В. С. ТАНАЕВ

ДЕКОМПОЗИЦИЯ И АГРЕГИРОВАНИЕ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

АКАДЕМИЯ НАУК БЕЛОРУССКОЙ ССР
ИНСТИТУТ ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

В. С. ТАНАЕВ

ДЕКОМПОЗИЦИЯ И АГРЕГИРОВАНИЕ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Под редакцией
члена-корреспондента АН БССР
А. Д. Закревского

МИНСК
«НАУКА И ТЕХНИКА»
1987

УДК 519.85

Танаев В. С. **Декомпозиция и агрегирование в задачах математического программирования** / Под ред. А. Д. Закревского.— Мн.: Наука и техника, 1987.— 183 с.

В книге излагаются основные подходы к построению процедур декомпозиции и агрегирования, предназначенных для решения большеразмерных задач математического программирования. При определенных условиях использование этих процедур позволяет получать решение исходной задачи в результате решения некоторой совокупности взаимосвязанных вспомогательных подзадач, каждая из которых в том или ином смысле проще исходной.

Рассчитана на научных сотрудников, инженеров и аспирантов, специализирующихся в области разработки и применения методов оптимизации, а также студентов старших курсов вузов соответствующих специальностей.

Ил. 27. Библиогр.— 36 назв.

Рецензенты:
В. А. Емеличев, д-р физ.-мат. наук,
Л. Ф. Верина, канд. физ.-мат. наук

$$T\frac{1502000000-160}{M316-87}46-87$$

ВВЕДЕНИЕ

Многие задачи оптимального проектирования, планирования, управления могут быть сформулированы в терминах математического программирования:

$$f(x) \to \min,$$

$$g_i(x) \leqslant 0, \quad i = \overline{1, m},$$

где $x = (x_1, x_2, \ldots, x_n) \in X \subseteq R^n$; R^n — n-мерное евклидово пространство; функции $f(x)$, $g_i(x)$, $i = \overline{1, m}$, определены на X и принимают действительные значения.

В зависимости от вида множества X и свойств функций $f(x)$ и $g_i(x)$, $i = \overline{1, m}$, можно выделить следующие классы задач математического программирования: линейное, квадратичное, выпуклое, целочисленное и т. д. В зависимости от сложности и степени изученности для каждого класса задач разработаны различные по эффективности и трудоемкости методы их решения. Сущность каждого из этих методов состоит в последовательном выполнении некоторой совокупности вычислений, часто включающих решение вспомогательных задач математического программирования. Вспомогательные задачи относятся либо к тому же классу задач, что и исходная, но содержат меньше переменных и(или) ограничений, либо к более простому классу задач.

Специфические особенности многих прикладных задач позволяют использовать при их решении такие эффективные приемы, как декомпозиция и агрегирование. В ряде случаев применение этих приемов дает возможность не только получать решение сложной большеразмерной задачи в результате решения некоторой совокупности более простых подзадач, но и добиваться определенной их независимости. Создаются реальные предпосылки для распараллеливания процесса решения задачи, что в свою очередь приводит к более полному использованию возможностей современной вычислительной техники (мультипроцессорные системы, сети ЭВМ и т. п.).

Пусть, например, исходная задача такова, что в результате перенумерации переменных и ограничений ее можно представить в виде

$$F(f_1(\tilde{x}_1), f_2(\tilde{x}_2), \ldots, f_\nu(\tilde{x}_\nu)) \to \min,$$

$$g_i(\tilde{x}_l) \leqslant 0, \quad i=\overline{m_{l-1}+1, m_l}, \quad l=\overline{1, \nu},$$

где $\tilde{x}_l \in X_l \subseteq R^{n_l}$; $f_l(\tilde{x}_l)$, $g_i(\tilde{x}_l)$, $i=\overline{m_{l-1}+1, m_l}$ — действительные функции, заданные на X_l, $l=\overline{1, \nu}$; F — действительная неубывающая функция ν переменных f_1, $f_2, \ldots, f_\nu$, $\sum\limits_{l=1}^{\nu} n_l = n$, $m_0=1$, $m_\nu=m$.

Нетрудно видеть, что эта задача допускает непосредственную декомпозицию (разложение) на ν независимых подзадач $(l=\overline{1, \nu})$:

$$f_l(\tilde{x}_l) \to \min,$$

$$g_i(\tilde{x}_l) \leqslant 0, \quad i=\overline{m_{l-1}+1, m_l},$$

$$\tilde{x}_l \subseteq X_l,$$

решение которых и составляет решение исходной задачи.

Аналогично пусть исходная задача представима в виде

$$F(f_1(\tilde{x}_1), f_2(\tilde{x}_2), \ldots, f_\nu(\tilde{x}_\nu)) \to \min,$$

$$g_i(f_1(\tilde{x}_1), f_2(\tilde{x}_2), \ldots, f_\nu(\tilde{x}_\nu)) \leqslant 0, \quad i=\overline{1, m},$$

где $\tilde{x}_l \in R^{n_l}$; $f_l(\tilde{x}_l)$ — действительная функция, $l=\overline{1, \nu}$; F, g_i, $i=\overline{1, m}$,— действительные функции ν переменных $f_1, f_2, \ldots, f_\nu$.

Приведенная задача допускает непосредственное агрегирование (объединение, сборку) переменных. Введем новые (агрегированные) переменные $y_1, y_2, \ldots, y_\nu$, полагая $y_l=f_l(\tilde{x}_l)$. Пусть $y_1^*, y_2^*, \ldots, y_\nu^*$ — решение агрегированной задачи

$$F(y_1, y_2, \ldots, y_\nu) \to \min;$$

$$g_i(y_1, y_2, \ldots, y_\nu) \leqslant 0, \quad i=\overline{1, m}.$$

Тогда решение исходной задачи может быть получено, очевидно, в результате решения ν независимых уравнений $(l=\overline{1, \nu})$:

$$f_l(\tilde{x}_l)=y_l^*.$$

Описанные ситуации являются, конечно, простейшими и на практике встречаются весьма редко. Однако они могут создаваться искусственно в процессе решения значительно более сложных практических задач. Следует отметить также, что формальное использование приемов декомпозиции и агрегирования часто допускает соответствующую содержательную интерпретацию в терминах решаемой прикладной задачи и тем самым позволяет более полно и всесторонне ее изучить.

Систематические исследования в области создания эффективных методов решения большеразмерных задач математического программирования проводятся в нашей стране и за рубежом с начала 60-х годов. Большое влияние на их развитие оказала бурно развивающаяся в последние годы математическая теория многоуровневых иерархических систем управления.

К настоящему времени сформировались различные направления исследований, разработаны новые подходы и методы, проведены обширные эксперименты, решен ряд сложных прикладных задач. Вместе с тем практика требует дальнейшего развития и совершенствования декомпозиционных методов, создания соответствующего программного обеспечения и расширения круга специалистов, владеющих этими методами и средствами.

В данной работе изложены современные подходы к созданию методов декомпозиции и агрегирования в задачах математического программирования. Основное внимание уделено выявлению исходных математических посылок, описанию характерных способов и наиболее распространенных схем декомпозиции и агрегирования.

В первых трех параграфах дается краткое описание известных методов минимизации функций одной и нескольких переменных, решения задач линейного и выпуклого программирования. В параграфе 4 рассматривается широко используемый на практике метод релаксации (ослабления) ограничений. Вопросам параметрической декомпозиции посвящены параграфы 5, 6. В последующих трех параграфах описывается подход

Дж. Данцига и Ф. Вулфа, а в параграфах 10, 11 — подход Дж. Бендерса. Основные положения теории двойственности и основанные на них схемы декомпозиции анализируются в параграфе 12. В параграфах 13, 14 рассматриваются вопросы агрегирования переменных в блочном программировании. Использованию метода возмущений при построении схем декомпозиции и агрегирования посвящен параграф 15.

Необходимые исходные сведения даются по ходу изложения материала. В книге используются обычные для математического программирования обозначения. За редким исключением не подчеркивается различие между вектор-строкой и вектор-столбцом.

Автор считает своим приятным долгом поблагодарить сотрудников лаборатории математических методов оптимизации Института технической кибернетики АН БССР за ряд полезных советов и замечаний. Большую помощь при подготовке рукописи к печати оказали С. Н. Воронкова и С. И. Крючкова. Автор искренне признателен А. Д. Закревскому, В. А. Емеличеву и Л. Ф. Вериной за доброжелательную критику и внимание к работе.

1. МЕТОДЫ БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ

В данном параграфе описываются наиболее распространенные методы минимизации функций одной и нескольких переменных без ограничений на значения переменных (за исключением, возможно, интервальных ограничений). Особое внимание уделяется методам минимизации недифференцируемых функций.

1.1. Выпуклые множества и функции. Множество X точек R^n называется *выпуклым,* если оно вместе с любыми своими точками $x^{(1)}$, $x^{(2)} \in X$ содержит и все точки вида

$$x = \alpha x^{(1)} + (1-\alpha) x^{(2)}, \quad 0 \leqslant \alpha \leqslant 1.$$

Геометрически это означает, что если концы некоторого отрезка принадлежат X, то и весь отрезок принадлежит X.

Из определения выпуклого множества непосредственно следует, что если точки $x^{(i)} \in X$, $i=\overline{1, r}$, то при всех $\alpha_i \geqslant 0$, $i=\overline{1, r}$, таких, что $\sum\limits_{i=1}^{r} \alpha_i = 1$, точка $x = \sum\limits_{i=1}^{r} \alpha_i x^{(i)}$ принадлежит X.

Пересечение любого числа выпуклых множеств выпукло.

Функция $f(x)$ является *выпуклой* на выпуклом множестве X, если

$$f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) \leqslant \alpha f(x^{(1)}) + (1-\alpha) f(x^{(2)})$$

для всех $x^{(1)}$, $x^{(2)} \in X$ и $0 \leqslant \alpha \leqslant 1$.

Отметим ряд свойств выпуклых на X функций.

1. Пусть $f_1(x)$ и $f_2(x)$ — выпуклые функции и c_1, c_2 — неотрицательные числа. Тогда $f(x) = c_1 f_1(x) + c_2 f_2(x)$ — выпуклая функция.

2. Если $f(x)$ — выпуклая функция, то $f(\alpha_1 x^{(1)} + \ldots + \alpha_r x^{(r)}) \leqslant \alpha_1 f(x^{(1)}) + \ldots + \alpha_r f(x^{(r)})$ для всех $x^{(i)} \in X$ и $\alpha_i \geqslant 0$, $i=\overline{1, r}$, при условии, что $\sum\limits_{i=1}^{r} \alpha_i = 1$.

3. Если $f_i(x)$, $i=\overline{1, r}$,— выпуклые функции, то $f(x) = \max\limits_{1 \leqslant i \leqslant r} f_i(x)$ выпуклая функция.

4. Функция $f(x)$ выпукла тогда и только тогда, когда при любых x и y из R^n выпукла функция одномерного аргумента $t: \varphi_{x,y}(t) \equiv f(x+ty), \quad x+ty \in X$.

5. Если $f(x)$ — выпуклая функция, то множество точек $x \in X$, удовлетворяющих неравенству $f(x) \leqslant c$, или пусто, или выпукло при любом числе c.

6. Если $f(x)$ — выпуклая функция, то она непрерывна во всех внутренних точках выпуклого множества X и имеет в каждой внутренней точке множества X производную по любому направлению.

7. Дифференцируемая функция $f(x)$ выпукла тогда и только тогда, когда $f(x^{(2)}) - f(x^{(1)} \geqslant (f'(x^{(1)}), x^{(2)} - x^{(1)})$ для любых $x^{(1)}, x^{(2)} \in X$. Здесь $f'(x)$ — градиент функции $f(x)$.

Выпуклая функция может иметь на выпуклом множестве X несколько локальных минимумов, каждый из которых глобальный. Множество точек минимума при этом является выпуклым.

Функция $f(x)$ — строго выпуклая на выпуклом множестве X, если

$$f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) < \alpha f(x^{(1)}) + (1-\alpha) f(x^{(2)})$$

для всех $x^{(1)}, x^{(2)} \in X, \quad x^{(1)} \neq x^{(2)}$ и $0 < \alpha < 1$.

Строго выпуклая функция имеет на выпуклом множестве X не более одного локального минимума, который является и глобальным.

Функция $f(x)$ называется *сильно выпуклой* на выпуклом множестве X, если существует такая константа $\rho > 0$, что для любых $x^{(1)}, x^{(2)} \in X$ и $0 < \alpha < 1$ выполняется неравенство

$$f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) \leqslant \alpha f(x^{(1)}) + (1-\alpha) f(x^{(2)}) -$$
$$- \alpha (1-\alpha) \rho \| x^{(1)} - x^{(2)} \|^2.$$

Если $f(x)$ — дважды непрерывно дифференцируемая функция, то условие сильной выпуклости на R^n эквивалентно условию

$$(f''(x) y, y) \geqslant m \|y\|^2$$

при любых x и y из R^n и $m = \text{const} > 0$. Здесь $f''(x)$ — матрица вторых частных производных функции $f(x)$.

Очевидно, сильно выпуклая функция является строго выпуклой, а следовательно, и выпуклой.

1.2. Одномерная минимизация. Рассмотрим задачу минимизации унимодальной функции $f(x)$ одной переменной x на интервале $[a, b]$.

Функция $f(x)$ называется *унимодальной* на заданном интервале $[a, b]$, если 1) она на этом интервале имеет единственную точку x^* минимума и 2) для любых двух точек $x_1 \in [a, b]$ и $x_2 \in [a, b]$ таких, что $x_1 < x_2$, из условия $x^* < x_1$ следует $f(x_1) < f(x_2)$, а из условия $x_2 < x^*$ — $f(x_1) > f(x_2)$.

Свойство унимодальности позволяет по результатам вычисления значений функции $f(x)$ в двух произвольных точках x_1 и x_2, $x_1 < x_2$, интервала $[a, b]$ указать интервал, в котором находится точка x^*, более узкий, чем исходный. Действительно, если $f(x_1) < f(x_2)$, то $x^* \in [a, x_2]$; если $f(x_1) > f(x_2)$, то $x^* \in [x_1, b]$; наконец, если $f(x_1) = f(x_2)$, то $x^* \in [x_1, x_2]$.

В дальнейшем, не нарушая общности, в качестве интервала $[a, b]$ будем рассматривать $[0, 1]$ (замена переменной $x' = (x-a)/(b-a)$).

Вычисление значения функции $f(x)$ в заранее выбранной точке $x \in [0, 1]$ обычно называют экспериментом.

В зависимости от характера проведения экспериментов поиск точки x^* минимума функции $f(x)$ на интервале $[0, 1]$ может быть пассивным или активным. В любом случае эффективность поиска определяется длиной получаемого в результате подынтервала интервала $[0, 1]$, содержащего точку x^* (так называемого интервала неопределенности).

При *пассивном* поиске эксперименты проводятся параллельно и независимо друг от друга. В этом случае целесообразно выполнять четное число экспериментов и осуществлять поиск так называемыми однородными парами. Если число экспериментов равно n (n — четное число), то оптимальная стратегия имеет вид

$$x_k = \frac{(1+\varepsilon)\left[(k+1)/2\right]}{n/2+1} - \left\{\left[\frac{k+1}{2}\right] - \left[\frac{k}{2}\right]\right\}\varepsilon,$$

$$k = \overline{1, n},$$

где ε — любое малое положительное число; $[c]$ — целая часть числа c.

Вычислив и сравнив значения функции $f(x)$ в точках x_k, $k = \overline{1, n}$, получим интервал неопределенности, длина которого не превышает величины $(1+\varepsilon)\Big/\left(\frac{n}{2}+1\right)$.

При *активном* поиске эксперименты проводятся последовательно с использованием на каждом шаге всей информации, полученной на предыдущих шагах. Наиболее распространенными методами активного поиска являются методы дихотомии, золотого сечения, Фибоначчи.

Используя *метод дихотомии*, на каждом шаге осуществляют два эксперимента: вычисление значений функции в двух точках, расположенных возможно ближе к середине полученного к данному шагу интервала неопределенности. По результатам экспериментов выделяется новый, более узкий интервал неопределенности и процесс повторяется. Если на каждом шаге выбирать пару точек, отстоящих от середины рассматриваемого на данном шаге интервала на $\varepsilon/2$ (справа и слева), то после $n/2$ шагов (n — общее число экспериментов) находим интервал неопределенности, длина которого не превышает величины $2^{-n/2}+(1-2^{-n/2})\varepsilon$.

Если заранее известно общее число n экспериментов, то наименьшую длину интервала неопределенности обеспечивает применение метода Фибоначчи. Введем в рассмотрение последовательность чисел Фибоначчи: $F_0=F_1=1$, $F_k=F_{k-1}+F_{k-2}$, $k=2, 3, \ldots$ Первый эксперимент проводится в точке, отстоящей от одного (безразлично какого) конца исходного единичного интервала на расстоянии

$$L_2=\frac{F_{n-2}+(-1)^n\varepsilon}{F_n},$$

где ε — любое малое положительное число.

Второй эксперимент выполняем в точке, симметричной предыдущей относительно середины интервала. В результате проведения экспериментов в исходном интервале выделяется подынтервал неопределенности, содержащий одну из двух указанных точек. Следующий эксперимент осуществляем в точке, симметричной оставшейся относительно середины подынтервала и т. д. После проведения n экспериментов получаем интервал неопределенности, длина которого не превышает величины $1/F_n+(F_{n-2}/F_n)\varepsilon$.

При нефиксированном числе n экспериментов, как уже отмечалось, может быть использован метод дихотомии.

Однако более предпочтителен *метод золотого сечения.* По этому методу первых два эксперимента проводятся в точках, отстоящих соответственно от правого и левого концов исходного единичного интервала на расстоянии $1/(1+\tau)$, где $\tau=\frac{1+\sqrt{5}}{2}\approx 1{,}618$. В результате проведения данных экспериментов выделяется подынтервал неопределенности, содержащий одну из этих точек. Следующий эксперимент выполняем в точке, симметричной оставшейся относительно середины подынтервала и т. д. После проведения n экспериментов получаем интервал неопределенности, длина которого не превышает величины $1/\tau^{n-1}$.

1.3. При минимизации одномерных функций особенно выпуклых широко распространены разнообразные *аппроксимационные методы,* в которых процесс минимизации функции $f(x)$ одной переменной x на интервале $[a, b]$ включает замену функции $f(x)$ некоторой аппроксимирующей функцией $\varphi(x)$ с последующей ее минимизацией на интервале $[a, b]$. Полученная в результате точка $\hat{x}$ минимума функции $\varphi(x)$ является приближенным решением исходной задачи. Если длина интервала $[a, b]$ мала и функция $\varphi(x)$ выбрана удачно, то погрешность этого решения, как правило, небольшая. Знание точки $\hat{x}$ обычно позволяет провести более точную аппроксимацию функции $f(x)$ и (или) сузить интервал поиска минимума. Выбор аппроксимирующих функций $\varphi(x)$ естественно ограничивается классом функций, минимизация которых не вызывает особых затруднений, во всяком случае менее трудоемка, чем непосредственная минимизация функции $f(x)$.

В *методе парабол* функция $f(x)$ аппроксимируется полиномом второй степени, график которого проходит через точки $(x_1, f(x_1))$, $(x_2, f(x_2))$, $(x_3, f(x_3))$, где $a\leqslant x_1<x_2<x_3\leqslant b$, $f(x_2)\leqslant \min\{f(x_1), f(x_3)\}$, $f(x_2)<\max\{f(x_1), f(x_3)\}$. Значение

$$\hat{x}=x_2-\frac{1}{2}\frac{(x_2-x_1)^2(f(x_2)-f(x_3))-(x_3-x_2)^2(f(x_2)-f(x_1))}{(x_2-x_1)(f(x_2)-f(x_3))-(x_3-x_2)(f(x_2)-f(x_1))}.$$

В *методе кубической аппроксимации* функция $f(x)$ на интервале $[a, b]$ аппроксимируется полиномом третьей степени. Предполагается, что $f(x)$ — дифференцируемая функция. Значение

$$\hat{x}=b-\frac{(f'(b)+B-A)(b-a)}{f'(a)+f'(b)+2B},$$

где

$$B=\frac{3(f(a)-f(b))}{b-a}+f'(a)+f'(b);$$

$$A=\sqrt{B^2-f'(a)f'(b)}.$$

В *методе касательных* функция $f(x)$ аппроксимируется кусочно-линейной функцией, составленной из «кусков» касательных к функции $f(x)$ в последовательно выбираемых точках интервала $[a, b]$. Предполагается, что функция $f(x)$ непрерывно дифференцируема на интервале $[a, b]$.

Введем в рассмотрение функцию $\Phi(x, y)$ двух переменных, полагая $\Phi(x, y)=f(y)+f'(y)(x-y)$. Выбирая точки $x_1, x_2, \ldots, x_k$ в интервале $[a, b]$, определяем функцию $\varphi_k(x)=\max\{\Phi(x, x_1), \Phi(x, x_2), \ldots, \Phi(x, x_k)\}$. Если x_{k+1} — точка минимума этой функции на интервале $[a, b]$, то x_{k+1} можно выбрать в качестве приближенного решения $\hat{x}$ исходной задачи либо добавить точку x_{k+1} к точкам $x_1, x_2, \ldots, x_k$ и перейти к функции $\varphi_{k+1}(x)$, аппроксимирующей функцию $f(x)$ с большей точностью.

1.4. При минимизации одномерных функций часто используется *способ удвоения шагов*. Выбираются точка x_1 и число $\alpha_1>0$. Если $f(x_1)>f(x_1+\alpha_1)>f(x_1+2\alpha_1)>\ldots>$ $>f(x_1+2^p\alpha_1)\leqslant f(x_1+2^{p+1}\alpha_1)$, то в качестве очередной точки x_2 выбирается точка $x_1+2^p\alpha_1$; если $f(x_1)\leqslant f(x_1+$ $+\alpha_1)$, $f(x_1)\leqslant f(x_1+\alpha_1/2)$, $\ldots$, $f(x_1)\leqslant f(x_1+\alpha_1/2^{p-1})$, $f(x_1)>f(x_1+\alpha_1/2^p)$, то выбирается точка $x_1+\alpha_1/2^p$. Аналогичные действия повторяются для точки x_2 и некоторого числа $\alpha_2>0$ и т. д. При минимизации функции $f(x)$ на интервале необходимо следить за тем, чтобы не выйти за границы рассматриваемого интервала.

1.5. Минимизация функций нескольких переменных. Рассмотрим задачу минимизации функции $f(x)=f(x_1, x_2, \ldots, x_n)$, $x=(x_1, x_2, \ldots, x_n)\in R^n$, нескольких переменных без ограничений на их значения.

Методы нулевого порядка позволяют проводить минимизацию функции $f(x)$ без нахождения ее производных.

Метод покоординатного спуска является простейшим методом нулевого порядка и имеет большое число различных модификаций. Согласно этому методу, выбирается произвольная точка $x^{(0)}$ и вычисляется значение $f(x^{(0)})$. Затем последовательно, начиная с первой, изменяется одна из координат точки $x^{(0)}$ при неизменных остальных координатах до тех пор, пока не будет получена точка $x^{(1)}$ такая, что $f(x^{(1)})<f(x^{(0)})$, либо установлена оптимальность точки $x^{(0)}$. Процедура повторяется для точки $x^{(1)}$ и т. д.

Изменение значения переменной осуществляется различными способами: от увеличения и уменьшения его на одну и ту же величину до решения задачи минимизации одномерной функции, получаемой из функции $f(x)$ путем закрепления всех переменных, кроме рассматриваемой.

В *методе прямого поиска* на каждом шаге производится так называемый исследующий поиск и поиск по образцу.

Исследующий поиск в точке x по заданному n-мерному вектору Δx заключается в нахождении такой точки $\bar{x}$, что $f(\bar{x}_1, \ldots, \bar{x}_k, x_{k+1}, \ldots, x_n) = \min \{f(\bar{x}_1, \ldots, \bar{x}_{k-1}, x_k, \ldots, x_n), f(\bar{x}_1, \ldots, \bar{x}_{k-1}, x_k+\Delta x_k, x_{k+1}, \ldots, x_n), f(\bar{x}_1, \ldots, \bar{x}_{k-1}, x_k-\Delta x_k, x_{k+1}, \ldots, x_n)\}$. Результат исследующего поиска считается успешным, если $f(\bar{x})<f(x)$.

Поиск по образцу заключается в нахождении точки $\tilde{x}$, доставляющей минимум функции $f(x)$ вдоль луча, исходящего из заданной точки x и проходящего через точку $\bar{x}$, т. е. в минимизации одномерной функции $f(x+\alpha(\bar{x}-x))$ при $\alpha \geqslant 0$. Результат поиска по образцу считается успешным, если успешным оказывается результат исследующего поиска в точке $\tilde{x}$ при некотором выбранном Δx.

Используя введенные понятия, метод прямого поиска может быть описан следующим образом. Выбираются некоторая начальная точка $x^{(0)}$, вектор $\Delta x^{(0)}$ и осуществляется исследующий поиск в точке $x^{(0)}$ по заданному $\Delta x^{(0)}$. Если результат такого поиска оказывается неуспешным, то изменяется $\Delta x^{(0)}$ и процесс повторяется. Пусть найдена точка $\bar{x}^{(0)}$ (значение $f(\bar{x}^{(0)})<f(x^{(0)})$). Выбирая точку $\bar{x}^{(0)}$ в качестве $x^{(0)}$, можно повторить исследующий поиск либо перейти к поиску по образцу — минимизации одномерной функции $f(x^{(0)}+\alpha(\bar{x}^{(0)}-x^{(0)}))$ при $\alpha \geqslant 0$. При этом используется обычно способ удвоения шагов. В результате поиска по образцу находится точка $\tilde{x}^{(0)}$, в которой в свою очередь осуществляется исследующий поиск. Если будет найдена точка x' такая, что $f(x')<f(\tilde{x}^{(0)})$ (т. е.

поиск по образцу оказывается успешным), то точка x' выбирается в качестве точки $x^{(1)}$ и процесс повторяется. В противном случае производится возврат в точку $x^{(0)}$ и организуется исследующий поиск в точке $x^{(0)}$ с новым вектором $\Delta x^{(0)}$.

Метод поиска по деформируемому многограннику состоит в следующем. Выбирается $n+1$ точка $x^{(1)}, x^{(2)}, \dots, x^{(n+1)}$ и в них вычисляются значения функции $f(x)$. Пусть

$$f(x^{(l)}) = \min \{f(x^{(i)}) \mid i = \overline{1, n+1}\}$$

и

$$f(x^{(h)}) = \max \{f(x^{(i)}) \mid i = \overline{1, n+1}\}.$$

Находится центр тяжести всех точек, кроме точки $x^{(h)}$:

$$x^{(c)} = \frac{1}{n} \left(\sum_{i=1}^{n+1} x^{(i)} - x^{(h)} \right)$$

и осуществляется минимизация функции $f(x)$ вдоль луча, исходящего из точки $x^{(h)}$ и проходящего через точку $x^{(c)}$. Любую точку, лежащую на этом луче, можно представить в виде $x^{\alpha} = x^{(h)} + \alpha(x^{(c)} - x^{(h)})$, $\alpha \geqslant 0$.

Выбираются три числа α_1, α_2 и α_3 так, что $0 < \alpha_3 < 1 < \alpha_1 < \alpha_2$. Если $f(x^{\alpha_1}) \leqslant f(x^{(l)})$ и $f(x^{\alpha_2}) < f(x^{(l)})$, то точка $x^{(h)}$ заменяется x^{α_2}. Если $f(x^{\alpha_1}) > f(x^{(i)})$ для всех i, $1 \leqslant i \leqslant n+1$, кроме $i = h$, то точка $x^{(h)}$ заменяется x^{α_3}. Если $f(x^{\alpha_1}) > f(x^{(h)})$, то каждая точка $x^{(i)}$ заменяется $x^{(l)} + \frac{1}{2}(x^{(i)} - x^{(l)})$, $i = \overline{1, n+1}$. В остальных случаях точка $x^{(h)}$ заменяется x^{α_1} (без изменения остальных точек). Описанный процесс повторяется для вновь полученного набора точек и т. д.

В качестве критерия окончания поиска может быть выбрано условие

$$\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - f(x^{(c)})]^2 \leqslant \varepsilon^2,$$

где ε — достаточно малое число.

Метод поиска с пошаговым изменением системы координат отличается от метода покоординатного спуска тем, что при переходе от точки $x^{(k)}$ к очередной точке

$x^{(k+1)}$ выбирается новая система координат, получаемая по результатам поиска точки $x^{(k+1)}$.

Поиск начинается с произвольной точки $x^{(0)}$ с использованием исходной системы координат и некоторого начального набора чисел $\alpha_1, \alpha_2, \ldots, \alpha_n$. Обозначим через $s^{(1)}, s^{(2)}, \ldots, s^{(n)}$ ортонормированные n-мерные векторы, определяющие текущую систему координат. На первом шаге это единичные векторы, параллельные осям координат.

Если $f(x^{(0)}+\alpha_1 s^{(1)}) \leqslant f(x^{(0)})$, то в качестве точки $x^{(0)}$ выбирается $x^{(0)}+\alpha_1 s^{(1)}$ и число α_1 заменяется $\beta\alpha_1$, $\beta>0$ (обычно $\beta=3$). Если $f(x^{(0)}+\alpha_1 s^{(1)})>f(x^{(0)})$, то число α_1 заменяется $\gamma\alpha_1$, $\gamma<0$ (обычно $\gamma=-0{,}5$). Затем осуществляются аналогичные действия с использованием направления $s^{(2)}$ и т. д. до тех пор, пока не будут рассмотрены все n направлений. В результате будут получены некоторая новая точка $x^{(0)}$ и новый набор чисел $\alpha_1, \alpha_2, \ldots, \alpha_n$. Процесс повторяется до тех пор, пока поиск по каждому из направлений не будет приводить к возрастанию функции $f(x)$. Последняя из точек $x^{(0)}$ выбирается в качестве точки $x^{(1)}$, а последний из наборов $\alpha_1, \alpha_2, \ldots, \alpha_n$ — в качестве исходного для организации поиска из точки $x^{(1)}$.

Характерная особенность описываемого метода состоит в том, что при организации поиска из каждой очередной точки $x^{(k+1)}$ система ортонормированных векторов $s^{(1)}, s^{(2)}, \ldots, s^{(n)}$, используемая на предыдущем шаге, заменяется некоторой новой системой. В качестве $s^{(1)}$ выбирается единичный вектор в направлении $(x^{(k+1)}-x^{(k)})$, а в качестве $s^{(2)}, \ldots, s^{(n)}$ — единичные взаимно ортогональные векторы, ортогональные вектору $s^{(1)}$. Процедура их нахождения может быть описана следующим образом. Пусть $s^{(1)}, s^{(2)}, \ldots, s^{(n)}$ — система ортонормированных векторов, используемая на k-м шаге поиска (с начальной точкой $x^{(k)}$), и $x^{(k+1)}$ — результирующая точка этого поиска. Обозначим через δ_i длину смещения точки $x^{(k+1)}$ относительно $x^{(k)}$ в направлении $s^{(i)}$, $i=\overline{1, n}$. Введем в рассмотрение векторы

$$a^{(i)}=\sum_{k=i}^{n} \delta_k s^{(k)}.$$

Положим $\bar{s}^{(1)}=a^{(1)}/\|a^{(1)}\|$, $\bar{s}^{(i)}=b^{(i)}/\|b^{(i)}\|$, где $b^{(i)}= =a^{(i)}-\sum\limits_{k=1}^{i-1}(a^{(i)}, \bar{s}^{(k)})\bar{s}^{(k)}$, $i=\overline{2, n}$, $\|c\|$ — норма вектора c.

Векторы $\bar{s}^{(1)}, \bar{s}^{(2)}, \ldots, \bar{s}^{(n)}$ образуют искомую систему ортонормированных векторов, используемую на $(k+1)$-м шаге поиска (с начальной точкой $x^{(k+1)}$).

1.6. Рассмотрим некоторые наиболее распространенные *методы первого порядка* минимизации функции $f(x)$ нескольких переменных без ограничений на значение переменных. Предполагается, что $f(x)$ — дифференцируемая функция и имеется возможность вычисления значений ее первых частных производных в любой заданной точке x. Каждый из описываемых ниже методов строит последовательность точек $x^{(0)}, x^{(1)}, \ldots$, связанных рекуррентным соотношением вида

$$x^{(k+1)} = x^{(k)} + \alpha_k v^{(k)} \tag{1}$$

и удовлетворяющих условию релаксации

$$f(x^{(k+1)}) < f(x^{(k)}), \quad k=0, 1, \ldots$$

Для нахождения направления $v^{(k)}$ спуска в этих методах необходимо знание значений первых частных производных (а иногда и значения функции) в точке $x^{(k)}$. Длина шага $\alpha_k > 0$ определяется обычно в результате использования того или иного метода одномерной минимизации.

При определенных условиях последовательность точек сходится к искомой оптимальной точке x^*.

В *методе градиентного спуска* в качестве направления $v^{(k)}$ спуска выбирается антиградиент $f'(x^{(k)})$ функции $f(x)$ в точке $x^{(k)}$. Соотношение (1) принимает вид

$$x^{(k+1)} = x^{(k)} - \alpha_k f'(x^{(k)}).$$

Если длина шага α_k выбирается так, чтобы соблюдалось условие

$$f(x^{(k)} - \alpha_k f'(x^{(k)})) = \min \{f(x^{(k)} - \alpha f'(x^{(k)})) \mid \alpha \geqslant 0\}, \tag{2}$$

то говорят, что используется *метод наискорейшего спуска.*

В *методе градиентного спуска с дроблением шага* длина шага α_k выбирается согласно условию

$$f(x^{(k)} - \alpha_k f'(x^{(k)})) - f(x^{(k)}) \leqslant -\varepsilon \alpha_k \| f'(x^{(k)}) \|^2,$$

где $0 < \varepsilon < 1$ — произвольно выбранная константа.

В частности, если градиент $f'(x)$ удовлетворяет условию Липшица (существует константа ρ такая, что неравенство $\|f'(x)-f'(y)\| \leqslant \rho\|x-y\|$ имеет место для любых точек x и y из R^n), то на каждом шаге k в качестве α_k можно выбирать одну и ту же величину $\alpha=(1-\varepsilon)/\rho$.

Если функция $f(x)$ ограничена снизу, а ее градиент $f'(x)$ удовлетворяет условию Липшица, то $\|f'(x^{(k)})\| \to 0$ при $k\to\infty$ независимо от выбора начальной точки $x^{(0)}$.

При минимизации сильно выпуклых функций градиентные методы сходятся со скоростью геометрической прогрессии: $\|x^{(k)}-x^*\| \leqslant Cq^k\|x^{(0)}-x^*\|$, где $0<q<1$, $C<\infty$; x^* — точка минимума функции $f(x)$. В частности, если $f(x)$ — дважды непрерывно дифференцируемая функция, матрица $f''(x)$ вторых частных производных которой удовлетворяет неравенствам

$$m\|y\|^2 \leqslant (f''(x)y,\, y) \leqslant M\|y\|^2,\quad M \geqslant m > 0$$

при любых x и y, то при использовании метода наискорейшего спуска знаменатель $q=\dfrac{M-m}{M+m}$.

При минимизации так называемых овражных функций широкое распространение получил *прием изменения масштабов переменных.*

Вводя соответствующую замену переменных вида $x_j = \mu_j y_j$, исходная функция $f(x)$ заменяется новой функцией $\tilde{f}(y)$ с менее выраженной овражностью. Если функция $f(x)$ является достаточно гладкой, то в качестве μ_j можно выбрать минимум функции $\left(\dfrac{\partial^2 f(x)}{\partial x_j^2}\right)^{-1/2}$ по x_j при $x_1=\ldots=x_{j-1}=x_{j+1}=\ldots=x_n=0$.

Масштабирование переменных в общем случае приводит к итерационному процессу вида

$$x^{(k+1)}=x^{(k)}-\alpha_k B_k f'(x^{(k)}),$$

где B_k — некоторая матрица, зависящая, вообще говоря, от шага k. Известны различные подходы к формированию матриц B_k.

Так, в *методе Давидона — Флетчера — Пауэлла* матрицы B_k вычисляются согласно следующему рекуррентному соотношению:

$$B_{k+1}=B_k+\frac{[\Delta_k x]\,[\Delta_k x]^\tau}{(\Delta_k x,\, \Delta_k f')}-\frac{[b_k]\,[b_k]^\tau}{(\Delta_k f',\, b_k)},$$

где векторы $\Delta_k x = x^{(k+1)} - x^{(k)}$, $\Delta_k f' = f'(x^{(k+1)}) - f'(x^{(k)})$, $b_k = B_k \Delta_k f'$; B_0 — единичная матрица. Если a — n-мерный вектор, то $[a]$ — квадратная матрица размерности n, в первом столбце которой расположены компоненты вектора a, в остальных — нули; τ — знак транспонирования.

В *проективном методе Ньютона — Рафсона*

$$B_{k+1} = B_k - \frac{[b_k][b_k]^\tau}{(\Delta_k f', b_k)}.$$

В *методе Бройдена*

$$B_{k+1} = B_k + \frac{[\Delta_k x - b_k][\Delta_k x - b_k]^\tau}{(\Delta_k x - b_k, \Delta_k f')}.$$

В *методе Пирсона*

$$B_{k+1} = B_k + \frac{[\Delta_k x - b_k][\Delta_k x]^\tau}{(\Delta_k x, \Delta_k f')}.$$

Перечисленные методы относятся к обширной группе *методов переменной метрики.*

Длина шага α_k в каждом из этих методов обычно выбирается исходя из соотношения (2). Часто используется прием восстановления матрицы B_k (через некоторое число шагов, скажем k_0, в качестве матрицы B_k выбирают матрицу B_0, т. е. $B_{l(k_0+1)} = B_0$, $l = 1, 2, \ldots$).

В *методе сопряженных направлений* направление $v^{(k)}$ спуска определяется соотношениями

$$v^{(0)} = -f'(x^{(0)}), \quad v^{(k)} = -f'(x^{(k)}) + \beta_k v^{(k-1)}, \quad k = 1, 2, \ldots,$$

а длина шага α_k выбирается согласно условию (2).

Различные варианты приведенного метода отличаются друг от друга в основном способом выбора параметров β_k.

В частности, если все $\beta_k = 0$, то получаем метод наискорейшего спуска. Если все $\beta_k = \beta > 0$, то говорят, что применяется *метод тяжелого шарика.*

В *методе Флетчера — Ривса*

$$\beta_k = \frac{(\Delta_k f', \Delta_k f')}{(\Delta_{k-1} f', \Delta_{k-1} f')}.$$

В *методе Полака — Рибьера*

$$\beta_k = \frac{(\Delta_k f' - \Delta_{k-1} f', \Delta_k f')}{(\Delta_{k-1} f', \Delta_{k-1} f')}.$$

При использовании последних двух методов обычно задается предельно допустимое значение β и если вычисленные по указанным формулам значения β_k оказываются больше β, то полагают $\beta_k=\beta$.

Наряду с перечисленными известны и другие варианты метода сопряженных направлений.

На практике широко используется прием так называемого обновления метода: после выполнения $k=ln$ шагов полагают $\beta_{k+1}=0$ и продолжают процесс вычислений, $l=1, 2, \ldots$

Скорость сходимости различных вариантов метода сопряженных направлений для сильно выпуклых функций оказывается не ниже сверхлинейной. В частности, метод Флетчера — Ривса обеспечивает квадратичную скорость сходимости на каждые n итераций $(\|x^{((k+1)n)}-x^*\|\leqslant \leqslant C\|x^{(kn)}-x^*\|^2$, $C=\text{const}>0$, $k\geqslant N=\text{const})$.

1.7. Наиболее известным *методом второго порядка* минимизации функции $f(x)$ нескольких переменных без ограничений на значения переменных является *метод Ньютона.* При его использовании предполагаем, что функция $f(x)$ достаточно гладкая (существуют вторые частные производные).

Метод Ньютона основан на аппроксимации функции $f(x)$ в окрестности точки $x^{(k)}$ квадратичной функцией с последующей ее минимизацией. Начиная с произвольной точки $x^{(0)}$, строится последовательность точек $x^{(1)}$, $x^{(2)}, \ldots$, удовлетворяющих рекуррентному соотношению

$$x^{(k+1)}=x^{(k)}-[f''(x^{(k)})]^{-1}f'(x^{(k)}),$$

где $f''(x^{(k)})$ — матрица вторых частных производных функции $f(x)$, вычисленных в точке $x^{(k)}$; вектор-столбец $f'(x^{(k)})$ — градиент функции $f(x)$ в точке $x^{(k)}$.

Если $f(x)$ — дважды непрерывно дифференцируемая функция и $f''(x^*)$ — невырожденная матрица (x^* — точка минимума функции $f(x)$), то существует такая окрестность точки x^*, что для любой точки $x^{(0)}$ из этой окрестности последовательность точек $x^{(1)}$, $x^{(2)}, \ldots$, построенная в соответствии с указанным соотношением, сходится к точке x^* с квадратичной скоростью: $\|x^{(k+1)}-x^*\|\leqslant \leqslant C\|x^{(k)}-x^*\|^2$, где C — некоторая неотрицательная константа.

Из-за затруднений, связанных с выбором начального приближения $x^{(0)}$, обычно используют *метод Ньютона* *с регулировкой шага*

$$x^{(k+1)}=x^{(k)}-\alpha_k[f''(x^{(k)})]^{-1}f'(x^{(k)}). \qquad (3)$$

Длина шага $\alpha_k \geqslant 0$ выбирается либо из условия

$$f(x^{(k)}-\alpha_k[f''(x^{(k)})]^{-1}f'(x^{(k)}))=$$
$$=\min\{f(x^{(k)}-\alpha[f''(x^{(k)})]^{-1}f'(x^{(k)}))\,|\,\alpha\geqslant 0\},$$

либо из условия

$$f(x^{(k)}-\alpha_k[f''(x^{(k)})]^{-1}f'(x^{(k)}))-f(x^{(k)})\leqslant$$
$$\leqslant -\varepsilon\alpha_k(f'(x^{(k)}),\ [f''(x^{(k)})]^{-1}f'(x^{(k)})),$$

где $0<\varepsilon<1/2$ — некоторое число.

Если используется второе неравенство и оно выполняется при $\alpha_k=1$, то длина шага принимается равной единице, в противном случае дробится до тех пор, пока это неравенство не выполнится.

Сходимость последовательности точек $x^{(0)}, x^{(1)}, \ldots$, построенной в соответствии с соотношением (3) и указанными способами выбора длины шага, не зависит от выбора начальной точки $x^{(0)}$ при условии, если $f(x)$ — дважды непрерывно дифференцируемая функция и существуют числа m, M и ρ такие, что $m\|y\|^2\leqslant(f''(x)y, y)\leqslant M\|y\|^2$ и $\|f''(x)-f''(y)\|\leqslant\rho\|x-y\|$ для любых x и y. При этом сохраняется квадратичная скорость сходимости с константой $C=\rho/m$.

Затруднения, связанные с вычислением матриц $f''(x^{(k)})$ и их обращением, а также с возможным нарушением положительной определенности матрицы $f''(x)$, привели к разработке различных модификаций метода Ньютона.

Так, в *методе Ньютона с постоянной матрицей вторых производных* матрица $f''(x)$ вычисляется и обращается только в начальной точке $x^{(0)}$ и далее остается неизменной: в соотношении (3) $f''(x^{(k)})=f''(x^{(0)})$, $k=1, 2, \ldots$ Это, очевидно, один из вариантов метода градиентного спуска с изменением масштабов переменных.

В *методе Ньютона с обновлением матрицы вторых производных* матрица $f''(x)$ вычисляется и обращается в точках $x^{(0)}, x^{(l+1)}, \ldots, x^{(2l+1)}, \ldots$, где l — некоторое целое положительное число. В соотношении (3) $f''(x^{(k)})=f''(x^{(0)})$, $k=1, 2, \ldots, l$, $f''(x^{(k)})=f''(x^{(l+1)})$, $k=l+1, \ldots, 2l$, и т. д.

Если матрица вторых производных не является положительно определенной (и, как следствие, последовательность точек, удовлетворяющая соотношению (3), не обязательно сходится), то используются приемы ее коррекции. В частности, на каждом шаге k к матрице $f''(x^{(k)})$ добавляется матрица $\lambda_k E$, где E — единичная матрица; числа λ_k выбираются из условия обеспечения сходимости процесса.

1.8. Многие методы декомпозиции задач математического программирования приводят к необходимости минимизации *недифференцируемых функций*. Для их минимизации могут быть использованы описанные выше методы нулевого порядка. Однако для выпуклых функций существуют более эффективные методы.

Пусть $f(x)$ — выпуклая функция, заданная на (выпуклом) множестве $X \subseteq R^n$, и $\hat{x}$ — внутренняя точка X.

Вектор $\gamma_f(\hat{x}) \in R^n$, удовлетворяющий соотношению

$$f(x) - f(\hat{x}) \geqslant (\gamma_f(\hat{x}),\ x - \hat{x})$$

для всех $x \in X$, называется *субградиентом* или *обобщенным градиентом* функции $f(x)$ в точке $\hat{x}$.

Множество $\Gamma_f(\hat{x})$ обобщенных градиентов функции $f(x)$ в точке $\hat{x}$ является непустым, ограниченным, выпуклым и замкнутым.

Как уже отмечалось, в любой внутренней точке $\hat{x} \in X$ выпуклая функция $f(x)$ имеет производную $f'_v(\hat{x})$ по любому направлению $v \in R^n$.

Имеет место соотношение

$$f'_v(\hat{x}) = \max\{(\gamma_f(\hat{x}),\ v) \mid \gamma_f(\hat{x}) \in \Gamma_f(\hat{x})\}.$$

Если $f(x)$ — дифференцируемая в точке $\hat{x}$ функция, то ее обобщенный градиент совпадает с градиентом. Напомним, что выпуклая функция почти везде дифференцируема. Следовательно, за исключением точек множества меры нуль, ее обобщенный градиент совпадает с градиентом. При этом градиент непрерывен на множестве точек, в которых он определен.

Направление наискорейшего спуска в точке $\hat{x}$ задается вектором $-v$, $v \in \Gamma_j(\hat{x})$, лежащим на кратчайшем расстоянии от начала координат. Если $f(x)$ — дифференцируемая функция, то это ее антиградиент.

Для того чтобы точка $\hat{x} \in X$ была точкой минимума функции $f(x)$, необходимо и достаточно, чтобы множество $\Gamma_f(\hat{x})$ содержало нулевой вектор.

Следующие утверждения позволяют легко находить обобщенные градиенты большинства из рассматриваемых в дальнейшем функций.

Пусть $f_i(x)$, $i=\overline{1, r}$,— выпуклые функции; c_i, $i=\overline{1, r}$,— неотрицательные числа. Тогда функция $f(x)=\sum_{i=1}^{r} c_i f_i(x)$ — выпуклая и множество $\Gamma_f(\hat{x})$ ее обобщенных градиентов в точке $\hat{x}$ состоит из всевозможных векторов вида $\gamma_f(\hat{x})=\sum_{i=1}^{r} c_i \gamma_{f_i}(\hat{x})$, где $\gamma_{f_i}(\hat{x}) \in \Gamma_{f_i}(\hat{x})$.

Аналогично функция $f(x)=\max\limits_{1 \leqslant i \leqslant r} f_i(x)$ — выпуклая и множество $\Gamma_f(\hat{x})$ ее обобщенных градиентов в точке $\hat{x}$ содержит обобщенные градиенты $\gamma_{f_i}(\hat{x})$ тех функций $f_i(x)$, для которых $f_i(\hat{x})=f(\hat{x})$.

1.9. Для минимизации выпуклых недифференцируемых функций используются методы, внешне похожие на методы градиентного спуска. В этих методах роль градиента играет обобщенный градиент, а для обеспечения сходимости итерационного процесса применяются специфические приемы регулировки длины шага. Получаемая в результате последовательность точек $x^{(0)}, x^{(1)}, \ldots$ не является, вообще говоря, релаксационной, т. е. необязательно $f(x^{(k+1)}) \leqslant f(x^{(k)})$, $k=0, 1, \ldots$.

В *методе обобщенного градиентного спуска*, начиная с некоторой начальной точки $x^{(0)}$, строится последовательность точек $x^{(1)}, x^{(2)}, \ldots$, удовлетворяющих следующему рекуррентному соотношению:

$$x^{(k+1)}=x^{(k)}-\alpha_k \frac{\gamma_f(x^{(k)})}{\|\gamma_f(x^{(k)})\|}$$

где $\gamma_f(x^{(k)})$ — обобщенный градиент минимизируемой функции $f(x)$ в точке $x^{(k)}$; $\|c\|$ — норма вектора c.

Последовательность положительных чисел α_k — длин шагов — выбирается так, чтобы

$$\lim_{k\to\infty} \alpha_k=0, \quad \sum_{k=0}^{\infty} \alpha_k=\infty.$$

Если $f(x)$ — выпуклая функция, определенная в R^n, область X^* минимумов которой ограничена, то при любом начальном приближении $x^{(0)}$ построенная указанным образом последовательность точек $x^{(1)}, x^{(2)}, \ldots$ обладает одним из следующих свойств: либо найдется такое $k=\bar{k}$, что $x^{(\bar{k})} \in X^*$, либо

$$\lim_{k\to\infty} \min_{x\in X^*} \|x^{(k)}-x\|=0 \quad \text{и} \quad \lim_{k\to\infty} f(x^{(k)}) = \min_{x\in R^n} f(x).$$

Аналогичным свойством обладает и последовательность точек $x^{(0)}, x^{(1)}, \ldots$, построенная по формуле

$$x^{(k+1)} = \begin{cases} x^{(k)}-\alpha_k \gamma_f(x^{(k)}) & \text{при} \quad \alpha_k\|\gamma_f(x^{(k)})\| \leqslant C, \\ x^{(0)} & \text{в противном случае,} \end{cases}$$

где $C>0$ — некоторая константа.

Если последовательность точек $x^{(0)}, x^{(1)}, \ldots$ построить по формуле $x^{(k+1)}=x^{(k)}-\alpha_k\gamma_f(x^{(k)})$, то она будет сходиться в том и только в том случае, когда последовательность обобщенных градиентов $\gamma_f(x^{(0)}), \gamma_f(x^{(1)}), \ldots$ ограничена.

Отметим, что если $f(x)$ — выпуклая кусочно-линейная функция с конечным числом «кусков», то указанная последовательность обобщенных градиентов всегда ограничена. В общем случае для любого начального приближения $x^{(0)}$ можно подобрать такое число $\delta>0$, что если $\alpha_k \leqslant \delta$, $k=0, 1, \ldots$, то последовательность $\gamma_f(x^{(k)})$, $k=0, 1, \ldots$, ограничена.

Наряду с перечисленными известны и другие варианты метода обобщенного градиентного спуска.

Стохастическим аналогом метода обобщенного градиентного спуска является *метод обобщенных стохастических градиентов*, относящийся к классу методов случайного поиска.

Пусть $f(x)$ — выпуклая функция с единственной точкой x^* минимума в R^n.

Начиная с произвольной точки $x^{(0)}$, построим последовательность точек $x^{(1)}, x^{(2)}, \ldots$, удовлетворяющих рекуррентному соотношению

$$x^{(k+1)}=x^{(k)}-\alpha_k\gamma_f^{\omega}(x^{(k)}),$$

где $\gamma_f^{\omega}(x^{(k)})$ — случайный вектор, математическое ожидание которого совпадает с обобщенным градиентом $\gamma_f(x^{(k)})$ функции $f(x)$ в точке $x^{(k)}$.

Если длины шагов $\alpha_k > 0$ выбирать так, чтобы

$$\sum_{k=0}^{\infty} \alpha_k = \infty \text{ и } \sum_{k=0}^{\infty} \alpha_k^2 < \infty,$$

то последовательность точек $x^{(0)}$, $x^{(1)}$, ... с вероятностью единица будет сходиться к точке x^* при условии, что математическое ожидание величины $\|\gamma_f^\omega(x^{(k)})\|$ не превосходит некоторой константы для всех $k=0, 1, \ldots$

1.10. При минимизации недифференцируемых функций (так же как и дифференцируемых) широко используется прием изменения метрики пространства в процессе построения последовательных приближений. Хорошие результаты получаются при растяжении пространства в направлении обобщенного градиента, разности двух последовательных градиентов и некоторых других линейных неортогональных преобразованиях пространства аргументов.

Приведем краткое описание *метода обобщенного градиентного спуска с растяжением пространства* в направлении градиента.

Пусть задан вектор $v \in R^n$ с $\|v\|=1$. Каждый вектор $x \in R^n$ однозначно представим в виде

$$x = \mu v + z, \quad (v, z) = 0,$$

где $\mu = \mu(x, v)$ — некоторое число; $z = z(x, v)$ — некоторый вектор из R^n, ортогональный вектору v.

Оператором растяжения пространства R^n в направлении v с коэффициентом $\beta \geqslant 0$ называется оператор P, действующий на $x = \mu v + z$ следующим образом:

$$P(x) = \beta\mu v + z = (\beta - 1)(x, v) v + x = (E + (\beta - 1) v v^\tau) x.$$

Здесь представлено несколько эквивалентных форм записи оператора P. Символом E обозначена единичная матрица размерности $n \times n$; τ — знак транспонирования.

В описываемом методе минимизации выпуклой функции $f(x)$ движение в направлении обобщенного градиента на каждом шаге сочетается с операцией растяжения пространства аргументов в этом же направлении.

Пусть после выполнения k шагов найдена точка $x^{(k)}$ (в качестве $x^{(0)}$ может быть выбрана произвольная точка). Определяется обобщенный градиент $\gamma_f(x^{(k)})$ функции $f(x)$ в точке $x^{(k)}$. Если $\gamma_f(x^{(k)}) = 0$, то $x^{(k)}$ — искомая точка минимума и вычисления прекращаются. В противном случае осуществляется линейное преобразование исходного пространства:

$$y=B_k^{-1}x, \quad \varphi_k(y)=f(B_k y),$$

где B_k — некоторый, известный к данному шагу оператор (неособая матрица).

В новом пространстве реализуется шаг обобщенного градиентного спуска для функции $\varphi_k(y)$ исходя из точки $y^{(k)}=B_k^{-1}x^{(k)}$, т. е. находится

$$y^{(k+1)}=y^{(k)}-\alpha_k \frac{\tilde{\gamma}_k}{\|\tilde{\gamma}_k\|} \tag{4}$$

($\tilde{\gamma}_k$ — обобщенный градиент функции $\varphi_k(y)$ в точке $y^{(k)}$).

Вектор

$$\tilde{\gamma}_k=B_k^*\gamma_f(x^{(k)}) \tag{5}$$

(B_k^* — оператор, сопряженный оператору B_k).

Возвращаясь в исходное пространство (применяя к обеим частям (4) оператор B_k), получаем

$$x^{(k+1)}=x^{(k)}-\alpha_k B_k \frac{\tilde{\gamma}_k}{\|\tilde{\gamma}_k\|}. \tag{6}$$

Рассматриваемый $(k+1)$-й шаг завершается нахождением оператора

$$B_{k+1}=B_k P_k, \tag{7}$$

где P_k — оператор растяжения пространства R^n в направлении $v=\tilde{\gamma}_k/\|\tilde{\gamma}_k\|$ с коэффициентом $\beta=1/\beta_k$; $\beta_k>0$ — некоторое число.

В качестве B_0 может быть выбрана любая неособая матрица, например единичная матрица E.

Используя формулы (5)—(7), при одних и тех же начальных $x^{(0)}$ и B_0 могут быть построены различные последовательности точек $x^{(1)}$, $x^{(2)}$, ... в зависимости от того, какими будут выбраны последовательности длин шагов α_0, α_1, ... и коэффициентов растяжения пространства β_0, β_1, ... В частности, если $B_0=E$ и $\beta_k=1$, $k=0, 1, \ldots$, то получаем обычный метод обобщенного градиентного спуска.

Для класса выпуклых функций можно подобрать такой вариант метода обобщенного градиентного спуска с растяжением пространства, при котором последовательность $x^{(0)}, x^{(1)}, \ldots$ сходится по функционалу со скоростью геометрической прогрессии, знаменатель которой зависит только от размерности области определения минимизируемой функции.

1.11. Качество каждого из описанных методов характеризуется многими факторами: классом минимизируемых функций, скоростью сходимости, трудоемкостью выполнения одного итерационного шага, объемом запоминаемой в процессе вычислений информации и т. п. Отдать априори предпочтение какому-либо одному или небольшой группе методов не представляется возможным. Выбор подходящего метода для минимизации конкретно рассматриваемой функции во многих случаях вызывает серьезные затруднения. На практике широкое распространение получили так называемые *гибридные алгоритмы*, позволяющие в процессе решения задачи осуществлять переход от одного метода к другому в зависимости от получаемых результатов.

2. МЕТОДЫ РЕШЕНИЯ ЭКСТРЕМАЛЬНЫХ ЗАДАЧ С ОГРАНИЧЕНИЯМИ

В данном параграфе описываются численные методы минимизации функций нескольких переменных при ограничениях на значения переменных.

Для определенности рассматриваются задачи вида

$$f(x)\to\min\ (=f^*), \tag{1}$$

$$g_i(x)\leqslant 0,\quad i=\overline{1,m}, \tag{2}$$

где $x=(x_1, x_2, \ldots, x_n)\in R^n$; $f(x)$ и $g_i(x)$, $i=\overline{1,m}$,— некоторые действительные функции от x.

Множество значений x, удовлетворяющих неравенствам (2), будем обозначать через X.

Для минимизации функции $f(x)$ на множестве X разработано много различных методов. Наиболее изучена ситуация, когда функция $f(x)$ и множество X выпуклы.

2.1. *Методы штрафных функций* позволяют получать решение задачи (1)—(2) в результате решения серии задач безусловной минимизации функций вида

$$f(x)+\Psi_k(x|X)\to\min\ (=f_k^*), \tag{3}$$

где функции $\Psi_k(x|X)$ подбираются таким образом, что $x_k^*\to x^*$ или по крайней мере $f_k^*\to f^*$ при $k\to\infty$. Здесь x^* и x_k^* — оптимальные значения вектора x; f^* и f_k^* — оптимальные значения целевых функций в задачах (1)—(2) и (3) соответственно.

В *методах внутренних штрафных функций* (или, что то же, *барьерных функций*) предполагается известной некоторая начальная точка $x^{(0)}$, удовлетворяющая неравенствам $g_i(x^{(0)})<0$, $i=\overline{1,m}$. В качестве функций $\Psi_k(x|X)$ обычно выбираются функции вида

$$\Psi_k(x|X)=-\beta_k\sum_{i=1}^{m}\ln(-g_i(x)),$$

или

$$\Psi_k(x|X)=-\beta_k\sum_{i=1}^{m}1/g_i(x),$$

где параметр штрафа $\beta_k>0$, $\beta_k\to 0$ при $k\to\infty$.

При приближении к границе множества X каждая из указанных функций $\Psi_k(x|X) \to \infty$ при $\beta_k > 0$.

В *методах внешних штрафных функций* функции $\Psi_k(x|X)$ строятся так, чтобы при всех $k=1, 2, \ldots$ имело место $\Psi_k(x|X)=0$, если $x \in X$ и $\Psi_k(x|X) > 0$, если $x \notin X$. В качестве таких функций могут быть выбраны, например, следующие:

$$\Psi_k(x|X) = \frac{1}{\beta_k} \sum_{i=1}^{m} [\max\{0, g_i(x)\}]^2,$$

или

$$\Psi_k(x|X) = \frac{1}{\beta_k} \sum_{i=1}^{m} \max\{0, g_i(x)\},$$

где параметр штрафа $\beta_k > 0$, $\beta_k \to 0$ при $k \to \infty$.

Наряду с перечисленными используются и другие штрафные функции. Поэтому при составлении задачи (3) можно обеспечить выполнимость желательных с вычислительной точки зрения условий, таких, как гладкость, выпуклость, простота вычисления значений минимизируемой функции и ее производных и т. п.

2.2. *Метод проекции градиента* является естественным обобщением метода градиентного спуска на случай, когда $X \neq R^n$.

Напомним, что проекцией точки $y \in R^n$ на множество $X \subseteq R^n$ называется ближайшая к y точка x множества X (обозначение $x = P_X(y)$). По определению, если $y \in X$, то $P_X(y) = y$. В частности, если $X = R^n$, то $P_X(y) = y$ для всех $y \in X$.

Если X — выпуклое замкнутое множество, то всякая точка y имеет, и притом единственную, проекцию x на это множество. В данном случае для того чтобы $x = P_X(y)$, необходимо и достаточно, чтобы $(x-y, z-x) \geqslant 0$ при всех $z \in X$.

Отыскание проекции точки y на множество X сопряжено с решением задачи минимизации (по x) квадратичной функции $\|y-x\|^2$ на множестве X. Поскольку в общем случае такая задача является достаточно сложной, то часто ограничиваются нахождением некоторого подходящего приближения $\tilde{x}$ точки $x = P_X(y)$ с соблюдением условия, что $\tilde{x} \in X$.

В методе проекции градиента строится последовательность точек $x^{(0)}$, $x^{(1)}$, ..., удовлетворяющая рекуррентному соотношению

$$x^{(k+1)}=P_X(x^{(k)}-\alpha_k f'(x^{(k)})), \qquad (4)$$

где $f'(x^{(k)})$ — градиент функции $f(x)$ в точке $x^{(k)}$.

Если множество X — выпуклое и замкнутое (т. е. все функции $g_i(x)$ — выпуклые) и длины α_k всех шагов заданы, то соотношение (4) однозначно определяет указанную последовательность точек. Использование оператора P_X позволяет обеспечить принадлежность этих точек множеству X.

Как и в методе градиентного спуска, выбор длины шага $\alpha_k \geqslant 0$ может осуществляться различными способами, приводя к разным вариантам метода проекции градиента.

Так, находя α_k в результате минимизации одномерной (относительно α) функции $f(Px(x^{(k)}-\alpha f'(x^{(k)})))$, получаем *аналог метода наискорейшего спуска.*

В *методе проекции градиента с дроблением шага выбор* α_k осуществляется согласно условию

$$f(P_X(x^{(k)}-\alpha_k f'(x^{(k)})))-f(x^{(k)}) \leqslant$$
$$\leqslant -\varepsilon \| P_X(x^{(k)}-\alpha_k f'(x^{(k)}))-x^{(k)}\|^2,$$

где C — некоторая положительная константа.

Если $f(x)$ — непрерывно дифференцируемая на X функция и ее градиент $f'(x)$ удовлетворяет условию Липшица на X с константой ρ, то в качестве α_k можно выбрать любое число, удовлетворяющее условиям $0<\varepsilon_0<$ $<\alpha_k \leqslant 2/(\rho+2\varepsilon)$ (ε_0, ε — положительные числа (параметры метода)). В этом случае для задач выпуклого программирования справедлива оценка $f(x^{(k)})-f^* \leqslant C/k$, где C — некоторая положительная константа.

Зачастую в качестве α_k выбирают любое положительное число, обеспечивающее выполняемость условия релаксации $f(x^{(k+1)})<f(x^{(k)})$.

2.3. *В методе возможных направлений* для нахождения очередной точки $x^{(k+1)}$ строится конус возможных направлений с вершиной $x^{(k)}$. В построенном конусе путем решения соответствующей задачи линейного программирования выбирается направление, в котором значение целевой функции может быть уменьшено (без нарушения ограничений задачи).

Вектор $v=(v_1, v_2, \ldots, v_n)\in R^n$, $v\neq 0$, называется возможным направлением в точке $x\in X$, если существует такое число $\varepsilon_0>0$, что точка $x(\varepsilon)=x+\varepsilon v\in X$ для всех $0\leqslant\varepsilon\leqslant\varepsilon_0$.

Если все функции $g_i(x)$, $i=\overline{1, m}$, являются непрерывно дифференцируемыми, то конус возможных направлений в точке $x\in X$ задается неравенствами вида $(g'_i(x), v)<0$, $i\in I_0$, где I_0 — множество всех индексов i, $1\leqslant i\leqslant n$, при которых $g_i(x)=0$.

В предположении, что $f(x)$ — непрерывно дифференцируемая функция, возможное направление ее убывания (если оно существует) должно удовлетворять неравенству $(f'(x), v)<0$.

В методе возможных направлений искомый вектор v выбирается таким образом, чтобы наибольшее из значений $(f'(x), v)$ и $(g'_i(x), v)$, $i\in I_0$, было минимальным. Иными словами, на каждом шаге $k+1$, $k=0, 1, \ldots$, решается задача линейного программирования вида

$$z\to\min,$$
$$(f'(x^{(k)}), v)\leqslant z,$$
$$(g'_i(x^{(k)}), v)\leqslant z, \quad i\in I_0^{(k)},$$
$$0\leqslant v_j\leqslant 1, \quad j=\overline{1, n}.$$

Двухсторонние ограничения на v_j добавлены для обеспечения конечности минимума z. Вместо этих ограничений иногда добавляется квадратичное ограничение нормировки: $\|v\|^2\leqslant 1$.

Указанная задача линейного программирования всегда имеет решение, независимо от того, существует искомое возможное направление $v^{(k)}$ или нет. Можно показать, что если оптимальное значение целевой функции неотрицательно, то в точке $x^{(k)}$ выполняются необходимые условия оптимальности. В противном случае осуществляется переход к точке $x^{(k+1)}=x^{(k)}+\alpha_k v^{(k)}$.

Выбор длины шага $\alpha_k>0$ может быть произведен, например, в результате решения задачи одномерной минимизации (по $\alpha\geqslant 0$):

$$f(x^{(k)}+\alpha v^{(k)})\to\min,$$
$$g_i(x^{(k)}+\alpha v^{(k)})\leqslant 0, \quad i=\overline{1, m}.$$

2.4. *Метод условного градиента* может использоваться для минимизации нелинейной дифференцируемой функции $f(x)$ на выпуклом множестве X в том случае, когда минимизация линейной функции на X не вызывает сколь-нибудь серьезных затруднений. Согласно этому методу, в окрестности каждой очередной точки $x^{(k)} \in X$ функция $f(x)$ аппроксимируется линейной функцией вида $f(x^{(k)})+(f'(x^{(k)}), x-x^{(k)})$, находится точка $\bar{x}^{(k)}$ ее минимума (или, что то же, функции $(f'(x^{(k)}), x-x^{(k)})$) на множестве X и определяется точка $x^{(k+1)}=x^{(k)}+\alpha_k v^{(k)}$, где $v^{(k)}=\bar{x}^{(k)}-x^{(k)}$.

Длина шага $\alpha_k>0$ может выбираться различным образом, например в результате минимизации одномерной (относительно α) функции $f(x^{(k)}+\alpha v^{(k)})$ на интервале $(0, 1)$.

Если $f(x)$ — непрерывно дифференцируемая функция и константа Липшица ρ для $f'(x)$ на X известна, то можно положить величину α_k, равной 2^{-j_0}, где j_0 — первый индекс $j=0, 1, \ldots$, для которого выполняется неравенство

$$f(x^{(k)}+2^{-j}v^{(k)}) \leqslant f(x^{(k)})+2^{-(j+1)}(f'(x^{(k)}), v^{(k)}).$$

В этом случае для выпуклой функции $f(x)$ справедлива оценка $f(x^{(k)})-f^* \leqslant C/k$, где C — некоторая положительная константа. Если X — сильно выпуклое множество (т. е. существует такое число $\delta>0$, что для любых x и y из X точки $\frac{x+y}{2}+v$ принадлежат X для всех $v \in R^n$ таких, что $\|v\| \leqslant \delta\|x-y\|^2$), то метод условного градиента сходится со скоростью геометрической прогрессии.

2.5. В отличие от предыдущего метода в *методе Ньютона* для нахождения направления спуска $v^{(k)}=\bar{x}^{(k)}-x^{(k)}$ используется квадратичная аппроксимация функции $f(x)$ в окрестности точки $x^{(k)}$. Иными словами, в качестве $\bar{x}^{(k)}$ выбирается точка минимума функции $(f'(x^{(k)}), x-x^{(k)})+\frac{1}{2}(f''(x^{(k)})(x-x^{(k)}), x-x^{(k)})$ на множестве X.

При определенных условиях скорость сходимости этого метода либо сверхлинейна, либо квадратична.

2.6. *В методе линеаризации* на каждом шаге $k+1$, $k=0, 1, \ldots$, осуществляется линеаризация функций $f(x)$

и $g_i(x)$, $i=\overline{1, m}$, в окрестности точки $x^{(k)}$. В результате получается некоторая задача линейного программирования, обычно не имеющая решения. Для обеспечения достаточной «малости» окрестности точки $x^{(k)}$ к линеаризованной целевой функции добавляется квадратичный член.

Пусть $g(x)=\max\limits_{1\leqslant i\leqslant m} g_i(x)$ и $I_\delta(x)$ — множество всех индексов i, $1\leqslant i\leqslant m$, при которых $g_i(x)\geqslant g(x)-\delta$ в точке x, число $\delta\geqslant 0$. Не нарушая общности, можно считать, что $g(x)\geqslant 0$ (добавив, например, к ограничениям (2) ограничение с функцией $g_i(x)$, тождественно равной нулю).

Для нахождения направления $v^{(k)}$ спуска в точке $x^{(k)}$ решается задача квадратичного программирования

$$(f'(x^{(k)}), v)+\frac{1}{2}\|v\|^2\to\min,$$

$$(g_i'(x^{(k)}), v)+g_i(x^{(k)})\leqslant 0, \quad i\in I_\delta(x^{(k)}).$$

Длина шага α_k полагается равной 2^{-j_0}, где j_0 — первое значение $j=0, 1, \ldots$, при котором выполняется неравенство

$$f(x^{(k)}+2^{-j}v^{(k)})+Cg(x^{(k)}+2^{-j}v^{(k)})\leqslant$$

$$\leqslant f(x^{(k)})+Cg(x^{(k)})-2^{-j}\varepsilon\|v^{(k)}\|^2,$$

где константы $C>0$ и $\varepsilon>0$ — параметры метода.

Очередная точка $x^{(k+1)}=x^{(k)}+\alpha_k v^{(k)}$.

При естественных предположениях в общем нелинейном случае метод линеаризации имеет геометрическую, а зачастую и квадратичную скорость сходимости. Если задача (1)—(2) является задачей линейного программирования, то этот метод становится конечным.

2.7. Ряд современных методов решения задач математического программирования основан на идее последовательного сужения области локализации искомой точки минимума x^*. На каждом шаге от множества X отсекается некоторое подмножество таким образом, что в оставшейся части множества X заведомо содержится хотя бы одна точка минимума функции $f(x)$. Отсечения обычно проводятся с использованием соответствующим образом подобранных гиперплоскостей.

Пусть для определенности $f(x)$, $g_i(x)$, $i=\overline{1, m}$,— выпуклые дифференцируемые функции, X — ограниченное множество и $\hat{x}$ — некоторая внутренняя точка этого множества, т. е. $g_i(\hat{x})<0$, $i=\overline{1, m}$.

Введем в рассмотрение гиперплоскость $(f'(\hat{x}), x-\hat{x})=0$ и обозначим через $\hat{X}$ множество всех точек из X, удовлетворяющих условию $(f'(\hat{x}), x-\hat{x})\leqslant 0$.

Этот прием позволяет свести исходную задачу минимизации $f(x)$ на множестве X к задаче минимизации $f(x)$ на множестве $\hat{X}$, объем которого меньше исходного. Выбирая в качестве X множество $\hat{X}$, можно провести те же рассуждения относительно вновь полученной задачи и т. д., пока точка x^* не будет локализована во множестве достаточно малого объема (определяемого заданной точностью решения исходной задачи).

Если в качестве точки $\hat{x}$ на каждом шаге выбирать центр тяжести очередного множества X, будет отсекаться не менее чем $(n/(n+1))^n$ часть объема предыдущего множества X. Такой метод, получивший название *метода централизованных отсечений,* обеспечивает убывание объема области локализации точки x^* со скоростью не меньшей, чем скорость убывания геометрической прогрессии со знаменателем $1-\left(\frac{n}{n+1}\right)^n\approx 1-\frac{1}{e}$ при больших n.

К сожалению, нахождение центра тяжести множества X в большинстве случаев (даже когда, например, X — многогранник) является весьма трудоемкой задачей. В то же время, если X — шар, эллипсоид, ортогональный симплекс (т. е. множество точек $x\in R^n$, удовлетворяющих неравенствам $\sum\limits_{j=1}^{n} x_j\leqslant n+1$, $x_j\geqslant 0$, $j=\overline{1, n}$) и т. п., то нахождение центра тяжести множества X не вызывает затруднений, что приводит к созданию достаточно эффективных методов решения задач математического программирования, на каждом шаге которых множество X погружается в некоторое более «простое» подмножество и проводится центрированное сечение последнего.

Так, *в методе эллипсоидов* исходное множество X или часть его, заведомо содержащая x^*, погружается в некоторый эллипсоид, обычно в шар. При решении практических задач это нетрудно сделать, поскольку, как правило, известны «разумные» границы изменения переменных. Проводится центрированное сечение эллипсоида и та часть его, которая заведомо содержит x^*, погружается в новый эллипсоид возможно меньшего объема. Проводится центрированное сечение нового эллипсоида и т. д., пока не будет получен эллипсоид достаточно малого объема. Центр его и выбирается в качестве x^*.

Формально эта процедура может быть описана следующим образом.

Введем в рассмотрение шар с центром в точке $x^{(0)}$ и радиусом, равным $r>0$ (т. е. множество точек $x \in R^n$, удовлетворяющих неравенству $\|x-x^{(0)}\| \leqslant r$), заведомо содержащий точку x^*. Положим $B_0=E$ — единичная матрица размерности $n \times n$, число $h_0=\dfrac{r}{n+1}$.

Пусть на k-м шаге алгоритма получены $x^{(k)}$, B_k, h_k. На $(k+1)$-м шаге:

1) вычисляем $\varphi^{(k)}=f'(x^{(k)})$, если $g_i(x^{(k)}) \leqslant 0$, $i=\overline{1, m}$, и $\varphi^{(k)}=g'_{i_k}(x^{(k)})$, если $g_{i_k}(x^{(k)}) = \max\limits_{1 \leqslant i \leqslant m} g_i(x^{(k)}) > 0$;

2) полагаем $\eta^{(k)}=B_k^* \varphi^{(k)}/\|B_k^* \varphi^{(k)}\|$, где B_k^* — матрица, сопряженная матрице B_k;

3) находим новую точку $x^{(k+1)}=x^{(k)}-h_k B_k \eta^{(k)}$;

4) определяем $B_{k+1}=B_k P_k$, где P_k — оператор растяжения пространства R^n в направлении $v=\eta^{(k)}$ с коэффициентом $\beta=\sqrt{\dfrac{n-1}{n+1}}$ (см. параграф 1, п. 1.10);

5) полагаем $h_{k+1}=\dfrac{h_k \eta}{\sqrt{n^2-1}}$.

Объем эллипсоидов, в которых локализуется точка x^*, убывает со скоростью геометрической прогрессии со знаменателем $\sqrt{\dfrac{n-1}{n+1}}\left(\dfrac{n}{\sqrt{n^2-1}}\right)^n < 1$.

Наряду с описанным широко используются и другие варианты метода эллипсоидов.

В заключение отметим, что приведенные выше рассуждения естественным образом обобщаются и на случай недифференцируемых функций. Для этого достаточно вместо градиента использовать субградиенты.

2.8. При минимизации функций многих переменных, особенно недифференцируемых и многоэкстремальных, широко используются *методы случайного поиска.* Они обычно обладают небольшой скоростью сходимости, но могут оказаться весьма полезными в случаях, когда применение других методов затруднительно либо вообще невозможно. Эффективность методов случайного поиска существенно возрастает, если имеется возможность организовать одновременное вычисление значений функции $f(x)$ во многих точках множества X (используя, например, мультипроцессорные вычислительные системы).

В методах случайного поиска (как, впрочем, и в большинстве описанных выше методах) строится релаксационная последовательность точек $x^{(0)}, x^{(1)}, \ldots$ из X согласно соотношению

$$x^{(k+1)}=x^{(k)}+\alpha_k v^{(k)}. \tag{5}$$

В качестве направления спуска $v^{(k)}$ выбирается какая-либо реализация случайного вектора $\xi^{(k)}=(\xi_1^{(k)}, \xi_2^{(k)}, \ldots$ $\ldots, \xi_n^{(k)})$ с известным законом распределения (изменяемым, вообще говоря, при переходе от шага к шагу).

В *методах случайного поиска без обучения* обычно полагают $\alpha_k=\alpha>0$ и $\xi^{(k)}=\xi/\|\xi\|$, $k=0, 1, \ldots$, где $\xi=$ $=(\xi_1, \xi_2, \ldots, \xi_n)$, ξ_j, $j=\overline{1, n}$,— независимые случайные величины, распределенные равномерно на интервале $[-1, 1]$.

Простейшим среди этих методов является *метод с возвратом при неудачном шаге.* Согласно этому методу, на шаге k с помощью соответствующего датчика случайных чисел получают некоторую реализацию случайного вектора ξ. Находится точка $x^{(k+1)}=x^{(k)}+\alpha\xi/\|\xi\|$. Если $x^{(k+1)}\notin X$ либо $x^{(k+1)}\in X$, но $f(x^{(k+1)})\geqslant f(x^{(k)})$, то осуществляется возврат в точку $x^{(k)}$. Если $x^{(k+1)}\in X$ и $f(x^{(k+1)})<$ $<f(x^{(k)})$, то осуществляется переход в точку $x^{(k+1)}$. В любом случае получают некоторую новую реализацию вектора ξ и продолжают процесс поиска x^*. При этом иногда приходится прибегать к изменению длины шага α (используя, например, способ удвоения шагов).

В *методе наилучшей пробы* задается некоторое целое число $l>0$ и берутся l реализаций случайного вектора ξ. Для каждой реализации находится точка вида $x^{(k)}+$ $+\alpha\xi/\|\xi\|$. Среди них выбираются точки, принадлежащие X, а среди последних точка $\hat{x}$ с наименьшим значением функции $f(x)$. Если $f(\hat{x})<f(x^{(k)})$, то полагают $x^{(k+1)}=\hat{x}$.

В противном случае процесс поиска $x^{(k+1)}$ повторяют, изменяя при необходимости длину шага α.

Существенно большей эффективностью обладают *методы случайного поиска с обучением*. В таких методах заложено стремление учесть априорную информацию о поведении функции $f(x)$ на множестве X, а также текущую информацию, получаемую в процессе поиска точки x^*. Достигается это посредством выбора подходящего закона распределения случайного вектора ξ на первом шаге и последующего его уточнения при переходе от шага к шагу. Соответствующим изменениям подвергается и длина шага.

В качестве примера приведем краткое описание *метода покоординатного обучения* в предположении, что $X=R^n$. Согласно этому методу, закон распределения случайного вектора $\xi=(\xi_1, \xi_2, \ldots, \xi_n)$ на каждом шаге описывается с помощью задания вектора параметров $w=(w_1, w_2, \ldots, w_n)$ следующим образом. Значение $\xi_j=1$ с вероятностью p_j и $\xi_j=-1$ с вероятностью $1-p_j$. Величина $p_j=0$, если $w_j<-1$; $p_j=\frac{1}{2}(1+w_j)$, если $|w_j|\leqslant 1$; $p_j=1$, если $w_j>1$.

На первом шаге полагаем $w_j^{(0)}=0$, $j=\overline{1, n}$, т. е. каждая компонента вектора $\xi^{(0)}$ равновероятно принимает значение -1 или 1. При какой-либо реализации этого вектора, согласно формуле (5), (при $v^{(0)}=\xi^{(0)}/\|\xi^{(0)}\|$) находим точку $x^{(1)}$. На втором шаге полагаем $w_j^{(1)}=0$, $j=\overline{1, n}$, и аналогичным образом определяем точку $x^{(2)}$.

Пусть известны точки $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$ и вектор параметров $w^{(k-1)}$. Полагаем

$$w_j^{(k)}=\beta w_j^{(k-1)}-\delta \operatorname{sign}\left[\left(f\left(x^{(k-1)}\right)-\right.\right.$$
$$\left.\left.-f\left(x^{(k-2)}\right)\right)\left(x_j^{(k-1)}-x_j^{(k-2)}\right)\right]$$

для всех $j=\overline{1, n}$.

Величина $\beta\geqslant 0$ называется параметром забывания, величина $\delta\geqslant 0$ — параметром интенсивности обучения, $\beta+\delta>0$.

Знание вектора параметров $w^{(k)}$ позволяет получить реализацию случайного вектора $\xi^{(k)}$, найти точку $x^{(k+1)}$ и продолжить вычислительный процесс.

Приведенное рекуррентное соотношение для нахождения вектора параметров $w^{(k)}$ имеет следующую содержательную интерпретацию. Если при переходе от точки $x^{(k-2)}$ к точке $x^{(k-1)}$ значение функции $f(x)$ уменьшилось, то поиск точки $x^{(k+1)}$ с большей вероятностью будет осуществляться в направлении $x^{(k-1)}-x^{(k-2)}$. При увеличении значения функции вероятность выбора указанного направления снижается.

Следует отметить, что в настоящее время разработан широкий спектр методов случайного поиска. Выбор конкретного метода определяется спецификой решаемой задачи и возможностями имеющихся в распоряжении исследователя средств вычислительной техники.

3. ЭЛЕМЕНТЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Будем рассматривать задачу линейного программирования, записанную для определенности в канонической форме:

$$(c,\ x)\to\min, \tag{1}$$

$$Ax=b, \tag{2}$$

$$x\geqslant 0, \tag{3}$$

где $x=(x_1,\ x_2,\ \ldots,\ x_n)\in R^n$ — n-мерный вектор; $A=(A_1,\ A_2,\ \ldots,\ A_n)$ — матрица размерности $m\times n$ с элементами a_{ij}, $i=\overline{1,\ m}$, $j=\overline{1,\ n}$; A_j — столбцы этой матрицы, $j=\overline{1,\ n}$; $c=(c_1,\ c_2,\ \ldots,\ c_n)\in R^n$ и $b=(b_1,\ b_2,\ \ldots,\ b_m)\in R^m$— векторы с n и m компонентами соответственно, $m<n$, $b\geqslant 0$.

3.1. Некоторые свойства задачи (1)—(3). Известно, что если система (2)—(3) совместна, то множество X значений x, удовлетворяющих условиям (2)—(3), является выпуклым, замкнутым многогранным множеством.

Точка $\tilde{x}\in X$ называется *крайней* точкой *(вершиной)* множества X, если в X не существует таких точек x' и x'', $x'\neq x''$, что $\tilde{x}=\alpha x'+(1-\alpha)x''$ при некотором $0<\alpha<1$. Для того чтобы точка $\tilde{x}$ была вершиной многогранного множества X, необходимо и достаточно, чтобы вектор $\tilde{x}$ был базисным допустимым решением (опорным планом) задачи (1)—(3), т. е. таким решением, при котором столбцы A_j матрицы A, отвечающие положительным компонентам вектора $\tilde{x}$, являются линейно независимыми. Ясно, что в базисном решении не более m положительных компонент. Вершине $\tilde{x}$ можно сопоставить *матрицу базиса* B, составленную из m линейно независимых столбцов A_j матрицы A и включающую все те столбцы A_j, которые соответствуют положительным компонентам вектора $\tilde{x}$. Если положительных компонент ровно m, то базисное решение (вершина многогранника X) невырождено.

Если X — ограниченное множество (многогранник), то оно совпадает с совокупностью точек вида

$$x=\sum_{t=1}^{T} z_t x^{(t)},\quad \sum_{t=1}^{T} z_t=1,\quad z_t\geqslant 0,\quad t=\overline{1,\ T}, \tag{4}$$

где $x^{(t)}$, $t=\overline{1, T}$,— вершины многогранника X *(теорема о представлении многогранника X).*

Если X — неограниченное множество, то наряду с его вершинами могут быть выделены неограниченные ребра вида $x=\tilde{x}+\mu\bar{x}$, где $\tilde{x}$ — вершина множества X; $\bar{x}$ — направляющий вектор ребра (решение однородной системы $Ax=0$); параметр μ изменяется от 0 до ∞. В этом случае множество X совпадает с совокупностью точек вида

$$x=\sum_{t=1}^{T_1} z_t x^{(t)}, \quad \sum_{t=1}^{T_1} \varepsilon_t z_t=1, \quad z_t \geqslant 0, \quad t=\overline{1, T_1}, \tag{5}$$

где $x^{(t)}$, $t=\overline{1, T}$ — вершины; $x^{(t)}$, $t=\overline{T+1, T_1}$,— направляющие векторы неограниченных ребер множества X; $\varepsilon_t=1$, $t=\overline{1, T}$, $\varepsilon_t=0$, $t=\overline{T+1, T_1}$ *(теорема о представлении многогранного множества X).*

Если множество X непусто и функция (1) ограничена снизу на этом множестве, то задача (1)—(3) обладает хотя бы одним решением и среди ее решений найдется хотя бы одна вершина множества X, т. е.

$$\min\{(c, x) \mid x \in X\}=\min\{(c, x^{(t)}) \mid t=\overline{1, T}\}. \tag{6}$$

Этот факт лежит в основе конечных методов линейного программирования, поскольку позволяет получить решение задачи (1)—(3) в результате вычисления значений целевой функции (1) в вершинах множества X (число которых конечно) с последующим выбором наименьшего значения. Существуют различные стратегии перебора вершин множества X и, следовательно, различные методы решения задач линейного программирования.

Сформулируем *признак оптимальности* вершины множества X. Пусть $\tilde{x}$ — вершина множества X (базисное решение, опорный план задачи (1)—(3)); $B=(A_{j_1}, A_{j_2}, \ldots, A_{j_m})$ — матрица базиса, соответствующая $\tilde{x}$; c_B — вектор коэффициентов целевой функции, соответствующий матрице B. Обозначим $\Delta_j=c_j-(c_B, B^{-1}A_j)$, $j=\overline{1, n}$. Ясно, что $\Delta_{j_l}=0$ для всех $l=\overline{1, m}$. Величины Δ_j называются обычно характеристическими разностями.

Если $\Delta_j \geqslant 0$ для всех $j=\overline{1, n}$, то $\tilde{x}$ — искомая оптимальная вершина.

Если найдется $j=k$, при котором $\Delta_k<0$ и $B^{-1}A_k\leqslant 0$, то множество X неограничено и функция (c, x) неограничена снизу на этом множестве.

Наконец, если при некотором $j=k$ значение $\Delta_k<0$ и вектор $B^{-1}A_k$ содержат хотя бы одну положительную компоненту, то существует вершина множества X, в которой значение функции (c, x) меньше, чем в исходной вершине $\tilde{x}$. В частности, такой вершиной является вершина $\tilde{x}'$, матрица базиса B' которой может быть получена из матрицы B замещением одного из ее столбцов столбцом A_k.

3.2. Методы решения задачи (1)—(3) *Симплексный метод (метод последовательного улучшения плана)* решения задач линейного программирования позволяет получить искомую оптимальную вершину множества X последовательным перебором вершин этого множества, при котором значение целевой функции убывает от вершины к вершине. При описании и обосновании данного метода обычно предполагается, что задача (1)—(3) невырождена, т. е. все вершины множества X невырождены. В противном случае не исключено, что вычислительный процесс зациклится и будет осуществляться циклический перебор одних и тех же вершин множества X. Для преодоления явления зацикливания используется так называемый метод возмущений, позволяющий в неопределенной ситуации решить, какой именно вектор матрицы B следует удалить при переходе к новой матрице базиса.

Сопоставим (невырожденной) вершине $\tilde{x}$ набор $(m+1)\times(n+1)$ параметров u_{ij}, $i=\overline{0, m}$, $j=\overline{0, n}$, где $u_{00}=(c, \tilde{x})$; $u_{0j}=\Delta_j$, $j=\overline{1, n}$; u_{ij} — i-я компонента вектора $B^{-1}A_j$, $i=\overline{1, m}$, $j=\overline{1, n}$; u_{i0} — i-я компонента вектора $B^{-1}b$. Ясно, что u_{00} — значение целевой функции в вершине $\tilde{x}$; u_{i0}, $i=\overline{1, m}$,— положительные компоненты вектора $\tilde{x}$ (в рассматриваемом случае $x_{j_1}, x_{j_2}, \ldots, x_{j_m}$, остальные $x_j=0$).

Пусть при некотором $j=k$ значение $u_{0k}<0$ и множество индексов I_k, при которых $u_{ik}>0$, непусто. В этом случае, как уже отмечалось, делается итерационный шаг, осуществляющий переход от вершины $\tilde{x}$ с матрицей базиса $B=(A_{j_1}, \ldots, A_{j_{s-1}}, A_{j_s}, A_{j_{s+1}}, \ldots, A_{j_m})$ к вершине $\tilde{x}'$ с матрицей базиса $B'=(A_{j_1}, \ldots, A_{j_{s-1}}, A_{j_{s+1}}, \ldots, A_{j_m}, A_k)$.

Исключаемый из базиса вектор A_{j_s} (в невырожденном случае) определяется исходя из условия: $u_{s0}/u_{sk}=$ $=\min\{u_{i0}/u_{ik}|i\in I_k\}$.

Существует простая рекуррентная связь между параметрами u_{ij} вершины $\tilde{x}$ и соответствующими параметрами u'_{ij} вершины $\tilde{x}'$:

$$u'_{ij}=u_{ij}-u_{ik}\frac{u_{sj}}{u_{sk}},\ i=\overline{0,\ m},\ i\neq s,\ j=\overline{0,\ m};$$

$$u'_{kj}=\frac{u_{sj}}{u_{sk}},\ j=\overline{0,\ n}.$$

Приведенные соотношения позволяют систематизировать процесс вычислений. Переход от одного набора параметров $\{u_{ij}\}$ к другому $\{u'_{ij}\}$ производится до тех пор, пока не будет выполнено условие: $u_{0j}\geqslant 0$ для всех $j=\overline{1,\ n}$. Для получения исходной вершины (исходного набора параметров $\{u_{ij}\}$) используют обычно метод искусственного базиса.

Модифицированный симплексный метод дает возможность устранить хранение на каждой итерации полного набора параметров $\{u_{ij}\}$, необходимых для вычисления $\{u'_{ij}\}$. В этом методе на каждом шаге определяют матрицу B^{-1}. Зная ее, можно найти $\Delta_j=c_j-(c_B,\ B^{-1}A_j)$, $B^{-1}b$ и другие параметры вершины $\tilde{x}$. Объем вычислений естественно возрастает, однако объем запоминаемой текущей информации существенно сокращается.

Если последовательно рассматриваемые матрицы базисов B и B' представлены в виде $B=(A_{j_1},\ \ldots,\ A_{j_{s-1}},\ A_{j_s},$ $A_{j_{s+1}},\ \ldots,\ A_{j_m})$ и $B'=(A_{j_1},\ \ldots,\ A_{j_{s-1}},\ A_k,\ A_{j_{s+1}},\ \ldots,\ A_{j_m})$, то элементы матриц $B^{-1}=(b_{ij})$ и $(B')^{-1}=(b'_{ij})$ связаны простыми рекуррентными соотношениями

$$b'_{ij}=b_{ij}-u_{ik}\frac{b_{sj}}{u_{sk}},\ i\neq s,\ b'_{sj}=\frac{b_{sj}}{u_{sk}},$$

позволяющими исключить выполнение довольно трудоемкой процедуры непосредственного обращения матрицы B', если известна матрица B^{-1}.

3.3. Наряду с прямыми методами широкое распространение получили так называемые *двойственные методы* линейного программирования и их комбинации с прямыми.

Задачей *двойственной* к задаче (1)—(3) называется задача

$$(b, \Lambda) \to \max, \tag{7}$$

$$A^{\tau}\Lambda \leqslant c, \tag{8}$$

где $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ — вектор двойственных переменных; τ — знак транспонирования.

Если x удовлетворяет ограничениям (2)—(3) прямой задачи, а Λ — ограничениям (8) двойственной задачи, то $(c, x) \geqslant (b, \Lambda)$.

Задачи (1)—(3) и (7)—(8) образуют двойственную пару. Если одна из задач этой пары разрешима, то и другая задача также разрешима. Если x^* и Λ^* — решения этих задач, то $(c, x^*) = (b, \Lambda^*)$ *(первая теорема двойственности)*.

Если для любого оптимального решения x^* (Λ^*) прямой (двойственной) задачи имеет место $x^*_j = 0$ (соответственно $\sum\limits_{i=1}^{m} a_{ij}\lambda^*_i = c_j$), то условие $x_j \geqslant 0$ (условие $\sum\limits_{i=1}^{m} a_{ij}\lambda_i \leqslant$ $\leqslant c_j$) называется закрепленным, в противном случае — свободным. Если задачи разрешимы, то в каждой паре их двойственных условий одно условие является свободным, другое закрепленным *(вторая теорема двойственности)*.

Двойственный симплексный метод (метод последовательного уточнения оценок) основан на том факте, что при решении линейной задачи симплексным методом наряду с ее решением можно получить и решение двойственной задачи (в прежних обозначениях $\Lambda = c_B B^{-1} = (B^{-1})^{\tau} c_B$). Тем самым можно осуществлять целенаправленный перебор вершин многогранного множества, описываемого ограничениями (8) двойственной задачи, при котором целевая функция (7) возрастает, обеспечивая в результате получение решения прямой задачи.

Метод одновременного решения прямой и двойственной задач (метод последовательного сокращения невязок) позволяет выполнять итерационный процесс, начиная с любой допустимой точки (необязательно вершины) множества (8).

Пусть $\Lambda^{(0)}$ удовлетворяет ограничениям (8) двойственной задачи и J_0 — множество индексов ограничений, выполняемых в точке $\Lambda^{(0)}$ со знаком равенства.

Сформулируем вспомогательную линейную задачу

$$\sum_{i=1}^{m} v_i \to \min,$$

$$\sum_{j \in J_0} A_j x_j + v = b,$$

$$x_j \geqslant 0,\ j \in J_0,\ v = (v_1, v_2, \ldots, v_m) \geqslant 0.$$

Известно, что если x_j^*, $j \in J_0$, $v^* = (v_1^*, v_2^*, \ldots, v_m^*)$ — решение этой задачи и $\sum_{i=1}^{m} v_i^* = 0$, то, полагая остальные $x_j^* = 0$, получаем решение исходной задачи.

В противном случае определяются векторы невязок $\Delta^{(0)} = \Lambda^{(0)} A - c$ и $\delta^* = \tilde{\Lambda}^* A$, где $\tilde{\Lambda}^*$ — вектор оптимальных значений двойственных переменных вспомогательной задачи. Если $\delta^* \leqslant 0$, то целевая функция задачи (1)—(3) неограничена снизу. Если вектор δ^* содержит положительные компоненты, то вычисляется величина $\mu_0 = \frac{\Delta_k^{(0)}}{\delta_k^*} = \max_{\delta_j^* > 0} \frac{\Delta_j^{(0)}}{\delta_j^*}$, находится новая допустимая точка $\Lambda^{(1)} = \Lambda^{(0)} - \mu_0 \tilde{\Lambda}^*$ двойственной задачи (7)—(8) и формируется новая вспомогательная задача.

Отметим, что для получения решения исходной вспомогательной задачи достаточно положить $v = b \geqslant 0$, $x_j = 0$, $j \in J_0$, и воспользоваться симплексным методом. В качестве начальной вершины новой вспомогательной задачи можно выбрать вершину, ненулевые координаты которой — положительные компоненты решения предыдущей вспомогательной задачи. При этом на первой итерации в матрицу базиса вводится вектор A_k.

Общим недостатком двойственных методов является то, что они вплоть до получения оптимальной точки не обеспечивают допустимость прямой задачи и не позволяют прекратить вычисления на какой-либо промежуточной итерации, ограничившись приближенным решением.

Кроме перечисленных, известны и другие методы решения задач линейного программирования, как конечные, так и итерационные.

4. РЕЛАКСАЦИЯ ОГРАНИЧЕНИЙ

Релаксация (ослабление) ограничений является одним из наиболее простых и широко распространенных декомпозиционных методов решения задач математического программирования. Согласно этому методу, вместо непосредственного решения исходной задачи решается накоторая последовательность так называемых релаксированных задач, каждая из которых (за возможным исключением последней) получается из исходной в результате удаления некоторых ограничений.

4.1. Прежде чем перейти к непосредственному изложению метода релаксации ограничений, введем в рассмотрение несколько новых понятий и сформулируем некоторые связанные с ними утверждения. При чтении этого параграфа предполагается предварительное ознакомление читателя с материалом, приведенным в п. 1.1 параграфа 1.

Функция $f(x)$ является *квазивыпуклой* на выпуклом множестве $X \subseteq R^n$, если $f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) \leqslant \max \{f(x^{(1)}), f(x^{(2)})\}$ для всех $x^{(1)}$, $x^{(2)} \in X$, $0 < \alpha < 1$. В этом случае множество точек $x \in X$, удовлетворяющих неравенству $f(x) \leqslant c$, или пусто, или выпукло при любом числе c.

Для *строго квазивыпуклой* на X функции выполняется неравенство $f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) < \max \{f(x^{(1)}), f(x^{(2)})\}$ для всех $x^{(1)}$, $x^{(2)} \in X$, $x^{(1)} \neq x^{(2)}$ и $0 < \alpha < 1$.

Для *полустрого квазивыпуклой* на X функции неравенство $f(\alpha x^{(1)} + (1-\alpha) x^{(2)}) < \max\{f(x^{(1)}), f(x^{(2)})\}$ выполняется для всех $x^{(1)}$, $x^{(2)} \in X$ таких, что $f(x^{(1)}) \neq f(x^{(2)})$ и $0 < \alpha < 1$.

Выпуклая функция является, очевидно, и квазивыпуклой, и полустрого квазивыпуклой.

Функция $f(x)$, определенная на (необязательно выпуклом) множестве X, *полунепрерывна снизу* в точке $x^{(0)} \in X$, если $\underline{\lim} f(x) = f(x^{(0)})$, $x \to x^{(0)}$, $x \in X$. Функция $f(x)$ полунепрерывна на множестве X, если она полунепрерывна в каждой точке множества X.

Рассмотрим *задачу квазивыпуклого программирования*:

$$f(x) \to \min,$$

$$g_i(x) \leqslant 0, \ i \in I = \{1, 2, \ldots, m\}, \tag{A}$$
$$x \in X,$$

где $x = (x_1, x_2, \ldots, x_n) \in R^n$, X — выпуклое множество, функция $f(x)$ — полустрого квазивыпуклая; функции $g_i(x)$ — непрерывные квазивыпуклые на X.

Предположим, что существует решение x^* этой задачи. Пусть $g_i(x^*) = 0$, $i \in \bar{I}$, $g_i(x^*) < 0$, $i \in I \setminus \bar{I}$ и $I / \bar{I} \neq \varnothing$.

Сформулируем задачу

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \ i \in \bar{I}, \tag{B}$$
$$x \in X.$$

Покажем, что x^* является решением задачи В.

Действительно, обозначим через $\tilde{X}$ — множество допустимых точек задачи В и предположим от противного, что существует точка $\bar{x} \in \tilde{X}$ такая, что $f(\bar{x}) < f(x^*)$. В силу выпуклости множество $\tilde{X}$ содержит все точки вида $x^\alpha = \alpha\bar{x} + (1-\alpha)x^*$, $0 \leqslant \alpha \leqslant 1$. Поскольку функция $f(x)$ полустрого квазивыпуклая, то $f(x^\alpha) < f(x^*)$ для всех $0 < \alpha < 1$. Поскольку функции $g_i(x)$, $i \in I$, непрерывны, то при достаточно малом значении $\alpha = \alpha_0 > 0$ выполняются неравенства $g_i(x^{\alpha_0}) \leqslant 0$, $i \in I \setminus \bar{I}$, т. е. x^{α_0} — допустимая точка задачи А и, следовательно, $f(x^*) \leqslant f(x^{\alpha_0})$. Получаем противоречие.

Покажем, что если x^* — единственное решение задачи А, то x^* — единственное решение задачи В.

Действительно, пусть x^* — единственное решение задачи А и существует точка $\bar{x} \in \tilde{X}$, $\bar{x} \neq x^*$, такая, что $f(\bar{x}) = f(x^*)$. Здесь через $\tilde{X}$ обозначено по-прежнему множество допустимых точек задачи В. Как показано выше, точка $x^\alpha = \alpha\bar{x} + (1-\alpha)x^*$ принадлежит $\tilde{X}$ при любом $0 \leqslant \alpha \leqslant 1$ и является допустимой точкой задачи А при $0 \leqslant \alpha \leqslant \alpha_0$, где α_0 — достаточно малое число. Поскольку x^* — единственное решение задачи А, то $f(x^*) < f(x^\alpha)$ для всех $0 \leqslant \alpha \leqslant \alpha_0$. Полагая $\alpha = \alpha_1$, $0 < \alpha_1 < \alpha_0$, в силу полустрогой квазивыпуклости функции $f(x)$ имеем $f(x^*) < f(x^{\alpha_1}) < f(x^{\alpha_0})$. Но $f(\bar{x}) = f(x^*)$ и, следовательно, $f(x^{\alpha_1}) > f(x^{\alpha_0}) > f(\bar{x})$. Получаем противоречие.

Отметим, что для существования решения задачи А достаточно потребовать, чтобы множество X было компактным, а функция $f(x)$ полунепрерывной снизу.

Пусть задача А является *задачей выпуклого программирования* (множество X — выпукло, функции $f(x)$ и $g_i(x)$, $i \in I$,— выпуклы на X) с регулярным множеством ограничений (существует точка $x' \in X$ такая, что $g_i(x') < 0$, $i \in I$).

Если x^* — решение этой задачи, то по известной теореме Куна — Таккера существуют такие числа $\lambda_i^* \geqslant 0$, $i \in I$ (оптимальные значения двойственных переменных), что

$$f(x^*) + \sum_{i \in I} \lambda_i^* g_i(x^*) \leqslant f(x) + \sum_{i \in I} \lambda_i^* g_i(x), \quad x \in X, \tag{1}$$

$$\lambda_i^* g_i(x^*) = 0, \quad i \in I. \tag{2}$$

Приведенные условия являются и достаточными условиями оптимальности точки x^* (при условии, что $g_i(x^*) \leqslant 0$, $i \in I$). Из этих условий непосредственно вытекает, что x^*— решение задачи В. Действительно, если $g_i(x^*) < 0$ при некотором $i \in I$, то из (2) следует, что $\lambda_i^* = 0$, что дает возможность в соотношениях (1)—(2) множество I заменить на $\bar{I}$.

Вообще если задача А является задачей выпуклого программирования, x^* — ее решение, Λ^* — вектор оптимальных значений двойственных переменных, то x^* — решение задачи

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \quad i \in \bar{I}', \\ x \in X, \end{gathered} \tag{B$'$}$$

где $\bar{I}'$ — множество всех тех $i \in I$, для которых $\lambda_i^* > 0$. Ясно, что $\bar{I} \subseteq \bar{I}'$.

4.2. *Схема релаксации ограничений для общей задачи* математического программирования приведена на рис. 4.1.

На первом шаге в исходной задаче

$$f(x) \to \min, \tag{3}$$

$$g_i(x) \leqslant 0, \quad i \in I, \tag{4}$$

$$x \in X \tag{5}$$

удаляются все ограничения (4), индексы которых принадлежат некоторому множеству $I \setminus I^{(1)}$. В частности, могут быть удалены вообще все ограничения (4).

Получаемая в результате релаксированная задача имеет вид

$$f(x) \to \min, \tag{6}$$

$$g_i(x) \leqslant 0, \ i \in I^{(1)}, \tag{7}$$

$$x \in X. \tag{8}$$

Если она неразрешима, то неразрешима, очевидно, и исходная задача. Если $x^{(1)}$ — решение задачи (6)—(8) и $g_i(x^{(1)}) \leqslant 0$, $i \in I \setminus I^{(1)}$, то $x^{(1)}$ — решение исходной задачи.

Если существует непустое множество $I_2^{(1)} \subseteq I \setminus I^{(1)}$ такое, что $g_i(x^{(1)}) > 0$, $i \in I_2^{(1)}$, то $x^{(1)}$ не является решением исходной задачи. В этом случае к ограничениям задачи (6)—(8) добавляется одно или несколько из нарушаемых в точке $x^{(1)}$ ограничений. Иными словами, выбирается некоторое множество $J_2^{(1)} \subseteq I_2^{(1)}$ и формируется новая релак-

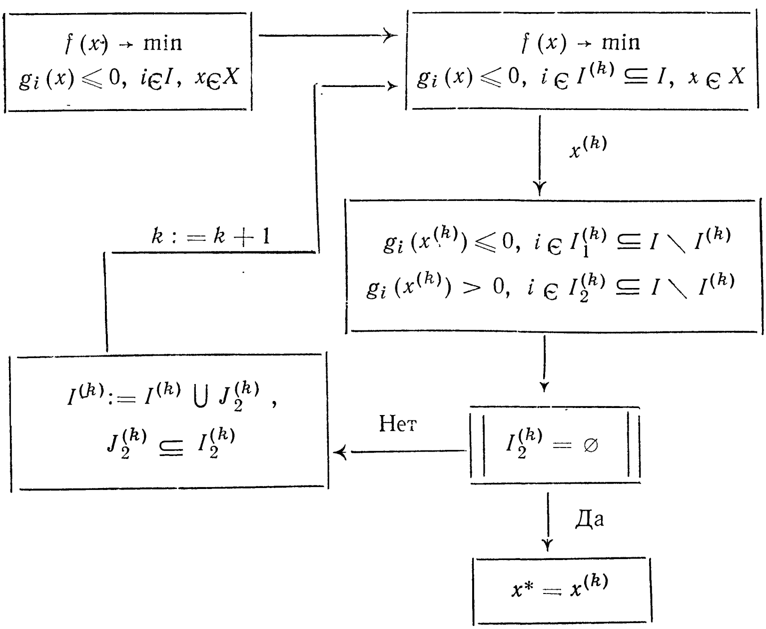

Рис. 4.1. Схема релаксации ограничений (общий случай)

сированная задача:

$$f(x) \to \min, \tag{9}$$

$$g_i(x) \leqslant 0, \ i \in I^{(2)} = I^{(1)} \cup J_2^{(1)}, \tag{10}$$

$$x \in X. \tag{11}$$

Относительно этой задачи можно повторить те же рассуждения. В результате либо будет установлена неразрешимость исходной задачи, либо получено ее решение, либо осуществится переход к новой релаксированной задаче с бо́льшим числом ограничений.

Поскольку на каждом шаге такого процесса ограничения только добавляются, то число шагов не превосходит m. Не исключается ситуация, когда последняя из «релаксированных» задач совпадает с исходной. В этом случае желаемого эффекта от применения метода релаксации, очевидно, не будет.

4.3. Для *задачи квазивыпуклого* программирования *схема релаксации ограничений* представлена на рис. 4.2.

Как и в общем случае на первом шаге решается задача (6)—(8) при некотором $I^{(1)} \subset I$ и, если выполняются приведенные выше условия, осуществляется переход к решению задачи (9)—(11) при некотором $I^{(2)} = I^{(1)} \cup J_2^{(1)}$, $J_2^{(1)} \subseteq I_2^{(1)}$.

Пусть $x^{(2)}$ — решение задачи (9) — (11). Если $g_i(x^{(2)}) \leqslant 0$, $i \in I \setminus I^{(2)}$, то $x^{(2)}$ — решение исходной задачи. В противном случае существует непустое множество $I_2^{(2)} \subseteq I \setminus I^{(2)}$ такое, что $g_i(x^{(2)}) > 0$, $i \in I_2^{(2)}$. Выберем некоторое множество $J_2^{(2)} \subseteq I_2^{(2)}$.

Если $f(x^{(2)}) = f(x^{(1)})$, то очередная релаксированная задача имеет вид

$$f(x) \to \min, \tag{12}$$

$$g_i(x) \leqslant 0, \ i \in I^{(3)} = I^{(2)} \cup J_2^{(2)}, \tag{13}$$

$$x \in X. \tag{14}$$

Если $f(x^{(2)}) > f(x^{(1)})$, то в полученной задаче (12)—(14) удаляются все ограничения (13), удовлетворяющие условию $g_i(x^{(2)}) < 0$. Обозначая множество индексов таких ограничений через $\tilde{I}^{(2)}$, очередную релаксированную задачу можно записать в виде

$$f(x)\to\min, \tag{15}$$

$$g_i(x)\leqslant 0,\ \ i\in I^{(3)}=(I^{(2)}\setminus\tilde{I}^{(2)})\cup J_2^{(2)}, \tag{16}$$

$$x\in X. \tag{17}$$

Поскольку исходная задача, а следовательно, и задача (12)—(14) являются задачами квазивыпуклого программирования, то, как показано выше, если $x^{(3)}$ — решение задачи (12)—(14), то $x^{(3)}$ — решение задачи (15)—(17).

Процесс повторяется до тех пор, пока не будет получено решение исходной задачи либо установлена ее неразрешимость. Каждая последующая релаксированная

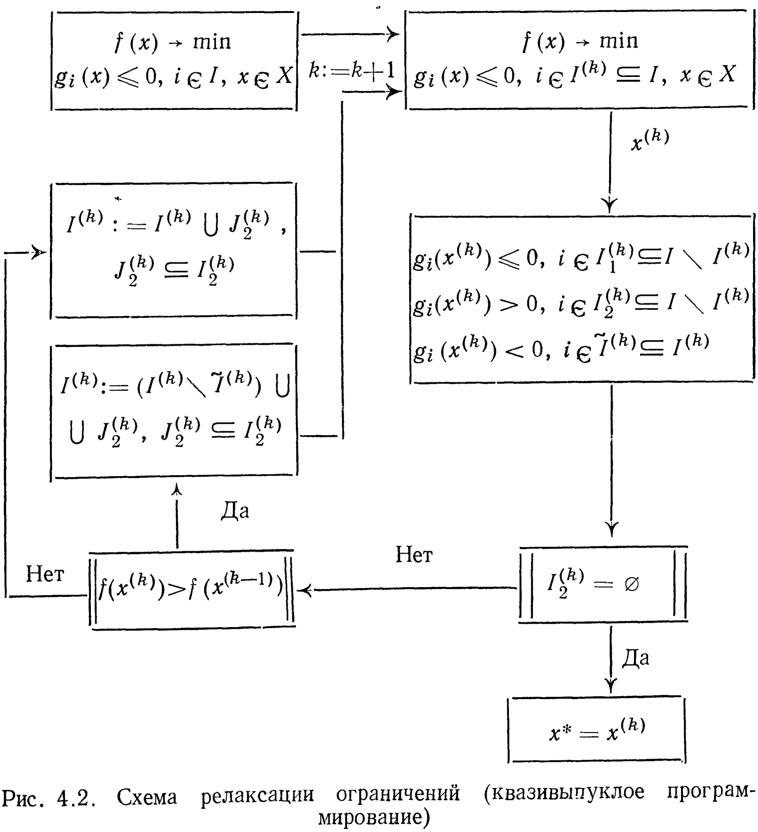

Рис. 4.2. Схема релаксации ограничений (квазивыпуклое программирование)

задача получается из предыдущей в результате добавления одного или нескольких новых ограничений. Удаление несущественных ограничений производится только в том случае, когда наблюдается увеличение оптимального значения целевой функции по сравнению с предыдущей задачей. Это обеспечивает конечность процесса релаксации ограничений.

При решении задач *выпуклого* программирования с использованием двойственных методов (обеспечивающих получение оптимальных значений двойственных переменных) в описанной выше ситуации можно удалять все ограничения с нулевыми λ_i^* (рис. 4.3).

В любом случае каждое очередное оптимальное значение целевой функции оказывается не меньше предыдущего: $f(x^{(k+1)}) \geqslant f(x^{(k)})$.

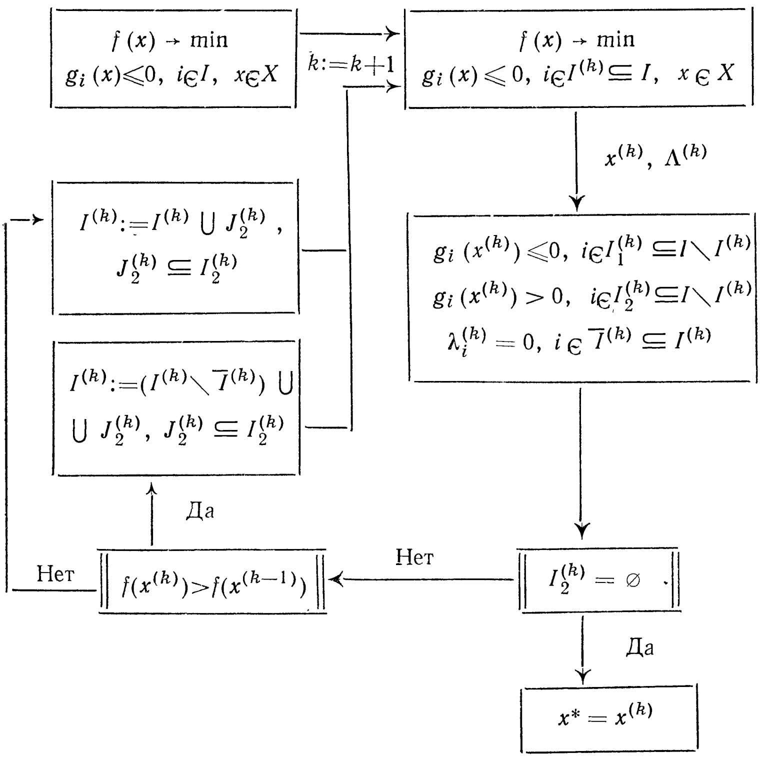

Рис. 4.3. Схема релаксации ограничений (выпуклое программирование)

4.4. Пример. Рассмотрим задачу линейного программирования вида

$$f(x)=x_1+x_2+6x_3+2x_4\to\min,$$

$$g_1(x)=x_1+x_2+2x_3+4x_4-12=0,$$

$$g_2(x)=x_1-x_2-2x_3+2x_4-6=0,$$

$$g_3(x)=-x_1\leqslant 0,$$

$$g_4(x)=-x_2\leqslant 0,$$

$$g_5(x)=-x_3\leqslant 0,$$

$$g_6(x)=-x_4\leqslant 0.$$

Удалим четыре последние ограничения (условия неотрицательности переменных). Из первых двух ограничений получаем

$$x_1=9-3x_4,\ \ x_2=3-2x_3-x_4.$$

Подставляя полученные выражения в целевую функцию, первую релаксированную задачу можно записать в виде

$$f(x)=12+4x_3-2x_4\to\min.$$

Очевидно, $f(x)\to-\infty$ при $x_3^{(1)}\to-\infty$, $x_4^{(1)}\to\infty$.

Добавляя ограничение $g_5(x)\leqslant 0$, т. е. $x_3\geqslant 0$, получаем $f(x)\to-\infty$ при $x_3^{(2)}=0$ и $x_4^{(2)}\to\infty$. При этом $x_1^{(2)}\to-\infty$ и $x_2^{(2)}\to-\infty$.

Добавляя ограничение $g_3(x)\leqslant 0$, т. е. $x_1=9-3x_4\geqslant 0$, или, что то же, $x_4\leqslant 3$, очередную релаксированную задачу можно записать в виде

$$f(x)=12+4x_3-2x_4\to\min,$$

$$x_3\geqslant 0,\ \ x_4\leqslant 3.$$

Решением этой задачи являются $x_3^{(3)}=0$ и $x_4^{(3)}=3$. При этом $x_1^{(3)}=0$ и $x_2^{(3)}=0$.

Полученные значения переменных являются решением исходной задачи.

4.5. Пример. Рассмотрим задачу линейного программирования вида

$$f(x)=-3x_1-x_2\to\min,$$

$$g_1(x) = 2x_1 + x_2 - 2 \leqslant 0,$$
$$g_2(x) = x_1 + 2x_2 - 2 \leqslant 0,$$
$$g_3(x) = 2x_1 - 2x_2 + 1 \leqslant 0,$$
$$x \in X = \{x = (x_1, x_2) \mid x_1 \geqslant 0,\ x_2 \geqslant 0\}.$$

Выберем в качестве первой релаксированной задачи следующую:

$$f(x) \to \min,\ g_2(x) \leqslant 0,\ x \in X.$$

Решением этой задачи является точка $x^{(1)} = (2, 0)$, значение $f(x^{(1)}) = -6$. В точке $x^{(1)}$ нарушаются ограничения $g_1(x) \leqslant 0$ и $g_3(x) \leqslant 0$. Добавим первое из них. Имеем

$$f(x) \to \min,\ g_1(x) \leqslant 0,\ g_2(x) \leqslant 0,\ x \in X.$$

Решением этой задачи является точка $x^{(2)} = (1, 0)$, значение $f(x^{(2)}) = -3$. В точке $x^{(2)}$ нарушается ограничение $g_3(x) \leqslant 0$. Поскольку $g_1(x^{(2)}) = 0$, $g_2(x^{(2)}) < 0$ и $f(x^{(2)}) > f(x^{(1)})$, то, добавляя ограничение $g_3(x) \leqslant 0$, можно удалить ограничение $g_2(x) \leqslant 0$. Очередная релаксированная задача запишется в виде

$$f(x) \to \min,\ g_1(x) \leqslant 0,\ g_3(x) \leqslant 0,\ x \in X.$$

Решением этой задачи является точка $x^{(3)} = (1/2, 1)$, значение $f(x^{(3)}) = -2\dfrac{1}{2}$. В точке $x^{(3)}$ нарушается ограничение $g_2(x) \leqslant 0$. Значения $g_1(x^{(3)}) = 0$, $g_3(x^{(3)}) = 0$. Добавляя ограничение $g_2(x) \leqslant 0$, получаем исходную задачу и, следовательно, выбранная схема релаксации неэффективна.

Если к первой релаксированной задаче добавить ограничение $g_3(x) \leqslant 0$ (вместо $g_1(x) \leqslant 0$), то решение полученной задачи, как нетрудно убедиться, явилось бы и решением исходной.

5. РАСЧЛЕНЕНИЕ ОГРАНИЧЕНИЙ

Описываемый в данном параграфе декомпозиционный подход обычно применяется при решении задач математического программирования с большим числом переменных и относительно небольшим числом ограничений, связывающих все или большую часть переменных.

5.1. Рассмотрим *параметрическую задачу выпуклого программирования* вида

$$f(x)\to\min\ (=f^*(y)),\tag{1}$$

$$g_i(x)\leqslant y_i,\quad i=\overline{1,\, m},\tag{2}$$

$$x\in X,\tag{3}$$

$$y\in Y,\tag{4}$$

где $x=(x_1,\, x_2,\, \ldots,\, x_n)\in R^n$; $y=(y_1,\, y_2,\, \ldots,\, y_m)\in R^m$; $X\subseteq R^n$ и $Y\subseteq R^m$ — выпуклые множества; $f(x)$, $g_i(x)$, $i=\overline{1,\, m}$,— выпуклые функции на X.

Предполагается, что при любом $y\in Y$ задача имеет решение. Обозначим его через $x^*(y)$, а соответствующее значение целевой функции через $f^*(y)$.

Покажем, что $f^*(y)$ — *выпуклая на Y функция.*

Действительно, выбирая произвольные точки $y^{(1)}$, $y^{(2)}\in Y$ и полагая $x^\alpha=\alpha x^*(y^{(1)})+(1-\alpha)x^*(y^{(2)})$, $0\leqslant\alpha\leqslant 1$, имеем $g_i(x^\alpha)=g_i(\alpha x^*(y^{(1)})+(1-\alpha)x^*(y^{(2)}))\leqslant \alpha g_i(x^*(y^{(1)}))+(1-\alpha)g_i(x^*(y^{(2)}))\leqslant \alpha y_i^{(1)}+(1-\alpha)y_i^{(2)}$ для всех $i=\overline{1,\, m}$ и $0\leqslant\alpha\leqslant 1$.

Поскольку множество X выпуклое и точки $x^*(y^{(1)})$ и $x^*(y^{(2)})$ принадлежат X, то $x^\alpha\in X$. Следовательно, точка x^α удовлетворяет ограничениям (2)—(3) при $y=\alpha y^{(1)}+(1-\alpha)y^{(2)}$, $0\leqslant\alpha\leqslant 1$. Отсюда и из выпуклости функции $f(x)$ на X имеем $f^*(\alpha y^{(1)}+(1-\alpha)y^{(2)})\leqslant f(x^\alpha)\leqslant \alpha f(x^*(y^{(1)}))+(1-\alpha)f(x^*(y^{(2)}))=\alpha f^*(y^{(1)})+(1-\alpha)\times f^*(y^{(2)})$ и, следовательно, $f^*(y)$ — выпуклая на Y функция.

Отметим, что если среди ограничений (2) существует хотя бы одно ограничение вида $g_i(x)=y_i$, то функция $f^*(y)$ необязательно выпуклая на Y.

5.2. Примеры.

1. Пусть $x=(x_1, x_2)\in R^2$, $X=\{x\,|\,x_1\geqslant 0,\ x_2\geqslant 0\}$, $f(x)=$ $=x_1^2-4x_2$, $g_1(x)=x_1^2+x_2^2$, $y\in R^1$, $Y=\{y\,|\,y\geqslant 0\}$.

При фиксированном $y\in Y$ получаем задачу выпуклого программирования

$$x_1^2-4x_2\to\min\ (=f^*(y)),$$

$$x_1^2+x_2^2\leqslant y,$$

$$x_1\geqslant 0,\quad x_2\geqslant 0.$$

Допустимая область этой задачи представлена на рис. 5.1. Ее решением, очевидно, является точка $x^*(y)=$ $=(x_1^*(y),\ x_2^*(y))=(0,\ \sqrt{y})$. Оптимальное значение целевой функции равно $f^*(y)=-4\sqrt{y}$. Функция $f^*(y)$ выпуклая на указанном выпуклом множестве Y.

2. Пусть $x\in R^1$, $X=\{x\,|\,x\geqslant 0\}$, $f(x)=x$, $g_1(x)=x^2$, $y\in$ $\in R^1$, $Y=\{y\,|\,y\geqslant 0\}$ и задача состоит в нахождении минимального значения $f^*(y)$ функции $f(x)$ при ограничениях $g_1(x)=y$ и $x\in X$ для всех $y\in Y$.

В рассматриваемом случае функции $f(x)$ и $g_1(x)$ выпуклые на X, а функция $f^*(y)=\sqrt{y}$ вогнутая на Y.

5.3. Рассмотрим *параметрическую задачу линейного программирования* вида

$$(c,\ x)\to\min\ (=f^*(y)), \tag{5}$$

$$Ax=b+b^{(0)}y, \tag{6}$$

$$x\geqslant 0,\quad y\in Y, \tag{7}$$

где $x=(x_1, x_2, \ldots, x_n)\in R^n$, $y=(y_1, y_2, \ldots, y_m)\in R^m$; A — матрица размерности $m\times n$; $c\in R^n$, b, $b^{(0)}\in R^m$; Y — выпуклое множество. По-прежнему предполагается, что при любом $y\in Y$ соответствующая задача линейного программирования имеет решение.

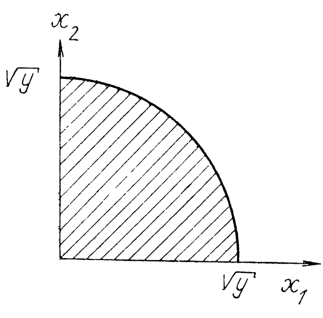

Рис. 5.1. Допустимая область задачи

Покажем, что $f^*(y)$ — *выпуклая, кусочно-линейная функция на* Y.

Выберем произвольные точки $y^{(1)}, y^{(2)} \in Y$ и изучим поведение функции $f^*(y)$ на отрезке, соединяющем эти точки: $y^\alpha = \alpha y^{(1)} + (1-\alpha) y^{(2)}$, $0 \leqslant \alpha \leqslant 1$. Обозначая $f^*(y^\alpha)$ через $\varphi(\alpha)$, имеем (при фиксированном α, $0 \leqslant \alpha \leqslant 1$)

$$(c, x) \to \min\ (=\varphi(\alpha)),$$

$$Ax = b^{(1)} + b^{(2)}\alpha,$$

$$x \geqslant 0,$$

где $b^{(1)} = b + b^{(0)} y^{(2)}$ и $b^{(2)} = b^{(0)}(y^{(1)} - y^{(2)})$ — не зависящие от α векторы.

Двойственной к данной задаче является следующая:

$$(b^{(1)} + b^{(2)}\alpha, \Lambda) \to \max\ (=\varphi(\alpha)), \tag{8}$$

$$A^\tau \Lambda \leqslant c \tag{9}$$

($\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ — вектор двойственных переменных; τ — знак транспонирования).

В последней задаче допустимая область, описываемая соотношением (9), не зависит от α. Целевая функция (8) при фиксированном α линейна по Λ, а при фиксированном Λ линейна по α. Обозначим через $\Lambda^*(\alpha)$ решение этой задачи (при фиксированном α). При изменении α от 0 до 1 можно выделить конечное число значений α, равных $\alpha_0, \alpha_1, \ldots, \alpha_r$ $(0 = \alpha_0 < \alpha_1 < \ldots < \alpha_r = 1)$ и удовлетворяющих условию: если $\Lambda^*(\alpha_k) = \Lambda^{(k)}$, то $\Lambda^*(\alpha) = \Lambda^{(k)}$ для всех α, $\alpha_k \leqslant \alpha < \alpha_{k+1}$, $0 \leqslant k \leqslant r-1$. На каждом из интервалов $[\alpha_k, \alpha_{k+1})$, $k = \overline{0, r-1}$, функция $\varphi(\alpha)$, очевидно, линейна и может быть представлена в виде $\varphi(\alpha) = v_k \alpha + w_k$, где $v_k = (b^{(2)}, \Lambda^{(k)})$ и $w_k = (b^{(1)}, \Lambda^{(k)})$.

Для произвольного k, $1 \leqslant k \leqslant r-1$, и α', $\alpha_k < \alpha' \leqslant \alpha_{k+1}$ имеем $v_k \alpha_k + w_k = v_{k-1}\alpha_k + w_{k-1}$ и $v_k \alpha' + w_k > v_{k-1}\alpha' + w_{k-1}$. Откуда $v_k > v_{k-1}$, т. е. тангенс угла наклона отрезка прямой $\varphi(\alpha)$ в интервале $[\alpha_{k-1}, \alpha_k)$ меньше тангенса угла наклона отрезка прямой $\varphi(\alpha)$ в интервале $[\alpha_k, \alpha_{k+1})$. Следовательно, функция $\varphi(\alpha)$ — выпуклая при $0 \leqslant \alpha \leqslant 1$.

5.4. Пример. Рассмотрим однопараметрическую задачу линейного программирования

$$-2x_1 - x_2 \to \min\ (=f^*(y)),$$

$$x_1+x_2\leqslant 6-y,$$
$$x_1-x_2\leqslant 3,$$
$$x_1+2x_2\geqslant 2,$$
$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad 1\leqslant y\leqslant 5.$$

Допустимая область этой задачи (в зависимости от значения y) изображена на рис. 5.2.

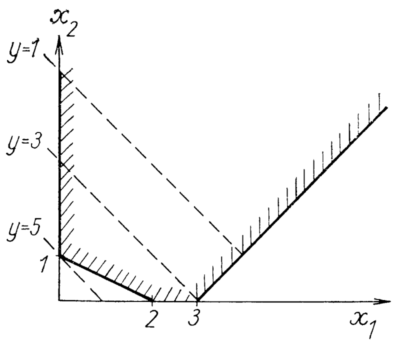

Рис. 5.2. Допустимая область задачи

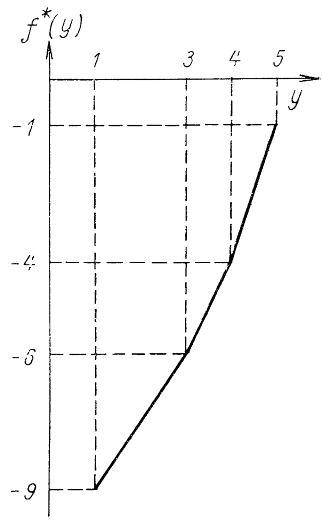

Рис. 5.3. График функции $f^*(y)$

При $4<y\leqslant 5$ $x_1^*(y)=10-2y$, $x_2^*(y)=y-4$, $f^*(y)=3y-16$; при $3<y\leqslant 4$ $x_1^*(y)=6-y$, $x_2^*(y)\equiv 0$, $f^*(y)=2y-12$; при $1\leqslant y\leqslant 3$ $x_1^*(y)=4\frac{1}{2}-\frac{y}{2}$, $x_2^*(y)=1\frac{1}{2}-\frac{y}{2}$, $f^*(y)=\frac{3y}{2}-10\frac{1}{2}$.

График функции $f^*(y)$ при $1\leqslant y\leqslant 5$ представлен на рис. 5.3.

5.5. *Метод расчленения ограничений* получил широкое распространение при решении задач сепарабельного программирования вида

$$\sum_{l=1}^{\nu} f_l(\tilde{x}_l) \to \min, \tag{10}$$

$$\sum_{l=1}^{\nu} g_i^{(l)}(\tilde{x}_l) \leqslant 0, \quad i=\overline{1, m}, \tag{11}$$

$$\tilde{x}_l \in X_l, \quad l=\overline{1, \nu}, \tag{12}$$

где $x=(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_\nu) \in R^n$; $\tilde{x}_l=(x_{k+1}, x_{k+2}, \ldots, x_{k+n_l}) \in \in R^{n_l}$; $k=\sum_{j=1}^{l-1} n_j$; множества $X_l \subseteq R^{n_l}$, $l=\overline{1, \nu}$.

Введем в рассмотрение $m\nu$ новых переменных — параметров y_{il}, $1 \leqslant i \leqslant m$, $1 \leqslant l \leqslant \nu$.

Представим ограничения (11) в виде

$$g_i^{(l)}(\tilde{x}_l) \leqslant y_{il}, \quad i=\overline{1, m}, \quad l=\overline{1, \nu}, \tag{13}$$

$$\sum_{l=1}^{\nu} y_{il} \leqslant 0, \quad i=\overline{1, m}. \tag{14}$$

При фиксированных значениях параметров y_{il} получаем ν *локальных* задач $(l=\overline{1, \nu})$:

$$f_l(\tilde{x}_l) \to \min \ (=f_l^*(y_l)), \tag{15}$$

$$g_i^{(l)}(\tilde{x}_l) \leqslant y_{il}, \quad i=\overline{1, m}, \tag{16}$$

$$\tilde{x}_l \in X_l, \tag{17}$$

где через $f_l^*(y_l)$ обозначено оптимальное значение целевой функции при $y_l=(y_{1l}, y_{2l}, \ldots, y_{ml})$, $l=\overline{1, \nu}$.

Решение исходной задачи сводится к решению задачи

$$f^*(y)=\sum_{l=1}^{\nu} f_l^*(y_l) \to \min, \tag{18}$$

$$\sum_{l=1}^{\nu} y_{il} \leqslant 0, \quad i=\overline{1, m}. \tag{19}$$

Здесь $y=(y_1, y_2, \ldots, y_\nu)$.

Отметим, что во всех или части ограничений (19) знак нестрогого неравенства можно заменить знаком равенства.

В приведенной задаче $m\nu$ переменных y_{il}, m линейных ограничений с единичными коэффициентами при переменных. Функция $f^*(y)$ задана неявно: для вычисления ее значения при фиксированном значении y требуется решить ν задач (15)—(17). Задача (18)—(19) обычно называется *координирующей.*

Схема расчленения ограничений представлена на рис. 5.4.

Отметим, что если в левой части ограничений (11) некоторое слагаемое $g_i^{(l)}(\tilde{x}_l) \equiv 0$, то нет необходимости вводить соответствующий параметр y_{il}. Поэтому эффективность метода расчленения ограничений обычно тем выше, чем больше в ограничениях задачи тождественно равных нулю слагаемых.

Если исходная задача (10)—(12) является задачей выпуклого программирования (множества X_l выпуклые, функции $f_l(\tilde{x}_l)$ и $g_i^{(l)}(\tilde{x}_l)$, $i=\overline{1, m}$, выпуклые на X_l, $l=\overline{1, \nu}$), то функции $f_l^*(y_l)$, $l=\overline{1, \nu}$, а следовательно, и

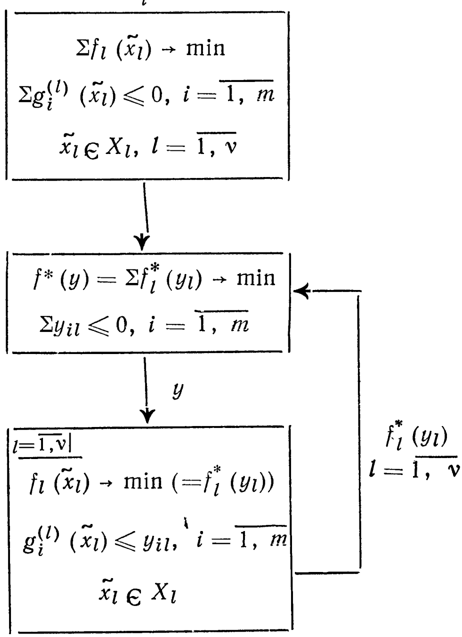

Рис. 5.4. Схема расчленения ограничений

функция $f^*(y)$ выпуклые на выпуклом множестве Y, описываемом ограничениями (19).

Функция $f^*(y)$, вообще говоря, недифференцируема на множестве Y. Поэтому для ее минимизации обычно используют методы обобщенного градиентного спуска.

5.6. Примеры.

1. Рассмотрим задачу

$$-x_1x_2-x_3^2\to\min,$$
$$x_3^2+x_3\leqslant 2,$$
$$x_1+x_2+x_3\leqslant 4,$$
$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad x_3\geqslant 0.$$

Вводя параметры y_1 и y_2, ограничения этой задачи можно переписать в виде

$$x_3^2+x_3\leqslant 2,$$
$$x_1+x_2\leqslant y_1,$$
$$x_3\leqslant y_2,$$
$$y_1+y_2\leqslant 4,$$
$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad x_3\geqslant 0.$$

При фиксированных значениях параметров y_1 и y_2 исходная задача распадается на две локальные (см. (15)—(17))

$$-x_1x_2\to\min\ (=f_1^*(y_1)),\qquad -x_3^2\to\min\ (=f_2^*(y_2)),$$
$$x_1+x_2\leqslant y_1,\qquad x_3^2+x_3\leqslant 2,$$
$$x_1\geqslant 0,\quad x_2\geqslant 0,\qquad 0\leqslant x_3\leqslant y_2.$$

Координирующая задача записывается в виде (см. (18)—(19))

$$f^*(y_1,\,y_2)=f_1^*(y_1)+f_2^*(y_2)\to\min,$$
$$y_1+y_2\leqslant 4,$$
$$y_1\geqslant 0,\quad y_2\geqslant 0.$$

Решая первую локальную задачу при $y_1\geqslant 0$, получаем

$x_1^*(y_1)=x_2^*(y_1)=\frac{y_1}{2}$, $f_1^*(y_1)=-\frac{y_1^2}{4}$. Решая вторую локальную задачу при $y_2\geqslant 0$, получаем $x_3^*(y_2)=y_2$ и $f_2^*(y_2)=-y_2^2$, если $y_2\leqslant 1$, и $x_3^*(y_2)\equiv 1$ и $f_2^*(y_2)\equiv -1$, если $y_2>1$.

Следовательно, $f^*(y_1, y_2)=-\frac{y_1^2}{4}-y_2^2$, если $y_2\leqslant 1$, и

$f^*(y_1, y_2)=-\frac{y_1^2}{4}-1$, если $y_2>1$.

Решением координирующей задачи является $y_1^*=3$, $y_2^*=1$. Отсюда $x_1^*=x_2^*=\frac{3}{2}$, $x_3^*=1$.

2. Рассмотрим задачу линейного программирования

$$-x_1-3x_2-2x_3-x_4\to \min,$$

$$2x_1+x_2+x_3+x_4\leqslant 8,$$

$$x_1+x_2\leqslant 5,$$

$$x_3-x_4\leqslant 4,$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0.$$

Введем два параметра y_1 и y_2. Представим первое из ограничений в виде $2x_1+x_2\leqslant y_1$, $x_3+x_4\leqslant y_2$, $y_1+y_2=8$ или, учитывая, что $y_2=8-y_1$ и обозначая y_1 через y, в виде $2x_1+x_2\leqslant y$, $x_3+x_4\leqslant 8-y$.

Поскольку $x_j\geqslant 0$, $j=\overline{1, 4}$, то $0\leqslant y\leqslant 8$. Тем самым координирующую задачу можно записать в виде

$$f^*(y)=f_1^*(y)+f_2^*(y),\quad 0\leqslant y\leqslant 8,$$

где $f_1^*(y)$ — оптимальное значение целевой функции локальной задачи

$$-x_1-3x_2\to \min\ (=f_1^*(y)),$$

$$2x_1+x_2\leqslant y,$$

$$x_1+x_2\leqslant 5,$$

$$x_1\geqslant 0,\ x_2\geqslant 0;$$

$f_2^*(y)$ — оптимальное значение целевой функции локальной задачи

$$-2x_3-x_4\to \min\ (=f_2^*\ (y)),$$
$$x_3+x_4\leqslant 8-y,$$
$$x_3-x_4\leqslant 4,$$
$$x_3\geqslant 0,\quad x_4\geqslant 0$$

при фиксированном значении y, $0\leqslant y\leqslant 8$.

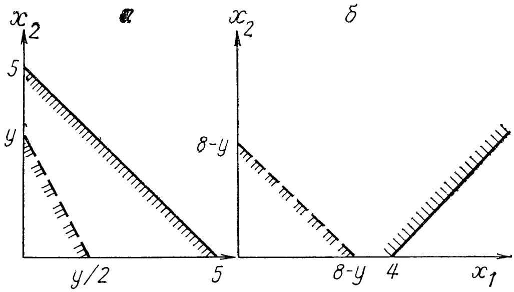

Рис. 5.5. Допустимые области локальных задач

Функция $f^*(y)$ является выпуклой, кусочно-линейной функцией одного аргумента. Для нахождения ее минимума на интервале [0, 8] может быть использован любой из методов минимизации унимодальных функций. Однако в рассматриваемом случае функцию $f^*(y)$ можно записать в явном виде.

На рис. 5.5 изображены допустимые области локальных задач. При $0\leqslant y\leqslant 5$ $x_1^*\ (y)\equiv 0$, $x_2^*\ (y)=y$, $f_1^*\ (y)=-3y$; при $5<y\leqslant 8$ $x_1^*\ (y)\equiv 0$, $x_2^*\ (y)\equiv 5$, $f_1^*(y)\equiv -15$; при $4\leqslant y\leqslant 8$ $x_3^*\ (y)=8-y$, $x_4^*\ (y)\equiv 0$, $f_2^*\ (y)=2y-16$; при $0\leqslant y<4$ $x_3^*\ (y)=6-\dfrac{y}{2}$, $x_4^*\ (y)=2-\dfrac{y}{2}$, $f_2^*\ (y)=\dfrac{3y}{2}-14$.

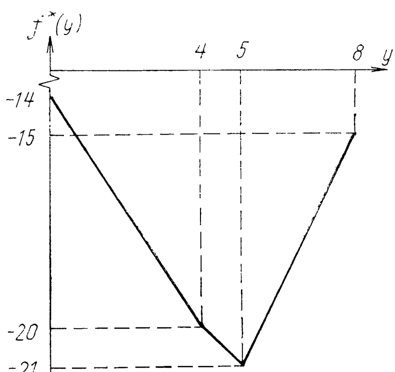

Рис. 5.6. График функции $f^*(y)$

Следовательно,

$$f^*(y) = \begin{cases} -\dfrac{3y}{2} - 14, & 0 \leqslant y < 4, \\ -y - 16, & 4 \leqslant y < 5, \\ 2y - 31, & 5 \leqslant y \leqslant 8. \end{cases}$$

График этой функции представлен на рис. 5.6.

Минимум функции $f^*(y)$ на интервале $[0, 8]$ достигается в точке $y^* = 5$. Следовательно, $x_1^* = x_1^*(5) = 0$, $x_2^* = x_2^*(5) = 5$, $x_3^* = x_3^*(5) = 3$ и $x_4^* = x_4^*(5) = 0$ — решение исходной задачи.

6. ПАРАМЕТРИЧЕСКАЯ ДЕКОМПОЗИЦИЯ

Описанный в предыдущем параграфе метод декомпозиции задач математического программирования состоит в специфическом преобразовании исходной задачи в некоторую новую задачу. Эта новая задача наряду с исходными переменными содержит дополнительные переменные — параметры. При фиксированных значениях параметров задача распадается на несколько задач меньшей размерности. Такой прием допускает естественное обобщение.

6.1. Пусть дана задача

$$f(x)\to\min\ (=f^*),\quad x\in X, \tag{A}$$

где X — некоторое множество (не обязательно из R^n); $f(x)$ — ограниченная снизу на X действительная функция.

Введем в рассмотрение множества Y, $Z\subseteq X\times Y$ и ограниченную снизу на Z действительную функцию $\tilde f(x, y)$.

Пара $\{Z, \tilde f(x, y)\}$ порождает экстремальную задачу

$$\tilde f(x, y)\to\min(=\tilde f^*),\quad (x, y)\in Z. \tag{B}$$

Предположим, что задачи А и В таковы, что если (x^*, y^*) — решение задачи В, то x^* — решение задачи А и $f^*=\tilde f^*$. В этом случае будем говорить, что задачи А и В являются *согласованными*. Решение исходной задачи А, очевидно, может быть получено в результате решения задачи В.

Решение задачи В в свою очередь может быть получено в результате решения (*координирующей*) задачи

$$\tilde f^*(y)\to\min,\quad y\in Y, \tag{B$''$}$$

где $\tilde f^*(y)$ — оптимальное значение целевой функции в *локальной* задаче

$$\begin{gathered}\tilde f(x, y)\to\min\ (=\tilde f^*(y)),\\ x\in X(y),\quad y=\mathrm{const}\in Y.\end{gathered} \tag{B$'$}$$

Здесь $X(y)$ — соответствующее сечение множества Z (т. е. множество точек $(x, y)\in Z$, где $y=\mathrm{const}\in Y$).

В условиях предыдущего параграфа задача (10) — (12) является задачей А (функция $f(x)$ записана в виде

(10), множество X описывается соотношениями (11)—(12)); задача (10), (12)—(14) является задачей В (функция $\tilde{f}(x, y) \equiv f(x)$, множество Y описывается ограничениями (14); множество Z — ограничениями (12)—(14); совокупность задач (15)—(17) представляет собой задачу В′ (функция $\tilde{f}(x, y) \equiv f(x)$, множество $X(y)$ описывается совокупностью ограничений (16)—(17)); наконец, задача (18)—(19) является задачей В″.

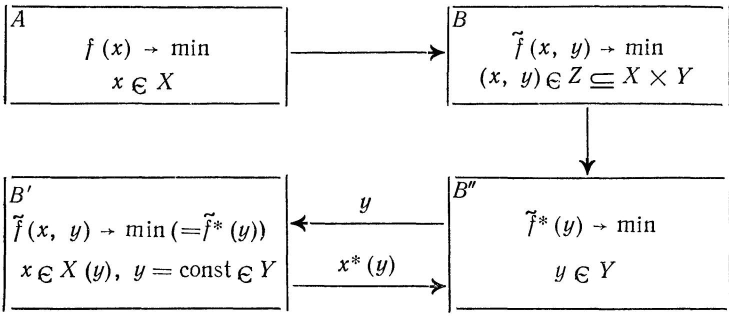

Рис. 6.1. Схема параметрической декомпозиции

Схема параметрической декомпозиции представлена на рис. 6.1. Через $x^*(y)$ обозначено решение задачи В′ при фиксированном значении y.

Для одной и той же задачи А может быть сформулирован целый спектр задач типа В (согласованных с исходной задачей А). Выбор конкретной задачи определяется двумя естественными, но зачастую противоречивыми тенденциями. Поскольку поиск решения задачи В″, как правило, сопряжен с необходимостью многократного решения задачи В′ (при различных значениях y), желательно получить возможно более «простую» задачу В′, в частности, распадающуюся на ряд автономных, хорошо изученных, «стандартных» подзадач. С другой стороны, эффективность поиска решения задачи В″ в значительной мере определяется тем, насколько сложна сама эта задача.

6.2. При систематическом использовании указанного приема для решения разнообразных экстремальных задач (в том числе задач математического программирования) целесообразно располагать *условиями*, соблюдение которых гарантировало бы согласованность задач А и В.

Следует также ограничиться рассмотрением класса задач типа В, двухуровневая схема решения которых позволяла бы получать по возможности более «простые» области поиска Y и «хорошо ведущие на них» функции $\tilde{f}^*(y)$. Эти условия желательно сформулировать таким образом, чтобы обеспечить возможность их практической проверки и указать конструктивные приемы формирования задач типа В по исходной задаче А.

Пусть задачи А и В таковы, что существует однозначное отображение ω множества X на множество $\bar{Y} \subseteq Y$ и выполняются следующие условия:

1°. $\tilde{f}(x, \omega(x)) = f(x)$ для всех $x \in X$;

2°. $\tilde{f}(x, y) \geqslant f(x)$ для всех $x \in X(y)$, $y \in Y$;

3°. $\tilde{f}^*(y) \leqslant \varphi(y) = \min\{\tilde{f}(x, y) \mid x \in \omega^{-1}(y)\}$ для всех $y \in \bar{Y}$.

Предполагается, что функция $\tilde{f}(x, y)$ определена на $X \times Y$ и достигает наименьшего значения на множествах $X(y)$ и $\omega^{-1}(y)$.

Покажем, что выполнение указанных условий гарантирует согласованность задач А и В.

Обозначим через $X^*(y)$ множество решений задачи В′ при фиксированном y (т. е. множество всех $x \in X(y)$ таких, что $\tilde{f}(x, y) = \tilde{f}^*(y)$).

Если $y \in Y$, $x \in X^*(y)$ и $y' = \omega(x)$, то $\tilde{f}^*(y) \geqslant f(x) \geqslant \geqslant \varphi(y') \geqslant \tilde{f}^*(y')$. Действительно, по условию 3° значение $\tilde{f}^*(y') \leqslant \varphi(y')$. По определению и условию 1° значение $\varphi(y') \leqslant \tilde{f}(x, y') = \tilde{f}(x, \omega(x)) = f(x)$. Наконец, по условию 2° значение $f(x) \leqslant \tilde{f}(x, y) = \tilde{f}^*(y)$.

Покажем, что если $\tilde{f}^*(y) = \tilde{f}^*$ и $x \in X^*(y)$, то $f(x) = f^*$ и значение $f^* = \tilde{f}^*$. По определению и, согласно условиям 1°—3°, для произвольного $x \in X$ имеем $f(x) = \tilde{f}(x, \omega(x)) \geqslant \geqslant \varphi(\omega(x)) \geqslant \tilde{f}^*(\omega(x)) \geqslant \tilde{f}^*$. Для $x \in X^*(y)$ (при условии, что $\tilde{f}^*(y) = \tilde{f}^*$) имеем $\tilde{f}^* = \tilde{f}^*(y) \geqslant f(x) \geqslant \tilde{f}^*(\omega(x)) \geqslant \tilde{f}^*$ и, следовательно, значение $f^* = f(x) = \tilde{f}^*$.

Нетрудно показать, что аналогичный результат получается и в том случае, когда минимумы в задачах А и В″ не достигаются. Если $\lim\limits_{k\to\infty} \tilde{f}^*(y^{(k)}) = \tilde{f}^*$ и $x^{(k)} \in X^*(y^{(k)})$, то $\lim\limits_{k\to\infty} f(x^{(k)}) = f^*$. Значение $f^* = \tilde{f}^*$.

Таким образом, при формировании задачи В по заданной задаче А достаточно обеспечить выполнение условий 1°—3°. При этом допускается существенное видоизменение как целевых функций, так и множеств допустимых значений переменных.

Рассмотрим две *простейшие крайние ситуации*, в первой из которых производится разбиение множества X на подмножества без изменения целевой функции задачи A, во второй целевая функция видоизменяется в каждой рассматриваемой точке $x \in X$ без изменения множества X.

1. Пусть $X(y) = \omega^{-1}(y)$ для всех $y \in \bar{Y} = Y$. В этом случае из условия 1° следует выполнимость условий 2° и 3°. Множество X разбивается на попарно пересекающиеся подмножества $X(y)$, $y \in Y$.

Задача B' состоит в нахождении наименьшего значения $f^*(y)$ функции $f(x)$ на множестве $X(y)$, $y = \mathrm{const} \in Y$. При решении задачи B'' находим y в Y, которому соответствует наименьшее значение функции $\tilde{f}^*(y)$.

Декомпозиция исходной задачи осуществляется непосредственным разбиением множества X допустимых значений x на подмножества.

2. Пусть $X = Y$, $X(y) = X$ и $\omega(x) = x^*$ для всех $x \in X$ и $y \in Y$. Здесь x^* — искомое решение задачи A. В этом случае для каждого $x' \in X$ выбирается функция $\tilde{f}(x, x')$, удовлетворяющая условиям 1° и 2°. Задача B′ состоит в нахождении наименьшего значения $\tilde{f}^*(x')$ этой функции на множестве X. В задаче B'' требуется минимизировать $f^*(x')$ при $x' \in X$.

Таким образом, в отличие от предыдущего случая множество допустимых значений X в задаче A не изменяется, целевая же функция может изменяться при переходе от одной точки множества к другой.

Решение задачи A получается в результате решения «цепочки» задач с различными целевыми функциями, что наблюдается, например, в случаях, когда производится первоначальное «огрубление» целевой функции с последующим ее «уточнением» по результатам решения задачи.

Во всех *известных схемах* параметрической декомпозиции задач математического программирования (см., например, предыдущий параграф) выполняется условие

$$3^\circ_1.\ \omega^{-1}(y) \subseteq X(y) \text{ для всех } y \in \bar{Y}$$

или, что то же,

$$3^\circ_2.\ \omega(x) \in Y(x) \text{ для всех } x \in X,$$

откуда непосредственно следует выполнимость условия 3°. Здесь $Y(x)$ — соответствующее сечение множества Z.

Покажем, что для выполнимости условия 3° выполнимость условия 3°_1 (3°_2) не является необходимой.

Рассмотрим следующий пример.

Задача A:

$$f(x)=\max(1/x_1,\ 1/x_2)+x_1+x_2\to\min,$$

$$X=\{x=(x_1,\ x_2)\,|\,1\leqslant x_1\leqslant 2,\ 1\leqslant x_2\leqslant 2\}.$$

Задача B:

$$\tilde f(x,\ y)=y+x_1+x_2,$$

$$Z=\{(x,\ y)\,|\,1\leqslant x_1\leqslant 2,\ 1\leqslant x_2\leqslant 2,\ y\leqslant 1,\ x_1y\geqslant 1,\ x_2y\geqslant 1,$$
$$-0{,}75x_1+x_2+y\geqslant 1\}.$$

Здесь $\omega(x)=\max(1/x_1,\ 1/x_2)$. Очевидно, выполняются условия 1° и 2°. Покажем, что выполняется и условие 3°. Действительно, $\varphi(y)=\min\{y+x_1+x_2\,|\,1\leqslant x_1\leqslant 2,\ 1\leqslant x_2\leqslant \leqslant 2,\ \max(1/x_1,\ 1/x_2)=y\}=y+2/y$ и точка $(x_1=1/y,\ x_2=1/y,\ y)$ при $1/2\leqslant y\leqslant 1$ принадлежит множеству Z. Следовательно, задачи A и B являются согласованными.

В то же время существует, например, при $y=1$ точка $x'=(x_1=2,\ x_2=1)$, такая, что $x'\in\omega^{-1}(1)$ и $x'\notin X(1)$.

6.3. Рассмотрим *некоторые особенности параметрической декомпозиции задач математического программирования.*

Будем предполагать, что множества X, Y и $X(y)$ описываются неравенствами $G(x)\leqslant 0$, $\Phi(y)\leqslant 0$ и $\tilde G(x,\ y)\leqslant 0$ соответственно. Вектор-функции G, Φ и $\tilde G$ предполагаются непрерывными на соответствующих множествах, а функция $\tilde f(x,\ y)$ достигает своего наименьшего значения на каждом замкнутом подмножестве множества X, $y\in Y$.

Исходная задача A состоит в минимизации функции $f(x)$ при $G(x)\leqslant 0$, задача B′ — в нахождении наименьшего значения $\tilde f^*(y)$ функции $\tilde f(x,\ y)$ при $\tilde G(x,\ y)\leqslant 0$, $y=\text{const}$. Наконец, в задаче B″ требуется минимизировать функцию $\tilde f^*(y)$ при $\Phi(y)\leqslant 0$.

В рассматриваемом случае условия 1°—3° целесообразно сформулировать в несколько иной, характерной для математического программирования терминологии.

По-прежнему будем предполагать, что существует однозначное отображение ω множества X во множество Y, для которого выполняется условие 1°, т. е. $\tilde{f}(x, \omega(x)) = f(x)$ для всех $x \in X$. Потребуем, чтобы, кроме этого условия, выполнялись следующие условия.

4°. Если $\Phi(y') \leqslant 0$, то система $\tilde{G}(x, y') \leqslant 0$ совместна, и если $\tilde{G}(x', y') \leqslant 0$, то $G(x') \leqslant 0$ и $\tilde{f}(x', y') \geqslant f(x')$.

5°. Если $G(x') \leqslant 0$ и $y' = \omega(x')$, то $\Phi(y') \leqslant 0$ и $\tilde{f}(x, y') \leqslant f(x')$ для некоторого x, такого, что $\tilde{G}(x, y') \leqslant 0$.

При выполнении условий 1°, 4°, 5° решение задачи A может быть получено в результате решения задачи B″. В свою очередь поиск решения задачи B″ требует, как правило, многократного решения задачи B′ при различных фиксированных значениях y.

Один из естественных *способов формирования* задач B′ и B″ по заданной исходной задаче A может быть описан следующим образом.

Пусть $G(x) = (g_1(x), g_2(x), \ldots, g_m(x))$. Обозначим $f(x)$ через $g_0(x)$. Каждую функцию $g_i(x)$ (включая функцию $g_0(x)$) представим в виде сложной функции $g_i(x) = \Gamma_i(\omega_{i1}, \omega_{i2}, \ldots, \omega_{i\nu_i})$ промежуточных параметров $\omega_{ij} = \omega_{ij}(x)$.

Обозначим множество всех (различных) функций ω_{ij} через $\Omega_0 = \{\omega_k \mid k = \overline{1, p_0}\}$, где $k = k(i, j)$ — номер функции ω_{ij} во множестве Ω_0.

Пусть Ω — произвольное подмножество Ω_0. Для определенности будем полагать $\Omega = \{\omega_k \mid k = \overline{1, p}\}$. Выбор Ω задает некоторое отображение ω множества X на множество $\bar{Y}$, при котором точка $x \in X$ отображается в точку $y = (y_1, y_2, \ldots, y_p) = \omega(x) = (\omega_1(x), \omega_2(x), \ldots, \omega_p(x))$.

Введем в рассмотрение функции $\tilde{g}_i(x, y) = \Gamma_i(z_{i1}, z_{i2}, \ldots, z_{i\nu_i})$, $i = \overline{0, m}$, где $z_{ij} = \omega_k(x)$, если $k = k(i, j) > p$, и $z_{ij} = y_k$ в противном случае.

Формирование вспомогательной задачи B проведем заменой целевой функции и всех или части ограничений исходной задачи некоторыми новыми ограничениями. Разобьем множество индексов $\{0, 1, \ldots, m\}$ на два подмножества I_1 и I_2, причем $0 \in I_1$ и необязательно $I_2 \neq \varnothing$.

В качестве целевой функции $\tilde{f}(x, y)$ в задаче B примем функцию $\tilde{g}_0(x, y)$, а в качестве ограничений $\tilde{g}_i(x, y) \leqslant 0$ при $i \in I_1 \setminus \{0\}$ и $g_i(x) \leqslant 0$ при $i \in I_2$. При этом если среди функций $\tilde{g}_i(x, y)$, $i \in I_1$, существуют неубывающие по y функции, вводим дополнительное ограничение $\omega(x)-y\leqslant 0$, а для каждого $i\in I_1$ — ограничение $g_i(x)-\tilde{g}_i(x, y)\leqslant 0$, если функция $\tilde{g}_i(x, y)$ не является монотонно возрастающей по y.

Функции, стоящие в левых частях ограничений задачи В, обычно можно разделить на две группы. К первой группе относятся те из них, которые зависят только от параметра y, а ко второй — остальные. В результате задачу В можно представить в виде $\tilde{f}(x, y)\to\min$ при ограничениях $\tilde{G}(x, y)\leqslant 0$ и $H(y)\leqslant 0$.

Пусть неравенства $Q(y)\leqslant 0$ описывают пересечение множества значений параметра y, удовлетворяющих неравенствам $H(y)\leqslant 0$, с проекцией множества, определяемого неравенствами $\tilde{G}(x, y)\leqslant 0$ на R^p.

Рассмотрим неравенства $\Phi(y)\leqslant 0$, удовлетворяющие условиям: если $y\in\omega(X)$, то $\Phi(y)\leqslant 0$, и если $\Phi(y)\leqslant 0$, то $Q(y)\leqslant 0$.

Тогда задача В$'$ представима в виде $\tilde{f}(x, y)\to\min$ при $\tilde{G}(x, y)\leqslant 0$ и фиксированном значении y, а задача В$''$ — в виде $\tilde{f}^*(y)\to\min$ при $\Phi(y)\leqslant 0$.

Покажем, что полученные таким образом задачи В$'$ и В$''$ удовлетворяют условиям 1°, 4°, 5° (следовательно, и условиям 1°—3°).

Действительно, по построению $\omega(x)$ — отображение множества X во множество Y и $\tilde{f}(x, \omega(x))=f(x)$ для всех $x\in X$.

Пусть $\Phi(y')\leqslant 0$. Тогда $Q(y')\leqslant 0$ и, следовательно, система $\tilde{G}(x, y')\leqslant 0$ совместна. Пусть $\tilde{G}(x', y')\leqslant 0$. Тогда, очевидно, $g_i(x')\leqslant 0$ для всех $i\in I_2$. Если существует неубывающая по y функция $\tilde{g}_i(x, y)$, $i\in I_1\setminus\{0\}$, то $g_i(x')=\tilde{g}_i(x', \omega(x'))\leqslant\tilde{g}_i(x', y')\leqslant 0$, поскольку в системе $\tilde{G}(x, y)\leqslant 0$ содержится ограничение $\omega(x)-y\leqslant 0$. Наконец, если функция $\tilde{g}_i(x, y)$, $i\in I_1\setminus\{0\}$, не является монотонно возрастающей по y, то $g_i(x')\leqslant\tilde{g}_i(x', y')\leqslant 0$. Аналогичным образом убеждаемся в справедливости соотношения $f(x')\leqslant\tilde{f}(x', y')$.

Пусть $G(x')\leqslant 0$ и $y'=\omega(x')$. По построению $\Phi(y')\leqslant 0$ и доказанному выше система $\tilde{G}(x, y')\leqslant 0$ совместна. В рассматриваемом случае $\tilde{G}(x', y')\leqslant 0$, т. е. $\omega^{-1}(y)\subseteq X(y)$, и, следовательно, $\tilde{f}(x, y')\leqslant f(x')$ для некоторого x, такого, что $\tilde{G}(x, y')\leqslant 0$. В частности, можно положить $x=x'$.

Таким образом, выполняются условия 1°, 4°, 5° и тем самым имеют место приведенные ранее утверждения относительно взаимосвязи исходной задачи А и получаемых задач В′ и В″.

6.4. Установим *условия (квази)выпуклости целевых функций и выпуклости допустимых областей* задач В′ и В″ в зависимости от свойств задачи А.

Пусть $X \subseteq R^n$, $Y \subseteq R^m$, Y — проекция $Z \subseteq X \times Y$ на R^m.

Если Z — выпуклое множество, то его проекция Y на множество R^m и любое сечение $X(y)$, $y \in Y$, очевидно, выпуклы.

Покажем, что если функция $\tilde{f}(x, y)$ выпукла на (выпуклом) множестве Z, то функция $\tilde{f}^*(y)$ выпукла на множестве Y.

Действительно, пусть $y^{(1)}, y^{(2)} \in Y$, число $0 < \alpha < 1$, $y = \alpha y^{(1)} + (1-\alpha) y^{(2)}$, $x^{(1)} \in X(y^{(1)})$, $x^{(2)} \in X(y^{(2)})$, $x = \alpha x^{(1)} + (1-\alpha) x^{(2)}$. Поскольку Z — выпуклое множество, то $(x, y) \in Z$, Y — выпуклое множество и, следовательно, $y \in Y$, $X(y)$ — выпуклое множество и, следовательно, $x \in X(y)$. Имеем $\tilde{f}(x, y) \leqslant \alpha \tilde{f}(x^{(1)}, y^{(1)}) + (1-\alpha) \tilde{f}(x^{(2)}, y^{(2)})$. Поскольку $\tilde{f}^*(y) = \min \{\tilde{f}(x', y) \mid x' \in X(y)\}$, то $\tilde{f}^*(y) \leqslant \alpha \tilde{f}^*(y^{(1)}) + (1-\alpha) \tilde{f}^*(y^{(2)})$.

Аналогично можно показать, что если функция $\tilde{f}(x, y)$ квазивыпукла на (выпуклом) множестве Z, то функция $\tilde{f}^*(y)$ квазивыпукла на множестве Y.

Таким образом, если задача В является задачей выпуклого программирования, то задачи В′ и В″ также задачи выпуклого программирования, если задача В — задача минимизации квазивыпуклой функции в выпуклой области, то и задачи В′ и В″ являются задачами минимизации квазивыпуклых функций в выпуклых областях. В частности, если В — задача линейного программирования, то задачи В′ — также задачи линейного программирования, а В″ — задача минимизации выпуклой (кусочно-линейной) функции при линейных ограничениях.

Будем говорить, что множество Z выпукло по y, если для любых $(x, y^{(1)}) \in Z$, $(x, y^{(2)}) \in Z$, и $0 < \alpha < 1$ точка $(x, \alpha y^{(1)} + (1-\alpha) y^{(2)}) \in Z$.

Если множество Z выпукло по y и $X(y^{(1)}) \cap X(y^{(2)}) \neq \varnothing$ для любых $y^{(1)}, y^{(2)} \in Y$, то множество Y является выпуклым.

Действительно, пусть $y^{(1)}, y^{(2)} \in Y$. По условию существует $x \in X(y^{(1)}) \cap X(y^{(2)})$. Следовательно, точки $(x, y^{(1)})$, $(x, y^{(2)}) \in Z$. Поскольку Z выпукло по y, то точка

$(x, \alpha y^{(1)}+(1-\alpha)y^{(2)})\in Z$ при любом $0<\alpha<1$. Следовательно, $x\in X(\alpha y^{(1)}+(1-\alpha)y^{(2)})$ и $\alpha y^{(1)}+(1-\alpha)y^{(2)}\in Y$.

Практически во всех известных схемах параметрической декомпозиции задач математического программирования условие $X(y^{(1)})\cap X(y^{(2)})\neq\varnothing$ выполняется при всех $y^{(1)}, y^{(2)}\in Y$. Тем самым для того чтобы гарантировать выпуклость множества Y, достаточно обеспечить выпуклость множества Z по y.

6.5. В заключение отметим, что рассматриваемые здесь вопросы имеют естественную интерпретацию в терминах математической теории *многоуровневых иерархических систем управления.*

Действительно, рассматривая задачу A как глобальную оптимизационную задачу, стоящую перед двухуровневой системой управления в целом, в результате ее параметрической декомпозиции получаем, вообще говоря, некоторую совокупность локальных оптимизационных задач типа B′, которые могут быть поставлены перед локальными решающими элементами нижнего уровня системы управления.

Конкретный вид задачи типа B′, стоящий перед отдельным решающим элементом, зависит от координирующего сигнала y, вырабатываемого решающим элементом верхнего уровня (координатором). Координация может осуществляться изменением как целей, так и ограничений.

Задача координатора состоит в выработке такого координирующего сигнала y^*, при котором (в результате решения соответствующих локальных задач) будет выработано искомое управляющее воздействие x^* и тем самым достигнута глобальная цель управления. Выполнимость условий 1°—3° гарантирует так называемую координируемость глобальной и локальных целей управления.

7. ГЕНЕРАЦИЯ СТОЛБЦОВ

В данном параграфе описываются методы декомпозиции задач линейного программирования, в основе которых лежит достаточно простая идея—получение столбцов матрицы ограничений, необходимых в процессе решения большеразмерной задачи, путем решения вспомогательных задач меньшей размерности. Предполагается предварительное ознакомление читателя с материалом, изложенным в параграфе 3.

7.1. Пусть дана *задача линейного программирования,* записанная для определенности в канонической форме

$$(c, x)\to\min, \tag{1}$$

$$Ax=b, \tag{2}$$

$$x\geqslant 0, \tag{3}$$

где $A=(A_1, A_2, \ldots, A_n)$ — матрица размерности $m\times n$ с элементами a_{ij}, $i=\overline{1, m}$, $j=\overline{1, n}$; A_j — столбцы этой матрицы; $x=(x_1, x_2, \ldots, x_n)$, $c=(c_1, c_2, \ldots, c_n)$, $b=(b_1, b_2, \ldots, b_m)$ — векторы указанных размерностей; $m<n$, $b\geqslant 0$.

Представим задачу (1)—(3) в виде

$$(c, x)\to\min, \tag{4}$$

$$A^{(0)}x=b^{(0)}, \tag{5}$$

$$A^{(1)}x=b^{(1)}, \tag{6}$$

$$x\geqslant 0. \tag{7}$$

Здесь $A^{(0)}$ и $A^{(1)}$ — матрицы размерности $m_0\times n$ и $m_1\times n$; $b^{(0)}$ и $b^{(1)}$ — векторы размерности m_0 и m_1 соответственно; $m_0+m_1=m$.

Предположим, что многогранное множество X, описываемое условиями (6)—(7), является *ограниченным.* Обозначим через $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ вершины многогранника X. Используя теорему о представлении многогранника, можно записать

$$x=\sum_{t=1}^{T} z_t x^{(t)}, \quad \sum_{t=1}^{T} z_t=1, \quad z_t\geqslant 0, \quad t=\overline{1, T},$$

для любой точки $x\in X$.

Подставляя последнее выражение в (4) и (5), исходную задачу можно переписать в виде

$$\sum_{t=1}^{T} \sigma_t z_t \to \min, \tag{8}$$

$$\sum_{t=1}^{T} P_t z_t = b^{(0)}, \tag{9}$$

$$\sum_{t=1}^{T} z_t = 1, \tag{10}$$

$$z_t \geqslant 0, \ t = \overline{1, T}, \tag{11}$$

где $\sigma_t = (c, x^{(t)})$; $P_t = A^{(0)} x^{(t)}$, $t = \overline{1, T}$.

Полученная задача эквивалентна задаче (1)—(3) в том смысле, что если $(z_1^*, z_2^*, \ldots, z_T^*)$ — ее решение, то $x^* = \sum\limits_{t=1}^{T} z_t^* x^{(t)}$ — решение задачи (1)—(3). Задачу (8)—(11) обычно называют *z-задачей*. В отличие от исходной она содержит меньшее число ограничений — (m_0+1), но число переменных z_t (равное числу вершин многогранника X) обычно велико. Непосредственная запись z-задачи сопряжена с вычислением коэффициентов σ_t целевой функции и вектор-столбцов P_t ограничений, что в свою очередь требует знания всех крайних точек многогранника X.

7.2. Пример. Рассмотрим задачу

$$(c, x) = -3x_1 - x_2 \to \min,$$

$$A^{(0)} x = 2x_1 + x_2 \leqslant 14,$$

$$A^{(1)} x = \begin{cases} x_1 + x_2 \leqslant 8, \\ x_1 - x_2 \leqslant 6, \end{cases}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0.$$

Многогранник X графически показан на рис. 7.1. Вершины этого многогранника: $x^{(1)} = (0, 0)$, $x^{(2)} = (6, 0)$, $x^{(3)} = (7, 1)$, $x^{(4)} = (0, 8)$. Любую его точку $x = (x_1, x_2)$ можно представить в виде

$$x = \sum_{t=1}^{4} z_t x^{(t)}, \quad \sum_{t=1}^{4} z_t = 1, \ z_t \geqslant 0, \ t = \overline{1, 4},$$

или

$$x_1=0z_1+6z_2+7z_3+0z_4,$$

$$x_2=0z_1+0z_2+1z_3+8z_4,$$

$$z_1+z_2+z_3+z_4=1,$$

$$z_1\geqslant 0,\ z_2\geqslant 0,\ z_3\geqslant 0,\ z_4\geqslant 0.$$

Подставляя полученные выражения в целевую функцию и первую группу ограничений, получаем z-задачу:

$$0z_1-18z_2-22z_3-8z_4\to\min,$$

$$0z_1+12z_2+15z_3+8z_4\leqslant 14,$$

$$z_1+z_2+z_3+z_4=1,$$

$$z_1\geqslant 0,\ z_2\geqslant 0,\ z_3\geqslant 0,\ z_4\geqslant 0,$$

или в канонической форме:

$$0z_1-18z_2-22z_3-8z_4+0z_5\to\min,$$

$$0z_1+12z_2+15z_3+8z_4+1z_5=14,$$

$$z_1+z_2+z_3+z_4+0z_5=1,$$

$$z_1\geqslant 0,\ z_2\geqslant 0,\ z_3\geqslant 0,\ z_4\geqslant 0,\ z_5\geqslant 0.$$

Значения $\sigma_1=0$, $\sigma_2=-18$, $\sigma_3=-22$, $\sigma_4=-8$, $\sigma_5=0$, векторы $P_1=(0)$, $P_2=(12)$, $P_3=(15)$, $P_4=(8)$, $P_5=(1)$. В данном примере z-задачу оказалось возможным записать в явном виде, поскольку число вершин многогранника X невелико.

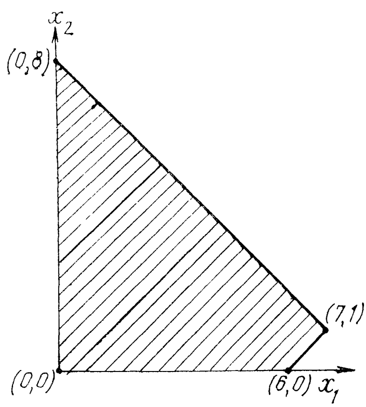

Рис. 7.1. Многогранник X

7.3. Метод или, точнее, *методы генерации столбцов* основаны на том факте, что при решении z-задачи каким-либо конечным методом (например, модифицированным симплексным методом) осуществляется целенаправленный перебор вершин многогранного множества, определяемого ограничениями этой задачи. Каждая вершина содержит не более m_0+1 положительных компонент z_t (остальные $z_t=0$) и ей сопоставляется матрица базиса $\bar{Q}$, содержащая m_0+1 столбцов вида $\bar{P}_t=\begin{pmatrix} P_t \\ 1 \end{pmatrix}$. При переходе от одной вершины к другой в матрице $\bar{Q}$ один из векторов $\bar{P}_t$, скажем $\bar{P}_s$, замещается новым вектором $\bar{P}_k$.

Если известны матрица базиса $\bar{Q}$ и вектор $\bar{P}_k$, то выбор подлежащего замещению вектора $\bar{P}_s$, как уже отмечалось в параграфе 3, не вызывает особых затруднений. Иначе обстоит дело с определением вектора $\bar{P}_k$. Следуя обычному порядку, необходимо для всех $t=\overline{1, T}$ вычислить величины $\Delta_t=\sigma_t-(\sigma_{\bar{Q}}, \bar{Q}^{-1}\bar{P}_t)$, где $\sigma_{\bar{Q}}$ — вектор коэффициентов целевой функции (8), соответствующий матрице базиса $\bar{Q}$. В качестве k может быть выбран любой индекс t, при котором $\Delta_t<0$. В частности, если $\Delta_{t'}=\min\{\Delta_t \mid t=\overline{1, T}\}<0$, то можно положить $k=t'$.

Покажем, что значение $\Delta_{t'}$ может быть определено в результате решения некоторой *вспомогательной (локальной) задачи линейного программирования*. Выражая Δ_t через параметры задачи (4)—(7), имеем

$$\Delta_t=\sigma_t-(\sigma_{\bar{Q}}, \bar{Q}^{-1}\bar{P}_t)=\sigma_t-(\sigma_{\bar{Q}}\bar{Q}^{-1}, \bar{P}_t)=\sigma_t-(\Lambda, P_t)-\lambda=$$

$$=(c, x^{(t)})-(\Lambda A^{(0)}, x^{(t)})-\lambda=(c-\Lambda A^{(0)}, x^{(t)})-\lambda,$$

где $(\Lambda, \lambda)=\sigma_{\bar{Q}}\bar{Q}^{-1}$ — вектор значений двойственных переменных z-задачи (λ соответствует ограничению (10)).

Следовательно, для определения значения $\Delta_{t'}$ достаточно вычислить $\min\{(c-\Lambda A^{(0)}, x^{(t)}) \mid t=\overline{1, T}\}$ или (используя соотношение (6) из параграфа 3 и учитывая, что $x^{(t)}$, $t=\overline{1, T}$,— вершины многогранника X) $\min\{(c-\Lambda A^{(0)}, x) \mid x\in X\}$, т. е. решить задачу линейного программирования вида

$$(c-\Lambda A^{(0)}, x)\to\min, \qquad (12)$$

$$A^{(1)}x=b^{(1)}, \qquad (13)$$

$$x\geqslant 0. \qquad (14)$$

Пусть $x^{(k)}$ — решение этой задачи, $g=(c-\Lambda A^{(0)}, x^{(k)})$ — оптимальное значение целевой функции. Если $g<\lambda$, то искомый вектор $P_k=A^{(0)}x^{(k)}$, значение $\sigma_k=(c, x^{(k)})$ и можно переходить к операции включения вектора $\bar{P}_k=\begin{pmatrix} P_k \\ 1 \end{pmatrix}$ в базис.

Если $g\geqslant\lambda$, то полученное на данном шаге решение z-задачи является искомым. Оно содержит не более m_0+1 положительных компонент z_t (для определенности $z^*_{t_1}, z^*_{t_2}, \ldots, z^*_{t_{m_0+1}}$), остальные $z_t=0$. Зная вершины $x^{(t_1)}, x^{(t_2)}, \ldots, x^{(t_{m_0+1})}$ многогранника X, решение исходной задачи (1)—(3) можно записать в виде $x^*=$ $=\sum\limits_{l=1}^{m_0+1} z^*_t x^{(t_l)}$.

Таким образом, зная некоторый исходный базис z-задачи (в невырожденном случае m_0+1 линейно независимых столбцов $\bar{P}_t$) и соответствующие значения σ_t, можно сформулировать локальную задачу (12)—(14). В результате ее решения может быть определен новый столбец $\bar{P}_k$, подлежащий включению в базис, либо установлена оптимальность рассматриваемого базисного решения z-задачи.

Иными словами, процесс решения z-задачи может быть осуществлен без записи ее в явном виде или, что то же, без знания всех вершин многогранника X. Достаточно на каждом шаге иметь набор из m_0+1 вершин (вместо всех T вершин многогранника X) и при необходимости заменять одну из вершин в рассматриваемом наборе новой, генерируемой в результате решения локальной задачи (12)—(14).

Описанная *процедура решения задач линейного программирования* схематически представлена на рис. 7.2.

7.4. Пример. В качестве исходной матрицы базиса z-задачи, сформулированной в примере п. 7.2, естественно выбрать матрицу $\bar{Q}_1=(\bar{P}_1, \bar{P}_5)=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, вектор $\sigma_{\bar{Q}_1}=$ $=(0, 0)$. Отметим, что этот выбор мог бы быть осуществлен и без записи z-задачи в явном виде.

Имеем $(\Lambda, \lambda)=\sigma_{\bar{Q}_1}\bar{Q}_1^{-1}=(0, 0)$, $\Lambda=0$, $\lambda=0$. Сформулируем локальную задачу (12)—(14):

$$(c-\Lambda A^{(0)}, x)=(c, x)=-3x_1-x_2\to\min,$$

$$A^{(1)}x=\begin{cases} x_1+x_2\leqslant 8,\\ x_1-x_2\leqslant 6,\end{cases}$$

$$x_1\geqslant 0,\ x_2\geqslant 0.$$

Ее решением является вершина $x^{(k)}=x^{(3)}=(7,\ 1)$, значение целевой функции равно -22. Поскольку $-22<<\lambda=0$, то в матрице базиса $\bar{Q}_1$ следует заменить один из столбцов новым столбцом $\bar{P}_k=\bar{P}_3=\begin{pmatrix} P_3\\ 1\end{pmatrix}$, где $P_3==A^{(0)}x^{(3)}=(15)$. Значение $\sigma_k=\sigma_3=(c,\ x^{(3)})=-22$. Новый столбец $\bar{P}_k$ и соответствующее значение σ_k получены без записи z-задачи в явном виде.

Обычными для симплексного метода рассуждениями приходим к заключению, что из базиса $\bar{Q}_1$ необходимо удалить столбец $\bar{P}_5$. В результате получаем новую матрицу базиса $\bar{Q}_2=(\bar{P}_1,\ \bar{P}_3)=\begin{pmatrix} 0 & 15\\ 1 & 1\end{pmatrix}$, $\sigma_{\bar{Q}_2}=(0,\ -22)$.

Непосредственно или используя приведенные в п. 3.2

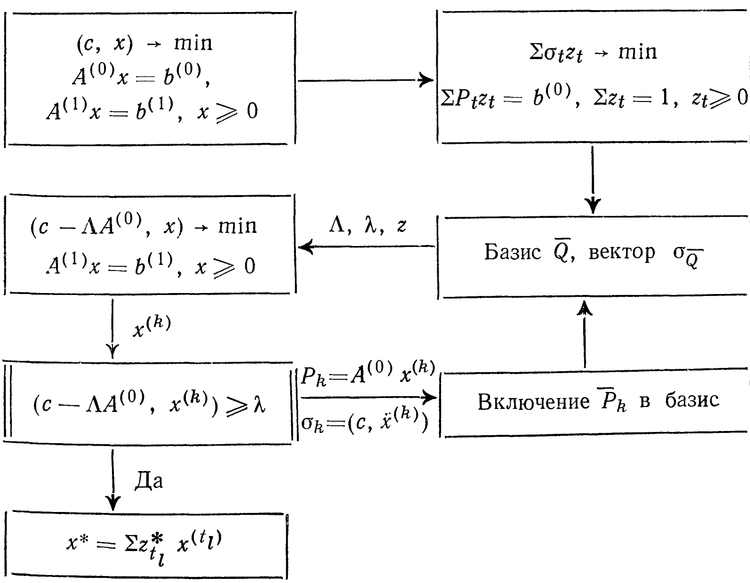

Рис. 7.2. Схема генерации столбцов

параграфа 3 формулы, находим $\overline{Q}_2^{-1} = \begin{pmatrix} -\frac{1}{15} & 1 \\ \frac{1}{15} & 0 \end{pmatrix}$. Имеем $(\Lambda,\ \lambda) = \sigma_{\overline{Q}_2} \overline{Q}_2^{-1} = \left(-\frac{22}{15},\ 0\right)$, $\Lambda = -\frac{22}{15}$, $\lambda = 0$.

Локальная задача записывается в виде

$$-\frac{1}{15} x_1 + \frac{7}{15} x_2 \to \min,$$
$$x_1 + x_2 \leqslant 8,$$
$$x_1 - x_2 \leqslant 6,$$
$$x_1 \geqslant 0,\ x_2 \geqslant 0.$$

Решением этой задачи является $x^{(k)} = x^{(2)} = (6,\ 0)$, значение целевой функции $-\frac{6}{15}$. Поскольку $-\frac{6}{15} < \lambda = 0$, то в матрице $\overline{Q}_2$ следует заменить один из столбцов столбцом $\overline{P}_k = \overline{P}_2 = \begin{pmatrix} P_2 \\ 1 \end{pmatrix}$, где $P_2 = A^{(0)} x^{(2)} = (12)$. Значение $\sigma_k = \sigma_2 = (c,\ x^{(2)}) = -18$. Имеем $\overline{Q}_3 = (\overline{P}_2,\ \overline{P}_3) = \begin{pmatrix} 12 & 15 \\ 1 & 1 \end{pmatrix}$, $\overline{Q}_3^{-1} = \begin{pmatrix} -\frac{1}{3} & 5 \\ \frac{1}{3} & -4 \end{pmatrix}$, $(\Lambda,\ \lambda) = \left(-\frac{4}{3},\ -2\right)$.

Локальная задача записывается в виде

$$-\frac{1}{3} x_1 + x_2 \to \min,$$
$$x_1 + x_2 \leqslant 8,$$
$$x_1 - x_2 \leqslant 6,$$
$$x_1 \geqslant 0,\ x_2 \geqslant 0.$$

Решением этой задачи является вершина $x^{(k)} = x^{(2)} = (6,\ 0)$, значение целевой функции -2 и $-2 = \lambda$. Следовательно, $\overline{Q}_3$ — матрица базиса, соответствующая оптимальному плану z-задачи. Имеем $(z_2^*, z_3^*) = \bar{Q}_3^{-1} b^{(1)} = \left(\frac{1}{3}, \frac{2}{3}\right)$, $x^{(2)} = (6, 0)$, $x^{(3)} = (7, 1)$. Значение $x_1^* = \frac{1}{3} \cdot 6 + \frac{2}{3} \cdot 7 = 6\frac{2}{3}$, $x_2^* = \frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1 = \frac{2}{3}$.

7.5. Перейдем к рассмотрению ситуации, когда множество X является *неограниченным.* Пусть $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ — вершины, а $x^{(T+1)}, x^{(T+2)}, \ldots, x^{T_1}$ — направляющие векторы неограниченных ребер этого множества. Используя соотношение (5), приведенное в параграфе 3, z-задачу можно записать в виде

$$\sum_{t=1}^{T_1} \sigma_t z_t \to \min, \tag{15}$$

$$\sum_{t=1}^{T_1} P_t z_t = b^{(0)}, \tag{16}$$

$$\sum_{t=1}^{T_1} \varepsilon_t z_t = 1, \tag{17}$$

$$z_t \geqslant 0, \quad t = \overline{1, T_1}, \tag{18}$$

где $\varepsilon_t = 1$ при $t = \overline{1, T}$ и $\varepsilon_t = 0$ при $t = \overline{T+1, T_1}$.

Значение $\Delta_t = (c - \Lambda A^{(0)}, x^{(t)}) - \varepsilon_t \lambda$, вектор $\bar{P}_t = \begin{pmatrix} P_t \\ \varepsilon_t \end{pmatrix}$ $t = \overline{1, T_1}$.

Если при решении локальной задачи (12)—(14) минимум достигается в вершине $x^{(k)}$, то, очевидно, $\Delta_t \geqslant \Delta_k$ для всех $t = \overline{1, T}$. Покажем, что $\Delta_t \geqslant 0$ для всех $t = \overline{T+1, T_1}$.

Действительно, пусть $\Delta_q < 0$ при некотором $T+1 \leqslant q \leqslant T_1$ и $x' = x^{(p)} + \mu x^{(q)}$ — неограниченное ребро множества X ($x^{(p)}$ — вершина X). Поскольку $\varepsilon_q = 0$, то $(c - \Lambda A^{(0)}, x^{(q)}) < 0$ и $(c - \Lambda A^{(0)}, x') \to \infty$ при $\mu \to \infty$. Получаем противоречие.

Таким образом, в текущий базис следует включить вектор $\bar{P}_k$, если $\Delta_k < 0$, или завершить решение z-задачи, если $\Delta_k \geqslant 0$.

Пусть при решении локальной задачи (12)—(14) установлено, что целевая функция (12) неограничена на множестве X. Это означает, что на некоторой итерации решения данной задачи найдена такая матрица, скажем B, и такой столбец $A_j^{(1)}$, что

$$c_j^{(1)}-(c_B^{(1)},\ B^{-1}A_j^{(1)})<0 \text{ и } B^{-1}A_j^{(1)}\leqslant 0. \qquad (19)$$

Здесь $c^{(1)}=(c-\Lambda A^{(0)})$ — вектор коэффициентов целевой функции (12); $c_j^{(1)}$ — j-я компонента этого вектора; $c_B^{(1)}$ — вектор коэффициентов, соответствующий матрице B.

Обозначим базисное решение задачи (12)—(14), полученное на предыдущей итерации, через $x^{(p)}$ (вершина множества X). Пусть для определенности $x^{(p)}=(x_1^{(p)}, x_2^{(p)}, \ldots, x_{m_1}^{(p)}, 0, \ldots, 0)$.

Рассмотрим вектор $x^{(q)}=(-(B^{-1}A_j^{(1)})^\tau, 0, \ldots, 0, x_j=1, 0, \ldots, 0)$. Ясно, что $A^{(1)}x^{(q)}=0$ и $x^{(q)}\geqslant 0$. Следовательно, вектор $x'=x^{(p)}+\mu x^{(q)}$ удовлетворяет условиям (13)—(14) при любом $\mu\geqslant 0$ и $x^{(q)}$ — направляющий вектор неограниченного ребра множества X.

Покажем, что $\Delta_q\leqslant 0$. Имеем $\Delta_q=(c-\Lambda A^{(0)}, x^{(q)})=(c^{(1)}, x^{(q)})=c_j^{(1)}-(c_B^{(1)}, B^{-1}A_j^{(1)})<0$ (условие (19)).

Следовательно, в очередной базис z-задачи может быть введен вектор $\bar{P}_q=\begin{pmatrix} P_q \\ 0 \end{pmatrix}=\begin{pmatrix} A^{(0)}x^{(q)} \\ 0 \end{pmatrix}$ и процесс решения z-задачи должен быть продолжен.

Таким образом, независимо от того, является ли множество X ограниченным или нет, процесс перехода от одной матрицы базиса z-задачи к другой может быть произведен в результате решения локальной задачи (12)—(14). Иными словами, решение z-задачи может быть получено в результате последовательной целенаправленной генерации ее столбцов, осуществляемой посредством решения цепочки локальных задач (отличающихся друг от друга значениями коэффициентов целевой функции). На каждой итерации вновь получаемый столбец замещает один из столбцов рассматриваемой на данном шаге матрицы базиса z-задачи. Следует отметить, что при таком подходе необязательно решать локальную задачу «до конца». Достаточно найти любую вершину $x^{(k)}$, в которой $(c-\Lambda A^{(0)}, x^{(k)})<\lambda$, что в ряде случаев позволяет значительно упростить процесс решения z-задачи.

Для построения исходного базиса z-задачи могут быть использованы стандартные методы искусственного базиса либо учтены специфические особенности конкретно рассматриваемой задачи. Если ограничения (5) записаны в форме неравенств, то исходный базис z-задачи определяется введением в нее дополнительных переменных.

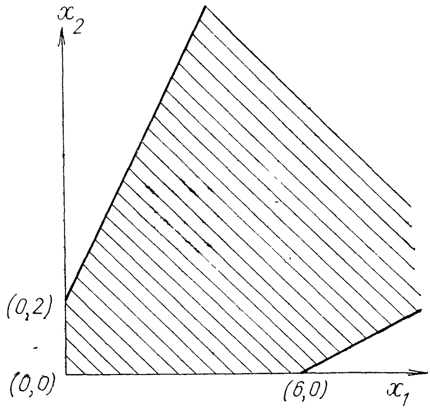

Рис. 7.3. Многогранное множество X

7.6. Пример. Рассмотрим задачу

$$(c, x) = -x_1 - 3x_2 \to \min,$$

$$A^{(0)}x = x_1 + x_2 \leqslant 10,$$

$$A^{(1)}x = \begin{cases} x_1 - 2x_2 \leqslant 6, \\ -2x_1 + x_2 \leqslant 2, \end{cases}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0.$$

Множество X представлено на рис. 7.3. В качестве исходной матрицы базиса z-задачи (при записи ее в канонической форме) можно выбрать $\bar{Q} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ с $\sigma_{\bar{Q}} = (0, 0)$.

На 1-й итерации решаем локальную задачу

$$-x_1 - 3x_2 \to \min,$$

$$x_1 - 2x_2 \leqslant 6,$$

$$-2x_1 + x_2 \leqslant 2,$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0,$$

или (записывая ее в канонической форме)

$$-x_1-3x_2+0x_3+0x_4\to\min,$$

$$x_1-2x_2+x_3+0x_4=6,$$

$$-2x_1+x_2+0x_3+x_4=2,$$

$$x_1\geqslant 0,\ \ x_2\geqslant 0,\ \ x_3\geqslant 0,\ \ x_4\geqslant 0.$$

Исходная матрица базиса этой задачи $(A_3^{(1)}, A_4^{(1)})=$ $=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, базисное решение $(0, 0, 6, 2)$.

Включая в базис вектор $A_2^{(1)}$, получаем новую матрицу базиса $B=(A_2^{(1)}, A_3^{(1)})=\begin{pmatrix} -2 & 1 \\ 1 & 0 \end{pmatrix}$, $c_B^{(1)}=(-3, 0)$, базисное решение $x^{(p)}=(0, 2, 10, 0)$.

Находим $B^{-1}=\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$. Поскольку для $j=1$, $A_1^{(1)}=$ $=\begin{pmatrix} 1 \\ -2 \end{pmatrix}$, $c_1^{(1)}=-1$, выполняются условия (19), приходим к заключению, что целевая функция локальной задачи неограничена снизу.

Вектор $B^{-1}A_1^{(1)}=(-2, -3)$. Следовательно, $x^{(q)}=(1, 2, 3, 0)$ — направляющий вектор неограниченного ребра с вершиной $x^{(p)}=(0, 2, 10, 0)$.

В очередной базис $\bar{Q}$ z-задачи включается вектор $\bar{P}_q=$ $=\begin{pmatrix} 3 \\ 0 \end{pmatrix}$.

8. СПЕЦИАЛЬНЫЕ СЛУЧАИ

Описанный в предыдущем параграфе подход к решению задач линейного программирования позволяет получить решение задачи размерности $m\times n$ в результате решения некоторой совокупности локальных подзадач размерности $m_1\times n$, где $m_1<m$. Разбиение ограничений исходной задачи на две группы определяется спецификой решаемой задачи и теми условиями (объем оперативной памяти, вычислительные мощности и т. п.), в которых реализуется процесс ее решения. Метод генерации столбцов может быть использован и при решении локальной задачи. Тем самым решение исходной задачи может быть сведено к решению серии задач линейного программирования с любым заданным числом ограничений.

Эффективность методов генерации столбцов существенно возрастает, если матрица A ограничений исходной задачи обладает определенными структурными особенностями, учет которых лежит в основе так называемых структурно-ориентированных методов декомпозиции.

В данном параграфе рассматриваются два метода такого рода. Первый из них ориентирован на решение задач, матрицы ограничений которых содержат блочно-диагональные подматрицы. Второй метод предназначен для решения задач линейного программирования транспортного типа.

8.1. Рассмотрим *задачу линейного программирования* вида

$$\sum_{l=1}^{\nu}(\tilde{c}_l,\ \tilde{x}_l)\to\min, \tag{1}$$

$$\sum_{l=1}^{\nu}\tilde{A}_l^{(0)}\tilde{x}_l=b^{(0)}, \tag{2}$$

$$\tilde{A}_l^{(1)}\tilde{x}_l=\tilde{b}_l^{(1)},\quad l=\overline{1,\ \nu}, \tag{3}$$

$$\tilde{x}_l\geqslant 0,\quad l=\overline{1,\ \nu}, \tag{4}$$

где $\tilde{x}_l$, $\tilde{c}_l$ — n_l-мерные векторы; $\sum\limits_{l=1}^{\nu}n_l=n$; $b^{(0)}$ — m_0-мерный вектор; $\tilde{b}_l^{(1)}$ — $m_1^{(l)}$-мерные векторы; $\sum_{l=1}^{\nu} m^{(l)}=m_1$; $m_0+m_1=m$; $\tilde{A}_l^{(0)}$ и $\tilde{A}_l^{(1)}$ — матрицы размерности $m_0\times n_l$ и $m_1^{(l)}\times n_l$ соответственно, $l=\overline{1, \nu}$.

В рассматриваемом случае матрица $A^{(1)}=(\tilde{A}_l^{(1)})_{l=\overline{1,\nu}}$ имеет блочно-диагональную структуру. Любая точка $x=(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_\nu)$ многогранного множества X, определяемого ограничениями (3)—(4), может быть представлена в виде

$$\tilde{x}_l=\sum_{t=1}^{T^{(l)}} z_t^{(l)}\tilde{x}_l^{(t)},\ \sum_{t=1}^{T^{(l)}} \varepsilon_t^{(l)} z_t^{(l)}=1,\ z_t^{(l)}\geqslant 0,\ t=\overline{1, T^{(l)}},\ l=\overline{1, \nu}. \tag{5}$$

Здесь $\tilde{x}_l^{(t)}$ — вершины или направляющие векторы неограниченных ребер многогранного множества, определяемого ограничениями $\tilde{A}_l^{(1)}\tilde{x}_l=b_l^{(1)}$ и $\tilde{x}_l\geqslant 0$; $\varepsilon_t^{(l)}$ принимает значение 1 или 0 в зависимости от того, является ли $\tilde{x}_l^{(t)}$ вершиной или направляющим вектором неограниченного ребра.

Подставляя, как и раньше, выражение (5) в целевую функцию (1) и ограничения (2) исходной задачи, получаем z-задачу

$$\sum_{l=1}^{\nu}\ \sum_{t=1}^{T^{(l)}} \sigma_t^{(l)} z_t^{(l)}\rightarrow \min, \tag{6}$$

$$\sum_{l=1}^{\nu}\ \sum_{t=1}^{T^{(l)}} P_t^{(l)} z_t^{(l)}=b^{(0)}, \tag{7}$$

$$\sum_{t=1}^{T^{(l)}} \varepsilon_t^{(l)} z_t^{(l)}=1,\quad l=\overline{1, \nu}, \tag{8}$$

$$z_t^{(l)}\geqslant 0,\quad t=\overline{1, T^{(l)}},\quad l=\overline{1, \nu}, \tag{9}$$

где $\sigma_t^{(l)}=(\tilde{c}_l, \tilde{x}_l^{(t)})$, $P_t^{(l)}=\tilde{A}_l^{(0)}\tilde{x}_l^{(t)}$, $t=\overline{1, T^{(l)}}$, $l=\overline{1, \nu}$.

Решая эту задачу методом генерации столбцов, на каждой итерации определяем $(m_0+\nu)$-мерный вектор $(\Lambda, \bar{\lambda})$ значений двойственных переменных (Λ соответствует ограничениям (7), $\bar{\lambda}$ — ограничениям (8)).

Новый столбец генерируется в результате решения локальной задачи, которая в рассматриваемом случае распадается на ν независимых подзадач $(l=\overline{1, \nu})$:

$$(\tilde{c}_l-\Lambda\tilde{A}_l^{(0)}, \tilde{x}_l)\to\min, \tag{10}$$

$$\tilde{A}_l^{(1)}\tilde{x}_l=\tilde{b}_l^{(1)}, \tag{11}$$

$$\tilde{x}_l\geqslant 0. \tag{12}$$

Процесс решения каждой подзадачи может завершаться либо получением решения (вершины соответствующего многогранного множества), либо установлением факта неограниченности снизу линейной формы (10) при ограничениях (11)—(12). В любом случае рассуждения, аналогичные описанным в предыдущем параграфе, позволяют получить новый столбец z-задачи для включения его в базис (либо убедиться в том, что полученное на данной итерации решение является искомым). Поскольку этот столбец определяется набором результатов решения ν локальных подзадач, при разработке конкретных вычислительных схем используются различные подходы. Так, на каждой итерации можно прекращать процесс решения подзадач, если в результате решения одной из них получена информация о необходимости перехода к новому базису. С другой стороны, располагая результатами решения всех подзадач, можно на одной и той же итерации провести несколько переходов от базиса к базису z-задачи.

Отметим, что аналогичный подход может применяться и при решении задач со связывающими переменными и блочно-диагональной структурой остальной части матрицы ограничений. Действительно, осуществляя переход к двойственной задаче (применяя двойственный симплексный метод), получаем рассмотренную выше ситуацию.

8.2. Пример. Решим описанным методом задачу, условие которой приведено в п. 5.6 параграфа 5:

$$-x_1-3x_2-2x_3-x_4\to\min,$$
$$2x_1+x_2+x_3+x_4\leqslant 8,$$
$$x_1+x_2\qquad\qquad\leqslant 5,$$
$$x_3-x_4\leqslant 4,$$
$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0.$$

В обозначениях этого параграфа $\nu=2$, $\tilde{x}_1=(x_1, x_2)$, $\tilde{x}_2=(x_3, x_4)$, $\tilde{c}_1=(-1, -3)$, $\tilde{c}_2=(-2, -1)$, $\tilde{A}_1^{(0)}=(2, 1)$, $\tilde{A}_2^{(0)}=(1, 1)$, $\tilde{A}_1^{(1)}=(1, 1)$, $\tilde{A}_2^{(1)}=(1, -1)$, $b^{(0)}=(8)$, $\tilde{b}_1^{(1)}=$ $=(5)$, $\tilde{b}_2^{(1)}=(4)$.

В рассматриваемом случае z-задачу (6)—(9) можно записать в явном виде. Вершинами многогранника, описываемого ограничениями $x_1+x_2\leqslant 5$, $x_1\geqslant 0$, $x_2\geqslant 0$, являются $\tilde{x}_1^{(1)}=(0, 0)$, $\tilde{x}_1^{(2)}=(5, 0)$, $\tilde{x}_1^{(3)}=(0, 5)$. Вершинами многогранного множества, описываемого ограничениями $x_3-x_4\leqslant 4$, $x_3\geqslant 0$, $x_4\geqslant 0$, являются $\tilde{x}_2^{(1)}=(0, 0)$, $\tilde{x}_2^{(2)}=$ $=(4, 0)$, а направляющими векторами неограниченных ребер — векторы $\tilde{x}_2^{(3)}=(0, 1)$ и $\tilde{x}_2^{(4)}=(1, 1)$. Соотношения (5) записываются в виде

$$\tilde{x}_1=(x_1, x_2)=(0, 0)z_1^{(1)}+(5, 0)z_2^{(1)}+(0, 5)z_3^{(1)},$$
$$z_1^{(1)}+z_2^{(1)}+z_3^{(1)}=1,\ z_1^{(1)}\geqslant 0,\ z_2^{(1)}\geqslant 0,\ z_3^{(1)}\geqslant 0,$$
$$\tilde{x}_2=(x_3, x_4)=(0, 0)z_1^{(2)}+(4, 0)z_2^{(2)}+(0, 1)z_3^{(2)}+(1, 1)z_4^{(2)},$$
$$z_1^{(2)}+z_2^{(2)}+0z_3^{(2)}+0z_4^{(2)}=1,\ z_1^{(2)}\geqslant 0,\ z_2^{(2)}\geqslant 0,\ z_3^{(2)}\geqslant 0,\ z_4^{(2)}\geqslant 0.$$

Подставляя эти выражения в целевую функцию и первое ограничение исходной задачи, получаем искомую z-задачу:

$$0z_1^{(1)}-5z_2^{(1)}-15z_3^{(1)}+0z_1^{(2)}-8z_2^{(2)}-z_3^{(2)}-3z_4^{(2)}\to\min,$$
$$0z_1^{(1)}+10z_2^{(1)}+5z_3^{(1)}+0z_1^{(2)}+4z_2^{(2)}+z_3^{(2)}+2z_4^{(2)}\leqslant 8,$$
$$z_1^{(1)}+z_2^{(1)}+z_3^{(1)}\qquad\qquad=1,$$
$$z_1^{(2)}+z_2^{(2)}+0z_3^{(2)}+0z_4^{(2)}=1,$$
$$z_1^{(1)}\geqslant 0,\ z_2^{(1)}\geqslant 0,\ z_3^{(1)}\geqslant 0,\ z_1^{(2)}\geqslant 0,\ z_2^{(2)}\geqslant 0,\ z_3^{(2)}\geqslant 0,\ z_4^{(2)}\geqslant 0.$$

Решение z-задачи осуществляется, конечно, без записи ее в явном виде. Для организации процесса решения достаточно знать исходную матрицу базиса $\bar{Q}$ и соответствующий ему вектор $\sigma_{\bar{Q}}$ коэффициентов целевой функции z-задачи. В рассматриваемом случае в качестве $\bar{Q}$ можно выбрать единичную матрицу размерности 3×3, а в качестве $\sigma_{\bar{Q}}$ — нулевой вектор $(0, 0, 0)$. Вектором $(\Lambda, \bar{\lambda})$ значений двойственных переменных при этом является вектор $(0, 0, 0)$, где $\Lambda=(0)$, $\bar{\lambda}=(0, 0)$.

Локальные задачи (10)—(12) записываются в виде

$$\begin{array}{ll} -x_1-3x_2\to \min, & -2x_3-x_4\to \min, \\ x_1+x_2\leqslant 5, & x_3-x_4\leqslant 4, \\ x_1\geqslant 0,\ x_2\geqslant 0, & x_3\geqslant 0,\ x_4\geqslant 0. \end{array}$$

Решением первой из этих задач является вершина $(0, 5)$, целевая функция второй задачи неограниченно убывает вдоль луча $(4, 0)+\mu(1, 1)$, $\mu\geqslant 0$.

Используя соотношения $\sigma_t^{(l)}=(\tilde{c}_l, \tilde{x}_l^{(t)})$ и $P_t^{(l)}=\tilde{A}_l^{(0)}\tilde{x}_l^{(t)}$, получаем столбцы $\begin{pmatrix} 5 \\ 1 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$ и соответствующие коэффициенты целевой функции z-задачи -15 и -3. Полученных данных достаточно для того, чтобы перейти к новой матрице базиса (используя, например, обычную технику симплекс-метода) либо убедиться в оптимальности полученного решения.

8.3. Рассмотрим *задачу линейного программирования транспортного типа*

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}\to \min, \tag{13}$$

$$\sum_{j=1}^{n} x_{ij}=a_i,\quad i=\overline{1, m}, \tag{14}$$

$$\sum_{i=1}^{m} x_{ij}=b_j,\quad j=\overline{1, n}, \tag{15}$$

$$x_{ij}\geqslant 0,\quad i=\overline{1, m},\quad j=\overline{1, n}, \tag{16}$$

где a_i и b_j — положительные числа; $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

Пусть $x_{ij}^{(t)}$ — компоненты t-й вершины множества X, описываемого ограничениями (15)—(16), $t=\overline{1, T}$. Множество X совпадает с совокупностью точек, компоненты которых следующие:

$$x_{ij} = \sum_{t=1}^{T} z_t x_{ij}^{(t)} \quad \sum_{t=1}^{T} z_t = 1, \quad z_t \geqslant 0, \quad t=\overline{1, T}. \tag{17}$$

Подставляя выражение (17) в (13)—(14), получаем z-задачу

$$\sum_{t=1}^{T} \sigma_t z_t \to \min, \tag{18}$$

$$\sum_{t=1}^{T} p_{it} z_t = a_i, \quad i=\overline{1, m}, \tag{19}$$

$$z_t \geqslant 0, \quad t=\overline{1, T}, \tag{20}$$

где $\sigma_t = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}^{(t)}$; $p_{it} = \sum_{j=1}^{n} x_{ij}^{(t)}$, $i=\overline{1, m}$, $t=\overline{1, T}$.

Условие $\sum_{t=1}^{T} z_t = 1$ опущено, поскольку его выполнимость обеспечивают остальные ограничения z-задачи. Действительно,

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} \sum_{t=1}^{T} p_{it} z_t = \sum_{i=1}^{m} \sum_{t=1}^{T} \sum_{j=1}^{n} x_{ij}^{(t)} z_t =$$

$$= \sum_{t=1}^{T} z_t \sum_{j=1}^{n} \left(\sum_{i=1}^{m} x_{ij}^{(t)} \right) = \sum_{t=1}^{T} z_t \left(\sum_{j=1}^{n} b_j \right) \text{ и } \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j > 0.$$

Обозначая через $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ вектор значений двойственных переменных z-задачи, локальную задачу можно записать в виде

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \lambda_i) x_{ij} \to \min, \tag{21}$$

$$\sum_{i=1}^{m} x_{ij}=b_j,\quad j=\overline{1, n}, \tag{22}$$

$$x_{ij}\geqslant 0,\quad i=\overline{1, m},\quad j=\overline{1, n}. \tag{23}$$

Поскольку каждая переменная этой задачи входит только в одно из ограничений (22), то ее решение может быть получено в результате решения n локальных подзадач $(j=\overline{1, n})$:

$$\sum_{i=1}^{m} (c_{ij}-\lambda_i) x_{ij}\to \min, \tag{24}$$

$$\sum_{i=1}^{m} x_{ij}=b_j, \tag{25}$$

$$x_{ij}\geqslant 0,\quad i=\overline{1, m}, \tag{26}$$

каждая из которых решается тривиально: если $c_{lj}-\lambda_l=\min\limits_{1\leqslant i\leqslant m} (c_{ij}-\lambda_i)$, то $x_{lj}=b_j$, остальные $x_{ij}=0$. В результате получаем решение $\{x_{ij}^{(k)}\}$ локальной задачи (21)—(23). Величина

$$\Delta_k=\sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}-\lambda_i) x_{ij}^{(k)}=\sum_{j=1}^{n} b_j \min_{1\leqslant i\leqslant m} (c_{ij}-\lambda_i).$$

Если $\Delta_k<0$, то определяются величины σ_k и p_{ik} и осуществляется переход к новому базису. В противном случае процесс решения z-задачи завершается. Соотношения (17) позволяют получить решение исходной задачи.

Описанный *процесс* схематически представлен на рис. 8.1. Его можно использовать при решении любой транспортной задачи, но наиболее целесообразен он, очевидно, в случаях, когда m сравнительно невелико, а $n\gg m$.

8.4. На примере решения задачи (1)—(4) проиллюстрируем применение *метода одновременного решения прямой и двойственной задач* для построения соответствующих декомпозиционных схем.

Будем предполагать (для простоты изложения), что многогранные множества, описываемые условиями (3)—

(4), являются ограниченными. В этом случае z-задачу (6)—(9) можно записать в виде

$$\sum_{l=1}^{\nu} \sum_{t=1}^{T^{(l)}} \sigma_t^{(l)} z_t^{(l)} \to \min, \tag{27}$$

$$\sum_{l=1}^{\nu} \sum_{t=1}^{T^{(l)}} P_t^{(l)} z_t^{(l)} = b^{(0)}, \tag{28}$$

$$\sum_{t=1}^{T^{(l)}} z_t^{(l)} = 1, \quad l = \overline{1, \nu}, \tag{29}$$

$$z_t^{(l)} \geqslant 0, \quad t = \overline{1, T^{(l)}}, \quad l = \overline{1, \nu}, \tag{30}$$

где $\sigma_t^{(l)} = (\tilde{c}_l, \tilde{x}_l^{(t)})$; $P_t^{(l)} = \tilde{A}_l^{(0)} \tilde{x}_l^{(t)}$, $\tilde{x}_l^{(t)}$, $t = \overline{1, T}$,— вершины многогранника

$$\tilde{A}_l^{(1)} \tilde{x}_l = \tilde{b}_l^{(1)}, \quad \tilde{x}_l \geqslant 0. \tag{31}$$

Перейдем к двойственной задаче. Пусть $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{m_0})$ — вектор двойственных переменных, соответствующих ограничениям (28); $\overline{\lambda} = (\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(\nu)})$ — вектор двойственных переменных, соответствующих огра-

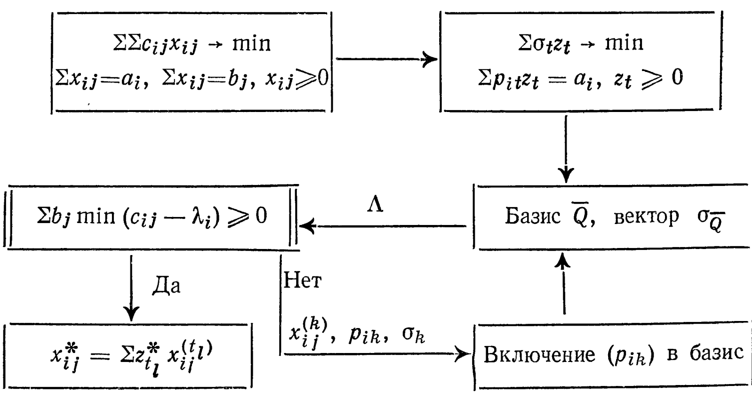

Рис. 8.1. Схема решения транспортной задачи методом генерации столбцов

ничениям (29). Задачу, двойственную к z-задаче, запишем в виде

$$(b^{(0)}, \Lambda)+\sum_{l=1}^{\nu}\lambda^{(l)}\to \max, \tag{32}$$

$$(P_t^{(l)}, \Lambda)+\lambda^{(l)}\leqslant\sigma_t^{(l)}, \quad t=\overline{1, T^{(l)}}, \quad l=\overline{1, \nu}. \tag{33}$$

Используя выражения $P_t^{(l)}$ и $\sigma_t^{(l)}$ через $\tilde{x}_l^{(t)}$, ограничения (33) можно записать в виде

$$\lambda^{(l)}\leqslant(\tilde{c}_l-\Lambda\tilde{A}_l^{(0)}, \tilde{x}_l^{(t)}), \quad t=\overline{1, T^{(l)}}, \quad l=\overline{1, \nu}. \tag{34}$$

Исходное для рассматриваемого метода решение $(\Lambda^{(0)}, \overline{\lambda}_0)$ двойственной задачи (32), (34) можно получить, полагая $\Lambda^{(0)}=0$ и $\lambda_0^{(l)}=\min\limits_{1\leqslant t\leqslant T^{(l)}}\{(\tilde{c}_l, \tilde{x}_l^{(t)})\}$, $l=\overline{1, \nu}$. Учитывая, что $\tilde{x}_l^{(t)}$ — вершины многогранника (31), для определения $\overline{\lambda}_0=(\lambda_0^{(1)}, \lambda_0^{(2)}, \ldots, \lambda_0^{(\nu)})$ достаточно решить ν локальных задач $(l=\overline{1, \nu})$:

$$(\tilde{c}_l, \tilde{x}_l)\to\min, \tag{35}$$

$$\tilde{A}_l^{(1)}\tilde{x}_l=\tilde{b}_l^{(1)}, \quad \tilde{x}_l\geqslant 0. \tag{36}$$

В результате их решения, наряду с получением значений $\lambda_0^{(l)}$ целевых функций, могут быть выделены вершины $x_l^{(t)}$, $t\in\tau_0^{(l)}\subseteq\{1, 2, \ldots, T^{(l)}\}$, многогранников (31), $l=\overline{1, \nu}$, на которых эти значения достигаются. Иными словами, могут быть найдены ограничения (34) двойственной задачи, выполняемые в точке $(\Lambda^{(0)}, \overline{\lambda}_0)$ со знаком равенства. Это позволяет сформулировать сокращенную задачу

$$\sum_{i=1}^{m_0} v_i+\sum_{l=1}^{\nu} v^{(l)}\to\min, \tag{37}$$

$$\sum_{l=1}^{\nu}\sum_{t\in\tau_0^{(l)}} P_t^{(l)}z_t^{(l)}+v=b^{(0)}, \tag{38}$$

$$\sum_{t\in\tau_0^{(l)}} z_t^{(l)}+v^{(l)}=1,\ l=\overline{1,\nu}, \tag{39}$$

$$z_t^{(l)}\geqslant 0,\ t\in\tau_0^{(l)},\ v^{(l)}\geqslant 0,\ l=\overline{1,\nu},$$
$$v=(v_1, v_2, \ldots, v_{m_0})\geqslant 0 \tag{40}$$

и найти оптимальные значения $(\widetilde{\Lambda}^*, \widetilde{\lambda}_*)$ двойственных ей переменных.

Компоненты векторов невязок записываются в виде

$$\Delta_{lt}^{(0)}=(P_t^{(l)}, \Lambda^{(0)})+\lambda_0^{(l)}-\sigma_t^{(l)}=(\Lambda^{(0)}\tilde{A}_l^{(0)}-\tilde{c}_l,\ x_l^{(t)})+\lambda_0^{(l)},$$

$$\delta^*_{lt}=(P_t^{(l)}, \widetilde{\Lambda}^*)+\lambda_*^{(l)}=(\widetilde{\Lambda}^*\tilde{A}_l^{(0)}, \tilde{x}_l^{(t)})+\lambda_*^{(l)}.$$

При переходе к новым значениям двойственных переменных $\Lambda^{(1)}=\Lambda^{(0)}-\mu_0\widetilde{\Lambda}^*$ и $\overline{\lambda}_1=\overline{\lambda}_0-\mu_0\widetilde{\lambda}_*$ в качестве μ_0 следует выбрать наибольшее из значений

$$\mu_0^{(l)}=\max_{\delta_{lt}>0}\frac{\Delta_{lt}^{(0)}}{\delta^*_{lt}},\ l=\overline{1,\nu}.$$

Непосредственное вычисление значений $\mu_0^{(l)}$ сопряжено с перебором всех вершин многогранников (31), $l=\overline{1,\nu}$, что практически нереализуемо. Исключить полный перебор позволяет тот факт, что дробно-линейная целевая функция, также, как и линейная, достигает экстремума в вершинах многогранника.

Следовательно, для вычисления значений $\mu_0^{(l)}$ достаточно решить ν локальных задач дробно-линейного программирования $(l=\overline{1,\nu})$:

$$\frac{(\Lambda\tilde{A}_l^{(0)}-\tilde{c}_l,\ \tilde{x}_l)+\lambda^{(l)}}{(\Lambda^*\tilde{A}_l^{(0)},\ \tilde{x}_l)+\lambda_*^{(l)}}\to\max(=\mu_0^{(l)}), \tag{41}$$

$$\tilde{A}_l^{(1)}\tilde{x}_l=\tilde{b}_l^{(1)},\ \tilde{x}_l\geqslant 0, \tag{42}$$

$$(\Lambda^*\tilde{A}_l^{(0)},\ \tilde{x}_l)+\lambda_*^{(l)}>0. \tag{43}$$

Ограничения (42) дают возможность при поиске максимума целевой функции (41) ограничиться рассмотрением вершин $x_l^{(t)}$, а ограничения (43) исключают те из них, для которых $\delta^*_{lt} \leqslant 0$.

Если хотя бы одна из задач (41)—(43) имеет решение, то находится величина $\mu_0 = \max\limits_{1 \leqslant l \leqslant \nu} \{\mu_0^{(l)}\}$ и осуществляется переход к новым значениям двойственных переменных z-задачи.

Если для всех $l = \overline{1, \nu}$ ограничения (42)—(43) противоречивы, то z-задача имеет неограниченное решение.

Таким образом, решение исходной задачи с блочно-диагональной матрицей ограничений (и связывающими ограничениями) может быть получено в результате решения серии локальных линейных (35), (36), дробно-линейных (41)—(43) и сокращенных (37)—(39) z-задач.

9. ОБОБЩЕННЫЙ ПОДХОД

Методы генерации столбцов могут быть использованы и при решении задач нелинейного, в основном выпуклого, программирования.

9.1. Рассмотрим задачу

$$f(x)\to\min, \tag{1}$$

$$G^{(0)}(x)\leqslant 0, \tag{2}$$

$$G^{(1)}(x)\leqslant 0, \tag{3}$$

где $f(x)$ — выпуклая функция; $G^{(0)}(x)$ и $G^{(1)}(x)$ — выпуклые вектор-функции.

Пусть точки $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ удовлетворяют ограничению (3). Рассмотрим линейную z-задачу следующего вида:

$$\sum_{t=1}^{T} f(x^{(t)})z_t\to\min, \tag{4}$$

$$\sum_{t=1}^{T} G^{(0)}(x^{(t)})z_t\leqslant 0, \tag{5}$$

$$\sum_{t=1}^{T} z_t=1, \tag{6}$$

$$z_t\geqslant 0,\quad t=\overline{1,\ T}. \tag{7}$$

Если $z^*=(z_1^*, z_2^*, \ldots, z_T^*)$ — решение данной задачи, то $x(z^*)=\sum\limits_{t=1}^{T} z_t^* x^{(t)}$ — приближенное решение исходной задачи. Значение целевой функции (4) при этом не меньше $f(x(z^*))$.

Действительно, поскольку $G^{(1)}(x)$ — выпуклая вектор-функция и точки $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ удовлетворяют ограничению (3), то точка $x(z^*)$ будучи выпуклой линейной комбинацией этих точек также удовлетворяет ограничению (3). Аналогично $G^{(0)}(x(z^*))=G^{(0)}(\sum\limits_{t=1}^{T} z_t^* x^{(t)})\leqslant$

$\leqslant \sum_{t=1}^{T} z_t^* G^{(0)}(x^{(t)}) \leqslant 0$ (ограничение (5)). Наконец, $f(x(z^*)) = f\left(\sum_{t=1}^{T} z_t^* x^{(t)}\right) \leqslant \sum_{t=1}^{T} z_t^* f(x^{(t)})$.

Естественно предполагать, что выбор бóльшего числа точек $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ для линеаризации исходной задачи приводит к получению более точного ее решения. Однако решение z-задачи при большом числе T переменных вызывает определенные затруднения, что приводит к необходимости рассматривать некоторую цепочку последовательно *«расширяющихся»* z-задач и получать в результате их решения последовательно уточняющиеся приближенные решения исходной задачи. Один из простейших способов построения такой цепочки z-задач состоит в том, что к рассматриваемой совокупности точек $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ добавляется новая точка $x^{(T+1)}$. К расширенной таким образом совокупности точек $x^{(1)}, x^{(2)}, \ldots, x^{(T+1)}$ добавляется новая точка $x^{(T+2)}$ и т. д. Выбор каждой новой точки сопряжен с введением новой переменной z_t и соответствующего дополнительного столбца в матрицу ограничений рассматриваемой на данном шаге z-задачи.

Целенаправленный выбор новой точки $x^{(T+1)}$ (удовлетворяющей ограничению (3)) по заданной совокупности точек $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ можно осуществить исходя из следующих соображений.

Пусть (Λ, λ) — вектор значений двойственных переменных задачи (4)—(7) (значение λ соответствует ограничению (6)). В рассматриваемом случае $\Delta_t = f(x^{(t)}) - (\Lambda, G^{(0)}(x^{(t)})) - \lambda$, $t = \overline{1, T}$.

Добавление новой точки $x^{(T+1)}$ к совокупности точек $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ приводит к получению новой z-задачи, отличающейся от задачи (4)—(7) наличием новой переменной z_{T+1}, соответствующего столбца $\begin{pmatrix} C^{(0)}(x^{(T+1)}) \\ 1 \end{pmatrix}$ матрицы ограничений и коэффициента $f(x^{(T+1)})$ целевой функции. Значение $\Delta_{T+1} = f(x^{(T+1)}) - (\Lambda, G^{(0)}(x^{(T+1)})) - \lambda$.

Если $\Delta_{T+1} < 0$, то решение исходной задачи может быть уточнено (введением $(T+1)$-го столбца в матрицу базиса). В противном случае полученное решение является искомым.

Таким образом, для нахождения точки $x^{(T+1)}$ следует решить *локальную* задачу вида

$$f(x)-(\Lambda,\ G^{(0)}(x))\to\min, \tag{8}$$

$$G^{(1)}(x)\leqslant 0. \tag{9}$$

Описанная *процедура решения задач выпуклого программирования* схематически представлена на рис. 9.1.

Следует отметить, что в отличие от линейного программирования в данном случае приходится запоминать все полученные в процессе решения столбцы матрицы ограничений и соответствующие коэффициенты целевой функции z-задачи. Определенные затруднения может вызвать и решение локальной задачи, поскольку в общем случае она не является линейной.

9.2. Пример. Рассмотрим задачу выпуклого программирования

$$f(x)=x_1^2+x_2^2\to\min,$$

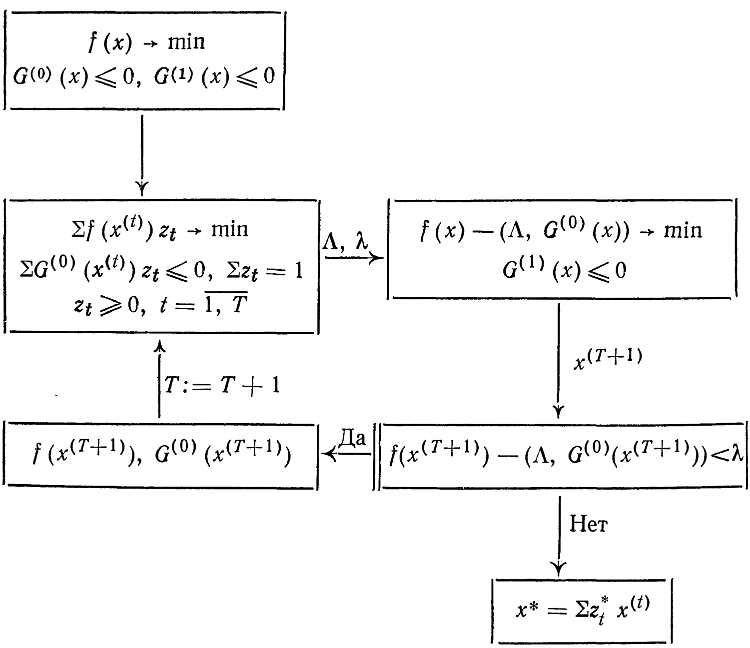

Рис. 9.1. Схема линеаризации с последовательной генерацией столбцов

$$G^{(0)}(x)=(x_1-3)^2+(x_2-3)^2-4\leqslant 0,$$
$$G^{(1)}(x)=x_1+x_2-6\leqslant 0.$$

Допустимая область представлена на рис. 9.2. Значения $x_1^*=x_2^*=3-\sqrt{2}\approx 1{,}6$, $f(x^*)\approx 5{,}2$.

Решим приведенную задачу методом генерации столбцов. Выберем в качестве исходных две точки $x^{(1)}=(2, 3)$ и $x^{(2)}=(1, 2)$, удовлетворяющие ограничению $G^{(1)}(x)\leqslant 0$. Проведем линеаризацию целевой функции и ограничения $G^{(0)}(x)\leqslant 0$ исходной задачи по этим двум точкам. Получаем z-задачу вида

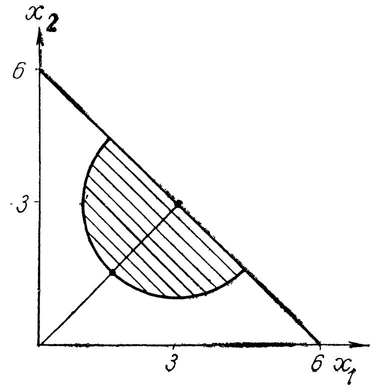

Рис. 9.2. Допустимая область задачи

$$13z_1+5z_2\to\min,$$
$$-3z_1+z_2\leqslant 0,$$
$$z_1+z_2=1,$$
$$z_1\geqslant 0,\ z_2\geqslant 0,$$

решением которой являются $z_1^*=1/4$, $z_2^*=3/4$. Значение $x(z^*)=\frac{1}{4}x^{(1)}+\frac{3}{4}x^{(2)}=\left(\frac{5}{4}, \frac{9}{4}\right)$, $f(x(z^*))\approx 6{,}6$.

Для нахождения новой точки $x^{(3)}$ определяем значения двойственных переменных z-задачи ($\Lambda=-2$, $\lambda=7$) и формулируем локальную задачу

$$f(x)-(\Lambda,\ G^{(0)}(x))=3[(x_1-2)^2+(x_2-2)^2]+4\to\min,$$
$$G^{(1)}(x)=x_1+x_2-6\leqslant 0.$$

В результате решения этой задачи получаем точку $x^{(3)}=(2, 2)$, значение целевой функции $4<\lambda=7$. Добавление точки $x^{(3)}$ к совокупности точек $x^{(1)}$ и $x^{(2)}$ позволяет сформулировать новую z-задачу:

$$13z_1+5z_2+8z_3\to\min,$$
$$-3z_1+z_2-2z_3\leqslant 0,$$
$$z_1+z_2+z_3=1,$$
$$z_1\geqslant 0,\ z_2\geqslant 0,\ z_3\geqslant 0$$

и продолжить процесс решения.

9.3. Следует отметить, что если $\Lambda\leqslant 0$, то $(\Lambda, G^{(0)}(x))\geqslant\geqslant 0$ для любой точки x, удовлетворяющей ограничению (2). В данном случае значение целевой функции задачи (8)—(9) является нижней оценкой значения целевой функции задачи (1)—(3). Таким образом, на каждой итерации в результате решения очередной z-задачи и соответствующей локальной задачи могут быть найдены верхняя и нижняя *оценки* искомого значения f^* целевой функции исходной задачи

$$f(x^{(T+1)})-(\Lambda,\ G^{(0)}(x^{(T+1)}))\leqslant f^*\leqslant f(x(z^*))\leqslant\sum_{t=1}^{T} z_t^* f(x^{(t)}).$$

В рассматриваемом выше примере величина, стоящая в левой части этого неравенства, равна 4, в правой 7, $f(x(z^*))\approx 6{,}6$, искомое значение $f^*\approx 5{,}2$.

9.4. Рассмотренные выше ситуации, связанные с необходимостью последовательного получения столбцов в процессе решения задачи, позволяют проследить некоторые *общие закономерности* формирования соответствующих вычислительных схем. Знание их в свою очередь дает возможность значительно расширить область применения метода генерации столбцов.

Как при решении задач линейного программирования описанным выше способом разбиения ее ограничений на две группы и формирования z-задачи, так и при решении задач нелинейного программирования способом линеаризации на сетке возникает необходимость решения некоторой линейной задачи с большим числом переменных и сравнительно небольшим числом ограничений. В общем случае не исключено, что она имеет бесконечное число переменных (что может наблюдаться, например, при линеаризации существенно нелинейных функций).

Запишем рассматриваемую линейную задачу в виде

$$\sum_{j=1}^{N} r_j y_j \to \min, \tag{10}$$

$$\sum_{j=1}^{N} P_j y_j = q, \tag{11}$$

$$y_j \geqslant 0,\ j=\overline{1, N}, \tag{12}$$

где $r=(r_1, r_2, \ldots, r_N) \in R^N$, $q=(q_1, q_2, \ldots, q_m) \in R^m$ — векторы; $P_1, P_2, \ldots, P_N$ — столбцы матрицы ограничений; y_j, $j=\overline{1, N}$,— переменные задачи.

Предположим, что известны некоторая матрица Q базиса этой задачи и вектор r_Q коэффициентов целевой функции, соответствующий матрице Q. Отметим, что размерность матрицы Q равна $m \times m$, т. е. по предположению сравнительно невелика, вектор r_Q содержит m компонент.

Если бы были известны все r_j и P_j, то при относительно небольшом N можно вычислить величину

$$\Delta = \min_{1 \leqslant j \leqslant N} \{\Delta_j\} = \min_{1 \leqslant j \leqslant n} \{r_j - (\Lambda, P_j)\},$$

где $\Lambda = r_Q Q^{-1}$ — вектор значений двойственных переменных. В зависимости от знака Δ можно сделать заключение об окончании процесса решения задачи либо о необходимости перехода к новой матрице базиса. Включаемый в базис вектор P_k обычно определяется из условия $r_k - (\Lambda, P_k) = \Delta$. Исключаемый из текущего базиса вектор P_s определяется в результате проведения сравнительно простых вычислений (см. параграф 3, п. 3.2).

Иначе обстоит дело, когда все r_j и P_j заранее неизвестны и процесс их получения сопряжен с определенными затруднениями (нахождение всех вершин многогранника X в линейном программировании, построение достаточно «частой» сетки выпуклого множества в нелинейном программировании и т. п.). В этом случае, как уже отмечалось, результативным оказывается следующий прием.

Рассмотрим вектор параметров $u=(u_1, u_2, \ldots, u_n) \in R^n$ с областью изменения U, содержащей точки $u^{(1)}$, $u^{(2)}, \ldots, u^{(N)}$. Зададим на U функцию $r(u)$ и вектор-функцию $P(u)$ таким образом, чтобы $r(u^{(j)})=r_j$ и $P(u^{(j)})=P_j$ для всех $j=\overline{1, N}$. Положим $\Delta(u)=r(u)-(\Lambda, P(u))$.

Пусть введенные конструкции таковы, что

$$\min_{u\in U}\{\Delta(u)\}=\min_{1\leqslant j\leqslant N}\{\Delta(u^{(j)})\}=\Delta \tag{13}$$

и выбранный способ минимизации $\Delta(u)$ на U обеспечивает получение экстремальной точки $u^{(k)}$ из множества $\{u^{(1)}, u^{(2)}, \ldots, u^{(N)}\}$. Тогда вся необходимая информация для следующего итерационного шага может быть получена в результате решения вспомогательной (локальной) задачи

$$r(u)-(\Lambda, P(u))\to\min, \tag{14}$$

$$u\in U. \tag{15}$$

Действительно, значение целевой функции этой задачи по определению равно Δ и минимум достигается в некоторой точке $u^{(k)}\in\{u^{(1)}, u^{(2)}, \ldots, u^{(N)}\}$. Если $\Delta<0$, то очередным включаемым в базис столбцом является $P_k=$ $=P(u^{(k)})$, соответствующее значение $r_k=r(u^{(k)})$.

Напомним, что при описании метода генерации столбцов в линейном программировании (см. параграф 7) роль u играл вектор x, в качестве U выбирался многогранник X (случай ограниченного множества X), точками $u^{(1)}$, $u^{(2)}, \ldots, u^{(N)}$ служили его вершины. Целевая функция (14) локальной задачи оказывалась линейной и тем самым обеспечивалась выполнимость условия (13). Для решения локальной задачи использовался любой метод, с помощью которого можно получить экстремальные точки — вершины многогранника X. Функции $r(u)$ и $P(u)$ в рассматриваемом случае имели вид (c, x) и $A^{(0)}x$ и позволяли найти очередной столбец матрицы ограничений и значение соответствующего коэффициента целевой функции z-задачи.

При решении нелинейных задач методом последовательной линеаризации (см. параграф 9, п. 9.1) в качестве u также выбирался вектор x, множество U описывалось ограничением $G^{(1)}(x)\leqslant 0$, роль $u^{(j)}$ играла любая точка этого множества (что могло приводить к сколь угодно большому «расширению» z-задачи).

9.5. Разбиение ограничений исходной задачи на группы и использование одной из них для описания множества U наиболее распространенный, но не единственный прием эффективного применения метода генерации столбцов. Возможны, конечно, и другие приемы, обусловленные особенностями конкретной задачи.

Рассмотрим простейшую задачу из класса *задач о раскрое*. Имеется неограниченное число рулонов бумаги длиной L. Пользователю необходимо иметь q_1 кусков бумаги длиной l_1, q_2 кусков длиной l_2 и т. д., q_m кусков длиной l_m. Значения $l_i < L$, $i=\overline{1, m}$. Требуется выбрать такие способы раскроя рулонов бумаги, чтобы удовлетворить потребности в кусках и разрезать при этом минимальное число рулонов.

Способ раскроя рулона характеризуется вектором $P_j = (p_{1j}, p_{2j}, \ldots, p_{mj})$, где p_{ij} — число кусков длиной l_i, получаемых из рулона. Число N способов раскроя, очевидно, конечно, но в практических ситуациях может быть чрезвычайно большим.

Полагая $q=(q_1, q_2, \ldots, q_m)$ и обозначая через y_j число рулонов, раскраиваемых по j-му способу, приходим к необходимости решения задачи (10)—(12) с $r_j=1$, $j=\overline{1, N}$, при дополнительном условии целочисленности переменных y_j. Это условие существенно осложняет процесс ее решения и на практике обычно ограничиваются решением обычной линейной задачи с последующим округлением результатов.

Итак, требуется решить задачу (10)—(12) с $r_j=1$, $j=\overline{1, N}$, и настолько большим N, что перечисление всех способов раскроя (столбцов P_j) практически исключено.

Пусть U — множество векторов $u=(u_1, u_2, \ldots, u_m)$ с неотрицательными целочисленными компонентами, удовлетворяющих условию $\sum_{i=1}^{m} l_i u_i \leqslant L$. Положим $P(u)=u$, $r(u) \equiv 1$.

Тогда для получения очередного столбца P_k достаточно, очевидно, решить локальную задачу вида

$$(\Lambda, u) \to \max, \tag{16}$$

$$\sum_{i=1}^{m} l_i u_i \leqslant L, \tag{17}$$

$$u_i \geqslant 0, \ u_i, \ i=\overline{1, m}, \text{— целые числа.} \tag{18}$$

Такая задача широко известна в литературе как задача о рюкзаке и при сравнительно небольшом значении m может быть решена любым из известных методов.

9.6. Пример. Пусть $L=10$, $l_1=2$, $l_2=3$, $l_3=4$, $q_1=9$, $q_2=12$, $q_3=8$.

Способов раскроя в рассматриваемой ситуации относительно много. Так, рулон можно раскроить на пять равных кусков длиной 2 — способ $P_{j_1}=(5, 0, 0)$; на два куска длиной 2 и два куска длиной 3 — способ $P_{j_2}=(2, 2, 0)$; на куски длиной 2, 3 и 4 (получаем в остатке кусок длиной 1) — способ $P_{j_3}=(1, 1, 1)$ и т. д.

Применяя «естественную» стратегию (сначала получить необходимое количество кусков длиной 2, затем длиной 3 и 4), приходим к заключению, что потребуется 10 рулонов. При этом в остатках будет пять кусков длиной 2 и четыре куска длиной 1.

Воспользуемся описанным приемом.

Обозначая через y_j количество рулонов, раскраиваемых по способу P_j, задачу можно записать в виде

$$y_1+y_2+y_3+\ldots+y_N\to\min,$$

$$5y_1+0y_2+1y_3+\ldots+p_{1N}y_N\geqslant 9,$$

$$0y_1+3y_2+0y_3+\ldots+p_{2N}y_N\geqslant 12,$$

$$0y_1+0y_2+2y_3+\ldots+p_{3N}y_N\geqslant 8,$$

$$y_1\geqslant 0,\ y_2\geqslant 0,\ y_3\geqslant 0,\ \ldots,\ y_N\geqslant 0.$$

Здесь N — число способов раскроя, условия целочисленности переменных y_j опущено, выделены способы раскроя (5, 0, 0), (0, 3, 0) и (1, 0, 2), векторы-столбцы которых образуют матрицу базиса

$$Q_1=\begin{pmatrix}5 & 0 & 1\\ 0 & 3 & 0\\ 0 & 0 & 2\end{pmatrix}.$$

Значение $\Lambda^{(1)}=r_{Q_1}Q_1^{-1}=\left(\frac{1}{5}, \frac{1}{3}, \frac{2}{5}\right)$. Следовательно, локальную задачу (16)—(18) можно записать в виде

$$\frac{1}{5}u_1+\frac{1}{3}u_2+\frac{2}{5}u_3\to\max,$$

$$2u_1+3u_2+4u_3\leqslant 10,$$

$$u_1\geqslant 0,\ u_2\geqslant 0,\ u_3\geqslant 0,$$

$$u_1,\ u_2,\ u_3 \text{ — целые числа.}$$

Решая приведенную задачу, имеем $u_1^{(1)}=0$, $u_2^{(1)}=2$, $u_3^{(1)}=1$. Значение целевой функции $\left(\frac{16}{15}\right)$ больше значения $r(u)\equiv 1$. Следовательно, в матрицу базиса Q_1 необходимо включить столбец $(0, 2, 1)$, а исключить $(0, 3, 0)$.

Получаем новую матрицу базиса

$$Q_2=\begin{pmatrix}5 & 0 & 1\\ 0 & 2 & 0\\ 0 & 1 & 2\end{pmatrix}.$$

Величина $\Lambda^{(2)}=\left(\frac{1}{5}, \frac{3}{10}, \frac{2}{5}\right)$ и новая локальная задача запишется в виде

$$\frac{1}{5}u_1+\frac{3}{10}u_2+\frac{2}{5}u_3\to\max$$

при ограничениях предыдущей локальной задачи. Значение целевой функции, очевидно, равно 1. Поскольку $r(u)\equiv 1$, то Q_2 — матрица базиса, соответствующая оптимальному плану исходной задачи.

Находим $Q_2^{-1}q=\left(1\frac{3}{5}, 6, 1\right)$. Округляя полученный результат, приходим к заключению, что два рулона следует раскроить по способу $(5, 0, 0)$, шесть — по способу $(0, 2, 1)$ и один — по способу $(1, 0, 2)$. Всего потребуется девять рулонов, остатки — два куска длиной 2.

10. ГЕНЕРАЦИЯ И РЕЛАКСАЦИЯ ОГРАНИЧЕНИЙ

Описываемый в данном параграфе декомпозиционный подход к решению задач линейного программирования является двойственным по отношению к методу генерации столбцов.

10.1. Рассмотрим *задачу линейного программирования,* переменные которой разбиты на две группы x и y:

$$(c, x)+(d, y)\to\min, \quad (1)$$

$$Ax+By\geqslant b, \quad (2)$$

$$x\geqslant 0,\ y\geqslant 0. \quad (3)$$

Здесь A и B — матрицы размерности $m\times n_1$ и $m\times n_2$; $n_1+n_2=n$, b, c, d, x, y — векторы соответствующих размерностей.

Если зафиксировать переменные y (выбрав некоторый вектор $y\geqslant 0$), то задача (1)—(3) сводится к задаче

$$(c, x)\to\min(=f^*(y)), \quad (4)$$

$$Ax\geqslant b-By, \quad (5)$$

$$x\geqslant 0. \quad (6)$$

Задача, двойственная к приведенной, имеет вид

$$(b-By, \Lambda)\to\max(=f^*(y)), \quad (7)$$

$$A^{\tau}\Lambda\leqslant c, \quad (8)$$

$$\Lambda\geqslant 0, \quad (9)$$

где Λ — вектор двойственных переменных; τ — знак транспонирования.

Предположим для простоты рассуждений, что исходная задача такова, что многогранное множество, описываемое ограничениями (8)—(9), является непустым и ограниченным. При этом условии задача (7)—(9) (а следовательно, и задача (4)—(6)) имеет решение при любом $y\geqslant 0$.

Обозначая через $\Lambda^{(1)}, \Lambda^{(2)}, \ldots, \Lambda^{(T)}$ вершины многогранника (8)—(9), приходим к заключению, что

$$f^*(y)=\max_{1\leqslant t\leqslant T}\{(b-By, \Lambda^{(t)})\}$$

и, следовательно, исходную задачу (1)—(3) можно представить в виде

$$\max_{1\leqslant t\leqslant T} \{(b-By, \Lambda^{(t)})\}+(d, y)\to\min, \quad y\geqslant 0.$$

Введем в рассмотрение переменную z и запишем последнюю задачу в виде

$$z\to\min, \tag{10}$$

$$(b-By, \Lambda^{(t)})\leqslant z-dy, \quad t=\overline{1, T}, \tag{11}$$

$$y\geqslant 0. \tag{12}$$

Задача (10)—(12) эквивалентна исходной задаче (1)—(3) в том смысле, что если (y^*, z^*) — решение задачи (10)—(12), а x^* — решение задачи (4)—(6) при $y=y^*$, то (x^*, y^*) — решение исходной задачи (1)—(3). Задача (10)—(12) является задачей линейного программирования и в общем случае содержит большое число ограничений вида (11) (равное числу вершин многогранника (8)—(9)). Для ее решения целесообразно использовать *метод релаксации ограничений.*

Пусть на некотором k-м шаге релаксированная задача имеет вид

$$z\to\min, \tag{13}$$

$$(b-By, \Lambda^{(t)})\leqslant z-dy, \quad t\in\theta^{(k)}, \tag{14}$$

$$y\geqslant 0, \tag{15}$$

где $\theta^{(k)}$ — некоторое подмножество множества $\{1, 2, \ldots, T\}$. На первом шаге можно положить $\theta^{(1)}=\varnothing$.

Обозначим через $(y^{(k)}, z^{(k)})$ решение этой задачи, а через $\Lambda^{(t_k)}$ — решение задачи (7)—(9) при $y=y^{(k)}$. Значение $t_k\in\{1, 2, \ldots, T\}$.

Если $(b-By^{(k)}, \Lambda^{(t_k)})>z^{(k)}-(d, y^{(k)})$, то ограничение $(b-By, \Lambda^{(t_k)})\leqslant z-(d, y)$ не выполняется в точке $(y^{(k)}, z^{(k)})$ и его следует добавить к ограничениям релаксированной задачи, т. е. положить $\theta^{(k+1)}=\theta^{(k)}\cup\{t_k\}$.

Если $(b-By^{(k)}, \Lambda^{(t_k)})=z^{(k)}-(d, y^{(k)})$, то $(y^{(k)}, z^{(k)})$ — решение задачи (10)—(12).

Процесс решения задачи (1)—(3), схематически представленный на рис. 10.1, состоит в последовательном решении релаксированных задач (13)—(15).

Каждое вновь добавляемое ограничение *генерируется* посредством решения локальной задачи (7)—(9) при соответствующем значении y.

Поскольку задача (10)—(12) — задача линейного программирования, то тогда, когда $z^{(k)} > z^{(k-1)}$, в релаксированной задаче (13)—(15) можно удалить все ограничения вида (14) с нулевыми значениями двойственных переменных (в частности, несущественные в точке $(y^{(k)}, z^{(k)})$ ограничения). Множество индексов удаляемых ограничений на рисунке обозначено через $\overline{\theta}^{(k)}$. В резуль-

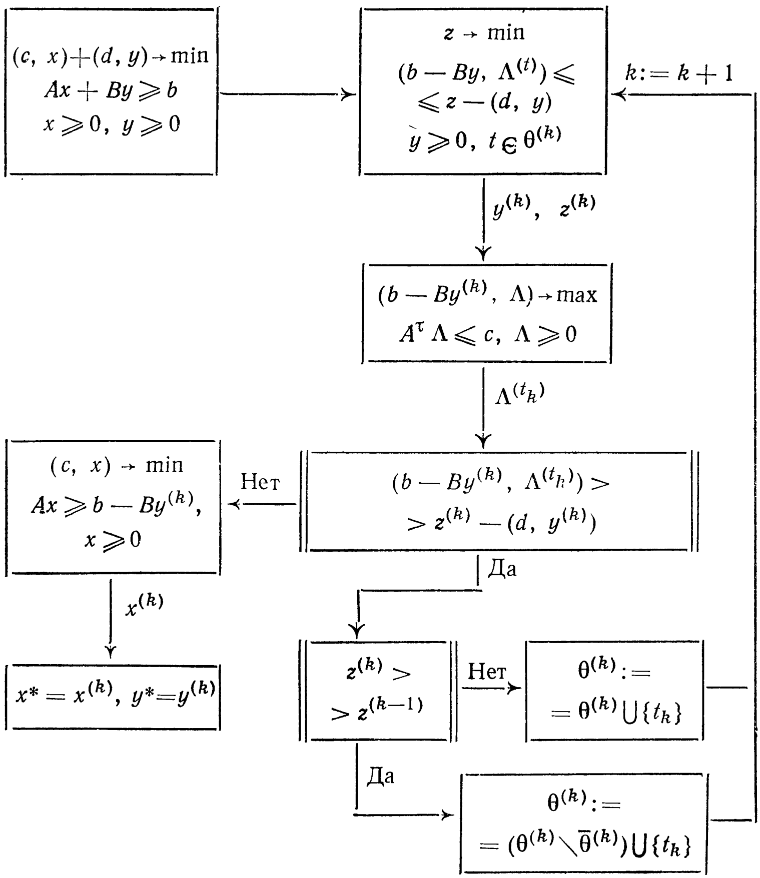

Рис. 10.1. Схема генерации и релаксации ограничений (линейное программирование)

тате этого процесса через конечное число шагов, скажем $k=\bar{k}$, будут получены такие $y^{(\bar{\bar{k}})}$ и $z^{(\bar{k})}$, что $(b-By^{(\bar{k})}, \Lambda^{(t_{\bar{k}})})=z^{(\bar{k})}-(d, y^{(\bar{\bar{k}})})$.

Решая задачу (4)—(6) при $y=y^{(\bar{\bar{k}})}$, получаем $x=x^{(\bar{k})}$.

Найденные значения $x^{(\bar{k})}$ и $y^{(\bar{\bar{k}})}$ являются решением исходной задачи (1)—(3).

10.2. Пример. Решим описанным выше методом задачу линейного программирования

$$8x_1+6x_2+14y\to\min,$$

$$x_1+x_2+2y\geqslant 3,$$

$$x_1-x_2+y\geqslant 1,$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ y\geqslant 0.$$

Полагая $y=\text{const}\geqslant 0$, получаем (см. задачу (4)—(6))

$$8x_1+6x_2\to\min(=f^*(y)),$$

$$x_1+x_2\geqslant 3-2y,$$

$$x_1-x_2\geqslant 1-y,$$

$$x_1\geqslant 0,\ x_2\geqslant 0.$$

Задача, двойственная к приведенной (см. задачу (7)—(9)), имеет вид

$$(3-2y)\lambda_1+(1-y)\lambda_2\to\max(=f^*(y)),$$

$$\lambda_1+\lambda_2\leqslant 8,$$

$$\lambda_1-\lambda_2\leqslant 6,$$

$$\lambda_1\geqslant 0,\ \lambda_2\geqslant 0.$$

Многогранник, описываемый ограничениями этой задачи, представлен на рис. 10.2. Вершинами многогранника являются точки $\Lambda^{(1)}=(0, 0)$, $\Lambda^{(2)}=(6, 0)$, $\Lambda^{(3)}=(7, 1)$, $\Lambda^{(4)}=(0, 8)$. Поскольку число вершин невелико, задачу (10)—(12) можно было бы записать в явном виде

$$z\to\min,$$

$$(3-2y)\cdot 0+(1-y)\cdot 0\leqslant z-14y,$$

$$(3-2y)\cdot 6+(1-y)\cdot 0\leqslant z-14y,$$
$$(3-2y)\cdot 7+(1-y)\cdot 1\leqslant z-14y,$$
$$(3-2y)\cdot 0+(1-y)\cdot 8\leqslant z-14y,$$
$$y\geqslant 0,$$

или

$$z\to\min,$$
$$z-14y\geqslant 0,$$
$$z-2y\geqslant 18,$$
$$z+y\geqslant 22,$$
$$z-6y\geqslant 8,$$
$$y\geqslant 0.$$

Следуя описанному методу, будем решать данную задачу, не располагая ее явной записью.

На первом шаге решаем релаксированную задачу $z\to\min$, $y\geqslant 0$. Ее решение: $y^{(1)}=0$, $z^{(1)}\to-\infty$ (в качестве $y^{(1)}$ можно выбрать любое неотрицательное число).

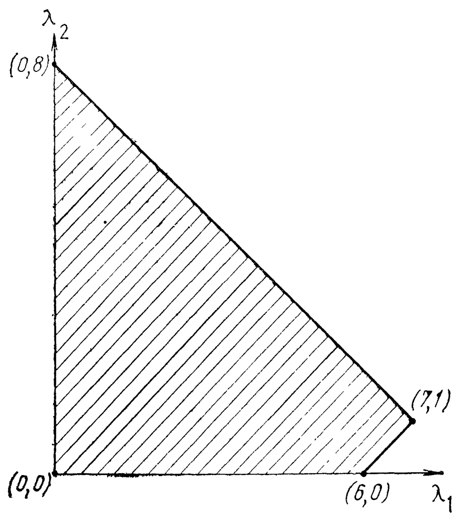

Рис. 10.2. Допустимая область двойственной задачи

Целевая функция двойственной задачи при $y=y^{(1)}=0$ имеет вид

$$3\lambda_1+\lambda_2\to\max(=f^*(y^{(1)})).$$

Следовательно, $\lambda_1^{(1)}=7$, $\lambda_2^{(1)}=1$. Поскольку оптимальное значение целевой функции двойственной задачи равно $22>z^{(1)}-14y^{(1)}\to-\infty$, то к ограничениям релаксированной задачи следует добавить ограничение $(3-2y)\cdot 7+ +(1-y)\cdot 1\leqslant z-14y$, т. е. $z+y\geqslant 22$.

На втором шаге решаем релаксированную задачу $z\to\min$, $z+y\geqslant 22$, $y\geqslant 0$. Ее решение: $y^{(2)}\to\infty$, $z^{(2)}\to-\infty$.

Коэффициенты целевой функции двойственной задачи при $y=y^{(2)}\to\infty$ отрицательны. Следовательно, $\lambda_1^{(2)}=0$ и $\lambda_2^{(2)}=0$. Поскольку оптимальное значение целевой функции двойственной задачи равно $0>z^{(2)}-14y^{(2)}\to-\infty$, то к ограничениям релаксированной задачи следует добавить ограничение $(3-2y)\cdot 0+(1-y)\cdot 0\leqslant z-14y$, т. е. $z-14y\geqslant 0$.

На третьем шаге решаем релаксированную задачу $z\to\min$, $z+y\geqslant 22$, $z-14y\geqslant 0$, $y\geqslant 0$. Ее решение: $y^{(3)}= =1\frac{7}{15}$, $z^{(3)}=20\frac{8}{15}$.

При $y=y^{(3)}=1\frac{7}{15}$ целевая функция двойственной задачи имеет вид

$$\frac{1}{15}\lambda_1-\frac{7}{15}\lambda_2\to\max.$$

Максимальное значение этой функции достигается при $\lambda_1^{(3)}=6$, $\lambda_2^{(3)}=0$ и равно $2/5$. Поскольку $2/5>z^{(3)}- -14y^{(3)}=0$, то к ограничениям релаксированной задачи следует добавить ограничение $(3-2y)\cdot 6+(1-y)\cdot 0\leqslant \leqslant z-14y$, т. е. $z-2y\geqslant 18$, и перейти к выполнению четвертого шага.

Решением релаксированной задачи $z\to\min$, $z+y\geqslant 22$, $z-14y\geqslant 0$, $z-2y\geqslant 18$, $y\geqslant 0$ являются $y^{(4)}=1\frac{1}{3}$, $z^{(4)}= =20\frac{2}{3}$.

Целевая функция двойственной задачи при $y=y^{(4)}= =1\frac{1}{3}$ имеет вид

$$\frac{1}{3}\lambda_1-\frac{1}{3}\lambda_2\to\max$$

и принимает наибольшее значение, равное 2 при $\lambda_1^{(4)}=6$, $\lambda_2^{(4)}=0$.

Поскольку $2=z^{(4)}-14y^{(4)}$, то $y^*=1\frac{1}{3}$.

Для нахождения x_1^* и x_2^* решаем задачу вида (4)—(6) при $y=y^*=1\frac{1}{3}$

$$8x_1+6x_2\to\min,$$

$$x_1+x_2\geqslant\frac{1}{3},$$

$$x_1-x_2\geqslant-\frac{1}{3},$$

$$x_1\geqslant0,\quad x_2\geqslant0.$$

В результате получаем $x_1^*=0$, $x_2^*=\frac{1}{3}$. Значения $x_1^*=0$, $x_2^*=\frac{1}{3}$, $y^*=1\frac{1}{3}$ — решение исходной задачи.

Отметим, что на первых трех шагах все ограничения релаксированных задач оказывались существенными.

10.3. Метод генерации и релаксации ограничений в линейном программировании является по существу методом, *двойственным* по отношению к методу генерации столбцов (см. параграф 7).

Действительно, рассмотрим задачу, двойственную к задаче (1)—(3):

$$(b,\ u)\to\max,\tag{16}$$

$$B^{\tau}u\leqslant d,\tag{17}$$

$$A^{\tau}u\leqslant c,\tag{18}$$

$$u\geqslant0.\tag{19}$$

Предполагая по-прежнему, что многогранное множество, описываемое ограничениями (18), (19), является непустым и ограниченным, соответствующую z-задачу можно записать в виде

$$\sum_{t=1}^{T} \sigma_t z_t \to \max, \tag{20}$$

$$\sum_{t=1}^{T} P_t z_t \leqslant d, \tag{21}$$

$$\sum_{t=1}^{T} z_t = 1, \tag{22}$$

$$z_t \geqslant 0, \quad t = \overline{1, T}, \tag{23}$$

где $\sigma_t = (b, u^{(t)})$, $P_t = B^{\tau} u^{(t)}$, $u^{(t)}$ — вершины многогранника (18)—(19), $t = \overline{1, T}$.

Задача, двойственная к z-задаче, запишется в виде

$$(d, y) + v \to \min,$$

$$(P_t, y) + v \geqslant \sigma_t, \quad t = \overline{1, T},$$

$$y \geqslant 0$$

или, обозначая $(d, y) + v$ через z и учитывая, что $\sigma_t = (b, u^{(t)})$ и $P_t = B^{\tau} u^{(t)}$, получаем

$$z \to \min,$$

$$(b - By, u^{(t)}) \leqslant z - (d, y), \quad t = \overline{1, T},$$

$$y \geqslant 0.$$

Здесь y — вектор двойственных переменных, соответствующих ограничениям (21); v соответствует ограничению (22).

Таким образом, задача, двойственная к z-задаче, является координирующей задачей (10)—(12).

Используя метод генерации столбцов для решения z-задачи (20)—(23), на каждом шаге решается локальная задача

$$(b - By, u) \to \max,$$

$$A^{\tau} u \leqslant c,$$

$$u \geqslant 0,$$

совпадающая с (7)—(9). В результате ее решения получается новый столбец, включаемый в очередную матрицу базиса z-задачи.

Аналогичная ситуация имеет место и в методе генерации и релаксации ограничений. Вновь генерируемое в результате решения локальной задачи ограничение добавляется ко множеству ограничений релаксированной задачи (13)—(15).

Наконец, признаками окончания процесса решения z-задачи (20)—(23) также, как и процесса решения координирующей задачи (10)—(12), служат аналогичные соотношения между оптимальным значением целевой функции локальной задачи и значением двойственной переменной v.

10.4. Пример. Задачей, двойственной к задаче, рассмотренной в предыдущем примере, является

$$3u_1+u_2\to \max,$$

$$2u_1+u_2\leqslant 14,$$

$$u_1+u_2\leqslant 8,$$

$$u_1-u_2\leqslant 6,$$

$$u_1\geqslant 0,\quad u_2\geqslant 0.$$

Поменяв знак целевой функции на противоположный и знак max на min, получим задачу, с точностью до обозначений совпадающую с задачей, рассмотренной в примере п. 7.2. Воспользовавшись приведенным там результатом, z-задачу можно записать в виде

$$0z_1+18z_2+22z_3+8z_4\to \max,$$

$$0z_1+12z_2+15z_3+8z_4\leqslant 14,$$

$$z_1+z_2+z_3+z_4=1,$$

$$z_1\geqslant 0,\ z_2\geqslant 0,\ z_3\geqslant 0,\ z_4\geqslant 0,$$

а двойственную к ней задачу в виде

$$14y+v\to \min,$$

$$0y+v\geqslant 0,$$

$$12y+v\geqslant 18,$$

$$15y+v\geqslant 22,$$
$$8y+v\geqslant 8,$$
$$y\geqslant 0.$$

Обозначая $14y+v$ через z, последнюю задачу можно переписать в виде

$$z\to\min,$$
$$z-14y\geqslant 0,$$
$$z-2y\geqslant 18,$$
$$z+y\geqslant 22,$$
$$z-6y\geqslant 8,$$
$$y\geqslant 0,$$

что совпадает с задачей, рассмотренной в предыдущем примере. Решая ее методом генерации и релаксации ограничений, на каждом шаге k необходимо решать локальную задачу

$$(3-2y^{(k)})\lambda_1+(1-y^{(k)})\lambda_2\to\max,$$
$$\lambda_1+\lambda_2\leqslant 8,$$
$$\lambda_1-\lambda_2\leqslant 6,$$
$$\lambda_1\geqslant 0,\ \ \lambda_2\geqslant 0.$$

Последняя задача с точностью до обозначений ($\Lambda=x$, $y=\Lambda$) совпадает с локальной, решаемой на шаге k, при решении z-задачи методом генерации столбцов.

Нетрудно убедиться также в эквивалентности критериев оптимальности, используемых в обоих методах.

11. НЕКОТОРЫЕ ОБОБЩЕНИЯ

Описанный в предыдущем параграфе подход к решению задач линейного программирования допускает естественные обобщения и может быть использован для решения задач нелинейного и частично целочисленного программирования. Прежде чем перейти к непосредственному описанию этих обобщений, сделаем несколько замечаний.

11.1. Рассмотрим систему линейных неравенств $Ax \leqslant b$. Здесь A — матрица размерности $m \times n$, x и b — векторы размерности n и m соответственно.

Известно, что непустое множество X значений x, удовлетворяющих системе неравенств $Ax \leqslant b$, является многогранником (ограниченным многогранным множеством) тогда и только тогда, когда однородная система неравенств $Ax \leqslant 0$ не имеет отличного от нуля решения. Если X — неограниченное множество, то направляющие векторы его неограниченных ребер совпадают с направляющими векторами ребер конуса, описываемого системой неравенств $Ax \leqslant 0$.

Рассмотрим систему линейных уравнений $Ax = b$. Для того чтобы существовал вектор $x \geqslant 0$, удовлетворяющий данной системе, необходимо и достаточно, чтобы для всех векторов $\Lambda \in R^m$ таких, что $A^\tau \Lambda \geqslant 0$, имело место неравенство $(b, \Lambda) \geqslant 0$. Здесь τ — знак транспонирования.

Такое утверждение непосредственно следует из соотношений двойственности в линейном программировании. Достаточно изучить пару двойственных задач

$$
\begin{array}{ll}
(0, x) \to \max\ (=0), & (b, \Lambda) \to \min, \\
Ax = b, & A^\tau \Lambda \geqslant 0, \\
x \geqslant 0. &
\end{array}
$$

Если они имеют решения и ими являются векторы x^* и Λ^*, то $(0, x^*) = (b, \Lambda^*) = 0$. При любых допустимых x и Λ значение $(b, \Lambda) \geqslant (0, x) = 0$. Если одна из пары двойственных друг другу линейных задач имеет неограниченное решение, то система ограничений второй задачи несовместна.

11.2. Рассмотрим задачу вида

$$(c, x)+\varphi(y)\to\min, \tag{1}$$

$$Ax+G(y)\leqslant b, \tag{2}$$

$$x\geqslant 0,\ y\in Y, \tag{3}$$

где A — матрица размерности $m\times n_1$; $G(y)=(g_1(y), g_2(y), \ldots, g_m(y))$ — вектор-функция с m компонентами; $\varphi(y)$, $g_i(y)$, $i=\overline{1, m}$,— функции от y; c и b — векторы размерности n_1 и m соответственно; x и y — искомые векторы размерности n_1 и n_2 соответственно; Y — некоторое множество n_2-мерных векторов.

При фиксированном $y\in Y$ задачу (1)—(3) можно переписать в виде

$$(c, x)\to\min\ (=f^*(y)), \tag{4}$$

$$Ax\leqslant b-G(y), \tag{5}$$

$$x\geqslant 0. \tag{6}$$

Сформулируем *условия*, при которых множество $X(y)$, описываемое системой неравенств (5)—(6), *не пусто*.

Введем в рассмотрение m-мерный вектор w и запишем неравенство (5) в виде равенства

$$Ax+Ew=b-G(y), \tag{7}$$

где $w\geqslant 0$; E — единичная матрица размерности.

Для того чтобы система уравнений (7) имела неотрицательное решение ($x\geqslant 0$, $w\geqslant 0$), необходимо и достаточно, чтобы для всех Λ таких, что $(AE)^\tau\Lambda\geqslant 0$, имело место неравенство $(b-G(y), \Lambda)\geqslant 0$. Или, что то же, для всех $\Lambda\geqslant 0$ таких, что $A^\tau\Lambda\geqslant 0$, имело место неравенство

$$(G(y)-b, \Lambda)\leqslant 0. \tag{8}$$

Обозначим через Λ_j^r, $j=\overline{1, n_r}$, направляющие векторы ребер конуса, описываемого системой неравенств $A^\tau\Lambda\geqslant 0$, $\Lambda\geqslant 0$, и представим любую точку Λ этого конуса в виде линейной комбинации указанных направляющих векторов:

$$\Lambda=\sum_{j=1}^{n_r}\mu_j\Lambda_j^r,\quad \mu_j\geqslant 0,\quad j=\overline{1, n_r}.$$

Подставляя найденное выражение для Λ в неравенство (8), получаем

$$\sum_{j=1}^{n_r} \mu_j (G(y)-b,\ \Lambda_j^r) \leqslant 0, \quad \mu_j \geqslant 0, \quad j=\overline{1,\ n_r}$$

или, поскольку μ_j — произвольные неотрицательные числа:

$$(G(y)-b,\ \Lambda_j^r) \leqslant 0, \quad j=\overline{1,\ n_r}. \tag{9}$$

Таким образом, множество $X(y) \neq \varnothing$ (или, что то же, система неравенств (5)—(6) совместна) тогда и только тогда, когда имеют место неравенства (9). Следовательно, при решении исходной задачи (1)—(3) можно рассматривать только такие значения $y \in Y$, которые удовлетворяют неравенствам (9).

Запишем задачу, *двойственную* к задаче (4)—(6):

$$(G(y)-b,\ \Lambda) \to \max\ (=f^*(y)), \tag{10}$$

$$A^{\tau}\Lambda \geqslant -c, \tag{11}$$

$$\Lambda \geqslant 0. \tag{12}$$

Если значение целевой функции данной задачи неограничено сверху, то система неравенств (5)—(6) несовместна (и, следовательно, несовместна система неравенств (9)).

Обозначим через Λ_i^p, $i=\overline{1,\ n_p}$, вершины многогранного множества, описываемого неравенствами (11)—(12). Имеем

$$f^*(y) = \max_{1 \leqslant i \leqslant n_p} \{(G(y)-b,\ \Lambda_i^p)\}.$$

В результате исходную задачу (1)—(3) можно переписать в виде

$$\max_{1 \leqslant i \leqslant n_p} \{(G(y)-b,\ \Lambda_i^p\} + \varphi(y) \to \min,$$

$$(G(y)-b,\ \Lambda_j^r) \leqslant 0, \quad j=\overline{1,\ n_r},$$

$$y \in Y$$

или, вводя новую переменную z:

$$z \to \min, \tag{13}$$

$$(G(y)-b,\ \Lambda_i^p) \leqslant z-\varphi(y),\quad i=\overline{1,\ n_p}, \tag{14}$$

$$(G(y)-b,\ \Lambda_j^r) \leqslant 0,\quad j=\overline{1,\ n_r}, \tag{15}$$

$$y \in Y. \tag{16}$$

Здесь Λ_i^p, $i=\overline{1,\ n_p}$,— вершины; Δ_j^r, $j=\overline{1,\ n_r}$,— направляющие векторы неограниченных ребер многогранного множества, описываемого системой неравенств (11)—(12).

Задача (13)—(16) эквивалентна задаче (1)—(3) в том смысле, что если будут найдены решение (y^*, z^*) задачи (13)—(16) и решение x^* задачи (4)—(6) при $y=y^*$, то (x^*, y^*) — решение исходной задачи (1)—(3).

Отметим, что задача (13)—(16), вообще говоря, нелинейна, а запись ее в явном виде в большинстве случаев невозможна из-за большого числа ограничений (14)—(15), равного числу вершин и неограниченных ребер многогранного множества (11)—(12).

11.3. Для решения задачи (13)—(16) целесообразно использовать *метод релаксации ограничений.*

Пусть на шаге k релаксированная задача имеет вид

$$z \to \min, \tag{17}$$

$$(G(y)-b,\ \Lambda_i^p) \leqslant z-\varphi(y),\quad i \in I^{(k)}, \tag{18}$$

$$(G(y)-b,\ \Lambda_j^r) \leqslant 0,\quad j \in J^{(k)}, \tag{19}$$

$$y \in Y, \tag{20}$$

где $I^{(k)} \subseteq \{1,\ 2,\ \ldots,\ n_p\}$; $J^{(k)} \subseteq \{1,\ 2,\ \ldots,\ n_r\}$. На первом шаге можно положить $I^{(1)}=J^{(1)}=\varnothing$.

Обозначим решение этой задачи через $y^{(k)}$, $z^{(k)}$ и перейдем к решению задачи (10)—(12) при $y=y^{(k)}$.

Возможны следующие случаи.

1. Задача (10)—(12) имеет решение $\Lambda_{i_k}^p$ и $f^*(y^{(k)}) > z^{(k)}-\varphi(y^{(k)})$. Ограничение $(G(y)-b,\ \Lambda_i^p) \leqslant z-\varphi(y)$ при $i=i_k$ нарушено в точке $(y^{(k)},\ z^{(k)})$ и его следует добавить

к ограничениям релаксированной задачи, т. е. положить $I^{(k+1)}=I^{(k)} \cup \{i_k\}$.

2. Значение целевой функции в задаче (10)—(12) неограничено возрастает вдоль ребра $\Lambda=\Lambda^p_{i_k}+\mu\Lambda^r_{j_k}$, $\mu \geqslant 0$, многогранного множества (11)—(12). Ограничение $(G(y)-b, \Lambda^r_j) \leqslant 0$ при $j=j_k$ нарушено в точке $(y^{(k)}, z^{(k)})$ и его следует добавить к ограничениям релаксированной задачи, т. е. положить $J^{(k+1)}=J^{(k)} \cup \{j_k\}$. Может оказаться, что в точке $(y^{(k)}, z^{(k)})$ нарушено и ограничение $(G(y)-b, \Lambda^p_i) \leqslant z-\varphi(y)$ при $i=i_k$. Его также следует добавить к ограничениям релаксированной задачи, т. е. положить $I^{(k+1)}=I^{(k)} \cup \{i_k\}$.

3. Задача (10)—(12) имеет решение $\Lambda^p_{i_k}$ и $f^*(y^{(k)})=$ $=z^{(k)}-\varphi(y^{(k)})$. В этом случае $(y^{(k)}, z^{(k)})$ является решением задачи (13)—(16). Если $x^{(k)}$ — решение (4)—(6) при $y=y^{(k)}$, то $(x^{(k)}, y^{(k)})$ — решение исходной задачи (1)—(3).

Описанный *процесс решения задачи (1)—(3)* схематически представлен на рис. 11.1.

Если задача (13)—(16) является задачей выпуклого или квазивыпуклого программирования, то наряду с добавлением указанных ограничений на шаге k могут удаляться несущественные в точке $(y^{(k)}, z^{(k)})$ ограничения (при условии, что $z^{(k)} > z^{(k-1)}$).

Отметим, что если система ограничений (11)—(12) несовместна, то значение целевой функции в задаче (4)—(6) при $y=y^{(k)}$ не ограничено снизу. Следовательно (при конечном $\varphi(y^{(k)})$), не ограничено снизу и значение целевой функции исходной задачи.

11.4. Пример. Применим описанный подход к решению задачи *частично целочисленного линейного программирования* следующего вида:

$$-x_1-x_2+x_3-2y_1-y_2 \to \min,$$

$$x_1+2x_2+0x_3+y_1+y_2 \leqslant 5,$$

$$3x_1+x_2-x_3+y_1-y_2 \leqslant 4,$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,$$

$$0 \leqslant y_1 \leqslant 10,\ 0 \leqslant y_2 \leqslant 10,$$

$$y_1,\ y_2 \text{ — целые числа.}$$

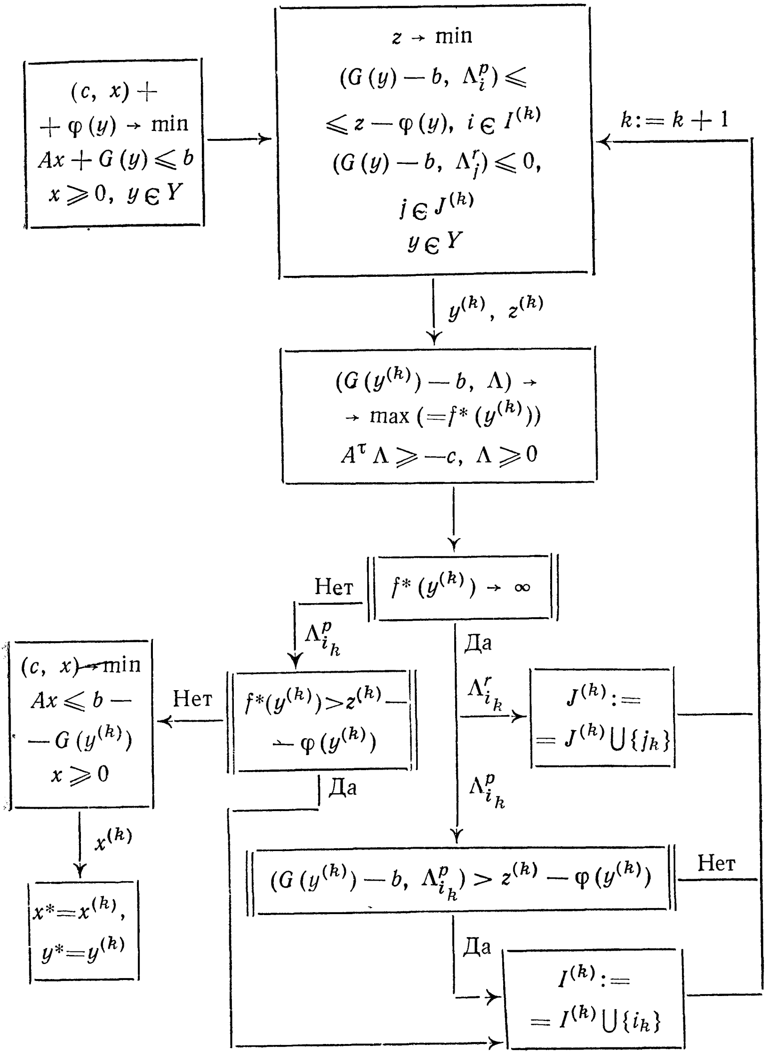

Рис. 11.1. Схема генерации и релаксации ограничений (нелинейное программирование)

Здесь $\varphi(y)=-2y_1-y_2$: $G(y)=(g_1(y), g_2(y))$, $g_1(y)=$ $=y_1+y_2$, $g_2(y)=y_1-y_2$, $y=(y_1, y_2)$; $Y=\{y \mid 0\leqslant y_1\leqslant 10$, $0\leqslant y_2\leqslant 10$, y_1, y_2 — целые числа$\}$.

При фиксированном $y\in Y$ имеем (см. задачу (4)—(6))

$$-x_1-x_2+x_3\to \min\ (=f^*(y)),$$

$$x_1+2x_2+0x_3\leqslant 5-y_1-y_2,$$

$$3x_1+x_2-x_3\leqslant 4-y_1+y_2,$$

$$x_1\geqslant 0,\ \ x_2\geqslant 0,\ \ x_3\geqslant 0.$$

Двойственной к этой задаче (см. задачу (10)—(12)) является задача

$$(y_1+y_2-5)\lambda_1+(y_1-y_2-4)\lambda_2\to \max\ (=f^*(y)),$$

$$\lambda_1+3\lambda_2\geqslant 1,$$

$$2\lambda_1+\lambda_2\geqslant 1,$$

$$\lambda_1\geqslant 0,\ \ 0\leqslant\lambda_2\leqslant 1.$$

Многогранное множество, описываемое ограничениями двойственной задачи, представлено на рис. 11.2.

На первом шаге полагаем $I^{(1)}=J^{(1)}=\varnothing$ и решаем релаксированную задачу (см. задачу (17)—(20)) $z\to\min$, $y\in Y$.

Значение целевой функции неограничено снизу на Y, т. е. $z^{(1)}\to-\infty$. В качестве $y^{(1)}$ можно выбрать любую точку из множества Y, для определенности $y^{(1)}=(10, 10)$.

При $y_1=y_1^{(1)}=10$ и $y_2=y_2^{(1)}=10$ целевая функция двойственной задачи записывается в виде $15\lambda_1-4\lambda_2\to$ $\to\max\ (=f^*(y^{(1)}))$ и неограничено возрастает вдоль ребра $\Lambda=\Lambda^p_{i_1}+\mu\Lambda^r_{j_1}$, $\mu\geqslant 0$, $\mu\to\infty$, где $\Lambda^p_{i_1}=(1, 0)$, $\Lambda^r_{j_1}=$ $=(1, 0)$. В точке $\Lambda^p_{i_1}=(1, 0)$ значение целевой функции двойственной задачи равно $15>z^{(1)}-\varphi(y^{(1)})\to-\infty$.

Следовательно, к ограничениям релаксированной задачи добавляются ограничения $(G(y)-b, \Lambda^r_{j_1})\leqslant 0$, т. е. $y_1+y_2-5\leqslant 0$, и $(G(y)-b, \Lambda^p_{i_1})\leqslant z-\varphi(y)$, т. е. y_1+y_2- $-5\leqslant z+2y_1+y_2$.

В результате получаем новую релаксированную задачу $z\to\min$, $y_1+y_2\leqslant 5$, $-y_1-5\leqslant z$, $y\in Y$.

Ее решением является $y^{(2)}=(5, 0)$, $z^{(2)}=-10$.

При $y_1=y_1^{(2)}=5$ и $y_2=y_2^{(2)}=0$ целевая функция двойственной задачи записывается в виде $0\lambda_1+\lambda_2\to$ $\to\max\ (=f^*(y^{(2)}))$ и достигает максимума, равного $f^*(y^{(2)})=1$, в любой точке $(\lambda_1,\ 1)$, $\lambda_1\geqslant 0$. Выберем для определенности точку $\Lambda^p_{i_2}=(0,\ 1)$. В этой точке значение $f^*(y^{(2)})=1>z^{(2)}-\varphi(y^{(2)})=-10+10=0$.

Следовательно, к ограничениям предыдущей релаксированной задачи добавляется ограничение $(G(y)-b,$ $\Lambda^p_{i_2})\leqslant z-\varphi(y)$, т. е. $y_1-y_2-4\leqslant z+2y_1+y_2$.

В результате получаем новую релаксированную задачу: $z\to\min$, y_1+ $+y_2\leqslant 5$, $-y_1-5\leqslant z$, $-y_1-2y_2-4\leqslant z$, $y\in Y$, решением которой является $y^{(3)}=(5,\ 0)$, $z^{(3)}=-9$.

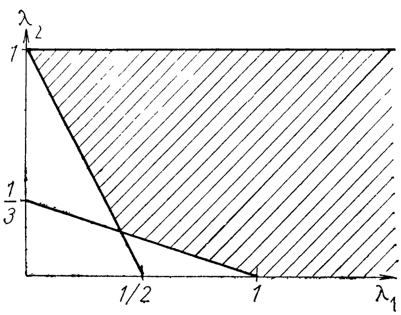

Рис. 11.2. Допустимая область двойственной задачи

Поскольку целевая функция двойственной задачи на этом шаге такая же, как и на предыдущем, т. е. $0\lambda_1+$ $+\lambda_2\to\max\ (=f^*(y^{(3)}))$, то $f^*(y^{(3)})=1$ и $f^*(y^{(3)})=z^{(3)}-$ $-\varphi(y^{(3)})=1$.

Таким образом, $y^*=(5,\ 0)$ и для нахождения x^* достаточно решить приведенную выше задачу линейного программирования при $y=y^*$:

$$-x_1-x_2+x_3\to\min,$$

$$x_1+2x_2+0x_3\leqslant 5-5-0=0,$$

$$3x_1+x_2-x_3\leqslant 4-5+0=-1,$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0.$$

Из первого ограничения и условий неотрицательности следует, что $x_1^*=x_2^*=0$ и задача сводится к минимизации x_3 при $x_3\geqslant 1$. Следовательно, $x_3^*=1$.

Окончательно $x_1^*=x_2^*=0$, $x_3^*=1$, $y_1^*=5$, $y_2^*=0$ — решение исходной задачи.

11.4. Таким образом, используя описанный в данном параграфе декомпозиционный подход, решение задачи частично целочисленного программирования оказалось возможным получить в результате решения серии задач линейного и целочисленного линейного программирования меньшей размерности.

При решении многих практических задач часто не требуется точное решение, достаточно найти приближенное с заданной абсолютной или относительной погрешностью по целевой функции. Покажем, что применение метода генерации и релаксации ограничений обеспечивает определение последовательностей *верхних и нижних оценок*, монотонно сходящихся к оптимальному значению целевой функции.

Действительно, решая релаксированную и двойственную задачи, на каждом шаге k вычисляем значения $y^{(k)}$, $z^{(k)}$ и $f^*(y^{(k)})$. Если величина $f^*(y^{(k)})$ ограничена, то система ограничений прямой задачи (4)—(6) при $y=y^{(k)}$ совместна и может быть найдено решение $x^{(k)}$ этой задачи. Поскольку $x^{(k)}$ и $y^{(k)}$ удовлетворяют ограничениям исходной задачи (1)—(3), то $(c, x^*)+\varphi(y^*)\leqslant(c, x^{(k)})+$ $+\varphi(y^{(k)})$. В рассматриваемом случае $(c, x^{(k)})=f^*(y^{(k)})$. Следовательно, $(c, x^*)+\varphi(y^*)\leqslant f^*(y^{(k)})+\varphi(y^{(k)})$. Это неравенство выполняется при любом k (в том числе и в случае, когда $f^*(y^{(k)})\to\infty$). Поэтому можно записать $(c, x^*)+\varphi(y^*)\leqslant\min\limits_{1\leqslant q\leqslant k}\{f^*(y^{(q)})+\varphi(y^{(q)})\}$.

Обозначая через z^* оптимальное значение целевой функции задачи (13)—(16), имеем $z^{(k)}\leqslant z^*=(c, x^*)+$ $+\varphi(y^*)$ при любом k.

Таким образом,

$$z^{(k)}\leqslant(c, x^*)+\varphi(y^*)\leqslant\min_{1\leqslant q\leqslant k}\{f^*(y^{(q)})+\varphi(y^{(q)})\}. \quad (21)$$

Поскольку $z^{(k-1)}\leqslant z^{(k)}$ и $\min\limits_{1\leqslant q\leqslant k}\{\cdot\}\leqslant\min\limits_{1\leqslant q\leqslant k-1}\{\cdot\}$, то по мере возрастания k указанный интервал нахождения оптимального значения целевой функции исходной задачи сужается до тех пор, пока (через конечное число шагов) его границы не совпадут друг с другом, что является, по существу, признаком окончания описанного процесса решения исходной задачи.

11.6. Примеры.

1. В условиях предыдущего примера имеем

$$y^{(1)}=(10,\ 10),\quad z^{(1)}\to-\infty,\quad f^*(y^{(1)})\to\infty,\quad \varphi(y^{(1)})=-30,$$

$$y^{(2)}=(5,\ 0),\quad z^{(2)}=-10,\quad f^*(y^{(2)})=1,\quad \varphi(y^{(2)})=-10,$$

$$y^{(3)}=(5,\ 0),\quad z^{(3)}=-9,\quad f^*(y^{(3)})=1,\quad \varphi(y^{(3)})=-10.$$

Подставляя последовательно эти значения в выражение (21), приходим к заключению, что оптимальное значение целевой функции исходной задачи содержится соответственно в интервалах $(-\infty,\ \infty)$, $[-10,\ -9]$, $[-9,\ -9]$.

Если бы исходную задачу требовалось решить приближенно с абсолютной погрешностью (по целевой функции), не превышающей 1, то можно было бы прекратить описанный процесс ее решения после выполнения второго шага.

2. Пусть дана *задача квадратичного программирования*

$$\begin{aligned}
&8x_1+5x_2+4x_3+y^2\to \min,\\
&2x_1+x_2\qquad\quad +y\geqslant 1,\\
&x_1+x_2\qquad\qquad\ \geqslant 3,\\
&x_1\qquad\ +x_3+y\geqslant 2,\\
&x_1\qquad\ -x_3\qquad \geqslant 1;\\
&x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ y\geqslant 0.
\end{aligned}$$

Найдем приближенное решение этой задачи, относительная погрешность которого (по целевой функции) не превышает 10%.

При $y=\text{const}\geqslant 0$ имеем

$$\begin{aligned}
&8x_1+5x_2+4x_3\to \min\ (=f^*(y)),\\
&2x_1+x_2\qquad \geqslant 1-y,\\
&x_1+x_2\qquad \geqslant 3,\\
&x_1\qquad +x_3\geqslant 2-y,\\
&x_1\qquad -x_3\geqslant 1,\\
&x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0.
\end{aligned}$$

Задача, двойственная к полученной, запишется в виде

$$\begin{aligned}
&(1-y)\lambda_1+3\lambda_2+(2-y)\lambda_3+\lambda_4\to \max\ (=f^*(y)),\\
&2\lambda_1+\lambda_2+\lambda_3+\lambda_4\leqslant 8,
\end{aligned}$$

$$\lambda_1+\lambda_2 \qquad \leqslant 5,$$

$$\lambda_3-\lambda_4\leqslant 4,$$

$$\lambda_1\geqslant 0,\ \lambda_2\geqslant 0,\ \lambda_3\geqslant 0,\ \lambda_4\geqslant 0.$$

На первом шаге, решая релаксированную задачу $z\to\min$, $y\geqslant 0$, получаем, что $z^{(1)}\to-\infty$ и для определенности $y^{(1)}=0$.

Целевая функция двойственной задачи при $y=y^{(1)}=0$ имеет вид $\lambda_1+3\lambda_2+2\lambda_3+\lambda_4\to\max\ (=f^*(y^{(1)}))$.

Приходим к ситуации, рассмотренной в п. 5.6 параграфа 5 (пример 2). Откуда $\lambda_1^{(1)}=0$, $\lambda_2^{(1)}=5$, $\lambda_3^{(1)}=3$, $\lambda_4^{(1)}=0$ и, следовательно, $f^*(y^{(1)})=21$. Поскольку $f^*(y^{(1)})>$ $>z^{(1)}-\varphi(y^{(1)})\to-\infty$, то к ограничениям релаксированной задачи следует добавить ограничение $15+(2-y)\times$ $\times 3\leqslant z-y^2$.

На втором шаге решаем релаксированную задачу $z\to\min$, $z\geqslant y^2-3y+21$, $y\geqslant 0$.

Функция $y^2-3y+21$ достигает наименьшего значения, равного $18\frac{3}{4}$ при $y=1\frac{1}{2}$. Следовательно, $y^{(2)}=$ $=1\frac{1}{2}$, $z^{(2)}=18\frac{3}{4}$.

Целевая функция двойственной задачи при $y=y^{(2)}=$ $=1\frac{1}{2}$ имеет вид $-\frac{1}{2}\lambda_1+3\lambda_2+\frac{1}{2}\lambda_3+\lambda_4\to$ $\to\max\ (=f^*(y^{(2)}))$.

Нетрудно убедиться, что решением двойственной задачи в этом случае являются $\lambda_1^{(2)}=0$, $\lambda_2^{(2)}=5$, $\lambda_3^{(2)}=0$, $\lambda_4^{(2)}=3$ и $f^*(y^{(2)})=18$.

Таким образом, оптимальное значение целевой функции исходной задачи не меньше $z^{(2)}=18\frac{3}{4}$ и не больше $f^*(y^{(2)})+\varphi(y^{(2)})=20\frac{1}{4}$, т. е. находится в интервале $\left[18\frac{3}{4},\ 20\frac{1}{4}\right]$, длина которого равна $1\frac{1}{2}$.

Следовательно, полагая в исходной задаче $y=y^{(2)}=$ $=1\frac{1}{2}$, получаем приближенное решение, относительная погрешность которого (по целевой функции) не превышает $\left(1\frac{1}{2} : 18\frac{3}{4}\right) \cdot 100\% = 8\%$.

Имеем

$$8x_1+5x_2+4x_3 \to \min\ (=f^*(y^{(2)})),$$

$$2x_1+x_2 \geqslant -\frac{1}{2},$$

$$x_1+x_2 \geqslant 3,$$

$$x_1+x_3 \geqslant \frac{1}{2},$$

$$x_1-x_3 \geqslant 1,$$

$$x_1 \geqslant 0,\quad x_2 \geqslant 0,\quad x_3 \geqslant 0.$$

Для решения этой задачи можно воспользоваться методом релаксации ограничений. Если исключить первое и третье неравенства, то минимум целевой функции будет достигаться в точке ($x_1=1$, $x_2=2$, $x_3=0$), удовлетворяющей первому и третьему неравенствам.

Таким образом, $x_1^{(2)}=1$, $x_2^{(2)}=2$, $x_3^{(2)}=0$, $y^{(2)}=1\frac{1}{2}$ — искомое приближенное решение исходной задачи.

12. ДВОЙСТВЕННОСТЬ И ДЕКОМПОЗИЦИЯ

При построении декомпозиционных схем решения задач математического программирования широко используются основные соотношения теории двойственности.

12.1. Рассмотрим *задачу выпуклого программирования*

$$f(x) \to \min, \tag{1}$$

$$g_i(x) \leqslant 0, \ i = \overline{1, m}, \tag{2}$$

$$x \in X, \tag{3}$$

где множество $X \subseteq R^n$ — выпуклое; функции $f(x)$ и $g_i(x)$, $i = \overline{1, m}$,— выпуклые на X.

Сопоставим приведенной задаче функцию Лагранжа

$$L(x, \Lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x), \tag{4}$$

где $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in R^m$.

Пара (x^0, Λ^0), $x^0 \in X$, $\Lambda^0 \geqslant 0$, называется *седловой точкой* функции Лагранжа, если для всех $x \in X$ и $\Lambda \geqslant 0$ выполняются неравенства $L(x^0, \Lambda) \leqslant L(x^0, \Lambda^0) \leqslant L(x, \Lambda^0)$.

Если (x^0, Λ^0) — седловая точка функции (4), то x^0 — решение задачи (1)—(3) и выполняется условие дополняющей нежесткости

$$\lambda_i^0 g_i(x^0) = 0, \quad i = \overline{1, m}. \tag{5}$$

Отметим, что это утверждение справедливо для любой (необязательно выпуклой) задачи математического программирования вида (1)—(3).

Для задачи выпуклого программирования вида (1)—(3), ограничения которой обладают свойством регулярности (существует точка $x' \in X$ такая, что $g_i(x') < 0$, $i = \overline{1, m}$), справедлива теорема Куна — Таккера: необходимым и достаточным условием оптимальности точки $x^0 \in X$ является существование такого $\Lambda^0 \geqslant 0$, что (x^0, Λ^0) — седловая точка функции Лагранжа (4). При этом выполняется условие дополняющей нежесткости (5).

Если все ограничения (2) линейные, а $f(x)$ по-прежнему выпуклая на X функция, то теорема Куна — Таккера справедлива без предположения о регулярности множества ограничений (2).

Указанные взаимосвязи между решением задачи математического программирования и седловой точкой функции Лагранжа позволяют в отмеченных ситуациях вместо непосредственного решения задачи (1)—(3) осуществлять поиск седловой точки функции (4).

Сформулируем *необходимые и достаточные условия*, при которых пара (x^0, Λ^0) является седловой точкой функции Лагранжа.

Для того чтобы пара (x^0, Λ^0) $x^0 \in X$, $\Lambda^0 \geqslant 0$, была седловой точкой функции Лагранжа, необходимо и достаточно, чтобы $g_i(x^0) \leqslant 0$, $i=\overline{1, m}$; $L(x^0, \Lambda^0) \leqslant L(x, \Lambda^0)$ для всех $x \in X$; $\lambda_i^0 g_i(x^0)=0$, $i=\overline{1, m}$.

Пусть функции $f(x)$ и $g_i(x)$, $i=\overline{1, m}$,— непрерывно дифференцируемые и $X=\{x|x \geqslant 0\}$. В этом случае пара (x^0, Λ^0), $x^0 \in X$, $\Lambda^0 \geqslant 0$ является седловой точкой функции Лагранжа тогда и только тогда, когда $g_i(x^0) \leqslant 0$, $i=\overline{1, m}$; $\partial L^0/\partial x_j \geqslant 0$, $j=\overline{1, n}$; $\lambda_i^0 g_i(x^0)=0$, $i=\overline{1, m}$; $(\partial L^0/\partial x_j) x_j^0=0$, $j=\overline{1, n}$. Здесь $\partial L^0/\partial x_j$ — значение частной производной от функции $L(x, \Lambda)$ по x_j, вычисленное в точке (x^0, Λ^0).

12.2. Введем понятия прямой и двойственной задачи и сформулируем основные соотношения двойственности. Предполагается, что ограничения (2) обладают свойством регулярности.

Перепишем задачу (1)—(3) в виде

$$\varphi(x) \to \min, \qquad (6)$$

$$x \in \bar{X}, \qquad (7)$$

где $\varphi(x)$ — оптимальное значение целевой функции в задаче $L(x, \Lambda) \to \max(=\varphi(x))$, $\Lambda \geqslant 0$; $\bar{X}$ — множество точек из X, в которых $\varphi(x) < \infty$.

Решение задачи (1)—(3) совпадает с решением задачи (6)—(7). Действительно, если $x' \in X$ и в точке x' нарушается хотя бы одно из ограничений (2), то $\varphi(x')=$

$=\infty$. Если $x'\in X$ и точка x' удовлетворяет всем ограничениям (2), то $\varphi(x')=f(x')$. Наконец, если $x'\in X$ и $\varphi(x')<<\infty$, то точка x' удовлетворяет ограничениям (2).

Задачу (6)—(7) или эквивалентную ей задачу (1)—(3) называют *прямой* задачей выпуклого программирования.

Двойственной называют следующую задачу:

$$\psi(\Lambda)\to\max, \tag{8}$$

$$\Lambda\in\bar{\Lambda}, \tag{9}$$

где $\psi(\Lambda)$ — оптимальное значение целевой функции в задаче $L(x, \Lambda)\to\min(=\psi(\Lambda))$, $x\in X$; $\bar{\Lambda}$ — множество точек $\Lambda\geqslant0$, в которых $\psi(\Lambda)>-\infty$.

Для задачи линейного программирования двойственную задачу (функцию $\psi(\Lambda)$ и множество $\bar{\Lambda}$) можно записать в явном виде.

Действительно, рассмотрим задачу

$$(c, x)\to\min,$$

$$Ax-b\leqslant0,$$

$$x\geqslant0.$$

Функция Лагранжа для такой задачи запишется в виде

$$L(x, \Lambda)=(c, x)+(\Lambda, Ax-b)=(c+A^{\tau}\Lambda, x)-(b, \Lambda),$$

где τ — знак транспонирования.

Если среди компонент вектора $c+A^{\tau}\Lambda$ существует хотя бы одна отрицательная компонента, то $\psi(\Lambda)==\min\{L(x, \Lambda)\,|\,x\geqslant0\}\to-\infty$. Следовательно, множество $\bar{\Lambda}$ описывается неравенствами $c+A^{\tau}\Lambda\geqslant0$, при этом $\psi(\Lambda)=-(b, \Lambda)$.

Таким образом, двойственную задачу можно записать в виде

$$-(b, \Lambda)\to\max,$$

$$A^{\tau}\Lambda\geqslant-c,$$

$$\Lambda\geqslant0.$$

12.3. Между прямой и двойственной задачами существует тесная связь, находящая отражения в так называемых *соотношениях двойственности*. Перечислим некоторые из них.

1. Для существования решения x^0 прямой задачи необходимо (но не достаточно) существование решения Λ^0 двойственной задачи.

2. Для того чтобы x^0 и Λ^0 были решениями прямой и двойственной задач, необходимо и достаточно, чтобы пара (x^0, Λ^0) была седловой точкой функции Лагранжа.

3. Если $x \in \bar{X}$ и $\Lambda \in \bar{\Lambda}$, то $\varphi(x) \geqslant \psi(\Lambda)$. Для того чтобы $x^0 \in \bar{X}$ и $\Lambda^0 \in \bar{\Lambda}$ были решениями прямой и двойственной задач, необходимо и достаточно, чтобы $\varphi(x^0) = \psi(\Lambda^0)$. При этом выполняется условие дополняющей нежесткости: $\lambda_i^0 g_i(x^0) = 0$, $i = \overline{1, m}$.

4. Если функция $\varphi(x)$ не ограничена снизу на $\bar{X}$, то $\bar{\Lambda} = \varnothing$. Если функция $\psi(\Lambda)$ не ограничена сверху на $\bar{\Lambda}$, то $\bar{X} = \varnothing$.

5. Справедливо соотношение (теорема о минимаксе):

$$\min_{x \in X} \max_{\Lambda \geqslant 0} L(x, \Lambda) = \max_{\Lambda \geqslant 0} \min_{x \in X} L(x, \Lambda).$$

Рассмотрим некоторые полезные в дальнейшем свойства целевой функции $\psi(\Lambda)$ двойственной задачи (8)—(9).

Покажем, что $\psi(\Lambda)$ является *вогнутой* функцией на любом выпуклом подмножестве $\bar{\Lambda}'$ множества $\bar{\Lambda}$.

Действительно, выбирая точки $\Lambda^{(1)}, \Lambda^{(2)} \in \bar{\Lambda}'$, имеем

$$\psi(\alpha\Lambda^{(1)} + (1-\alpha)\Lambda^{(2)}) = \min\{L(x, \alpha\Lambda^{(1)} + (1-\alpha)\Lambda^{(2)}) \mid x \in X\} = \min\{\alpha L(x, \Lambda^{(1)}) + (1-\alpha) L(x, \Lambda^{(2)}) \mid x \in X\} \geqslant \alpha \min\{L(x, \Lambda^{(1)}) \mid x \in X\} + (1-\alpha)\min\{L(x, \Lambda^{(2)}) \mid x \in X\} = \alpha\psi(\Lambda^{(1)}) + (1-\alpha)\psi(\Lambda^{(2)})$$

для всех $0 \leqslant \alpha \leqslant 1$.

Из дифференцируемости функций $f(x)$ и $g_i(x)$, $i = \overline{1, m}$, на X не следует, вообще говоря, дифференцируемость функции $\psi(\Lambda)$ на $\bar{\Lambda}$.

12.4. П р и м е р. Рассмотрим задачу выпуклого программирования

$$f(x) = x_1^2 + x_2 \to \min,$$

$$g_1(x) = 1 - x_1 - x_2 \leqslant 0,$$

$$x \in X = \{x = (x_1, x_2) \mid 0 \leqslant x_1 \leqslant 1,\ 0 \leqslant x_2 \leqslant 1\}.$$

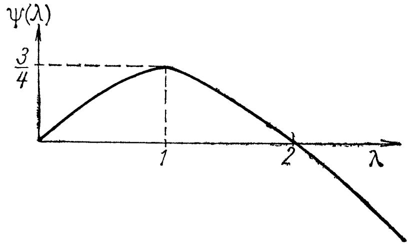

Рис. 12.1. График функции $\psi(\lambda)$

Функция Лагранжа в данном случае имеет вид

$$L(x, \lambda) = (x_1^2 + x_2) + \lambda(1 - x_1 - x_2) = \left(x_1 - \frac{\lambda}{2}\right)^2 + (1 - \lambda) x_2 + \lambda - \frac{\lambda^2}{4}.$$

Минимизируя ее по $x \in X$, получаем $\psi(\lambda) = \lambda - \frac{\lambda^2}{4}$ при $0 \leqslant \lambda \leqslant 1$, $\psi(\lambda) = 1 - \frac{\lambda^2}{4}$ при $1 < \lambda \leqslant 2$ и $\psi(\lambda) = 2 - \lambda$ при $\lambda > 2$.

График функции $\psi(\lambda)$ представлен на рис. 12.1. Эта функция является дифференцируемой во всех точках $\lambda \geqslant 0$, кроме $\lambda = 1$.

12.5. При сравнительно общих предположениях можно найти выражения для *частных производных и производных по направлению* функции $\psi(\Lambda)$. Для этого достаточно воспользоваться следующим известным утверждением.

Пусть дана некоторая непрерывная функция $H(y, z)$, где $y = (y_1, y_2, \ldots, y_p) \in R^p$, $z = (z_1, z_2, \ldots, z_q) \in R^q$, $y \in Y$, Y — ограниченное замкнутое множество в R^p. Предположим, что существуют частные производные функции H по z_i, $i = \overline{1, q}$, и эти производные непрерывны.

Введем в рассмотрение функцию $h(z) = \min\{H(y, z) \mid y \in Y\}$ и множество $Y^*(z) = \{y \mid H(y, z) = h(z)\}$.

Односторонней производной функции $h(z)$ по направлению $s=(s_1, s_2, \ldots, s_q) \in R^q$ в точке z называется предел отношения $h(z+\alpha s)-h(z)$ к α при $\alpha \to 0$, $\alpha \geqslant 0$. При указанных предположениях (односторонние) производные функции $h(z)$ по любому направлению s в любой точке z существуют и равны

$$\min \left\{ \sum_{i=1}^{q} s_i \frac{\partial H}{\partial z_i} \,\middle|\, y \in Y^*(z) \right\}.$$

Отсюда следует, например, что *правосторонняя* частная производная функции $h(z)$ по z_i

$$\left(\frac{\partial h}{\partial z_i}\right)^+ = \min \left\{ \frac{\partial H}{\partial z_i} \,\middle|\, y \in Y^*(z) \right\},$$

а *левосторонняя*

$$\left(\frac{\partial h}{\partial z_i}\right)^- = \max \left\{ \frac{\partial H}{\partial z_i} \,\middle|\, y \in Y^*(z) \right\}.$$

Полагая $y=x$, $z=\Lambda$, $Y=X$ и выбирая в качестве функции $H(y, z)$ функцию $L(x, \Lambda)$, приходим к заключению, что для существования односторонних производных функции $\psi(\Lambda)$ при любых $\Lambda \in \overline{\Lambda}$ достаточно потребовать, чтобы множество X было замкнутым и ограниченным, а функции $f(x)$ и $g_i(x)$, $i=\overline{1, m}$, непрерывными на X.

При этом (односторонняя) производная функции $\psi(\Lambda)$ в точке Λ по направлению $s=(s_1, s_2, \ldots, s_m) \in R^m$ равна

$$\min \left\{ \sum_{i=1}^{m} s_i g_i(x) \,\middle|\, x \in X^*(\Lambda) \right\},$$

где $X^*(\Lambda) = \{x \mid L(x, \Lambda) = \psi(\Lambda)\}$.

Частные производные функции $\psi(\Lambda)$ в точке Λ справа и слева вычисляются по формулам

$$\left(\frac{\partial \psi}{\partial \lambda_i}\right)^+ = \min \{g_i(x) \mid x \in X^*(\Lambda)\}$$

и

$$\left(\frac{\partial \psi}{\partial \lambda_i}\right)^- = \max \{g_i(x) \mid x \in X^*(\Lambda)\}.$$

Если множество X ограничено и замкнуто, а функции $f(x)$ и $g_i(x)$, $i=\overline{1, m}$, непрерывны на X, то функция $\psi(\Lambda)$ дифференцируема в точке $\Lambda \in \overline{\Lambda}$ тогда и только тогда, когда все функции $g_i(x)$ постоянны на $X^*(\Lambda)$. В этом случае в точке $\Lambda \in \overline{\Lambda}$

$$\frac{\partial \psi}{\partial \lambda_i}=g_i(x), \quad x \in X^*(\Lambda), \quad i=\overline{1, m}.$$

Ситуация такого рода имеет место, в частности, в том случае, когда множество $X^*(\Lambda)$ состоит только из одного элемента, т. е. $|X^*(\Lambda)|=1$. Единственность минимума функции $L(x, \Lambda)$ при фиксированном $\Lambda \in \overline{\Lambda}$ гарантируется, например, выпуклостью функций $g_i(x)$, $i=\overline{1, m}$, и строгой выпуклостью функции $f(x)$ на X.

Отметим, что приведенные сведения относительно дифференцируемости функции $\psi(\Lambda)$ справедливы для любой (необязательно выпуклой) задачи вида (1)—(3), обладающей указанными свойствами. Для задачи выпуклого программирования непрерывность функций $f(x)$ и $g_i(x)$, $i=\overline{1, m}$, во внутренних точках множества X непосредственно следует из их выпуклости на X.

12.6. Пример. По условиям предыдущего примера для любого числа $\lambda \geqslant 0$ имеем

$$\left(\frac{\partial \psi}{\partial \lambda}\right)^{+}=\min \{1-x_1-x_2 \mid x \in X^*(\lambda)\}$$

и

$$\left(\frac{\partial \psi}{\partial \lambda}\right)^{-}=\max \{1-x_1-x_2 \mid x \in X^*(\lambda)\}.$$

Если $\lambda=1$, то $L(x, \lambda)=L(x, 1)=(x_1-1/2)^2+3/4$. Множество $X^*(\lambda)=X^*(1)$ точек $x \in X$, в которых достигается минимум функции $L(x, 1)$, описывается соотношениями $x_1=1/2$, $0 \leqslant x_2 \leqslant 1$. Следовательно, $\left(\frac{\partial \psi}{\partial \lambda}\right)^{+}_{\lambda=1}=-\frac{1}{2}$, а $\left(\frac{\partial \psi}{\partial \lambda}\right)^{-}_{\lambda=1}=\frac{1}{2}$.

Если $0 \leqslant \lambda < 1$, то множество $X^*(\lambda)$ состоит из единственной точки $x_1 = \frac{\lambda}{2}$, $x_2 = 0$ и, следовательно, $\frac{\partial\psi}{\partial\lambda} = \left(\frac{\partial\psi}{\partial\lambda}\right)^+ = \left(\frac{\partial\psi}{\partial\lambda}\right)^- = 1 - \frac{\lambda}{2}$. Аналогично если $1 < \lambda \leqslant 2$, то множество $X^*(\lambda)$ состоит из единственной точки $x_1 = \frac{\lambda}{2}$, $x_2 = 1$ и $\frac{\partial\psi}{\partial\lambda} = -\frac{\lambda}{2}$. Наконец, если $\lambda > 2$, то множество $X^*(\lambda)$ состоит из единственной точки $x_1 = 1$, $x_2 = 1$ и $\frac{\partial\psi}{\partial\lambda} = -1$.

Отметим также, что седловой точкой функции Лагранжа в рассматриваемом случае является пара (x^0, λ^0), где $\lambda^0 = 1$, $x^0 \in X^*(1)$ и выполняется условие дополняющей нежесткости: $1(1 - x_1^0 - x_2^0) = 0$. Отсюда $x_1^0 = x_2^0 = \frac{1}{2}$.

12.7. Наряду с функцией Лагранжа $L(x, \Lambda)$ в современной вычислительной практике широкое распространение получили так называемые модифицированные функции Лагранжа, обладающие тем же множеством седловых точек и играющие ту же роль в теории оптимизации, что и классическая функция Лагранжа. Вместе с тем при соответствующей модификации функции $L(x, \Lambda)$ оказывается возможным избавиться от некоторых ее недостатков, в частности обеспечить гладкость целевой функции двойственной задачи.

Обычно в качестве модифицированной функции Лагранжа выбирают функцию вида

$$\tilde{L}(x, \Lambda) = f(x) + \sum_{i=1}^{m} \Phi_i(g_i(x), \lambda_i),$$

где каждая из функций $\Phi_i(y, z)$ двух переменных y и z является выпуклой и неубывающей по y (при $z \geqslant 0$) и вогнутой по z (при любом y). Предполагается также, что $\Phi_i(0, z) = 0$ для всех $z \geqslant 0$, функция $\Phi_i(y, z)$ дифференцируема по y в нуле и $\frac{\partial\Phi_i(0, z)}{\partial y} = z$ при любом $z \geqslant 0$.

При выполнении перечисленных условий любая функция вида $\tilde{L}(x, \Lambda)$ обладает тем же множеством седловых точек, что и функция $L(x, \Lambda)$. Достаточно разнообразные примеры функций $\tilde{L}(x, \Lambda)$ описываются выражениями вида

$$\Phi_i(y, z) = \eta_i(y) z + \theta_i(y),$$

где $\eta_i(y)$ и $\theta_i(y)$ — выпуклые и неубывающие функции одной переменной y; $\eta_i(y) < 0$ при $y < 0$, $\eta_i(0) = 0$; $\theta_i(y) = 0$ при $y \leqslant 0$; $\eta_i(y)$ и $\theta_i(y)$ дифференцируемы в нуле и $\eta_i'(0) = 1$.

Полагая $\eta_i(y) = y$, $\theta_i(y) \equiv 0$, $i = \overline{1, m}$, и тем самым $\Phi_i(y, z) = yz$, получаем обычную функцию Лагранжа

$$\tilde{L}(x, \Lambda) = L(x, \Lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x).$$

Если положить, например, $\eta_i(y) = y$, $\theta_i(y) = \beta_i [\max(0, y)]^2$, где число $\beta_i > 0$, $i = \overline{1, m}$, то $\Phi_i(y, z) = yz + \beta_i [\max(0, y)]^2$ и модифицированная функция Лагранжа имеет вид

$$\tilde{L}(x, \Lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{m} \beta_i [\max(0, g_i(x))]^2.$$

Перечень примеров модифицированных функций Лагранжа может быть естественно продолжен. Как уже отмечалось, эти функции в математическом программировании играют ту же роль, что и функция $L(x, \Lambda)$. В частности, с использованием модифицированных функций Лагранжа может быть построена такая же теория двойственности, как и с помощью обычной функции Лагранжа.

Обозначая через $\tilde{\varphi}(x)$ оптимальное значение целевой функции в задаче $\tilde{L}(x, \Lambda) \to \max$, $\Lambda \geqslant 0$, исходную задачу (1)—(3) можно переписать в виде

$$\tilde{\varphi}(x) \to \min, \tag{10}$$

$$x \in \tilde{X}, \tag{11}$$

где $\tilde{X}$ — множество точек из X, в которых $\tilde{\varphi}(x) < \infty$.

Двойственной к этой задаче является задача

$$\tilde{\psi}(\Lambda) \to \max, \tag{12}$$

$$\Lambda \in \tilde{\Lambda}, \tag{13}$$

где $\tilde{\psi}(\Lambda)$ — оптимальное значение целевой функции в задаче $\tilde{L}(x, \Lambda) \to \min$, $x \in X$; $\tilde{\Lambda}$ — множество точек $\Lambda \geqslant 0$, в которых $\tilde{\psi}(\Lambda) > -\infty$.

Для прямой и двойственной задач (10)—(11) и (12)—(13) справедливы те же соотношения двойственности, что и для задач (6)—(7) и (8)—(9).

При подходящем выборе функции $\tilde{L}(x, \Lambda)$ можно получить пару задач вида (10)—(11) и (12)—(13), более простых (в том или ином смысле), чем задачи (6)—(7) и (8)—(9). В частности, можно обеспечить весьма важное с вычислительной точки зрения условие гладкости целевой функции $\tilde{\psi}(\Lambda)$ двойственной задачи. Для этого достаточно, например, положить

$$\Phi_i(y, z) = \frac{1}{2\beta}\{[\max(0, z+\beta y)]^2 - z^2\}, \quad i = \overline{1, m},$$

где β — произвольное положительное число. Нетрудно убедиться, что эта функция обладает всеми перечисленными выше свойствами.

Соответствующая модифицированная функция Лагранжа запишется в виде

$$\tilde{L}(x, \Lambda) = f(x) + \frac{1}{2\beta}\sum_{i=1}^{m}[\max(0, \lambda_i + \beta g_i(x))]^2 - \frac{1}{2\beta}\sum_{i=1}^{m}\lambda_i^2. \tag{14}$$

Целевая функция $\tilde{\psi}(\Lambda)$ двойственной задачи оказывается дифференцируемой в любой точке $\Lambda \geqslant 0$ (предполагается, что $\tilde{\psi}(\Lambda) > -\infty$ для всех $\Lambda \geqslant 0$).

12.8. Пример. В условиях рассмотренного в п. 12.4 примера модифицированная функция Лагранжа (14) запишется в виде (полагаем для определенности $\beta = 1$)

$$\tilde{L}(x_1, x_2, \lambda) = (x_1^2 + x_2) + \frac{1}{2}[\max(0, \lambda + 1 - x_1 - x_2)]^2 - \frac{1}{2}\lambda^2.$$

Для нахождения $\tilde{\psi}(\lambda)$ данную функцию необходимо минимизировать на множестве X при фиксированном $\lambda \geqslant 0$. Напомним, что множество X в рассматриваемом случае описывается неравенствами $0 \leqslant x_1 \leqslant 1$, $0 \leqslant x_2 \leqslant 1$.

Определим частные производные функции $\tilde{L}(x_1, x_2, \lambda)$ по x_1 и x_2 и приравняем их нулю:

$$\begin{cases} 2x_1 - \max(0, \lambda+1-x_1-x_2) = 0, \\ 1 - \max(0, \lambda+1-x_1-x_2) = 0. \end{cases}$$

Получаем $x_1^*(\lambda) = \frac{1}{2}$, $x_2^*(\lambda) = \lambda - \frac{1}{2}$. Поскольку при $\frac{1}{2} < \lambda \leqslant 1\frac{1}{2}$ точка $\left(\frac{1}{2}, \lambda - \frac{1}{2}\right) \in X$, то $\tilde{\psi}(\Lambda) = -\frac{1}{2}\lambda^2 + \lambda + \frac{1}{4}$, $\frac{1}{2} < \lambda \leqslant 1\frac{1}{2}$.

Если $\lambda > 1\frac{1}{2}$, то минимум функции $\tilde{L}(x_1, x_2, \lambda)$ на множестве X достигается при $x_2 = 1$. Приходим к необходимости минимизации функции $\tilde{L}(x_1, 1, \lambda) = (x_1^2 + 1) + \frac{1}{2}(\lambda - x_1)^2 - \frac{1}{2}\lambda^2$ при условии, что $0 \leqslant x_1 \leqslant 1$. Приравнивая нулю производную этой функции по x_1, получаем $3x_1 - \lambda = 0$, откуда $x_1^*(\lambda) = \frac{1}{3}\lambda$. Поскольку при $1\frac{1}{2} < \lambda \leqslant 3$ точка $\left(\frac{1}{3}\lambda, 1\right) \in X$, то $\tilde{\psi}(\lambda) = -\frac{1}{6}\lambda^2 + 1$, $1\frac{1}{2} < \lambda \leqslant 3$.

Аналогично, если $0 \leqslant \lambda \leqslant \frac{1}{2}$, то минимум функции $\tilde{L}(x_1, x_2, \lambda)$ на множестве X достигается при $x_2 = 0$. Имеем: $\tilde{L}(x_1, 0, \lambda) = x_1^2 + \frac{1}{2}(\lambda + 1 - x_1)^2 - \frac{1}{2}\lambda^2$, $\frac{\partial \tilde{L}}{\partial x_1} = 2x_1 - (\lambda + 1 - x_1) = 0$, $x_1^*(\lambda) = \frac{\lambda+1}{3}$, $\tilde{\psi}(\lambda) = -\frac{1}{6}\lambda^2 + \frac{2}{3}\lambda + \frac{1}{3}$, $0 \leqslant \lambda \leqslant \frac{1}{2}$.

Наконец, если $\lambda > 3$, то минимум функции $\tilde{L}(x_1, x_2, \lambda)$

на множестве X достигается при $x_1=1$ и $x_2=1$. Следовательно, $\tilde{\psi}(\lambda)=-\lambda+2\frac{1}{2}$, $\lambda>3$.

Таким образом, $\tilde{\psi}(\lambda)=-\frac{1}{6}\lambda^2+\frac{2}{3}\lambda+\frac{1}{3}$ при $0\leqslant\lambda\leqslant\frac{1}{2}$, $\tilde{\psi}(\lambda)=-\frac{1}{2}\lambda^2+\lambda+\frac{1}{4}$ при $\frac{1}{2}<\lambda\leqslant 1\frac{1}{2}$, $\tilde{\psi}(\lambda)=-\frac{1}{6}\lambda^2+1$ при $1\frac{1}{2}<\lambda\leqslant 3$ и $\tilde{\psi}(\lambda)=-\lambda+2\frac{1}{2}$ при $\lambda>3$. График этой функции представлен на рис. 12.2.

Функция $\tilde{\psi}(\lambda)$ является вогнутой, дифференцируемой и для ее максимизации (при $\lambda\geqslant 0$) могут быть использованы, например, градиентные методы. Максимум этой функции, равный $\frac{3}{4}$, достигается при $\lambda=1$. Соответствующие значения $x_1^*(1)=\frac{1}{2}$, $x_2^*(1)=\frac{1}{2}$. Следовательно, седловой точкой функции $\tilde{L}(x_1, x_2, \lambda)$ является пара (x^0, λ^0), где $x^0=\left(\frac{1}{2}, \frac{1}{2}\right)$, $\lambda^0=1$.

12.9. *Методы декомпозиции, основанные на использовании функций Лагранжа,* получили широкое распространение при решении задач выпуклого сепарабельного программирования с блочными ограничениями. Основные идеи этого подхода можно проиллюстрировать, рассмотрев общую задачу линейного программирования вида

$$(c, x)\to\min,$$

$$Ax\leqslant b,$$

$$x\geqslant 0,$$

где A — матрица размерности $m\times n$; c, b, x — векторы соответствующих размерностей.

Представим рассматриваемую задачу в виде

$$(c, x)\to\min, \tag{15}$$

$$A^{(0)}x\leqslant b^{(0)}, \tag{16}$$

$$A^{(1)}x\leqslant b^{(1)}, \tag{17}$$

$$x\geqslant 0, \tag{18}$$

где $A^{(0)}$ и $A^{(1)}$ — матрицы размерности $m_0 \times n$ и $m_1 \times n$ соответственно, $m_0 + m_1 = m$; вектор $(b^{(0)}, b^{(1)}) = b$.

Обозначим многогранное множество, описываемое ограничениями (17)—(18), через X. Тогда исходную задачу можно записать в виде

$$(c, x) \to \min, \tag{19}$$

$$A^{(0)}x - b^{(0)} \leqslant 0, \tag{20}$$

$$x \in X. \tag{21}$$

По элементам этой задачи составим функцию Лагранжа: $L(x, \Lambda) = (c, x) + (\Lambda, A^{(0)}x - b^{(0)}) = (c + (A^{(0)})^{\tau}\Lambda, x) - (b^{(0)}, \Lambda)$, где $\Lambda \in R^{m_0}$.

Решение исходной задачи сводится к нахождению седловой точки $(x^{(0)}, \Lambda^{(0)})$, $x^0 \in X$, $\Lambda^{(0)} \geqslant 0$, функции $L(x, \Lambda)$, что в свою очередь сводится к решению *(координирующей)* задачи

$$\psi(\Lambda) \to \max, \tag{22}$$

$$\Lambda \geqslant 0, \tag{23}$$

где $\psi(\Lambda)$ — значение целевой функции *(локальной)* задачи

$$L(x, \Lambda) \to \min (= \psi(\Lambda)), \tag{24}$$

$$x \in X. \tag{25}$$

Обозначая вектор $(c + (A^{(0)})^{\tau}\Lambda)$ через d_Λ, локальную задачу можно записать в виде

$$(d_\Lambda, x) \to \min (= \psi_0(\Lambda)),$$

$$A^{(1)}x \leqslant b^{(1)},$$

$$x \geqslant 0,$$

а координирующую — в виде

$$\psi(\Lambda) = \psi_0(\Lambda) - (b^{(0)}, \Lambda),$$

$$\Lambda \geqslant 0.$$

Локальная задача при фиксированном Λ является задачей линейного программирования. Коэффициенты целевой функции этой задачи линейно зависят от Λ. Функция $\psi_0(\Lambda)$ — вогнутая кусочно-линейная. Координирующая задача состоит в максимизации вогнутой кусочно-линейной функции $\psi(\Lambda)$ при условии неотрицательности переменных. Отметим, что функция $\psi(\Lambda)$ задана неявно: для вычисления ее значения в некоторой точке Λ' необходимо решить локальную задачу при $\Lambda=\Lambda'$.

Если матрица $A^{(1)}$ — блочно-диагональная, то локальная задача (при фиксированном Λ) распадается на несколько подзадач меньшей размерности.

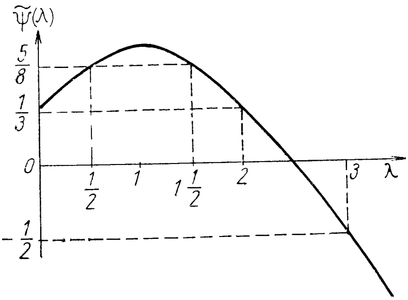

Рис. 12.2. График функции $\tilde{\psi}(\lambda)$

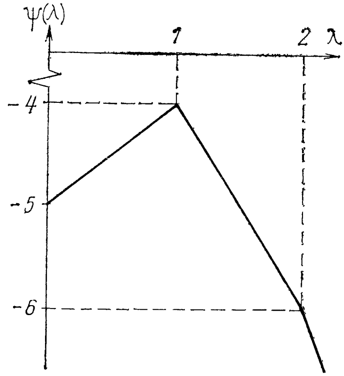

Рис. 12.3. График функции $\psi(\lambda)$

12.10. Пример. Рассмотрим задачу линейного программирования

$$\begin{aligned} -x_1-x_2-2x_3-3x_4&\to\min,\\ 2x_1+x_2+x_3+2x_4&\leqslant 3,\\ x_1+x_2&\leqslant 2,\\ x_3+x_4&\leqslant 1,\\ x_1\geqslant 0,\ x_2\geqslant 0,\ x_3&\geqslant 0,\ x_4\geqslant 0. \end{aligned}$$

Полагая $x=(x_1, x_2, x_3, x_4)$, $X=\{x\,|\,x_1+x_2\leqslant 2,\ x_3+x_4\leqslant 1,\ x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0\}$, задачу можно записать в виде

$$\begin{aligned} -x_1-x_2-2x_3-3x_4&\to\min,\\ 2x_1+x_2+x_3+2x_4-3&\leqslant 0,\\ x&\in X. \end{aligned}$$

Функция Лагранжа:

$$L(x, \lambda) = (-x_1 - x_2 - 2x_3 - 3x_4) + \lambda(2x_1 + x_2 + x_3 + 2x_4 - 3) =$$
$$= (2\lambda - 1)x_1 + (\lambda - 1)x_2 + (\lambda - 2)x_3 + (2\lambda - 3)x_4 - 3\lambda.$$

Локальная задача

$$(2\lambda-1)x_1 + (\lambda-1)x_2 + (\lambda-2)x_3 +$$
$$+ (2\lambda-3)x_4 \to \min\,(=\psi_0(\lambda)),$$
$$x_1 + x_2 \leqslant 2, \ x_3 + x_4 \leqslant 1,$$
$$x_1 \geqslant 0, \ x_2 \geqslant 0, \ x_3 \geqslant 0, \ x_4 \geqslant 0$$

при фиксированном λ распадается на две подзадачи:

$$\begin{aligned} &(2\lambda-1)x_1 + (\lambda-1)x_2 \to \min\,(=\psi_1(\lambda)), \\ &x_1 + x_2 \leqslant 2, \ x_1 \geqslant 0, \ x_2 \geqslant 0 \end{aligned} \qquad \text{(A)}$$

и

$$\begin{aligned} &(\lambda-2)x_3 + (2\lambda-3)x_4 \to \min\,(=\psi_2(\lambda)), \\ &x_3 + x_4 \leqslant 1, \ x_3 \geqslant 0, \ x_4 \geqslant 0. \end{aligned} \qquad \text{(B)}$$

Координирующая задача записывается в виде: $\psi(\lambda) = \psi_1(\lambda) + \psi_2(\lambda) - 3\lambda \to \max$, $\lambda \geqslant 0$. Отметим, что в рассматриваемом случае функции $\psi_1(\lambda)$ и $\psi_2(\lambda)$ (а следовательно, и функция $\psi(\lambda)$) могут быть выписаны в явном виде.

Действительно, при $0 \leqslant \lambda < 1$ решением подзадачи A является точка $x_1^*(\lambda) = 0$, $x_2^*(\lambda) = 2$; при $\lambda > 1$ — точка $x_1^*(\lambda) = 0$, $x_2^*(\lambda) = 0$. Если $\lambda = 1$, то $x_1^*(\lambda) = 0$, а $x_2^*(\lambda)$ может принимать любое значение из интервала $[0, 2]$. Следовательно, $\psi_1(\lambda) = 2\lambda - 2$ при $0 \leqslant \lambda < 1$ и $\psi_1(\lambda) \equiv 0$ при $\lambda \geqslant 1$.

Аналогично при $0 \leqslant \lambda < 1$ решением подзадачи B служит точка $x_3^*(\lambda) = 0$, $x_4^*(\lambda) = 1$; при $1 < \lambda < 2$ — точка $x_3^*(\lambda) = 1$, $x_4^*(\lambda) = 0$; при $\lambda > 2$ — точка $x_3^*(\lambda) = 0$, $x_4^*(\lambda) = 0$. Если $\lambda = 1$, то любая точка (x_3, x_4) с неотрицательными компонентами, удовлетворяющая условию $x_3 + x_4 = 1$, может быть решением подзадачи B. Наконец, $x_4^*(2) = 0$ и $x_3^*(2)$ может принимать любое значение из интервала $[0, 1]$. Следовательно, $\psi_2(\lambda) = 2\lambda - 3$ при $0 \leqslant \lambda < 1$, $\psi_2(\lambda) = \lambda - 2$ при $1 \leqslant \lambda < 2$ и $\psi_2(\lambda) \equiv 0$ при $\lambda \geqslant 2$.

Поскольку $\psi(\lambda)=\psi_1(\lambda)+\psi_2(\lambda)-3\lambda$, то $\psi(\lambda)=\lambda-5$ при $0\leqslant\lambda<1$, $\psi(\lambda)=-2\lambda-2$ при $1\leqslant\lambda<2$ и $\psi(\lambda)=-3\lambda$ при $\lambda\geqslant 2$. График функции $\psi(\lambda)$ представлен на рис. 12.3.

Максимум функции $\psi(\lambda)$ достигается в точке $\lambda^0=1$. Значения $x_1^*(1)=0$, $0\leqslant x_2^*(1)\leqslant 2$, $x_3^*(1)+x_4^*(1)=1$, $x_3^*(1)\geqslant 0$, $x_4^*(1)\geqslant 0$. Следовательно, компонента $x^0=(x_1^0, x_2^0, x_3^0, x_4^0)$ седловой точки (x^0, λ^0) функции Лагранжа удовлетворяет условиям: $x_1^0=0$, $0\leqslant x_2^0\leqslant 2$, $x_3^0+x_4^0=1$, $x_3^0\geqslant 0$, $x_4^0\geqslant 0$ и условию дополняющей нежесткости: $1(2x_1^0+x_2^0+x_3^0+2x_4^0-3)=0$. Отсюда $x_1^0=0$, $1\leqslant x_2^0\leqslant 2$, $x_3^0=x_2^0-1$, $x_4^0=2-x_2^0$.

Таким образом, в рассматриваемом случае функция Лагранжа имеет бесконечно много седловых точек, а исходная задача — бесконечно много решений. Для получения некоторого конкретного решения достаточно выбрать некоторое значение x_2^0 из интервала $[1, 2]$ и вычислить значения x_3^0 и x_4^0 по приведенным выше формулам (в любом случае $x_1^0=0$).

12.11. Эффективность методов декомпозиции, основанных на использовании функций Лагранжа (как и большинства известных методов декомпозиции), определяется тем, насколько «сбалансированными» оказываются трудоемкости решения координирующей и локальной задач. Это в свою очередь зависит от того, какие именно ограничения отнесены к описанию множества X, а какие использованы при формировании функции Лагранжа.

Рассмотрим, например, задачу выпуклого программирования

$$x_1^2+x_2^2\to\min,$$

$$(x_1-3)^2+(x_2-3)^2\leqslant 4,$$

$$x_1+2x_2\leqslant 9.$$

1. Обозначим через X множество точек $x=(x_1, x_2)$, удовлетворяющих неравенству $x_1+2x_2\leqslant 9$.

Функция Лагранжа:

$$L(x, \lambda)=(x_1^2+x_2^2)+\lambda[(x_1-3)^2+(x_2-3)^2-4]=$$

$$=(1+\lambda)\left[\left(x_1-\frac{3\lambda}{1+\lambda}\right)^2+\left(x_2-\frac{3\lambda}{1+\lambda}\right)^2+\frac{14\lambda-4\lambda^2}{(1+\lambda)^2}\right].$$

Минимизируя ее по x без ограничений, получаем $x_1^*(\lambda)=$ $=x_2^*(\lambda)=\dfrac{3\lambda}{1+\lambda}$ и $x_1^*(\lambda)+2x_2^*(\lambda)=\dfrac{9\lambda}{1+\lambda}\leqslant 9$ для всех $\lambda\geqslant 0$.

Следовательно, $\psi(\lambda)=\dfrac{14\lambda-4\lambda^2}{1+\lambda}$. Для нахождения максимума этой функции при $\lambda\geqslant 0$ вычислим ее производную $\psi'(\lambda)=\dfrac{14-8\lambda-4\lambda^2}{(1+\lambda)^2}$ и решим уравнение $\psi'(\lambda)=0$, т. е. $2\lambda^2+4\lambda-7=0$. Исследуя полученные решения, приходим к заключению, что $\lambda^0=-1+\dfrac{3\sqrt{2}}{2}\approx$ $\approx 1{,}12$.

Отсюда $x_1^0=x_2^0\approx x_1^*(1{,}12)=x_2^*(1{,}12)=1{,}58$ — решение исходной задачи.

2. Обозначим через X множество точек $x=(x_1,\ x_2)$, удовлетворяющих неравенству $(x_1-3)^2+(x_2-3)^2\leqslant 4$.

Функция Лагранжа:

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

$$=\left(x_1+\frac{\lambda}{2}\right)^2+(x_2+\lambda)^2-\frac{5}{4}\lambda^2-9\lambda.$$

Описанный выше релаксационный прием минимизации этой функции по x не применим, поскольку точка $(-\lambda/2,\ -\lambda)$ не принадлежит X при $\lambda\geqslant 0$. Тем самым приходим к необходимости непосредственной минимизации квадратичной функции $L(x,\ \lambda)$ по x (при фиксированном $\lambda\geqslant 0$) при квадратичном ограничении $(x_1-3)^2+$ $+(x_2-3)^2\leqslant 4$, описывающем множество X.

Поскольку множество X представляет собой круг радиуса 2 с центром в точке $(3,\ 3)$, а уравнение $z=L(x,\ \lambda)$ при фиксированных z и λ $\left(z>-\dfrac{5}{4}\lambda^2-9\lambda,\ \lambda\geqslant 0\right)$ описывает окружность с центром в точке $(-\lambda/2,\ -\lambda)$, то для минимизации $L(x,\ \lambda)$ по x достаточно найти ближайшую к $(-\lambda/2,\ -\lambda)$ точку пересечения прямой, проходящей через точки $(-\lambda/2,\ -\lambda)$ и $(3,\ 3)$ со множеством X. Иными словами, достаточно решить систему уравнений

$$\begin{cases}(x_1-3)^2+(x_2-3)^2=4,\\ (2x_1+\lambda)(3+\lambda)=(\cancel{x_2}+\lambda)(6+\lambda).\end{cases}$$

Имеем (выбирая подходящий знак корня)

$$x_1^*(\lambda)=\frac{-2(6+\lambda)}{\sqrt{5\lambda^2+36\lambda+72}}+3,$$

$$x_2^*(\lambda)=\frac{-4(3+\lambda)}{\sqrt{5\lambda^2+36\lambda+72}}+3.$$

Подставляя найденные значения x в $L(x, \lambda)$, получаем

$$\psi(\lambda)=-2\sqrt{5\lambda^2+36\lambda+72}+22.$$

Максимизируя эту функцию по $\lambda\geqslant 0$, приходим к заключению, что $\lambda^0=0$. Следовательно, $x_1^0=x_2^0=x_1^*(0)=$ $=x_2^*(0)=3-\dfrac{2}{\sqrt{2}}\approx 1{,}58.$

3. Обозначим через X множество всех точек $x=$ $=(x_1, x_2)\in R^2$. Положим $\Lambda=(\lambda_1, \lambda_2)$.

Функция Лагранжа:

$$L(x, \Lambda)=(x_1^2+x_2^2)+\lambda_1[(x_1-3)^2+(x_2-3)^2-4]+$$
$$+\lambda_2(x_1+2x_2-9)=(1+\lambda_1)\left[\left(x_1-\frac{6\lambda_1-\lambda_2}{2(1+\lambda_1)}\right)^2+\right.$$
$$\left.+\left(x_2-\frac{3\lambda_1-\lambda_2}{1+\lambda_1}\right)^2\right]+\frac{56\lambda_1-36\lambda_2-16\lambda_1^2-5\lambda_2^2}{4(1+\lambda_1)}.$$

Минимизируя ее по x (без ограничений), получаем

$$x_1^*(\Lambda)=\frac{6\lambda_1-\lambda_2}{2(1+\lambda_1)} \quad \text{и} \quad x_2^*(\Lambda)=\frac{3\lambda_1-\lambda_2}{1+\lambda_1}.$$

Максимизируя функцию

$$\psi(\Lambda)=\frac{56\lambda_1-36\lambda_2-16\lambda_1^2-5\lambda_2^2}{4(1+\lambda_1)}$$

при условии неотрицательности переменных, получаем $\lambda_2^0=0$ (поскольку коэффициенты при λ_2 отрицательны) и $\lambda_1^0=\dfrac{3\sqrt{2}}{2}-1\approx 1{,}12$ (находим производную $\psi(\Lambda)$ по λ_1 при условии, что $\lambda_2=0$, и приравниваем ее нулю).

Значения $x_1^0 = x_2^0 \approx x_1^* (1{,}12; 0) = x_2^* (1{,}12; 0) = 1{,}58$.

Таким образом, во всех случаях получено решение исходной задачи, но трудоемкость его получения далеко не одинакова в каждом из рассмотренных случаев.

12.12. При разработке методов декомпозиции задач математического программирования наряду с классической функцией Лагранжа могут быть использованы различные ее *модификации*.

Так, при решении задачи линейного программирования, представленной в форме (19)—(21), вместо функции $L(x, \Lambda)$ можно выбрать модифицированную функцию $\tilde{L}(x, \Lambda)$ вида (14):

$$\tilde{L}(x, \Lambda) = (c, x) + \frac{1}{2\beta} \left| (\Lambda + \beta (A^{(0)} x - b^{(0)}))^+ \right|^2 - \frac{1}{2\beta} |\Lambda|^2,$$

где β — произвольное положительное число и приняты следующие обозначения: если вектор $a = (a_1, a_2, \ldots, a_p)$, то $a^+ = (a_1^+, a_2^+, \ldots, a_p^+)$, $a_i^+ = \max(0, a_i)$, $|a|^2 = a_1^2 + a_2^2 + \ldots + a_p^2$.

В результате решение исходной задачи сводится к решению *(координирующей)* задачи

$$\tilde{\psi}(\Lambda) \to \max, \tag{26}$$

$$\Lambda \geqslant 0, \tag{27}$$

где $\tilde{\psi}(\Lambda)$ — значение целевой функции *(локальной)* задачи

$$\tilde{L}(x, \Lambda) \to \min (= \tilde{\psi}(\Lambda)), \tag{28}$$

$$A^{(1)} x \leqslant b^{(1)}, \tag{29}$$

$$x \geqslant 0. \tag{30}$$

Функция $\tilde{\psi}(\Lambda)$, как и ранее рассмотренная $\psi(\Lambda)$, является вогнутой. Однако в отличие от функции $\psi(\Lambda)$ она дифференцируема при любом $\Lambda \geqslant 0$ (при условии, что $\tilde{\psi}(\Lambda) > -\infty$). В этом отношении координирующая задача (26)—(27) имеет определенные преимущества перед координирующей задачей (22)—(23).

Функция $\tilde{L}(x, \Lambda)$ при фиксированном $\Lambda \geqslant 0$ в отличие от функции $L(x, \Lambda)$ не является линейной, более того, она не является и сепарабельной. Следовательно, если матрица $A^{(1)}$ имеет блочно-диагональную структуру, то локальная задача (28)—(30) (в отличие от локальной задачи (24)—(25)) не распадается на подзадачи меньшей размерности. Задача (28)—(30) относится к классу задач, близких к задачам квадратичного программирования, и ее решение требует использования соответствующих методов.

12.13. Пример. В условиях рассмотренного в п. 12.10 примера, полагая для определенности $\beta = 1$, имеем

$$\tilde{L}(x, \lambda) = (-x_1 - x_2 - 2x_3 - 3x_4) +$$

$$+ \frac{1}{2}[\max(0, \lambda + 2x_1 + x_2 + x_3 + 2x_4 - 3)]^2 - \frac{1}{2}\lambda^2.$$

Локальная задача (28)—(30) состоит в минимизации этой функции (при фиксированном $\lambda \geqslant 0$) на множестве X, описываемом неравенствами: $x_1 + x_2 \leqslant 2$, $x_3 + x_4 \leqslant 1$, $x_1 \geqslant 0$, $x_2 \geqslant 0$, $x_3 \geqslant 0$, $x_4 \geqslant 0$. Обозначая решение данной задачи через $x^*(\lambda)$, получаем $\tilde{\psi}(\lambda) = \tilde{L}(x^*(\lambda), \lambda)$.

Координирующая задача состоит в максимизации функции $\tilde{\psi}(\lambda)$ на множестве неотрицательных значений λ.

Производная функция $\tilde{\psi}(\lambda)$ в рассматриваемом случае записывается в явном виде:

$$\tilde{\psi}'(\lambda) = \max(0, \lambda + 2x_1^*(\lambda) + x_2^*(\lambda) + x_3^*(\lambda) +$$

$$+ 2x_4^*(\lambda) - 3) - \lambda.$$

Решение λ^0 координирующей задачи может быть получено путем организации итерационного процесса вида

$$\lambda_{k+1} = \lambda_k + \alpha_k \tilde{\psi}'(\lambda_k), \quad \alpha_k > 0, \quad k = 0, 1, \ldots,$$

на каждом шаге k которого решается локальная задача и вычисляется значение производной функции $\psi(\lambda)$ при $\lambda = \lambda_k$.

13. АГРЕГИРОВАНИЕ ПЕРЕМЕННЫХ ИЗ РАЗНЫХ БЛОКОВ

В настоящем параграфе рассматриваются задачи блочного линейного программирования со связывающими переменными. Декомпозиция исходной задачи на ряд задач меньшей размерности осуществляется путем введения новых переменных — агрегатов, представляющих собой сумму исходных переменных, взятых по одной из каждого блока.

13.1. Рассмотрим *задачу блочного линейного программирования* следующего вида:

$$\sum_{l=1}^{\nu} (\tilde{c}_l, \tilde{x}_l) \to \min, \tag{1}$$

$$\sum_{l=1}^{\nu} \tilde{A}_l \tilde{x}_l \leqslant b, \tag{2}$$

$$\tilde{B}_l \tilde{x}_l \leqslant \tilde{b}_l, \quad l = \overline{1, \nu}, \tag{3}$$

$$\tilde{x}_l \geqslant 0, \quad l = \overline{1, \nu}, \tag{4}$$

где $\tilde{A}_l = (a_{ij}^{(l)})$ — матрица размерности $m \times r$; $\tilde{B}_l = (b_{ij}^{(l)})$ — матрица размерности $m_l \times r$, $l = \overline{1, \nu}$; $\tilde{c}_l$, $\tilde{x}_l$, b, $\tilde{b}_l$ — векторы соответствующих размерностей.

Отметим, что каждый искомый вектор $\tilde{x}_l = (x_{l1}, x_{l2}, \ldots, x_{lr})$ имеет одну и ту же размерность, равную r. Если в исходной постановке векторы $\tilde{x}_l$ имеют различную размерность, то задачу можно свести к рассматриваемой, «удлинив» векторы до одинаковой размерности за счет введения новых компонент и «расширив» матрицы $\tilde{A}_l$ и $\tilde{B}_l$ соответствующим количеством нулевых столбцов. Векторы $\tilde{c}_l$ можно дополнить любыми неотрицательными компонентами до требуемой размерности. Можно также вновь введенные компоненты априори положить равными нулю.

Введем в рассмотрение новые переменные X_q, полагая

$$X_q = \sum_{l=1}^{\nu} x_{lq}, \quad q = \overline{1, r}. \tag{5}$$

Эти переменные называются *агрегированными* или *агрегатами* и представляют собой сумму соответствующих исходных переменных, взятых по одной из каждого блочного ограничения (3).

Соотношения (5) можно записать в виде

$$x_{lq}=\alpha_{lq}X_q,\ \alpha_{lq}\geqslant 0,\ \sum_{l=1}^{\nu}\alpha_{lq}=1,\ q=\overline{1,r},\ l=\overline{1,\nu},\qquad (6)$$

где величины α_{lq} характеризуют «вклад», который «вносят» переменные x_{lq}, $l=\overline{1,\nu}$, в суммарную, агрегированную переменную X_q, и называются *весами агрегирования.*

Проводя замену переменных, согласно соотношению (6), задачу (1)—(4) можно переписать в эквивалентной форме:

$$(c(\alpha),\ X)\to\min,$$

$$A(\alpha)X\leqslant b,$$

$$B_l(\alpha)X\leqslant\tilde{b}_l,\ l=\overline{1,\nu},$$

$$X=(X_1,\ X_2,\ \ldots,\ X_r)\geqslant 0,$$

$$\alpha=(\alpha_{11},\ \alpha_{12},\ \ldots,\ \alpha_{\nu r})\geqslant 0,$$

$$\sum_{l=1}^{\nu}\alpha_{lq}=1,\ q=\overline{1,r},$$

где вектор $c(\alpha)=(c_1(\alpha),\ c_2(\alpha),\ \ldots,\ c_r(\alpha))$, $c_q(\alpha)=\sum_{l=1}^{\nu}c_{lq}\alpha_{lq}$; матрица $A(\alpha)=(a_{iq}(\alpha))$, $a_{iq}(\alpha)=\sum_{l=1}^{\nu}a_{iq}^{(l)}\alpha_{lq}$, $i=\overline{1,m}$, $q=\overline{1,r}$; матрицы $B_l(\alpha)=(b_{iq}^{(l)}(\alpha))$, $b_{iq}^{(l)}(\alpha)=b_{iq}^{(l)}\alpha_{lq}$, $i=\overline{1,m_l}$, $l=\overline{1,\nu}$, $q=\overline{1,r}$.

Эта задача является *многопараметрической* задачей линейного программирования с относительно простыми ограничениями на параметры — веса агрегирования α_{lq}. Для ее решения можно использовать следующую двухуровневую схему.

Фиксируя веса агрегирования α_{lq}, получаем *(локальную)* задачу

$$(c,\ (\alpha),\ X)\to\min(=f^*(\alpha)),\qquad (7)$$

$$A(\alpha)X \leqslant b, \qquad (8)$$

$$B_l(\alpha)X \leqslant \tilde{b}_l, \quad l=\overline{1, \nu}, \qquad (9)$$

$$X \geqslant 0, \qquad (10)$$

содержащую в ν раз меньше переменных, чем исходная задача (1)—(4). Обозначим ее решение через $X^*(\alpha)$.

Поиск оптимальных весов агрегирования α^*_{lq} осуществляется путем решения *(координирующей)* задачи

$$f^*(\alpha) \to \min, \qquad (11)$$

$$\sum_{l=1}^{\nu} \alpha_{lq} = 1, \quad \alpha_{lq} \geqslant 0, \quad l=\overline{1, \nu}, \quad q=\overline{1, r}. \qquad (12)$$

Решением исходной задачи являются, очевидно, значения $x^*_{lq} = \alpha^*_{lq} X^*(\alpha^*_{lq})$.

13.2. Опишем одну из возможных *итерационных схем нахождения* α^*_{lq}.

Задачей, двойственной к локальной задаче (7)—(10), является следующая:

$$-(b, \Lambda) - \sum_{l=1}^{\nu} (\tilde{b}_l, \Lambda_l) \to \max(=f^*(\alpha)), \qquad (13)$$

$$[A(\alpha)]^{\tau}\Lambda + \sum_{l=1}^{\nu} [B_l(\alpha)]^{\tau}\Lambda_l \geqslant -c(\alpha), \qquad (14)$$

$$\Lambda \geqslant 0, \quad \Lambda_l \geqslant 0, \quad l=\overline{1, \nu}, \qquad (15)$$

где вектор двойственных переменных Λ соответствует ограничениям (8); векторы Λ_l, $l=\overline{1, \nu}$,— ограничениям (9); τ — знак транспонирования.

Обозначим решение этой задачи (при фиксированных α_{lq}) через $\Lambda^*(\alpha)$, $\Lambda^*_l(\alpha)$, $l=\overline{1, \nu}$.

Сформулируем исходную задачу (1)—(4) в терминах нахождения седловой точки функции Лагранжа

$$L(x, \Lambda) = \sum_{l=1}^{\nu} (\tilde{c}_l, \tilde{x}_l) + \left(\Lambda, \sum_{l=1}^{\nu} \tilde{A}_l\tilde{x}_l - b\right) =$$

$$= \sum_{l=1}^{\nu} (\tilde{c}_l + \tilde{A}^{\tau}_l \Lambda, \tilde{x}_l) - (b, \Lambda)$$

при ограничениях (3)—(4). Здесь $x=(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_\nu)$ и $\Lambda \geqslant 0$.

При фиксированных Λ задача минимизации функции $L(x, \Lambda)$ при ограничениях (3)—(4) распадается на ν подзадач $(l=\overline{1, \nu})$:

$$(\tilde{c}_l+\tilde{A}_l^\tau \Lambda, \tilde{x}_l) \to \min(=\varphi_l^*(\Lambda)), \tag{16}$$

$$\tilde{B}_l\tilde{x}_l \leqslant \tilde{b}_l, \quad \tilde{x}_l \geqslant 0. \tag{17}$$

Обозначим решения этих подзадач через $\tilde{x}_l^*(\Lambda)$, $l=\overline{1, \nu}$. Минимальное значение функции $L(x, \Lambda)$ при фиксированном Λ и ограничениях (3)—(4) равно

$$\varphi^*(\Lambda) = \sum_{l=1}^{\nu} \varphi_l^*(\Lambda) - (b, \Lambda).$$

Напомним, что если (x^0, Λ^0) — седловая точка функции $L(x, \Lambda)$, то $\varphi^*(\Lambda^0)=L(x^0, \Lambda^0)$, $\tilde{x}_l^0$, $l=\overline{1, \nu}$,— решение исходной задачи (1)—(4) и выполняется условие дополняющей нежесткости

$$\left(\Lambda^0, \sum_{l=1}^{\nu} \tilde{A}_l\tilde{x}_l^0 - b\right)=0. \tag{18}$$

Если при $\alpha_{lq}=\alpha_{lq}^0$, удовлетворяющих ограничениям (12), значение $f^*(\alpha^0)=\varphi^*(\Lambda^0)$, то α_{lq}^0 — искомые оптимальные веса агрегирования, т. е. $\alpha_{lq}^*=\alpha_{lq}^0$.

Итак, выберем произвольные веса агрегирования $\alpha^{(1)}=(\alpha_{lq}^{(1)})$ (удовлетворяющие ограничениям (12)). Решая прямую и двойственную локальные задачи (7)—(10) и (13)—(15) при $\alpha=\alpha^{(1)}$, находим $X^*(\alpha^{(1)})$ и $\Lambda^*(\alpha^{(1)})$.

Зная значения $X^*(\alpha^{(1)})$ агрегированных переменных и веса агрегирования $\alpha^{(1)}$, определяем значения исходных переменных $x_{lq}^{(1)}=\alpha_{lq}^{(1)}X_q^*(\alpha^{(1)})$ (соотношение (6)).

Решая подзадачи (16)—(17) при $\Lambda=\Lambda^*(\alpha^{(1)})$, вычисляем другие значения исходных переменных $\hat{x}_{lq}^{(1)}=$ $=x_{lq}^*(\Lambda^*(\alpha^{(1)}))$.

Если $f^*(\alpha^{(1)})=\varphi^*(\Lambda^*(\alpha^{(1)}))$, то $\alpha^*=\alpha^{(1)}$ и $x_{lq}^{(1)}$ — искомое решение исходной задачи. В противном случае приступаем к поиску новых весов агрегирования $\alpha^{(2)}=$ $=(\alpha_{lq}^{(2)})$.

Введем в рассмотрение νr функций одной переменной ρ:

$$\alpha_{lq}(\rho)=\frac{x_{lq}^{(1)}+\rho(\hat{x}_{lq}^{(1)}-x_{lq}^{(1)})}{\sum\limits_{l=1}^{\nu}(x_{lq}^{(1)}+\rho(\hat{x}_{lq}^{(1)}-x_{lq}^{(1)}))}$$

где $0\leqslant\rho\leqslant 1$.

Очевидно, значения $\alpha_{lq}(\rho)$ при любом фиксированном ρ, $0\leqslant\rho\leqslant 1$, можно выбрать в качестве новых весов агрегирования, поскольку выполняются соотношения (12).

Подставляя $\alpha_{lq}=\alpha_{lq}(\rho)$ в (7)—(10), получаем *однопараметрическую* задачу линейного программирования

$$(c(\rho),X)\to\min\ (=\psi^*(\rho)),$$

$$A(\rho)X\leqslant b,$$

$$B_l(\rho)X\leqslant\tilde{b}_l,\quad l=\overline{1,\nu},$$

$$X\geqslant 0,$$

где $c(\rho)=(c_1(\rho),\ c_2(\rho),\ \ldots,\ c_r(\rho))$, $c_q(\rho)=\sum\limits_{l=1}^{\nu}c_{lq}\alpha_{lq}(\rho)$; матрица $A(\rho)=(a_{iq}(\rho))$, $a_{iq}(\rho)=\sum\limits_{l=1}^{\nu}a_{iq}^{(l)}\alpha_{lq}(\rho)$, $i=\overline{1,m}$, $q=\overline{1,r}$; матрицы $B_l(\rho)=(b_{iq}^{(l)}(\rho))$, $b_{iq}^{(l)}(\rho)=b_{iq}^{(l)}\alpha_{lq}(\rho)$, $i=$ $=\overline{1,m_l}$, $l=\overline{1,\nu}$, $q=\overline{1,r}$.

Выберем в качестве новых весов агрегирования $\alpha^{(2)}=$ $=(\alpha_{lq}^{(2)})$ значения $\alpha_{lq}(\rho^*)$, где ρ^* — точка минимума функции $\psi^*(\rho)$ на интервале $[0,1]$.

Все приведенные рассуждения можно повторить для $\alpha^{(2)}$ и при необходимости перейти к $\alpha^{(3)}$ и т. д.

Описанный *процесс решения задачи (1)—(4)* схематически представлен на рис. 13.1. В результате этого процесса формируется последовательность значений $(x_{lq}^{(k)})$, $k=1,2,\ldots$, удовлетворяющих ограничениям (2)—(4). При достаточно общих условиях из данной по-

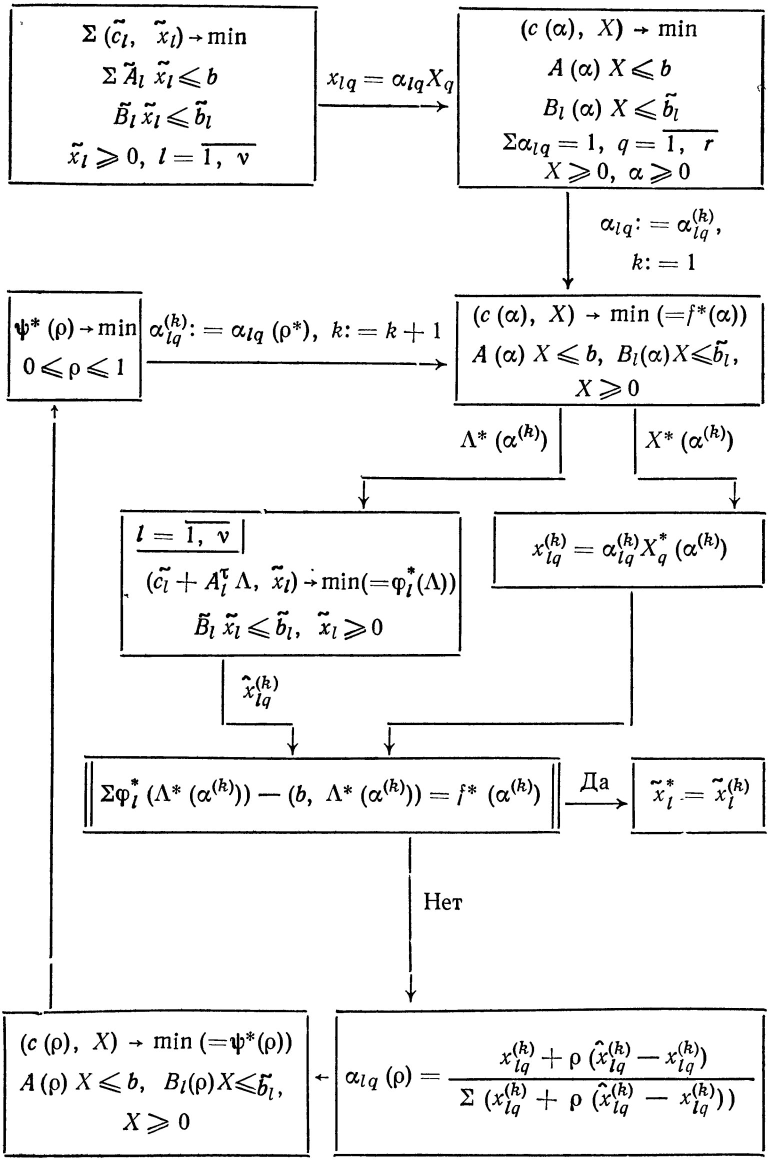

Рис. 13.1. Схема агрегирования переменных из разных блоков

следовательности можно выделить подпоследовательность, сходящуюся к решению исходной задачи (1)—(4).

13.3. Пример. Решим описанным методом задачу, рассмотренную в п. 5.6, § 5 (пример 2):

$$-x_1-3x_2-2x_3-x_4\to\min,$$
$$2x_1+x_2+x_3+x_4\leqslant 8,$$
$$x_1+x_2\leqslant 5,$$
$$x_3-x_4\leqslant 4,$$
$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0.$$

Полагая $x_1+x_3=X_1$, $x_2+x_4=X_2$, $x_1=\alpha_1X_1$, $x_3=\alpha_3X_1$, $x_2=\alpha_2X_2$, $x_4=\alpha_4X_2$, задачу можно переписать в агрегированном виде

$$-(\alpha_1+2\alpha_3)X_1-(3\alpha_2+\alpha_4)X_2\to\min,$$
$$(2\alpha_1+\alpha_3)X_1+(\alpha_2+\alpha_4)X_2\leqslant 8,$$
$$\alpha_1X_1+\alpha_2X_2\leqslant 5,$$
$$\alpha_3X_1-\alpha_4X_2\leqslant 4,$$
$$X_1\geqslant 0,\ X_2\geqslant 0,$$
$$\alpha_1+\alpha_3=1,\ \alpha_2+\alpha_4=1,$$
$$\alpha_1\geqslant 0,\ \alpha_2\geqslant 0,\ \alpha_3\geqslant 0,\ \alpha_4\geqslant 0$$

или, проводя очевидные преобразования:

$$-(1+\alpha_3)X_1-(1+2\alpha_2)X_2\to\min,$$
$$(1+\alpha_1)X_1+X_2\leqslant 8,$$
$$\alpha_1X_1+\alpha_2X_2\leqslant 5,$$
$$\alpha_3X_1-\alpha_4X_2\leqslant 4,$$
$$X_1\geqslant 0,\ X_2\geqslant 0,$$
$$\alpha_1+\alpha_3=1,\ \alpha_2+\alpha_4=1,$$
$$\alpha_1\geqslant 0,\ \alpha_2\geqslant 0,\ \alpha_3\geqslant 0,\ \alpha_4\geqslant 0.$$

При фиксированных α получаем локальную задачу (см. выражения (7)—(10))

$$-(1+\alpha_3)X_1-(1+2\alpha_2)X_2\to\min\ (=f^*(\alpha)),$$

$$(1+\alpha_1)X_1+X_2\leqslant 8,$$

$$\alpha_1 X_1+\alpha_2 X_2\leqslant 5,$$

$$\alpha_3 X_1-\alpha_4 X_2\leqslant 4,$$

$$X_1\geqslant 0,\quad X_2\geqslant 0.$$

Координирующая задача записывается в виде (см. выражения (11)—(12))

$$f^*(\alpha)\to\min,$$

$$\alpha_1+\alpha_3=1,\quad \alpha_2+\alpha_4=1,$$

$$\alpha_1\geqslant 0,\quad \alpha_2\geqslant 0,\quad \alpha_3\geqslant 0,\quad \alpha_4\geqslant 0.$$

Задача, двойственная к локальной, имеет вид (см. выражения (13)—(15))

$$-8\lambda-5\lambda_1-4\lambda_2\to\max\ (=f^*(\alpha)),$$

$$(1+\alpha_1)\lambda+\alpha_1\lambda_1+\alpha_3\lambda_2\geqslant 1+\alpha_3,$$

$$\lambda+\alpha_2\lambda_1-\alpha_4\lambda_2\geqslant 1+2\alpha_2,$$

$$\lambda\geqslant 0,\quad \lambda_1\geqslant 0,\quad \lambda_2\geqslant 0.$$

Используя связывающее ограничение исходной задачи, составим для нее функцию Лагранжа:

$$L(x,\lambda)=(-x_1-3x_2-2x_3-x_4)+\lambda(2x_1+x_2+x_3+x_4-8)=$$
$$=(-1+2\lambda)x_1+(-3+\lambda)x_2+(-2+\lambda)x_3+$$
$$+(-1+\lambda)x_4-8\lambda.$$

Минимизация этой функции по x при фиксированном λ и блочных ограничениях исходной задачи сводится к решению двух задач (см. (16)—(17)):

а) $(-1+2\lambda)x_1+(-3+\lambda)x_2\to\min\ (=\varphi_1^*(\lambda)),$

$$x_1+x_2\leqslant 5,$$

$$x_1\geqslant 0,\quad x_2\geqslant 0;$$

б) $(-2+\lambda)x_3+(-1+\lambda)x_4\to\min\ (=\varphi_2^*(\lambda)),$

$$x_3-x_4\leqslant 4,$$

$$x_3\geqslant 0,\quad x_4\geqslant 0.$$

Используя описанный итерационный процесс, на первом шаге положим для определенности $\alpha_1^{(1)}=1$, $\alpha_2^{(1)}=1$, $\alpha_3^{(1)}=0$, $\alpha_4^{(1)}=0$. При этих значениях α локальная задача запишется в виде

$$-X_1-3X_2\to\min,$$
$$2X_1+X_2\leqslant 8,$$
$$X_1+X_2\leqslant 5,$$
$$X_1\geqslant 0,\quad X_2\geqslant 0,$$

а двойственная к ней — в виде

$$-8\lambda-5\lambda_1\to\max,$$
$$2\lambda+\lambda_1\geqslant 1,$$
$$\lambda+\lambda_1\geqslant 3,$$
$$\lambda\geqslant 0,\quad \lambda_1\geqslant 0.$$

Решая эти задачи, получаем $X_1^{(1)}=0$, $X_2^{(1)}=5$, $\lambda^{(1)}=0$, $\lambda_1^{(1)}=3$. Отсюда $x_1^{(1)}=0$, $x_2^{(1)}=5$, $x_3^{(1)}=0$, $x_4^{(1)}=0$, $f^*(\alpha^{(1)})=$ $=-15$.

Решая задачи а) и б) при $\lambda=\lambda^{(1)}=0$, получаем $\hat{x}_1^{(1)}=$ $=0$, $\hat{x}_2^{(1)}=5$, $\hat{x}_3^{(1)}=W$, $\hat{x}_4^{(1)}=W-4$, $\varphi_1^*(\lambda^{(1)})=-15$, $\varphi_2^*(\lambda^{(1)})=-3W+4$, где $W\to\infty$. Значение $\varphi^*(\lambda^{(1)})=$ $=\varphi_1^*(\lambda^{(1)})+\varphi_2^*(\lambda^{(1)})-8\lambda^{(1)}=-3W-11\neq f^*(\alpha^{(1)})=-15$.

Сформируем новый набор значений α, полагая

$$\alpha_1(\rho)=\frac{x_1^{(1)}+\rho(\hat{x}_1^{(1)}-x_1^{(1)})}{[x_1^{(1)}+\rho(\hat{x}_1^{(1)}-x_1^{(1)})]+[x_3^{(1)}+\rho(\hat{x}_3^{(1)}-x_3^{(1)})]}=$$
$$=\frac{0}{\rho W}=0,$$

$$\alpha_2(\rho)=\frac{x_2^{(1)}+\rho(\hat{x}_2^{(1)}-x_2^{(1)})}{[x_2^{(1)}+\rho(\hat{x}_2^{(1)}-x_2^{(1)})]+[x_4^{(1)}+\rho(\hat{x}_4^{(1)}-x_4^{(1)})]}=$$
$$=\frac{5}{5+\rho(W-4)}\to 0\ \text{при}\ W\to\infty,$$

$$\alpha_3(\rho)=\frac{x_3^{(1)}+\rho(\hat{x}_3^{(1)}-x_3^{(1)})}{[x_1^{(1)}+\rho(\hat{x}_1^{(1)}-x_1^{(1)})]+[x_3^{(1)}+\rho(\hat{x}_3^{(1)}-x_3^{(1)})]}=$$

$$=\frac{\rho W}{\rho W}=1,$$

$$\alpha_4(\rho)=\frac{x_4^{(1)}+\rho(\hat{x}_4^{(1)}-x_4^{(1)})}{[x_2^{(1)}+\rho(\hat{x}_2^{(1)}-x_2^{(1)})]+[x_4^{(1)}+\rho(\hat{x}_4^{(1)}-x_4^{(1)})]}=$$

$$=\frac{\rho(W-4)}{5+\rho(W-4)}\to 1 \text{ при } W\to\infty.$$

Найденные значения $\alpha(\rho)$ не зависят от ρ, $0<\rho\leqslant 1$, и можно положить $\alpha_1^{(2)}=0$, $\alpha_2^{(2)}=0$, $\alpha_3^{(2)}=1$, $\alpha_4^{(2)}=1$.

На втором шаге при $\alpha_1=\alpha_1^{(2)}=0$, $\alpha_2=\alpha_2^{(2)}=0$, $\alpha_3=$ $=\alpha_3^{(2)}=1$ и $\alpha_4=\alpha_4^{(2)}=1$ решаем локальную и двойственную к ней задачи. Получаем $X_1^{(2)}=6$, $X_2^{(2)}=2$, $\lambda^{(2)}=1\frac{1}{2}$, $\lambda_2^{(2)}=\frac{1}{2}$. Отсюда $x_1^{(2)}=0$, $x_2^{(2)}=0$, $x_3^{(2)}=6$, $x_4^{(2)}=2$, $f^*(\alpha^{(2)})=-14$.

Решая задачи а) и б) при $\lambda=\lambda^{(2)}=1\frac{1}{2}$, определяем $\hat{x}_1^{(2)}=0$, $\hat{x}_2^{(2)}=5$, $\hat{x}_3^{(2)}=4$, $\hat{x}_4^{(2)}=0$, $\varphi_1^*(\lambda^{(2)})=-7\frac{1}{2}$, $\varphi_2^*(\lambda^{(2)})=-2$. Значение $\varphi^*(\lambda^{(2)})=\varphi_1^*(\lambda^{(2)})+\varphi_2^*(\lambda^{(2)})-$ $-8\lambda^{(2)}=-21\frac{1}{2}\neq f(\alpha^{(2)})=-14$.

Используя приведенные выше формулы, находим $\alpha_1(\rho)\equiv 0$, $\alpha_2(\rho)=\frac{5\rho}{3\rho+2}$, $\alpha_3(\rho)\equiv 1$, $\alpha_4(\rho)=\frac{2-2\rho}{3\rho+2}$. Подставляя эти выражения вместо α в запись исходной задачи в агрегированном виде, имеем параметрическую задачу

$$-2X_1-\frac{13\rho+2}{3\rho+2}X_2\to\min\ (=\psi^*(\rho)),$$

$$X_1+X_2\leqslant 8,$$

$$X_1+\frac{2\rho-2}{3\rho+2}X_2\leqslant 4,$$

$$X_1\geqslant 0,\ \frac{3\rho+2}{\rho}\geqslant X_2\geqslant 0.$$

При $0\leqslant\rho\leqslant 2/7$ решением этой задачи является $X_1^*(\rho)=\frac{-4\rho+24}{\rho+4}$, $X_2^*(\rho)=\frac{12\rho+8}{\rho+4}$ и $\psi^*(\rho)=$ $=\frac{-132\rho^2-256\rho-112}{(3\rho+2)(\rho+4)}$.

При $\frac{2}{7}<\rho\leqslant\frac{2}{5}$ значения $X_1^*(\rho)\equiv 0$, $X_2^*(\rho)\equiv 8$ и $\psi^*(\rho)=\frac{-104\rho-16}{3\rho+2}$.

Наконец, при $2/5<\rho\leqslant 1$ значения $X_1^*(\rho)=\frac{5\rho-2}{\rho}$, $X_2^*(\rho)=\frac{3\rho+2}{\rho}$ и $\psi^*(\rho)=\frac{-23\rho+2}{\rho}$.

Минимизируя $\psi^*(\rho)$ на интервале $[0, 1]$, получаем $\rho^*=1$. Следовательно, $\alpha_1^{(3)}=0$, $\alpha_2^{(3)}=1$, $\alpha_3^{(3)}=1$, $\alpha_4^{(3)}=0$.

На третьем шаге решаем локальную и двойственную к ней задачи при $\alpha_1=\alpha_1^{(3)}=0$, $\alpha_2=\alpha_2^{(3)}=1$, $\alpha_3=\alpha_3^{(3)}=1$, $\alpha_4=\alpha_4^{(3)}=0$. Получаем $X_1^{(3)}=3$, $X_2^{(3)}=5$, $\lambda^{(3)}=2$, $\lambda_1^{(3)}=1$, $\lambda_2^{(3)}=0$. Отсюда $x_1^{(3)}=0$, $x_2^{(3)}=5$, $x_3^{(3)}=3$, $x_4^{(3)}=0$, $f^*(\alpha^{(3)})=-21$.

Решая задачи а) и б) при $\lambda=\lambda^{(3)}=2$, находим $\hat{x}_1^{(3)}=$ $=0$, $\hat{x}_2^{(3)}=5$, $\hat{x}_3^{(3)}=0$, $\hat{x}_4^{(3)}=0$, $\varphi_1^*(\lambda^{(3)})=-5$, $\varphi_2^*(\lambda^{(3)})=0$.

Поскольку $\varphi^*(\lambda^{(3)})=\varphi_1^*(\lambda^{(3)})+\varphi_2^*(\lambda^{(3)})-8\lambda^{(3)}=-21=$ $=f^*(\alpha^{(3)})$, то $x_1^*=x_1^{(3)}=0$, $x_2^*=x_2^{(3)}=5$, $x_3^*=x_3^{(3)}=3$, $x_4^*=$ $=x_4^{(3)}=0$ — решение исходной задачи.

14. ПОБЛОЧНОЕ АГРЕГИРОВАНИЕ ПЕРЕМЕННЫХ

Широкое распространение на практике получило агрегирование переменных, принадлежащих одному и тому же блоку.

14.1. Рассмотрим *задачу линейного программирования* следующего вида:

$$\sum_{l=1}^{\nu} (\tilde{c}_l, \tilde{x}_l) + (\tilde{d}_0, \tilde{y}_0) + \sum_{l=1}^{\nu} (\tilde{d}_l, \tilde{y}_l) \to \min, \tag{1}$$

$$\sum_{l=1}^{\nu} \tilde{A}_l \tilde{x}_l + \tilde{D}_0 \tilde{y}_0 = b, \tag{2}$$

$$\tilde{B}_l \tilde{x}_l + \tilde{D}_l \tilde{y}_l = \tilde{b}_l, \quad l = \overline{1, \nu}, \tag{3}$$

$$\tilde{x}_l \geqslant 0, \ \tilde{y}_l \geqslant 0, \ l = \overline{1, \nu}, \ \tilde{y}_0 \geqslant 0, \tag{4}$$

где $\tilde{x}_l$, $\tilde{y}_l$, $l = \overline{1, \nu}$, $\tilde{y}_0$ — искомые векторы; $\tilde{c}_l$, $\tilde{d}_l$, $\tilde{b}_l$, $l = \overline{1, \nu}$, $\tilde{d}_0$, b — векторы; $\tilde{A}_l$, $\tilde{B}_l$, $\tilde{D}_l$, $l = \overline{1, \nu}$, $\tilde{D}_0$ — матрицы соответствующих размерностей.

В дальнейшем будем предполагать, что множества точек, описываемые соотношениями (3)—(4), являются ограниченными, т. е. многогранниками.

Проведем *поблочное* агрегирование x. Пусть $x = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_\nu)$, $\tilde{x}_l = (x_{l1}, x_{l2}, \ldots, x_{lr_l})$. Введем новые (*агрегированные*) переменные $X_1, X_2, \ldots, X_\nu$ и неотрицательные *веса агрегирования* α_{lq}, полагая

$$x_{lq} = \alpha_{lq} X_l, \quad \alpha_{lq} \geqslant 0, \quad q = \overline{1, r_l}, \quad l = \overline{1, \nu}. \tag{5}$$

Тогда задачу (1)—(4) можно переписать в виде

$$\sum_{l=1}^{\nu} c(\tilde{\alpha}_l) X_l + (\tilde{d}_0, \tilde{y}_0) + \sum_{l=1}^{\nu} (\tilde{d}_l, \tilde{y}_l) \to \min, \tag{6}$$

$$\sum_{l=1}^{\nu} a(\tilde{\alpha}_l) X_l + \tilde{D}_0 \tilde{y}_0 = b, \tag{7}$$

$$b(\tilde{\alpha}_l) X_l + \tilde{D}_l \tilde{y}_l = \tilde{b}_l, \quad l = \overline{1, \nu}, \tag{8}$$

$$X_l \geqslant 0, \ \tilde{\alpha}_l \geqslant 0, \ \tilde{y}_l \geqslant 0, \ l = \overline{1, \nu}, \ \tilde{y}_0 \geqslant 0, \tag{9}$$

где $\tilde{\alpha}_l=(\alpha_{l1}, \alpha_{l2}, \ldots, \alpha_{lr_l})$; числа $c(\tilde{\alpha}_l)=(\tilde{c}_l, \tilde{\alpha}_l)$; векторы $a(\tilde{\alpha}_l)=\tilde{A}_l\tilde{\alpha}_l$, $b(\tilde{\alpha}_l)=\tilde{B}_l\tilde{\alpha}_l$, $l=\overline{1, \nu}$.

Рассматриваемая задача является *многопараметрической* задачей линейного программирования. Фиксируя параметры — веса агрегирования α_{l_q}, получаем задачу линейного программирования, содержащую столько же ограничений и неизвестных y, сколько их содержит исходная задача (1)—(4). Однако вместо $\sum\limits_{l=1}^{\nu} r_l$ неизвестных x в исходной задаче в задаче (5)—(8) содержится только ν новых неизвестных X_l.

Соотношения (5) позволяют вычислить (допустимые) значения исходных x по найденным значениям X_l при заданных весах агрегирования α_{lq}. Обозначим оптимальное значение целевой функции (6) при фиксированных α_{lq} через $f^*(\alpha)$.

Таким образом, решение исходной задачи может быть получено в результате нахождения оптимальных весов агрегирования $\alpha^*_{l_q}$ (доставляющих минимум функции $f^*(\alpha)$ при ограничениях $\alpha_{lq}\geqslant 0$) и решения задачи (6)—(9) при $\alpha_{lq}=\alpha^*_{lq}$.

14.2. Опишем одну из возможных *итерационных схем нахождения оптимальных весов агрегирования* α^*_{lq}.

Сформулируем задачу (1)—(4) как задачу нахождения седловой точки функции Лагранжа

$$L(x, y, \Lambda)=\sum_{l=1}^{\nu}(\tilde{c}_l, \tilde{x}_l)+(\tilde{d}_0, \tilde{y}_0)+\sum_{l=1}^{\nu}(\tilde{d}_l, \tilde{y}_l)+$$

$$+\left(\Lambda, \sum_{l=1}^{\nu}\tilde{A}_l\tilde{x}_l+\tilde{D}_0\tilde{y}_0-b\right)=\sum_{l=1}^{\nu}(\tilde{c}_l+\tilde{A}^{\tau}_l\Lambda, \tilde{x}_l)+$$

$$+(\tilde{d}_0+\tilde{D}^{\tau}_0\Lambda, \tilde{y}_0)+\sum_{l=1}^{\nu}(\tilde{d}_l, \tilde{y}_l)-(b, \Lambda)$$

при ограничениях (3)—(4). Здесь $y=(\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_\nu)$; Λ — вектор двойственных переменных; τ — знак транспонирования.

При фиксированном векторе Λ задача минимизации функции $L(x, y, \Lambda)$ по x и y, удовлетворяющим ограничениям (3)—(4), распадается на ν подзадач вида $(l=\overline{1, \nu})$:

$$\theta_l(\tilde{x}_l, \tilde{y}_l) = (\tilde{c}_l + \tilde{A}_l^{\tau}\Lambda, \tilde{x}_l) + (\tilde{d}_l, \tilde{y}_l) \to \min, \qquad (10)$$

$$\tilde{B}_l\tilde{x}_l + \tilde{D}_l\tilde{y}_l = \tilde{b}_l, \qquad (11)$$

$$\tilde{x}_l \geqslant 0, \quad \tilde{y}_l \geqslant 0 \qquad (12)$$

и подзадачу

$$(\tilde{d}_0 + \tilde{D}_0^{\tau}\Lambda, \tilde{y}_0) \to \min,$$

$$\tilde{y}_0 \geqslant 0.$$

Обозначим решение задачи (10)—(12) через $\tilde{x}_l^*(\Lambda)$ и $\tilde{y}_l^*(\Lambda)$.

При фиксированном векторе x исходная задача (1)—(4) распадается на ν подзадач вида $(l=\overline{1, \nu})$:

$$(\tilde{d}_l, \tilde{y}_l) \to \min,$$

$$-\tilde{D}_l\tilde{y}_l = \tilde{B}_l\tilde{x}_l - \tilde{b}_l,$$

$$\tilde{y}_l \geqslant 0$$

и подзадачу

$$(\tilde{d}_0, \tilde{y}_0) \to \min,$$

$$-\tilde{D}_0\tilde{y}_0 = \sum_{l=1}^{\nu} \tilde{A}_l\tilde{x}_l - b,$$

$$\tilde{y}_0 \geqslant 0.$$

Обозначим через $y^*(x)$ решение этих подзадач, а через $\Lambda^*(x)$ оптимальный вектор двойственных переменных последней подзадачи.

Иными словами, $\Lambda^*(x)$ является решением задачи

$$\left(\sum_{l=1}^{\nu} \tilde{A}_l\tilde{x}_l - b, \Lambda\right) \to \max, \qquad (13)$$

$$\tilde{D}_0^{\tau}\Lambda \geqslant -\tilde{d}_0 \qquad (14)$$

при фиксированном x.

Процесс поиска оптимальных весов агрегирования α_{lq}^* состоит в следующем.

На первом шаге задаются произвольные неотрицательные веса $\alpha_{lq}^{(1)}$ и решается агрегированная задача (6)—(9). Найденные значения $X_1^{(1)}, X_2^{(1)}, \ldots, X_\nu^{(1)}$ позволяют вычислить значения $x_{lq}^{(1)} = \alpha_{lq}^{(1)} X_l^{(1)}$, $q=\overline{1, r_l}$, $l=\overline{1, \nu}$. Полученное значение y обозначим через $y^{(1)}$.

Решая задачу (13)—(14) при $x=x^{(1)}$, имеем $\Lambda^*(x^{(1)})$. Решая задачи (10)—(12) при $\Lambda=\Lambda^*(x^{(1)})$, находим векторы $\hat{x}_l^{(1)}=\tilde{x}_l^*(\Lambda^*(x^{(1)})$ и $\hat{y}_l^{(1)}=\tilde{y}_l^*(\Lambda^*(x^{(1)}))$, $i=\overline{1, \nu}$.

Очевидно, что $\theta_l(\hat{x}_l^{(1)}, \hat{y}_l^{(1)})\leqslant\theta_l(\tilde{x}_l^{(1)}, \tilde{y}_l^{(1)})$, $l=\overline{1, \nu}$. Выберем те индексы $l\in\{1, 2, \ldots, \nu\}$, при которых в указанном соотношении имеет место знак строгого неравенства. Обозначим множество всех таких индексов через $L^{(1)}$.

Если $L^{(1)}=\varnothing$, то $\alpha_{lq}^*=\alpha_{lq}^{(1)}$ и $x^{(1)}$, $y^{(1)}$ — решение исходной задачи.

Если $L^{(1)}\neq\varnothing$, то выберем новые веса агрегирования $\alpha_{lq}^{(2)}$, полагая $\alpha_{lq}^{(2)}=\rho x_{lq}^{(1)}+(1-\rho)\hat{x}_{lq}^{(1)}$ для всех $l\in L^{(1)}$ и $\alpha_{lq}^{(2)}=x_{lq}^{(1)}$ для остальных l. Здесь ρ — число, удовлетворяющее неравенству $0\leqslant\rho\leqslant 1$.

Для нахождения наилучшего значения $\rho=\rho^*$ подставим выражения для $\alpha_{lq}^{(2)}$ в агрегированную задачу (6)—(9). В результате получаем *однопараметрическую* (относительно ρ) задачу линейного программирования

$$\sum_{l=1}^{\nu} c_l(\rho)X_l+(\tilde{d}_0, \tilde{y}_0)+\sum_{l=1}^{\nu}(\tilde{d}_l, \tilde{y}_l)\to\min,$$

$$\sum_{l=1}^{\nu} a_l(\rho)X_l+\tilde{D}_0\tilde{y}_0=b,$$

$$b_l(\rho)X_l+\tilde{D}_l\tilde{y}_l=\tilde{b}_l,\quad l=\overline{1, \nu},$$

$$X_l\geqslant 0,\quad \tilde{y}_l\geqslant 0,\quad l=\overline{1, \nu},\quad \tilde{y}_0\geqslant 0,\quad 0\leqslant\rho\leqslant 1,$$

решение которой и дает искомое значение ρ^*.

Процесс повторяется для вновь найденных весов агрегирования $\alpha_{lq}^{(2)}$ (при $\rho=\rho^*$) и т. д., пока на некотором шаге $k=\bar{k}$ не будет получено $L^{(\bar{k})}=\varnothing$. Значения $\alpha_{lq}^*=$ $=\alpha_{lq}^{(\bar{k})}$.

При определенных условиях данный процесс сходится. Его схема представлена на рис. 14.1.

14.3. Пример. Решим описанным методом следующую задачу линейного программирования:

$$\begin{gathered}\Sigma(\tilde c_l,\ \tilde x_l)+(\tilde d_0,\ \tilde y_0)+\\+\Sigma(\tilde d_l,\ \tilde y_l)\to\min\\ \Sigma\tilde A_l\tilde x_l+\tilde D_0\tilde y_0=b\\ \tilde B_l\tilde x_l+\tilde D_l\tilde y_l=\tilde b_l\\ \tilde x_l\geqslant 0,\ \tilde y_l\geqslant 0,\ l=\overline{1,\ \nu},\ \tilde y_0\geqslant 0\end{gathered}$$

$$\downarrow\ \alpha_{lq}:=\alpha_{lq}^{(k)}\geqslant 0,\quad k:=1$$

$$\begin{gathered}\Sigma\, c(\tilde\alpha_l)\,X_l+(\tilde d_0,\ \tilde y_0)+\\+\Sigma(\tilde d_l,\ \tilde y_l)\to\min\\ \Sigma\, a(\tilde\alpha_l)\,X_l+\tilde D_0\tilde y_0=b\\ b(\tilde\alpha_l)\,X_l+\tilde D_l\tilde y_l=\tilde b_l\\ X_l\geqslant 0,\ \ \tilde y_l\geqslant 0,\ l=\overline{1,\ \nu},\ \tilde y_0\geqslant 0\end{gathered}$$

$$\downarrow\ X_l^{(k)},\ \tilde y_l^{(k)},\ \tilde y_0^{(k)}$$

$$x_{lq}^{(k)}=\alpha_{lq}^{(k)}X_l^{(k)}$$

$$\downarrow$$

$$\begin{gathered}(\Sigma\tilde A_l\tilde x_l-b,\ \Lambda)\to\max\\ \tilde D_0^{\tau}\Lambda\geqslant-\tilde d_0\end{gathered}$$

$$\downarrow\ \Lambda^*(x^{(k)})$$

$$\begin{gathered}\underline{l=\overline{1,\ \nu}}\\ \theta_l(\tilde x_l,\ \tilde y_l)=(\tilde c_l+\tilde A_l^{\tau}\Lambda,\ \tilde x_l)+\\+(\tilde d_l,\ \tilde y_l)\to\min\\ \tilde B_l\tilde x_l+\tilde D_l\tilde y_l=\tilde b_l\\ \tilde x_l\geqslant 0,\ \tilde y_l\geqslant 0\end{gathered}$$

$$\downarrow\ \hat x_{lq}^{(k)}$$

$$\uparrow$$

$$\alpha_{lq}(\rho)=\begin{cases}\rho x_{lq}^{(k)}+(1-\rho)\hat x_{lq}^{(k)}, & l\in L^{(k)}\\ x_{lq}^{(k)}, & l\notin L^{(k)}\end{cases}$$

$$\uparrow$$

$$\begin{gathered}\Sigma\, c_l(\rho)\,X_l+(\tilde d_0,\ \tilde y_0)+\\+\Sigma(\tilde d_l,\ \tilde y_l)\to\min(=\psi^*(\rho))\\ \Sigma\, a_l(\rho)\,X_l+\tilde D_0\tilde y_0=b\\ b_l(\rho)\,X_l+\tilde D_l\tilde y_l=\tilde b_l\\ X_l\geqslant 0,\ \ \tilde y_l\geqslant 0,\\ l=\overline{1,\ \nu},\ \tilde y_0\geqslant 0\end{gathered}$$

$$\uparrow$$

$$\psi^*(\rho)\to\min,\ 0\leqslant\rho\leqslant 1$$

$$\uparrow\ \rho^*$$

$$\alpha_{lq}^{(k)}:=\alpha_{lq}(\rho^*)$$

$$\xleftarrow{k:=k+1}$$

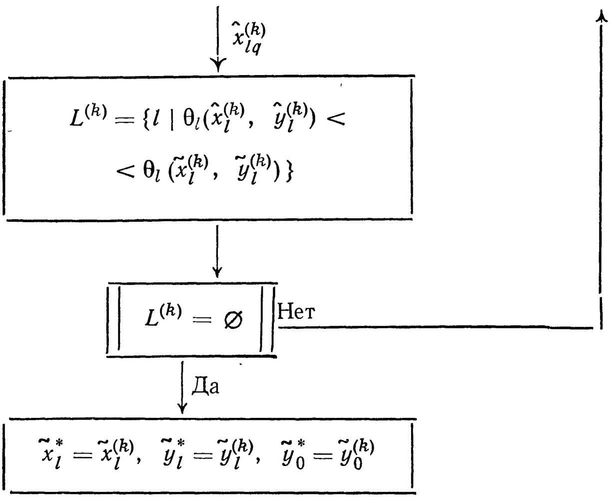

Рис. 14.1. Схема поблочного агрегирования переменных

$$
\begin{aligned}
&-2x_1-x_2-x_3-3x_4-2x_5-y_1 \to \min,\\
&x_1+x_2+2x_3+x_4+x_5+y_1=6,\\
&2x_1+x_2 \leqslant 2,\\
&x_3+x_4+3x_5 \leqslant 4,\\
&x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0,\ x_5 \geqslant 0,\ y_1 \geqslant 0.
\end{aligned}
$$

Запишем эту задачу в каноническом виде

$$
\begin{aligned}
&-2x_1-x_2-x_3-3x_4-2x_5-y_1+0y_2+0y_3 \to \min,\\
&x_1+x_2+2x_3+x_4+x_5+y_1=6,\\
&2x_1+x_2+y_2=2,\\
&x_3+x_4+3x_5+y_3=4,\\
&x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0,\ x_5 \geqslant 0,\ y_1 \geqslant 0,\ y_2 \geqslant 0,\ y_3 \geqslant 0.
\end{aligned}
$$

В обозначениях этого параграфа $\tilde{x}_1=(x_1, x_2)$, $\tilde{x}_2= =(x_3, x_4, x_5)$, $\tilde{y}_0=y_1$, $\tilde{y}_1=y_2$, $\tilde{y}_2=y_3$.

Введем агрегированные переменные X_1, X_2, неотрицательные веса агрегирования α_1, α_2, α_3, α_4, α_5, так, что $x_1=\alpha_1 X_1$, $x_2=\alpha_2 X_1$, $x_3=\alpha_3 X_2$, $x_4=\alpha_4 X_2$, $x_5=\alpha_5 X_2$.

Подставляя приведенные выражения для x в целевую функцию и ограничения, получаем агрегированную задачу (см. выражения (6)—(9))

$$
\begin{aligned}
&(-2\alpha_1-\alpha_2)X_1+(-\alpha_3-3\alpha_4-2\alpha_5)X_2-y_1+0y_2+0y_3\to\min,\\
&(\alpha_1+\alpha_2)X_1+(2\alpha_3+\alpha_4+\alpha_5)X_2+y_1=6,\\
&(2\alpha_1+\alpha_2)X_1+y_2=2,\\
&(\alpha_3+\alpha_4+3\alpha_5)X_2+y_3=4,\\
&X_1\geqslant 0,\ X_2\geqslant 0,\ y_1\geqslant 0,\ y_2\geqslant 0,\ y_3\geqslant 0,\\
&\alpha_1\geqslant 0,\ \alpha_2\geqslant 0,\ \alpha_3\geqslant 0,\ \alpha_4\geqslant 0,\ \alpha_5\geqslant 0.
\end{aligned}
$$

Локальные задачи (10)—(12) в рассматриваемом случае записываются в виде

$$
\begin{gathered}
(-2+\lambda)x_1+(-1+\lambda)x_2\to\min,\\
2x_1+x_2\leqslant 2,\\
x_1\geqslant 0,\ x_2\geqslant 0
\end{gathered}
$$

и

$$
\begin{gathered}
(-1+2\lambda)x_3+(-3+\lambda)x_4+(-2+\lambda)x_5\to\min,\\
x_3+x_4+3x_5\leqslant 4,\\
x_3\geqslant 0,\ x_4\geqslant 0,\ x_5\geqslant 0,
\end{gathered}
$$

а двойственная задача (13)—(14) в виде

$$
\begin{gathered}
(x_1+x_2+2x_3+x_4+x_5-6)\lambda\to\max,\\
\lambda\geqslant 1.
\end{gathered}
$$

На первом шаге выбираем произвольные неотрицательные веса агрегирования, полагая, например, $\alpha_1^{(1)}=0$, $\alpha_2^{(1)}=1$, $\alpha_3^{(1)}=1$, $\alpha_4^{(1)}=0$, $\alpha_5^{(1)}=1$, и решаем агрегированную задачу

$$
\begin{aligned}
&-X_1-3X_2-y_1+0y_2+0y_3\to\min,\\
&X_1+3X_2+y_1=6,\\
&X_1+y_2=2,\\
&4X_2+y_3=4,\\
&X_1\geqslant 0,\ X_2\geqslant 0,\ y_1\geqslant 0,\ y_2\geqslant 0,\ y_3\geqslant 0.
\end{aligned}
$$

Получаем $X_1^{(1)}=2$, $X_2^{(1)}=1$, $y_1^{(1)}=1$, $y_2^{(1)}=0$, $y_3^{(1)}=0$. Отсюда $x_1^{(1)}=0$, $x_2^{(1)}=2$, $x_3^{(1)}=1$, $x_4^{(1)}=0$, $x_5^{(1)}=1$.

Двойственная задача (13)—(14) при найденных значениях x имеет вид $\lambda\to\min$, $\lambda\geqslant 1$. Ее решением является $\lambda=1$.

Локальные задачи (10)—(12) при $\lambda=1$ запишем в виде

$$-x_1+0x_2\to\min,$$

$$2x_1+x_2\leqslant 2,$$

$$x_1\geqslant 0,\ x_2\geqslant 0$$

и

$$x_3-2x_4-x_5\to\min,$$

$$x_3+x_4+3x_5\leqslant 4,$$

$$x_3\geqslant 0,\ x_4\geqslant 0,\ x_5\geqslant 0.$$

Решая рассматриваемые задачи, получаем $\hat{x}_1^{(1)}=1$, $\hat{x}_2^{(1)}=0$, $\hat{x}_3^{(1)}=0$, $\hat{x}_4^{(1)}=4$, $\hat{x}_5^{(1)}=0$. Оптимальные значения целевых функций равны -1 и -8. При $x=x^{(1)}$ их значения равны соответственно 0 и 0.

Следовательно, новые веса

$$\alpha_1^{(2)}=\rho x_1^{(1)}+(1-\rho)\hat{x}_1^{(1)}=\rho\cdot 0+(1-\rho)\cdot 1=1-\rho,$$

$$\alpha_2^{(2)}=\rho x_2^{(1)}+(1-\rho)\hat{x}_2^{(1)}=\rho\cdot 2+(1-\rho)\cdot 0=2\rho,$$

$$\alpha_3^{(2)}=\rho x_3^{(1)}+(1-\rho)\hat{x}_3^{(1)}=\rho\cdot 1+(1-\rho)\cdot 0=\rho,$$

$$\alpha_4^{(2)}=\rho x_4^{(1)}+(1-\rho)\hat{x}_4^{(1)}=\rho\cdot 0+(1-\rho)\cdot 4=4-4\rho,$$

$$\alpha_5^{(2)}=\rho x_5^{(1)}+(1-\rho)\hat{x}_5^{(1)}=\rho\cdot 1+(1-\rho)\cdot 0=\rho.$$

Подставляя эти выражения в агрегированную задачу, получаем однопараметрическую задачу вида

$$\begin{aligned}&-2X_1+(9\rho-12)X_2-y_1\to\min,\\&(1+\rho)X_1+(4-\rho)X_2+y_1=6,\\&X_1\leqslant 1,\\&X_2\leqslant 1,\\&X_1\geqslant 0,\ X_2\geqslant 0,\ y_1\geqslant 0,\ 0\leqslant\rho\leqslant 1.\end{aligned}$$

Нетрудно видеть, что при $0 \leqslant \rho \leqslant 1$ решением данной задачи являются $X_1^*(\rho) \equiv 1$, $X_2^*(\rho) \equiv 1$, $y_1^*(\rho) \equiv 1$. Оптимальное значение целевой функции равно $9\rho - 15$ и достигает наименьшего значения при $\rho = \rho^* = 0$.

На втором шаге веса агрегирования $\alpha_1^{(2)} = 1$, $\alpha_2^{(2)} = 0$, $\alpha_3^{(2)} = 0$, $\alpha_4^{(2)} = 4$, $\alpha_5^{(2)} = 0$.

Агрегированная задача при этих весах имеет вид

$$-2X_1 - 12X_2 - y_1 + 0y_2 + 0y_3 \to \min,$$

$$\begin{aligned} X_1 + 4X_2 + y_1 \qquad\qquad &= 6, \\ 2X_1 \qquad\qquad + y_2 \qquad &= 2, \\ 4X_2 \qquad\qquad + y_3 &= 4, \end{aligned}$$

$$X_1 \geqslant 0,\ X_2 \geqslant 0,\ y_1 \geqslant 0,\ y_2 \geqslant 0,\ y_3 \geqslant 0.$$

Ее решение: $X_1^{(2)} = 1$, $X_2^{(2)} = 1$, $y_1^{(2)} = 1$, $y_2^{(2)} = 0$, $y_3^{(2)} = 0$. Отсюда $x_1^{(2)} = 1$, $x_2^{(2)} = 0$, $x_3^{(2)} = 0$, $x_4^{(2)} = 4$, $x_5^{(2)} = 0$.

Двойственная задача (13)—(14) при найденных значениях x и локальные задачи (10)—(12) являются такими же, что и на предыдущем шаге. Следовательно, $\hat{x}_1^{(2)} = 1$, $\hat{x}_2^{(2)} = 0$, $\hat{x}_3^{(2)} = 0$, $\hat{x}_4^{(2)} = 4$, $\hat{x}_5^{(2)} = 0$. Оптимальные значения целевых функций локальных задач равны -1 и -8. При $x = x^{(2)}$ их значения также равны -1 и -8 соответственно.

Таким образом, $x_1^* = x_1^{(2)} = 1$, $x_2^* = x_2^{(2)} = 0$, $x_3^* = x_3^{(2)} = 0$, $x_4^* = x_4^{(2)} = 4$, $x_5^* = x_5^{(2)} = 0$, $y_1^* = y_1^{(2)} = 0$ — решение исходной задачи.

15. МЕТОД ВОЗМУЩЕНИЙ, ДЕКОМПОЗИЦИЯ И АГРЕГИРОВАНИЕ

В настоящем параграфе описывается своеобразный подход к решению вопросов декомпозиции и агрегирования в задачах математического программирования, в основе которого лежат некоторые понятия и результаты теории возмущений.

15.1. Пусть дана *задача математического программирования*

$$f(x)\to\min, \tag{1}$$

$$g_i(x)\leqslant 0,\ i=\overline{1,m}. \tag{2}$$

Рассмотрим *однопараметрическое* семейство задач вида

$$f(x)+\varepsilon\hat{f}(x)\to\min, \tag{3}$$

$$g_i(x)+\varepsilon\hat{g}_i(x)\leqslant 0,\ i=\overline{1,m}. \tag{4}$$

Задачу (1)—(2) обычно называют *порождающей*, а задачу (3)—(4) — *возмущенной*. Функции $\hat{f}(x)$, $\hat{g}_i(x)$, $i=\overline{1,m}$, показывают характер возмущения, а параметр ε — его интенсивность.

Предположим, что 1) необходимо решить задачу вида (3)—(4) при некотором конкретном значении $\varepsilon=\varepsilon_0$, 2) непосредственное решение этой задачи вызывает затруднение и 3) решение порождающей задачи (1)—(2) может быть получено сравнительно просто.

Если значение параметра $\varepsilon=\varepsilon_0$ невелико, то вполне естественной является тактика, согласно которой сначала находится решение порождающей задачи, а затем при необходимости вносится поправка, определяемая конкретным характером и интенсивностью возмущения. Изучение возможностей и способов реализации такой тактики привело к созданию так называемого *метода возмущений.*

Для более детального описания этого метода ограничимся рассмотрением *задачи линейного программирования.*

Порождающая задача (1)—(2) имеет вид

$$(c, x)\to\min(=f^*), \tag{5}$$

$$Ax=b, \tag{6}$$

$$x\geqslant 0, \tag{7}$$

а возмущенная (3)—(4)

$$(c+\varepsilon\hat{c}, x)\to\min(=f^*(\varepsilon)), \tag{8}$$

$$(A+\varepsilon\hat{A})x=b+\varepsilon\hat{b}, \tag{9}$$

$$x\geqslant 0, \tag{10}$$

где $x=(x_1, x_2, \ldots, x_n)\in R^n$; A и $\hat{A}$ — матрицы размерности $m\times n$, $(m<n)$; c, $\hat{c}$, b, $\hat{b}$ — векторы соответствующих размерностей; $\varepsilon>0$ — параметр.

Обозначим через f^* и $f^*(\varepsilon)$ — оптимальные значения целевых функций порождающей и возмущенной задач, а через X^* и $X^*(\varepsilon)$ соответствующие множества их решений. Предполагается, что X^* и $X^*(\varepsilon)$ — непустые ограниченные множества.

В методе возмущений особую роль играет *вспомогательная* задача

$$(\hat{c}-\hat{A}^\tau\Lambda^*, x)\to\min(=\hat{f}^*), \tag{11}$$

$$x\in X^*, \tag{12}$$

где Λ^* — решение задачи, двойственной к порождающей:

$$(b, \Lambda)\to\max,$$

$$A^\tau\Lambda\leqslant c;$$

τ — знак транспонирования.

Предполагается, что последняя задача имеет единственное решение. Обозначим через $\hat{f}^*$ оптимальное значение целевой функции вспомогательной задачи, а через $\hat{X}^*$ множество ее решений.

Между решениями возмущенной и вспомогательной задач существует весьма тесная взаимосвязь. Так, можно показать, что при невырожденности порождающей задачи (5)—(7) множество предельных точек $X^*(\varepsilon)$, получаемых при $\varepsilon\to 0$, содержится в $\hat{X}^*$ и имеет место соотношение

$$f^*(\varepsilon)=f^*+\varepsilon(\hat{f}^*+(\hat{b}, \Lambda^*))+0(\varepsilon).$$

Пусть $\hat{x}^*\in\hat{X}^*$ — произвольное базисное решение (опорный план) вспомогательной задачи (11)—(12), B и $\hat{B}$ — соответствующие ему матрицы базиса (подматрицы матриц A и $\hat{A}$). Введем в рассмотрение вектор $x(\varepsilon)$, у которого те же нулевые компоненты, что и у вектора $\hat{x}^*$, а остальные (базисные) компоненты равны $(B+\varepsilon\hat{B})^{-1}\times$ $\times(b+\varepsilon\hat{b})$. Можно показать, что при всех достаточно малых ε вектор $x(\varepsilon)$ является допустимым для возмущенной задачи (8)—(10) и значение целевой функции (8) при $x=x(\varepsilon)$ отличается от оптимального $f^*(\varepsilon)$ не более чем на $0(\varepsilon^2)$. Если $\hat{x}^*$ — единственное решение вспомогательной задачи (т. е. $\hat{X}^*=\{\hat{x}^*\}$), то при всех достаточно малых ε решение $x^*(\varepsilon)$ возмущенной задачи также единственно (т. е. $X^*(\varepsilon)=\{x^*(\varepsilon)\}$) и $x^*(\varepsilon)=x(\varepsilon)$.

Эти утверждения позволяют получать точное или приближенное решение возмущенной задачи (8)—(10) при заданном (достаточно малом) значении $\varepsilon=\varepsilon_0$ посредством решения порождающей (5)—(7) и вспомогательной (11)—(12) задач. Сначала идентифицируется множество X^* решений порождающей задачи. Затем находится базисное решение $\hat{x}^*$ вспомогательной задачи и фиксируются индексы, соответствующие базисным переменным. Определяется базисное решение $x(\varepsilon_0)$ исходной возмущенной задачи. Индексы базисных переменных вектора $x(\varepsilon_0)$ совпадают с индексами базисных переменных вектора $\hat{x}^*$, значения которых вычисляются указанным выше способом (полагаются равными $(B+\varepsilon\hat{B})^{-1}(b+\varepsilon\hat{b})$). Если ε_0 достаточно мало, то вектор $x(\varepsilon_0)$ удовлетворяет ограничениям (9)—(10) возмущенной задачи при $\varepsilon=\varepsilon_0$ и может быть выбран в качестве приближенного решения интересующей нас возмущенной задачи. Если решение вспомогательной задачи единственно, то $x(\varepsilon_0)$ — точное решение возмущенной задачи. Если $x(\varepsilon_0)$ не удовлетворяет условиям оптимальности, то его можно использовать для дальнейших вычислений, применяя, например, для решения возмущенной задачи симплекс-метод.

Отметим, что все утверждения справедливы, если в ограничениях исходной задачи знаки равенства заменить на неравенства.

15.2. Пример. Пусть порождающая задача (5)—(7) имеет вид

$$0x_1+0x_2-x_3\to\min,$$

$$x_1-x_2+0x_3=1,$$

$$0x_1+0x_2+x_3=2,$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,$$

а возмущенная (8)—(10)

$$\varepsilon x_1+\varepsilon x_2-x_3\to\min,$$

$$x_1-x_2+\varepsilon x_3=1+\varepsilon,$$

$$\varepsilon x_1+\varepsilon x_2+x_3=2+\varepsilon,$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0.$$

В обозначениях этого параграфа

$$A=\begin{pmatrix}1 & -1 & 0\\ 0 & 0 & 1\end{pmatrix},\ \hat{A}=\begin{pmatrix}0 & 0 & 1\\ 1 & 1 & 0\end{pmatrix},\ b=\begin{pmatrix}1\\ 2\end{pmatrix}\ \hat{b}=\begin{pmatrix}1\\ 1\end{pmatrix},$$

$c=(0,\ 0,\ -1)$, $\hat{c}=(1,\ 1,\ 0)$.

Необходимо решить возмущенную задачу при некотором достаточно малом $\varepsilon=\varepsilon_0$, скажем, при $\varepsilon_0=0{,}1$.

Множество X^* решений порождающей задачи описывается, очевидно, соотношениями $x_1-x_2=1$, $x_3=2$, $x_1\geqslant 0$, $x_2\geqslant 0$. Значение $f^*=-2$.

Двойственная к порождающей задача записывается в виде

$$\lambda_1+2\lambda_2\to\max,$$

$$\lambda_1+0\lambda_2\leqslant 0,$$

$$-\lambda_1+0\lambda_2\leqslant 0,$$

$$0\lambda_1+\lambda_2\leqslant -1.$$

Отсюда $\Lambda^*=(\lambda_1^*,\ \lambda_2^*)=(0,\ -1)$ и вспомогательная задача (11)—(12) имеет вид

$$2x_1+2x_2+0x_3\to\min,$$

$$x_1-x_2=1,\ x_3=2,$$

$$x_1\geqslant 0,\ x_2\geqslant 0.$$

Эта задача имеет единственное решение $\hat{x}^*=(\hat{x}_1^*,\ \hat{x}_2^*,\ \hat{x}_3^*)=(1,\ 0,\ 2)$. Значение $\hat{f}^*=2$. Переменные x_1 и x_3 — базисные. Соответствующими базисными подматрицами матриц A и $\hat{A}$ являются

$$B=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\ \text{и}\ \hat{B}=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}.$$

Имеем

$$B+\varepsilon\hat{B}=\begin{pmatrix}1 & \varepsilon\\ \varepsilon & 1\end{pmatrix},\quad (B+\varepsilon\hat{B})^{-1}=\begin{pmatrix}\dfrac{1}{1-\varepsilon^2} & \dfrac{-\varepsilon}{1-\varepsilon^2}\\ \dfrac{-\varepsilon}{1-\varepsilon^2} & \dfrac{1}{1-\varepsilon^2}\end{pmatrix},$$

$$(B+\varepsilon\hat{B})^{-1}(b+\varepsilon\hat{b})=\left(\frac{1-\varepsilon-\varepsilon^2}{1-\varepsilon^2},\ \frac{2-\varepsilon^2}{1-\varepsilon^2}\right).$$

Отсюда $x_1(\varepsilon)=\dfrac{1-\varepsilon-\varepsilon^2}{1-\varepsilon^2}$, $x_2(\varepsilon)=0$, $x_3(\varepsilon)=\dfrac{2-\varepsilon^2}{1-\varepsilon^2}$ и при достаточно малом ε вектор $x(\varepsilon)$ — решение возмущенной задачи. В частности, при интересующем нас значении $\varepsilon=0{,}1$ вектор $x(0,1)\approx(0{,}899;\ 0;\ 2{,}010)$. Значение $f^*(0,1)\approx 1{,}92$.

15.3. Использование метода возмущений при решении задач математического программирования, обладающих специфическими структурными особенностями, позволяет *декомпозировать* их на задачи меньшей размерности. Рассмотрим возмущенную задачу *линейного программирования*

$$(c,\ x)\to\min, \tag{13}$$

$$(A+\varepsilon\hat{A})x=b, \tag{14}$$

$$x\geqslant 0, \tag{15}$$

где A — блочно-диагональная матрица; $\tilde{A}_l$, $l=\overline{1,\ \nu}$,— ее блоки.

Тогда *порождающую* задачу можно записать в виде

$$\sum_{l=1}^{\nu}(\tilde{c}_l,\ \tilde{x}_l)\to\min, \tag{16}$$

$$\tilde{A}_l\tilde{x}_l=\tilde{b}_l,\ l=\overline{1,\ \nu}, \tag{17}$$

$$\tilde{x}_l\geqslant 0,\ l=\overline{1,\ \nu}, \tag{18}$$

где векторы $\tilde{c}_l$, $\tilde{b}_l$, $\tilde{x}_l$ определяются блоками $\tilde{A}_l$ матрицы A.

Пусть требуется решить задачу (13)—(15) при некотором заданном достаточно малом ε. Следуя общему подходу, на первом шаге решаем порождающую задачу (16)—(18), которая распадается на ν подзадач ($l=\overline{1,\ \nu}$):

$$(\tilde{c}_l, \tilde{x}_l) \to \min, \tag{19}$$

$$\tilde{A}_l \tilde{x}_l = \tilde{b}_l, \tag{20}$$

$$\tilde{x}_l \geqslant 0. \tag{21}$$

Если решения $\tilde{x}_l^*$, $l = \overline{1, \nu}$, этих подзадач единственны, то $x^* = (\tilde{x}_1^*, \tilde{x}_2^*, \ldots, \tilde{x}_\nu^*)$ с точностью до $0(\varepsilon)$ является решением задачи (13)—(15).

Если в какой-либо из подзадач (скажем, l) решение неединственно, то необходимо идентифицировать множество $\tilde{X}_l^*$ всех ее решений.

Обозначим через $\tilde{\Lambda}_l^*$ решение задачи, двойственной к задаче (19)—(21). Целевая функция (11) *вспомогательной* задачи в рассматриваемом случае записывается в виде $(-\hat{A}^\tau \Lambda^*, x)$, а ограничения (12) — в виде $\tilde{x}_l \in \tilde{X}_l^*$, $l = \overline{1, \nu}$, где $\Lambda^* = (\tilde{\Lambda}_1^*, \tilde{\Lambda}_2^*, \ldots, \tilde{\Lambda}_\nu^*)$. Следовательно, задача распадается на ν подзадач $(l = \overline{1, \nu})$:

$$(\tilde{d}_l, \tilde{x}_l) \to \min, \tag{22}$$

$$\tilde{x}_l \in \tilde{X}_l^*, \tag{23}$$

где $\tilde{d}_l$ — вектор, содержащий те компоненты вектора $(-\hat{A}^\tau \Lambda^*)$, которые соответствуют вектору $\tilde{x}_l$.

Предполагая, что каждая из задач (22)—(23) имеет единственное решение $\hat{x}_l^*$, можно принять в качестве приближенного (с точностью до $0(\varepsilon)$) решения исходной задачи вектор $\hat{x}^* = (\hat{x}_1^*, \hat{x}_2^*, \ldots, \hat{x}_\nu^*)$.

Для уточнения решения, как и в общем случае, выделяются все базисные переменные всех векторов $\hat{x}_l^*$, $l = \overline{1, \nu}$. Пусть эти переменные образуют вектор $\bar{x}$. Обозначим соответствующие вектору $\bar{x}$ подматрицы матриц A и $\hat{A}$ через B и $\hat{B}$. Для нахождения новых значений базисных переменных достаточно решить систему уравнений $(B + \varepsilon \hat{B}) \bar{x} = b$.

Проверка опорного плана на оптимальность осуществляется обычным для линейного программирования способом. Обозначим через $\bar{c}$ вектор, образованный компонентами вектора c, соответствующими вектору $\bar{x}$. Опорный план оптимален, если $\Lambda(\tilde{A}+\varepsilon\hat{A})\leqslant c$, где $\Lambda=\bar{c}(B+ +\varepsilon\hat{B})^{-1}$.

Схема решения задачи (13)—(15) представлена на рис. 15.1.

15.4. Пример. Решим описанным методом возмущенную задачу

$$-x_1-3x_2-x_3-x_4\to\min,$$
$$2x_1+x_2+\varepsilon(x_3+2x_4)\leqslant 8,$$
$$x_1+x_2+\varepsilon(x_3-x_4)\leqslant 5,$$
$$\varepsilon(x_1+2x_2)+x_3+x_4\leqslant 4,$$
$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0$$

при достаточно малом $\varepsilon=\varepsilon_0$.

Порождающая задача

$$-x_1-3x_2-x_3-x_4\to\min,$$
$$\begin{aligned} 2x_1+x_2 \qquad\qquad &\leqslant 8,\\ x_1+x_2 \qquad\qquad &\leqslant 5,\\ x_3+x_4 &\leqslant 4,\end{aligned}$$
$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0$$

распадается на две подзадачи:

$$\begin{array}{ll} -x_1-3x_2\to\min, & -x_3-x_4\to\min,\\ 2x_1+x_2\leqslant 8, & x_3+x_4\leqslant 4,\\ x_1+x_2\leqslant 5, & x_3\geqslant 0,\ x_4\geqslant 0.\\ x_1\geqslant 0,\ x_2\geqslant 0, & \end{array}$$

Решением первой подзадачи является точка $x_1^*=0$, $x_2^*=5$, решением второй — множество точек, описываемое соотношениями $x_3+x_4=4$, $x_1\geqslant 0$, $x_2\geqslant 0$.

Двойственными к указанным подзадачам являются следующие задачи:

$$\begin{array}{ll} -8\lambda_1-5\lambda_2\to\max, & -4\lambda_3\to\max,\\ 2\lambda_1+\lambda_2\geqslant 1, & \lambda_3\geqslant 1,\\ \lambda_1+\lambda_2\geqslant 3, & \\ \lambda_1\geqslant 0,\ \lambda_2\geqslant 0. & \end{array}$$

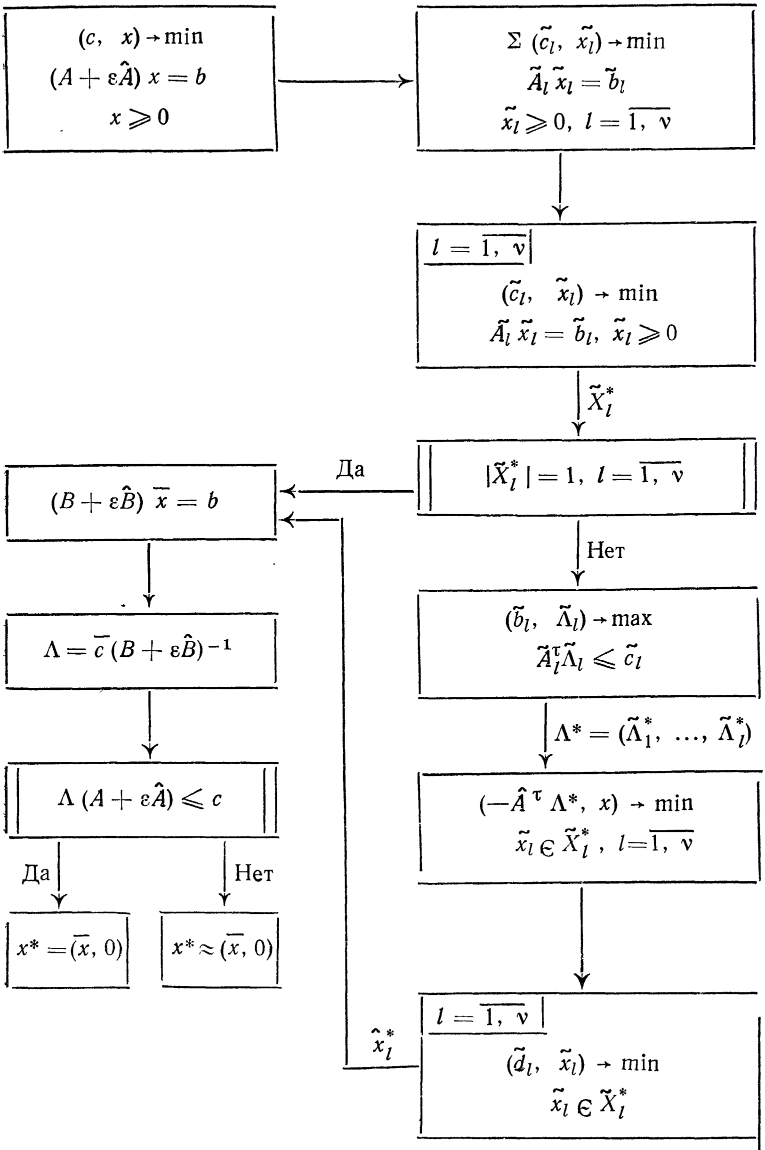

Рис. 15.1. Схема декомпозиции задач линейного программирования при блочно-диагональной матрице A и малом $\varepsilon > 0$

Имеем $\lambda_1^*=0$, $\lambda_2^*=3$, $\lambda_3^*=1$. В рассматриваемом случае матрица

$$\hat{A}=\begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & -1 \\ 1 & 2 & 0 & 0 \end{pmatrix}.$$

Следовательно, вспомогательная задача

$$-x_1-2x_2-3x_3+3x_4\to\min,$$

$$x_1=0,\ x_2=5,\ x_3+x_4=4,\ x_3\geqslant 0,\ x_4\geqslant 0$$

распадается на две подзадачи. Ее решение: $\hat{x}_1^*=0$, $\hat{x}_2^*=5$, $\hat{x}_3^*=4$, $\hat{x}_4^*=0$.

Базисными являются переменные x_2 и x_3. Заменяя в последних двух ограничениях исходной задачи знаки неравенства на равенства и приравнивая нулю переменные x_1 и x_4, получаем систему уравнений

$$\begin{cases} x_2+\varepsilon x_3=5, \\ 2\varepsilon x_2+x_3=4, \end{cases}$$

решение которой $x_2(\varepsilon)=\dfrac{5-4\varepsilon}{1-2\varepsilon^2}$ и $x_3(\varepsilon)=\dfrac{4-10\varepsilon}{1-2\varepsilon^2}$. Первое из неравенства при малом ε в точке $\hat{x}^*$ не существенно, т. е. выполняется со знаком строгого неравенства.

Итак, $x_1(\varepsilon_0)=0$, $x_2(\varepsilon_0)=\dfrac{5-4\varepsilon_0}{1-2\varepsilon_0^2}$, $x_3(\varepsilon_0)=\dfrac{4-10\varepsilon_0}{1-2\varepsilon_0^2}$, $x_4(\varepsilon_0)=0$ — решение исходной задачи (при заданном достаточно малом ε_0).

15.5. Ряд практических ситуаций приводит к необходимости решения возмущенных задач, порождающие задачи которых допускают непосредственное *агрегирование* переменных.

Рассмотрим, например, *задачу линейного программирования* следующего вида:

$$\sum_{l=1}^{\nu}\sum_{j\in N_l}(c_l+\varepsilon d_{lj})x_j\to\min, \tag{24}$$

$$\sum_{l=1}^{\nu}\sum_{j\in N_l}(A_l+\varepsilon B_{lj})x_j\leqslant b, \tag{25}$$

$$x_j\geqslant 0,\ j=\overline{1,n}. \tag{26}$$

Здесь N_l, $l=\overline{1, \nu}$,— попарно-непересекающиеся непустые подмножества множества $N=\{1, 2, \ldots, n\}$, $\bigcup\limits_{l=\overline{1, \nu}} N_l=N$; c_l, d_{lj} — скалярные величины; A_l, B_{lj}, b — векторы; ε — параметр.

Введем в рассмотрение *агрегированные* переменные X_l, $l=\overline{1, \nu}$, полагая

$$X_l=\sum_{j\in N_l} x_j. \tag{27}$$

Тогда задачу (24)—(26) можно переписать в виде

$$\begin{gathered}\sum_{l=1}^{\nu} c_l X_l+\varepsilon \sum_{l=1}^{\nu} \sum_{j\in N_l} d_{lj} x_j \to \min, \\ \sum_{l=1}^{\nu} A_l X_l+\varepsilon \sum_{l=1}^{\nu} \sum_{j\in N_l} B_{lj} x_j \leqslant b, \\ X_l \geqslant 0, \quad l=\overline{1, \nu}, \quad x_j \geqslant 0, \quad j=\overline{1, n}.\end{gathered} \tag{28}$$

Порождающая задача в агрегированных переменных записывается в виде

$$\sum_{l=1}^{\nu} c_l X_l \to \min, \tag{29}$$

$$\sum_{l=1}^{\nu} A_l X_l \leqslant b, \tag{30}$$

$$X_l \geqslant 0, \quad l=\overline{1, \nu}. \tag{31}$$

Пусть эта задача имеет единственное решение X_l^*, $l=\overline{1, \nu}$. Обозначим множество индексов l, $1\leqslant l\leqslant\nu$, при которых $X_l^*>0$, через L.

Переходя к исходным переменным x_j (согласно соотношению (27)) множество решений порождающей задачи можно описать соотношениями

$$\sum_{j\in N_l} x_j=X_l^*, \quad x_j\geqslant 0, \quad j\in N_l, \quad l\in L,$$

$$x_j=0, \quad j\in N_l, \quad l\notin L.$$

Задачей, двойственной к порождающей (29)—(31), является следующая:

$$-(b, \Lambda) \to \max, \tag{32}$$

$$\sum_{l=1}^{\nu} A_l^{\tau} \Lambda \geqslant -c, \tag{33}$$

$$\Lambda \geqslant 0. \tag{34}$$

Пусть Λ^* — решение этой задачи. Как уже отмечалось, решение исходной задачи (24)—(26) при достаточно малом ε с точностью до $0(\varepsilon^2)$ по функционалу совпадает с решением *вспомогательной* задачи

$$\sum_{l \in L} \sum_{j \in N_l} (d_{lj} - B_{lj}^{\tau} \Lambda^*) x_j \to \min, \tag{35}$$

$$\sum_{j \in N_l} x_j = X_l^*, \quad x_j \geqslant 0, \quad j \in N_l, \quad l \in L, \tag{36}$$

которая распадается на $|L|$ задач вида

$$\sum_{j \in N_l} (d_{lj} - B_{lj}^{\tau} \Lambda^*) x_j \to \min, \tag{37}$$

$$\sum_{j \in N_l} x_j = X_l^*, \quad x_j \geqslant 0, \quad j \in N_l. \tag{38}$$

Решение каждой из таких задач тривиально: достаточно найти индекс $j = j_l \in N_l$, соответствующий наименьшему коэффициенту целевой функции, и положить $\hat{x}_{j_l}^* = X_l^*$. Все остальные $\hat{x}_j^*$ полагаются равными нулю.

Если все j_l, $l \in L$, определяются однозначно, то ненулевые компоненты решения исходной задачи могут быть найдены в результате решения системы уравнений, получаемой из (25) путем замены знака неравенства на знак равенства и приравниванием нулю всех переменных x_j кроме x_{j_l}, $l \in L$.

Если индексы j_l определяются неоднозначно, то указанное решение отличается от оптимального (по функционалу) не более чем на $0(\varepsilon^2)$.

Описанная *схема агрегирования переменных* представлена на рис. 15.2.

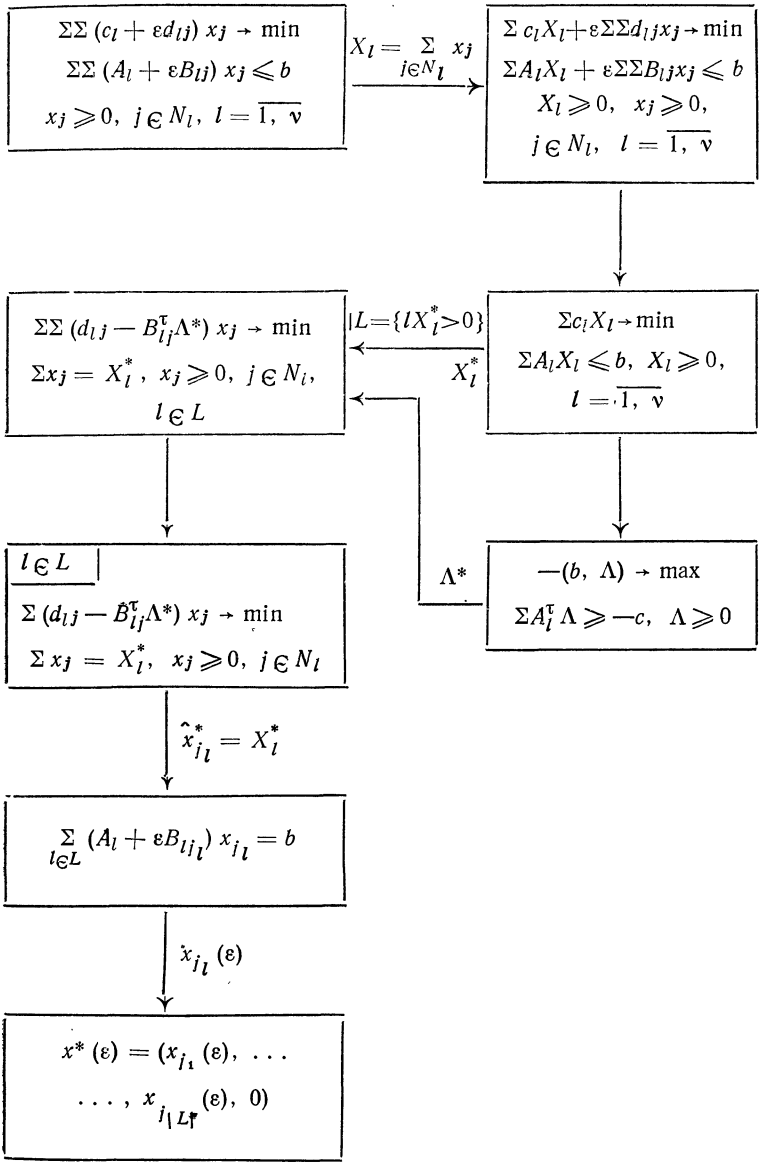

Рис. 15.2. Схема агрегирования переменных при малом $\varepsilon > 0$

12. Зак. 1201

15.6. Пример. Пусть требуется решить задачу

$$-2(x_1+x_2)-(x_3+x_4)-\varepsilon(x_1+2x_2+3x_3-x_4)\to\min,$$

$$(x_1+x_2)+(x_3+x_4)+\varepsilon(x_1+2x_2+x_3+x_4)\leqslant 4,$$

$$(x_1+x_2)-(x_3+x_4)-\varepsilon(x_1+x_2-x_3-x_4)\leqslant 2,$$

$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad x_3\geqslant 0,\quad x_4\geqslant 0$$

при достаточно малом $\varepsilon=\varepsilon_0$.

Введем агрегированные переменные X_1 и X_2, полагая

$$X_1=x_1+x_2,\quad X_2=x_3+x_4.$$

Порождающая задача (29)—(31) в рассматриваемом случае имеет вид

$$-2X_1-X_2\to\min,$$

$$X_1+X_2\leqslant 4,$$

$$X_1-X_2\leqslant 2,$$

$$X_1\geqslant 0,\quad X_2\geqslant 0.$$

Ее решением являются $X_1^*=3$, $X_2^*=1$.

Двойственная к порождающей задача (32)—(34) имеет вид

$$-4\lambda_1-2\lambda_2\to\max,$$

$$\lambda_1+\lambda_2\geqslant 2,$$

$$\lambda_1-\lambda_2\geqslant 1,$$

$$\lambda_1\geqslant 0,\quad \lambda_2\geqslant 0.$$

Ее решением являются $\lambda_1^*=1\frac{1}{2}$, $\lambda_2^*=\frac{1}{2}$.

Сформулируем вспомогательную задачу (35)—(36)

$$-2x_1-4\frac{1}{2}x_2-5x_3-x_4\to\min,$$

$$x_1+x_2=3,\quad x_3+x_4=1,$$

$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad x_3\geqslant 0,\quad x_4\geqslant 0,$$

которая распадается на две подзадачи вида (37)—(38)

$$-2x_1-4\frac{1}{2}x_2\to\min,\quad -5x_3-x_4\to\min,$$

$$x_1+x_2=3,\quad x_3+x_4=1,$$

$$x_1\geqslant 0,\quad x_2\geqslant 0,\quad x_3\geqslant 0,\quad x_4\geqslant 0,$$

имеющих тривиальные решения: $\hat{x}_1^*=0$, $\hat{x}_2^*=3$, $\hat{x}_3^*=1$, $\hat{x}_4^*=0$.

Для нахождения решения исходной задачи (при малом $\varepsilon=\varepsilon_0$) достаточно решить систему уравнений

$$\begin{cases} x_2+x_3+\varepsilon_0(2x_2+x_3)=4,\\ x_2-x_3-\varepsilon_0(x_2-x_3)=2. \end{cases}$$

Имеем $x_2(\varepsilon_0)=\dfrac{2\varepsilon_0-6}{3\varepsilon_0^2-\varepsilon_0-2}$, $x_3(\varepsilon_0)=\dfrac{8\varepsilon_0-2}{3\varepsilon_0^2-\varepsilon_0-2}$, $x_1(\varepsilon_0)=0$, $x_4(\varepsilon_0)=0$.

ЛИТЕРАТУРА

1. *Алексейчик М. И., Калюжный Г. В.* Теоретическое обоснование и алгоритм процедуры релаксации для задач квазивогнутого программирования // Изв. АН СССР. Техн. кибернетика. 1980. № 4. С. 37—42.

2. *Бейко И.В., Бублик Б. Н., Зинько П. Н.* Методы и алгоритмы решения задач оптимизации. Киев, 1983. 511 с.

3. *Васильев Ф. П.* Численные методы решения экстремальных задач. М., 1980. 518 с.

4. *Верина Л. Ф., Танаев В. С.* Декомпозиционные подходы к решению задач математического программирования // Эконом. и мат. методы. 1975. Т. 11, № 6. С. 1160—1172.

5. *Вулис В. И., Шерман Ю. С.* Метод декомпозиции для решения больших задач транспортного типа // Эконом. и мат. методы. 1985. Т. 21, № 2. С. 304—314.

6. *Габасов Р., Кириллова Ф. М.* Методы оптимизации // Минск, 1981. 350 с.

7. *Гольштейн Е. Г., Третьяков Н. В.* Модифицированные функции Лагранжа и их применение // Эконом. и мат. методы. 1983. Т. 19, № 3. С. 528—547.

8. *Гольштейн Е. Г., Юдин Д. Б.* Новые направления в линейном программировании. М., 1966. 524 с.

9. *Данциг Дж.* Линейное программирование, его обобщения и применения. М., 1966. 600 с.

10. *Данциг Дж., Вольф Ф.* Алгоритм разложения для задач линейного программирования // Математика. 1964. Т.8, № 1. С. 151—160.

11. *Ермольев Ю. М., Ермольева Л. Г.* Метод параметрической декомпозиции //Кибернетика. 1973. № 2. С. 66—69.

12. *Карманов В. Г.* Математическое программирование. М., 1980. 256 с.

13. *Корнаи И., Липтак Т.* Планирование на двух уровнях // Применение матем. в эконом. исслед. М., 1965. С. 107—136.

14. *Левин Г. М., Танаев В. С.* Декомпозиционные методы оптимизации проектных решений. Минск, 1978. 240 с.

15. *Лэсдон Л. С.* Оптимизация больших систем. М., 1975. 431 с.

16. *Медницкий В. Г.* Об оптимальности агрегирования в блочной задаче линейного программирования // Матем. методы решения эконом. задач. 1972. Вып. 3. С. 3—17.

17. *Месарович М., Мако Д., Такахара И.* Теория иерархических многоуровневых систем. М., 1973. 344 с.

18. *Моисеев Н. Н., Иванилов Ю. П., Столярова Е. М.* Методы оптимизации. М., 1978. 351 с.

19. *Оскорбин Н. М.* О схемах численных методов блочного программирования // Эконом. и мат. методы. 1981. Т. 17, № 5. С. 964—972.

20. *Первозванский А. А., Гайцгори В. Г.* Декомпозиция, агрегирование и приближенная оптимизация. М., 1979. 342 с.

21. *Пшеничный Б. Н., Данилин Ю. М.* Численные методы в экстремальных задачах. М., 1975. 319 с.

22. *Сергиенко И. В.* Математические модели и методы решения задач дискретной оптимизации. Киев, 1985. 381 с.

23. *Телле В.* Применение модифицированной функции Лагранжа в блочном программировании // Эконом. и мат. методы. 1975. Т. 11, № 3. С. 525—534.

24. *Тенев Т. П.* Декомпозиция задач линейного раскроя // Изв. АН СССР. Техн. кибернетика. 1978. № 5. С. 56—61.

25. *Химмельблау Д.* Прикладное нелинейное программирование. М., 1975. 534 с.

26. *Цурков В. И.* Декомпозиция в задачах большой размерности. М., 1981. 351 с.

27. *Шор Н. З.* Методы минимизации недифференцируемых функций и их приложения. Киев, 1979. 199 с.

28. *Юдин Д. Б., Гольштейн Е. Г.* Линейное программирование. М., 1963. 775 с.

29. *Beer K.* Lösung grober linearer Optimierung-saufgaben // Berlin: VEB Dtsch. Verl. Wiss. 1977. 254 S.

30. *Benders J. F.* Partitioning procedures for solving mixed variables programming problems // Numericshe Mathematik. 1962. V. 4, N 3. P. 238—252.

31. *Dantzig G. B.* Mathematical programming: decomposition // Progr. Operat. Res. New York—London—Sydney—Toronto. 1969. P. 175—193.

32. *Geoffrion A. M.* Generalized Benders decomposition // J. of Opt. theory and its Appl. 1972. V. 10, N 4. P. 237—260.

33. *Hagelschuer P. B.* Theorie der liniaren Dekomposition. Berlin: Springer. 1971. V. 11. 191 S.

34. *Himmelblau D. M.* (editor). Decomposition of large-scale problems. Amsterdam: North-Holland. 1973. 571 p.

35. *Tomlin J. A.* Large scale mathematical programming systems // Comput. and Chem. Eng. 1983. V. 7, N 5. P. 575—582.

36. *Wismer D. A.* (editor). Optimization methods for large-scale systems with application. New York: McGraw-Hill. 1972. 355 p.

БИБЛИОГРАФИЧЕСКАЯ СПРАВКА

Вопросам декомпозиции и агрегирования в задачах математического программирования посвящена обширная литература, в том числе монографии Е. Г. Гольштейна и Д. Б. Юдина [8], Г. М. Левина и В. С. Танаева [14], И. В. Сергиенко [22], А. А. Первозванского и В. Г. Гайцгори [20], В. И. Цуркова [26], Л. С. Лэсдона [15], К. Беера [29], П. Б. Хагельшуера [33], Д. Химмельблау [34], Д. А. Висмера [36], обзорные статьи Л. Ф. Вериной и В. С. Танаева [4], Дж. Б. Данцига [31], Дж. А. Томлина [35] и др. Многие из этих вопросов можно интерпретировать в терминах теории многоуровневых иерархических систем (см., например, монографию М. Месаровича, Д. Мако и И. Такахары [17]).

С различными методами оптимизации можно более детально ознакомиться по книгам В. Г. Карманова [12], Н. Н. Моисеева, Ю. П. Иванилова и Е. М. Столяровой [18], Ф. П. Васильева [3], Б. Н. Пшеничного и Ю. М. Данилина [21], Р. Габасова и Ф. М. Кирилловой [6], Н. З. Шора [27], Д. Б. Юдина и Е. Г. Гольштейна [28], Дж. Данцига [9], Д. Химмельблау [25[. В справочном пособии И. В. Бейко,Б. Н. Бублика и П. Н. Зинько [2] содержатся краткие описания большинства известных методов и алгоритмов оптимизации.

Содержание параграфа 4 составили статья М. И. Алексейчика и Г. В. Калюжного [1] и соответствующий раздел монографии Л. С. Лэсдона [15]. В основу параграфа 5 положены статьи И. Корнаи и Т. Липтака [13], Ю. М. Ермольева и Л. Г. Ермольевой [11].

Более полное описание метода параметрической декомпозиции, его обобщений и приложений содержится в монографии Г. М. Левина и В. С. Танаева [14].

Изложенный в параграфах 7—9 декомпозиционный подход известен в литературе как метод Данцига—Вулфа (см. [8], [10], [15], [19], [29] и др.), а подход, изложенный в параграфах 10, 11,—как метод Бендерса (см. [15], [30], [32] и др.). Декомпозиционные методы решения достаточно сложных задач транспортного типа и задач линейного раскроя предложены в статьях В. И. Вулиса и Ю. С. Шермана [5] и Т. П. Тенева [24].

С теорией двойственности можно ознакомиться по книгам Л. С. Лэсдона [15], Р. Габасова и Ф. М. Кирилловой [6], В. П. Васильева [3] и др. Параграф 12, посвященный модифицированным функциям Лагранжа, написан на основе статьи Е. Г. Гольштейна и Н. В. Третьякова [7]. См. также статью В. Телле [23].

Детальное изложение вопросов агрегирования переменных из разных блоков (далеко выходящее за рамки параграфа 13) содержится в монографии В. И. Цуркова [26]. В основу параграфа 14 положена статья В. Г. Медницкого [16] и соответствующий раздел монографии В. И. Цуркова [26].

Анализу различных аспектов применения метода возмущений для решения задач математического программирования и управления динамическими системами посвящена монография А. А. Первозванского и В. Г. Гайцгори [20].

ОГЛАВЛЕНИЕ

Вячеслав Сергеевич Танаев

ДЕКОМПОЗИЦИЯ И АГРЕГИРОВАНИЕ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Заведующая редакцией Н. Т. Ломако
Редактор С. В. Машканова
Художник Л. М. Гоманов
Художественный редактор Л. И. Усачев
Технический редактор В. А. Витенко
Корректор Е. Т. Трусова

ИБ № 2965

Печатается по постановлению РИСО АН БССР. Сдано в набор 21.08.86. Подписано в печать 23.10.86. АТ 17417. Формат 84×108[1]/32. Бум. тип. № 2. Гарнитура литературная. Высокая печать. Усл. печ. л. 9.66. Усл. кр.-отт. 9,97. Уч.-изд. л. 7,88. Тираж 1550 экз. Зак. № 1201. Цена 1 р. 30 к.
Издательство «Наука и техника» Академии наук БССР и Государственного комитета БССР по делам издательств, полиграфии и книжной торговли. 220600. Минск, Ленинский проспект, 68. Типография им. Франциска (Георгия) Скорины издательства «Наука и техника». 220600. Минск, Ленинский проспект, 68.