Б.Н. ПШЕНИЧНЫЙ

МЕТОД ЛИНЕАРИЗАЦИИ

Б. Н. ПШЕНИЧНЫЙ

МЕТОД ЛИНЕАРИЗАЦИИ

МОСКВА "НАУКА"
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1983

~~22.18~~
П 93
УДК 519.6

Метод линеаризации. П ш е н и ч н ы й Б. Н. – М.: Наука. Главная редакция физико-математической литературы, 1983. – 136 с.

Книга посвящена систематическому изложению метода линеаризации – одного из универсальных методов решения общих задач математического программирования, а также его применениям к задачам безусловной оптимизации, линейного и квадратичного программирования, нахождению решений систем неравенств, задачам минимакса. Изложение доведено до стадии описания конкретных алгоритмов для конкретных задач.

Табл. 6 Библ. 47 назв.

Борис Николаевич П ш е н и ч н ы й

МЕТОД ЛИНЕАРИЗАЦИИ

Редакторы *А.Д. Вайнштейн, И.В. Викторенкова*
Тех. редактор *С.В. Геворкян*
Корректоры *Т.В. Обод, Т.А. Печко*

ИБ № 12230

Сдано в набор 11.11.82. Подписано к печати 22.03.83
Т–06977. Бумага 60×90/16 офсетная. Печать офсетная
Усл.печ.л. 8,50. Уч.изд.л. 8,82. Тираж 7500 экз.
Тип.зак. 562. Цена 1 р. 10 к.

Издательство "Наука"
Главная редакция физико-математической литературы
117071 Москва, В–71, Ленинский проспект, 15
4-я типография издательства "Наука"
630077, Новосибирск, 77, ул. Станиславского, 25

П $\frac{1702070000-072}{053(02)-83}$ 25-83

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Обычно в предисловии принято писать о практической важности проблемы и характеризовать содержание книги. Но практическая важность решения задач оптимизации давно уже не вызывает каких-либо сомнений. Этой тематике посвящена такая обширная научная и научно-популярная литература, что вряд ли есть необходимость еще раз повторять, что теория и методы оптимизации находят приложение в многочисленных проблемах экономики, автоматическом управлении, инженерном деле и т.д. С другой стороны, краткое содержание подробно описано в первом вводном параграфе этой книги. В связи с этим на нем также нет смысла останавливаться. Вместо этого я позволю себе высказать несколько общих замечаний относительно теории и численных методов оптимизации и их взаимодействия в процессе решения реальной сложной задачи.

Электронные вычислительные машины начали применяться для решения задач оптимизации с первых лет своего появления. Первоначально это применение было связано с относительно простыми по своей структуре задачами линейного программирования, которые можно было решать разработанными регулярными методами, и со сравнительно несложными нелинейными задачами, решение которых достигалось за счет простых интуитивных соображений в соединении со способностью вычислительной техники производить огромное количество вычислительных операций. Но появление все новых и новых задач возрастающего объема, содержащих сложные нелинейности, не позволяло ограничиться простыми приемами и грубой силой. Решение достаточно общих нелинейных задач требовало как более глубокого теоретического исследования свойств их решений, так и более тонких и сложных приемов для получения численного результата в разумное для данной конкретной задачи время. Процесс создания такого арсенала средств для решения проблемы оптимизации интенсивно развивался в течение последних двадцати лет.

Представляется, что метод линеаризации — предмет исследований в настоящей книге — является одним из довольно многих плодов, полученных в результате этого процесса.

Действительно, метод линеаризации, как это видно из дальнейшего изложения, тесно связан с методом Ньютона решения систем уравнений, методом штрафных функций, в частности, негладких штрафных функций, а в связи с последним обстоятельством и с методами недифференцируемой оптимизации. Успешное применение метода линеаризации невозможно без эффективного решения задач квадратичного программирования, что связывает его с методами сопряженных градиентов и переменной

метрики. Желание решать задачи большого объема требует привлечения аппарата, разработанного в линейном программировании, т.е. приемов работы с разреженными матрицами, мультипликативного представления обратных матриц и т.п.

Наконец, исследование области и скорости сходимости невозможно без привлечения теории необходимых условий экстремума, множителей и функций Лагранжа, понятия двойственной задачи. Как следует из сказанного, достаточно законченное изложение свойств метода линеаризации невозможно без привлечения широкого круга понятий, разработанного как в самой абстрактной теории, так и при самой конкретной машинной реализации. Естественно, что все это в большей или меньшей степени нашло отражение в книге. Не всем вопросам можно было уделить одинаковое внимание при разумном ограничении объема книги. Однако, даже если какие-то проблемы упоминаются лишь бегло (например – методы сопряженных направлений и переменной метрики, методы работы с разреженными матрицами), то это не значит, что они не имеют существенного значения. Чаще наоборот, как это показывает пример с разреженными матрицами. Без умения успешно работать с ними время работы с задачами большого объема катастрофически возрастает.

Укажем еще на одну и очень существенную причину, в силу которой специалист, заинтересованный в первую очередь в конкретном применении метода, должен тем не менее быть хотя бы знакомым с понятиями и идеями, связанными с методами. Дело в том, что метод рассчитан на решение общей задачи нелинейного программирования и может в силу этого затрачивать на решение каких-то подзадач данной задачи значительное время. Но конкретная задача (или класс задач) всегда имеет свою специфику (например, большинство ограничений имеет очень простой вид), и поэтому учет этой специфики за счет изменения каких-то блоков алгоритма может привести к существенному сокращению времени решения и требуемого объема памяти. Ясно, что такое изменение нельзя провести успешно без понимания движущих пружин, которые делают алгоритм сходящимся.

Хотя данная книга продиктована стремлением подвести некоторый итог исследованиям в области метода линеаризации, автор надеется, что она вовсе не будет служить конечной точкой в этой области. Свидетельством тому является все увеличивающееся число статей, посвященных этой и связанной с ней тематике. Кроме того, имеется целый ряд проблем, требующих более глубокого и детального изучения. В частности, требуют дальнейшего исследования вопросы выбора правил перехода с простого на ускоренный метод линеаризации, уточнение правил выбора шага при таком переходе, алгоритм изменения константы в функции штрафа, способы аппроксимации матрицы вторых производных функций Лагранжа в решении. Необходимо более детально рассмотреть специфику метода линеаризации применительно к задачам большего объема, методы декомпозиции исходной и вспомогательной задач и т.д. Все эти проблемы нетривиальны, и автор будет рад, если данная книга послужит не только для расширения сферы практического применения метода, но и отправным пунктом для его совершенствования и развития.

Б.Н. Пшеничный

Глава 1.

ЗАДАЧИ ВЫПУКЛОГО И КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

§ 1. ВВЕДЕНИЕ

Предлагаемая читателю книга целиком, за исключением § 8, посвящена одному методу решения задач нелинейного программирования – методу линеаризации. Этим она отличается от большинства книг, посвященных тому же предмету, в которых обычно рассматриваются различные методы. Описание в монографиях различных алгоритмов и подходов не случайно. Оно связано с тем, что многолетняя практика решения нелинейных задач оптимизации привела специалистов к достаточно единодушному мнению о невозможности создания универсального алгоритма, который бы одинаково успешно решил все задачи. И автор этой книги с таким мнением полностью согласен. Действительно, задачи нелинейной оптимизации чрезвычайно разнообразны. Они отличаются структурой вхождения нелинейности, количеством переменных и ограничений, требуемым объемом памяти. И практический опыт показывает, что существуют классы задач, в которых, казалось бы, самый неэффективный с теоретической точки зрения метод благодаря простоте реализации и специфике задачи дает хорошие результаты. В связи с этим вычислительная практика в целом требует набора различных алгоритмов. Сосредоточение же в этой книге на одном методе связано с желанием достаточно глубоко выяснить его свойства и возможности, выделить те особенности и преимущества, которые он может дать на практике. Тем более что использование метода линеаризации на практике в течение многих лет показало его высокую эффективность при решении достаточно широких классов задач. Поэтому здесь мы постараемся выделить наиболее характерные черты, обеспечивающие его высокую эффективность.

1. Более точная постановка задачи и требования к ней будут даны по ходу изложения. Здесь же мы не будем себя ограничивать особой математической строгостью.

Итак, пусть $I = \{1, \ldots, m\}$ – конечное множество индексов. Рассматривается задача нахождения минимума функции $f_0(x)$ при ограничениях $f_i(x) \leqslant 0$, $i = 1, \ldots, m$. Более коротко,

$$\min_x \{f_0(x) : f_i(x) \leqslant 0, \quad i = 1, \ldots, m\}. \tag{1.1}$$

Хотя можно без особого труда в методе линеаризации рассматривать и ограничения вида $f_i(x) = 0$, для упрощения здесь мы этого делать не будем.

По аналогии с известным методом Ньютона решения систем нелинейных уравнений попробуем нелинейную задачу (1.1) в данной точке x линеаризовать и приращение аргумента вычислить из решения соответствующей

линейной задачи:

$$\min_p \{ f_0(x) + (f_0'(x), p):\ f_i(x) + (f_i'(x), p) \leqslant 0,\ i = 1, \ldots, m \}, \qquad (1.2)$$

где $f_i'(x)$ – градиенты функций $f_i(x)$. В качестве нового приближения можно было бы взять точку $x + p$. Однако, как правило, задача (1.2) не будет иметь решения – минимум в ней будет равен $-\infty$. Это связано с тем, что линейное приближение к нелинейной функции справедливо лишь в некоторой окрестности точки x. Поэтому естественно как-то попытаться учесть этот факт. Такой учет можно сделать двумя способами: прямым ограничением нормы вектора p, т.е. дополнением задачи (1.2) ограничением

$$\| p \| \leqslant \delta,\ \ \delta > 0, \qquad (1.3)$$

либо с помощью штрафа за большие уклонения. Второй способ приводит к следующей вспомогательной задаче:

$$\min_p \left\{ (f_0'(x), p) + \tfrac{1}{2} \| p \|^2:\ f_i(x) + (f_i'(x), p) \leqslant 0,\ i = 1, \ldots, m \right\}. \qquad (1.4)$$

В дальнейшем рассматривается только второй способ.

Методы, основанные на введении ограничения (1.3) во вспомогательную задачу, достаточно часто рассматривались в литературе, в частности, подробно они исследованы в [37].

Таким образом, если x – какое-либо приближение к решению задачи (1.1), а $p(x)$ – решение задачи (1.4), то следующее приближение ищется в виде $x + p(x)$. Однако хорошо известно, что даже при решении систем уравнений методом Ньютона такой выбор следующего приближения в лучшем случае обеспечивает лишь локальную сходимость, т.е. сходимость с достаточно хорошего начального приближения. Для практики же хотелось бы обеспечить сходимость из широкой области начальных приближений, лучше всего с любого начального приближения. Поэтому будем искать следующее приближение в виде $x_1 = x + \alpha p(x)$, где α – число из полуинтервала $(0,1]$, определяющее шаг вдоль направления $p(x)$. Естественно, что α, вообще говоря, зависит от x и возникает новая проблема – требуется дать правило выбора этого шага.

В задачах безусловной оптимизации правило выбора шага определяется условием, что минимизируемая функция убывает достаточно существенно в заданном направлении. Однако теперь мы имеем дело с задачей минимизации при ограничениях и просто взять для этой цели минимизируемую функцию нельзя. Вместо этого берется целевая функция задачи плюс штраф за нарушение ограничений. Более точно, пусть

$$F(x) = \max\{0, f_1(x), f_2(x), \ldots, f_m(x)\},$$

$$\Phi_N(x) = f_0(x) + N F(x).$$

Будем выбирать шаг α следующим образом: берем $\alpha = 1$ и делим пополам эту величину до первого выполнения неравенства

$$\Phi_N(x + \alpha p(x)) \leqslant \Phi_N(x) - \epsilon \alpha \| p(x) \|^2,\ \ 0 < \epsilon < 1. \qquad (1.5)$$

То первое значение α, при котором неравенство (1.5) выполняется, и берется в качестве шага.

Оказывается, что если

$$N \geqslant \sum_{i=1}^{m} u^i(x), \tag{1.6}$$

где $u^i(x)$ – множители Лагранжа вспомогательной задачи (1.4), то неравенство (1.5) будет выполнено после конечного числа делений пополам.

Таким образом, алгоритм метода линеаризации состоит в следующем. Если приближение x_k уже построено, то для построения следующего приближения решается задача (1.4) при $x = x_k$, полагается $p_k = p(x_k)$ и

$$x_{k+1} = x_k + \alpha_k p_k ,$$

где α_k выбирается, исходя из неравенства (1.5) при $x = x_k$, описанным выше способом.

2. Рассмотрим теперь вопрос о сходимости предложенного алгоритма. Как показывают исследования, проведенные в § 4, решающим для сходимости алгоритма является выполнение неравенства (1.6), т.е. правильный выбор числа N. Конечно, оно заранее неизвестно. Но в задачу (1.4) число N не входит. Поэтому величины $u^i(x)$ могут вычисляться независимо от него. Проверка же выполнения неравенства (1.6) позволяет производить корректировку этого числа, увеличивая его, если первоначальный выбор был неудачным.

Если число N правильно скорректировано, то алгоритм сходится в следующем смысле: порожденная им последовательность точек x_k имеет своими предельными точками только такие, которые удовлетворяют всем ограничениям задачи (1.1), и в них выполняются необходимые условия минимума. В частности, если рассматривается задача выпуклого программирования, то любая такая предельная точка является ее решением. Более того, если рассматривается задача линейного программирования, то можно показать, что алгоритм сходится за конечное число шагов.

Заметим теперь, что алгоритм может быть применен к решению систем неравенств. В самом деле, если положить $f_0(x) = 0$, то решение задачи (1.1) эквивалентно просто решению системы неравенств $f_i(x) \leqslant 0$, $i = 1, \ldots, m$. Как показано в § 5, после небольшой корректировки выбора шага алгоритм при достаточно естественных предположениях в этом случае будет сходиться квадратично, т.е. как обычный метод Ньютона решения систем уравнений. Замечательно, что такой же особенностью он будет обладать и для задачи (1.1), если только в решении задачи (1.1) удовлетворятся как равенства n независимых ограничений $f_i(x) \leqslant 0$ (здесь n – размерность вектора x).

В § 7 изучается задача дискретного минимакса, т.е. задача минимизации функции

$$F(x) = \max_{1 \leqslant i \leqslant m} f_i(x),$$

и дается модификация алгоритма линеаризации, пригодная для решения этой задачи. В этом случае оказывается, что сходимость алгоритма квадратичная, если точка минимума функции $F(x)$ является так называемой

чебышевской точкой – предположение, которое, как правило, выполняется в задачах наилучшего равномерного приближения. Кстати, заметим, что и без этого предположения, как показала практика, алгоритм обладает высокой эффективностью применительно к минимаксным задачам.

В общем случае алгоритм линеаризации обладает лишь сходимостью со скоростью геометрической прогрессии. Это достаточно легко усмотреть из того, что для задачи без ограничений он превращается в обычный градиентный спуск, для которого именно такая сходимость строго обоснована. В ряде задач сходимость со скоростью геометрической прогрессии может оказаться неудовлетворительной. Поэтому в § 6 предлагается ряд приемов ускорения сходимости. Остановимся здесь лишь на одном из них. А именно, предлагается на каждом итерационном шаге вместо (1.4) решать задачу

$$\min_p\{(f_0'(x_k), p) + \tfrac{1}{2}(A_k p_k, p_k):\ f_i(x_k) + (f_i'(x_k), p) \leqslant 0,\ i = 1, \ldots, m\}.$$

В § 6 матрица A_k выбирается равной матрице вторых производных по x от функции Лагранжа

$$L(x, u) = f_0(x) + \sum_{i=1}^{m} u^i f_i(x)$$

и показывается, что при определенных предположениях это ведет к сходимости быстрее любой геометрической прогрессии. Но вычисление (даже по разностным формулам) матрицы вторых производных может быть весьма трудоемким. Поэтому было бы интересно провести дальнейшие исследования, связанные с методами рекуррентной переработки матрицы A_k в A_{k+1} так, как это делается, например, в [46]. При этом для обеспечения ускоренной сходимости достаточно лишь обеспечить сходимость A_k к матрице вторых производных функции Лагранжа в решении.

3. Остановимся теперь еще на ряде общих моментов, связанных с методом линеаризации. Как явствует из вышеизложенного, он сочетает в себе черты целого ряда известных методов – метода Ньютона, метода функций штрафов, обычного градиентного спуска. В частности, метод линеаризации можно рассматривать как метод минимизации штрафной функции $\Phi_N(x)$. Как известно (см. § 2), функция $\Phi_N(x)$ при достаточно большом N обладает той особенностью, что при некоторых предположениях ее точки минимума совпадают с решением задачи (1.1). Однако эта функция недифференцируема и к ней неприменимы обычные эффективные методы безусловной оптимизации. Метод линеаризации позволяет находить ее точки минимума, и при этом выбор направления сдвига p от точки к точке получается из решения задачи (1.4) и не зависит от выбора числа N. Далее, выбор этого направления инвариантен по отношению к масштабированию функций $f_i(x)$, $i = 1, \ldots, m$, задающих ограничения, т.е. к замене функций $f_i(x)$ на функции $a_i f_i(x)$, $a_i > 0$, что также важно для вычислительной практики.

При исследовании метода приходится использовать весь арсенал фактов из теории оптимизации. Поэтому в книгу включен § 2, содержащий основные теоремы теории выпуклых функций и необходимых условий экстремума. Этот параграф может служить для справок и хорошо осве-

домленным читателем может быть опущен. Кроме того, в § 2 многие теоремы приведены без доказательства. Однако во всех остальных параграфах приведены подробные доказательства. Это связано не только с желанием соблюсти математическую строгость. Дело в том, что, по мнению автора, доказательства сходимости алгоритмов являются не просто формальным обоснованием. Они содержат нечто большее – анализ причин сходимости и тех фактов, которые этому могут препятствовать. И знание этих фактов может служить основой анализа тех случаев, когда алгоритм не приводит к успешному решению задачи.

Хорошо известно, что самый лучший алгоритм можно загубить плохой реализацией на электронной вычислительной машине. Поэтому чрезвычайно важно использование всего арсенала приемов решения задачи квадратичного программирования при решении вспомогательной задачи (1.4), так как именно решение этой задачи требует основных затрат времени. В § 3 дается описание алгоритма решения задачи квадратичного программирования, обобщающего симплекс-метод задачи линейного программирования в мультипликативной форме. Использование именно такого алгоритма позволяет применить все методы экономии памяти и вычислений, работы с разреженными матрицами и т.п., разработанные в настоящее время. Впрочем, в силу специфического вида задачи (1.4) удобнее всего перейти к двойственной задаче, которая в данном случае имеет простые ограничения на переменные и может решаться некоторым обобщением метода сопряженных направлений, широко используемым для минимизации квадратичных функций без ограничений.

4. В книге используются стандартные обозначения, но, чтобы не возникало недоразумений, мы их коротко опишем.

Все рассмотрение ведется в n-мерном пространстве $\mathbf{R}^n$ вектор-столбцов x, y, z и т.п. Матрицы обозначаются буквами A, B, C, а транспортирование – с помощью звездочки сверху. I_n – единичная матрица порядка n. Как обычно, (x,y) – скалярное произведение векторов x и y:

$$(x, y) = \sum_{i=1}^{n} x^i y^i,$$

причем компоненты векторов обозначаются индексами сверху. Всюду используется евклидова норма, т.е.

$$\| x \| = \left(\sum_{i=1}^{n} (x^i)^2 \right)^{1/2}.$$

Для обозначения верхнего (нижнего) предела используются знаки $\overline{\lim}$ или lim sup (соответственно $\underline{\lim}$, lim inf). Знак $\lambda \downarrow 0$ показывает, что величина λ стремится к нулю, монотонно убывая.

Отметим еще обозначения производных от функций: $f'(x)$ – градиент, вектор-строка с компонентами $\partial f(x)/\partial x^i$, $i = 1, \ldots, n$; $f''(x)$ – матрица вторых производных, т.е.

$$f''(x) = \left\{ \frac{\partial^2 f(x)}{\partial x^i \, \partial x^j} \right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,n}}$$

Иногда, когда неясно, по какому аргументу берется производная, этот аргумент пишется внизу. Так, если

$$L(x,u) = f_0(x) + \sum_{i=1}^{n} u^i f_i(x)$$

есть функция Лагранжа, то $L''_{xx}(x, u)$ обозначает матрицу вторых производных по x.

Наконец, $\min_x \{f(x).\ x \in M\}$ обозначает значение минимума функции f по аргументу x, меняющемуся в множестве M

§ 2. НЕОБХОДИМЫЕ УСЛОВИЯ МИНИМУМА И ДВОЙСТВЕННОСТЬ

Современные методы решения задач математического программирования базируются на теоретическом фундаменте, развитом в выпуклом анализе и теории необходимых условий экстремума. Цель этого параграфа – дать краткую сводку результатов, которые так или иначе используются в дальнейшем.

Таким образом, данный параграф служит своеобразным справочником по теории выпуклого анализа, двойственности и необходимым условиям экстремума. Хотя читатель, интересующийся лишь практической стороной дела, т.е. вычислительными алгоритмами, может опустить вначале этот параграф, обращаясь к нему по мере необходимости за справками, тем не менее и ему, возможно, будет интересно взглянуть на сжатое содержание того теоретического материала, на котором базируются современные вычислительные методы. Более подробное изложение материала этого параграфа можно найти в [7, 23, 29, 47].

1. Выпуклые множества. Множество M в пространстве $\mathbf{R}^n$ называется *выпуклым*, если вместе с любыми двумя точками $x_1, x_2 \in M$ оно содержит и весь отрезок, их соединяющий, т.е. $\lambda_1 x_1 + \lambda_2 x_2 \in M$ при всех $\lambda_1, \lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$. Это свойство, характеризующее выпуклое множество, легко обобщить. А именно, если $x_i \in M$, $i = 1, \ldots, m$, и M выпукло, то $\lambda_1 x_1 + \ldots + \lambda_m x_m \in M$ для всех $\lambda_i \geqslant 0, \lambda_1 + \ldots + \lambda_m = 1$.

Непосредственно из определений соответствующих понятий вытекает, что внутренность int M и замыкание cl M выпуклого множества также выпуклы.

Важнейшее свойство выпуклых множеств состоит в том, что точку, не принадлежащую выпуклому множеству, можно отделить от него. Более точно, если M – замкнутое выпуклое множество и точка x_0 не принадлежит ему, то существуют вектор $a \in \mathbf{R}^n$ и $\epsilon > 0$ такие, что

$$(x, a) \leqslant (x_0, a) - \epsilon$$

для всех $x \in M$. При построении двойственных алгоритмов часто используется тот факт, что в качестве вектора a можно взять вектор $a = x_0 - y$, где y – ближайшая к точке x_0 точка множества M, т.е.

$$\|x_0 - y\| = \min_x \{\|x_0 - x\| : x \in M\},\ y \in M.$$

Среди выпуклых множеств своими особыми свойствами выделяются выпуклые конусы. Выпуклое множество K называется *выпуклым конусом*, если из того, что $x \in K$, вытекает, что $\lambda x \in K$ при всех $\lambda > 0$.

С каждым выпуклым конусом тесно связан сопряженный ему конус K^*. По определению

$$K^* = \{ y \in \mathbf{R}^n : \ (x,y) \geqslant 0 \quad \forall x \in K \}. \tag{2.1}$$

Если с каждым $y \in \mathbf{R}^n$ связать линейную функцию (x,y), то можно сказать, что *сопряженный конус* K^* есть множество линейных функций, принимающих на конусе K неотрицательные значения.

Характерным примером конуса является множество, задаваемое системой линейных неравенств:

$$K = \{ x : \ (a_i \ x) \geqslant 0, \ \ i = 1, \ \ldots, m \}.$$

Известно, что сопряженный конус K^* состоит из элементов y, представимых в виде

$$y = \sum_{i=1}^{n} u^i a_i, \ \ u^i \geqslant 0, \ \ i = 1, \ldots, m.$$

2. Выпуклые функции. Будем рассматривать функции, определенные для $x \in \mathbf{R}^n$ и принимающие значения в расширенной действительной оси. Таким образом, для $f(x)$ допускают значения $-\infty$ и $+\infty$. Такое соглашение о значениях функции f удобно при рассмотрении двойственных задач выпуклого программирования. В целом же в дальнейшем изложении функциями, принимающими несобственные значения, мы злоупотреблять не будем.

Свяжем с каждой функцией два множества – ее *надграфик* $\operatorname{epi} f$, т.е. множество пар $\{x, \alpha\}$ таких, что $x \in \mathbf{R}^n$, $\alpha \in \mathbf{R}$, $\alpha \geqslant f(x)$, и *область определения* $\operatorname{dom} f$ – множество x таких, что $f(x) < +\infty$.

Итак, по определению

$$\operatorname{epi} f = \{ \{x, \alpha\} \in \mathbf{R}^{n+1} : \ \alpha \geqslant f(x) \} \ .$$

$$\operatorname{dom} f = \{ x \in \mathbf{R}^n : \ f(x) < +\infty \} .$$

Общая *выпуклая функция* определяется как функция, надграфик которой есть выпуклое множество. Легко видеть, что область определения выпуклой функции есть выпуклое множество.

В дальнейшем будут рассматриваться лишь собственные выпуклые функции, т.е. такие, которые не принимают значения $-\infty$ и не равны тождественно $+\infty$. Отметим здесь только следующий факт. Пусть $f(y) = -\infty$ в какой-то точке y, а $x \in \operatorname{int} \operatorname{dom} f$ и f – выпуклая функция. При достаточно малом $\epsilon > 0$ справедливо $x_1 = x + \epsilon(x \quad y) \in \operatorname{dom} f$. Легко видеть, что

$$x = \frac{1}{1+\epsilon} \cdot x_1 + \frac{\epsilon}{1+\epsilon} - y.$$

Так как $f(y) = -\infty$, то $\{y, \beta\} \in \operatorname{epi} f$ при любом β. Пусть $\alpha_1 \geqslant f(x_1)$, т.е.

$\{x_1, \alpha_1\} \in \operatorname{epi} f$. В силу выпуклости надграфика

$$\left\{\frac{1}{1+\epsilon}x_1 + \frac{\epsilon}{1+\epsilon}y,\ \frac{1}{1+\epsilon}\alpha_1 + \frac{\epsilon}{1+\epsilon}\right\}\beta \in \operatorname{epi} f,$$

т.е

$$f(x) = f\left(\frac{1}{1+\epsilon}x_1 + \frac{\epsilon}{1+\epsilon}y\right) \leqslant \frac{1}{1+\epsilon}\alpha_1 + \frac{\epsilon}{1+\epsilon}\beta.$$

Так как β произвольно, то $f(x) = -\infty$. Итак, если выпуклая функция f принимает значение $-\infty$, то $f(x) = -\infty$ для всех $x \in \operatorname{int} \operatorname{dom} f$.

Для собственных выпуклых функций, которые в дальнейшем только и будут рассматриваться, определение через выпуклость надграфика эквивалентно обычному: функция f называется *выпуклой*, если

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leqslant \lambda_1 f(x_1) + \lambda_2 f(x_2),$$
$$\lambda_1, \lambda_2 \geqslant 0,\ \lambda_1 + \lambda_2 = 1. \qquad (2.2)$$

Выпуклость функции влечет за собой целый ряд аналитических свойств. Так, для того чтобы выпуклая функция была непрерывна в точке x_0, необходимо и достаточно, чтобы она была ограничена сверху в некоторой окрестности этой точки. Более того, в этом случае она удовлетворяет условию Липшица. Если функция обладает первыми или вторыми производными, то неравенства

$$f(y) - f(x) \geqslant (y - x, f'(x)) \quad \forall y, x,$$

$$(f''(x)p, p) \geqslant 0 \quad \forall p, x$$

эквивалентны выпуклости. В частности, если $f(x) = \frac{1}{2}(x, Cx) + (d, x)$, то f выпукла тогда и только тогда, когда $(p, Cp) \geqslant 0$, т.е. матрица C положительно определена.

Вообще говоря, выпуклая функция не является гладкой и не обладает непрерывными производными. Однако ее *производная по направлению*

$$f'(x, p) = \lim_{\lambda \downarrow 0} \frac{f(x + \lambda p) - f(x)}{\lambda}$$

существует и конечна в любой точке непрерывности. При этом разностное отношение $\lambda^{-1}[f(x + \lambda p) - f(x)]$, монотонно убывая, стремится к $f'(x, p)$.

Введём для выпуклой функции следующее определение. Вектор x^* называется *субградиентом* функции f в точке x, если

$$f(y) - f(x) \geqslant (y - x, x^*)$$

при всех y. Множество всех субградиентов обозначается через $\partial f(x)$ и называется *субдифференциалом*. Итак,

$$\partial f(x) = \{x^* \in \mathbf{R}^n:\ f(y) - f(x) \geqslant (y - x, x^*)\ \ \forall y\}. \qquad (2.3)$$

Если функция дифференцируема, то субдифференциал состоит из одного вектора $f'(x)$.

Субдифференциал и производные по направлению тесно связаны. Так, если f непрерывна в точке x, то

$$f'(x, p) = \max_{x^*} \{(p, x^*): \ x^* \in \partial f(x)\} \tag{2.4}$$

и $\partial f(x) \neq \phi$.

Будем говорить, что функция f *замкнута*, если ее надграфик есть замкнутое множество. Следующие три утверждения эквивалентны:

1) функция f замкнута;

2) множества уровня $\{x: \ f(x) \leqslant \alpha\}$ при любом α замкнуты;

3) функция f полунепрерывна снизу, т.е.

$$\lim_{y \to x} \inf f(y) \geqslant f(x).$$

Так же, как с выпуклым конусом тесно связан сопряженный конус, так и для выпуклой функции можно определить *сопряженную функцию*

$$f^*(x^*) = \sup_{x} \{(x, x^*) - f(x)\}. \tag{2.5}$$

Эта функция всегда выпукла и замкнута. Связь между исходной и сопряженной функциями дается следующей важной теоремой.

Т е о р е м а 2.1. *Пусть f – собственная выпуклая функция, полунепрерывная снизу в точке x. Тогда*

$$f(x) = f^{**}(x),$$

где

$$f^{**}(x) = \sup_{x^*} \{(x, x^*) - f^*(x^*)\}.$$

Эта теорема имеет многочисленные приложения. В частности, как будет видно в следующем пункте, она служит основой теории двойственности в выпуклом программировании.

3. Основы выпуклого программирования. Задача выпуклого программирования состоит в минимизации выпуклой функции на выпуклом множестве. В зависимости от формы представления выпуклого множества будут записываться и необходимые условия экстремума.

Для начала рассмотрим простейшую задачу. Пусть f – гладкая выпуклая функция, а M – выпуклое множество. Требуется охарактеризовать точку x_0, в которой f достигает своего минимума на M.

Пусть

$$K_M(x_0) = \{p: \ p = \lambda(x - x_0), \lambda > 0, x \in M\}.$$

Если $p \in K_M(x_0)$, то

$$x_0 + \alpha p = (1 - \alpha\lambda) \ x_0 + \alpha\lambda x \in M$$

при $\alpha \leqslant \lambda^{-1}$. Таким образом, при малом сдвиге из x_0 вдоль направления p

точка $x_0 + \alpha p$ не выходит за пределы множества M. Поэтому конус $K_M(x_0)$ носит название *конуса допустимых направлений.*

Т е о р е м а 2.2. *Для того чтобы точка x_0 была точкой минимума гладкой выпуклой функции f на выпуклом множестве M, необходимо и достаточно, чтобы $f'(x_0) \in K_M^*(x_0)$.*

Д о к а з а т е л ь с т в о. Пусть x_0 – точка минимума. Для любой точки $x \in M$ и $0 < \lambda \leqslant 1$

$$f((1-\lambda)x_0 + \lambda x) = f(x_0 + \lambda(x - x_0)) \geqslant f(x_0),$$

или

$$\frac{f(x_0 + \lambda p) - f(x_0)}{\lambda} \geqslant 0.$$

Переходя к пределу, получаем

$$f'(x_0, p) = (p, f'(x_0)) \geqslant 0, \quad p = x - x_0,$$

а значит,

$$(p, f'(x_0)) \geqslant 0, \quad p \in K_M(x_0). \tag{2.6}$$

Согласно определению сопряженного конуса последнее неравенство означает, что $f'(x_0) \in K_M^*(x_0)$.

Обратно, если (2.6) выполнено, то в силу выпуклости функции f

$$f(x) - f(x_0) \geqslant (x - x_0, f'(x_0)) \geqslant 0, \quad x \in M,$$

т.е. x_0 – точка минимума.

Рассмотрим случай, когда множество M задано системой линейных неравенств:

$$M = \{x: (a_i, x) \leqslant \alpha_i, \quad i = 1, \ldots, m\}. \tag{2.7}$$

Пусть

$$I(x) = \{i: (a_i, x) = \alpha_i, \quad i = 1, \ldots, m\}. \tag{2.8}$$

Нетрудно видеть, что только такие направления p, которые удовлетворяют неравенствам $(a_i, p) \leqslant 0$, $i \in I(x_0)$, обладают тем свойством, что при малом сдвиге из точки x_0 вдоль этого направления точка $x_0 + \alpha p$ остается в M. Поэтому

$$K_M(x_0) = \{p: (a_i, p) \leqslant 0, \quad i \in I(x_0)\}.$$

Используя приведенное в п. 1 утверждение о конусе, сопряженном к конусу, заданному системой линейных неравенств, получаем, что $y \in K_M^*(x_0)$ тогда и только тогда, когда

$$y = -\sum_{i \in I(x_0)} u^i a_i, \quad u^i \geqslant 0, \quad i \in I(x_0).$$

Итак, если x_0 – точка минимума гладкой выпуклой функции f на множестве M, заданном (2.7), то

$$f'(x_0) = \sum_{i \in I(x_0)} u^i a_i, \quad u^i \geqslant 0, \quad i \in I(x_0). \tag{2.9}$$

Придадим этому результату несколько другую форму.

Теорема 2.3. *Для того чтобы точка x_0 была точкой минимума гладкой выпуклой функции f при ограничениях*

$$(a_i, x) \leqslant \alpha_i, \quad i = 1, \ldots, m,$$

необходимо и достаточно, чтобы нашлись такие числа $u^i \geqslant 0$, $i = 1, \ldots, m$, что

$$f'(x_0) + \sum_{i=1}^{m} u^i a_i = 0,$$
$$u^i [(a_i, x) - \alpha_i] = 0, \quad i = 1, \ldots, m. \tag{2.10}$$

Доказательство теоремы сразу следует из предыдущего, если положить $u^i = 0$, $i \notin I(x_0)$, и вспомнить определение (2.8) множества $I(x_0)$.

Замечание. Этот результат легко обобщается на случай, когда в определение множества M входят равенства $(a_i, x) = \alpha_i$. Действительно, это равенство эквивалентно двум неравенствам $(a_i, x) \leqslant \alpha_i$, $(-a_i, x) \leqslant -\alpha_i$. Поэтому в (2.10) войдут векторы a_i, $-a_i$ соответственно с множителями $u^i_+ \geqslant 0$, $u^i_- \geqslant 0$. Собирая подобные члены и обозначая $u^i = u^i_+ - u^i_-$, получим то же соотношение (2.10), только на знак u^i уже не будет ограничения.

Используя это замечание, нетрудно получить необходимые и достаточные условия для следующей стандартной формы задачи.

Теорема 2.4. *Для того чтобы точка x_0 была точкой минимума гладкой выпуклой функции f при ограничениях*

$$(a_i, x) = \alpha_i, \quad i = 1, \ldots, m,$$
$$x^j \geqslant 0, \quad j = 1, \ldots, n,$$

необходимо и достаточно, чтобы нашлись такие числа u^i и вектор $v \in \mathbf{R}^n$ с компонентами v^j, что

$$f'(x_0) + \sum_{i=1}^{m} u^i a_i = v,$$
$$v^j \geqslant 0, \quad v^j x_0^j = 0, \quad j = 1, \ldots, n.$$

Доказательство получается непосредственным применением теоремы 2.3 и сделанного замечания, если учесть, что неравенство $x^j \geqslant 0$ эквивалентно неравенству $(-e_j, x) \leqslant 0$, где $e_j^* = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ – j-й единичный орт. При этом v^j – множитель, соответствующий этому ограничению.

Пусть теперь $f_i(x)$, $i = 0, 1, \ldots, m$, – выпуклые непрерывные функции, M – выпуклое множество. Рассмотрим задачу $P(0)$:

$$\min_x \{ f_0(x): \ f_i(x) \leqslant 0, \quad i = 1, \ldots, m, \ x \in M \}.$$

Для дальнейшего удобно эту задачу вложить в семейство задач $P(y)$,

зависящих от вектора параметров $y \in \mathbf{R}^m$. Положим

$$V(y) = \inf_x \{ f_0(x): f_i(x) \leqslant y^i, \quad i = 1, \ldots, m, x \in M \}.$$

Условимся, что $V(y) = +\infty$, если при данном y ограничения задачи несовместимы. Исходная задача $P(0)$ получается при $y = 0$.

Лемма 2.1. *Функция $V(y)$ выпукла. Если существует такая точка $x \in M$, что*

$$f_i(\bar{x}) < 0, \quad i = 1, \ldots, m,$$

и $V(0)$ – конечное число, то $V(y)$ непрерывна в окрестности точки $y = 0$ и $\partial V(0) \neq \phi$.

Доказательство. Пусть $\beta_j > V(y_j)$, $j = 1,2$. Тогда существуют такие точки $x_j \in M$, что

$$f_0(x_j) < \beta_j, \quad f_i(x_j) \leqslant y_j^i, \quad j = 1,2.$$

В силу выпуклости входящих в задачу функций справедливы неравенства

$$f_0(\lambda_1 x_1 + \lambda_2 x_2) = \lambda_1 f_0(x_1) + \lambda_2 f_0(x_2) < \lambda_1 \beta_1 + \lambda_2 \beta_2,$$

$$f_i(\lambda_1 x_1 + \lambda_2 x_2) \leqslant \lambda_1 y_1^i + \lambda_2 y_2^i, \quad i = 1, \ldots, m,$$

$$\lambda_1 x_1 + \lambda_2 x_2 \in M, \quad \lambda_1 + \lambda_2 = 1, \quad \lambda_1, \lambda_2 \geqslant 0.$$

Поэтому

$$V(\lambda_1 y_1 + \lambda_2 y_2) < \lambda_1 \beta_1 + \lambda_2 \beta_2,$$

откуда следует, что epi V есть выпуклое множество, и, значит, функция V выпукла.

Если существует указанная в лемме точка $\bar{x}$, то очевидно, что $f_i(\bar{x}) \leqslant y^i$, $i = 1, \ldots, m$, при малых y и поэтому $V(y) \leqslant f_0(\bar{x})$ и $0 \in \operatorname{int} \operatorname{dom} V$. Так как $V(0)$ – конечное число, то согласно сказанному в п. 2 V не может принимать значения $-\infty$.

Таким образом, V – выпуклая конечная в окрестности нуля функция, ограниченная в этой окрестности сверху числом $f_0(\bar{x})$. Согласно сказанному в п. 2 $V(y)$ непрерывна в окрестности 0 и $\partial V(0) \neq \phi$, что и требовалось доказать.

Вектор $u \in \mathbf{R}^m$ называется *вектором Куна – Таккера* задачи выпуклого программирования, если $u \geqslant 0$ и

$$V(0) = \inf_x \{ L(x, u): x \in M \},$$

где

$$L(x, u) = f_0(x) + \sum_{i=1}^{\infty} u^i f_i(x).$$

Функция L называется *функцией Лагранжа*.

Теорема 2.5. *Пусть $V(0)$ – конечное число. Вектор u является вектором Куна – Таккера тогда и только тогда, когда $-u \in \partial V(0)$.*

Доказательство. По определению субдифференциала $-u \in \partial V(0)$ тогда и только тогда, когда $V(y) \geqslant V(0) - (u, y)$ $\forall y$, т.е. когда

$$\inf_y \{ V(y) + (u, y) \} = V(0).$$

Подставляя сюда выражение для $V(y)$, получаем

$$\inf_y \inf_x \{f_0(x)+(u,y):\ f_i(x)\leqslant y^i,\quad i=1,\ldots,m,\quad x\in M\}=V(0).$$

Нетрудно заметить, что если для некоторого i соответствующая компонента u^i вектора u отрицательна, $u^i<0$, то, устремляя y^i к $+\infty$, в левой части приведенного неравенства получим $-\infty$, что противоречит конечности $V(0)$. Итак, $u\geqslant 0$, и в левой части минимум по y достигается, когда $y^i=f_i(x)$. Поэтому

$$\inf_x \{f_0(x)+\sum_{i=1}^m u^i f_i(x):\ x\in M\}=V(0),$$

т.е.

$$\inf_x \{L(x,u):\ x\in M\}=V(0)$$

и вектор u есть вектор Куна – Таккера. Теорема доказана.

Т е о р е м а 2.6. *Пусть существует точка $\bar{x}\in M$ такая, что $f_i(\bar{x})<0$ для $i=1,\ldots,m$ и $V(0)$ – конечное число. Тогда существует вектор Куна – Таккера. Если x_0 – решение задачи $P(0)$, то*

$$f_0(x_0)=L(x_0,u)\leqslant L(x,u),\quad x\in M,$$
$$u^i\geqslant 0,\ u^i f_i(x_0)=0,\quad i=1,\ldots,m. \tag{2.11}$$

Обратно, если точка x_0 удовлетворяет ограничениям задачи $P(0)$ и совместно с вектором u удовлетворяет условиям (2.11), то x_0 – решение задачи $P(0)$, а u – вектор Куна – Таккера.

Д о к а з а т е л ь с т в о. Существование вектора Куна – Таккера следует из леммы 2.1 и теоремы 2.5. Пусть теперь $x_0\in M$ – решение задачи $P(0)$. Тогда

$$f_i(x_0)\leqslant 0,\quad i=1,\ldots,m,$$
$$f_0(x_0)=V(0). \tag{2.12}$$

Так как $u\geqslant 0$, то

$$V(0)=f_0(x_0)\geqslant f_0(x_0)+\sum_{i=1}^m u^i f_i(x_0)=L(x_0,u).$$

Но по определению вектора Куна – Таккера $V(0)\leqslant L(x_0,u)$. Поэтому

$$f_0(x_0)=L(x_0,u)=V(0)\leqslant L(x,u),\ x\in M.$$

Кроме того, из $u\geqslant 0$, (2.12) и равенства

$$f_0(x_0)=L(x_0,u)=f_0(x_0)+\sum_{i=1}^m u^i f_i(x_0)$$

следует, что

$$u^i f_i(x_0)=0,\ i=1,\ldots,m.$$

Докажем оставшуюся часть теоремы. Пусть векторы u и x_0, удовлетворяющие ограничениям задачи $P(0)$, таковы, что выполняются соотношения

(2.11). Пусть x – любая точка, удовлетворяющая ограничениям задачи $P(0)$. Тогда из (2.11) следует, что

$$f_0(x_0) = L(x_0, u) \leqslant L(x, u) = f_0(x) + \sum_{i=1}^{m} u^i f_i(x) \leqslant f_0(x),$$

т.е. x_0 – решение задачи. Поэтому $V(0) = f_0(x_0)$ и из (2.11) вытекает, что u – вектор Куна – Таккера.

Т е о р е м а 2.7. *Пусть в условиях теоремы* 2.6 *функции* f_i *непрерывно дифференцируемы. Тогда для того чтобы точка* x_0 *была решением задачи* $P(0)$, *необходимо и достаточно существование такого вектора* $u \in \mathbf{R}^m$, *что выполнены соотношения*

$$L'_x(x_0, u) \in K^*_M(x_0), \tag{2.13}$$

$$u \geqslant 0, \quad u^i f_i(x_0) = 0, \quad i = 1, \ldots, m.$$

Д о к а з а т е л ь с т в о. В рассматриваемом гладком случае соотношения (2.13) полностью эквивалентны соотношениям (2.12). Действительно, первое из условий (2.12) утверждает, что точка x_0 является точкой минимума $L(x, u)$ по $x \in M$. Но согласно теореме 2.2 первое соотношение в (2.13) есть необходимое и достаточное условие для этого. Все остальные соотношения в (2.12) и (2.13) совпадают.

4. Двойственность в выпуклом программировании. Рассмотренной в предыдущем пункте задаче $P(0)$ можно поставить в соответствие двойственную задачу. Для того чтобы это сделать, вычислим функцию, сопряженную к функции $V(0)$, введенной в п. 3.

По определению

$$V^*(u) = \sup_y \{(u, y) - V(y)\} =$$

$$= \sup_y \{(u, y) - \inf_x \{f_0(x): f_i(x) \leqslant y^i, i = 1, \ldots, m, x \in M\}\} =$$

$$= \sup_x \sup_y \{-f_0(x) + \sum_{i=1}^{\infty} u^i y^i: f_i(x) \leqslant y^i, i = 1, \ldots, m, x \in M\}.$$

Легко вычислить верхнюю грань по y. Это даст

$$V^*(u) = \begin{cases} \sup\limits_x \{-f_0(x) + \sum\limits_{i=1}^{m} u_i f_i(x): x \in M\}, u \leqslant 0, \\ +\infty, \ u^i > 0 \ \text{для некоторого } i. \end{cases}$$

Иначе

$$V^*(u) = \begin{cases} -\inf\limits_x \{L(x, -u): x \in M\}, u \leqslant 0, \\ +\infty, \ u^i > 0 \ \text{для некоторого } i. \end{cases} \tag{2.14}$$

Т е о р е м а 2.8. *Если функция* $V(y)$ *полунепрерывна снизу при* $y = 0$, *в частности, если* $V(0)$ *конечна и существует такая точка* $\bar{x} \in M$, *что*

$f_i(\bar{x}) < 0, i = 1, \ldots, m$, то

$$V(0) = \sup_{u \geqslant 0} \inf_x \{ L(x, u): x \in M \}. \qquad (2.15)$$

Если существует вектор Куна – Таккера, то (2.15) *выполняется и на этом векторе достигается верхняя грань.*

Доказательство. Из теоремы 2.1 и (2.14) вытекает, что

$$V(0) = V^{**}(0) = \sup_u \{(u, 0) - V^*(u)\} = \sup_{u \leqslant 0} \{ \inf_x L(x, -u): x \in M \}.$$

Заменяя u на $-u$, получаем первое утверждение теоремы.

Пусть теперь $u \geqslant 0$ – произвольный вектор. Имеем

$$\inf_x \{ L(x, u): x \in M \} = \inf_x \{ f_0(x) + \sum_{i=1}^{m} u^i f_i(x): x \in M \} \leqslant$$

$$\leqslant \inf_x \{ f_0(x): f_i(x) \leqslant 0, i = 1, \ldots, m, x \in M \} = V(0),$$

Итак, для любого $u \geqslant 0$

$$\inf_x \{ L(x, u): x \in M \} \leqslant V(0).$$

Но если $u_0 \geqslant 0$ – вектор Куна – Таккера, то

$$V(0) = \inf_x \{ L(x, u_0): x \in M \}.$$

Это завершает доказательство.

Положим

$$\varphi(u) = \inf_x \{ L(x, u): x \in M \}.$$

Задача $\sup_{u \geqslant 0} \varphi(u)$ называется *двойственной задачей выпуклого программирования.*

Таким образом, теорема 2.8 утверждает, что при некоторых предположениях верхняя грань в двойственной задаче равна значению $V(0)$ исходной задачи, а если вектор Куна – Таккера существует, то на нем достигается верхняя грань в двойственной задаче.

Рассмотрим теперь вновь задачу минимизации гладкой выпуклой функции f при линейных ограничениях $(a_i, x) \leqslant \alpha_i$, $i = 1, \ldots, m$. Если x_0 – ее решение, то согласно теореме 2.3 существует вектор $u \geqslant 0$ такой, что выполняются соотношения (2.10). Но для данной задачи

$$L(x, u) = f(x) + \sum_{i=1}^{m} u^i |(a_i, x) - \alpha_i|, \qquad (2.16)$$

и поэтому соотношения (2.10) в точности совпадают с (2.13), если учесть, что в рассматриваемой задаче $M = \mathbf{R}^n$, а значит, $K_M(x_0) = \mathbf{R}^n$ и $K_M^*(x_0) = \{0\}$. Учитывая теперь эквивалентность (2.13) и (2.11), из теоремы 2.6 заключаем, что справедлив следующий результат.

Теорема 2.9. *Если в задаче минимизации гладкой выпуклой функции при линейных ограничениях минимум достигается, то существует вектор Куна – Таккера, справедлива формула* (2.15) *и вектор Куна – Таккера даст решение двойственной задачи.*

5. Необходимые условия экстремума. Общая задача. Откажемся теперь от условий выпуклости и рассмотрим общую задачу математического программирования. Для дальнейшего изложения достаточно рассмотреть только гладкие задачи.

Итак, пусть $f_0(x)$, $f_i(x)$, $i \in I$, – непрерывно дифференцируемые функции, а M – выпуклое множество. Рассмотрим задачу

$$\min_x \{f_0(x):\ f_i(x) \leqslant 0,\ i \in I^-,\ f_i(x) = 0,\ i \in I^0,\ x \in M\}, \tag{2.17}$$

где I – конечное множество индексов и $I^- \cup I^0 = I$, $I^- \cap I^0 = \phi$.

Пусть x_0 – решение задачи (2.17). Тогда справедлив следующий результат [23].

Т е о р е м а 2.10. *Для того чтобы точка x_0 была решением задачи* (2.17), *необходимо, чтобы нашлись такие, не равные нулю одновременно числа u^i, $i \in \{0\} \cup I$, что*

$$u^0 f_0'(x_0) + \sum_{i \in I} u^i f_i'(x_0) \in K_M^*(x_0). \tag{2.18_1}$$

$$u^i \geqslant 0,\ i \in \{0\} \cup I^-,\ u^i f_i(x_0) = 0,\ i \in I^- \tag{2.18_2}$$

Величины u^i, фигурирующие в соотношениях (2.18), называются *множителями Лагранжа.*

З а м е ч а н и е. Если $M = \mathbf{R}^n$, то $K_M^*(x_0) = \{0\}$, и поэтому первое из условий (2.18) превращается в равенство

$$u^0 f_0'(x_0) + \sum_{i \in I} u^i f_i'(x_0) = 0. \tag{2.19}$$

В условиях (2.18_1) присутствует множитель u^0 при градиенте функции $f_0(x)$. Обращение этого коэффициента в нуль означает вырожденность задачи, так как в необходимых условиях экстремума фактически не участвует минимизируемая функция. Поэтому разумно следующее определение, выделяющее регулярный случай. Точка минимума x_0 в задаче (2.17) *регулярна*, если число u^0, фигурирующее в (2.18_1), строго положительно.

З а м е ч а н и е. Как нетрудно заметить, в этом случае можно положить, что впредь мы и будем делать, $u^0 = 1$, поделив, если надо, все соотношения (2.18) на u^0.

Если точка минимума x_0 регулярна, то условие (2.18_1) может быть переписано в виде

$$L_x'(x_0, u) \in K_M^*(x_0), \tag{2.20}$$

где $L(x,u)$ – функция Лагранжа. Сравнивая (2.20), (2.18) и (2.11) получаем, что в теоремах 2.6 и 2.7 рассмотрен регулярный случай задач выпуклого программирования.

В общем случае условия регулярности, хотя они и чрезвычайно существенны при построении алгоритмов, достаточно трудно сформулировать в эффективно проверяемом виде. Как видно из предыдущего, если функции f_i выпуклы, $I^0 = \phi$, то таким условием служит существование точки $\bar{x} \in M$ такой, что $f_i(\bar{x}) < 0$ для $i \in I^-$. Если не предполагать выпуклости f_i, то можно сформулировать лишь существенно более слабое условие.

Пусть $M = \mathbf{R}^n$. Точка минимума x_0 задачи (2.17) называется *сильно регулярной*, если векторы $f_i'(x_0)$ линейно независимы, $i \in I^-(x_0) \cup I^0$, где $I^-(x_0) = \{ i \in I: f_i(x_0) = 0 \}$.

Если учесть (2.18$_2$), откуда следует, что $u^i = 0$ для $i \notin I^-(x_0)$, то в условиях сильной регулярности из (2.19) вытекает, что $u^0 > 0$, так как равенство $u^0 = 0$ означало бы линейную зависимость векторов $f_i'(x_0)$, $i \in I^-(x_0) \cup I^0$, что противоречит сильной регулярности.

Соотношение (2.19) можно переписать в виде

$$f_0'(x_0) + \sum_{i \in I^-(x_0) \cup I^0} u^i f_i'(x_0) = 0.$$

Поскольку векторы $f_i'(x_0)$, $i \in I^-(x_0) \cup I^0$, линейно независимы, то последнее равенство определяет u^i однозначно. Поэтому справедлива следующая теорема.

Теорема 2.11. *Пусть точка минимума x_0 задачи* (2.17) *сильно регулярна. Тогда она регулярна, т.е. u^0 в* (2.18) *можно положить равным* 1, *а все остальные величины u^i определены однозначно.*

6. Необходимые условия экстремума второго порядка. Доказательства результатов, приведенных в предыдущем пункте, используют вариации аргументов первого порядка малости, и поэтому в формулировку теорем входят лишь первые производные от исходных функций. Изучение вариаций более высокого порядка приводит к необходимым условиям, включающим вторые производные. Для дальнейшего изложения будет необходим следующий результат.

Пусть $M = \mathbf{R}^n$, функции f_i дважды непрерывно дифференцируемы.

Теорема 2.12. *Пусть точка минимума x_0 в задаче* (2.17) *сильно регулярна и $u^i > 0$, $i \in I^-(x_0)$. Тогда*

$$(L''_{xx}(x_0, u)p, p) \geqslant 0 \qquad (2.21)$$

для всех $p \in \mathbf{R}^n$, удовлетворяющих соотношениям

$$f_i'(x_0)p = 0, \quad i \in I^-(x_0) \cup I^0.$$

7. Задача о минимаксе. Пусть требуется найти точку минимума функции

$$f_0(x) = \max_{k \in I} \varphi_k(x), \qquad (2.22)$$

где I — конечное множество индексов, а функции φ_k непрерывно дифференцируемы. При этом x меняется во всем пространстве $\mathbf{R}^n$. Эта задача легко сводится к общей задаче математического программирования введением дополнительной переменной x^0. А именно, задача минимизации функции $f(x)$, задаваемой формулой (2.22), эквивалентна задаче

$$\min_{\{x,\, x^0\}} \{ x^0: \varphi_k(x) \leqslant x^0, \quad k \in I \}. \qquad (2.23)$$

Применение к этой задаче теоремы 2.10 (при $M = \mathbf{R}^{n+1}$) приводит к следующему результату.

Теорема 2.13. *Для того чтобы точка x_0 была точкой минимума функции $f_0(x)$, определенной соотношением* (2.22), *необходимо, чтобы*

нашлись такие числа u^k, $k \in I$, *что*

$$\sum_{k \in I} u^k \varphi_k'(x_0) = 0,$$

$$u^k \geqslant 0, \quad u^k (\varphi_k(x_0) \quad f_0(x_0)) = 0,$$

$$\sum_{k \in I} u^k = 1.$$

Легко видеть, что если имеются ограничения на x в виде ограничений (2.17), задаваемых равенствами и неравенствами, при которых требуется минимизировать функцию $f_0(x)$, то она подобным же образом может быть сведена к общей задаче математического программирования. Однако при этом она уже не обладает какими-либо специфическими особенностями, выделяющими ее из общей задачи, и поэтому ограничимся здесь лишь констатацией этого факта.

8. **Метод штрафных функций.** Метод штрафных функций достаточно популярен при решении задач математического программирования. Он сводит эту задачу к некоторой задаче безусловной оптимизации. Методу штрафных функций посвящена обширная литература, к которой мы и отсылаем читателя, желающего более подробно с ним познакомиться. Здесь будут изложены лишь некоторые факты, непосредственно относящиеся к методу линеаризации.

Пусть $f_0, f_1, \ldots, f_m$ – непрерывные функции, M – некоторое множество. Положим

$$V(\nu) = \inf_x \{ f_0(x) : f_i(x) \leqslant \nu^i, \ i = 1, \ldots, m, \ x \in M \}. \tag{2.24}$$

Задачу минимизации, стоящую в правой части соотношения (2.24), будем обозначать $P(\nu)$. Естественно, что в первую очередь нас будет интересовать задача $P(0)$ – как исходная.

Заметим также, что рассмотрение в (2.24) лишь ограничений типа неравенств не уменьшает общности, так как каждое ограничение типа равенства $f(x) = 0$ может быть записано в виде двух неравенств: $f(x) \leqslant 0$, $-f(x) \leqslant 0$.

Положим

$$F(x) = \max \{ 0, f_1(x), \ldots, f_m(x) \},$$

$$\Phi_N(x) = f_0(x) + NF(x).$$

Совершенно очевидна следующая цепочка равенств:

$$\inf_{x \in M} \Phi_N(x) = \inf_{x, \lambda} \{ f_0(x) + N\lambda : 0 \leqslant \lambda, f_1(x) \leqslant \lambda, \ldots, f_m(x) \leqslant \lambda, x \in M \} =$$

$$= \inf_{\lambda \geqslant 0} [V(\lambda \cdot 1) + N\lambda], \quad 1 = \begin{bmatrix} 1 \\ \ldots \\ 1 \end{bmatrix} \in \mathbf{R}^m. \tag{2.25}$$

Отсюда следует, что если нижняя грань в правой части (2.25) достигается только при $\lambda = 0$, т.е.

$$V(\lambda \cdot 1) - V(0) > \quad N\lambda, \ \lambda > 0, \tag{2.26}$$

то

$$V(0) = \inf_x \{\Phi_N(x) : x \in M\}. \tag{2.27}$$

Таким образом, если (2.26) выполнено, то нижняя грань в задаче $P(0)$ совпадает с нижней гранью функции $\Phi_N(x)$ при $x \in M$. Покажем, что справедлив более сильный факт, а именно, точки, в которых достигается минимум в задачах $P(0)$, и точки минимума $\Phi_N(x)$ совпадают.

Т е о р е м а 2.14. *Пусть*

$$\inf_{\lambda > 0} \frac{V(\lambda \cdot 1) - V(0)}{\lambda} = -l > -\infty \tag{2.28}$$

и $N > l$. Тогда точки минимума задачи $P(0)$ и задачи $\inf_x \{\Phi_N(x): x \in M\}$ совпадают.

З а м е ч а н и е. Так как $V(\lambda \cdot 1)$, очевидно, есть убывающая функция λ (нижняя грань по более узкой области больше, чем нижняя грань в более широкой), то $l \geqslant 0$.

Д о к а з а т е л ь с т в о. Из (2.28) следует, что $V(\lambda \cdot 1) + N\lambda > V(0)$ при $\lambda > 0$.

Пусть $x_0 \in M$ и $\Phi_N(x_0) = \inf_x \{\Phi_N(x): x \in M\}$. Далее,

$$\inf_x \{f_0(x) + NF(x): x \in M\} \leqslant \inf_x \{f_0(x): f_i(x) \leqslant 0,\ i = 1, \ldots, m,\ x \in M\},$$

т.е.

$$V(0) \geqslant \inf_x \{\Phi_N(x): x \in M\}. \tag{2.29}$$

Покажем, что $F(x_0) = 0$. Допустим противное, т.е. допустим, что $\lambda_0 = F(x_0) > 0$. Имеем

$$f_0(x_0) + NF(x_0) = \min_x \{f_0(x) + NF(x): x \in M\} \leqslant$$

$$\leqslant \min_x \{f_0(x) + NF(x): x \in M,\ F(x) \leqslant \lambda_0\} \leqslant$$

$$\leqslant \min_x \{f_0(x) + N\lambda_0: x \in M,\ F(x) \leqslant \lambda_0\} = V(\lambda_0 \cdot 1) + N\lambda_0.$$

Но, с другой стороны,

$$f_0(x_0) + NF(x_0) = f_0(x_0) + N\lambda_0 \geqslant \min_x \{f_0(x) + N\lambda_0: x \in M,$$

$$F(x) \leqslant \lambda_0\} \geqslant V(\lambda_0 \cdot 1) + N\lambda_0.$$

Таким образом,

$$f_0(x_0) + NF(x_0) = V(\lambda_0 \cdot 1) + N\lambda_0.$$

Отсюда, из (2.28), (2.29) и того, что $N > l$, получаем

$$V(0) \geqslant f_0(x_0) + NF(x_0) = V(\lambda_0 \cdot 1) + N\lambda_0 > V(0),$$

так как $\lambda_0 = F(x_0) > 0$. Получено противоречие.

Итак, $F(x_0) = 0$, откуда следует, что точка $x_0 \in M$ удовлетворяет всем ограничениям задачи $P(0)$. Поэтому $f_0(x_0) \geqslant V(0)$. Но согласно (2.29)

$V(0) \geqslant f_0(x_0) + NF(x_0) = f_0(x_0)$. Окончательно: $f_0(x_0) = V(0)$, т.е. x_0 – точка минимума задачи $P(0)$.

Обратно, пусть x_0 – решение задачи $P(0)$. Тогда

$$f_0(x_0) + NF(x_0) = f_0(x_0) = V(0) \leqslant$$

$$\leqslant \inf_{\lambda \geqslant 0} [V(\lambda \cdot 1) + N\lambda] = \inf_x \{\Phi_N(x): x \in M\},$$

т.е. x_0 минимизирует $\Phi_N(x)$ на M.

Из полученного результата можно извлечь ряд следствий.

Следствие 1. *Пусть функции $f_i(x)$, $i = 0, 1, \ldots, m$, и множество M выпуклы и существует вектор Куна – Таккера u. Тогда при*

$$N > \sum_{i=1}^{m} u^i$$

решения задачи $P(0)$ и точки минимума $\Phi_N(x)$ на M совпадают.

Доказательство. Поскольку рассматривается задача выпуклого программирования, то $V(y)$ – выпуклая функция и согласно теореме 2.5 $-u \in \partial V(0)$. Поэтому

$$V(\lambda \cdot 1) - V(0) \geqslant -(u, \lambda \ 1) = -\lambda \sum_{i=1}^{m} u^i,$$

$$\inf_{\lambda \geqslant 0} \frac{V(\lambda \cdot 1) - V(0)}{\lambda} \geqslant -\sum_{i=1}^{m} u^i,$$

и доказываемое утверждение вытекает из теоремы.

Следствие 2. *Пусть в общей задаче $M \in \mathbf{R}^n$, все функции дифференцируемы и выполнено соотношение* (2.28). *Тогда любая точка минимума регулярна.*

Доказательство. На основании теоремы 2.14 можно заключить, что если x_0 – точка минимума задачи $P(0)$, то она доставляет минимум функции $\Phi_N(x)$ при достаточно большом N. Но

$$\Phi_N(x) = \max\{\varphi_i(x): i = 0, 1, \ldots, m\},$$

где

$$\varphi_0(x) = f_0(x), \ \varphi_i(x) = f_0(x) + Nf_i(x), \ i = 1, \ldots, m.$$

Таким образом, задача минимизации $\Phi_N(x)$ есть минимаксная задача и к ней можно применить теорему 2.13. Значит, существуют такие числа $u^i \geqslant 0$, $i = 0, 1, \ldots, m$, что

$$u^0 f_0'(x_0) + \sum_{i=1}^{m} u^i (f_0'(x_0) + Nf_i'(x_0)) = 0,$$

$$u^0 (f_0(x_0) - \Phi_N(x_0)) = 0,$$

$$u^i [f_0(x_0) + Nf_i(x_0) - \Phi_N(x_0)] = 0, \ i = 1, \ldots, m,$$

$$u^0 + \sum_{i=1}^{m} u^i = 1.$$

Но

$$\Phi_N(x_0) = f_0(x_0) + NF(x_0) = f_0(x_0),$$

так как x_0 – решение задачи $P(0)$. Поэтому, если обозначить $\bar{u}^i = Nu^i$, $i = 1, \ldots, m$, то предыдущие соотношения можно переписать в виде

$$f_0'(x_0) + \sum_{i=1}^{m} \bar{u}^i f_i'(x_0) = 0,$$

$$\bar{u}^i f_i(x_0) = 0, \quad \bar{u}^i = 0, \quad i = 1, \ldots, m.$$

Это обычные необходимые условия минимума, в которых $\bar{u}^0 = 1$, что соответствует регулярному случаю.

§ 3. ЗАДАЧА КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

Задача квадратичного программирования состоит в нахождении минимума квадратичной функции с положительно определенной матрицей при линейных ограничениях на аргумент. Представляя интерес и сама по себе, в частности потому, что она является обобщением задачи линейного программирования, в контексте этой книги она важна, так как является основной вспомогательной задачей при решении общих нелинейных проблем. Практика расчетов показала, что от того, насколько эффективен используемый алгоритм решения задачи квадратичного программирования, зависит и эффективность всего алгоритма нелинейного программирования в целом.

Существует много различных алгоритмов квадратичного программирования. Для дальнейшего будут важны в первую очередь конечные алгоритмы, т.е. такие, которые дают решение за конечное число шагов. Это обусловлено тем, что по ходу общего алгоритма линеаризации приходится решать большое число задач квадратичного программирования, и поэтому нельзя допускать внутри общего процесса процедур с бесконечной длительностью.

1. **Метод сопряженных направлений.** Начнем с простейшей задачи, т.е. задачи минимизации функции

$$f(x) = \tfrac{1}{2}(x, Cx) + (d, x), \tag{3.1}$$

где C – симметричная положительно определенная матрица, т.е. $(p, Cp) \geqslant 0$ при всех $p \in \mathbf{R}^n$.

Векторы p_i, $i = 1, \ldots, n$, называются *сопряженными относительно матрицы* C, если они линейно независимы и

$$(p_i, Cp_j) = 0, \quad i \neq j. \tag{3.2}$$

Сопряженные векторы существуют всегда, хотя определены неоднозначно. Из линейной алгебры известно, что симметричная матрица имеет n ортогональных собственных векторов s_i, соответствующих собственным числам λ_i, т.е.

$$Cs_i = \lambda_i s_i, \quad i = 1, \ldots, n,$$

$$(s_i, s_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Поэтому

$$(s_i, C s_j) = \lambda_j (s_i, s_j) = \begin{cases} \lambda_j, & i = j, \\ 0, & i \neq j, \end{cases}$$

так что векторы s_i являются сопряженными. Так как для положительно определенной матрицы $\lambda_j \geqslant 0$, но, возможно, $\lambda_i = 0$ при некотором i, то, вообще говоря, число $(s_i, C s_i)$ может быть равным нулю.

Знание сопряженной системы векторов позволяет легко решить задачу минимизации функции (3.1). В самом деле, пусть x_0 – произвольная точка. Любая другая точка x может быть однозначно представлена в виде

$$x = x_0 + \sum_{i=1}^{n} \alpha_i p_i.$$

Подставив это выражение в (3.1), получим

$$f(x) = f(x_0) + \frac{1}{2} \sum_{i=1}^{n} \alpha_i^2 (p_i, C p_i) + \sum_{i=1}^{n} \alpha_i (p_i, f'(x_0)). \tag{3.3}$$

где учтено, что $f'(x) = Cx + d$. Теперь легко найти минимум правой части (3.3) по α_i.

Если $(p_i, C p_i) = 0$, то в (3.3) отсутствует квадратичный член, и если $(p_i, f'(x_0)) \neq 0$, то нижняя грань по α_i равна $-\infty$. Если же $(p_i, f'(x_0)) = 0$, то α_i может быть выбрано произвольно, так как f от α_i фактически не зависит.

Заметим теперь, что x_0 – произвольная точка. Поэтому можно сделать следующий вывод: для того чтобы функция $f(x)$ (3.1) была ограничена снизу, необходимо и достаточно, чтобы выполнялось соотношение

$$(p_i, f'(x)) = 0$$

для всех x и всех i, для которых $(p_i, C p_i) = 0$.

Если $(p_i, C p_i) \neq 0$, то минимизация (3.3) по α_i дает

$$\alpha_i = -(p_i, f'(x_0)) / (p_i, C p_i). \tag{3.4}$$

Из только что сказанного вытекает, что если C – положительно определенная матрица, то квадратичная функция f либо неограничена снизу, либо достигает минимума в некоторой точке.

Если минимум достигается, то точка минимума имеет вид

$$x_* = x_0 + \sum_{i=1}^{n} \alpha_i p_i,$$

причем α_i задаются формулой (3.4), если $(p_i, C p_i) \neq 0$, и произвольны в противном случае. Для определенности будем полагать в этом случае $\alpha_i = 0$.

Положим теперь

$$x_{i+1} = x_i + \alpha_{i+1} p_{i+1}, \quad i = 0, 1, \ldots, n-1.$$

Заметим, что

$$f'(x_i) = f'(x_i) - f'(x_{i-1}) + f'(x_{i-1}) - f'(x_{i-2}) + \ldots$$

$$\ldots + f'(x_1) - f'(x_0) + f'(x_0) =$$

$$= \alpha_i C p_i + \alpha_{i-1} C p_{i-1} + \ldots + \alpha_1 C p_1 + f'(x_0).$$

Поэтому

$$(p_{i+1},\ f'(x_i)) = (p_{i+1},\ f'(x_0)).$$

Отсюда следует, что точки x_i, $i = 1, \ldots, n$, могут быть получены по рекуррентным формулам

$$x_{i+1} = x_i + \alpha_{i+1} p_{i+1},$$

$$\alpha_{i+1} = -(p_{i+1},\ f'(x_i))/(p_{i+1},\ Cp_{i+1}),\quad i = 0, 1, \ldots, n-1,$$

и $x_n = x_*$ – решение задачи.

Если по ходу процесса оказывается, что $(p_i,\ Cp_i) = 0$, но $(p_i,\ f'(x_{i-1})) \neq 0$, то отсюда сразу следует неограниченность снизу функции $f(x)$.

До сих пор оставался открытым вопрос о методах построения сопряженных направлений. Этот вопрос может решаться многими способами, в зависимости от которых возникают различные алгоритмы. По этому поводу читатель может ознакомиться с литературой, указанной в библиографии. Здесь приводится лишь один алгоритм, по-видимому, наиболее простой и удобный для минимизации квадратичной функции.

2. Алгоритм метода сопряженных направлений. Здесь приводится одна из форм алгоритма метода сопряженных направлений, достаточно удобная для минимизации квадратичной функции.

Начальная точка x_0 выбирается произвольно. Все остальные вычисления при $i = 0, 1, \ldots, n-1$ производятся по рекуррентным формулам:

$$\beta_{i+1} = \begin{cases} 0, & i = 0, \\ \dfrac{(f'(x_i),\ f'(x))}{(f'(x_{i-1}),\ f'(x_{i-1}))}, & i > 0, \end{cases}$$

$$p_{i+1} = \begin{cases} f'(x_i), & i = 0, \\ f'(x_i) + \beta_{i+1} p_i, & i > 0, \end{cases}$$

$$\alpha_{i+1} = -\frac{(p_{i+1},\ f'(x_i))}{(p_{i+1},\ Cp_{i+1})},$$

$$x_{i+1} = x_i + \alpha_{i+1} p_{i+1}.$$

Комментарии. 1. Как видно из изложенного в п. 1, знание вектора p_{i+1} необходимо лишь после того, как построена точка x_i. Приведенный выше процесс как раз и строит сопряженные векторы по мере возникновения надобности в них.

2. Условием останова процесса является обращение очередного градиента $f'(x_i)$ в нуль. Это обязательно произойдет при $i = n$, но может случиться и ранее, если удачно выбрана точка x_0 или часть собственных чисел матрицы C равна нулю.

3. Ограниченность снизу квадратичной функции $f(x)$ априори неизвестна. Поэтому при вычислении очередной величины α_{i+1} может возникнуть ситуация, когда

$$(p_{i+1},\ Cp_{i+1}) = 0,\quad (p_{i+1},\ f'(x_i)) \neq 0.$$

Появление такой ситуации как раз и характеризует неограниченность снизу функции $f(x)$.

4. Вычислительный опыт показывает, что, когда матрица C плохо обусловлена, т.е. когда она невырождена, но отношение максимального собственного числа к минимальному очень велико, приведенный процесс накапливает погрешность вычислений. Поэтому, если $f'(x_n)$ не равно нулю с достаточной точностью, необходимо повторить применение алгоритма, беря точку x_n за начальную, т.е. использовать приведенный алгоритм как один шаг более общего итерационного процесса.

3. **Существование решения.** Обратимся теперь к общей задаче квадратичного программирования. Это есть задача минимизации квадратичной функции

$$f(x) = \tfrac{1}{2}(x, Cx) + (d, x)$$

при ограничениях

$$\begin{aligned} &(a_i, x) \leqslant \alpha_i, \quad i \in I^-, \\ &(a_i, x) = \alpha_i, \quad i \in I^0, \end{aligned} \tag{3.5}$$

где матрица C положительно определена, а I^-, I^0 — конечные множества индексов. В этом пункте нас будет интересовать вопрос о разрешимости задачи квадратичного программирования.

Заметим, что в ограничениях (3.5) равенства можно исключить, если из этих равенств часть неизвестных выразить через остальные и подставить в оставшиеся неравенства и выражение для функции f. Это приведет к сокращению числа неизвестных и к присутствию только неравенств в ограничениях задачи. Итак, по крайней мере для целей, преследуемых в этом пункте, можно считать, что задача имеет вид

$$\min \{f(x): (a_i, x) \leqslant \alpha_i, \ i = 1, \ldots, m\}. \tag{3.6}$$

Т е о р е м а 3.1. *В задаче квадратичного программирования минимум либо достигается в некоторой точке, либо функция f неограничена снизу.*

Д о к а з а т е л ь с т в о. Используем индукцию по числу неравенств m. При $m = 0$ ограничения отсутствуют и теорема справедлива в силу сказанного в п. 1. Допустим, что она справедлива для m неравенств, и покажем ее справедливость для $m + 1$ неравенства.

Введем обозначения:

$$v_{m+1} = \inf_x \{f(x): (a_i, x) \leqslant \alpha_i, \ i = 1, \ldots, m+1\},$$

$$v_m = \inf_x \{f(x): (a_i, x) \leqslant \alpha_i, \ i = 1, \ldots, m\},$$

$$W = \inf_x \{f(x): (a_i, x) \leqslant \alpha_i, \ i = 1, \ldots, m, (a_{m+1}, x) = \alpha_{m+1}\}.$$

Задачи, стоящие в правых частях этих формул, обозначим P_{m+1}, P_m, P_0 соответственно. Очевидно, что $v_m \leqslant v_{m+1} \leqslant W$. Далее, в задаче P_0 минимум достигается или равен $-\infty$, так как эта задача содержит фактически только m неравенств.

Если $W = -\infty$, то $v_{m+1} = -\infty$ и все доказано. Предположим поэтому, что W конечно, а x — точка минимума. Далее, если $v_{m+1} = -\infty$, то также все доказано. Поэтому необходимо считать, что v_{m+1} конечно, и доказать, что

найдется точка z, удовлетворяющая ограничениям задачи P_{m+1} и такая, что $f(z) = v_{m+1}$.

Теперь возможны два случая.

а) v_m конечно, и минимум достигается в точке $\bar{z}$. Если

$$(a_{m+1}, \bar{z}) \leqslant \alpha_{m+1},$$

то $\bar{z}$ доставляет минимум и в P_{m+1} и все доказано.

Пусть

$$(a_{m+1}, \bar{z}) > \alpha_{m+1}. \tag{3.7}$$

Возьмем любую точку x, удовлетворяющую ограничениям задачи P_{m+1}, и поставим ей в соответствие точку

$$\bar{x} = x + t(\bar{z} - x), \quad 0 \leqslant t \leqslant 1,$$

где t выбрано так, что $(a_{m+1}, \bar{x}) = \alpha_{m+1}$. Так как

$$(a_{m+1}, x) \leqslant \alpha_{m+1}$$

и выполнено (3.7), то такое t найдется.

Очевидно, что $\bar{x}$ удовлетворяет ограничениям задачи P_0 (ведь x удовлетворяет P_{m+1}, а $\bar{z} \in P_m$), а так как f – выпуклая функция, то

$$f(\bar{x}) \leqslant (1-t) f(x) + t f(\bar{z}) \leqslant f(x),$$

ибо $f(\bar{z}) = v_m \leqslant v_{m+1} \leqslant f(x)$. Таким образом, если y – решение задачи P_0, то y удовлетворяет ограничениям задачи P_{m+1} и $f(y) \leqslant f(\bar{x}) \leqslant f(x)$, т.е. y – решение задачи P_{m+1} и минимум в P_{m+1} достигается.

б) $v_m = -\infty$. Пусть x_k, $k = 1, 2, \ldots$, – минимизирующая последовательность для задачи P_m, т.е. $f(x_k) \to -\infty$. Если среди точек последовательности x_k как угодно много удовлетворяют ограничениям задачи P_{m+1}, то $v_{m+1} = -\infty$. Поэтому допустим, что для достаточно больших k выполняется неравенство

$$(a_{m+1}, x_k) > \alpha_{m+1}. \tag{3.8}$$

Если x – допустимая точка для задачи P_{m+1}, то выберем теперь настолько большое k, что удовлетворяется (3.8) и $f(x) \geqslant f(x_k)$, и поставим точке x в соответствие точку

$$\bar{x} = x + t(x_k - x), \quad 0 \leqslant t \leqslant 1,$$

такую, что

$$(a_{m+1}, \bar{x}) = \alpha_{m+1}.$$

Теперь вновь $\bar{x}$ – допустимая точка для задачи P_0 и

$$f(\bar{x}) \leqslant (1-t) f(x) + t f(x_k) \leqslant f(x).$$

Отсюда, как и ранее, заключаем, что решение задачи P_0 – точка y – является и решением задачи P_{m+1}. Теорема доказана.

4. Необходимые условия экстремума и двойственная задача. Результаты § 2 позволяют легко выписать необходимые условия экстремума в задаче квадратичного программирования. В самом деле, применение теоремы 2.3 и сделанного к ней замечания приводит к следующему результату.

Т е о р е м а 3.2. *Пусть x_0 – точка минимума задачи квадратичного программирования с ограничениями* (3.5). *Тогда существуют такие числа* u^i, $i \in I = I \cup I^0$, *что*

$$f'(x_0) + \sum_{i \in I} u^i a_i = 0,$$

$$u^i \geqslant 0, \quad u^i |(a_i, x_0) - \alpha_i| = 0, \quad i \in I^-$$

Для построения двойственной задачи запишем функцию Лагранжа

$$L(x, u) = \tfrac{1}{2}(x, Cx) + (x, d) + \sum_{i \in I} u^i |(a_i, x) - \alpha_i| =$$

$$= \tfrac{1}{2}(x, Cx) + (x, d + \sum_{i \in I} u^i a_i) - \sum_{i \in I} u^i \alpha_i =$$

$$= \tfrac{1}{2}(x, Cx) + (x, d + A^* u) - (b, u),$$

где b – вектор с компонентами α_i, $i \in I$, u – вектор с компонентами u^i, $i \in I$, A – матрица, строками которой служат векторы a_i^*, $i \in I$. Напомним, что a_i – вектор-столбцы, так что a_i^* – вектор-строки. Чтобы построить двойственную задачу согласно п. 4 § 2, необходимо вычислить

$$\varphi(u) = \inf_x L(x, u).$$

Допустим, что матрица C невырождена. Приравнивая производные от L по x нулю, получаем

$$L'_x(x, u) = Cx + d + A^* u = 0,$$
$$x = -C^{-1}[d + A^* u]. \tag{3.9}$$

Подставляя это в выражение для $L(x, u)$, имеем

$$\varphi(u) = \tfrac{1}{2}(C^{-1}[d + A^* u], [d + A^* u]) -$$

$$-(C^{-1}[d + A^* u], d + A^* u) - (b, u) =$$

$$= -\tfrac{1}{2}(C^{-1}[d + A^* u], d + A^* u) - (b, u).$$

Итак, если C – положительно определенная невырожденная матрица, то согласно п. 4 § 2 двойственная задача квадратичного программирования состоит в максимизации квадратичной функции

$$\varphi(u) = -\tfrac{1}{2}(C^{-1}[d + A^* u], d + A^* u) - (b, u) \tag{3.10}$$

при ограничениях $u^i \geqslant 0$, $i \in I^-$.

При этом учтено, что ограничения (3.5), соответствующие $i \in I^0$, могут быть расписаны как два неравенства, что согласно замечанию к теореме 2.3 приводит к неопределенности знака соответствующего множителя u^i.

Как было показано в п. 3, минимум в задаче квадратичного программирования всегда достигается, и поэтому применима теорема 2.9. Из этой теоремы следует, что существует вектор Куна – Таккера и он является решением двойственной задачи.

Если теперь решение u задачи (3.10) найдено, то, подставляя его в (3.9), получим решение исходной задачи. В самом деле, так как C положительно определена и невырождена, то нетрудно показать, что исходная задача имеет единственное решение x_0. Согласно теореме 2.6 точка x_0 должна достав-

лять минимум $L(x, u)$, если u – вектор Куна – Таккера, т.е. решение задачи (3.10). Но этот минимум единственный и доставляется выражением, даваемым формулой (3.9).

Т е о р е м а 3.3. *Пусть C – положительно определенная невырожденная матрица. Тогда задача максимизации функции* (3.10) *при ограничениях $u^i \geqslant 0$, $i \in I^-$, является двойственной задачей квадратичного программирования. Если u – ее решение, то $x_0 = -C^{-1}[d + A^* u]$ является решением исходной задачи.*

Применим эту теорему к частному случаю, когда нет ограничений типа неравенств, т.е. когда $I^- = \phi$. Тогда согласно введенным обозначениям ограничения (3.5), соответствующие $i \in I^0$, можно записать в виде

$$Ax \quad b = 0. \tag{3.11}$$

Согласно теореме 3.3 решение задачи минимизации функции $f(x)$, задаваемой формулой (3.1), при ограничениях (3.11) имеет вид (3.9), причем на знак компонент u нет ограничений. Подставляя формулу (3.9) в (3.11), получаем

$$AC^{-1}[d + A^* u] + b = 0. \tag{3.12}$$

Заметим теперь, что если строки матрицы A, т.е. a_i^*, $i \in I^0$, линейно независимы, то матрица $AC^{-1}A^*$ обратима. В самом деле, эта матрица необратима, если она отображает некоторый ненулевой вектор v в нуль:

$$AC^{-1}A^* v = 0.$$

Но тогда

$$v^* AC^{-1} A^* v = (C^{-1}(A^* v), A^* v) = 0,$$

что возможно лишь, если $A^* v = 0$, т.е. если строки матрицы A, являющиеся столбцами матрицы A^*, линейно зависимы, что противоречит предположению. В этом предположении из (3.12) и (3.9) получаем

$$u = -(AC^{-1}A^*)^{-1}[b + AC^{-1} d], \tag{3.13}$$

$$x_0 = -C^{-1}[d + A^* u]. \tag{3.14}$$

Таким образом, доказана следующая теорема.

Т е о р е м а 3.4. *Пусть в задаче минимизации функции* (3.1) *при ограничениях* (3.11) *матрица C невырождена, а строки матрицы A линейно независимы. Тогда решение прямой и двойственной задачи дается формулами* (3.13), (3.14).

5. Приложение. Проектирование на подпространство. К задаче квадратичного программирования сводится задача проектирования вектора на подпространство. По определению *проекцией* вектора y на подпространство $M = \{x: Ax = 0\}$ называется точка $x_0 \in M$, расстояние которой от y минимально по евклидовой норме. Поэтому построение проекции сводится к нахождению решения задачи

$$\min_x \{ \tfrac{1}{2} \| x - y \|^2 : Ax = 0 \}.$$

Учитывая, что

$$\tfrac{1}{2}\| x - y \|^2 = \tfrac{1}{2}(x, x) - (x, y) + \tfrac{1}{2}(y, y),$$

проекцию x_0 можно получить, применяя теорему 3.4, причем в формулах (3.13), (3.14) теперь надо положить $C = I$ – единичная матрица, $b = 0$, $d = -y$.

Таким образом,

$$u = (AA^*)^{-1} Ay, \quad x_0 = y - A^*(AA^*)^{-1} Ay.$$

Обозначим

$$P = A^*(AA^*)^{-1} A. \tag{3.15}$$

Теорема 3.5. *Пусть* $M = \{x: Ax = 0\}$ *и строки матрицы* A *линейно независимы. Тогда проекция вектора* y *на подпространство* M *дается формулой*

$$x = (I - P)y.$$

Приведем некоторые свойства матрицы P, сразу следующие из ее определения (3.15):

а) $PA^* = A^*$;

б) $P^2 = P$, $P(I - P) = 0$, $P^* = P$;

в) $A(I - P) = 0$.

Из свойства б) получаем, что для любого вектора y

$$y = Py + (I - P)y,$$

$$(Py, (I - P)y) = (y, P(I - P)y) = 0.$$

Таким образом, каждый вектор y разложим в сумму двух ортогональных векторов, один из которых есть проекция y на M, а другой принадлежит ортогональному дополнению M – подпространству $M^\perp$. В самом деле, для любого $x \in M$ в силу б)

$$(Py, x) = (y, Px) = 0,$$

что и означает, что $Py \in M^\perp$.

Если теперь y – произвольный вектор, $x \in M^\perp$, то справедлива цепочка равенств

$$\| y - x \|^2 = \| Py - x + (I - P)y \|^2 = \| Py - x \|^2 +$$

$$+ 2(Py - x, (I - P)y) + \| (I - P)y \|^2.$$

Но согласно свойству б) и тому, что $x \in M^\perp$, $(I - P)y \in M$, справедливо

$$(Py, (I - P)y) = 0, \quad (x, (I - P)y) = 0,$$

так что

$$\| y - x \|^2 = \| Py - x \|^2 + \| (I - P)y \|^2.$$

Отсюда видно, что расстояние y до $x \in M^\perp$ минимально, когда $x = Py$. Таким образом, P – оператор проектирования на подпространство $M^\perp$.

6. Алгоритм для задачи квадратичного программирования. Перейдем теперь к исследованию возможностей практического решения задачи квадратичного программирования. Здесь будет изложен один из эффективных алгоритмов, хотя их существует достаточно большое количество. Выбор предлагаемого метода обусловлен тремя причинами:

1) этот метод конечен, т.е. сходится за конечное число шагов;

2) метод является обобщением симплекс-метода линейного программирования, и поэтому здесь может быть использован весь арсенал приемов, используемых при решении задач линейного программирования, в частности, приемы работы с разреженными матрицами;

3) метод может быть распространен на задачу минимизации произвольной функции при линейных ограничениях.

Рассмотрим задачу

$$\min\{\ \tfrac{1}{2}(x, Cx) + (d, x):\ Ax = b,\ \ x \geqslant 0\}\ , \tag{3.16}$$

где C – положительно определенная матрица, A – матрица размеров $m \times n$ $b \in \mathbf{R}^n$. Задача (3.16) называется *задачей квадратичного программирования в стандартной форме.*

Любая задача с общими ограничениями (3.5) может быть приведена к стандартной форме следующими приемами.

Если в (3.5) $I^- \neq \phi$, то для каждого $i \in I^-$ вводится дополнительная переменная z^i и неравенство

$$(a_i, x) \leqslant \alpha_i$$

заменяется соотношением

$$(a_i, x) + z^i = \alpha_i,\ \ z^i \geqslant 0.$$

Таким образом, в новых переменных все ограничения являются ограничениями типа равенств, а ограничения типа неравенств имеют простой вид $x^j \geqslant 0$ или $z^i \geqslant 0$.

Если после этих преобразований на какую-то компоненту x^j нет требования положительности, то вместо x^j подставляются две новые переменные x^{j+} и x^{j-} из соотношения

$$x^j = x^{j+} - x^{j-},\ \ x^{j+} \geqslant 0,\ \ x^{j-} \geqslant 0,$$

после чего задача приобретает стандартный вид. На практике такую постановку не надо реально делать, а следует просто учесть, что

$$x^j = x^{j+},\ \ x^{j-} = 0, \qquad \text{если } x^j \geqslant 0,$$

$$x^j = -x^{j-},\ \ x^{j+} = 0, \qquad \text{если } x^j \leqslant 0.$$

Таким образом, не уменьшая общности, можно рассматривать задачу только в стандартной форме.

Договоримся о некоторых обозначениях, которые будут иметь силу лишь внутри этого параграфа.

Будем обозначать столбцы матрицы A через A_j. Таким образом,

$$Ax = \sum_{j=1}^{n} A_j x^j.$$

Пусть J – любое подмножество индексов из $1, 2, \ldots, n$. Через $x|_J$ будет обозначаться вектор с компонентами x^j, $j \in J$, причем порядок следования компонент соответствует возрастанию индексов. Точно так же $A|_J$ – матрица размеров $m \times |J|$, где $|J|$ – число элементов в множестве J, со столбцами A_j, $j \in J$. Такому обозначению соответствует обозначение через $f'|_J(x)$ вектор-строки с компонентами $\partial f/\partial x^j$, $j \in J$. Удобно будет также обозначать через $\bar{J}$ дополнение J до $\{1, \ldots, n\}$.

Во введенных обозначениях, например, справедливо равенство

$$A|_J x|_J + A|_{\bar{J}} x|_{\bar{J}} = Ax.$$

Будем говорить, что векторы A_j, $j \in \bar{J}$, образуют *базис*, если $|\bar{J}| = m$ и A_j линейно независимы. Соответствующие переменные x^j будем называть *базисными*. Таким образом, если A_j, $j \in \bar{J}$, – базис, то определена матрица

$$B_{\bar{J}} = (A|_{\bar{J}})^{-1}.$$

При заданном базисе базисные переменные легко выражаются через остальные из соотношения

$$Ax = A|_J x|_{\bar{J}} + A|_J x|_J = b$$

по формуле

$$x|_{\bar{J}} = B_{\bar{J}} b - B_{\bar{J}} A|_J x|_J. \tag{3.17}$$

Если обозначить через

$$\partial x|_{\bar{J}} / \partial x|_J$$

матрицу с компонентами $\partial x^j / \partial x^k$, $j \in \bar{J}$, $k \in J$, то из (3.17) следует, что

$$\partial x|_{\bar{J}} / \partial x|_J = -B_{\bar{J}} A|_J.$$

Пусть базисные переменные выражены через небазисные по формуле (3.17) и подставлены в $f(x)$. Обозначим соответствующую функцию через $f|_J(x|_J)$. Тогда согласно известным формулам дифференцирования

$$(f|_J)' = f'|_J + f'|_{\bar{J}} \frac{\partial x|_{\bar{J}}}{\partial x|_J},$$

или

$$(f|_J)' = f'|_J - f'|_{\bar{J}} B_J A|_J. \tag{3.18}$$

Заметим, что здесь $(f|_J)'$ – вектор-строка, составленная из производных сложной функции $f|_J$ от переменных x^j, $j \in J$, в то время как, например, $f'|_J$ – это вектор-строка из производных $\partial f/\partial x^j$, $j \in J$, от исходной функции.

Поясним теперь основную идею алгоритма. Пусть выбрана начальная точка x, удовлетворяющая ограничениям задачи. Выберем базис $\bar{J}$. Оставшиеся индексы, т.е. множество J, по некоторому признаку, о котором будет сказано отдельно, будут разделены на активные и пассивные: J_a и J_p, $J_a \cup J_p = J$. Это разделение будет зависеть от точки x. Заметим только, что всегда

$$x^j = 0, \quad j \in J_p. \tag{3.19}$$

Удерживая теперь пассивные переменные в нуле, меняем активные переменные, выражая базисные переменные через них по формуле (3.17). Так как базисные переменные линейно выражаются через остальные, то после подстановки в $f(x)$ мы получим снова квадратичную функцию от активных переменных. Эту функцию будем минимизировать методом сопряженных направлений по активным переменным. Однако из-за ограничения на знак переменных необходимо следить за ходом процесса сопряженных направлений. Если шаг этого процесса приводит к отрицательности одной из переменных, то длина шага ограничивается так, чтобы в точности обратить эту переменную в нуль. После этого обратившаяся в нуль переменная относится к пассивным. При этом, если обратилась в нуль активная переменная, то далее идет процесс метода сопряженных направлений по оставшимся активным переменным. Если в нуль обратилась базисная переменная, то базис меняется – обратившаяся в нуль переменная относится к пассивным, а в базис вводится одна из активных переменных.

Как видно из изложенного, указанный процесс не может продолжаться бесконечно, так как число пассивных переменных все время растет, а число активных уменьшается. Поэтому в какой-то момент число активных переменных зафиксируется и метод сопряженных направлений найдет минимум квадратичной функции по активным переменным. Полученная точка выбирается за исходную и процесс продолжается.

Конечность алгоритма будет обусловлена следующими факторами:

1. Поскольку начальная точка удовлетворяет ограничениям задачи, а базисные переменные выражаются по формуле (3.17), то в ходе процесса выполняются все ограничения задачи.

2. Весь процесс состоит из больших шагов, подобных описанному выше, и каждый шаг кончается после конечного числа операций нахождением минимума по активным переменным при фиксированных пассивных.

3. Число возможных выборов групп пассивных переменных конечно.

4. Процесс монотонен, т.е. функция $f(x)$ все время строго убывает, что исключает возможность повторения группы пассивных переменных, при равенстве которых нулю найден минимум.

5. После каждого большого итерационного шага процесс начинается сначала с полученной точки, и при этом, если она не является решением задачи, происходит ненулевое уменьшение функции f.

Таким образом, алгоритм должен быть построен так, чтобы перечисленные выше факторы были удовлетворены.

Для того чтобы обеспечить успешную работу алгоритма, необходимо наложить на задачу требование невырожденности, близкое к аналогичному требованию в линейном программировании.

Условие невырожденности. Пусть x удовлетворяет ограничениям $Ax = b$, $x \geqslant 0$, а $\overline{J}(x) = \{j: x^j > 0\}$. Тогда существует такое множество $\overline{J} \subseteq \overline{J}(x)$, $|\overline{J}| = m$, что векторы A_j, $j \in \overline{J}$, линейно независимы.

Предполагая в дальнейшем выполнение этого условия, сформулируем алгоритм. Для этого мы опишем один большой шаг алгоритма.

0. Пусть x_0 – точка, удовлетворяющая ограничениям $Ax = b$, $x \geqslant 0$.

Выберем базис $\overline{J}^0$ из подмножества столбцов $\overline{J}(x_0)$. В силу условия невырожденности это возможно.

1. Положим J^0 равным дополнению $\bar{J}^0$ до $\{1, \ldots, n\}$. Вычислим $B_{\bar{J}^0}$ и вектор

$$(f|_{J^0})' = f'|_{J^0} - f'|_{\bar{J}^0} B_{\bar{J}^0} A|_{J^0},$$

соответствующий точке x_0. Если для $j \in J^0$

$$(f|_{J^0})'|_j = 0 \qquad \text{при} \quad x_0^j > 0,$$

$$(f|_{J^0})'|_j \geqslant 0 \qquad \text{при} \quad x_0^j = 0,$$

то x_0 – решение задачи.

2. Пусть

$$J_a^0 = \{j:\ x_0^j > 0,\ (f|_{\bar{J}^0})'|_j \neq 0$$

$$\text{или } (f|_{J^0})'|_j < 0,\ x_0^j \doteq 0,\ j \in J^0\}, \quad J_p^0 = J^0 \setminus J_a^0.$$

3. Для $k = 0, 1, \ldots$ применяем метод сопряженных направлений для минимизации функции $f|_{J^k}$, получающейся из f подстановкой вместо $x|_{\bar{J}^k}$ выражения (3.17):

$$x|_{\bar{J}^k} = B_{\bar{J}^k} b - B_{\bar{J}^k} A|_{J^k}\ x|_{J^k}.$$

Градиент этой функции вычисляется по формуле (3.18). При этом минимизация производится только по аргументам x^j, $j \in J_a^k$.

Функция $f|_{J^k}$ – квадратичная по $x|_{J^k}$, но матрица C_{J^k}, определяющая квадратичную часть, явно не выписана. Поэтому для вычисления шага α вдоль направления $p|_{J_a^k}$ из точки x применяется следующий прием.

Вектор $p|_{J_a^k}$ определяется по формулам метода сопряженных направлений. Полагаем также

$$p|_{J_p^k} = 0, \quad p|_{J^k} = -B_{\bar{J}^k} A|_{J^k}\, p|_{J^k}.$$

По формулам метода сопряженных направлений

$$\alpha = -(p|_{J^k}, (f|_{J^k})'(x)) / (p|_{J^k},\ C_{J^k}\, p|_{J^k}). \tag{3.20}$$

Числитель в (3.20) вычисляется по приведенным выше формулам для $(f|_{J^k})'$, а

$$C_{J^k}\, p|_{J^k} = (f|_{J^k})'(x + p) - (f|_{J^k})'(x),$$

так как для любой квадратичной функции f

$$f'(x) = Cx + d, \quad f'(x + p) - f'(x) = Cp.$$

Далее учитывается несколько моментов. Если знаменатель в (3.20) равен нулю и числитель тоже равен нулю, то процесс применения метода сопряженных направлений окончен, текущая точка выбирается в качестве начальной и происходит возвращение к шагу 0.

Если же числитель меньше нуля (только этот случай может возникнуть по построению метода сопряженных направлений), то полагаем $\alpha = +\infty$.

Вычисляем

$$\bar{\alpha} = \min\{-x^j / p^j:\ j \in J_a^k \cup \bar{J}^k,\ p^j < 0\}.$$

Если $\{j \in J_a^k \cup \bar{J}^k:\ p^j < 0\} = \phi$, то $\bar{\alpha} = +\infty$.

Если $\alpha < \bar{\alpha}$, то строится новая точка $x + \alpha p$ и процесс минимизации $f|_{J^k}$ по x^j, $j \in J_g^k$, продолжается.

Если $\alpha \geqslant \bar{\alpha}$, то строится новая точка $\bar{x} = x + \bar{\alpha} p$.

В случае, если $\bar{\alpha} = +\infty$, функция f неограничена снизу и работа алгоритма окончена. Если $\bar{\alpha} < +\infty$, то происходит перестройка множеств J_a^k, J_p^k, $\bar{J}^k$. А именно, пусть $j \in J_a^k \cup \bar{J}^k$ – такой индекс, что $\bar{\alpha} = -x^j / p^j$. Тогда полагаем

$$J_p^{k+1} = J_p^k \cup \{j\}, \quad J_a^{k+1} = J_a^k \setminus \{j\}, \quad \bar{J}^{k+1} = \bar{J}^k,$$

если $j \in J_a^k$. Если же $j \in \bar{J}^k$, то

$$\bar{J}^{k+1} = \{\bar{J}^k \setminus \{j\}\} \cup \{l\}, \quad J_a^{k+1} = J_a^k \setminus \{l\},$$

где $l \in \bar{J}(x + \alpha p) \setminus \bar{J}^k$ – такой индекс, для которого $x^l + \bar{\alpha} p^l > 0$, и векторы A_j, $j \in \bar{J}^{k+1}$, линейно независимы. После этого происходит возврат к началу шага 3.

После конечного числа операций для какого-то k (ведь множество J_p^k все время расширяется) текущая точка x доставит минимум $f|_{J^k}$ по x^j, $j \in J_a^k$. Эта точка берется за исходную, и происходит возврат к шагу 0.

4. Окончание процесса происходит на шаге 2, если оказывается, что $J_a^0 = \phi$.

Обоснование конечной сходимости алгоритма в основных чертах уже приведено выше. Необходимо отметить лишь ряд деталей.

Если множество J_a^0 непусто, то на первом шаге применения метода сопряженных градиентов компоненты вектора p, вдоль которого совершается сдвиг из точки x_0, для $j \in J_a^0$ совпадают со взятыми со знаком минус производными функции $f|_{J^0}$. Эти производные отличны от нуля, если $x_0^j > 0$, и строго отрицательны, если $x_0^j = 0$. Поэтому первый шаг всегда приведет к успешному ненулевому сдвигу и строгому уменьшению целевой функции.

Допустим, выполнился критерий останова в п. 1. Положим $u \in \mathbf{R}^m$, $v \in \mathbf{R}^n$:

$$u^* = -f'|_{\bar{J}^0} B_{\bar{J}^0}, \tag{3.21}$$

$$v^*|_{J^0} = (f|_{J^0})', \tag{3.22}$$

$$v^*|_{\bar{J}^0} = 0. \tag{3.23}$$

Тогда в силу (3.18) и (3.21) – (3.23)

$$f'|_{J^0} + u^* A|_{J^0} = (f|_{J^0})' = v^*|_{J^0},$$

$$f'|_{\bar{J}^0} + u^* A|_{\bar{J}^0} = 0 = v^*|_{\bar{J}^0},$$

где использовано то, что $B_{\bar{J}} = (A|_{\bar{J}})^{-1}$. Полученные два равенства, учитывая сделанные обозначения, можно объединить в одно:

$$f' + u^* A = v^*.$$

Заметим теперь, что по определению останова в п. 1 алгоритма

$$v \geqslant 0, \quad v^j x_0^j = 0.$$

Сопоставляя это с теоремой 2.4, получаем, что x_0 – решение задачи квадратичного программирования. Этим сходимость алгоритма полностью обоснована.

Заметим, что попутно мы получили формулы (3.21) – (3.23) для вычисления множителей Куна – Таккера.

Обратим также внимание на то, что равенство $J_a^k = \phi$, $k \geqslant 1$, указывает лишь на завершение процесса минимизации по методу сопряженных градиентов. Равенство $J_a^k = \phi$ указывает лишь на то, что при фиксированных пассивных переменных нельзя добиться дальнейшего уменьшения значения функции.

Далее, по построению базисные переменные всегда строго положительны. Это, а также возможность перестройки базиса при обращении одной из них в нуль, обеспечивается условием невырожденности.

7. **Вычислительные аспекты.** В предыдущем изложении был оставлен без внимания ряд существенных практических вопросов, без решения которых приведенный алгоритм не может рассматриваться как эффективный. К счастью, эти вопросы уже давно успешно решены в линейном программировании и здесь будут кратко изложены. Более детальное их рассмотрение читатель найдет в литературе к этому параграфу.

Для начала работы алгоритма необходима какая-либо точка, удовлетворяющая ограничениям $Ax = b$, $x \geqslant 0$. Если такая точка неизвестна из каких-либо априорных соображений, то следует использовать какую-либо стандартную программу линейного программирования. Такие программы достаточно широко распространены, и поэтому на этом вопросе мы не будем останавливаться. Сосредоточим внимание на вопросах вычисления матриц B_J, $B_J A_J$, которые необходимы в ходе работы алгоритма.

Пусть решается система $m \times n$ уравнений

$$Ax = b, \tag{3.24}$$

причем среди столбцов матрицы A имеется m линейно независимых. Обозначим строки матрицы A через a_i, столбцы через A_j, а элементы через a_{ij}.

Стандартный прием решения системы (3.24) состоит в том, что из i-го уравнения, если $a_{ij} \neq 0$, выражается x^j через остальные переменные и подставляется в остальные уравнения.

Такой метод исключения неизвестных называется *методом исключения Гаусса – Жордана.* Из преобразованной системы исключается следующее неизвестное и т.д.

Опишем этот прием в удобной для дальнейшего форме. Известно, что он эквивалентен следующему преобразованию: следует умножить i-ю строку матрицы на множитель t_k и прибавить ее к k-й строке, выбрав t_k так, чтобы преобразованный элемент a_{kj} равнялся нулю, если $k \neq i$, и единице, если $k = i$.

Если преобразованную матрицу обозначить через $\bar{A}$, то

$$\begin{aligned} &\bar{a}_k = a_k + t_k a_i, \quad t_k = -a_{kj}/a_{ij}, \quad k \neq i, \\ &\bar{a}_i = t_i a_i, \quad t_i = 1/a_{ij}. \end{aligned} \tag{3.25}$$

Из этих формул следует, что если $a_{il} = 0$, то

$\bar{a}_{kl} = a_{kl}$ для всего столбца l.

Введем матрицу

$$T_{ij} = \begin{bmatrix} 1 & 0 & \dots & 0 & t_1 & \overbrace{0 \dots 0}^{m-i} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & t_{i-1} & 0 & \dots & 0 \\ 0 & \dots & 0 & t_i & 0 & \dots & 0 \\ 0 & \dots & 0 & t_{i+1} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & t_m & 0 & \dots & 0 \quad 1 \end{bmatrix}. \tag{3.26}$$

Ее определитель, очевидно, равен $t_i \neq 0$, так что матрица невырождена. Тогда указанное преобразование, как нетрудно убедиться, приводит к новой системе

$$\bar{A}x = \bar{b}, \quad \bar{A} = T_{ij}A, \quad \bar{b} = T_{ij}b.$$

При этом переменная x^i входит только в i-е уравнение с коэффициентом 1:

$$\bar{a}_{kj} = 0, \quad k \neq i, \quad \bar{a}_{ij} = 1.$$

По-другому это можно записать в виде

$$\bar{A}_j = e_i,$$

где e_i — вектор, у которого все компоненты равны нулю, кроме i-й, которая равна 1.

Поступим теперь следующим образом. Возьмем $i = 1$ и найдем $a_{1j_1} \neq 0$. Обозначим $j(1) = j_1$. Положим

$$T^{(1)} = T_{1j_1}, \quad A^{(1)} = T^{(1)}A, \quad b^{(1)} = T^{(1)}b.$$

Выбрав $i = 2$, аналогичным образом среди элементов $a^{(1)}_{2j}$ находим ненулевой: $a^{(1)}_{2j_2} \neq 0$. Такой элемент обязательно должен найтись, так как в противном случае вся вторая строка матрицы $A^{(1)}$ была бы нулевой и ранг $A^{(1)}$ был бы меньше m. Но так как ранг матрицы A равен m, а матрица $T^{(1)}$ невырождена, то ранг $A^{(1)}$ также равен m. Кроме того, $j_2 \neq j_1$, так как $a_{kj_1} = 0$, $k \neq 1$, по построению $A^{(1)}$, так что $a_{2j_1} = 0$.

Полагаем

$$j(2) = j_2, \quad T^{(2)} = T_{2j_2}, \quad A^{(2)} = T^{(2)}A^{(1)} = T^{(2)}T^{(1)}A, \quad b^{(2)} = T^{(2)}T^{(1)}b,$$

причем элементы матрицы T_{2j_2} строятся по элементам матрицы $A^{(1)}$. Заметим при этом, что так как $a^{(1)}_{2j} = 0$, то $A^{(2)}_{j_1} = A^{(1)}_{j_1} = e_1$.

Продолжая аналогично, после m шагов получаем матрицы

$$A^{(m)} = T^{(m)} \dots T^{(1)},$$

$$A = T^{(m)}A^{(m-1)}, \quad b^{(m)} = T^{(m)} \dots T^{(1)}, \quad b = T^{(m)}b^{(m-1)} \tag{3.27}$$

и взаимно однозначное отображение $j(i) = j_i$. В силу взаимной однозначности определено и обратное отображение $i(j)$ для $j \in \{j_1, j_2, \dots, j_m\}$: $i(j_k) = k$. По построению столбец с номером j_k совпадает с k-м единичным ортом e_k:

$$A^{(m)}_{j_k} = e_k. \tag{3.28}$$

Обозначим $\bar{J} = \{j_1, \ldots, j_m\}$. Столбцы $A_j^{(m)}, j \in \bar{J}$, образуют базис для $A_\cdot^m$, так как они совпадают с различными единичными ортами. Так как делались невырожденные преобразования, то и столбцы A_j, $j \in \bar{J}$, исходной матрицы линейно независимы и образуют базис. Вообще, свойство быть линейно зависимыми или независимыми сохраняется для столбцов исходной и преобразованных матриц.

Образуем теперь матрицу $\bar{A}_{\bar{J}} = |A_{j_1} \ \ldots \ A_{j_m}|$. Из формул (3.27), (3.28) следует, что

$$\bar{B}_{\bar{J}} \bar{A}_{\bar{J}} = I_m, \ \bar{B}_{\bar{J}} = T^{(m)} \ \ldots \ T^{(1)}, \tag{3.29}$$

где I_m – единичная матрица порядка $m \times m$.

Таким образом, $\bar{B}_{\bar{J}}$ является обратной к матрице $\bar{A}_{\bar{J}}$. Однако $\bar{B}_{\bar{J}}$ не совпадает с $B_{\bar{J}} = (A|_{\bar{J}})^{-1}$, фигурировавшей в предыдущем пункте. Дело в том, что в матрице $A|_{\bar{J}}$ столбцы A_j упорядочены в естественном первоначальном порядке. Однако нетрудно видеть, что это не вызывает никаких сложностей. Действительно, перестановка столбцов в A эквивалентна просто перестановке переменных и на суть задачи не влияет. Поэтому, если базис $\bar{J}$ выбран и известна функция $j(i)$, $i = 1, \ldots, m$, то во всех вычислениях, необходимых для работы алгоритма предыдущего пункта, в которых участвует базис, необходимо считать, что перестановка уже произведена и элементы $j \in \bar{J}$ берутся в порядке $j_1, \ldots, j_m$, а вместо матрицы $B_{\bar{J}}$ использовать матрицу $\bar{B}_{\bar{J}}$. Так, например, $f'|_{\bar{J}}$ теперь есть вектор-строка с компонентами $\partial f/\partial x^j$, $j = j_1, \ldots, j_m$:

$$f'|_{\bar{J}} = \{\partial f/\partial x^{j_1}, \ \partial f/\partial x^{j_2}, \ \ldots, \ \partial f/\partial x^{j_m}\}.$$

Резюмируя сказанное, заключаем, что если начальная точка x_0 известна, то исходный базис $\bar{J}^0$ и матрица $B_{\bar{J}^0}$ могут быть эффективно построены с помощью только что описанной процедуры исключения Гаусса – Жордана, примененной к матрице $A|_{\bar{J}(x_0)}$.

Допустим теперь, что на некотором этапе имеется текущая точка x, исходная матрица A преобразовывалась r раз, так что имеются текущая матрица $A^{(r)}$, базис $\bar{J}$, отображение $j(i)$, $i = 1, \ldots, m$, и матрица $\bar{B}_{\bar{J}}$:

$$\bar{B}_{\bar{J}} = T^{(r)} \ \ldots \ T^{(1)}, \ A^{(r)} = \bar{B}_{\bar{J}} A, \ A^{(r)}_{j(i)} = e_i, \ i = 1, \ldots, m. \tag{3.30}$$

Пусть базисная компонента x^{j_0} обратилась в нуль. По условию невырожденности среди A_l, $l \in \bar{J}(x) = \{j: x^j > 0\}$, найдется не менее m линейно независимых векторов. Поэтому, если $i_0 = i(j_0)$, где $i(j)$ – обратное отображение к $j(i)$ (ведь каждому i однозначно соответствовал номер j), то среди элементов $a^{(r)}_{i_0 l}$, $i \in \bar{J}(x) \setminus \bar{J}$, должен найтись ненулевой. В самом деле, по предположению число элементов в $\bar{J}(x)$ не меньше m. Индекс $j_0 \in \bar{J}$, для которого $a_{i_0 j_0} = 1$, в это множество не попадает. Далее $a^{(r)}_{i_0 j} = 0$, $j \neq j_0$, $j \in \bar{J}$, по построению матрицы $A^{(r)}$ и базиса.

Итак, если $a_{i_0 l} = 0$ для всех $i \in \bar{J}(x) \setminus \bar{J}$, то все векторы $A^{(r)}_j$, $j \in \bar{J}(x)$, имеют нулевую i_0-ю компоненту, т.е. фактически являются $(m-1)$-мерными. В то же время их число не меньше m, и, значит, они линейно зависимы. Это противоречит невырожденности задачи.

Итак, если $i_0 = i(j_0)$, то найдется индекс $i \in \bar{J}(x) \setminus \bar{J}$, т.е. $l \in J_a$, такой, что $a^{(r)}_{i_0 l} \neq 0$. Так как $a^{(r)}_{i_0 j} = 0$, $j \neq j_0$, $j \in \bar{J}$, то $A^{(r)}_l$ линейно не зависит от A_j, $j \in \bar{J} \setminus \{j_0\}$, и поэтому

$$\bar{J} = (\bar{J} \setminus \{j_0\}) \cup \{l\}$$

есть индексное множество нового базиса.

Исходя теперь из элемента $a^{(r)}_{il}$ и матрицы $A^{(r)}$ проведем преобразование $T^{(r+1)} = T_{il}$ и получим

$$A^{(r+1)} = T^{(r+1)} A^{(r)}, \quad \bar{B}^{(r+1)} = T^{(r+1)} \bar{B}^{(r)}.$$

Что касается отображения $j(i)$, то для $i \neq i_0$ оно остается прежним, а $j(i_0) = l$. Итак, показано, что изменение базиса также может быть осуществлено за счет операции исключения Гаусса – Жордана.

В заключение остановимся на вопросе хранения информации. Существуют два способа.

1. Хранятся $A^{(r)}$, $\bar{B}^{(r)}$ и отображение $j(i)$. (Последнее, очевидно, требует $2m$ ячеек памяти и не составляет проблемы.) При таком способе хранения матрица $A^{(r)}$ преобразуется непосредственно, т.е. если выбран элемент $a^{(r)}_{i_0 l}$ для преобразования, что строки $a^{(r+1)}_k$ матрицы $A^{(r+1)}$ получаются по формулам

$$a^{(r+1)}_k = a^{(r)}_k + t^{(r)}_k a^{(r)}_{i_0}, \quad t^{(r)}_k = -a^{(r)}_{kl} / a^{(r)}_{i_0 l}, \quad k \neq i_0,$$

$$a^{(r+1)}_{i_0} = t^{(r)}_{i_0} a^{(r)}_{i_0}, \quad t^{(r)}_{i_0} = 1/a^{(r)}_{i_0 l}.$$

Точно так же $\bar{B}^{(r+1)} = T^{(r+1)} \bar{B}^{(r)}$.

Такой способ хранения прост и удобен для преобразований, но имеет следующие недостатки: а) может накапливаться погрешность вычислений; б) растет число ненулевых элементов в $A^{(r)}$, так что если в исходной матрице A их было мало, то $A^{(r)}$ при большом r может быть сильно заполненной.

2. Хранятся исходная матрица A, матрицы $T^{(1)}, \ldots, T^{(r)}$ и функция $j(i)$. Очевидно, что для матрицы $T^{(r)}$ необходимо хранить только ненулевые элементы $t^{(r)}_k$, $k = 1, \ldots, m$, и номер i_r строки, на основе которой производится преобразование.

Преимущества такого способа хранения состоят в том, что, как правило, среди $t^{(r)}_k$ также много нулевых элементов, которые можно не хранить, что сокращает требуемую память. Кроме того, время от времени, зная текущий базис на основе функции $j(i)$ и исходную матрицу A, можно вычеркнуть все ранее хранившиеся матрицы $T^{(1)}, \ldots, T^{(r)}$ и пересчитать новые матрицы

$$T^{(i)} = T_{ij(i)}, \quad i = 1, \ldots, m,$$

где, естественно, каждая очередная матрица $T^{(i+1)}$ считается на основе элементов предыдущей матрицы

$$A^{(i)} = T^{(i)} \ldots T^{(1)} A.$$

Как мы видели раньше,

$$B^{(r)} = \overline{B_J} = [A_{j_1} \ \ldots \ A_{jm}]^{-1}, \quad A^{(r)} = B^{(r)} A,$$

и поэтому $A^{(r)}$ и $B^{(r)}$ не зависят от последовательности матриц $T^{(1)}, \ldots, T^{(r)}$, а только от текущего базиса $\overline{J}$ и отображения $j(i)$.

Пересчет матриц $T^{(r)}$ может быть обусловлен их чрезмерным накоплением либо необходимостью предотвратить накопление ошибок вычислений.

8. Алгоритм для простых ограничений. Обобщение. Приведенный в п. 6 алгоритм существенно упрощается, если имеются только ограничения на знак переменных, т.е. если рассматривается задача

$$\min \{ f(x) = \tfrac{1}{2}(x, Cx) + (d, x): \ x \geqslant 0 \}.$$

В этом случае нет никаких базисов, а имеется только разделение переменных на пассивные и активные.

Алгоритм приобретает следующий вид. Пусть x_0 — начальная точка.

0. Если $\partial f/\partial x^j = 0, x_0^j > 0$, и $\partial f/\partial x^j \geqslant 0, \ x_0^j = 0$, то x_0 — решение задачи.

1. Полагаем

$$J_a^0 = \{ j: \ x_0^j > 0 \ \text{ или } \ x_0^j = 0, \ \partial f/\partial x^j < 0 \}, \quad J_p^0 = \{ 1, \ldots, n \} \setminus J_a^0.$$

2. Если $J_a^0 \neq \phi$, то для $k = 0, 1, \ldots$ производим минимизацию функции $f(x)$ по переменным $x^j, \ j \in J_a^k$, методом сопряженных направлений с проверкой условий на знак переменных, так же, как в основном алгоритме. Если по ходу процесса какая-то координата $x^j, j \in J_a^k$, обратилась в нуль, то

$$J_a^{k+1} = J_a^k \setminus \{j\}, \quad J_p^{k+1} = J_p^k \cup \{j\}$$

и возвращаемся к началу шага 2.

Если при каком-то J_a^k процесс метода сопряженных направлений завершился нахождением минимума по переменным $x^j, j \in J_a^k$, то полученная точка берется за исходную и происходит возврат к шагу 1 алгоритма.

Сходимость алгоритма следует из доказанной сходимости общего алгоритма. Однако в данном случае доказательство существенно проще и основано на том, что по ходу процесса множества J_p^k все время расширяются и поэтому шаг 2 алгоритма должен закончиться после конечного числа операций нахождением минимума по $x^j, \ j \in J_a^k$. Так как в ходе процесса функция $f(x)$ все время убывает, то конечные множества J_a^k на всех этапах процесса не могут повторяться. А так как таких множеств, очевидно, конечное число, то и весь алгоритм сходится после конечного числа операций.

Сделаем несколько заключительных замечаний.

1. Если на какую-то из переменных x^j нет ограничения по знаку, то ее сразу же целесообразно включать в базис, после чего можно не обращать внимания на ее знак в ходе процесса.

2. Если на переменные есть ограничения сверху: $0 \leqslant x^j \leqslant w^j$, то алгоритм также может быть модифицирован. При этом пассивными будут как переменные, обращающиеся в нуль, так и достигающие верхней грани.

3. Вместо описанного выше алгоритма метода сопряженных направлений может быть использована любая другая его модификация, описанная в литературе. Единственное существенное требование, которое при этом необходимо удовлетворить, — это, чтобы начальный шаг процесса выбирался вдоль направления антиградиента.

Глава 2.

МЕТОД ЛИНЕАРИЗАЦИИ

§ 4. ОБЩИЙ АЛГОРИТМ

В этом параграфе мы рассмотрим метод решения общей задачи математического программирования, не делая каких-либо допущений о выпуклости встречающихся функций. Существенной особенностью метода является возможность учета нелинейных ограничений типа равенств, что является камнем преткновения для большинства других методов.

Пусть требуется минимизировать функцию $f_0(x)$, $x \in E^n$, при ограничениях

$$f_i(x) \leqslant 0, \quad i \in I^-, \quad f_i(x) = 0, \quad i \in I^0, \tag{4.1}$$

где I^-, I^0 – конечные множества индексов. Предположим, что все функции $f_i(x)$ непрерывно дифференцируемы. Более полно ограничения, при которых исследуется задача, будут оговорены ниже. Заменим в точке x_0 все ограничения (4.1) и $f_0(x)$ на линейные, линеаризовав $f_i(x)$ в точке x_0. В результате получится некоторая задача линейного программирования. Естественно было бы решение линеаризованной задачи взять в качестве следующего приближения, как это делается в методе Ньютона для решения систем нелинейных уравнений. К сожалению, прямо этот путь не приводит к цели, так как обычно вспомогательная задача линейного программирования не имеет решения. Поэтому необходимо наложить некоторые ограничения на приращение вектора x в точке x_0, чтобы решение линеаризованной задачи в точке x_0 не уходило слишком далеко от x_0, оставаясь в такой окрестности x_0, в которой линеаризация еще справедлива. Это и будет проделано ниже путем добавления квадратичного члена к линеаризованной целевой функции.

Заметим, что каждое равенство $f_i(x) = 0$ эквивалентно двум неравенствам $f_i(x) \leqslant 0$, $-f_i(x) \leqslant 0$. Поэтому можно ограничиться рассмотрением лишь случая наличия только ограничений типа неравенств. Такое ограничение удобно по крайней мере при теоретическом обосновании алгоритма, хотя удвоение числа неравенств может быть неудобным при вычислениях. Ниже будет изложено теоретическое обоснование алгоритма для задачи минимизации $f_0(x)$ при ограничениях

$$f_i(x) \leqslant 0, \quad i \in I. \tag{4.2}$$

О модификации алгоритма для общей задачи (4.1) будет сказано отдельно.

Таким образом, не теряя общности, мы будем исследовать алгоритм для задачи (4.2). Ясно, что всегда можно предполагать наличие среди неравенств (4.2) тривиального: $0 \leqslant 0$. Поэтому будем предполагать, что среди функций $f_i(x)$, $i \in I$, имеется одна, равная тождественному нулю.

1. Основные предположения. Положим

$$F(x) = \max_{i \in I} f_i(x),$$
$$I_\delta(x) = \{i \in I:\ f_i(x) \geqslant F(x) - \delta\}, \quad \delta \geqslant 0. \tag{4.3}$$

В силу ранее сделанного предположения $F(x) \geqslant 0$ при всех x. Предположим, что существуют такие константы $N > 0$, $\delta > 0$, что

а) множество

$$\Omega_N = \{x:\ f_0(x) + NF(x) \leqslant C_0\}, \quad C_0 = f_0(x_0) + NF(x_0),$$

ограничено;

б) градиенты функций $f_i(x)$, $i \in \{0\} \cup I$, в Ω_N удовлетворяют условию Липшица, т.е.

$$\| f_i'(x_1) - f_i'(x_2) \| \leqslant L \| x_1 - x_2 \|;$$

в) задача квадратичного программирования

$$\min ((f_0'(x), p) + \tfrac{1}{2} \| p \|^2),$$
$$(f_i'(x), p) + f_i(x) \leqslant 0, \quad i \in I_\delta(x), \tag{4.4}$$

разрешима относительно $p \in E^n$ при любом $x \in \Omega_N$, и существуют такие множители Лагранжа $u^i(x)$, $i \in I_\delta(x)$, что $\sum_{i \in I_\delta(x)} u^i(x) \leqslant N$; здесь и всюду в этом параграфе $\| p \|$ обозначает евклидову норму вектора p.

В дальнейшем решение задачи (4.4) будем обозначать через $p(x)$, а множители Лагранжа – через $u^i(x)$, $i \in I_\delta(x)$.

2. Формулировка алгоритма. Пусть x_0 – начальное приближение и выбрано ϵ, $0 < \epsilon < 1$. Пусть в процессе работы алгоритма уже получена точка x_k. Построение следующего приближения производим в два этапа.

1. Решаем задачу (4.4) при $x = x_k$, находим ее решение – вектор $p_k = p(x_k)$.

2. Находим первое значение $i = 0, 1, \ldots$, при котором будет выполнено неравенство

$$f_0(x_k + (1/2)^i p_k) + NF(x_k + (1/2)^i p_k) \leqslant f_0(x_k) + NF(x_k) - (1/2)^i \epsilon \| p_k \|^2.$$

Если это неравенство впервые выполнилось при $i = i_0$, то полагаем

$$\alpha_k = 2^{-i_0}, \quad x_{k+1} = x_k + \alpha_k p_k.$$

Таким образом, на каждом шаге выполняется следующее неравенство:

$$f(x_{k+1}) + NF(x_{k+1}) \leqslant f(x_k) + NF(x_k) - \alpha_k \epsilon \| p_k \|^2. \tag{4.5}$$

3. Сходимость алгоритма. Покажем, что выбор шага α_k на каждой итерации осуществляется за конечное число делений единицы пополам, и обоснуем сходимость алгоритма.

Из результатов, изложенных в § 2, следует, что $p(x)$ есть решение задачи (4.4) тогда и только тогда, когда существуют такие $u^i(x) \geqslant 0$, $i \in I_\delta(x)$, что

$$f_0'(x) + (p(x))^* + \sum_{i \in I_\delta(x)} u^i(x) f_i'(x) = 0,$$
$$u^i(x)((f_i'(x), p(x)) + f_i(x)) = 0, \quad i \in I_\delta(x). \tag{4.6}$$

Напомним, что в наших обозначениях градиент $f'(x)$ есть вектор-строка.

Поскольку $p(x)$ – вектор-столбец, он входит в (4.6) в транспонированном виде, фигурируя здесь как градиент функции $\frac{1}{2}\|p\|^2$. Поэтому

$$(f_0'(x),\ p(x)) = -\sum_{i \in I_\delta(x)} u^i(x)(f_i'(x),\ p(x)) -$$

$$-\|p(x)\|^2 = \sum_{i \in I_\delta(x)} u^i(x) f_i(x) - \|p(x)\|^2. \qquad (4.7)$$

Лемма 4.1. *Для того чтобы точка x удовлетворяла неравенствам* (4.2) *и в ней выполнялись необходимые условия минимума $f_0(x)$ при ограничениях* (4.2), *необходимо и достаточно выполнения равенства $p(x) = 0$.*

Доказательство. Пусть точка x удовлетворяет (4.2) и в ней выполняются необходимые условия минимума для $f_0(x)$. Тогда существуют такие числа $u^i \geqslant 0$, $i \in I$, что

$$f_0'(x) + \sum_{i \in I} u^i f_i'(x) = 0,\quad u^i f_i(x) = 0,\quad i \in I. \qquad (4.8)$$

Если x удовлетворяет (4.2), то $F(x) = 0$ и поэтому $I_0(x)$ совпадает с множеством тех i, для которых $f_i(x) = 0$. Кроме того, в силу второго соотношения (4.8) $u^i = 0$, если $f_i(x) < 0$, т.е. если $i \notin I_0(x)$. Поэтому, учитывая, что $I_\delta(x) \supseteq I_0(x)$, (4.8) можно переписать в виде

$$f_0'(x) + \sum_{i \in I_\delta(x)} u^i f_i'(x) = 0,\quad u^i f_i(x) = 0,\quad i \in J_\delta(x).$$

Но сопоставление последних соотношений с (4.6) показывает, что вектор $p = 0$ есть решение задачи (4.4), ибо при $p = 0$ удовлетворяются все ограничения (4.4) (так как (4.2) удовлетворяются), а выполнение соотношений (4.6) при $p = 0$ есть необходимое и достаточное условие того, чтобы вектор $p = 0$ был решением (4.4).

Пусть теперь $p(x) = 0$. Это значит, что ограничения задачи (4.4) удовлетворяются при $p = 0$, т.е. $f_i(x) \leqslant 0$, $i \in I_\delta(x)$. Так как для $i \notin I_\delta(x)$

$$f_i(x) \leqslant F(x) - \delta \leqslant f_j(x) \leqslant 0,$$

где $j \in I_\delta(x)$, то точка x удовлетворяет всем ограничениям (4.2). Кроме того, при $p = 0$ соотношения (4.6) переходят в (4.8), если положить $u^i = 0$, $i \notin I_\delta(x)$. Таким образом, необходимые условия минимума $f_0(x)$ при ограничениях (4.2) также удовлетворяются, что завершает доказательство.

Оценим теперь изменение всех входящих в задачу функций при сдвиге из точки x_k в направлении p_k. Для $i \in I_\delta(x_k)$, используя формулу Тейлора, получаем

$$f_i(x_k + \alpha p_k) = f_i(x_k) + \alpha(p_k,\ f_i'(x_k)) + \alpha(p_k,\ f_i'(\theta_i) - f_i'(x_k)),$$

где $\theta_i = x_k + \alpha\xi_i p_k$, $0 \leqslant \xi_i \leqslant 1$. Так как p_k – решение (4.4) при $x = x_k$, то

$$f_i(x_k + \alpha p_k) \leqslant f_i(x_k) - \alpha f_i(x_k) + \alpha^2 \|p_k\|^2 L \leqslant$$
$$\leqslant (1 - \alpha) f_i(x_k) + \alpha^2 \|p_k\|^2 L. \qquad (4.9)$$

(При выводе (4.9) мы воспользовались тем, что градиенты $f_i(x)$ удовлетворяют условию Липшица.)

Для $i \notin I_\delta(x)$

$$f_i(x_k + \alpha p_k) = f_i(x_k) + \alpha(p_k,\ f_i'(\theta_i)) \leqslant F(x_k) - \delta + \alpha K \| p_k \|, \tag{4.10}$$

где K — величина, ограничивающая $\| f_i'(x) \|$ в Ω_N.

Так как

$$(1 - \alpha) F(x_k) \geqslant F(x_k) - \delta + \alpha K \| p_k \|$$

для

$$\alpha \leqslant 1,\ 0 \leqslant \alpha \leqslant \frac{\delta}{F(x_k) + K \| p_k \|}, \tag{4.11}$$

то из (4.9) и (4.10) следует, что для всех i имеет место неравенство

$$f_i(x_k + \alpha p_k) \leqslant (1 - \alpha) F(x_k) + \alpha^2 L \| p_k \|^2. \tag{4.12}$$

Аналогично предыдущим оценкам получаем

$$f_0(x_k + \alpha p_k) = f_0(x_k) + \alpha(p_k,\ f_0'(x_k)) + \alpha(p_k,\ f_0'(\theta_0) - f_0(x_k)),$$

$$\theta_0 = x_k + \alpha \xi_0 p_k,\ 0 \leqslant \xi_0 \leqslant 1.$$

Воспользовавшись (4.7) и условием Липшица для градиентов, получаем

$$f_0(x_k + \alpha p_k) \leqslant f_0(x_k) + \alpha \Big(\sum_{i \in I_\delta(x_k)} u^i(x_k)\ f_i(x_k) \Big) -$$

$$- \alpha \| p_k \|^2 + \alpha^2 L \| p_k \|^2.$$

Отсюда и из (4.12) следует, что

$$f_0(x_k + \alpha p_k) + N F(x_k + \alpha p_k) \leqslant f_0(x_k) +$$

$$+ NF(x_k) + \alpha \Big(\sum_{i \in I_\delta(x_k)} u^i(x_k) f_i(x_k) - NF(x_k) \Big) - \alpha \| p_k \|^2 +$$

$$+ \alpha^2 (N + 1) L \| p_k \|^2. \tag{4.13}$$

Вспомним теперь, что $u^i(x_k) \geqslant 0$, $F(x_k) \geqslant 0$ и

$$\sum_{i \in I_\delta(x_k)} u^i(x_k) \leqslant N.$$

Поэтому

$$\sum_{i \in I_\delta(x_k)} u^i(x_k) f_i(x_k) - NF(x_k) \leqslant 0.$$

Но тогда (4.13) перепишется в виде

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant$$

$$\leqslant f_0(x) + NF(x_k) - \alpha \| p_k \|^2 (1 - \alpha(N + 1) L).$$

Если

$$0 \leqslant \alpha \leqslant \frac{1 - \epsilon}{(N + 1) L}, \tag{4.14}$$

то

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant f_0(x) + NF(x_k) - \alpha\epsilon \| p_k \|^2. \tag{4.15}$$

Итак, ссли

$$0 \leqslant \alpha \leqslant \bar{\alpha}_k,$$

$$\bar{\alpha}_k = \min\left(1, \frac{\delta}{F(x_k) + K \| p_k \|}, \frac{1-\epsilon}{(N+1)L}\right),$$

то выполняется неравенство (4.15). Но это означает, что неравенство (4.5) будет выполнено после конечного числа проб $\alpha = 2^{-i}$, $i = 0, 1, \ldots$, и при этом будет иметь место неравенство

$$\alpha_k > \bar{\alpha}_k / 2. \tag{4.16}$$

Докажем теперь следующую теорему о сходимости процесса.

Т е о р е м а 4.1. *Если выполнены сделанные в п.* 1 *предположения, то процесс обладает следующими свойствами*:

а) $F(x_k) \to 0$ *при* $k \to \infty$;

б) *в любой предельной точке* x_* *последовательности* x_k, $k = 0, 1, \ldots$, *выполняются неравенства* (4.2) *и необходимые условия минимума* $f_0(x)$ *при ограничениях* (4.2).

З а м е ч а н и е. Стремление $F(x_k)$ к нулю означает, что последовательность x_k все более точно удовлетворяет ограничениям (4.2).

Д о к а з а т е л ь с т в о. Все точки x_k принадлежат области Ω_N, так как функция $f_0(x) + NF(x)$ в силу (4.15) убывает от шага к шагу. Более того, так как Ω_N — компактное множество, то $f_0(x) + NF(x)$ ограничена на этом множестве, ибо эта функция непрерывна. Отсюда следует, что

$$\alpha_k \| p_k \|^2 \to 0 \quad \text{при} \quad k \to \infty, \tag{4.17}$$

ибо в противном случае $f_0(x) + NF(x)$ неограниченно убывает вдоль последовательности x_k.

Докажем, что $p_k \to 0$. Действительно, если $p_k \not\to 0$, то из (4.17) следует, что $\alpha_k \to 0$ вдоль некоторой подпоследовательности индексов k. Но из (4.16) и выражения для $\bar{\alpha}_k$ в этом случае следует, что для больших k

$$\alpha_k \geqslant \frac{1}{2}\bar{\alpha}_k = \frac{\delta}{2(F(x_k) + K \| p_k \|)},$$

поэтому должна стремиться к нулю правая часть последнего неравенства. Так как $F(x)$ — непрерывная функция на компактном множестве Ω_N, то $F(x)$ ограничена сверху и выражение $\delta/(F(x_k) + K \| p_k \|)$ может стремиться к нулю, лишь если $\| p_k \| \to +\infty$. Но из (4.6)

$$\| p(x_k) \| = \| f_0'(x_k) + \sum_{i \in I_\delta(x_k)} u^i(x_k) f_i'(x_k) \| \leqslant K(N+1).$$

Таким образом, мы пришли к противоречию, предположив, что $p_k \not\to 0$.

По определению p_k выполняются соотношения

$$(f_i'(x_k), p_k) + f_i(x_k) \leqslant 0, \quad i \in I_\delta(x_k).$$

Поэтому

$$f_i(x_k) \leqslant -(f_i'(x_k), p_k) \leqslant K \| p_k \|, \quad i \in I_\delta(x_k).$$

Но $f_j(x_k) \leqslant f_i(x_k)$, $j \notin I_\delta(x_k)$, $i \in I_\delta(x_k)$. Отсюда

$$F(x_k) = \max_{i \in I} f_i(x_k) \leqslant K \| p_k \|.$$

Значит, $F(x_k) \to 0$ при $k \to \infty$, ибо $F(x_k) \geqslant 0$. Далее, положим $u^i(x) = 0$, $i \notin I_\delta(x)$. Тогда вдоль последовательности x_k соотношения (4.6) можно переписать в виде

$$\begin{aligned} &f_0'(x_k) + p_k^* + \sum_{i \in I} u^i(x_k) f_i'(x_k) = 0, \\ &u^i(x_k)((f_i'(x_k), p_k) + f_i(x_k)) = 0, \quad i \in I. \end{aligned} \tag{4.18}$$

Пусть теперь x_* – предельная точка последовательности $\{x_k\}$. Так как $x_k \in \Omega_N$, Ω_N компактно, то предельные точки всегда существуют. Без ограничения общности мы можем считать, что $x_k \to x_*$. Кроме того, так как $u^i(x) \geqslant 0$, $i \in I$, и их сумма ограничена, то можно считать, что $u^i(x_k) \to u^i$ при $k \to \infty$.

Переходя к пределу в (4.18), получаем

$$f_0'(x_*) + \sum_{i \in I} u^i f_i'(x_*) = 0, \quad u^i f_i(x_*) = 0, \ i \in I.$$

Кроме того, $u^i \geqslant 0$, ибо $u^i(x_k) \geqslant 0$, а точка x_* удовлетворяет всем ограничениям (4.2). Действительно, $f_i(x_k) \leqslant F(x_k)$ и $F(x_k) \to 0$, откуда с помощью предельного перехода получаем, что $f_i(x_*) \leqslant 0$. Тем самым мы убедились, что необходимые условия минимума в точке x_* выполняются. Теорема доказана.

Следствие. *Если единственной точкой, в которой выполняются необходимые условия минимума, является точка минимума, то порождаемая алгоритмом последовательность сходится в точке минимума* $f_0(x)$ *при ограничениях* (4.2).

Действительно, в этом случае в силу теоремы 4.1 единственной предельной точкой последовательности x_k может быть только точка минимума.

4. Вычислительные аспекты. Основной операцией, требующей значительных вычислений при реализации алгоритма на каждом шаге, является решение задачи (4.4). Это задача квадратичного программирования. При выборе метода решения этой задачи необходимо учитывать, что, поскольку задача (4.4) является вспомогательной, ее решение необходимо получить за конечное число шагов. Кроме того, поскольку константа N заранее, вообще говоря, неизвестна, для контроля правильности выбора N при решении задачи (4.4) удобно получить и соответствующие множители Лагранжа $u^i(x)$. С этих позиций представляется целесообразным при решении (4.4) перейти к двойственной задаче, а ее уже решать методом сопряженных градиентов, изложенным в п. 8 § 3.

Построим двойственную задачу для задачи (4.4). Согласно § 2 целевая функция двойственной задачи имеет вид

$$\varphi(u) = \min_p \left[(f_0'(x), p) + \tfrac{1}{2} \| p \|^2 + \sum_{i \in I_\delta(x)} u^i((f_i'(x), p) + f_i(x))\right]. \tag{4.19}$$

Приравнивая нулю производные по p от правой части последнего равенства,

находим, что минимум достигается при

$$p = - f_0'(x) - \sum_{i \in I_\delta(x)} u^i f_i'(x). \qquad (4.20)$$

Таким образом, точка p однозначно определяется вектором u с компонентами u^i, $i \in I_\delta(x)$.

Подставив (4.20) в правую часть (4.19), получаем

$$\varphi(u) = - \tfrac{1}{2} \| f_0'(x) + \sum_{i \in I_\delta(x)} u^i f_i'(x) \|^2 + \sum_{i \in I_\delta(x)} u^i f_i(x). \qquad (4.21)$$

Итак, мы вычислили целевую функцию двойственной задачи. Сама двойственная задача состоит теперь в максимизации $\varphi(u)$ при ограничениях $u^i \geqslant 0$, $i \in I_\delta(x)$.

Таким образом, получилась задача максимизации квадратичной формы при простых ограничениях, которую удобно решать методом сопряженных градиентов (п. 8 § 3). В результате решения мы получим множители Лагранжа $u^i(x)$ – решение двойственной задачи, а согласно изложенному в § 3 подстановка $u^i(x)$ в (4.20) дает вектор $p(x)$ – решение исходной задачи.

Другая проблема – это выбор констант N, δ. Величина N, вообще говоря, неизвестна. Выбирать ее слишком большой невыгодно, так как в силу формулы (4.14) это может повести к значительному дроблению шага. Поэтому целесообразно оценивать ее по ходу работы алгоритма. Например, если на каком-то шаге оказалось, что

$$N \leqslant \sum_{i \in I_\delta(x_k)} u^i(x_k), \qquad (4.22)$$

то N следует изменить, заменив на

$$N = 2 \sum_{i \in I_\delta(x_k)} u^i(x_k). \qquad (4.23)$$

Практический опыт показывает, что такая коррекция приводит к успеху. Кроме того, из теоретических соображений ясно, что если x_k достаточно близки к предельной точке, то в регулярном случае $u^i(x_k)$ будут близки к множителям Лагранжа в точке x_*, являющейся решением задачи, и поэтому формула (4.23) приведет к успеху. Подробнее о поведении множителей $u^i(x_k)$ будет сказано ниже.

Что касается величины δ, то ее следует уменьшать в случае, если вспомогательная задача (4.4) окажется неразрешимой на каком-то шаге. Однако опыт решения задач показал, что на практике δ следует брать как можно больше, учитывая при решении вспомогательной задачи все ограничения исходной задачи, если только это позволяет память машины.

5. Некоторые обобщения. В начале этого параграфа уже говорилось о том, что в случае наличия ограничений типа равенств, т.е. когда ограничения имеют вид (4.1), задача сводится к виду (4.2) путем замены каждого равенства двумя неравенствами. Таким образом, алгоритм применим и к

общей задаче (4.1). При этом надо только учитывать, что если при некотором x

$$f_i(x) \geqslant F(x) - \delta, \quad -f_i(x) \geqslant F(x) - \delta, \tag{4.24}$$

где $i \in I_0$, то в систему (4.4) входят два неравенства

$$(f_i'(x), p) + f_i(x) \leqslant 0, \quad -(f_i'(x), p) - f_i(x) \leqslant 0, \tag{4.25}$$

которые эквивалентны одному равенству

$$(f_i'(x), p) + f_i(x) = 0. \tag{4.26}$$

Поэтому целесообразно при решении вспомогательной задачи этот факт учитывать и заменять в (4.4) пары неравенств вида (4.25) на одно равенство (4.26). При переходе к двойственной задаче это приведет к тому, что соответствующий множитель u^i будет иметь произвольный знак, что, однако, не нарушает возможности применения алгоритма сопряженных градиентов (п. 8 § 3).

Допустим теперь, что в исходной задаче кроме ограничений (4.2) имеется ограничение, заданное условием, что точка x принадлежит некоторому множеству X, имеющему простую структуру. В этом случае целесообразно, чтобы получаемые приближения лежали в множестве X. Укажем, как в этом случае модифицируется алгоритм. Как и ранее, без ограничения общности рассмотрим только случай наличия неравенств в ограничениях.

Итак, пусть требуется минимизировать $f_0(x)$, $x \in E^n$, при ограничениях

$$f_i(x) \leqslant 0, \ i \in I, x \in X, \tag{4.27}$$

где I – конечное множество индексов, X – выпуклое замкнутое множество. Предполагается, что существует такой индекс i, что $f_i(x) = 0$.

Предположим, что существуют такие константы $N > 0$, $\delta > 0$, что выполнены следующие условия:

а) множества

$$\Omega_N = \{x: \ f_0(x) + NF(x) \leqslant C_0, \ x \in X\},$$

$$C_0 = f_0(x_0) + NF(x_0),$$

ограничены, и начальное приближение x_0 принадлежит X;

б) градиенты функций $f_i(x)$, $i \in \{0\} \cup I$, в Ω_N удовлетворяют условию Липшица, т.е.

$$\| f_i'(x_1) - f_i'(x_2) \| \leqslant L \| x_1 - x_2 \|;$$

в) задача

$$\begin{aligned} &\min (f_0'(x), p) + \tfrac{1}{2} \| p \|^2, \\ &(f_i'(x), p) + f_i(x) \leqslant 0, \ i \in I_\delta(x), \ x + p \in X, \end{aligned} \tag{4.28}$$

разрешима относительно p при любых $x \in \Omega_N$, и существуют такие множители Лагранжа $u^i(x)$, $i \in I_\delta(x)$, что

$$\sum_{i \in I_\delta(x)} u^i(x) \leqslant N.$$

Замечание. Напомним, что множители Лагранжа для задачи (4.28) – это такие неотрицательные числа, что выполняются условия

$$(f_0'(x), p(x)) + (p(x), p(x)) +$$

$$+ \sum_{i \in I_\delta(x)} u^i(x)\,[(f_i'(x),\ p(x)) + f_i(x)] \leqslant (f_0'(x), p) + (p(x), p) +$$

$$+ \sum_{i \in I_\delta(x)} u^i(x)\,[(f_i'(x),\ p) + f_i(x)] \tag{4.29}$$

для всех p, удовлетворяющих условию

$$x + p \in X. \tag{4.30}$$

Кроме того,

$$u^i(x)\,[(f_i'(x),\ p(x)) + f_i(x)] = 0,\quad i \in I_\delta(x). \tag{4.31}$$

Таким образом, условие в) предполагает не только разрешимость вспомогательной задачи (4.28), но и то, что в точке минимума $p = p(x)$ выполняются необходимые и достаточные условия, требуемые теоремой Куна – Таккера.

Алгоритм решения задачи (4.27) теперь строится точно так же, как это изложено в п. 2. Только теперь в качестве p_k берут вектор $p(x_k)$, являющийся решением новой вспомогательной задачи (4.28).

Покажем сходимость алгоритма, т.е. справедливость выводов теоремы 4.1, а также что $x_k \in X$ при всех k. Из последнего утверждения, в частности, следует, что всякая предельная точка последовательности x_k лежит в X. Поскольку доказательство сходимости лишь некоторыми деталями отличается от доказательства теоремы 4.1, то нет нужды приводить это доказательство подробно. Отметим лишь основные отличительные детали.

Во-первых, так как $x_k + p_k \in X$ и X выпукло, то $x_k + \alpha p_k \in X$ при всех α, лежащих между 0 и 1. Поэтому, если $x_k \in X$, то и $x_{k+1} \in X$. А так как $x_0 \in X$ по предположению, то вся последовательность $\{x_k\}_{k=0}^{\infty}$ лежит в X.

Во-вторых, из (4.29) – (4.31) при $p = 0$ получается, что

$$(f_0'(x),\ p(x)) + \|p(x)\|^2 \leqslant \sum_{i \in I_\delta(x)} u^i(x) f_i(x),$$

т.е.

$$(f_0'(x),\ p(x)) \leqslant \sum_{i \in I_\delta(x)} u^i f_i(x) - \|p(x)\|^2. \tag{4.32}$$

Это неравенство заменяет соотношение (4.7), использованное при получении оценки (4.13). Все остальные выкладки при получении оценок остаются без изменения.

Наконец, если $p(x_*) = 0$, то из (4.29) – (4.31) следует, что выполняются условия

$$\begin{aligned} &(f_0'(x_*),\ p) + \sum_{i \in I_\delta(x)} u^i(x_*)\,(f_i'(x_*),\ p) \geqslant 0, \\ &x_* + p \in X,\quad u^i(x_*) f_i(x_*) = 0,\quad i \in I_\delta(x_*). \end{aligned} \tag{4.33}$$

Кроме того, из (4.28) в этом случае следует, что

$$f_i(x_*) \leqslant 0, \quad i \in I_\delta(x_*), \quad x_* \in X,$$

и, кроме того, очевидно, что

$$f_i(x_*) < 0, \quad i \notin I_\delta(x_*).$$

Таким образом, точка x_* удовлетворяет всем ограничениям (4.27), а условия (4.33) показывают, что в этой точке выполнены необходимые условия экстремума.

Итак, как и ранее, мы показали, что если $p(x_*) = 0$, то в точке x_* выполняются необходимые условия экстремума. Нетрудно показать и обратное, так что условие $p(x) = 0$ является необходимым и достаточным условием того, чтобы точка x была подозрительной на экстремум.

Доказательство того, что каждая предельная точка x_* последовательности x_k, $k = 0, 1, \ldots$, удовлетворяет необходимым условиям экстремума, проводится точно так же, как при доказательстве теоремы 4.1, путем предельного перехода от соотношений (4.29) – (4.31), удовлетворяющихся в точках x_k, к соотношениям (4.33) в предельной точке.

6. Задача линейного программирования. Пусть теперь в задаче (4.2) все функции $f_0(x)$, $f_i(x)$, $i \in I$, линейны. Таким образом, получается задача линейного программирования. Хотя изложенный выше алгоритм имеет наибольшее значение для нелинейного случая, его применение для задачи линейного программирования также не лишено смысла. В частности, если множество I содержит большое число индексов, то получается задача линейного программирования с большим числом ограничений. В то же время при малом δ вспомогательная задача (4.4) будет иметь лишь небольшое число ограничений, так что общая задача сводится к решению серии более простых задач. Кроме того, в отличие от симплекс-метода, предлагаемый метод не будет накапливать вычислительную погрешность, так как не преобразует исходную матрицу ограничений от шага к шагу.

Для задачи линейного программирования условия а), в) (условие б) удовлетворяется автоматически) основного предположения являются излишне жесткими для сходимости алгоритма. Мы не будем здесь останавливаться на условиях сходимости для задачи линейного программирования, поскольку основная цель – получение алгоритма для нелинейного случая. Ниже будет показано, что по крайней мере при выполнении сделанных предположений а), в) для задачи линейного программирования алгоритм будет сходиться за конечное число шагов. Этот факт в определенной степени будет характеризовать нам скорость сходимости алгоритма.

Т е о р е м а 4.2. *Пусть выполнены предположения а), в) п. 1 и все функции* $f_0(x)$, $f_i(x)$, *определяющие задачу* (4.2), *имеют вид* $f_i(x) = (a_i, x) - b_i$. *Тогда алгоритм п. 2 сходится за конечное число шагов.*

Д о к а з а т е л ь с т в о. Заметим сразу, что в рассматриваемом случае шаг α_k равен 1 для достаточно больших k. Действительно, поскольку все $f_i(x)$ линейны, то константа Липшица L просто равна нулю. Поэтому из

формулы п. 3 для $\bar{\alpha}_k$ следует, что

$$\bar{\alpha}_k = \min\left(1,\ \frac{\delta}{F(x_k)+K\|p_k\|},\ \frac{1-\epsilon}{(N+1)L}\right) =$$

$$= \min\left(1,\ \frac{\delta}{F(x_k)+K\|p_k\|}\right). \tag{4.34}$$

Но выше было доказано, что $F(x_k) \to 0$, $\|p_k\| \to 0$. Поэтому для достаточно больших k имеем $\delta/(F(x_k) + K\|p_k\|) \geqslant 1$ и $\bar{\alpha}_k = 1$. Но $\bar{\alpha}_k$ построено так, что неравенство (4.15) выполняется при $\alpha = \bar{\alpha}_k$. Так как выбор α_k на каждой итерации начинается путем деления пополам $\alpha = 1$, то отсюда следует, что неравенство (4.5), определяющее выбор α_k, будет сразу же удовлетворено без дополнительных делений и шаг α_k будет просто равен 1.

Пусть теперь x_* – какая-либо предельная точка последовательности x_k, полученной в результате работы алгоритма. Как уже известно, эта точка является решением задачи (4.2), ибо она удовлетворяет всем ограничениям задачи, а кроме того, в ней согласно теореме 4.1 выполняются необходимые условия минимума, которые в рассматриваемом случае задачи линейного программирования оказываются и достаточными.

Положим

$$I_0(x_*) = \{i \in I:\ f_i(x_*) = 0\}. \tag{4.35}$$

Тогда $f_i(x_*) < 0$ для $i \notin I_0(x_*)$, так что

$$\epsilon_0 = \max_{i \notin I_0(x_*)} f_i(x_*) < 0. \tag{4.36}$$

Для упрощения дальнейших обозначений без ограничения общности будем считать, что вся последовательность x_k сходится к x_*.

Рассмотрим вспомогательную задачу (4.4) в точках последовательности x_k:

$$\min\,(f_0'(x_k),\ p) + \tfrac{1}{2}\|p\|^2, \tag{4.37}$$

$$(f_i'(x_k),\ p) + f_i(x_k) \leqslant 0,\quad i \in I_\delta(x),$$

где $p_k = p(x_k)$ – решение (4.4). Обозначим соответствующие множители Лагранжа через u_k^i, $i \in I_\delta(x_k)$, так что

$$u_k^i\,[(f_i'(x_k),\ p_k) + f_i(x_k)] = 0. \tag{4.38}$$

Покажем теперь, что $I_0(x_*) \subset I_\delta(x_k)$ для всех достаточно больших k. Действительно, если $i \notin I_\delta(x_k)$, то

$$f_i(x_k) < F(x_k) - \delta.$$

Переходя к пределу по k и учитывая, что $F(x_k) \to 0$, получаем $f_i(x_*) \leqslant -\delta$, что противоречило бы принадлежности i множеству $I_0(x_*)$.

Далее, обозначим

$$\tilde{I}(x_k) = \{i \in I_\delta(x_k):\ u_k^i > 0\}.$$

Следующее утверждение состоит в том, что для больших индексов k

$$\widetilde{I}(x_k) \subseteq I_0(x_*). \tag{4.39}$$

Действительно, если $i \notin I_0(x_*)$, то $f_i(x_*) \leqslant \epsilon_0$. Так как $p_k \to 0$, $f_i'(x_k)$ ограничены и $x_k \to x_*$, то при больших k

$$|(f_i'(x_k),\ p_k)| \leqslant -\epsilon_0/4,\ \ f_i(x_k) \leqslant \epsilon_0/2$$

и поэтому

$$(f_i'(x_k),\ p_k) + f_i(x_k) \leqslant \epsilon_0/4 < 0.$$

Поэтому, если $u_k^i > 0$, то

$$u_k^i\ [(f_i'(x_k),\ p_k) + f_i(x_k)] < 0,$$

что противоречит (4.38).

З а м е ч а н и е. Все проведенные рассуждения не использовали линейность $f_i(x)$, и поэтому утверждения, что $I_0(x_*) \subset I_\delta(x_k)$ и $\widetilde{I}(x_k) \subset I_0(x_*)$, верны в общем случае нелинейной задачи. Они будут использованы в дальнейшем.

Как было показано в п. 4 этого параграфа, задача, двойственная к вспомогательной задаче (4.37), состоит в максимизации функции (4.21) при ограничениях $u^i \geqslant 0$, $i \in I_\delta(x_k)$. При этом множители Лагранжа u_k^i являются решением двойственной задачи и имеет место равенство между оптимальными значениями в прямой и двойственной задачах, т.е.

$$(f_0'(x_k),\ p_k) + \tfrac{1}{2}\|p_k\|^2 = -\tfrac{1}{2}\|f_0'(x_k) + \sum_{i \in I_\delta(x_k)} u_k^i f_i'(x_k)\|^2 + \sum_{i \in I_\delta(x_k)} u_k^i f_i(x_k).$$

Так как $p_k \to 0$, то левая часть последнего соотношения стремится к нулю, а значит,

$$-\tfrac{1}{2}\|f_0'(x_k) + \sum_{i \in I_\delta(x_k)} u_k^i f_i'(x_k)\|^2 + \sum_{i \in I_\delta(x_k)} u_k^i f_i(x_k) \to 0. \tag{4.40}$$

Заметим теперь, $u_k^i > 0$, только если $i \in \widetilde{I}(x_k)$. Кроме того, $f_i(x) = (a_i, x) - b_i$, $i \in \{0\} \cup I$, так что $f_i'(x) = a_i$ и не зависит от x. Поэтому (4.40) может быть переписано в виде

$$-\tfrac{1}{2}\|a_0 + \sum_{i \in \widetilde{I}(x_k)} u_k^i a_i\|^2 + \sum_{i \in \widetilde{I}(x_k)} u_k^i f_i(x_k) \to 0.$$

Но $\widetilde{I}(x_k) \subset I_0(x_k)$, как было показано выше, и поэтому $f_i(x_k) \to f_i(x_*) = 0$, ибо $f_i(x_*) = 0$ для $i \in I_0(x_*)$ по определению. Поэтому

$$-\tfrac{1}{2}\|a_0 + \sum_{i \in \widetilde{I}(x_k)} u_k^i a_i\|^2 \to 0.$$

Но

$$-\tfrac{1}{2}\|a_0 + \sum_{i \in \widetilde{I}(x_k)} u_k^i a_i\|^2 \leqslant$$

$$\leqslant \max_{u^i \geqslant 0,\ i \in \widetilde{I}(x_k)} -\tfrac{1}{2}\|a_0 + \sum_{i \in \widetilde{I}(x_k)} u^i a_i\|^2 \leqslant 0. \tag{4.41}$$

Положим

$$\omega(\bar{I}) = \max_{u^i \geqslant 0,\ i \in \bar{I}} - \| a_0 + \sum_{i \in \bar{I}} u^i a_i \|^2 ;$$

$\omega(\bar{I})$ есть функция, определенная на множестве индексов $\bar{I}, \bar{I} \subset I$. Так как $\bar{I} \subset I$, то эта функция может принимать лишь конечное число значений. Из (4.41) следует, что $\omega(\tilde{I}(x_k)) \to 0$. Но это означает, что

$$\omega(\tilde{I}(x_k)) = 0 \tag{4.42}$$

для всех достаточно больших k, ибо, как только что было сказано, $\omega(\bar{I})$ принимает лишь конечное число значений. Выберем теперь k настолько большим, что $\alpha_k = 1$, выполняется условие (4.42) и $\tilde{I}(x_k) \subset I_0(x_*)$. Так как $\alpha_k = 1$, то $x_{k+1} = x_k + p_k$. Так как $x_k \to x_*, p_k \to 0$, то можно считать, что

$$f_i(x_{k+1}) \leqslant \epsilon_0 / 2 < 0, \quad i \notin I_0(x_*). \tag{4.43}$$

Рассмотрим снова вспомогательную задачу (4.37). Так как p_k удовлетворяет ограничениям (4.37), а $f_i(x)$ линейны, то

$$f_i(x_{k+1}) = (f_i'(x), p_k) + f_i(x_k) \leqslant 0 \tag{4.44}$$

для $i \in I_\delta(x_k)$, а значит, и для $i \in I_0(x_*)$, ибо $I_0(x_*) \subset I_\delta(x_k)$. Тем самым показано, что x_{k+1} удовлетворяет всем ограничениям задачи (4.2).

Покажем, что на самом деле x_{k+1} — решение задачи (4.2). Действительно, из (4.38) и определения множества $\tilde{I}(x_k)$ следует, что

$$f_i(x_{k+1}) = 0, \quad i \in \tilde{I}(x_k). \tag{4.45}$$

Но (4.42) означает, что найдутся такие числа $u_0^i \geqslant 0, i \in \tilde{I}(x_k)$, что

$$a_0 + \sum_{i \in \tilde{I}(x_k)} u_0^i a_i = 0. \tag{4.46}$$

Положив теперь $u_0^i = 0, \ i \notin \tilde{I}(x_k)$, мы получим, что нашлись такие числа $u_0^i \geqslant 0$, что выполнены условия

$$a_0 + \sum_{i \in I} u_0^i a_i = 0, \ u_0^i f_i(x_{k+1}) = 0.$$

Но последние соотношения (см. § 2) являются необходимыми и достаточными условиями того, чтобы точка x_{k+1} была решением задачи линейного программирования.

Таким образом, алгоритм действительно приводит к решению за конечное число шагов, что и требовалось доказать.

7. Метод линеаризации при ограничениях типа равенств. Рассмотрим задачу

$$\min \{ f_0(x) : f_i(x) = 0, i = 1, \ldots, m \}. \tag{4.47}$$

Применительно к этой задаче

$$F(x) = \max \{ 0, f_1(x), \ldots, f_m(x), -f_1(x), \ldots, -f_m(x) \} = \max_{1 \leqslant i \leqslant m} |f_i(x)|.$$

Выберем $\delta = +\infty$, так что вспомогательная задача квадратичного программирования включает все ограничения и приобретает вид

$$\min_p \{ (f_0'(x), p) + \tfrac{1}{2} \| p \|^2 : \ (f_i'(x), p) + f_i(x) = 0, i = 1, \ldots, m \}. \tag{4.48}$$

К этой задаче можно применить результаты п. 4§3. Формулы (3.13) и (3.4) с учетом того, что в рассматриваемом случае

$$C = I_n,\ d = (f_0'(x))^*,\ b = -f(x),\ A = f'(x),$$

где $f(x) \in \mathbf{R}^m$, $f'(x)$ – матрица $m \times n$ со строчками $f_i'(x)$, $i = 1, \ldots, m$, дают

$$u(x) = -(f'(x)(f'(x))^*)^{-1}\,|-f(x) + f'(x)(f_0'(x))^*|, \tag{4.49}$$

$$p(x) = -|(f_0'(x))^* + (f'(x))^* u(x)|. \tag{4.50}$$

Подставляя (4.49) в (4.50), получаем

$$p(x) = |(I_n - \Pi(x))(f_0'(x))^* + (f'(x))^*(f'(x)(f'(x))^*)^{-1} f(x)|, \tag{4.51}$$

где

$$\Pi(x) = (f'(x))^*(f'(x)(f'(x))^*)^{-1} f'(x). \tag{4.52}$$

Все приведенные выкладки согласно п. 4§3 справедливы, если градиенты $f_i'(x)$, $i = 1, \ldots, m$, линейно независимы, что мы выше неявно предполагали.

Формулы (4.49), (4.50) могут служить для явного вычисления $u(x)$ и $p(x)$. Однако, по-видимому, более экономный путь вычислений состоит в прямом решении задачи (4.48) путем перехода к двойственной:

$$\max_u \left\{ -\tfrac{1}{2} \| f_0'(x) + \sum_{i=1}^m u^i f_i'(x) \|^2 + \sum_{i=1}^m u^i f_i(x) \right\}. \tag{4.53}$$

При этом, так как в (4.48) участвуют лишь ограничения типа равенств, то нет ограничений на знак переменных и возможно прямое применение метода сопряженных направлений (п. 1§3).

Таким образом, алгоритм метода линеаризации в рассматриваемом случае приобретает следующий вид:

0. Пусть N – достаточно большое число, x_k уже построено.

1. Решаем задачу (4.53) при $x = x_k$ и находим множители $u_k = u(x_k)$. Подставляя их в (4.50), находим $p_k = p(x_k)$.

2. Делим пополам $\alpha = 1$ до удовлетворения неравенству

$$f_0(x_k + \alpha_k p_k) + NF(x_k + \alpha_k p_k) \leqslant f_0(x_k) + NF(x_k) - \epsilon\alpha_k \| p_k \|^2,$$

находим α_k.

3. Полагаем $x_{k+1} = x_k + \alpha_k p_k$ и возвращаемся к п. 1.

4. Критерий останова: $\| p_k \| = 0$.

8. Простые ограничения. Пусть теперь в задаче имеются только ограничения на изменения каждой координаты в отдельности, т.е. задача имеет вид

$$\min_x \{ f_0(x): a^j \leqslant x^j \leqslant b^j,\ j = 1, \ldots, n \}, \tag{4.54}$$

причем значения $a^j = -\infty$, $b^j = +\infty$ не исключаются.

Для такой задачи целесообразно включить во вспомогательную задачу все ограничения. При этом

$$F(x) = \max \{ 0, a^1 - x^1, \ldots, a^n - x^n, x^1 - b^1, \ldots, x^n - b^n \},$$

а вспомогательная задача приобретает вид

$$\min \left\{ \sum_{j=1}^{n} \frac{\partial f_0}{\partial x^j} p^j + \frac{1}{2} \sum_{j=1}^{n} (p^j)^2 : a^j \leqslant x^j + p^j \leqslant b^j,\ j = 1, \ldots, n \right\}. \quad (4.55)$$

Так как теперь целевая функция представляет собой сумму членов, каждый из которых зависит от своей переменной, меняющейся независимо от остальных, то задача легко решается:

$$p^j(x) = \begin{cases} a^j - x^j, \text{ если } -\dfrac{\partial f_0}{\partial x^j} \leqslant a^j - x^j, \\ -\dfrac{\partial f_0}{\partial x^j}, \text{ если } a^j - x^j \leqslant -\dfrac{\partial f_0}{\partial x^j} \leqslant b^j - x^j, \\ b^j - x^j, \text{ если } -\dfrac{\partial f_0}{\partial x^j} \geqslant b^j - x^j. \end{cases} \quad (4.56)$$

Зная $p^j(x)$, нетрудно найти множители Куна – Таккера. Обозначим через u^j_-, u^j_+ множители, соответствующие неравенствам

$$x^j + p^j - b^j \leqslant 0, \quad -x^j - p^j + a^j \leqslant 0.$$

Тогда функция Лагранжа приобретает вид

$$\sum_{j=1}^{n} \left[\frac{1}{2} (p^j)^2 + \frac{\partial f_0}{\partial x^j} p^j + u^j_+ (a^j - x^j - p^j) + u^j_- (x^j + p^j - b^j) \right].$$

По определению согласно теореме 2.6 множители Куна – Таккера должны обладать следующими свойствами. Во-первых, минимум функции Лагранжа по p должен достигаться в решении $p(x)$. Отсюда путем дифференцирования получаем

$$\frac{\partial f_0}{\partial x^j} - u^j_+ + u^j_- + p^j = 0,\ j = 1, \ldots, n.$$

Во-вторых,

$$u^j_+ \geqslant 0,\ u^j_+ (a^j - x^j - p^j) = 0,$$

$$u^j_- \geqslant 0,\ u^j_- (x^j + p^j - b^j) = 0,\ j = 1, \ldots, n.$$

Так как $p = p(x)$ теперь известно, то из полученых соотношений и (4.56) легко определить u^j_+, u^j_-:

$$u^j_-(x) = 0,\ u^j_+(x) = \frac{\partial f}{\partial x^j} + p^j(x) = \frac{\partial f}{\partial x^j} + a^j - x^j, \text{ если } -\frac{\partial f}{\partial x^j} \leqslant a^j - x^j;$$

$$u^j_-(x) = u^j_+(x) = 0, \text{ если } a^j - x^j < -\frac{\partial f}{\partial x^j} < b^j - x^j;$$

$$u^j_+(x) = 0,\ u^j_-(x) = -\frac{\partial f}{\partial x^j} - p^j(x) = -\frac{\partial f}{\partial x^j} - b^j + x^j,$$

$$\text{если } -\frac{\partial f_0}{\partial x^j} \geqslant b^j - x^j.$$

Итак, в случае простых ограничений вспомогательная задача легко решается и также определяются множители Куна – Таккера.

Впрочем, если начальная точка x_0 удовлетворяет ограничениям $a \leqslant x_0 \leqslant b$, а по построению $a \leqslant x + p(x) \leqslant b$, то $a \leqslant x + \alpha p(x) \leqslant b$, $0 \leqslant \alpha \leqslant 1$, и поэтому $F(x + \alpha p(x)) = 0$, $0 \leqslant \alpha \leqslant 1$.

Отсюда следует, что для выбора α на каждом шаге алгоритма линеаризации по формуле (4.5) нет необходимости в знании константы N, а значит, нет необходимости вычислять множители Куна – Таккера.

В связи со сказанным при простых ограничениях алгоритм приобретает следующий вид :

0. Выбирается точка x_0 такая, что $a^j \leqslant x_0^j \leqslant b^j$, $j = 1, \ldots, n$.

1. Если точка x_k уже построена, то вычисляется вектор $p_k = p(x_k)$ по формулам (4.56).

2. Делим пополам $\alpha = 1$ до выполнения неравенства

$$f_0(x_k + \alpha p_k) \leqslant f_0(x_k) - \epsilon \alpha \| p_k \|^2.$$

Полагаем α_k равным полученному α' и $x_{k+1} = x_k + \alpha_k p_k$. Возвращаемся к шагу 1.

3. Критерий останова: $p_k = 0$.

Т е о р е м а 4.3. *Если числа a^j, b^j, $j = 1, \ldots, n$, конечны и градиент функции f_0 удовлетворяет условию Липшица, то во всякой предельной точке последовательности x_k, $k = 0, 1, \ldots$, удовлетворяются необходимые условия минимума.*

Доказательство получается прямой ссылкой на теорему 4.1. Действительно, в данном случае последовательность x_k не выходит за пределы ограниченной области, вспомогательная задача (4.55) всегда разрешима, а множители u_+^j, u_-^j ограничены, как это видно непосредственно из приведенных выше формул. Поэтому выполнены все предположения, при которых справедлива теорема 4.1.

9. Выбор параметров в методе линеаризации. Модифицированный алгоритм. Как видно из изложенного в пп. 1, 2, 3, для сходимости метода необходимо выбрать параметры N, δ, которые, вообще говоря, априори неизвестны. Поэтому необходимы какие-либо обоснованные критерии выбора либо изменения этих параметров.

Как уже говорилось выше, весь вычислительный опыт показывает, что параметр δ лучше всего брать большим, так чтобы с самого начала учитывались все ограничения. Поэтому мы ограничимся здесь случаем, когда $\delta = +\infty$.

Теперь вспомогательная задача будет иметь вид

$$\min\{(f_0'(x), p) + \tfrac{1}{2} \| p \|^2 : \ (f_i'(x), p) + f_i(x) \leqslant 0, \ i \in I, \ x + p \in M\}, \quad (4.57)$$

где M – выпуклое множество простой структуры.

Заметим существенную особенность вспомогательной задачи: ее решение, определяющее направление сдвига на итерации, не зависит от выбора N. Величина N влияет лишь на выбор шага α.

Метод линеаризации может быть интерпретирован и как метод минимизации штрафной функции

$$\Phi_N(x) = f_0(x) + N \max\{0, f_i(x) : \ i \in I\}$$

на множестве M, так как на каждом шаге происходит убывание этой функции. Казалось бы, если метод линеаризации сходится, то он должен приводить к точке x_*, которая является точкой минимума $\Phi_N(x)$ на M. Как показывает простой пример, это не так.

Действительно, пусть $f_0(x) = -x^2$, $f_1(x) = -x$, $f_2(x) = x - 1$, $M = \mathbf{R}$, т.е. решается задача

$$\min\{-x^2:\ -x \leqslant 0,\ x - 1 \leqslant 0\} = \min\{-x^2:\ 0 \leqslant x \leqslant 1,\ -\infty \leqslant x \leqslant +\infty\}.$$

Очевидно, что ее решение $x_* = 1$. Метод линеаризации, если он начинается с любой точки $x > 0$, $x \leqslant 1$, немедленно приводит к решению. В то же время штрафная функция

$$\Phi_N(x) = -x^2 + N \max\{0, -x, x - 1\}$$

имеет нижнюю грань, равную $-\infty$ при любом $N \geqslant 0$.

Таким образом, методом штрафных функций исходную задачу решить нельзя. В точке $x_* = 1$ для $\Phi_N(x)$, как легко проверить, удовлетворяются лишь необходимые условия минимума, однако она не является точкой минимума, а лишь точкой локального минимума.

Из сказанного следует, что величина N для невыпуклых задач может оказаться никак не связанной с методом штрафных функций. С другой стороны, если все входящие в задачу функции выпуклы, а N выбрано так, чтобы удовлетворить условия основного предположения п. 1, то

$$N > \sum_{i \in I} u_*^i \ ,$$

где u_* – вектор Куна – Таккера задачи (4.2), так как вектор u_* одновременно является таковым и для вспомогательной задачи (4.4) при $x = x_*$. Поэтому в силу следствия 1 теоремы 2.14 решение задачи (4.2) является одновременно и точкой минимума штрафной функции $\Phi_N(x)$. Итак, в выпуклом случае метод линеаризации может рассматриваться одновременно и как некоторая модификация метода штрафных функций.

Укажем теперь алгоритм изменения N, который при разумных предположениях гарантирует сходимость метода линеаризации.

Модифицированный алгоритм метода линеаризации. Пусть точка x_0 и число $N_0 > 0$ выбраны. Положим $C_0 = f_0(x) + N_0 F(x_0)$. Если x_k, N_k, C_k уже построены, то переход к их следующим значениям происходит по правилу:

1. Решаем вспомогательную задачу (4.57) при $x = x_k$ и определяем $p_k = p(x_k)$ и множители Лагранжа $u_k^i = u^i(x_k)$, $i \in I$.

2. Если $N_k > \sum_{i \in I} u_k^i$, то

$$N_{k+1} = N_k, \quad C_{k+1} = C_k, \quad x_{k+1} = x_k + \alpha_k p_k,$$

где α_k выбирается делением пополам $\alpha = 1$ до выполнения неравенства (4.5) с $N = N_{k+1}$.

3. Если $N_k \leqslant \sum_{i \in I} u_k^i$, то

$$N_{k+1} = 2 \sum_{i \in I} u_k^i.$$

4. Если $f_0(x_k) + N_{k+1} F(x_k) \leqslant C_k$, то

$$C_{k+1} = C_k, \; x_{k+1} = x_k + \alpha_k p_k,$$

где α_k выбирается, как и выше.

5. Если $f(x_k) + N_{k+1} F(x_k) > C_k$, то

$$C_{k+1} = f(x_k) + N_{k+1} F(x_k), \quad x_{k+1} = x_0.$$

Т е о р е м а 4.4. *Пусть выполнены следующие условия:*

а) *в любой компактной области* $f_i(x), i \in \{0\} \cup I$, *удовлетворяют условию Липшица с константой, которая может зависеть от области;*

б) *в любой компактной области вспомогательная задача* (4.57) *разрешима и сумма множителей Лагранжа ограничена константой, которая зависит от области;*

в) $\inf\limits_x \{f_0(x): \; x \in M\} = \mu > -\infty$;

г) *множество*

$$W_\alpha = \{x: \; F(x) \leqslant \alpha, \; x \in M\}$$

ограничено.

Тогда модифицированный метод линеаризации сходится в том смысле, что любая предельная точка порожденной им последовательности $\{x_k\}$ *удовлетворяет ограничениям*

$$f_i(x) \leqslant 0, \; i \in I, \; x \in M,$$

и в ней выполняются необходимые условия минимума.

Д о к а з а т е л ь с т в о. Заметим, что по построению алгоритма N_k может только увеличиваться. При каждом увеличении $N_{k+1} \geqslant 2N_k$, так что если увеличение происходит бесконечное число раз, то $N_k \to +\infty$. Далее, по построению $x_k \in M$ при всех k.

Положим

$$\alpha = F(x_0) + \frac{f(x_0) - \mu}{N_0}, \tag{4.58}$$

и пусть N_α – константа, которая по предположению б) ограничивает сумму множителей Лагранжа вспомогательной задачи в области W_α.

Вся порожденная алгоритмом последовательность $\{x_k\}$ может быть разбита на отрезки, начальная точка которых есть x_0, а из конечной происходит возврат в x_0. Для определенности рассмотрим первый из этих отрезков $\{x_k\}_{k=0}^{l}$, так что $x_{l+1} = x_0$. Внутри этого отрезка $C_k = C_0$.

Покажем, что для всех $k = 0, 1, \ldots, l$ выполняется соотношение

$$f_0(x_k) + N_k F(x_k) \leqslant C_0. \tag{4.59}$$

Действительно, если при данном k это соотношение выполнено, то для $k+1$ в силу правил 2–4 алгоритма, независимо от соотношения между N_{k+1} и N_k, имеем

$$f(x_{k+1}) + N_{k+1} F(x_{k+1}) < f_0(x_k) + N_{k+1} F(x_k) \leqslant C_k = C_0.$$

Из (4.59) следует, что

$$F(x_k) \leqslant \frac{C_0 - f_0(x_k)}{N_k} \leqslant \frac{C_0 - \mu}{N_0} = \alpha,$$

так как $N_k \geqslant N_0$, $f(x_k) \geqslant \mu$.

Таким образом, $x_k \in W_\alpha$ для всех k из выбранного отрезка, а значит, при всех k, так как каждый из выбранных отрезков последовательности $\{x_k\}$ начинается с x_0.

Но из того, что $x_k \in W_\alpha$, и предположений б) и г) вытекает, что после конечного числа увеличений N_k эта величина превзойдет N_α и перестанет меняться. С этого момента алгоритм начнет работать как метод линеаризации с фиксированной константой N, и поэтому его сходимость вытекает из теоремы 4.1.

Заметим, что возврат в точку x_0 последовательности $\{x_k\}$ должен происходить достаточно редко, так как уже первая оценка величины N, получаемая в точке x_0, дает хорошее приближение. Кроме того, если $F(x_k) = 0$ (а практически, если $F(x_k)$ достаточно мало), то согласно шагу 4 алгоритма также не должно происходить возврата в x_0. Видимо, этим объясняется тот факт что при практических расчетах достаточно было просто всегда увеличивать N на шаге 3, не возвращаясь в начальную точку.

Из условий теоремы 4.4 единственным сложно проверяемым условием является б). Но если правильно оценить, что на самом деле дает утверждение теоремы, то это условие вовсе не покажется жестким. Ведь из утверждения теоремы следует разрешимость систем нелинейных неравенств. Этот факт имеет смысл зафиксировать.

Т е о р е м а 4.5. *В условиях теоремы* 4.4 *система неравенств*

$$f_i(x) \leqslant 0, \quad i \in I, x \in M,$$

имеет решение.

Приведем теперь два достаточно важных случая, когда выполнены условия теоремы 4.4. Для того чтобы не повторяться, будем ниже предполагать, что условие а) теоремы 4.4 выполнено.

Т е о р е м а 4.6. *Пусть в любой компактной области градиенты* $f_i'(x)$ *функций* $f_i(x)$, $i = 1, \ldots, m$, *линейно независимы,* $f_0(x) \geqslant \mu > -\infty$ *для всех* $x \in \mathbf{R}^n$ *и множество*

$$W_\alpha = \left\{x: F(x) = \max_{i=1,\ldots,m} |f_i(x)| \leqslant \alpha\right\}$$

ограничено. Тогда существует решение задачи

$$\min_x \{f_0(x): f_i(x) = 0, \ i = 1, \ldots, m\},$$

а модифицированный алгоритм метода линеаризации строит последовательность, любая предельная точка которой удовлетворяет ограничениям этой задачи и необходимым условиям экстремума.

Д о к а з а т е л ь с т в о. Как было показано в п. 7, для рассматриваемой задачи решение вспомогательной задачи и множители Лагранжа даются

формулами (4.49), (4.50). В силу предположений теоремы 4.6. матрица $[f'(x)\,(f'(x))^*]^{-1}$ существует и непрерывна, а поэтому ограничена в компактной области. Формулы (4.49) показывают, что вектор $u(x)$ также ограничен в любой компактной области. Таким образом, выполнены все предположения теоремы 4.4, откуда и следует результат.

Обратимся теперь к выпуклому случаю.

Теорема 4.7. *Пусть функции* $f_i(x), i \in I$, *и множество* M *выпуклы и существует точка* $\bar{x} \in M$ *такая, что*

$$f_i(\bar{x}) \leqslant -\gamma < 0\,, i \in I\,.$$

Тогда для множителей Лагранжа вспомогательной задачи (4.57) *справедлива оценка*

$$\sum_{i \in I} u^i(x) \leqslant \frac{1}{2\gamma}\,[\,\|f_0'(x)\|^2 + \|\bar{x} - x\|\,]^2\,.$$

Доказательство. Пусть $p(x)$ и $u^i(x)$ – соответственно решение и множители Лагранжа задачи (4.57). Тогда

$$(f_0'(x), p(x)) + \tfrac{1}{2}\,\|p(x)\|^2 \leqslant L(p, u(x)) =$$

$$= (f_0'(x), p) + \tfrac{1}{2}\,\|p\|^2 + \sum_{i \in I} u^i(x)\,[\,(f_i'(x), p) + f_i(x)\,]\,, x + p \in M\,.$$

Кроме того, в силу выпуклости функций f_i

$$(f_i'(x), p) + f_i(x) \leqslant f_i(x + p), i \in I\,,$$

$$(f_0'(x), p(x)) + \tfrac{1}{2}\,\|p(x)\|^2 \geqslant \min_p\,[(f_0'(x), p) + \tfrac{1}{2}\,\|p\|^2] = -\tfrac{1}{2}\,\|f_0'(x)\|^2.$$

Отсюда и из предыдущего неравенства для $\bar{p} = \bar{x} - x$ получаем

$$-\tfrac{1}{2}\,\|f_0'(x)\|^2 \leqslant (f_0'(x), p) + \tfrac{1}{2}\,\|\bar{p}\|^2 + \sum_{i \in I} u^i(x) f_i(\bar{x}) \leqslant$$

$$\leqslant \|f_0'(x)\|\,\|\bar{p}\| + \tfrac{1}{2}\,\|\bar{p}\|^2 - \gamma \sum_{i \in I} u^i(x)\,,$$

или, после простых преобразований,

$$\sum_{i \in I} u^i(x) \leqslant \frac{1}{2\gamma}\,[\,\|f_0'(x)\| + \|\bar{p}\|\,]^2\,,$$

что и требовалось доказать.

Сопоставляя полученный результат с теоремой 4.4, приходим к следующему выводу.

Теорема 4.8. *Пусть выполнены следующие условия:*

а) *функция* $f_i(x), i \in I$, *и множество* M *выпуклы, и существует такая точка* $\bar{x} \in M$, *что*

$$f_i(\bar{x}) \leqslant -\gamma < 0, i \in I\,;$$

б) *множество*

$$W_\alpha = \{x: f_i(x) \leqslant \alpha, i \in I, x \in M\}$$

ограничено;

в) $\inf\limits_x \{f_0(x): x \in M\} = \mu > -\infty$.

Тогда модифицированный алгоритм сходится в смысле, указанном в теореме 4.4.

Заметим, что мы не предполагали выпуклости функции $f_0(x)$. Если же предположить и это, то любая предельная точка последовательности, порожденной алгоритмом, будет решением задачи выпуклого программирования

$$\min\limits_x \{f_0(x): f_i(x) \leqslant 0,\ i \in I, x \in M\}.$$

§ 5. РЕШЕНИЕ СИСТЕМ РАВЕНСТВ И НЕРАВЕНСТВ

Возможность решать системы нелинейных равенств и неравенств, очевидно, является одним из необходимых элементов при решении задач математического программирования. Алгоритмы предыдущего параграфа естественным образом приспособлены для этого. Действительно, достаточно положить минимизируемую функцию равной тождественно нулю и любая точка, удовлетворяющая ограничениям, будет и решением задачи математического программирования.

Однако задача нахождения решения системы равенств и неравенств обладает рядом свойств, которые позволяют обеспечить гораздо более высокую скорость сходимости, что, как будет видно из дальнейшего, не удается сделать в общей задаче оптимизации без дополнительных усилий.

Итак, пусть рассматривается система

$$f_i(x) \leqslant 0,\ i \in I^-,\ f_i(x) = 0,\ i \in I^0, \tag{5.1}$$

где I^-, I^0 – конечные множества индексов. Как и в предыдущем параграфе, без ограничения общности, в дальнейшем будет рассматриваться только система неравенств

$$f_i(x) \leqslant 0,\ i \in I = \{1, \ldots, m\}. \tag{5.2}$$

Решение системы (5.1) сводится к решению системы (5.2) способами, которые указаны в § 4.

Положим

$$F(x) = \max\{0, f_1(x), \ldots, f_m(x)\},$$
$$I_\delta(x) = \{i \in I: f_i(x) \geqslant F(x) - \delta\},\ \delta \geqslant 0. \tag{5.3}$$

На протяжении всего этого параграфа будет предполагаться, что градиенты $f_i'(x)$ удовлетворяют условию Липшица в любой компактной области.

1. Вспомогательная задача. Свяжем с каждой точкой x вспомогательную задачу квадратичного программирования

$$\min_{p} \{ \tfrac{1}{2} \| p \|^2 : (f_i'(x), p) + f_i(x) \leqslant 0, \ i \in I_\delta(x) \}. \tag{5.4}$$

Решение этой задачи обозначим через $p(x)$, а через $u_i(x) \geqslant 0$, $i \in I_\delta(x)$, – соответствующие множители Лагранжа. Согласно изложенной в § 2 теории необходимых условий экстремума должны выполняться следующие соотношения:

$$\begin{gathered}(p(x))^* + \sum_{i \in I_\delta(x)} u^i(x) f_i'(x) = 0, \\ u^i(x) [(f_i'(x), p(x)) + f_i(x)] = 0, \ i \in I_\delta(x).\end{gathered} \tag{5.5}$$

На этих соотношений путем умножения первого из них скалярно на $p(x)$ и с учетом второго легко получаем

$$\| p(x) \|^2 = \sum_{i \in I_\delta(x)} u^i(x) f_i(x). \tag{5.6}$$

2. Алгоритм. Сформулируем теперь вычислительную процедуру, которая при некоторых предположениях обеспечит получение решения системы (5.2).

Пусть начальная точка x_0 и ϵ, $0 < \epsilon < 1$, выбраны. Если точка x_k уже построена, то решаем задачу (5.4) и получаем $p_k = p(x_k)$. Путем деления пополам $\alpha = 1$ до выполнения неравенства

$$F(x_k + \alpha_k p_k) \leqslant (1 - \epsilon \alpha_k) F(x_k) \tag{5.7}$$

вычисляем α_k. Полагаем

$$x_{k+1} = x_k + \alpha_k p_k.$$

Итерация окончена.

Ниже будут приведены условия, при которых этот алгоритм сходится.

3. Сходимость алгоритма. Приведем некоторые оценки. Пусть $i \in I_\delta(x)$ и задача (5.4) разрешима. Тогда для $p = p(x)$ получаем с помощью формулы о среднем значении:

$$f_i(x + \alpha p) = f_i(x) + \alpha (f_i'(x + \theta \alpha p), p) = f_i(x) + \alpha (f_i'(x), p) +$$

$$+ \alpha (f_i'(x + \theta \alpha p) - f_i'(x), p) \leqslant f_i(x) - \alpha f_i(x) + \alpha^2 L \| p \|^2,$$

где $0 \leqslant \theta \leqslant 1$, L – константа Липшица для градиентов $f_i'(x)$. Кроме того, был использован тот факт, что вектор p удовлетворяет ограничениям задачи (5.4).

Полученную формулу можно переписать в виде

$$f_i(x + \alpha p) \leqslant (1 - \alpha) F(x) + \alpha^2 L \| p \|^2, \ i \in I_\delta(x), \tag{5.8}$$

для $0 \leqslant \alpha \leqslant 1$.

Если $i \notin I_\delta(x)$, то

$$f_i(x+\alpha p) = f_i(x) + \alpha (f_i'(x+\theta p), p) \leqslant$$

$$\leqslant f_i(x) + \alpha K \| p \| \leqslant F(x) - \delta + \alpha K \| p \|, \tag{5.9}$$

где K – константа, ограничивающая норму градиента в достаточно широкой области. Далее,

$$(1 - \alpha) F(x) \geqslant F(x) - \delta + \alpha K \| p \|,$$

если

$$\alpha \leqslant \alpha_1 = \frac{\delta}{F(x) + K \| p \|} . \tag{5.10}$$

Поэтому при $\alpha \leqslant \min |1, \alpha_1|$ из (5.8), (5.9) следует неравенство

$$f_i(x+\alpha p) \leqslant (1 - \alpha) F(x) + \alpha^2 L \| p \|^2,$$

справедливое при всех $i \in I$, так что

$$F(x+\alpha p) \leqslant (1 - \alpha) F(x) + \alpha^2 L \| p \|^2, \tag{5.11}$$

$$\alpha \leqslant \min| 1, \alpha_1 |.$$

Основываясь на оценке (5.11), приведем несколько критериев сходимости.

Т е о р е м а 5.1. *Пусть множество* $\Omega_0 = \{x: F(x) \leqslant F(x_0)\}$ *компактно и вспомогательная задача* (5.4) *имеет в* Ω_0 *равномерно ограниченное решение. Тогда алгоритм п. 2 порождает последовательность* x_k, *для которой* $F(x_k) \to 0$.

Д о к а з а т е л ь с т в о. По предположению для некоторой константы C

$$\| p(x) \| \leqslant C, \quad x \in \Omega_0.$$

Оценка (5.11) применительно к точке $x = x_k$ может теперь быть переписана в виде

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha) F(x_k) + \alpha^2 L C^2 = F(x_k) -$$

$$- \alpha F(x_k) \left[1 - \alpha \frac{L C^2}{F(x_k)} \right].$$

Если

$$1 - \alpha \frac{L C^2}{F(x_k)} \geqslant \epsilon,$$

т.е. если

$$\alpha \leqslant \frac{1 - \epsilon}{L C^2} F(x_k),$$

то

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha \epsilon) F(x_k). \tag{5.12}$$

Заметим, что неравенство (5.12) справедливо при

$$\alpha \leqslant \min\left[1, \frac{\delta}{F(x_k)+K\|p_k\|}, \frac{1-\epsilon}{LC^2}F(x_k)\right].$$

Вспомним теперь, что α_k выбирается делением пополам единицы до первого выполнения неравенства (5.7) Поэтому

$$\alpha_k \geqslant \frac{1}{2}\min\left[1, \frac{\delta}{F(x_k)+K\|p_k\|}, \frac{1-\epsilon}{LC^2}F(x_k)\right]. \qquad (5.13)$$

В силу (5.12) $F(x_k)$ монотонно убывает, и поэтому, если $F(x_k)$ не стремится к нулю, то $F(x_k) \geqslant v > 0$ для всех k.

Используя (5.13) и очевидные огрубления этого неравенства, получаем

$$\alpha_k \geqslant \frac{1}{2}\min\left[1, \frac{\delta}{F(x_0)+KC}, \frac{1-\epsilon}{LC^2}v\right] = \bar{\alpha} > 0.$$

Поэтому

$$F(x_k + \alpha_k p_k) = F(x_{k+1}) \leqslant (1 - \bar{\alpha}\epsilon)F(x_k),$$

откуда следует, что $F(x_k) \to 0$, что завершает доказательство теоремы.

З а м е ч а н и е. Так как $F(x_k) \to 0$, то из (5.13) следует неравенство

$$\alpha_k \geqslant \frac{1}{2}\frac{1-\epsilon}{LC^2}F(x_k).$$

Подставляя правую часть этого неравенства в (5.12) вместо α, получаем более грубое неравенство

$$F(x_{k+1}) \leqslant (1 - \alpha_k \epsilon)F(x_k) \leqslant \left(1 - \frac{1-\epsilon}{2LC^2}F(x_k)\right)F(x_k).$$

В [28, с. 205] показано, что из приведенного неравенства следует существование такой константы $C_1 > 0$, что

$$F(x_k) \leqslant C_1 / k.$$

Таким образом, сходимость может быть достаточно медленной.

Естественно, такая медленная оценка скорости сходимости обусловлена не тем, что сходимость действительно плохая, а большой грубостью исходного предположения о том, что $p(x)$ просто ограничено. Если сделать более точные предположения, гарантирующие стремление $p(x)$ к нулю вместе с $F(x)$, то и оценка получается более точной.

Т е о р е м а 5.2. *Если множество* $\Omega_0 = \{x: F(x) \leqslant F(x)\}$ *компактно, вспомогательная задача* (5.4) *в этом множестве разрешима и ее множители Лагранжа равномерно ограничены, то* $F(x_k) \to 0$ *и имеет место неравенство*

$$F(x_{k+1}) \leqslant q F(x_k), \quad 0 < q < 1.$$

Д о к а з а т е л ь с т в о. Пусть для всех $x \in \Omega_0$

$$\sum_{i \in I_\delta(x)} u^i(x) \leqslant N.$$

Тогда из (5.6) вытекает, что

$$\| p(x) \|^2 \leqslant N F(x), \tag{5.14}$$

поэтому неравенство (5.11) теперь дает оценку

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha) F(x_k) + \alpha^2 L N F(x_k) =$$

$$= [1 - \alpha (1 - \alpha L N)] F(x_k).$$

Следовательно, при

$$\alpha \leqslant (1 - \epsilon) / (L N) \tag{5.15}$$

имеет место неравенство

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha \epsilon) F(x_k). \tag{5.16}$$

Итак, если

$$\alpha \leqslant \min \left[1, \frac{\delta}{F(x_k) + K \| p_k \|}, \frac{1 - \epsilon}{L N} \right], \tag{5.17}$$

то справедливо неравенство (5.16).

Учитывая способ выбора α_k путем деления пополам, получаем

$$F(x_k + \alpha_k p_k) \leqslant (1 - \alpha_k \epsilon) F(x_k),$$

$$\alpha_k \geqslant \frac{1}{2} \min \left[1, \frac{\delta}{F(x_k) + K \| p_k \|}, \frac{1 - \epsilon}{L N} \right].$$

Теперь очевидно, что α_k остается больше некоторой величины $\bar{\alpha}$, так что в итоге имеем

$$F(x_{k+1}) \leqslant q F(x_k), \quad q = 1 - \bar{\alpha} \epsilon.$$

Заметим, что теперь из $F(x_k) \to 0$ вытекает, что $\| p_k \| \to 0$ (см. (514)) и, значит, при больших k

$$\frac{\delta}{F(x_k) + K \| p_k \|} \geqslant 1.$$

Поэтому при больших k

$$\alpha_k \geqslant \tfrac{1}{2} \min [1, (1 - \epsilon) / (L N)]$$

и можно положить

$$\bar{\alpha} = \tfrac{1}{2} \min [1, (1 - \epsilon) / (L N)].$$

Следствие. *Если функции* $f_i(x)$, $i \in I$, *выпуклы, множество* Ω_0 *компактно и существует такая точка* $\bar{x}$, *что*

$$f_i(\bar{x}) \leqslant \gamma < 0, \quad i \in I,$$

то

$$F(x_{k+1}) \leqslant q\, F(x_k), \quad 0 \leqslant q < 1.$$

Действительно, в силу теоремы 4.7 при $f_0(x) = 0$

$$\sum_{i \in I} u^i(x) \leqslant \frac{1}{2\gamma} \| x - \bar{x} \|^2.$$

Таким образом, выполнены все предположения теоремы 5.2.

Покажем теперь, что при достаточно естественных предположениях можно сделать более сильное утверждение.

Теорема 5.3. *Пусть выполнены условия теоремы* 5.2 *и, кроме того, для всех* x, *удовлетворяющих неравенству* $F(x) \leqslant w$ *для некоторого* $w > 0$, *существует такая константа* C, *что*

$$\| p(x) \| \leqslant C F(x). \tag{5.18}$$

Тогда последовательность x_k *сходится к некоторой точке* x_*, *являющейся решением системы* (5.2), *и для достаточно больших* k

$$F(x_{k+1}) \leqslant L C^2 F^2(x_k). \tag{5.19}$$

Доказательство. Согласно предыдущей теореме $F(x_k) \to 0$. Поэтому при больших k имеем $F(x_k) \leqslant w$. Кроме того, можно также считать, что

$$\frac{\delta}{F(x_k) + K \| p_k \|} \geqslant 1,$$

поэтому неравенство (5.11) совместно с (5.18) показывает, что

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha) F(x_k) + \alpha^2 L \| p_k \|^2 \leqslant (1 - \alpha) F(x_k) +$$

$$+ \alpha^2 L C^2 F^2(x_k) \tag{5.20}$$

при $0 \leqslant \alpha \leqslant 1$ и достаточно больших k. Отсюда

$$F(x_k + \alpha p_k) \leqslant [1 - \alpha (1 - \alpha L C^2 F(x_k))]\, F(x_k).$$

Если

$$1 - \alpha L C^2 F(x_k) \geqslant \epsilon,$$

т.е. если

$$\alpha \leqslant \frac{1 - \epsilon}{LC^2 F(x_k)},$$

то

$$F(x_k + \alpha p_k) \leqslant (1 - \alpha\epsilon) F(x_k) \tag{5.21}$$

при

$$\alpha \leqslant \min \left[1, \frac{1 - \epsilon}{LC^2 F(x_k)} \right].$$

Но для достаточно больших k

$$\frac{1 - \epsilon}{LC^2 F(x_k)} \geqslant 1,$$

и поэтому в силу правила выбора α_k эта величина станет равной единице.

Итак, α_k выберется равным 1 при достаточно больших k. Соотношения (5.20) и (5.21) теперь приобретают вид

$$F(x_{k+1}) \leqslant LC^2 F^2(x_k). \tag{5.22}$$

$$F(x_{k+1}) \leqslant (1 - \epsilon) F(x_k). \tag{5.23}$$

Эти неравенства выполняются при $k \geqslant k_1$, где k_1 достаточно велико. Из (5.23) теперь следует, что

$$F(x_k) \leqslant (1 - \epsilon)^{k - k_1} F(x_{k_1}).$$

Сопоставляя это с (5.18), получаем

$$\| p_k \| \leqslant C (1 - \epsilon)^{k - k_1} F(x_{k_1}).$$

Следовательно, ряд $x_{k_1} + p_{k_1} + \ldots + p_{k-1} = x_k$ при $k \to \infty$ сходится, так как мажорируется сходящейся геометрической прогрессией.

Итак,

$x_k \to x_*$,

$$F(x_*) = \lim_{k \to \infty} F(x_k) = 0,$$

что полностью завершает доказательство теоремы.

Выясним теперь некоторые условия, при которых справедливо неравенство (5.18). Эти условия покажут, что выполнение неравенства (5.18) достаточно естественно и оно просто характеризует определенную невырожденность системы неравенств (5.2).

Теорема 5.4. *Пусть множество* $\Omega_0 = \{ x : F(x) \leqslant F(x_0) \}$ *компактно, вспомогательная задача для* $x \in \Omega_0$ *разрешима и ее множители Лагранжа равномерно ограничены. Пусть, кроме того, существуют такие* $\omega > 0$, $\delta_0 > 0$, *что для* x, *удовлетворяющих неравенству* $0 < F(x) \leqslant \omega$, *величина*

$$l(x, \delta_0) = \min_{\lambda} \left\{ \| \sum_{i \in I_{\delta_0}(x)} \lambda_i f_i'(x) \| : \lambda_i \geqslant 0, \sum_{i \in I_{\delta_0}(x)} \lambda_i = 1 \right\}$$

равномерно ограничена снизу числом $\gamma > 0$. *Тогда алгоритм решения системы неравенств сходится, т.е.* $F(x_k) \to 0$, $x_k \to x_*$, $F(x_*) = 0$, *и*

$$F(x_{k+1}) \leqslant C_1 F^2(x_k)$$

при достаточно больших k.

Доказательство. На основании теорем 5.2 и 5.3 ясно, что достаточно доказать, что предположения теоремы 5.4 обеспечивают выполнение неравенства (5.18) для точек x_k с достаточно большим номером.

Так как все предположения теоремы 5.2 выполняются, то $F(x_k)$ и $p(x_k)$ стремятся к нулю. Рассмотрим индексы k, настолько большие, что

$$F(x_k) \leqslant \delta_0 / 2, \quad K \| p(x_k) \| \leqslant \delta_0/2.$$

Пусть $i \in I_\delta(x_k)$ и $u^i(x_k) > 0$, т.е. ограничение, соответствующее индексу i во вспомогательной задаче, активно:

$$(f_i'(x_k), p(x_k)) + f_i(x_k) = 0.$$

Отсюда

$$|f_i(x_k)| = |(f_i'(x_k), p(x_k))| \leqslant K \| p(x_k) \| \leqslant \delta_0 / 2,$$

$$F(x_k) - f_i(x_k) \leqslant \delta_0 / 2 - f_i(x_k) \leqslant \delta_0,$$

т.е. $i \in I_{\delta_0}(x_k)$.

Итак, при больших k все активные индексы во вспомогательной задаче принадлежат $I_{\delta_0}(x_k)$. Воспользовавшись (5.5), теперь получаем

$$\| p(x_k) \| = \Big\| \sum_{i \in I_{\delta_0}(x_k)} u^i(x_k) f_i'(x_k) \Big\| =$$

$$= \Big(\sum_{i \in I_{\delta_0}(x_k)} u^i(x_k) \Big) \Big\| \sum_{i \in I_{\delta_0}(x_k)} \lambda_i f_i'(x_k) \Big\| \geqslant$$

$$\geqslant \Big(\sum_{i \in I_{\delta_0}(x_k)} u^i(x_k) \Big) l(x_k, \delta_0),$$

где введено обозначение

$$\lambda_i = \frac{u^i(x_k)}{\sum\limits_{i \in I_{\delta_0}(x_k)} u^i(x_k)}.$$

Окончательно можно записать

$$\sum_{i \in I_{\delta_0}(x_k)} u^i(x_k) \leqslant \frac{\| p(x_k) \|}{l(x_k, \delta_0)} \leqslant \frac{1}{\gamma} \| p(x_k) \|.$$

Сопоставляя это с (5.6), получаем неравенство

$$\| p(x_k) \|^2 \leqslant \Big(\sum_{i \in I_{\delta_0}(x_k)} u^i(x_k) \Big) F(x_k) \leqslant \frac{F(x_k)}{\gamma} \| p(x_k) \|,$$

или

$$\| p(x_k) \| \leqslant (1/\gamma) F(x_k).$$

Таким образом, выполнено неравенство (5.18), и поэтому справедливы все выводы теоремы 5.3. Доказательство завершено.

Теорема 5.4 позволяет полностью рассмотреть случай выпуклых неравенств.

Теорема 5.5. *Пусть*

а) $f_i(x)$, $i \in I$, – *выпуклые непрерывно дифференцируемые функции, производные которых удовлетворяют условию Липшица;*

б) *множество* $\Omega_0 = \{x: F(x) \leqslant F(x_0)\}$ *компактно, и существует такая точка* $\bar{x}$, *что*

$$f_i(\bar{x}) \leqslant -\gamma < 0, \; i \in I.$$

Тогда алгоритм строит последовательность x_k, *сходящуюся к решению системы неравенств, и скорость сходимости квадратичная:*

$$F(x_{k+1}) \leqslant C_1 F^2(x_k).$$

Доказательство. Пусть $\delta_0 = \gamma/2$. Тогда для $i \in I_{\delta_0}(x)$

$$f_i(x) \geqslant F(x) - \delta_0 \geqslant -\gamma/2,$$

т.е.

$$f_i(x) + \gamma/2 \geqslant 0, \; i \in I_{\delta_0}(x). \tag{5.24}$$

Пусть $F(x) > 0$, $p = \bar{x} - x$. Так как $f_i(\bar{x}) \leqslant -\gamma, i \in I$, то норма вектора $\bar{p}$ всегда больше некоторой константы, ибо для x, близких к $\bar{x}$, $F(x) = 0$.

Так как функции $f_i(x)$ выпуклы, то

$$(f_i'(x), \bar{x} - x) + f_i(x) \leqslant f_i(\bar{x}) \leqslant -\gamma. \tag{5.25}$$

Поэтому

$$(f_i'(x), \bar{p}) + f_i(x) + \gamma \leqslant 0,$$

что с учетом неравенства (5.24) приводит к соотношениям

$$(f_i'(x), \bar{p}) + \gamma/2 \leqslant 0, i \in I_{\delta_0}(x). \tag{5.26}$$

Выберем теперь произвольные $\lambda_i \geqslant 0$ такие, что

$$\sum_{i \in I_{\delta_0}(x)} \lambda_i = 1.$$

Тогда из (5.26) получаем

$$\left(\sum_{i \in I_{\delta_0}(x)} \lambda_i f_i'(x), \bar{p}\right) + \gamma/2 \leqslant 0, \quad -\| \sum_{i \in I_{\delta_0}(x)} \lambda_i f_i'(x) \| \; \| \bar{p} \| + \gamma/2 \leqslant 0$$

и в силу производительности λ_i

$$-l(x, \delta_0) \| \bar{p} \| + \gamma/2 \leqslant 0.$$

Поэтому

$$l(x, \delta_0) \geqslant \gamma / (2 \| \bar{p} \|).$$

Но это как раз то неравенство, которое требуется в теореме 5.4. Таким образом, теорема доказана.

§ 6. УСКОРЕНИЕ СХОДИМОСТИ МЕТОДА ЛИНЕАРИЗАЦИИ

Метод линеаризации может быть применен и в случае, когда ограничения вообще отсутствуют, т.е. когда требуется минимизировать функцию $f_0(x)$ во всем пространстве $\mathbf{R}^n$. В этом случае вспомогательная задача (4.4) приобретает вид

$$\min_p \left\{ (f_0'(x),\ p) + \tfrac{1}{2} \| p \|^2 :\ p \in \mathbf{R}^n \right\}.$$

Ее очевидным решением является $p = -(f_0'(x))^*$. (Напомним, что f_0' — вектор-строка, а звездочка обозначает транспонирование.)

Из полученного выражения для направления сдвига p из точки x видно, что в рассматриваемом случае метод просто совпадает с одним из градиентных методов [28]. Поэтому и скорость сходимости метода линеаризации в целом не может быть выше, чем у градиентного метода.

Ниже будет проведен анализ скорости сходимости метода линеаризации. На основе этого анализа будет построен алгоритм, обладающий высокой скоростью. Интересно также отметить, что наличие ограничений может приводить к ускорению сходимости, даже к квадратичной сходимости в некоторых случаях, и без специальных модификаций алгоритма.

Описанный в § 4 метод линеаризации обладает глобальной сходимостью, т.е. сходится с достаточно плохого начального приближения. В то же время скорость сходимости оценивается в окрестности решения. При этом алгоритмы, обеспечивающие высокую скорость сходимости, работают в окрестности решения, истинные размеры которой заранее неизвестны. Поэтому возникает довольно сложная задача совмещения глобальной сходимости и ее высокой скорости путем организации такого способа вычислений, который бы автоматически переводил алгоритм, обеспечивающий глобальную сходимость, в локальный алгоритм, быстро уточняющий решение. Ниже эта задача будет решена определенным способом. Однако этот способ не во всем является удовлетворительным с вычислительной точки зрения, так что указанная задача перехода нуждается в дополнительной проработке.

1. Основные предположения. Мы будем рассматривать задачу минимизации

$$\min_x \left\{ f_0(x):\ f_i(x) \leqslant 0,\ \ i \in I \right\},\ \ I = \left\{ 1, 2, \ldots, m \right\}. \tag{6.1}$$

Как и в § 4, мы ограничиваемся только ограничениями типа неравенств, так как ограничения типа равенств могут быть рассмотрены с помощью очевидных изменений в алгоритме так же, как это делалось в § 4. Напомним, что

$$F(x) = \max \left\{ 0,\ f_1(x),\ \ldots,\ f_m(x) \right\},$$

$$I_\delta(x) = \left\{ i \in I:\ f_i(x) \geqslant F(x) - \delta \right\},\ \ \delta > 0, \tag{6.2}$$

$$\Phi_N(x) = f_0(x) + NF(x),\ \ N \geqslant 0.$$

Пусть x_* — решение задачи (6.1). Согласно теоремам §2 в этой точке выполняются необходимые условия экстремума

$$f_0'(x_*) + \sum_{i=1}^{m} u_*^i f_i'(x_*) = 0,$$

$$u_*^i \geqslant 0,\ \ u_*^i f_i(x_*) = 0,\ \ i = 1, \ldots, m. \tag{6.3}$$

Обозначим

$$I_* = \{ i \in I:\ f_i(x_*) = 0 \}.$$

Множество I_* есть множество индексов активных ограничений в решении. Всюду в дальнейшем мы будем предполагать, что выполнены следующие допущения:

1) все функции $f_i(x)$, $i = 0, 1, \ldots, m$, трижды непрерывно дифференцируемы;

2) градиенты $f_i'(x)$, $i \in I_*$, линейно независимы, и $u_*^i > 0$, $i \in I_*$;

3) условие

$$(L''_{xx}(x_*, u_*)\, p,\ p) > 0 \tag{6.4}$$

выполнено для всех $p \neq 0$, удовлетворяющих равенствам

$$(f_i'(x_*),\ p) = 0,\quad i \in I_*. \tag{6.5}$$

В силу сказанного в § 2 эти допущения не являются слишком обременительными и требуют лишь определенной регулярности решения.

2. Локальный анализ вспомогательной задачи. Согласно § 4 вспомогательная задача в точке x имеет вид

$$\min_p \{ (f_0'(x),\ p) + \tfrac{1}{2} \| p \|^2 :\ (f_i'(x),\ p) + f_i(x) \leqslant 0,\ i \in I_\delta(x) \}. \tag{6.6}$$

Ее решение и множители Лагранжа обозначаются соответственно через $p(x)$ и $u^i(x)$, $i \in I_\delta(x)$. Обозначим также через $I_\delta^0(x)$ множество активных ограничений задачи (6.6), т.е.

$$I_\delta^0(x) = \{ i \in I_\delta(x):\ (f_i'(x),\ p(x)) + f_i(x) = 0 \}. \tag{6.7}$$

Займемся теперь исследованием поведения решения задачи (6.6) в окрестности точки x_*.

Лемма 6.1. *Существует окрестность точки x_*, в которой $I_* \subseteq I_\delta(x)$.*

Доказательство. Легко видеть, что в качестве такой окрестности можно взять множество точек x, удовлетворяющих неравенствам

$$F(x) < \delta/2,\quad |f_i(x)| < \delta/2,\quad i \in I_*.$$

Лемма 6.2. *Существует окрестность точки x_*, в которой $p(x)$, $u^i(x)$, $i \in I_*$, дифференцируемы, их производные удовлетворяют условию Липшица, и в этой окрестности $I_\delta^0(x) = I_*$.*

Доказательство. В силу необходимых условий экстремума для задачи (6.6) выполняются соотношения

$$p(x) + (f_0'(x))^* + \sum_{i \in I_\delta(x)} u^i(x)\,(f_i'(x))^* = 0, \tag{6.8}$$

$$u^i(x) \geqslant 0,\quad u^i(x)\,|(f_i'(x), p(x)) + f_i(x)| = 0,\quad i \in I_\delta(x). \tag{6.9}$$

При $x = x_*$ $p(x_*) = 0$, $u^i(x_*) = u_*^i$. Кроме того, ясно, что $I_\delta^0(x_*) = I_*$.

Рассмотрим систему уравнений

$$\begin{aligned} &\tilde{p} + (f_0'(x))^* + \sum_{i \in I_*} \tilde{u}^i (f_i'(x))^* = 0, \\ &f_i'(x)\,\tilde{p} + f_i(x) = 0,\quad i \in I_*. \end{aligned} \tag{6.10}$$

относительно неизвестных $\tilde{p}, \tilde{u}^i$, $i \in I_*$. Если обозначить через $f'(x)$ матрицу

со строками $f_i'(x)$, $i \in I_*$, через $f'(x)$ – вектор-столбец с компонентами $f_i(x)$, $i \in I_*$, а через $\widetilde{u}$ – вектор-столбец с компонентами $\widetilde{u}^i$, $i \in I_*$, то систему (6.10) можно переписать в виде

$$\begin{aligned} &\widetilde{p} + (f'(x))^* \widetilde{u} = -(f_0'(x))^*, \\ &f'(x)\, p = -f(x). \end{aligned} \tag{6.11}$$

Нетрудно решение этой системы записать в явном виде, если выразить p через u из первого уравнения и подставить во второе. После простых выкладок получаются следующие выражения:

$$\widetilde{u}(x) = (f'(x) f'^*(x))^{-1} \, | f(x) - f'(x)\, f_0'^*(x) |, \tag{6.12}$$

$$\widetilde{p}(x) = -(I \quad \Pi(x)) f_0'^*(x) - f'^*(x)\, v(x), \tag{6.13}$$

где

$$\Pi(x) = f'^*(x) \, [f'(x) f'^*(x)]^{-1} \, f'(x), \tag{6.14}$$

$$v(x) = (f'(x) f'^*(x))^{-1} \, f(x). \tag{6.15}$$

Обратная матрица, фигурирующая в этих выражениях, корректно определена, так как по допущению векторы $f_i'(x_*)$, $i \in I_*$, линейно независимы, а значит, они линейно независимы и в некоторой окрестности точки x_*.

Покажем, что в некоторой окрестности точки x_* функции $\widetilde{p}(x)$ и $\widetilde{u}(x)$ являются одновременно и решением вспомогательной задачи (6.6). Из формул (6.12) – (6.15) и дифференциальных свойств функций f_i вытекает, что $\widetilde{p}(x)$ и $\widetilde{u}(x)$ дважды непрерывно дифференцируемы. Кроме того, эти формулы показывают, что система (6.10) однозначно разрешима. Так как при $x = x_*$ система (6.10) имеет решение $\widetilde{p} = 0$, $\widetilde{u} = u_*$, то

$$\widetilde{p}(x_*) = 0, \quad \widetilde{u}(x_*) = u_*.$$

Но $u_* > 0$ по допущению, и поэтому в некоторой окрестности x_*

$$\widetilde{u}(x_*) \geqslant 0. \tag{6.16}$$

Далее, так как $f_i(x_*) < 0$, $i \notin I_*$, то

$$(f_i'(x), \widetilde{p}(x)) + f_i(x) < 0, \quad i \notin I_*, \tag{6.17}$$

в некоторой окрестности x_*.

Положим теперь $\widetilde{u}^i = 0$, $i \notin I_*$. Тогда в силу (6.10), (6.16) и (6.17) можно заключить, что в достаточно малой окрестности x_* вектор $\widetilde{p}(x)$ удовлетворяет ограничениям вспомогательной задачи и выполняются соотношения (6.8), (6.9). Но эти последние соотношения являются не только необходимыми, но и достаточными условиями того, чтобы вектор $\widetilde{p}(x)$ был решением вспомогательной задачи (6.6). Тем самым мы действительно показали, что вектор $\widetilde{p}(x)$ является решением вспомогательной задачи (6.6), а $\widetilde{u}(x)$ – соответствующие ненулевые множители Лагранжа. При этом в силу (6.10), (6.17) активными во вспомогательной задаче являются только ограничения $i \in I_*$, т.е. $I_0^0(x) = I_*$, что завершает доказательство леммы.

Из доказанной леммы вытекает, что при исследовании локальных свойств вектора $p(x)$ в окрестности точки x_* можно исходить из системы (6.10) (или (6.11)), рассматривая ее как систему, неявно задающую p и u как функции x. При этом можно пользоваться теоремой о неявных функциях

[23], из которой вытекает, что если матрица частных производных от левых частей (6.10) по искомым переменным p и u^i (матрица Якоби) при $x = x_*$ невырождена, то p и u^i дифференцируемы по x и их производные можно найти, продифференцировав (6.10) и разрешив получающуюся систему относительно искомых производных.

Заметим теперь, что система (6.10) линейна по p и u и поэтому матрица Якоби в точке $x = x_*$ имеет вид

$$\begin{bmatrix} I_n & f'^*(x_*) \\ f'(x) & 0 \end{bmatrix}.$$

Ниже будет показано для более сложного случая, что эта матрица невырождена. Поэтому, как только что было сказано, для вычисления матрицы

$$p'(x_*) = \left\{ \partial p^i / \partial x^j \right\}_{i,\, j\, =\, 1,\ \ldots,\ n}$$

необходимо просто продифференцировать формулы (6.10) в точке $x = x_*$.

Выполнив это дифференцирование с учетом того, что $p(x_*) = 0$, получаем

$$p'(x_*) + f_0''(x_*) + \sum_{i \in I_*} u_*^i f_i''(x_*) + \sum_{i \in I_*} f_i'^*(x_*)\,(u^i(x_*))' = 0,$$

$$f'(x_*)\,p'(x_*) + f'(x_*) = 0,$$

или, в матричном виде,

$$\begin{aligned} &p'(x_*) + L''_{xx}(x_*, u_*) + \sum_{i \in I_*} f_i'^*(x_*)\,(u^i(x_*))' = 0, \\ &f'(x_*)\,p'(x_*) + f'(x_*) = 0. \end{aligned} \tag{6.18}$$

Положим теперь $\Pi_* = \Pi(x_*)$. Из формулы (6.14) и сказанного в п. 5 § 3 вытекает, что Π_* есть оператор проектирования на подпространство, ортогональное подпространству $\{p:\ f'(x_*)\,p = 0\}$, при этом

$$\Pi_*^2 = \Pi_*, \quad \Pi_*^* = \Pi_*, \quad \Pi_* f'^*(x_*) = f'^*(x_*). \tag{6.19}$$

Умножив второе из соотношений (6.18) на $f'^*(x_*)\,[f'(x_*)f'^*(x_*)]^{-1}$, получим

$$\Pi_*\,p'(x_*) + \Pi_* = 0. \tag{6.20}$$

Если теперь первое из соотношений (6.18) умножить на $I_n - \Pi_*$, то в силу (6.19) получим

$$(I_n - \Pi_*)\,p'(x_*) + (I_n - \Pi_*)\,L''_{xx}(x_*, u_*) = 0. \tag{6.21}$$

Складывая (6.20) и (6.21), окончательно получаем

$$p'(x_*) = -\,[\Pi_* + (I_n - \Pi_*)\,L''_{xx}(x_*, u_*)]. \tag{6.22}$$

Л е м м а 6.3. *Собственные числа γ_j матрицы $p'(x_*)$ могут быть охарактеризованы следующим образом: $\gamma_j = -1$, $j = 1, \ldots, |I_*|$, где $|I_*|$ – число индексов в множестве I_* активных ограничений. Остальные $n - |I_*|$ собственных чисел совпадают со взятыми с обратным знаком ненулевыми собственными значениями матрицы $(I - \Pi_*)\,L''_{xx}(x_*, u_*)\,(I - \Pi_*)$.*

Доказательство. Пусть σ – собственное число, y – собственный вектор матрицы $p'(x_*)$. Тогда согласно формуле (6.22)

$$-\Pi_* y - (I_n - \Pi_*) L''_{xx}(x_*, u_*) y = \sigma y = \sigma \Pi_* y + \sigma (I_n - \Pi_*) y,$$

и поэтому

$$\Pi_* y = -\sigma \Pi_* y, \quad (I_n - \Pi_*) L''_{xx}(x_*, u_*) y = -\sigma (I_n - \Pi_*) y. \tag{6.23}$$

Если $\Pi_* y \neq 0$, то $\sigma = -1$, как это следует из (6.23). Если $\Pi_* y = 0$, то $y = (I_n - \Pi_*) y$ и второе из соотношений (6.23) переписывается в виде

$$(I_n - \Pi_*) L''_{xx}(x_*, u_*) (I_n - \Pi_*) y = -\sigma y,$$

т.е. σ – собственное число указанной в формулировке леммы матрицы. Эта матрица симметрична, так как Π_*, L''_{xx} – симметричные матрицы. Более того, рассматриваемая матрица неотрицательно определена. Действительно, для любого w

$$(w, (I_n - \Pi_*) L''_{xx}(I_n - \Pi_*) w) = (z, L''_{xx} z),$$

где

$$z = (I_n - \Pi_*) w, \quad L''_{xx} = L''_{xx}(x_*, u_*).$$

Но $f'(x_*) z = f'(x_*) (I_n - \Pi_*) w = 0$ в силу последнего соотношения (6.19), и поэтому

$$(z, L''_{xx} z) \geqslant 0$$

в силу третьего условия основных допущений, причем равенство нулю возможно, лишь если $z = (I_n - \Pi_*) w = 0$. Из симметрии рассматриваемой матрицы следует, что ее собственные числа и собственные векторы действительны. Так как $y \neq 0$, то из $\Pi_* y = 0$ следует, что $(I_n - \Pi_*) y \neq 0$, и поэтому из соотношения (6.23) следует, что

$$-\sigma(v, y) = (y, (I_n - \Pi_*) L''_{xx}(I_n - \Pi_*) y) = (v, L''_{xx} y) > 0.$$

Таким образом, $\sigma \neq 0$ и, значит, $-\sigma > 0$ и совпадает со строго положительным собственным числом рассматриваемой матрицы.

Осталось установить число собственных значений матрицы $p'(x_*)$, равных -1. По построению матрицы Π_* (см. (6.19))

$$\Pi_* f_i'^*(x_*) = f_i'^*(x_*), \quad i \in I_*,$$

т.е. матрица $(I_n - \Pi_*)$ имеет $|I_*|$ нулевых собственных значений. Поэтому и матрица $(I_n - \Pi_*) L''_{xx}(I_n - \Pi_*)$ имеет $|I_*|$ нулевых собственных значений. С другой стороны, матрица $p'(x_*)$, как мы убедились, имеет n собственных чисел, отличных от нуля. Поэтому ровно $|I_*|$ собственных чисел $p'(x_*)$ в точности равны -1. Лемма доказана.

Полученная характеристика собственных чисел позволяет дать некоторую локальную характеристику скорости сходимости метода линеаризации, если воспользоваться теоремой Островского [12, с. 130].

Пусть $g(x)$ – отображение $\mathbf{R}^n$ в $\mathbf{R}^n$, непрерывно дифференцируемое и обладающее неподвижной точкой x_*, $x_* = g(x_*)$. Если все собственные числа матрицы $g'(x_*)$ по модулю меньше единицы, то существует такая окрестность точки x_*, что итерационный процесс $x_{k+1} = g(x_k)$ сходится к точке x_*,

и при этом для достаточно больших k и любого $\epsilon > 0$ справедлива оценка

$$\| x_k - x_* \| \leqslant C(q+\epsilon)^k,$$

где C зависит только от ϵ, $q = \max_j |\lambda_j|$, λ_j – собственные числа матрицы $q'(x_*)$.

Теорема 6.1. *Пусть выполнены приведенные выше основные предположения 1) – 3) п. 1, $p(x)$ – решение вспомогательной задачи* (6.6). *Тогда существуют такая достаточно малая окрестность точки x_* – решения задачи* (6.1) *– и такое достаточно малое $\alpha > 0$, что итерационный процесс*

$$x_{k+1} = x_k + \alpha p(x_k) \tag{6.24}$$

сходится к x_* из этой окрестности и справедлива оценка

$$\| x_k - x_* \| \leqslant C(q+\epsilon)^k, \quad q < 1.$$

Доказательство. Если положить $g(x) = x + \alpha p(x)$, то x_* является неподвижной точкой отображения g, так как $p(x_*) = 0$. При этом

$$g'(x_*) = I_n + \alpha p'(x_*).$$

Поэтому собственные числа матрицы $g'(x_*)$ согласно лемме 6.3 имеют вид $1 - \alpha$ или $1 - \alpha\sigma$, где $\sigma > 0$ – ненулевые собственные числа матрицы $A = (I_n - \Pi_*) L''_{xx} (I_n - \Pi_*)$. Если $\alpha > 0$ выбрано так, что $1 - \alpha > -1$, $1 - \alpha\sigma > -1$, т.е. $\alpha < \min\{2,\ 2/\sigma_{\max}\}$, где $\sigma_{\max}$ – максимальное собственное число матрицы A, то все величины $1 - \alpha$ и $1 - \alpha\sigma$ будут лежать внутри отрезка $[-1, +1]$ и поэтому согласно теореме Островского итерационный процесс (6.24) будет локально сходящимся. Вообще говоря, доказанная теорема не дает точной оценки для метода линеаризации, так как процесс (6.24) идет с постоянным шагом α, о способе выбора которого ничего не говорится. Однако теорема 6.1 может служить ориентиром, давая общее представление о характере сходимости. В частности, любопытно и важно такое ее следствие.

Следствие. *Если $|I_*| = n$, т.е. в решении активными являются n линейно независимых ограничений, то при $\alpha = 1$ скорость сходимости процесса сверхлинейная.*

Доказательство. Если $|I_*| = n$ и векторы $f'_i(x_*)$ линейно независимы, то матрица $f'(x_*)$ обратима и поэтому

$$\Pi_* = f'^*(x_*) [f'(x_*) f'^*(x_*)]^{-1} f'(x_*) =$$

$$= f'^*(x_*) (f'^*(x_*))^{-1} (f'(x_*))^{-1} f'(x_*) = I_n.$$

Поэтому согласно формуле (6.22)

$$p'(x_*) = -\Pi_* = -I_n,$$

так что все собственные числа этой матрицы равны -1. Значит, $1 - \alpha = 0$ при $\alpha = 1$, т.е.

$$g'(x_*) = I_n - p'(x_*) = 0,$$

поэтому будет справедлива оценка сходимости процесса (6.22) в виде

$$\| x_k - x_* \| \leqslant C(\epsilon)\epsilon^k,$$

показывающая, что x_k стремится к x_* быстрее любой геометрической прогрессии.

3. Предварительные леммы. В следующем пункте этого параграфа будет сформулирован алгоритм решения задачи (6.1), обладающий высокой скоростью сходимости. Здесь мы докажем ряд вспомогательных результатов, которые, с одной стороны, дают основу для формулировки алгоритма, а с другой стороны, позволяют его строго обосновать.

Обозначим через $A(x, h)$ матрицу размеров $n \times n$ с элементами

$$h^{-2}\,[L(x+(e_i+e_j)h,\ u(x)) - L(x+e_ih,\ u(x)) -$$
$$- L(x+e_jh,\ u(x)) + L(x,\ u(x))],$$

где e_i – единичные орты пространства $\mathbf{R}^n$. Нетрудно видеть, что $A(x,h)$ есть приближение с точностью до порядка h к матрице $L''_{xx}(x,\ u(x))$. Заметим, что во всем изложении этого пункта рассмотрение ведется в достаточно малой окрестности точки x_* и поэтому вектор $u(x)$ согласно лемме 6.2 однозначно определен. То же можно будет сказать и обо всех остальных величинах, встречающихся в дальнейшем и связанных с решением вспомогательной задачи (6.6).

Пусть $B(x, h) = -[\Pi(x) + (I - \Pi(x))\,A(x, h)]$.

Лемма 6.4. *В некоторой окрестности точки x_* имеет место оценка*

$$\|p'(x_*) - B(x, h)\| \leqslant C(\|x - x_*\| + h).$$

Доказательство. Действительно, $A(x, h)$ отличается от $L''_{xx}(x, u(x))$ на величину порядка h, а выражение

$$-[\Pi(x) + (I - \Pi(x))\,L''_{xx}(x, u(x))]$$

дифференцируемо по x и поэтому в силу формулы (6.22) отличается от $p'(x_*)$ на величину порядка $\|x - x_*\|$.

Лемма 6.5. *В некоторой окрестности точки x_* справедлива оценка*

$$\|(p'(x_*))^{-1}\,p(x) - B^{-1}(x, h)\,p(x)\| \leqslant C(\|x - x_*\| + h)\,\|x - x_*\|.$$

Доказательство. Действительно, пусть

$$p'(x_*)\,y = -p(x),\quad B(x, h)\,\bar{y} = -p(x).$$

Тогда легко подсчитать, что

$$\bar{y} - y = (p'(x_*))^{-1}\,(B(x, h) - p'(x_*))\,B^{-1}(x, h)\,p(x). \tag{6.25}$$

Так как $p(x_*) = 0$ и вектор $p(x)$ дифференцируем, то $\|p(x)\|$ имеет тот же порядок малости, что и $\|x - x_*\|$. Учитывая это замечание и лемму 6.4, получаем из (6.25) утверждение леммы 6.5.

Лемма 6.6. *В достаточно малой окрестности точки x_* справедлива оценка*

$$\|x - B^{-1}(x, h)\,p(x) - x_*\| \leqslant C(2\,\|x - x_*\| + h)\,\|x - x_*\|.$$

Доказательство. Так как производная вектора $p(x)$ удовлетворяет условию Липшица, то нетрудно получить оценку

$$\|p(x) - p'(x_*)\,(x - x_*)\| \leqslant C\,\|x - x_*\|^2. \tag{6.26}$$

Для упрощения дальнейших выкладок без ограничения общности будем считать, что $x_* = 0$. Тогда

$$\| x - B^{-1}(x, h) p(x) \| \leqslant \| x - (p'(x_*))^{-1} p(x) \| +$$

$$+ \| (p'(x_*))^{-1} p(x) - B^{-1}(x, h) p(x) \| \leqslant$$

$$\leqslant \| x - (p'(x_*))^{-1} [p'(x_*) x + (p(x) - p'(x_*) x)] \| +$$

$$+ C(\| x \| + h) \| x \| \leqslant C \| x \|^2 + C(\| x \| + h) \| x \|,$$

где использованы лемма 6.5 и оценка (6.26).

Л е м м а 6.7. *Пусть y – решение системы уравнений*

$$B(x, h) y = -p(x). \tag{6.27}$$

Тогда y является решением системы уравнений

$$\begin{aligned} &A(x, h) y + (f_0'(x))^* + (f'(x))^* v = 0, \\ &f'(x) y + f(x) = 0 \end{aligned} \tag{6.28}$$

относительно неизвестных $y \in \mathbf{R}^n$ и $v \in \mathbf{R}^l$, где $l = |I_*|$ – число элементов множества I_*. Справедливо и обратное утверждение.

Д о к а з а т е л ь с т в о. Из выражения для $B(x, h)$ и формулы (6.13) для $p(x)$ следует, что уравнение (6.27) можно переписать в виде

$$\Pi(x) y + (I - \Pi(x)) A(x, h) y = -(I - \Pi(x))(f_0'(x))^* - f'^*(x) v(x). \tag{6.29}$$

Так как справедливы легко проверяемые соотношения

$$\Pi^*(x) = \Pi(x), \quad \Pi^2(x) = \Pi(x), \quad \Pi(x)(I - \Pi(x)) = 0, \quad \Pi(x) f'^*(x) = f'^*(x),$$

то, умножая (6.29) на $\Pi(x)$ и $(I - \Pi(x))$, получаем

$$\Pi(x) y = -f'^*(x) v(x), \tag{6.30}$$

$$(I - \Pi(x)) [A(x, h) y + (f_0'(x))^*] = 0. \tag{6.31}$$

Умножая соотношение (6.30) на $f'(x)$, с учетом формул (6.14), (6.15) получаем

$$f'(x) y = -f(x). \tag{6.32}$$

Далее, оператор $\Pi(x)$ есть оператор проектирования на подпространство, натянутое на векторы $(f_i'(x))^*$, $i \in I_*$. Равенство (6.31) означает, что вектор $A(x, h) y + (f_0'(x))^*$ лежит в этом подпространстве, т.е.

$$A(x, h) y + (f_0'(x))^* = -f'^*(x) v. \tag{6.33}$$

Соотношения (6.32) и (6.33) доказывают лемму.

4. Алгоритм линеаризации с ускоренной сходимостью. Покажем теперь, как может быть модифицирован метод линеаризации, чтобы была достигнута высокая скорость сходимости. Мы будем предполагать, что выполняются условия сходимости метода линеаризации, изложенные в начале § 4, а также основные допущения 1) – 3), сделанные в начале этого параграфа.

Пусть выбраны $N > 0$, $\delta > 0$, начальное приближение x_0, а также числа $0 < \gamma < 1$, $0 < \epsilon < 1$, $h > 0$. Положим $C_0 = +\infty$.

Опишем общий шаг алгоритма. Пусть x_k и C_k уже построены.

1. Решая задачу (6.6) при $x = x_k$, вычисляем $p_k = p(x_k)$, $u_k^i = u^i(x_k)$.

2. Если $\|p_k\| \leqslant C_k$, то полагаем $h_k = \min(h, \|p_k\|)$ и вычисляем y_k, решая систему уравнений

$$A(x_k, h_k)y + (f_0'(x_k))^* + \sum_{i \in I(x_k)} v^i (f_i'(x_k))^* = 0,$$
$$f_i'(x_k)y + f_i(x_k) = 0, \quad i \in I_\delta^0(x_k). \tag{6.34}$$

Если система не имеет решения, то полагаем $C_{k+1} = \gamma \|p_k\|$ и переходим к 4.

Если $\|p_k\| > C_k$, то полагаем $C_{k+1} = C_k$ и переходим к 4.

3. Если система (6.34) имеет решение y_k, то полагаем

$$\bar{x} = x_k + y_k \tag{6.35}$$

и вычисляем, решая задачу (6.6) при $x = \bar{x}$, вектор $p(\bar{x})$.

Если

$$\|p(\bar{x})\| \leqslant \gamma \|p_k\|, \tag{6.36}$$

то полагаем $x_{k+1} = \bar{x}$, $C_{k+1} = \gamma \|p_k\|$ и переходим к 1.

Если $\|p(\bar{x})\| > \gamma \|p_k\|$, то полагаем $C_{k+1} = \gamma \|p_k\|$ и переходим к следующему шагу.

4. Полагая вначале $\alpha = 1$, дробим его путем деления пополам до выполнения неравенства

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant f_0(x_k) + NF(x_k) - \alpha\epsilon \|p_k\|^2. \tag{6.37}$$

Полагаем $x_{k+1} = x_k + \alpha p_k$. Переходим к 1.

Т е о р е м а 6.2. *Пусть выполнены основные предположения* п. 1 § 4. *Тогда сформулированный алгоритм строит последовательность x_k, для которой $F(x_k) \to 0$, и в любой предельной точке этой последовательности удовлетворяются необходимые условия экстремума. Если дополнительно выполнены основные допущения* п. 1 *этого параграфа и решение задачи* (6.1) – *точка x_* – является единственной точкой, в которой выполнены необходимые условия экстремума, то последовательность x_k квадратично сходится к x_*, причем для достаточно больших k переход от x_k к x_{k+1} будет осуществляться согласно шагу* 3 *алгоритма.*

Д о к а з а т е л ь с т в о. Как видно из построения алгоритма, переход от x_k и x_{k+1} происходит либо по формулам (6.34), (6.35), либо путем использования метода линеаризации с выбором шага согласно формуле (6.37). При этом вся последовательность $\{x_k\}$ разбивается на отрезки $\{x_{k_j}, x_{k_j+1}, \ldots, x_{q_j}, \ldots, x_{k_{j+1}-1}\}$, $j = 1, \ldots, q_j \geqslant k_j$, обладающие следующими свойствами. Для индексов $k = k_j, k_j + 1, \ldots, q_j - 1$ переход от x_k к x_{k+1} происходит по формулам (6.34), (6.35) и при этом

$$\|p_{k+1}\| \leqslant \gamma \|p_k\|, \quad k = k_j, \ldots, q_j - 1.$$

Таким образом,

$$\|p_{q_j}\| \leqslant \gamma^{(q_j - k_j)} \|p_{k_j}\|.$$

Для индексов k от q_j до $k_{j+1} - 1$ работает метод линеаризации. Согласно алгоритму $C_{q_j} = \gamma \|p_{q_j}\|$ и $C_k = C_{q_j}$ для всех $k \geqslant q_j$, пока не выполнится неравенство $\|p_{k_{j+1}}\| \leqslant C_{k_{j+1}} = C_{q_j} = \gamma \|p_{q_j}\|$. Первым выполнением этого не-

равенства и определяется номер k_{j+1}. Как показано при исследовании метода линеаризации, это неравенство обязательно выполнится. Таким образом,

$$\| p_{k_{j+1}} \| \leqslant \gamma \| p_{q_j} \| \leqslant \gamma^{(q_j - k_j)+1} \| p_{k_j} \|.$$

Отсюда следует, что $\| p_{k_j} \| \to 0$, а поэтому согласно § 4 любая сходящаяся подпоследовательность последовательности $\{x_{k_j}\}$ будет иметь пределом точку, в которой выполняются ограничения и необходимые условия экстремума. Этим первое утверждение теоремы доказано.

Перейдем к доказательству второго утверждения. Рассмотрим блочную матрицу

$$\begin{bmatrix} L''_{xx}(x_*, u_*) & f'^*(x_*) \\ f'(x_*) & 0 \end{bmatrix} \tag{6.38}$$

и покажем, что эта матрица невырождена.

В самом деле, если эта матрица вырождена, то существуют не равные нулю одновременно векторы $y \in \mathbf{R}^n$, $v \in \mathbf{R}^l$, удовлетворяющие системе

$$L''_{xx}(x_*, u_*)\, y + f'^*(x_*)\, v = 0,$$
$$f'(x_*)\, y = 0.$$

Умножая первое из этих соотношений скалярно на y, с учетом второго получаем

$$(y, L''_{xx}(x_*, u_*)\, y) = 0,$$
$$f'(x_*)\, y = 0,$$

откуда следует, что $y = 0$, так как выполнены соотношения (6.4), (6.5). Но тогда при ненулевом v имеет место равенство $f'^*(x_*)\, v = 0$, что противоречит линейной независимости векторов $f'_i(x_*)$, $i \in I_*$.

Согласно лемме 6.2 $I_0^0(x) = I_*$, если x достаточно близко к x_*. Поэтому матрица системы (6.34) будет иметь вид

$$\begin{bmatrix} A(x_k, h_k) & f'^*(x_k) \\ f'(x_k) & 0 \end{bmatrix}, \tag{6.39}$$

если только x_k близко к x_*. Учитывая, что $h_k = \min(h, \| p_k \|) \leqslant \| p(x_k) \|$ и, значит, h_k также мало, можно утверждать, что матрица (6.39) близка к матрице (6.38) и также невырождена. Поэтому система (6.34) для достаточно больших $k = k_j$ будет разрешима. В этом случае система (6.34) согласно лемме 6.7 эквивалентна уравнению

$$B(x_k, h_k)\, y_k = -p(x_k).$$

так что формула (6.35) дает

$$\bar{x} = x_k - B^{-1}(x_k, h_k)\, p(x_k). \tag{6.40}$$

Из леммы 6.6 теперь следует, что

$$\| \bar{x} - x_* \| \leqslant C(2\| x_k - x_* \| + h_k) \| x_k - x_* \|. \tag{6.41}$$

Заметим, что по лемме 6.2 в окрестности x_*

$$p(x) = p'(x_*)(x - x_*) + \omega(x), \quad \| \omega(x) \| \leqslant C \| x - x_* \|^2. \tag{6.42}$$

Можно показать, что если матрица (6.38) невырождена, то невырождена и матрица $p'(x_*)$. Поэтому существует такое число $m > 0$, что

$$\| p'(x_*)(x - x_*) \| \geqslant m \| x - x_* \|.$$

Отсюда

$$\| p(x) \| \geqslant \| p'(x_*)(x - x_*) \| - \| \omega(x) \| \geqslant m \| x - x_* \| - C \| x - x_* \|^2 \geqslant (m - C \| x - x_* \|) \| x - x_* \|.$$

Значит, для достаточно малых $\| x - x_* \|$

$$\| p(x) \| \geqslant (m/2) \| x - x_* \|. \tag{6.43}$$

Возвратимся теперь к формуле (6.41). Так как $h_k \leqslant \| p(x_k) \| \leqslant C \| x_k - x_* \|$, то, переобозначая константы, формулу (6.41) можно переписать в виде

$$\| \bar{x} - x_* \| \leqslant C \| x_k - x_* \|^2, \quad k = k_j. \tag{6.44}$$

Теперь

$$\| p(\bar{x}) \| \leqslant C \| \bar{x} - x_* \| \leqslant C_1 \| x_k - x_* \|^2 \leqslant ((2C_1/m) \| x_k - x_* \|)(m/2) \| x_k - x_* \| \leqslant ((2C_1/m) \| x_k - x_* \|) p(x_k).$$

Если

$$(2C_1/m) \| x_k - x_* \| \leqslant \gamma,$$

то окончательно получаем: в малой окрестности точки x_*

$$\| p(x_{k+1}) \| = \| p(\bar{x}) \| \leqslant \gamma \| p(x_k) \|.$$

Таким образом, если $k = k_j$ достаточно велико, то алгоритм осуществляет переход от x_k к x_{k+1} по формулам (6.34), (6.35) и справедлива оценка (6.44). Эта оценка также показывает, что если x_k лежит в нужной окрестности x_* и $C \| x_k - x_* \| < 1$, то x_{k+1} будет также лежать в этой окрестности.

Резюмируем сказанное. При достаточно большом j точка x_{k_j} лежит в окрестности x_*, в которой выполняются все предположения, при которых справедливы проведенные выше выкладки. Справедливо соотношение

$$\| x_{k+1} - x_* \| \leqslant C \| x_k - x_* \|^2, \tag{6.45}$$

и точка x_{k+1} снова лежит в нужной окрестности. Тем самым доказано, что, начиная с некоторого k, будет справедлива формула (6.45), доказывающая квадратичную сходимость процесса.

5. Линейные преобразования задачи. Сделаем теперь небольшое отступление и рассмотрим, как меняются некоторые параметры задачи (6.1) и вспомогательной задачи (6.6) при преобразованиях координат и умножении функций f_i на положительные константы.

Напомним, что если x_* – решение задачи (6.1), u_* – соответствующий вектор множителей Лагранжа, то

$$f_0'(x_*) + \sum_{i=1}^{m} u_*^i f_i'(x_*) = 0,$$
$$u_*^i \geqslant 0, \quad u_*^i f_i(x_*) = 0, \quad i \in I. \tag{6.46}$$

Пусть $\widetilde{f}_i = a_i f_i$, $a_i > 0$, $i = 0, \ldots, m$. Очевидно, что решение задачи

$$\min_x \{ \widetilde{f}_0(x) : \widetilde{f}_i(x) \leqslant 0, \ i \in I \}$$

совпадает с x_*. С другой стороны, $\widetilde{f}_i' = a_i f_i'$ и (6.46) легко преобразуется к виду

$$\widetilde{f}_0'(x_*) + \sum_{i=1}^{m} \widetilde{u}_*^i \widetilde{f}_i'(x_*) = 0,$$

$$\widetilde{u}_*^i \geqslant 0, \ \widetilde{u}_*^i \widetilde{f}_i(x_*) = 0, \ i \in I,$$

где

$$\widetilde{u}_*^i = (a_0/a_i)\, u_i. \tag{6.47}$$

Таким образом, умножение всех функций f_i на положительные константы приводит к преобразованию (6.47) множителей Лагранжа.

Посмотрим теперь, что произойдет, если делается преобразование координат

$$x = Ay, \tag{6.48}$$

где A – невырожденная матрица. Напомним, что если $f(x)$ – дважды дифференцируемая функция, $f'(x)$ – ее градиент (вектор-строка), а $f''(x)$ – матрица вторых производных, то для функции $\widetilde{f}(y) = f(Ay)$ получаем

$$\widetilde{f}'(y) = f'(x) A, \tag{6.49}$$

$$\widetilde{f}''(y) = A^* f''(x) A, \tag{6.50}$$

где в левых частях штрихи уже означают производные по y.

Рассмотрим теперь задачу

$$\min_y \{ \widetilde{f}_0(y) : \widetilde{f}_i(y) \leqslant 0, \ i \in I \}. \tag{6.51}$$

Очевидно, что ее решение есть $y^* = A^{-1} x^*$. Если умножить (6.46) справа на A, то с учетом (6.49) сразу же получим необходимые условия экстремума для задачи (6.51). Таким образом, при преобразованиях координат множители Лагранжа не меняются.

Запишем вспомогательную задачу метода линеаризации для задачи (6.51). Легко видеть, что если x и y связаны соотношением (6.48), то $I_\delta(y) = I_\delta(x)$ и, с учетом (6.49), вспомогательная задача приобретает вид

$$\min_{\widetilde{p}} \{ f_0'(x) A\widetilde{p} + \tfrac{1}{2}(\widetilde{p}, \widetilde{p}) : f_i'(x) A\widetilde{p} + f_i(x) \leqslant 0, \ i \in I_\delta(x) \}. \tag{6.52}$$

При выводе (6.52) мы воспользовались тем, что $\widetilde{f}'$ – вектор-строка, и поэтому

$$(\widetilde{f}'(y), \widetilde{p}) = f'(x) A\widetilde{p},$$

где справа произведение вычисляется по правилам умножения матриц.

Обозначим решение задачи (6.52) через $\widetilde{p}(y)$. Положим в (6.52) $p = A\widetilde{p}$. Тогда (6.52) превратится в задачу

$$\min_p \{ f_0'(x) p + \tfrac{1}{2}(A^{-1} p, A^{-1} p) : f_i'(x) p + f_i(x) \leqslant 0, \ i \in I_\delta(x) \}. \tag{6.53}$$

Положив

$$C = (A^{-1})^* A^{-1}, \tag{6.54}$$

получим задачу

$$\min_p \{ f_0'(x)\, p + \tfrac{1}{2}(p, Cp):\ f_i'(x)\, p + f_i(x) \leqslant 0,\ i \in I_\delta(x) \}. \tag{6.55}$$

Итак, если $p_C(x)$ – решение задачи (6.55), то имеет место однозначная связь

$$p_C(x) = A\tilde{p}(y). \tag{6.56}$$

Заметим также, что в силу сказанного выше множители Лагранжа задач (6.52) и (6.55) совпадают.

Представим теперь, что мы выполнили преобразование (6.48) и применяем метод линеаризации к задаче (6.51). Тогда переход из точки y в новую точку $\bar{y} = y + \alpha\tilde{p}(y)$ происходит путем выбора шага α по правилу: делим пополам $\alpha = 1$ до выполнения неравенства

$$\tilde{f}_0(y + \alpha\tilde{p}(y)) + N\tilde{F}(y + \alpha\tilde{p}(y)) \leqslant \tilde{f}_0(y) + N\tilde{F}(y) - \alpha\epsilon \| \tilde{p}(y) \|^2.$$

Но $\tilde{f}_0(y) = f_0(Ay)$, $\tilde{F}(y) = F(Ay)$, $x = Ay$, и поэтому последнее неравенство приобретает вид

$$f_0(Ay + \alpha A\tilde{p}(y)) + NF(Ay + \alpha A\tilde{p}(y)) \leqslant$$
$$\leqslant f_0(Ay) + NF(Ay) - \alpha\epsilon \| \tilde{p}(y) \|^2,$$

или, с учетом (6.54), (6.56),

$$f_0(x + \alpha p_C(x)) + NF(x + \alpha p_C(x)) \leqslant$$
$$\leqslant f_0(x) + NF(x) - \alpha\epsilon(p_C(x),\ Cp_C(x)). \tag{6.57}$$

Итак, на самом деле нет необходимости переходить к новым координатам. Вместо этого все вычисления можно производить в старых координатах, вычисляя $p_C(x)$ как решение задачи (6.55) и переходя в новую точку $\bar{x} = x + \alpha p_C(x)$, выбрав шаг α из условия (6.57).

С другой стороны, если выбрана произвольная строго положительная матрица C (т.е. $(p, Cp) > 0$ при $p \neq 0$), то известно [46], что ее можно представить в виде $C = B^*B$ (при этом B можно даже выбрать верхней треугольной матрицей). Полагая $A = B^{-1}$, получаем, что процесс минимизации, определяемый формулами (6.55), (6.57), эквивалентен стандартному методу линеаризации в координатах y, связанных с x соотношениями

$$x = Ay = B^{-1}y, \quad C = B^*B. \tag{6.58}$$

Интересно вычислить в новых координатах y матрицу $\tilde{p}'(y_*)$. Для этого воспользуемся формулой (6.22). В новых координатах с учетом того, что множители Лагранжа не меняются, справедливо

$$L''_{yy}(y_*, u_*) = A^* L''_{xx}(x_*, u_*) A,$$

а $\tilde{\Pi}_*$ – оператор проектирования на подпространство, ортогональное под-

пространству $\widetilde{f}''(y_*)\widetilde{p} = 0$, или $f'(x_*)A\widetilde{p} = 0$. Поэтому формула (6.22) приобретает вид

$$\widetilde{p}'(y_*) = -[\widetilde{\Pi}_* + (I_n - \widetilde{\Pi}_*)A^* L''_{xx}(x_*, u_*)A]. \tag{6.59}$$

Допустим теперь, что матрица $L''_{xx}(x_*, u_*)$ строго положительно определена. Тогда ее можно представить в виде

$$L''_{xx}(x_*, u_*) = (A_0^{-1})^* A_0^{-1}. \tag{6.60}$$

Если теперь при преобразовании координат положить $A = A_0$, то

$$A_0^* L''_{xx}(x_*, u_*) A_0 = I_n$$

и формула (6.59) приобретает вид

$$\widetilde{p}'(y_*) = -[\widetilde{\Pi}_* + (I_n - \widetilde{\Pi}_*) I_n] = -I_n.$$

Итак, если преобразование координат $x = A_0 y$ выбрано из условия (6.60), то $\widetilde{p}'(y_*) = -I_n$.

6. Модификации метода линеаризации. Из сказанного в п. 5 вытекает, что при реализации метода линеаризации не обязательно использовать вспомогательную задачу (6.6), а можно исходить из решения задачи (6.55) и выбирать шаг, исходя из формулы (6.57). Мы пойдем дальше и будем менять матрицу C от итерации к итерации.

Модифицированный алгоритм линеаризации. Пусть x_0 выбрано, C_k — строго положительно определенные симметричные матрицы, $k = 0, 1, \ldots, N > 0$, $\delta > 0$, $0 < \epsilon < 1$. Если x_k уже построено, то x_{k+1} строится по следующему правилу:

1. Решается задача квадратичного программирования

$$\min_p \left\{ (f'_0(x_k), p) + \tfrac{1}{2}(p, C_k p):\ (f'_i(x_k), p) + f_i(x_k) \leqslant 0,\ i \in I_\delta(x_k) \right\}. \tag{6.61}$$

Пусть ее решение p_k существует, u_k^i — соответствующие множители Лагранжа.

2. Начиная с $\alpha = 1$, последовательно делим пополам α до выполнения неравенства

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant f_0(x_k) + NF(x_k) - \alpha\epsilon(p_k, C_k p_k). \tag{6.62}$$

Получаем величину шага $\alpha_k = \alpha$.

3. Полагаем $x_{k+1} = x_k + \alpha_k p_k$ и переходим к 1.

Теорема 6.3. *Пусть выполнены следующие условия*:

а) *множество*

$$\Omega_0 = \left\{ f_0(x) + NF(x) \leqslant f_0(x_0) + NF(x_0) \right\}$$

компактно;

б) *градиенты* $f'_i(x)$ *в этой области удовлетворяют условию Липшица с константой* L;

в) *существуют такие числа* $M \geqslant m > 0$, *что*

$$m \| p \|^2 \leqslant (p, C_k p) \leqslant M \| p \|^2;$$

г) *на всех итерациях выполняется соотношение*

$$\sum_i u_k^i \leqslant N.$$

Тогда $F(x_k) \to 0$ *и любая предельная точка последовательности* $\{x_k\}$ *удовлетворяет ограничениям задачи* (6.1) *и необходимым условиям экстремума.*

Доказательство. Необходимые и достаточные условия минимума для решения p_k задачи (6.61) имеют вид

$$C_k p_k^* + f_0'(x_k) + \sum_{i \in I_\delta(x_k)} u_k^i f_i'(x_k) = 0,$$
$$u_k^i \geqslant 0, \quad u_k^i [(f_i'(x_k), p_k) + f_i(x_k)] = 0, \quad i \in I_\delta(x_k). \tag{6.63}$$

Умножая первое из этих соотношений скалярно на p_k, получаем соотношение

$$(f_0'(x_k), p_k) = \sum_{i \in I_\delta(x_k)} u_k^i f_i(x_k) - (p_k, C_k p_k). \tag{6.64}$$

Повторяя теперь дословно все выкладки, предшествовавшие теореме 4.1, с той лишь разницей, что вместо соотношения (4.7) используется (6.64), получаем

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant f_0(x_k) + NF(x_k) -$$
$$- \alpha\epsilon(p_k, C_k p_k) \leqslant f_0(x_k) + NF(x_k) - \alpha\epsilon m \| p_k \|^2 \tag{6.65}$$

при $0 \leqslant \alpha \leqslant \bar{\alpha}_k$,

$$\bar{\alpha}_k = \min\left(1, \frac{\delta}{F(x_k) + K\| p_k \|}, \frac{1-\epsilon}{(N+1)L}\right).$$

Соотношение (6.65) позволяет повторить все рассуждения, приведенные в доказательстве теоремы 4.1, а тем самым доказать и теорему 6.3.

В формулировке теоремы 6.3 есть условие г), которое заранее невозможно проверить. Однако оно может эффективно проверяться по ходу процесса и может быть произведена корректировка выбранного первоначально значения N так же, как это делалось в § 4 при рассмотрении основного алгоритма метода линеаризации.

Воспользуемся теперь полученными результатами, чтобы построить алгоритм с высокой скоростью сходимости для задачи выпуклого программирования. Основная идея его будет состоять в следующем. Согласно предыдущему, если бы точка x_* и множители Лагранжа были известны, то можно было бы перейти к новым координатам y, взяв матрицу A_0 преобразования, исходя из формулы (6.60). При этом матрица $\tilde{p}'(y_*)$ равна $-I_n$, т.е. все ее собственные числа равны -1, и поэтому согласно сказанному в п.2 процесс $y_{k+1} = y_k + \tilde{p}(y_k)$ сходился бы к точке y_* быстрее любой геометрической прогрессии из некоторой окрестности точки y_*. Но точки x_* и u_* неизвестны. Поэтому попробуем использовать на каждом шаге $L_{xx}''(x_k, u_{k-1})$ вместо $L_{xx}''(x_*, u_*)$, где u_{k-1} – вектор множителей Лагранжа, взятый с предыдущей итерации. При этом глобальная сходимость будет обеспечиваться тем, что мы будем использовать только что изложенный модифици-

рованный алгоритм, а локально сверхлинейная сходимость – приемом, который уже был использован при построении алгоритма в п. 4.

Алгоритм для задачи выпуклого программирования. Пусть x_0, $N>0$, $\delta>0$, $0<\epsilon<1$, $0<\gamma<1$ заданы. Положим $C_0 = +\infty$, $u_{-1}=0$, $u_{-1}\in\mathbf{R}^m$.

Пусть точка x_k, вектор множителей Лагранжа u_{k-1} и число C_k уже построены.

1. Решаем задачу

$$\min_p \{(f_0'(x_k), p) + \tfrac{1}{2}(p, L_{xx}''(x_k, u_{k-1})p):$$

$$(f_i'(x_k), p) + f_i(x_k) \leqslant 0,\ i\in I_\delta(x_k)\}. \tag{6.66}$$

Пусть p_k – ее решение, u_k^i, $i\in I_\delta(x_k)$, – соответствующие множители Лагранжа. Полагаем $u_k^i=0$ для $i\notin I_\delta(x_k)$.

2. Если $\|p_k\| > C_k$, то полагаем $C_{k+1}=C_k$ и переходим к 4.

Если $\|p_k\| \leqslant C_k$, то переходим к 3.

3. Решаем задачу (6.66), в которой вместо x_k и u_{k-1} подставлены $\bar{x} = x_k + p_k$ и u_k соответственно. Если ее решение $\bar{p}$, $\bar{u}$ таково, что $\|\bar{p}\| \leqslant \gamma\|p_k\|$, то полагаем

$$x_{k+1}=x_k+p_k,\quad C_{k+1}=\gamma\|p_k\|$$

и переходим к 1. (Заметим, что теперь задача (6.66) для новых данных уже фактически решена, так как $p_{k+1}=\bar{p}$, а $u_{k+1}=\bar{u}$, и нет необходимости в ее повторном решении.)

Если же $\|\bar{p}\| > \gamma\|p_k\|$, то полагаем $C_{k+1}=\gamma\|p_k\|$ и переходим к 4.

4. Начиная с $\alpha=1$, путем деления пополам дробим α до выполнения неравенства

$$f_0(x_k+\alpha p_k)+NF(x_k+\alpha p_k) \leqslant$$
$$\leqslant f_0(x_k)+NF(x_k)-\alpha\epsilon(p_k, L_{xx}''(x_k, u_{k-1})p_k)$$

и таким образом вычисляем α_k. Полагаем $x_{k+1}=x_k+\alpha_k p_k$. Возвращаемся к 1.

Критерий останова: $\|p_k\| \leqslant w$, где w – заданная точность.

Сформулируем условия, гарантирующие сходимость алгоритма.

Теорема 6.4. *Пусть $f_i(x)$, $i=0, 1, \ldots, m$, – дважды непрерывно дифференцируемые выпуклые функции и существует число $m>0$ такое, что*

$$(p, f_0''(x)p) \geqslant m\|p\|^2 \tag{6.67}$$

для всех x и p. Пусть, кроме того, в точке минимума x_* градиенты активных ограничений линейно независимы и соответствующие множители Лагранжа положительны, а для последовательности $\{x_k\}$, порожденной алгоритмом, выполняется неравенство

$$\sum_{i=1}^{m} u_k^i \leqslant N.$$

Тогда последовательность $\{x_k\}$ сходится к точке x_* быстрее любой геометрической прогрессии.

Доказательство. Предположения, указанные в формулировке теоремы, позволяют сразу сделать несколько выводов. В силу условия сильной выпуклости (6.67) функции $f_0(x)$ рассматриваемая задача имеет единственную точку минимума x_*. Кроме того, из этого же условия следует, что $f_0(x) \to \infty$ при $x \to \infty$ и поэтому множество $\{ x: \ f_0(x) + NF(x) \leqslant C \}$ компактно при любом N и C, так как $F(x) \geqslant 0$. Далее, в силу выпуклости функций $f_i(x)$, $i = 1, \ldots, m$,

$$(p, f_i''(x)p) \geqslant 0.$$

Учитывая (6.67), получаем

$$(p, L_{xx}''(x_k, u_{k-1})p) \geqslant (p, f_0''(x)p) \geqslant m \| p \|^2.$$

Так как множители Лагранжа u_k равномерно ограничены, то верно и неравенство

$$(p, L_{xx}''(x_k, u_{k-1})p) \leqslant M \| p \|^2.$$

Таким образом, выполнены все условия теоремы 6.3, если положить в ней $C_k = L_{xx}''(x_k, u_{k-1})$. Поэтому, если бы применялся модифицированный алгоритм линеаризации, то он бы порождал последовательность, сходящуюся к x_*.

Заметим теперь, что условия теоремы гарантируют и выполнение основных предположений 1) – 3) п. 1, так что верны леммы 6.1 и 6.2 и мы в дальнейшем можем пользоваться вытекающими из них результатами.

Перейдем к доказательству сходимости. Покажем, что последовательность C_k стремится к нулю. Действительно, C_k уменьшается по крайней мере в γ раз, как только $\| p_k \| \leqslant C_k$. Если число C_k к нулю не стремится, то это означает, что оно, начиная с некоторого момента, остается постоянным и все время $\| p_k \| > C_k$. Но тогда алгоритм использует только шаги 1 и 4, т.е. фактически работает как модифицированный метод линеаризации, который, как указано выше, будет сходиться. Но тогда, в противоречии с предположением, $p_k \to 0$ и, значит, в некоторый момент $\| p_k \| \leqslant C_k$, после чего $C_{k+1} = \gamma \| p_k \| \leqslant \gamma C_k$, в противоречии с тем, что C_k постоянно.

Итак, $C_k \to 0$. Пусть J – множество тех индексов k, при которых происходит изменение C_k. Для таких индексов $\| p_k \| \leqslant C_k$, и поэтому p_k стремится к нулю, когда $k \to \infty$, $k \in J$.

Если теперь обратиться к доказательству теоремы 4.1, которое фактически является основой доказательства теоремы 6.3, то, используя линейную независимость градиентов активных ограничений, получаем, что

$$x_k \to x_*, \quad u_k \to u_*, \tag{6.68}$$

когда $k \to \infty$, $k \in J$.

Покажем, теперь, что, начиная с некоторого момента, переходы от x_k к x_{k+1} будут происходить по формуле

$$x_{k+1} = x_k + p_k \tag{6.69}$$

согласно шагу 3 алгоритма.

Начнем со следующего замечания: $k \in J$, если $\| p_k \| \leqslant C_k$, и при этом делается попытка сделать шаг по формуле (6.69). Если она согласно шагу 3 алгоритма безуспешна, то работает модифицированный метод линеариза-

ции. Если вообще успешных попыток использовать формулу (6.69) было конечное число, то все время работал модифицированный метод линеаризации и $x_k \to x_*$, $u_k \to u_*$ при $k \to \infty$. Допустим теперь, что бесконечное число раз попытка применить формулу (6.69) была успешной.

Обозначим множество соответствующих индексов k через J^0. Очевидно, $J^0 \subseteq J$. Но успешность применения формулы (6.69) согласно шагу 3 алгоритма означает, что

$$\| p_{k+1} \| \leqslant \gamma \| p_k \|.$$

Поэтому $\| p_{k+1} \| \to 0$ при $k \to \infty$, $k \in J^0$. Отсюда, как и ранее, можно заключить, что

$$x_{k+1} \to x_*, \quad u_{k+1} \to u_*, \tag{6.70}$$

когда $k \to \infty$, $k \in J^0$.

Совмещая (6.68) и (6.70), получаем

$$x_{k+1} \to x_*, \; u_k \to u_*,$$

если $k \to \infty$, $k \in J^0$, и поэтому

$$L''_{xx}(x_{k+1}, u_k) \to L''_{xx}(x_*, u_*), \quad k \to \infty, \quad k \in J^0. \tag{6.71}$$

Пусть теперь A_0 вычислено, исходя из формулы (6.60). Сделаем преобразование координат $x = A_0 y$.

Пусть $\tilde{p}_0(y)$ – решение задачи (6.52) при $A = A_0$. Согласно сказанному в п. 4

$$\tilde{p}'_0(y_*) = -I_n.$$

С другой стороны, если $p_0(x) = p_{C_0}(x)$ – решение задачи (6.55) при $C = C_0 = L''_{xx}(x_*, u_*)$, то в соответствии с формулой (6.56)

$$p_0(x) = A_0 \tilde{p}_0(y) = A_0 \tilde{p}_0(A_0^{-1} x)$$

и поэтому

$$p'_0(x_*) = A_0 \tilde{p}'_0(y_*) A_0^{-1} = -I_n. \tag{6.72}$$

Отсюда вытекает, что

$$p_0(x) = -I_n(x - x_*) + \omega(x - x_*) = -(x - x_*) + \omega(x - x_*), \tag{6.73}$$

$$\lim_{x \to x_*} \omega(x - x_*) / \| x - x_* \| = 0. \tag{6.74}$$

Далее, согласно леммам 6.1 и 6.2 множество активных индексов во вспомогательных задачах (6.55) при $C = C_0$ и (6.66) при $k - 1 \in J^0$ и достаточно больших k (здесь используется (6.70)) совпадает с I_* – множеством активных индексов в основной задаче (6.1). Поэтому согласно необходимым условиям экстремума (6.63) имеют место соотношения

$$C_0 p_0(x_k) + f_0'^*(x_k) + f'^*(x_k) u_0(x_k) = 0,$$

$$f'(x_k) p_0(x_k) + f(x_k) = 0,$$

$$C_k p_k + f_0'^*(x_k) + f'^*(x_k) u_k = 0,$$

$$f'(x_k) p_k + f(x_k) = 0,$$

$$C_0 = L''_{xx}(x_*, u_*), \quad C_k = L''_{xx}(x_k, u_{k-1}), \quad k - 1 \in J^0.$$

Вычитая эти соотношения, получаем

$$C_k(p_k - p_0(x_k)) + f'^*(x_k)(u_k - u_0(x_k)) - (C_0 - C_k)p_0(x_k) = 0,$$
$$f'(x_k)(p_k - p_0(x_k)) = 0.$$

Пользуясь тем, что эта система, рассматриваемая относительно $(p_k - p_0(x_k))$, $(u_k - u_0(x_k))$, невырождена, нетрудно получить оценку

$$\begin{aligned} \| p_k - p_0(x_k) \| &\leqslant K \| C_0 - C_k \| \, \| p_0(x_k) \|, \\ \| u_k - u_0(x_k) \| &\leqslant K \| C_0 - C_k \| \, \| p_0(x_k) \|. \end{aligned} \tag{6.75}$$

Используя эти оценки, получаем

$$\begin{aligned} &\bar{x} = x_k + p_k = x_k + p_0(x_k) + (p_k - p_0(x_k)) = \\ &= x_* + \omega(x_k - x_*) + (p_k - p_0(x_k)), \\ &\| \bar{x} - x_* \| \leqslant \| \omega(x_k - x_*) \| + \| p_k - p_0(x_k) \| \leqslant \\ &\leqslant K \left[\frac{\| \omega(x_k - x_*) \|}{\| x_k - x_* \|} + \| C_0 - C_k \| \right] \| p_0(x_k) \| \leqslant \\ &\leqslant K_1 \left[\frac{\| \omega(x_k - x_*) \|}{\| x_k - x_* \|} + \| C_0 - C_k \| \right] \| x_k - x_* \|, \end{aligned} \tag{6.76}$$

где мы воспользовались тем, что в силу (6.73), (6.74) нормы $p_0(x)$ и $x - x_*$ имеют один и тот же порядок малости.

Мы не будем проводить дальше достаточно простые, но громоздкие формальные выкладки. Важно то, что при достаточно большом k в силу формул (6.71) и (6.74) коэффициент

$$q_k = K_1 \left[\frac{\| \omega(x_k - x_*) \|}{\| x_k - x_* \|} + \| C_0 - C_k \| \right] \tag{6.77}$$

как угодно мал и поэтому $x_{k+1} = \bar{x} = x_k + p_k$ гораздо лучше приближает x_*, чем x_k. Близость p_k и $p_0(x_k)$, обеспечиваемая формулами (6.75), гарантирует то, что $\| p_{k+1} \| \leqslant \gamma \| p_k \|$. Поэтому для достаточно больших k переход от итерации к итерации будет осуществляться по формуле (6.69), и, значит, если $k - 1 \in J^0$, то и $k \in J^0$. Отсюда следует, что, начиная с некоторого момента, J^0 содержит все индексы.

В силу (6.76)

$$\| x_{k+1} - x_* \| \leqslant q_k \| x_k - x_* \|,$$

а формула (6.77) показывает, что $q_k \to 0$. Этим сверхлинейная сходимость, а значит, и теорема доказаны.

В условиях теоремы 6.4, как и теоремы 6.3, неприятным является невозможность заранее проверить ограниченность множителей Лагранжа. Как уже отмечалось, этот недостаток несколько компенсируется тем, что это условие можно контролировать по ходу процесса. Кроме того, возможно, полезной будет следующая теорема, доказательство которой легко следует из формул, выведенных при доказательстве теоремы 6.4.

Т е о р е м а 6.5. *Пусть выполнены все условия теоремы* 6.4, *за исключением условия об ограниченности множителей Лагранжа. Тогда процесс, построенный согласно приведенному выше алгоритму, сходится к решению* x_* *задачи* (6.1) *быстрее любой геометрической прогрессии, если начальное приближение* x_0 *достаточно близко к* x_*.

В самом деле, если начальное приближение достаточно близко к x_*, то будет справедлива формула (6.76), алгоритм будет работать, используя только формулу (6.69), и величина константы N вовсе не понадобится.

Можно выделить еще один случай, когда ни величина N, ни множители Лагранжа не влияют на процесс работы алгоритма. А именно, если функции $f_i(x)$, $i = 1, \ldots, m$, задающие ограничения, линейны по x, то

$$L''_{xx}(x, u) = f''_0(x).$$

Далее, если $\delta = +\infty$, то во вспомогательной задаче участвуют все ограничения и в силу их линейности

$$f_i(x + p) = (f'_i(x), p) + f_i(x) \leqslant 0,$$

так что если начальная точка x этим ограничениям удовлетворяла, то им будут удовлетворять и все точки $x + \alpha p$. Поэтому $F(x + \alpha p) = 0$ и формула (6.65), используемая для выбора шага, приобретает вид

$$f_0(x_k + \alpha p_k) \leqslant f_0(x_k) - \epsilon\alpha(p_k, f''_0(x_k) p_k). \tag{6.78}$$

Таким образом, на основании вышесказанного можно заключить, что верна следующая теорема.

Т е о р е м а 6.6. *Пусть* $f_0(x)$ – *выпуклая дважды непрерывно дифференцируемая функция,* $f_i(x)$, $i = 1, \ldots, m$, – *линейные функции, т.е.*

$$f_i(x) = (a_i, x) - b_i,$$

и пусть

$$(p, f''_0(x) p) \geqslant m \| p \|^2, \quad m > 0.$$

Тогда приведенный выше алгоритм для выпуклого программирования, в котором вместо задачи (6.66) *решается задача*

$$\min_p \left\{ (f'_0(x_k), p) + \tfrac{1}{2} (p, f''_0(x_k) p) : \ (a_i, p) + f_i(x) \leqslant 0, \ i \in I \right\},$$

а для выбора шага используется формула (6.78), *сходится быстрее любой геометрической прогрессии с любого начального приближения* x_0, *удовлетворяющего ограничениям задачи.*

Глава 3.

ЗАДАЧА ДИСКРЕТНОГО МИНИМАКСА И АЛГОРИТМЫ

§ 7. ЗАДАЧА ДИСКРЕТНОГО МИНИМАКСА

Часто встречающаяся на практике задача дискретного минимакса состоит в минимизации максимума конечного числа функций. Формально, если $I = \{1, \ldots, m\}$ – конечное множество индексов и для каждого $i \in I$ задана функция $f_i(x)$, то задача состоит в минимизации функции

$$F(x) = \max_{1 \leqslant i \leqslant m} f_i(x), \tag{7.1}$$

когда x меняется во всем пространстве $\mathbf{R}^n$.

Можно было бы ставить задачу, потребовав, чтобы аргумент менялся в некоторой заданной области M. Но это лишило бы задачу ряда специфических особенностей и фактически привело бы к решению общей задачи нелинейного программирования. А задача

$$\min_x \{F(x): \ x \in M\}$$

очевидным образом эквивалентна задаче

$$\min_{x,\,\beta} \{\beta: \ f_i(x) \leqslant \beta, \ i \in I, \ x \in M\}, \tag{7.2}$$

которая в свою очередь эквивалентна общей задаче нелинейного программирования, так как очевидными рассуждениями любую задачу в форме (4.2) введением дополнительной переменной можно свести к виду (7.2).

Итак, цель этого параграфа – построение алгоритмов для решения задачи

$$\min_x \{F(x): \ x \in \mathbf{R}^n\}. \tag{7.3}$$

Согласно вышесказанному эта задача эквивалентна задаче

$$\min_{x,\,\beta} \{\beta: \ f_i(x) \leqslant \beta, \ x \in \mathbf{R}^n\}. \tag{7.4}$$

Выпишем некоторые обозначения и факты, которые будут необходимы для дальнейшего изложения.

Всюду в этом параграфе будет предполагаться, что функции $f_i(x)$, $i \in I$, непрерывно дифференцируемы и их градиенты удовлетворяют условию Липшица на каждом компакте. Кроме того, будет предполагаться, что множества

$$C_\alpha = \{x: \ F(x) \leqslant \alpha\} \tag{7.5}$$

компактны.

Очевидно, что сделанные предположения гарантируют существование минимума (7.3). Пусть x_* – какая-либо точка минимума. Тогда согласно п.7

§ 2 (теорема 2.13) в точке x_* выполняется условие: существует такой вектор $u_* \in \mathbf{R}^m$, что

$$\sum_{i=1}^{m} u_*^i f_i'(x_*) = 0,$$

$$u_*^i \geqslant 0, \quad u_*^i (f_i(x_*) - F(x_*)) = 0, \quad i = 1, \ldots, m, \tag{7.6}$$

$$\sum_{i=1}^{m} u_*^i = 1.$$

Положим

$$I_\delta(x) = \{ i \in I: \ f_i(x) \geqslant F(x) - \delta \}, \quad \delta > 0.$$

Основная идея дальнейшего изложения та же, что и при построении алгоритмов §§ 4 – 6. Отличие будет состоять в том, что используется специфика задачи (7.3), которая позволяет снять ряд сложно проверяемых требований на исходную задачу и алгоритм.

1. Вспомогательная задача. Пусть x – некоторая точка и A – симметричная строго положительно определенная матрица, так что

$$(p, Ap) \geqslant m \| p \|^2. \tag{7.7}$$

Рассмотрим вспомогательную задачу

$$\min_{p, \beta} \{ \beta + \tfrac{1}{2} (p, Ap): \ (f_i'(x), p) + f_i(x) - \beta \leqslant 0, \ i \in I_\delta(x) \}. \tag{7.8}$$

Так как значения $p = 0$, $\beta \geqslant F(x)$, очевидно, удовлетворяют ограничениям задачи (7.8), то она всегда разрешима.

В силу ограничений (7.8)

$$\beta \geqslant f_i(x) + (f_i'(x), p) \geqslant f_i(x) - \| p \| \| f_i'(x) \|, \quad i \in I_\delta(x),$$

т.е.

$$\beta \geqslant c_1(x) - \| p \| c_2(x),$$

$$c_1(x) = \min_i \{ f_i(x): \ i \in I_\delta(x) \},$$

$$c_2(x) = \max_i \{ \| f_i'(x) \|: \ i \in I_\delta(x) \}.$$

Поэтому, если β и p удовлетворяют ограничениям задачи (7.8), то

$$\beta + \tfrac{1}{2} (p, Ap) \geqslant c_1(x) - \| p \| c_2(x) + m \| p \|^2.$$

Эта оценка показывает, что если $\beta \to \infty$, $p \to \infty$, то целевая функция задачи (7.8) стремится к $+\infty$. Отсюда нетрудно сделать вывод, что минимум в задаче (7.8) достигается. Более того, если учесть, что при фиксированном p минимум по β в (7.8) достигается при

$$\beta = \max_i \{ (f_i'(x), p) + f_i(x): \ i \in I_\delta(x) \},$$

то задачу (7.8) можно переписать в виде

$$\min_p \{ \tfrac{1}{2} (p, Ap) + \max_i \{ (f_i'(x), p) + f_i(x): \ i \in I_\delta(x) \} \},$$

которая эквивалентна исходной. В силу (7.7) эта задача есть задача

минимизации сильно выпуклой функции [23], и поэтому ее минимум достигается в единственной точке.

Все сказанное позволяет сформулировать следующую теорему.

Теорема 7.1. *Если выполнено условие* (7.7), *то задача* (7.8) *имеет единственную точку минимума* $p_A(x)$, $\beta_A(x)$ *и*

$$\beta_A(x) = \max_i \{ (f_i'(x),\ p_A(x)) + f_i(x):\ i \in I_\delta(x) \}.$$

Для вывода необходимых условий экстремума и формулировки двойственной задачи построим функцию Лагранжа задачи (7.8). Согласно п.3 § 2

$$L(p,\ \beta,\ u) = \beta + \tfrac{1}{2}(p, Ap) + \sum_{i \in I_\delta(x)} u^i [(f_i'(x),\ p) +$$

$$+ f_i(x) - \beta] = [1 - \sum_{i \in I_\delta(x)} u^i]\,\beta + \tfrac{1}{2}(p,\ Ap) +$$

$$+ \Big(\sum_{i \in I_\delta(x)} f_i'(x)\, u^i,\ p \Big) + \sum_{i \in I_\delta(x)} u^i f_i(x). \qquad (7.9)$$

Согласно вышесказанному и теоремам 2.9, 2.6 для точки минимума $p = p_A(x)$, $\beta = \beta_A(x)$ существуют такие множители $u_A^i(x)$, что

$$u_A^i(x) \geqslant 0,\ \ u_A^i(x)\,[(f_i'(x),\ p_A(x)) + f_i(x) - \beta_A(x)] = 0 \qquad (7.10)$$

и при подстановке $u_A^i(x)$ в (7.9) $L(p, \beta, u_A(x))$ достигает минимума по $p \in \mathbf{R}^n$, $\beta \in \mathbf{R}^{-1}$ при $p = p_A(x)$, $\beta = \beta_A(x)$. Дифференцируя (7.9) по β и p и приравнивая производные нулю, получим

$$\sum_{i \in I_\delta(x)} u_A^i(x) = 1,\ \ Ap_A(x) + \sum_{i \in I_\delta(x)} u_A^i(x)\,(f_i'(x))^* = 0. \qquad (7.11)$$

Итак, соотношения (7.10), (7.11) являются необходимыми и достаточными условиями, которые связывают точку минимума и множители Лагранжа задачи (7.8).

Для построения задачи, двойственной к (7.8), согласно п. 4 § 2 следует в (7.9) вычислить минимум по p и β при фиксированных u^i, $i \in I_\delta(x)$, $u^i \geqslant 0$. Но

$$\varphi(u) = \min_{p,\,\beta} L(p,\ \beta,\ u) =$$

$$= \begin{cases} -\infty, & \sum\limits_{i \in I_\delta(x)} u^i \neq 1, \\ -\dfrac{1}{2}\Big(\sum\limits_{i \in I_\delta(x)} u^i f_i'(x),\ \sum\limits_{i \in I_\delta(x)} u^i A^{-1} (f_i'(x))^* \Big) + & \\ + \sum\limits_{i \in I_\delta(x)} u^i f_i(x), & \sum\limits_{i \in I_\delta(x)} u^i = 1. \end{cases}$$

Таким образом, согласно п. 4 § 2 задача, двойственная к (7.8), имеет вид

$$\max_{u \geqslant 0} \left\{ -\frac{1}{2}\Big(\sum_{i \in I_\delta(x)} u^i f_i'(x), \Big/ \sum_{i \in I_\delta(x)} u^i A^{-1}(f_i'(x))^* \Big) + \right.$$

$$\left. + \sum_{i \in I_\delta(x)} u^i f_i(x) : \sum_{i \in I_\delta(x)} u^i = 1 \right\}. \tag{7.12}$$

При этом $u_A^i(x)$ являются решением задачи (7.12). Соображения единственности $p_A(x)$, $\beta_A(x)$, которые использовались при доказательстве теоремы 3.3, показывают, что $p_A(x)$ может быть выражено через $u_A(x)$, исходя из формулы (7.11), т.е.

$$p_A(x) = -\sum_{i \in I_\delta(x)} u_A^i(x) A^{-1}(f_i'(x))^*. \tag{7.13}$$

2. Некоторые оценки. Дальнейшее построение алгоритмов базируется на на ряде оценок изменения функций $f_i(x)$ при сдвигах вдоль направлений, определяемых решением задачи (7.8). Умножая скалярно второе из соотношений (7.11) на $p_A(x)$ и учитывая первое соотношение и (7.10), получаем

$$(p_A(x), Ap_A(x)) - \sum_{i \in I_\delta(x)} u^i f_i(x) + \beta_A(x) = 0.$$

Из этого соотношения с учетом (7.11) следует, что

$$(p_A(x), Ap_A(x)) \leqslant F(x) - \beta_A(x), \tag{7.14}$$

так как $f_i(x) \leqslant F(x)$.

Пусть теперь x_0 фиксировано, $\alpha = F(x_0)$ и в области C_α

$$K = \max_{x \in C_\alpha,\ i \in I} \| f_i'(x) \|,$$

L – соответствующая C_α константа Липшица для градиентов, т.е.

$$\| f_i'(x_1) - f_i'(x_2) \| \leqslant L \| x_1 - x_2 \|, \quad x_1, x_2 \in C_\alpha.$$

Используя известную из математического анализа формулу средних значений, получаем, что для некоторого θ, $0 \leqslant \theta \leqslant 1$,

$$f_i(x + \alpha p) = f_i(x) + \alpha(f_i'(x + \theta\alpha p), p) = f_i(x) + \alpha(f_i'(x), p) +$$

$$+ \alpha(f_i'(x + \theta\alpha p) + f_i(x), p) \leqslant f_i(x) + \alpha(f_i'(x), p) + \alpha^2 L \| p \|^2. \tag{7.15}$$

Пусть $i \in I_\delta(x)$ и

$$p = p_A(x). \tag{7.16}$$

Тогда, учитывая, что p удовлетворяет ограничениям задачи (7.8)* и неравенству (7.14), получаем при $0 \leqslant \alpha \leqslant 1$

$$f_i(x + \alpha p) \leqslant f_i(x) + \alpha(\beta_A(x) - f_i(x)) + \alpha^2 L \| p \|^2 =$$

$$= (1 - \alpha) f_i(x) + \alpha\beta_A(x) + \alpha^2 L \| p \|^2 \leqslant (1 - \alpha) F(x) +$$

$$+ \alpha\beta_A(x) + \alpha^2 L \| p \|^2 = F(x) - \alpha(F(x) - \beta_A(x)) +$$

$$+ \alpha^2 L \| p \|^2 \leqslant F(x) - \alpha(p, Ap) + \alpha^2 L \| p \|^2.$$

Итак, для $i \in I_b(x)$, $p = p_A(x)$ справедливо неравенство

$$f_i(x + \alpha p) \leqslant F(x) - \alpha(p, Ap) + \alpha^2 L \| p \|^2, 0 \leqslant \alpha \leqslant 1. \tag{7.17}$$

Пусть теперь $i \notin I_b(x)$. Тогда

$$f_i(x + \alpha p) = f_i(x) + \alpha(f_i'(x + \theta\alpha p), p) \leqslant F(x) - \delta + \alpha K \| p \|. \tag{7.18}$$

Но справедливо неравенство

$$F(x) - \alpha(p, Ap) \geqslant F(x) - \delta + \alpha K \| p \|, \tag{7.19}$$

если

$$\alpha \leqslant \frac{\delta}{(p, Ap) + K \| p \|}.$$

Поэтому, сопоставляя (7.17) – (7.19), получаем

$$F(x + \alpha p) \leqslant F(x) - \alpha(p, Ap) + \alpha^2 L \| p \|^2 \tag{7.20}$$

при

$$0 \leqslant \alpha \leqslant \min\left[1, \frac{\delta}{(p, Ap) + K \| p \|}\right]. \tag{7.21}$$

Пусть $0 < \epsilon < 1$. Тогда

$$(p, Ap) - \alpha L \| p \|^2 \geqslant \epsilon(p, Ap) \tag{7.22}$$

при

$$0 \leqslant \alpha \leqslant \frac{1 - \epsilon}{L} \frac{(p, Ap)}{\| p \|^2}. \tag{7.23}$$

Сопоставляя (7.20) – (7.23), убеждаемся в справедливости следующего результата.

Л е м м а 7.1. *Для параметра α, удовлетворяющего неравенству*

$$0 \leqslant \alpha \leqslant \min\left[1, \frac{\delta}{(p, Ap) + K \| p \|}, \frac{1 - \epsilon}{L} \frac{(p, Ap)}{\| p \|^2}\right],$$

справедлива оценка

$$F(x + \alpha p_A(x)) \leqslant F(x) - \alpha\epsilon(p_A(x), Ap_A(x)). \tag{7.24}$$

Отметим также справедливость следующего утверждения.

Л е м м а 7.2. *Функция $p_A(x)$ равна нулю тогда и только тогда, когда в точке x выполняются необходимые условия минимума функции $F(x)$.*

Действительно, если $p_A(x) = 0$, то $\beta_A(x) = F(x)$ и соотношения (7.10), (7.11) переходят в соотношения (7.6). Обратно, если соотношения (7.6) справедливы, то при $p_A(x) = 0$, $\beta_A(x) = F(x)$ они превращаются в соотношения (7.10), (7.11), которые являются достаточными условиями того, что $p_A(x)$ и $\beta_A(x)$ являются решением задачи (7.8).

3. Алгоритмы. Сформулируем теперь алгоритм метода линеаризации применительно к задаче о нахождении минимакса.

Пусть выбраны числа $\delta > 0$, $0 < \epsilon < 1$ и начальная точка x_0.

О б щ и й ш а г. Если приближение x_k уже построено, то выбираем симметричную строго положительно определенную матрицу A_k.

1. Решаем задачу (7.8) при $x = x_k$, $A = A_k$, и пусть

$p_k = p_{A_k}(x_k)$, $u_k = u_{A_k}(x_k)$.

2. Выбираем шаг α_k путем деления пополам единицы до первого выполнения неравенства

$$F(x_k + \alpha_k p_k) \leqslant F(x_k) - \alpha_k \epsilon(p_k, A_k p_k). \tag{7.25}$$

3. Полагаем $x_{k+1} = x_k + \alpha_k p_k$ и возвращаемся к 1.

Условие останова: $p_k = 0$ или $\| p_k \| \leqslant \epsilon_1$, где ϵ_1 – задаваемая точность.

Условия сходимости алгоритма даются следующей теоремой.

Теорема 7.2. *Если*

$\| A_k \| \leqslant M$, $(p, A_k p) \geqslant m \| p \|^2$

для всех k, $p \in \mathbf{R}^n$, *то любая предельная точка* x_* *последовательности* $\{x_k\}$ *удовлетворяет необходимым условиям экстремума* (7.6) *и* $\| p_k \| \to 0$.

Доказательство. Рассматриваемый алгоритм строит последовательность точек x_k, вдоль которой функция $F(x)$ убывает. Поэтому все точки x_k принадлежат компактному множеству

$C_\alpha = \{x: \ F(x) \leqslant \alpha\}$, $\alpha = F(x_0)$.

Из (7.11) следует, что

$$\| p_k \| = \| \sum_{i \in I_\delta(x_k)} u_k^i A_k^{-1} (f_i'(x_k))^* \| \leqslant$$

$$\leqslant \max_{i \in I} \| A_k^{-1} \| \| f_i'(x_k) \| \leqslant K/m.$$

Таким образом, все векторы p_k равномерно ограничены по норме. Из леммы 7.1, если огрубить в ней оценку на величину шага α, получаем

$$F(x_k + \alpha p_k) \leqslant F(x_k) - \alpha \epsilon(p_k, A_k p_k),$$

если $0 \leqslant \alpha \leqslant \alpha_0$, где

$$\alpha_0 = \min \left[1, \ \frac{\delta}{M(K/m)^2 + K^2/m} \cdot \frac{1 - \epsilon}{L} \ m \right] > 0.$$

Отсюда следует, что шаг α_k, получаемый делением пополам единицы до первого выполнения неравенства (7.25), не может быть меньше, чем половина α_0, т.е. $\alpha_k \geqslant \alpha_0/2$. Так как C_α – компакт, то функция $F(x)$ ограничена на нем снизу. Поэтому из (7.25) следует, что

$$\alpha_k (p_k, A_k p_k) \to 0,$$

или, так как $\alpha_k \geqslant 0{,}5\alpha_0$,

$$(p_k, A_k p_k) \geqslant m \| p_k \|^2 \to 0.$$

Таким образом, $p_k \to 0$.

Пусть теперь $J \subseteq \{0, 1, \ldots\}$ – какая-либо последовательность индексов k, вдоль которой $x_k \to x_*$. В силу компактности множества C_α такая точка x_* и последовательность J всегда существуют. Так как множители u_k^i

неотрицательны и в сумме равны единице, то без ограничения общности можно считать что $u_k^i \to u_*^i$, $k \in J$, и

$$u_*^i \geqslant 0, \quad \sum_{i=1}^{m} u_*^i = 1.$$

При этом мы полагали, что $u_k^i = 0$, $i \notin I_\delta(x_k)$, так что множители u_k^i определены при всех $i \in I$.

Далее, $\beta_k = \beta_{A_k}(x_k) \leqslant F(x_k)$ в силу (7.14). С другой стороны,

$$\beta_k \geqslant f_i(x_k) + (f_i'(x_k), p_k) \geqslant f_i(x_k) - K \| p_k \|, \quad i \in I_\delta(x_k).$$

Поэтому

$$F(x_k) \geqslant \beta_k \geqslant F(x_k) - K \| p_k \|,$$

и так как $\| p_k \| \to 0$, то

$$\beta_k \to F(x_*).$$

В силу (7.10), (7.11) и того, что по соглашению $u_k^i = 0$, $i \notin I_\delta(x_k)$,

$$u_k^i \geqslant 0, \quad u_k^i \left| (f_i'(x_k), p_k) + f_i(x_k) - \beta_k \right| = 0, \quad i \in I,$$

$$A_k p_k + \sum_{i=1}^{m} u_k^i (f_i'(x_k))^* = 0,$$

$$\sum_{i=1}^{m} u_k^i = 1.$$

Устремляя теперь $k \in J$ к бесконечности, из приведенных соотношений немедленно получаем соотношение (7.6), так как $p_k \to 0$, $u_k \to u_*$, $\beta_k \to F(x_*)$. Теорема доказана.

4. **Алгоритм при $A_k = I_n$.** Алгоритм, в котором все матрицы A_k совпадают с единичной, наиболее прост в реализации. Дело в том, что в этом случае двойственная задача к вспомогательной задаче (7.8) имеет простой вид

$$\max_{u \geqslant 0} \left\{ -\frac{1}{2} \Big\| \sum_{i \in I_\delta(x)} u^i (f_i'(x))^* \Big\|^2 + \right.$$

$$\left. + \sum_{i \in I_\delta(x)} u^i f_i(x); \quad \sum_{i \in I_\delta(x)} u^i = 1, \ u^i \geqslant 0 \right\}. \tag{7.26}$$

а решение исходной задачи через решение двойственной выражается по простой формуле

$$p(x) = -\sum_{i \in I_\delta(x)} u^i (f_i'(x))^*.$$

В дальнейшем при $A = I_n$ будем обозначать $p_A(x)$ и $u_A(x)$ просто через $p(x)$ и $u(x)$.

Остановимся более детально на скорости сходимости алгоритма при $A_k = I_n$. Для этого нам потребуются некоторые дополнительные предположения.

1. Пусть x_* – единственная точка минимума $F(x)$ и только в ней выполняются необходимые условия минимума (7.6).

2. Функции $f_i(x)$, $i \in I$, дважды непрерывно дифференцируемы.

3. Если

$$I_* = I_0(x_*) = \{ i \in I: f_i(x_*) = F(x_*) \},$$

то $u_*^i > 0$, $i \in I_*$, векторы $f_i'(x_*) - f_j'(x_*)$, $i \in I_*$, $i \neq j$, где j – любой индекс из I_*, линейно независимы.

4. $(p, L''_{xx}(x_*, u_*)\, p) > 0$ для всех $p \neq 0$, где

$$L(x, u) = \sum_{i=1}^{m} u^i f_i(x),$$

а L''_{xx} – матрица вторых производных от L по x.

В дальнейшем изложение в этом пункте аналогично § 6, и поэтому мы позволим себе несколько сократить доказательства.

Л е м м а 7.3. *Существует окрестность точки x_*, в которой $I_* \subseteq I_0(x)$.*

Доказательство аналогично доказательству леммы 6.1.

Л е м м а 7.4. *В некоторой окрестности точки x_* функции $p(x)$ и $u(x)$ определяются однозначно из решения системы*

$$p(x) + \sum_{i \in I_*} u^i(x) (f_i'(x))^* = 0, \tag{7.27}$$

$$(f_i'(x), p(x)) + f_i(x) = \beta(x), \quad i \in I_*, \tag{7.28}$$

$$\sum_{i \in I_*} u^i(x) = 1 \tag{7.29}$$

и являются дифференцируемыми функциями x.

Д о к а з а т е л ь с т в о. Очевидно, что при $x = x_*$ в силу (7.6) решением системы (7.27) является $p(x_*) = 0$, $u^i(x_*) = u_*^i$, $\beta(x_*) = F(x_*)$.

Пусть теперь j – произвольный индекс из I_* и

$$\widetilde{I}_* = I_* \setminus \{ j \},$$

$$\widetilde{f}_i(x) = f_i(x) - f_j(x), \quad i \in I_*, \quad i \neq j, \quad \widetilde{f}_0(x) = f_j(x).$$

Тогда, вычитая уравнение (7.28), соответствующее индексу j, из остальных уравнений (7.28) и сделав очевидные преобразования над (7.27) и (7.29), получаем, что система (7.27) – (7.29) эквивалентна системе

$$p(x) + (\widetilde{f}_0'(x))^* + \sum_{i \in \widetilde{I}_*} u^i(x) (\widetilde{f}_i'(x))^* = 0,$$

$$(\widetilde{f}_i'(x), p(x)) + \widetilde{f}_i(x) = 0, \quad i \in \widetilde{I}_*, \tag{7.30}$$

$$u_j(x) = 1 - \sum_{i \in \widetilde{I}_*} u^i(x).$$

Но система (7.30) полностью аналогична системе (6.10). Поэтому, рассуждая так же, как при доказательстве леммы 6.2, убедимся, что система (7.30), а значит и (7.27) – (7.29), имеет в окрестности точки x_* единственное решение, которое непрерывно дифференцируемо, и $u^i(x) \geqslant 0$ для $i \in I_*$. Сравнивая (7.27) – (7.29) с необходимыми и достаточными условиями минимума (7.10), (7.11), положив $u^i(x) = 0$, $i \notin I_*$, убеждаемся, что в выбранной окрестности решение системы (7.27) – (7.29) является одновременно решением вспомогательной задачи (7.8).

Пользуясь аналогичностью систем (7.30) и (6.10), можно дальше установить свойства собственных чисел матрицы $p'(x_*)$, что позволяет сформулировать теорему и следствие, аналогичные теореме 6.1 и ее следствию.

Т е о р е м а 7.3. *Если выполнены предположения* 1 – 4 *этого пункта, то существуют такая окрестность точки* x_* *и такое* $\alpha > 0$, *что процесс*

$$x_{k+1} = x_k + \alpha p(x_k)$$

сходится к x_* *со скоростью геометрической прогрессии со знаменателем, меньшим единицы.*

С л е д с т в и е. *Если выполнены предположения* 1 – 3 *и множество* I_* *содержит* $n+1$ *индекс, то процесс*

$$x_{k+1} = x_k + p(x_k)$$

сходится из некоторой окрестности точки x_* *сверхлинейно.*

Следствие требует небольшого комментария. Множество $\tilde{I}_*$ содержит на один индекс меньше, чем множество I_*. В следствии теоремы 6.1 фигурирует множество, которому соответствует теперь множество $\tilde{I}_*$. Поэтому, чтобы множество $\tilde{I}_*$ содержало n индексов, необходимо, чтобы мощность множества I_* была равна $n+1$.

Отметим, теперь, что ситуация, описанная в следствии теоремы 6.1, достаточно типична в минимаксных задачах, в частности, в задачах теории приближений. (Более подробно с этим вопросом читатель может ознакомиться в [6,7].) Поэтому желательно показать, что алгоритм метода линеаризации с $A_k = I_n$ обеспечивает в этой ситуации единичный шаг в окрестности точки x_*, а тем самым и сверхлинейную сходимость. То, что это действительно так, показано в работе [4], которой мы и последуем.

Л е м м а 7.5. *Пусть выполнены предположения* 1 – 3 *этого пункта и* $|I_*| = n+1$. *Тогда существуют такие числа* $r > 0$, $\eta > 0$, *что*

$$\min_{\|p\|=1} \max_{i \in I_*} (f_i'(x), p) \geqslant \eta$$

при всех x *таких, что* $\|x - x_*\| \leqslant r$.

Д о к а з а т е л ь с т в о. Покажем, что

$$\max_{i \in I_*} (f_i'(x_*), p) > 0$$

при любом p, $\|p\| = 1$. Действительно, если это не так, то найдется такой вектор p, что

$$(f_i'(x_*), p) \leqslant 0, \quad i \in I_*. \tag{7.31}$$

Но из (7.6) и предположений 1 – 3 следует, что

$$\sum_{i \in I_*} u_*^i f_i'(x_*) = 0, \quad u_*^i > 0, \quad i \in I_*.$$

Умножая первое равенство скалярно на p, получаем

$$\sum_{i \in I_*} u_*^i (f_i'(x_*), p) = 0. \tag{7.32}$$

Так как в силу (7.31) здесь все слагаемые неположительны, то (7.32)

возможно лишь, если

$$(f_i'(x_*),\ p) = 0,\quad i \in I_*,$$

или

$$(f_i'(x_*) - f_j'(x_*),\ p) = 0,\quad i \in \tilde{I}_*.$$

Отсюда получается, что вектор p ортогонален n линейно независимым векторам и поэтому $p = 0$, в противоречии с тем, что $\| p \| = 1$.

Заметим теперь, что функции

$$\varphi_0(x, p) = \max_{i \in I_*} (f_i'(x),\ p),\quad \varphi_1(x) = \min_{\| p \| = 1} \varphi_0(x, p)$$

непрерывны по x и p. Этот факт установлен, например, в [6, 23]. Поэтому из доказанного следует, что

$$\varphi_0(x_*, p) > 0,\quad \| p \| = 1,\quad \varphi_1(x_*) > 0$$

и существует такая окрестность точки x_*, что

$$\varphi_1(x) \geqslant \tfrac{1}{2}\, \varphi_1(x_*).$$

Полагая r равным радиусу этой окрестности и $\eta = \frac{1}{2}\, \varphi_1(x_*)$, получаем требуемый результат.

Заметим теперь, что если точка x достаточно близка к x_*, то и $p(x)$ достаточно близко к $p(x_*) = 0$. Поэтому точка $x + p(x)$ также близка к x_*. Отсюда нетрудно сделать вывод, что в некоторой окрестности точки x_* всегда найдется такой индекс $i \in I_*$ (он может зависеть от x), что

$$F(x + p(x)) = f_i(x + p(x)).$$

Выберем эту окрестность настолько малой, чтобы в ней выполнялось неравенство

$$\max_{i \in I_*} \| f_i'(x + \theta_i p(x)) - f_i'(x) \| \leqslant (1 - \epsilon)\, \eta$$

при любых θ_i, $0 \leqslant \theta_i \leqslant 1$, где $0 < \epsilon < 1$. Учитывая теперь (7.28), получаем

$$\begin{aligned} F(x) - \beta(x) &\geqslant \max_{i \in I_*} (f_i(x) - \beta(x)) = \\ &= \max_{i \in I_*} (f_i'(x),\ -p(x)) \geqslant \eta \| p(x) \|, \\ F(x + p(x)) - \beta(x) &= \max_{i \in I_*} [f_i(x + p(x)) - \\ &- f_i(x) - (f_i'(x),\ p(x))] = \max_{i \in I_*} (f_i'(x + \theta_i p(x)) - f_i'(x),\ p(x)) \leqslant \\ &\leqslant (1 - \epsilon)\, \eta \| p(x) \| \leqslant (1 - \epsilon)\, [F(x) - \beta(x)]. \end{aligned} \tag{7.33}$$

Так как из (7.14) вытекает, что

$$\| p(x) \|^2 \leqslant F(x) - \beta(x),$$

ибо $A = I_n$, то

$$\begin{aligned} &F(x) - F(x + p(x)) = F(x) - \beta(x) - \\ &- [F(x + p(x)) - \beta(x)] \geqslant \epsilon\, [F(x) - \beta(x)] \geqslant \epsilon \| p(x) \|^2. \end{aligned}$$

Итак, доказана следующая лемма.

Лемма 7.6. *В предположениях леммы 7.5 существует такая окрестность точки x_*, в которой выполнено неравенство*

$$F(x + p(x)) \leqslant F(x) - \epsilon \| p(x) \|^2 .$$

Теперь можно установить основной результат.

Теорема 7.4. *Пусть выполнены предположения 1 – 3 и $|I_*| = n + 1$. Тогда алгоритм метода линеаризации с $A_k = I_n$ сходится к точке минимума x_* функции $F(x)$ сверхлинейно, т.е. быстрее любой геометрической прогрессии.*

Доказательство. Так как точка x_* – единственная точка, в которой выполняются необходимые условия минимума функции $F(x)$, то согласно теорема 7.2 построенная алгоритмом последовательность может иметь единственной предельной точкой точку x_* и поэтому $x_k \to x_*$. Согласно алгоритму шаг α_k выбирается путем деления пополам единицы, исходя из неравенства (7.25), которое теперь приобретает вид

$$F(x_k + \alpha_k p_k) \leqslant F(x_k) - \alpha_k \epsilon \| p_k \|^2 ,$$

$$p_k = p(x_k).$$

Но лемма 7.6 показывает, что это неравенство при достаточно больших k, когда x_k близко к x_*, будет выполняться при $\alpha_k = 1$, так что при больших k справедлива формула

$$x_{k+1} = x_k + p(x_k).$$

Применение следствия теоремы 7.3 завершает доказательство.

5. Ускорение сходимости в выпуклом случае. При исследовании скорости сходимости решающую роль играла лемма 7.4, задававшая систему уравнений, которой удовлетворяют $p(x)$ и $u(x)$ в окрестности точки минимума $F(x)$. При этом вектор $p(x)$ соответствовал матрице $A = I_n$. Нетрудно, однако, видеть, что если взять произвольную положительно определенную матрицу A, то $p_A(x)$ будет в окрестности x_* удовлетворять системе, аналогичной (7.27) – (7.29):

$$Ap_A(x) + \sum_{i \in I_*} u_A^i(x) (f_i'(x))^* = 0,$$

$$(f_i'(x),\ p_A(x)) + f_i(x) = \beta_A(x),\quad x \in I_*, \tag{7.34}$$

$$\sum_{i \in I_*} u_A^i(x) = 1.$$

Эта система непосредственно получается из (7.10), (7.11), если только выполнено предположение 3 п. 4. При этом, как и в лемме 7.4, $p_A(x)$ и $u_A^i(x)$ непрерывно дифференцируемы в окрестности точки x_*. Действительно, система (7.34) линейна относительно p_A, u_A и β_A, причем матрица этой системы в точке $x = x_*$ невырождена. (Мы предоставляем читателю убедиться в этом, опираясь на выполнение предположения 3.) Поэтому $p_A(x)$, $u_A(x)$ и $\beta_A(x)$ однозначно выражаются через параметры задачи (7.34), и если $f_i(x)$ дважды непрерывно дифференцируемы, то p_A, u_A и β_A непрерывно дифференцируемы.

Пусть $u^i_A(x) = 0$, $i \notin I_*$, $f'(x)$ – матрица со строчками $f'_i(x)$, $i \in I_*$. Дифференцируя (7.34) по x при $x = x_*$ с учетом того, что $p_A(x_*) = 0$, $u_A(x_*) = u_*$, $\beta_A(x_*) = F(x_*)$, получаем

$$Ap'_A(x_*) + L''_{xx}(x_*, u_*) + \sum_{i \in I_*} (f'_i(x_*))^* (u^i_A(x_*))' = 0,$$

$$f'(x_*) p'_A(x_*) + f'(x_*) = 1 \cdot \beta'_A(x_*), \qquad (7.35)$$

$$\sum_{i \in I_*} (u^i_A(x_*))' = 0,$$

где 1 – вектор размерности $|I_*|$, все компоненты которого равны единице.

Так как матрица системы (7.34) была невырождена, то согласно теореме о неявных функциях [23], система (7.35) для производных однозначно разрешима. Пусть $A = L''_{xx}(x_*, u_*)$ и выполнено предположение 4 п. 4. При таком выборе A

$$p'_A(x_*) = -I_n, \quad (u^i_A(x_*))' = 0, \quad \beta'_A(x_*) = 0 \qquad (7.36)$$

является решением системы (7.35).

Таким образом, установлена справедливость следующего результата.

Теорема 7.5. *Пусть выполнены предположения* 1 – 4 *пункта* 4. *Тогда при* $A = L''_{xx}(x_*, u_*)$ *решение вспомогательной задачи* (7.8) *дифференцируемо в некоторой окрестности точки* x_* *и имеют место формулы* (7.36).

Наличие формул (7.36) позволяет сформулировать следующий алгоритм, от которого следует ожидать более высокой скорости сходимости.

Алгоритм. Пусть x_0, $\epsilon \in (0, 1)$, $\gamma \in (0, 1)$, $\delta > 0$ выбраны. Положим $C_0 = +\infty$.

Общий шаг: если точки x_k, u_{k-1} и число C_k уже построены, то поступаем следующим образом:

1. Полагаем $A_k = L''_{xx}(x_k, u_{k-1})$, $k \geqslant 1$, $A_0 = I_n$.

2. Решаем задачи (7.8) с $x = x_k$, $A = A_k$ и вычисляем $p_k = p_{A_k}(x_k)$, $u_k = u_{A_k}(x_k)$.

3. Если $\| p_k \| \leqslant C_k$, то полагаем

$$\bar{x} = x_k + p_k, \quad A = L''_{xx}(\bar{x}, u_k)$$

и вычисляем вектор $p_A(\bar{x})$.

Если $\| p_A(\bar{x}) \| \leqslant \gamma \| p_k \|$, то $x_{k+1} = \bar{x}$, $C_{k+1} = \gamma \| p_k \|$.

Если $\| p_A(\bar{x}) \| > \gamma \| p_k \|$, то полагаем $C_{k+1} = \gamma \| p_k \|$ и переходим к 5.

4. Если $\| p_k \| > C_k$, то $C_{k+1} = C_k$ и переходим к 5.

5. Начиная с $\alpha = 1$, делим пополам эту величину до выполнения неравенства

$$F(x_k + \alpha_k p_k) \leqslant F(x_k) - \epsilon \alpha_k (p_k, A_k p_k).$$

Полагаем

$$x_{k+1} = x_k + \alpha_k p_k$$

и возвращаемся к 1.

Теорема 7.6. *Пусть $f_i(x)$, $i \in I$, – дважды непрерывно дифференцируемые выпуклые функции и существуют такие константы $M \geqslant m > 0$, что*

$$m \| p \|^2 \leqslant (f_i''(x) p, p) \leqslant M \| p \|^2, \quad i \in I. \tag{7.37}$$

Пусть x_* – единственная точка минимума функции $F(x)$ и векторы $f_i'(x_*) - f_j'(x_*)$, $i \in I_* \setminus \{j\}$, где j – какой-либо индекс из I_*, линейно независимы, а $u_*^i > 0$ при $i \in I_*$. Тогда построенный алгоритм порождает последовательность x_k, сходящуюся к x_*, и

$$\| x_{k+1} - x_* \| \leqslant q_k \| x_k - x_* \|, \quad q_k \to 0.$$

Доказательство этой теоремы в основном повторяет доказательство теорем 6.2 и 6.4.

Сделаем сразу несколько замечаний. В силу условия (7.37) функция $F(x)$ сильно выпукла и достигает своего минимума в единственной точке x_*. Если $u^i \geqslant 0$ и сумма u^i равна единице, то из (7.37) следует, что

$$m \| p \|^2 \leqslant (L_{xx}''(x, u) p, p) \leqslant M \| p \|^2.$$

Поэтому все матрицы $A_k = L_{xx}''(x_k, u_{k-1})$ удовлетворяют условиям теоремы 7.2. Если в рассматриваемом алгоритме не было бы шагов 3, 4, то он совпадал бы просто с методом линеаризации и по теореме 7.2 давал бы сходящуюся к x_* последовательность.

Рассуждая теперь точно так же, как при доказательстве теоремы 6.4, убедимся, что существует такая бесконечная последовательность индексов J^0, что

$$x_k \to x_*, \quad u_{k-1} \to u_*, \quad A_k = L_{xx}''(x_k, u_{k-1}) \to L_{xx}''(x_*, u_*) = A.$$

Пусть $p_0(x)$, $u_0^i(x)$ – решение задачи (7.8) при $A = L_{xx}''(x_*, u_*)$, которое, как доказано выше, удовлетворяет системе (7.34) с соответствующей матрицей A. С другой стороны, p_k, u_k удовлетворяют системе (7.34) при $A = A_k$, $x = x_k$ при достаточно больших $k \in J^0$. Вычисляя эти системы одну из другой, получаем

$$A_k(p_k - p_0(x_k)) + \sum_{i \in I_*} (u_k^i - u_0^i(x_k)) (f_i'(x_*))^* = -(A_k - A) p_0(x_k),$$

$$f'(x_k)(p_k - p_0(x_k)) - (\beta_k - \beta_0(x_k)) = 0,$$

$$\sum_{i \in I_*} (u_k^i - u_0^i(x_k)) = 0.$$

Матрица этой системы уравнений относительно разностей – такая же, как и у системы (7.34), т.е. невырождена, а правая часть равна $(A_k - A) p_0(x_k)$. Поэтому, разрешая эту систему относительно разностей $p_k - p_0(x_k)$, $u_k - u_0(x_k)$, $\beta_k - \beta_0(x_k)$, получим

$$\| p_k - p_0(x_k) \| \leqslant C \| A_k - A \| \| p_0(x_k) \|,$$

$$\| u_k - u_0(x_k) \| \leqslant C \| A_k - A \| \| p_0(x_k) \|,$$

$$| \beta_k - \beta_0(x_k) | \leqslant C \| A_k - A \| \| p_0(x_k) \|.$$

Вспомним теперь, что согласно соотношениям (7.36)

$$p_0(x) = -(x - x_*) + \|x - x_*\| \omega_0(x, x_*),$$
$$u_0(x) = u_* + \|x - x_*\| \omega_1(x, x_*),$$
$$\beta_0(x) = F(x_*) + \|x - x_*\| \omega_2(x, x_*), \tag{7.38}$$
$$\lim_{x \to x_*} \omega_j(x, x_*) = 0, \quad j = 0, 1, 2.$$

Значит, если $k \in J^0$, то

$$\bar{x} = x_k + p_k = x_k + p_0(x_k) + (p_k - p_0(x_k)) =$$
$$= x_* + \|x_k - x_*\| \omega_0(x_k, x_*) + (p_k - p_0(x_k)),$$
$$\|\bar{x} - x_*\| \leqslant \|x_k - x_*\| \omega_0(x_k, x_*) + C\|A_k - A\| \|p_0(x_k)\| \leqslant \tag{7.39}$$
$$\leqslant [\omega_0(x_k, x_*) + C\|A_k - A\|] C_1 \|x_k - x_*\|,$$

где мы воспользовались тем, что $p_0(x)$ и $\|x - x_*\|$ есть величины одного порядка малости. По этой же причине из (7.38) и (7.39) вытекает, что

$$\|p_0(\bar{x})\| \leqslant [\omega_0(x_k, x_*) + C\|A_k - A\|] C_2 \|p_0(x_k)\|.$$

Учитывая теперь оценку отклонения p_k от $p_0(x_k)$ и то, что в силу приведенных оценок

$$\|u_k - u_*\| \leqslant \|x_k - x_*\| \omega_1(x_k, x_*) + C\|A_k - A\| \|p_0(x_k)\| \leqslant$$
$$\leqslant [\omega_1(x_k, x_*) + C_3\|A_k - A\|] C_4 \|x_k - x_*\|,$$

можно будет убедиться, что

$$\|p_{L''_{xx}(\bar{x}, u_k)}(\bar{x})\| \leqslant g_k \|p_k\|,$$

где $g_k \to 0$, $k \in J^0$.

Поэтому $g_k < \gamma$ при больших k и шаг 3 алгоритма успешно проработает. Итак, при больших $k \in J^0$ шаг алгоритма будет успешным и $x_{k+1} = \bar{x}$ и u_k будут существенно ближе к x_* и u_*. Повторяя приведенные выше рассуждения, убедимся, что и в точке x_{k+1} шаг 3 алгоритма будет успешным. Отсюда следует, что, начиная с некоторого момента, алгоритм будет работать по формуле $x_{k+1} = x_k + p_k$ и будет справедлива оценка (7.39):

$$\|x_{k+1} - x_*\| \leqslant q_k \|x_k - x_*\|, \quad q_k = C_1[\omega_0(x_k, x_*) + C\|A_k - A\|].$$

Но $\omega_0(x_k, x_*) \to 0$, $A_k \to A$, и поэтому $q_k \to 0$, что завершает доказательство.

§ 8. ДВОЙСТВЕННЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ

В последние годы широкую популярность приобрел метод модифицированных функций Лагранжа. Этому методу посвящено значительное число работ [21, 34]. В этом параграфе предлагается и исследуется метод решения задачи выпуклого программирования, весьма близкий к методу модифицированных функций Лагранжа. Однако его свойства изучаются способами, отличными от приемов, использованных в указанных выше работах. При этом получаемый метод обладает всеми преимуществами, свойственными методу модифицированных функций Лагранжа, а причины

хорошей сходимости метода и способы ускорения этой сходимости становятся совершенно ясными.

Пусть $f_i(x), i = 0, 1, \ldots, m$, – выпуклые функции в $\mathbf{R}^n$, области определения которых содержат выпуклое множество $X \subseteq \mathbf{R}^n$. Задача

$$\min_x \left\{ f_0(x):\ f_i(x) \leqslant 0, i = 1, \ldots, m, x \in X \right\} \tag{8.1}$$

является задачей выпуклого программирования. Рассмотрим в $\mathbf{R}^{m+1}$ множество

$$M = \left\{ z \in \mathbf{R}^{m+1}:\ z = f(x) + v, x \in X, v \geqslant 0 \right\} = f(X) + \mathbf{R}_+^{m+1}, \tag{8.2}$$

где вектор z имеет компоненты $z^0, z^1, \ldots, z^m$, $f(x)$ – вектор с компонентами $f_i(x)$, $i = 0, 1, \ldots, m$, $\mathbf{R}_+^{m+1}$, как обычно, означает положительный ортант в $\mathbf{R}^{m+1}$.

Нетрудно видеть, что M – выпуклое множество, а задача выпуклого программирования (8.1) эквивалентна задаче

$$\min \left\{ \lambda:\ \lambda e \in M \right\}, \tag{8.3}$$

где e – вектор с компонентами $e^0 = 1, e^i = 0, i = 1, \ldots, m$. Поэтому в ближайших пунктах все внимание будет сосредоточено на решении задачи (8.3). Обозначим через λ_* ее решение, так что $\lambda e \notin M$, $\lambda < \lambda_*$, $\lambda_* e \in M$.

Отвлечемся сейчас от конкретного вида (8.2) множества M. Будем предполагать, что M – замкнутое выпуклое множество, а вектор e – произвольный единичный вектор.

1. Двойственный алгоритм. Сформулируем алгоритм решения задачи (8.3). Пусть

$$\lambda_0 < \lambda_*, y_0 = \lambda_0 e, z_0 = \arg\min \left\{ \| z - y_0 \|:\ z \in M \right\}.$$

Положим также

$$r_0 = \| z_0 - y_0 \|,$$

$$\eta_0 = (z_0 - y_0)/r_0, \| \eta_0 \| = 1,$$

$$\lambda_1 = (\eta_0, z_0)/(\eta_0, e).$$

Известно (см. п. 1 § 2), что выполняется неравенство

$$(z, \eta_0) \geqslant (z_0, \eta_0), z \in M.$$

Определим теперь общий шаг алгоритма. Пусть $\lambda_k \leqslant \lambda_*$, η_{k-1}, z_{k-1} уже построены и удовлетворяют соотношениям

$$\begin{aligned} &\lambda_k (\eta_{k-1}, e) = (\eta_{k-1}, z_{k-1}) \leqslant (\eta_{k-1}, z), \\ &z \in M, z_{k-1} \in M, \| \eta_{k-1} \| = 1. \end{aligned} \tag{8.4}$$

Положим

$$\begin{aligned} &y_k = \lambda_k e - \alpha \eta_{k-1}, \alpha \geqslant 0, \qquad z_k = \arg\min_z \left\{ \| z - y_k \|:\ z \in M \right\}, \\ &r_k = \| z_k - y_k \|, \qquad \eta_k = (z_k - y_k)/r_k, \lambda_{k+1} = (\eta_k, z_k)/(\eta_k, e). \end{aligned} \tag{8.5}$$

Из того, что z_k есть точка минимума $\|z - y_k\|$ на M, следует, что выполнено соотношение

$$(\eta_k, z) \geqslant (\eta_k, z_k) = \lambda_{k+1}(\eta_k, e), z \in M, \tag{8.6}$$

так что λ_{k+1}, η_k, z_k удовлетворяют соотношению, аналогичному (8.4).

Заметим прежде всего, что в силу (8.4) $(\eta_{k-1}, z_k - \lambda_k e) \geqslant 0$, поэтому

$$r_k^2 = \|z_k - y_k\|^2 = \|z_k - \lambda_k e\|^2 + 2(z_k - \lambda_k e, \lambda_k e - y_k) +$$

$$+ \|\lambda_k e - y_k\|^2 = \|z_k - \lambda_k e\|^2 + 2\alpha(z_k - \lambda_k e, \eta_{k-1}) + \alpha^2 \geqslant \alpha^2 .$$

Таким образом, всегда $r_k \geqslant \alpha$. При этом, если $r_k = \alpha$, то $\|z_k - \lambda_k e\| = 0$, $z_k = \lambda_k e \in M$, и так как $\lambda_k \leqslant \lambda_*$, то $\lambda_k = \lambda_*$ есть решение задачи (8.3). Обратно, если $\lambda_k = \lambda_*$, то $\lambda_k e \in M$ и в силу выбора z_k

$$r_k^2 = \|z_k - \lambda_k e\|^2 + 2\alpha(z_k - \lambda_k e, \eta_{k-1}) + \alpha^2 \leqslant \|\lambda_k e - y_k\|^2 = \alpha^2 ,$$

т.е. $r_k \leqslant \alpha$.

Таким образом, равенство $r_k = \alpha$ является признаком того, что задача (8.3) решена.

Для обоснования возможности работы алгоритма необходимо показать, что если $\lambda_k < \lambda_*$, то $\lambda_{k+1} \leqslant \lambda_*$, и что (η_k, e) строго положительно.

Пусть $\lambda_k < \lambda_*$. Из (8.4) следует, что

$$(\eta_{k-1}, z_k) \geqslant (\eta_{k-1}, \lambda_k e) = (\eta_{k-1}, y_k) + \alpha(\eta_{k-1}, \eta_{k-1}) ,$$

или, так как $\|\eta_{k-1}\| = 1$,

$$(\eta_{k-1}, z_k - y_k) = r_k(\eta_{k-1}, \eta_k) \geqslant \alpha . \tag{8.7}$$

Отсюда видно, что $(\eta_{k-1}, \eta_k) \geqslant 0$, $r_k \geqslant \alpha$. Далее, из (8.6) следует, что

$$(\eta_k, z) \geqslant (\eta_k, z_k - y_k) + (\eta_k, y_k) = r_k + \lambda_k(\eta_k, e) - \alpha(\eta_k, \eta_{k-1}) .$$

Подставляя $z = \lambda_* e$, получаем

$$(\lambda_* - \lambda_k)(\eta_k, e) \geqslant r_k - \alpha(\eta_k, \eta_{k-1}) \geqslant r_k - \alpha \geqslant 0 . \tag{8.8}$$

Поэтому

$$(\eta_k, e) > 0 ,$$

ибо из того, что $(\eta_k, e) = 0$, следовало бы, что $r_k = \alpha$, $\lambda_k = \lambda_*$. Подставляя теперь $z = \lambda_* e$ в (8.6), получаем

$$(\lambda_* - \lambda_{k+1})(\eta_k, e) \geqslant 0 ,$$

т.е. $\lambda_{k+1} \leqslant \lambda_*$.

Сформулируем полученный результат.

Теорема 8.1. *Если $\alpha \geqslant 0$, $\lambda_0 < \lambda_*$, то алгоритм* (8.5) *для* $k = 1, 2, \ldots$ *строит последовательность λ_k такую, что $\lambda_k \leqslant \lambda_*$, $r_k \geqslant \alpha$. При этом выполнение на некотором шаге условия $r_k = \alpha$ означает, что $\lambda_k = \lambda_*$, $z_k = \lambda_* e$, т.е. условие $r_k = \alpha$ является признаком останова алгоритма.*

Из формул (8.5) следует, что

$$\lambda_{k+1} = \frac{(\eta_k, z_k - y_k) + \lambda_k(\eta_k, e) - \alpha(\eta_k, \eta_{k-1})}{(\eta_k, e)},$$

$$\lambda_{k+1} = \lambda_k + \frac{r_k - \alpha(\eta_k, \eta_{k-1})}{(\eta_k, e)}. \tag{8.9}$$

Так как $\|\eta_k\| = \|\eta_{k-1}\| = 1$, то

$$\|\eta_k - \eta_{k-1}\|^2 = 2(1 - (\eta_k, \eta_{k-1})).$$

Поэтому (8.9) можно переписать в виде

$$\lambda_{k+1} = \lambda_k + \frac{r_k - \alpha}{(\eta_k, e)} + \alpha \frac{\|\eta_k - \eta_{k-1}\|^2}{2(\eta_k, e)}. \tag{8.10}$$

Отсюда следует, что λ_k – возрастающая последовательность.

Теорема 8.2. *Построенная алгоритмом последовательность* λ_k, *монотонно возрастая, стремится к* λ_*. *При этом* $z_k \to \lambda_* e$, *а любая предельная точка* η *последовательности* η_k *есть опорный вектор к* M *в точке* $\lambda_* e$, *т.е.*

$$(z - \lambda_* e, \eta) \geqslant 0, z \in M.$$

Доказательство. Как показано выше, если алгоритм остановится, т.е. $r_k = \alpha$ при некотором k, то $\lambda_k = \lambda_*$, $z_k = \lambda_* e$. Поэтому рассмотрим случай когда $r_k > \alpha$ при всех k и процесс бесконечен.

Из (8.10) получаем, что $\lambda_{k+1} - \lambda_k \geqslant r_k - \alpha > 0$, так как $0 < (\eta_k, e) \leqslant 1$. Из того, что λ_k – монотонно возрастающая ограниченная сверху последовательность, следует, что $\lambda_{k+1} - \lambda_k \to 0$ и поэтому $r_k \to \alpha$. Как было показано выше,

$$r_k^2 = \|z_k - \lambda_k e\|^2 + 2\alpha(z_k - \lambda_k e, \eta_{k-1}) + \alpha^2,$$
$$(z_k - \lambda_k e, \eta_{k-1}) \geqslant 0,$$

т.е. $r_k^2 - \alpha^2 \geqslant \|z_k - \lambda_k e\|^2$, поэтому $\|z_k - \lambda_k e\| \to 0$.

Если $\underline{\lambda}$ – предел ограниченной сверху возрастающей последовательности λ_k, то $z_k \to \underline{\lambda} e$, $\underline{\lambda} \leqslant \lambda_*$. Но $z_k \in M$ и M замкнуто. Поэтому $\underline{\lambda} e \in M$, так что $\underline{\lambda} = \lambda_*$.

Последнее утверждение теоремы сразу получается путем предельного перехода в (8.6).

Замечание. Если M не замкнуто, то $\underline{\lambda} e \in \overline{M}$, где $\overline{M}$ – замыкание M, а $\underline{\lambda} = \min\{\lambda: \lambda e \in \overline{M}\}$.

В этом нетрудно убедиться.

Теорема 8.3. *Если* M – *многогранное множество, то алгоритм сходится за конечное число шагов.*

Доказательство. Пусть M задано конечной системой линейных неравенств

$$(a_i, z) - b_i \leqslant 0, \quad i \in I. \tag{8.11}$$

Для $z \in M$ обозначим $I(z) = \{i \in I:\ (a_i, z) = b_i\}$. Как показано выше, $z_k \to \lambda_* e$. Поэтому $I(z_k) \subseteq I(\lambda_* e)$ для достаточно больших k. С другой стороны, z_k есть точка минимума функции $\frac{1}{2}\|z - y_k\|^2$ на множестве M, заданном неравенствами (8.11), и поэтому в силу необходимости условий минимума

$$z_k - y_k = -\sum_{i \in I(z_k)} \lambda_i a_i,\ \lambda_i \geqslant 0,$$

или

$$\eta_k = -\sum_{i \in I(z_k)} \gamma_i a_i,\ \gamma_i = \lambda_i / r_k \geqslant 0.$$

Так как $I(z_k) \subseteq I(\lambda_* e)$, то

$$(a_i, z_k) = (a_i, \lambda_* e) = b_i,\ i \in I(z_k),$$

и поэтому

$$(\eta_k, \lambda_* e) = (\eta_k, z_k),$$

т.е.

$$\lambda_* = (\eta_k, z_k)/(\eta_k, e) = \lambda_{k+1},$$

откуда и следует конечная сходимость алгоритма.

2. Оценка скорости сходимости. Как обычно, более точная оценка скорости сходимости заставляет сделать некоторые добавочные предположения.

Так как множества $\{\lambda e:\ \lambda < \lambda_*\}$ и M не пересекаются, то их можно отделить, т.е. найдется такой вектор e_1, что

$$(\lambda e, e_1) \leqslant (z, e_1),\ \lambda < \lambda_*,\ z \in M.$$

Отсюда следует, что (e, e_1) неотрицательно и что

$$(\lambda_* e, e_1) \leqslant (z, e_1),\ z \in M. \tag{8.12}$$

Будем предполагать, что существует вектор e_1, удовлетворяющий (8.12) и строгому неравенству

$$(e, e_1) > 0. \tag{8.13}$$

Подставим в (8.12) вместо z вектор

$$z_k = y_k + r_k \eta_k = \lambda_k e + r_k \eta_k - \alpha \eta_{k-1}.$$

Получаем

$$r_k (e_1, \eta_k) \geqslant (\lambda_* - \lambda_k)(e, e_1) + \alpha (e_1, \eta_{k-1}), \tag{8.14}$$

или, более грубо,

$$r_k \geqslant \delta_k (e, e_1) + \alpha (e_1, \eta_{k-1}), \tag{8.15}$$

$$\delta_k = \lambda_* - \lambda_k.$$

Отбросив в выражении (8.10) для λ_{k+1} последний член и заменив r_k на (8.15), получаем

$$\lambda_{k+1} \geqslant \lambda_k + \delta_k \frac{(e, e_1)}{(e, \eta_k)} - \alpha \frac{1 - (e_1, \eta_{k-1})}{(e, \eta_k)},$$

или

$$\delta_{k+1} \leqslant \left(1 - \frac{(e, e_1)}{(e, \eta_k)}\right)\delta_k + \alpha \frac{\|e_1 - \eta_{k-1}\|^2}{2(e, \eta_k)}. \qquad (8.16)$$

Формула (8.16) позволяет сделать сразу некоторые выводы.

Т е о р е м а 8.4. *Пусть* $\alpha = 0$, *вектор* e_1, *удовлетворяющий соотношению* (8.12), *единственный. Пусть также выполнено неравенство* (8.13). *Тогда* δ_k *стремится к нулю сверхлинейно.*

В самом деле, в условиях теоремы 8.4 из теоремы 8.2 вытекает, что $\eta_k \to e_1$. Поэтому $\delta_{k+1} \leqslant \gamma_k \delta_k$, $\gamma_k = 1 - (e, e_1)/(e, \eta_k) \to 0$.

Изучим более детально случай $\alpha > 0$. Предположим, что множество M – достаточно гладкое в окрестности точки $\lambda_* e$, т.е. может быть задано в некоторой окрестности этой точки неравенством $\varphi(z) \leqslant 0$, где φ – дважды непрерывно дифференцируемая выпуклая функция. Очевидно, что в этом случае

$$e_1 = -\varphi'(\lambda_* e) \| \varphi'(\lambda_* e) \|^{-1},$$

где $\varphi'(z)$ – градиент функции φ.

Будем представлять теперь каждый вектор z в виде

$$z = \beta e + w, \quad (e, w) = 0.$$

Нетрудно видеть, что

$$\beta = (z, e), \quad w = z - (z, e)\, e.$$

Обозначим через $P = I - ee^*$, где e^* – транспонированный вектор e (т.е. вектор-строка), матрицу проектирования на подпространство $(e, z) = 0$. Ясно, что

$$z = (I - P)\, z + Pz, \quad \beta e = (I - P)\, z, \quad w = Pz.$$

Если условие (8.13) выполнено, то нетрудно показать, что в окрестности точки $\lambda_* e$ множество M может быть описано неравенством

$$\beta \geqslant \omega(w), \qquad (8.17)$$

где ω – дважды непрерывно дифференцируемая функция. При этом, так как $\beta = \lambda_*$, $w = 0$ для $z = \lambda_* e$, то

$$\lambda_* = \omega(0).$$

Пусть z – граничная точка M, т.е. $\beta = \omega(w)$. Обозначим через $\eta(z)$ единичную нормаль к M в z. Легко видеть, что

$$\eta(z) = \frac{1}{\sqrt{1 + \| \omega'(w) \|^2}} \begin{bmatrix} 1 \\ -\omega'(w) \end{bmatrix},$$

где $\omega'(w)$ – производная ω по w, причем размерность w на единицу мень-

шс размерности z. Кроме того, $\eta(0) = e_1$. Найдем точку пересечения касательной к M в точке z гиперплоскости с прямой λe. В координатах (β, w) соответствующее уравнение имеет вид

$$-(\lambda - \beta) + (\omega'(w), -w) = 0,$$

$$\lambda(z) = \beta - (\omega'(w), w) = \omega(w) - (\omega'(w), w).$$

Поэтому разложение $\omega(w)$ и $\omega'(w)$ по w в ряд Тейлора в окрестности точки $w = 0$ до членов второго порядка даст

$$\lambda(z) = \omega(0) - (\omega''(0)\, w, w)/2 + o(\|w\|^2),$$

$$\delta(z) = \lambda_* - \lambda(z) = (\omega''(0)\ w, w)/2 + o(\|w\|^2).$$

Пусть теперь

$$\mu = \min_{w \neq 0} \frac{(\omega''(0)\, w, w)}{\|w\|^2} > 0.$$

Тогда

$$\delta(z) \geqslant (\mu/4)\|w\|^2 \tag{8.18}$$

для точек z границы M при достаточно малых w. Далее, если w задано, то соответствующая ему граничная точка z множества M имеет координату $\beta = \omega(w)$. Поэтому определенная в граничных точках z нормаль $\eta(z)$ есть функция w, и так как $\omega(w)$ дважды непрерывно дифференцируема, то η как функция w удовлетворяет в окрестности $w = 0$ условию Липшица с константой L.

С учетом (8.18) получаем

$$\frac{\|e_1 - \eta(z)\|^2}{2} = \frac{\|\eta(0) - \eta(z)\|^2}{2} \leqslant \frac{L^2}{2}\|w\|^2 \leqslant \frac{2L^2}{\mu}\delta(z) \tag{8.19}$$

Воспользуемся теперь этой оценкой для преобразования оценки (8.16). Так как $\eta_{k-1} = \eta(z_{k-1})$, $\delta_{k-1} = \delta(z_{k-1})$, то

$$\delta_{k+1} \leqslant \left(1 - \frac{(e, e_1)}{(e, \eta_k)}\right)\delta_k + \frac{2L^2}{\mu}\,\alpha\,\frac{\delta_{k-1}}{(e, \eta_k)}. \tag{8.20}$$

Т е о р е м а 8.5. *Пусть в окрестности точки $\lambda_* e$ множество M может быть задано неравенством* (8.17), *где $\omega(w)$ – дважды непрерывно дифференцируемая выпуклая функция, и*

$$\mu = \min_{\substack{w \neq 0 \\ (e, w) = 0}} \frac{(\omega''(0)\, w, w)}{\|w\|^2} > 0.$$

Тогда скорость сходимости построенного алгоритма при больших k определяется формулой (8.20).

Поскольку последовательность δ_k – убывающая и при сделанных предположениях $\eta_k \to \eta(0) = e_1$, то из (8.20) следует, что

$$\overline{\lim_{k\to\infty}} \frac{\delta_{k+1}}{\delta_{k-1}} \leqslant \frac{2L^2}{\mu(e, e_1)} \alpha. \tag{8.21}$$

Отсюда видно, что уменьшение α ведет к ускорению процесса сходимости.

Покажем, что простым преобразованием можно добиться уменьшения константы L, что также приведет к ускорению сходимости. Сделаем преобразование координат, сводящееся к растяжению в K раз составляющей вектора z, ортогонального вектору e. Если P – оператор ортогонального проектирования на гиперплоскость, ортогональную e, то

$$P = I - ee^*, \; P^2 = P, \; P(I - P) = 0.$$

Будем в дальнейшем все величины в новых координатах обозначать тильдой. Преобразование координат теперь дается формулой

$$\tilde{z} = (I - P)z + KPz,$$

откуда нетрудно получить, что

$$z = (I - P)\tilde{z} + K^{-1}P\tilde{z} = (ee^* + K^{-1}P)\tilde{z}.$$

Кроме того,

$$\tilde{w} = P\tilde{z} = KPz = Kw.$$

Рассмотрим, как при этом преобразуется вектор нормали к множеству M в граничной точке этого множества. Напомним, что граничная точка z однозначно определяется своей координатой $w = Pz$, так как координата β вдоль оси e находится из уравнения $\beta = \omega(w)$.

Пусть η – единичная нормаль к M в точке z, т.е.

$$(\eta, z_1 - z) \geqslant 0, \; z_1 \in M, \; \|\eta\| = 1.$$

Подставляя

$$z_1 = (ee^* + K^{-1}P)\tilde{z}_1, \; z = (ee^* + K^{-1}P)\tilde{z},$$

получаем

$$(\eta, (ee^* + K^{-1}P)(\tilde{z}_1 - \tilde{z})) \geqslant 0, \; \tilde{z}_1 \in \tilde{M}, \tag{8.22}$$

где $\tilde{M}$ – множество M в новых координатах. Так как P – симметричная матрица, то (8.22) переписывается в виде

$$(\tilde{\eta}, \tilde{z}_1 - \tilde{z}) \geqslant 0, \; \tilde{z}_1 \in \tilde{M}, \tag{8.23}$$

где

$$\tilde{\eta} = \frac{(ee^* + K^{-1}P)\eta}{\|(ee^* + K^{-1}P)\eta\|}.$$

Так как векторы e и $P\eta$ ортогональны, то нетрудно посчитать, что

$$\|(ee^* + K^{-1}P)\eta\| = \sqrt{(e, \eta)^2 + K^{-2}(1 - (e, \eta)^2)}.$$

Учитывая эту формулу и (8.23), окончательно получаем, что вектор $\tilde{\eta}(\tilde{w})$

в новых координатах выражается через вектор $\eta(w)$ по формуле

$$\tilde{\eta}(\tilde{w}) = \frac{(e, \eta(w))e + K^{-1} P\eta(w)}{\sqrt{(e, \eta(w))^2 + K^{-2}(1 - (e, \eta(w))^2)}}. \qquad (8.24)$$

Так как $\tilde{e} = e, e_1 = \eta(0), (Pe_1, e) = 0$, то из (8.24) следует, что при $K > 1$

$$(\tilde{e}, \tilde{e}_1) = \frac{(e, e_1) + K^{-1}(Pe_1, e)}{\sqrt{(e, e_1)^2 + K^{-2}(1 - (e, e_1)^2)}} =$$

$$= \frac{(e, e_1)}{\sqrt{(e, e_1)^2 + K^{-2}(1 - (e, e_1)^2)}} > (e, e_1),$$

т.е.

$$(\tilde{e}, \tilde{e}_1) > (e, e_1). \qquad (8.24')$$

Таким образом, скалярное произведение в правой части (8.24′) при преобразовании координат возрастает.

С другой стороны, не очень сложный, но громоздкий анализ выражения (8.24) показывает, что в новых координатах константа Липшица $\tilde{L}$ для $\tilde{\eta}(\tilde{w})$ выражается через L по следующей формуле:

$$\tilde{L} = CL / K^2,$$

где C – некоторое число. Кроме того, так как неравенство

$$\beta \geqslant \omega(w),$$

задающее M в окрестности точки $\lambda_* e$, при преобразовании $\tilde{\beta} = \beta$, $\tilde{w} = Kw$ переходит в неравенство

$$\tilde{\beta} \geqslant \omega(\tilde{w}K) = \tilde{\omega}(\tilde{w}),$$

то

$$\tilde{\omega}''(0) = K^{-2}\omega''(0).$$

Поэтому

$$\tilde{\mu} = K^{-2}\mu.$$

В итоге в новых координатах формула (8.21) переписывается в виде

$$\overline{\lim_{k\to\infty}} \frac{\tilde{\delta}_{k+1}}{\tilde{\delta}_{k-1}} \leqslant \frac{2(\tilde{L})^2}{\tilde{\mu}(\tilde{e}, \tilde{e}_1)} \alpha \leqslant \frac{2C^2L^2}{\mu(e, e_1)} K^{-2}\alpha. \qquad (8.25)$$

Из этой формулы видно, что растяжение координат в гиперплоскости, ортогональной e, приводит при больших K к существенному ускорению сходимости.

3. Алгоритм для задачи выпуклого программирования. Возвратимся к исходной задаче

$$\min_x \{ f_0(x) : f_i(x) \leqslant 0, i = 1, \ldots, m, x \in X \} \qquad (8.26)$$

и рассмотрим, к чему сводится сформулированный алгоритм. Как уже

говорилось в начале параграфа, для этой задачи

$$M = \{ f(x) + v:\ x \in X,\ v \geqslant 0 \},$$

где $f(x) \in \mathbf{R}^{m+1}$ – вектор с компонентами $f_k(x)$, $i = 0, 1, \ldots, m$, v – вектор с компонентами v_i, $i = 0, \ldots, m$. Нетрудно показать, что если $f_i(x)$ – непрерывные выпуклые функции, множество X выпукло и замкнуто, а множество

$$X_c = \{ x \in X:\ f_0(x) \leqslant C,\ f_i(x) \leqslant 0, i = 1, \ldots, m \}$$

ограничено при некотором C, то M – замкнутое выпуклое множество и задача (8.26) имеет хотя бы одно решение x_*. При этом $\lambda_* = f_0(x_*)$,

$$e = \begin{bmatrix} 1 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \in \mathbf{R}^{m+1},\quad z_* = \lambda_* e = \begin{bmatrix} f_0(x_*) \\ 0 \\ \cdots \\ 0 \end{bmatrix} \in \mathbf{R}^{m+1}.$$

Пусть существует вектор Куна – Таккера $u \in \mathbf{R}^m$, т.е.

$$u \geqslant 0,\ u^i f_i(x_*) = 0,\ i = 1, \ldots, m,$$

$$f_0(x_*) + \sum_{i=1}^{m} u^i f_i(x_*) \leqslant f_0(x) + \sum_{i=1}^{m} u^i f_i(x),\ x \in X.$$

Нетрудно видеть, что в этом случае вектор

$$e_1 = \begin{bmatrix} 1/\sqrt{1 + \|u\|^2} \\ u^1/\sqrt{1 + \|u\|^2} \\ \cdots \\ u^m/\sqrt{1 + \|u\|^2} \end{bmatrix} \in \mathbf{R}^{m+1}$$

является единичной нормалью к M в точке z_*.

Основная процедура приведенного в первом пункте алгоритма состоит в нахождении точки $z_k \in M$, ближайшей к точке y_k. Учитывая специальную структуру множества M, получаем

$$r_k^2 = \min_{z \in M} \sum_{i=0}^{m} (z^i - y_k^i)^2 = \min_{\substack{x \in X \\ v \geqslant 0}} \sum_{i=0}^{m} (f_i(x) + v^i - y_k^i)^2 =$$

$$= \min_{x \in X} \sum_{i=0}^{m} (f_i(x) - y_k^i)_+^2, \qquad (8.27)$$

где, как обычно, $t_+ = \max(0, t)$. Если минимум в (8.27) достигается в точке x_k, то соответствующая точка z_k имеет вид

$$z_k^i = \begin{cases} y_k^i, \text{если } y_k^i - f_i(x_k) \geqslant 0, \\ f_i(x_k), \text{если } y_k^i - f_i(x_k) < 0. \end{cases} \qquad (8.28)$$

Теперь легко вычисляется и вектор $\eta_k = r_k^{-1}(z_k - y_k)$:

$$\eta_k^i = r_k^{-1}(f_i(x_k) - y_k^i)_+,\ i = 0, \ldots, m. \qquad (8.29)$$

Итак, применительно к задаче (8.26) алгоритм приобретает следующий вид: на шаге с номером $k \geqslant 0$

$$y_k = \lambda_k e - \alpha \eta_{k-1}, \quad k, \alpha = 0,$$

$$x_k = \arg\min \left\{ \sum_{i=0}^{m} (f_i(x) - y_k^i)_+^2 : x \in X \right\},$$

$$\eta_k^i = r_k^{-1} (f_i(x_k) - y_k^i)_+, \quad i = 0, \ldots, m, \tag{8.30}$$

$$\lambda_{k+1} = \frac{1}{\eta_k^0} \sum_{i=0}^{m} \eta_k^i z_k^i = \frac{1}{\eta_k^0} \sum_{i=0}^{m} \left[f_i(x_k) + (y_k^i - f_i(x_k))_+ \right] \eta_k^i =$$

$$= \frac{1}{\eta_k^0} \sum_{i=0}^{m} f_i(x_k) \eta_k^i,$$

где при выводе формулы для λ_{k+1} использованы (8.28) и (8.29). Величина λ_0 должна быть выбрана из условия $\lambda_0 \leqslant f_0(x_*)$. Критерием останова служит выполнение неравенства $r_k - \alpha \leqslant \epsilon$, где $\epsilon > 0$ – наперед заданная точность.

Как следует из приведенных выше теорем, алгоритм всегда сходится. Если исходная задача является задачей линейного программирования, то он сходится за конечное число шагов.

Рассмотрим вопрос о скорости сходимости в общем случае. Множество M, лежащее в $\mathbf{R}^{m+1}$, можно рассматривать как надграфик некоторой функции $\omega(w)$, где $w \in \mathbf{R}^n$ – вектор с компонентами $z^1, \ldots, z^m$. Тогда

$$\omega(w) = \min_z \left\{ z^0 : (z^0, z^1, \ldots, z^m)^* \in M \right\} =$$

$$= \min \left\{ f_0(x) : f_i(x) \leqslant z^i, \ i = 1, \ldots, m, \ x \in X \right\},$$

$$M = \left\{ z : z^0 \geqslant \omega(w) \right\}.$$

Известно, что $\omega(w)$ – выпуклая функция. При этом $\lambda_* = f_0(x_*) = \omega(0)$.

Теорема 8.6. *Если функция $\omega(w)$ дважды непрерывно дифференцируема в точке $w = 0$ и*

$$(\omega''(0) w, w) \geqslant \mu \| w \|^2, \mu > 0,$$

то скорость сходимости алгоритма (8.30) *определяется соотношением*

$$\overline{\lim_{k \to \infty}} \frac{\delta_{k+1}}{\delta_{k-1}} \leqslant \frac{2L^2}{\mu} \sqrt{1 + \| u \|^2} \alpha,$$

где L – константа Липшица для единичной нормали к M в окрестности точки $z^0 = f_0(x_*), z^i = 0, i = 1, \ldots, m$, $u \in \mathbf{R}^m$ – единственный вектор Куна – Таккера задачи (8.26).

Доказательство. Так как функция $\omega(w)$ – гладкая, то согласно известным результатам § 2 вектор Куна – Таккера определяется един-

ственным образом и совпадает с вектором $-\omega'(0)$. Далее, в силу специального вида векторов e и e_1 в рассматриваемом случае

$$(e, e_1) = 1 / \sqrt{1 + \|u\|^2}.$$

Теорема 8.6 теперь получается как прямое следствие теоремы 8.5 и формулы (8.21).

Приведенный результат характеризует сходимость к нулю величин δ_k. На практике же нас интересует сходимость точек x_k к точке минимума x_*, или по крайней мере поведение величин $f_i(x_k)$, $i = 0,1, \ldots, m$.

Напомним, что $\delta_k = \lambda_* - \lambda_k = f_0(x_*) - \lambda_k$. С другой стороны, из формулы (8.18) следует, что

$$\sum_{i=1}^{m} (z_k^i)^2 \leqslant 4\delta_k / \mu$$

и, в частности,

$$|z_k^i| \leqslant \sqrt{4\delta_k/\mu}.$$

Но формулы (8.28) показывают, что $f_i(x_k) \leqslant |z_k^i|$. Поэтому

$$f_i(x_k) \leqslant \sqrt{4\delta_k / \mu}, i = 1, \ldots, m,$$

так что степень удовлетворения точке x_k ограничений задачи зависит от величины $\sqrt{\delta_k}$.

Как показывает теорема 8.6, скорость сходимости алгоритма определяется величиной α: чем меньше $\alpha > 0$, тем выше скорость.

В предыдущем пункте был приведен еще один способ ускорения сходимости за счет растяжения в подпространстве, ортогональном вектору e. Учитывая конкретный вид этого вектора, нетрудно усмотреть, что такое растяжение в данном случае сводится к умножению всех координат z^i, $i = 1, \ldots, m$, на величину $K \geqslant 1$, т.е. применительно к задаче (8.26), к замене всех функций $f_i(x)$, $i = 1, \ldots, m$, на функции $K f_i(x)$. Легко также показать, что при этом вектор u Куна – Таккера заменяется на вектор $K^{-1}u$. Использование результатов предыдущего пункта, в частности, формулы (8.25), убеждает нас в справедливости следующей теоремы.

Теорема 8.7. *Пусть выполнены условия теоремы* 8.6. *Тогда применение алгоритма, описываемого формулами* (8.30), *с заменой всех функций* $f_i(x)$, $i = 1, \ldots, m$, *на функции* $K f_i(x)$, $K \geqslant 1$, *обеспечивает скорость сходимости, задаваемую соотношением*

$$\overline{\lim_{k \to \infty}} \frac{\delta_{k+1}}{\delta_{k-1}} \leqslant \frac{2C^2L^2}{\mu} \sqrt{1 + K^2 \|u\|^2} \frac{\alpha}{K^2}.$$

Приведем теорему, гарантирующую выполнение условий теоремы 8.6.

Теорема 8.8. *Пусть* x_* – *решение задачи* (8.26) *и выполнены следующие условия:*

а) *функции* $f_i(x)$ *выпуклы и дважды непрерывно дифференцируемы;*

б) *градиенты* $f_i'(x)$, $i \in I_0$, *где*

$$I_0 = \{ i : f_i(x_*) = 0, i = 1, \ldots, m \},$$

линейно независимы, и компоненты вектора Куна – Таккера строго положительны;

в) *матрица вторых производных по x от функции Лагранжа*

$$L(x,u)=f_0(x)+\sum_{i=1}^{m}u^i f_i(x)$$

строго положительно определена в точке x_*, *т.е.*

$$(L''_{xx}(x_*,u)p,p)>0, p\neq 0.$$

Тогда выполнены условия теоремы 8.6.

Доказательство. Без ограничения общности будем считать, что множество I_0 содержит все индексы $i=1,\ldots,m$. Для w из окрестности нуля запишем систему уравнений

$$L'_x(x(w),u(w))=0, \tilde{f}(x(w))-w=0, \tag{8.31}$$

где w – вектор с компонентами $z^1,\ldots,z^m$, $\tilde{f}(x)$ имеет компоненты $f_1(x),\ldots,f_m(x)$. В этой системе x и u рассматриваются как неизвестные функции вектора w. При $w=0$ система (8.31) представляет собой запись необходимых и достаточных условий экстремума в задаче (8.26) и имеет в силу предположений теоремы единственное решение $x(0)=x_*$, $u(0)=u>0$. С другой стороны, матрица производных левой части системы (8.31) по x и u в точке $x=x(0)$, $u=u(0)$ имеет вид

$$\begin{bmatrix} L''_{xx}(x_*,u) & \tilde{f}'^{*}(x_*) \\ \tilde{f}'(x_*) & 0 \end{bmatrix}.$$

где $\tilde{f}'(x_*)$ – матрица размерности $m\times n$ с компонентами $\partial f_i(x)/\partial x^j$, $i=1,\ldots,m$, $j=1,\ldots,n$, причем x^j – j-я компонента вектора $x\in\mathbf{R}^n$. Как показано в § 6, при сделанных предположениях эта матрица невырождена, а поэтому по теореме о неявных функциях система (8.31) однозначно разрешима в окрестности $w=0$. При этом матрицы $x'_w(0)$, $u'_w(0)$ производных от x и u по w, имеющие вид

$$x'_w(0)=\begin{bmatrix} \dfrac{\partial x^1(0)}{\partial z^1} & \cdots & \dfrac{\partial x^1(0)}{\partial z^m} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial x^n(0)}{\partial z^1} & \cdots & \dfrac{\partial x^n(0)}{\partial z^m} \end{bmatrix},$$

$$u'_w(0)=\begin{bmatrix} \dfrac{\partial u^1(0)}{\partial z^1} & \cdots & \dfrac{\partial u^1(0)}{\partial z^m} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial u^m(0)}{\partial z^1} & \cdots & \dfrac{\partial u^m(0)}{\partial z^m} \end{bmatrix}.$$

удовлетворяют соотношениям

$$L''_{xx}(x_*, u)\, x'_w(0) + \tilde{f}'^*(x_*)\, u'_w(0) = 0,$$
$$\tilde{f}'(x_*)\, x'_w(0) - I_m = 0,$$

где I_m – единичная $m \times m$- матрица. Отсюда получаем

$$u'_w(0) = -(\tilde{f}'(x_*)(L''_{xx}(x_*, u))^{-1}\, \tilde{f}'^*(x_*))^{-1}.$$

В предположениях теоремы 8.8 матрица $u'_w(0)$ строго отрицательно определена.

Так как $u = u(0) > 0$, то $u(w) > 0$ при малых w и $x(w)$ есть решение задачи минимизации $f_0(x)$ при ограничениях $f_i(x) \leqslant z^i$, $i = 1, \ldots, m$, а $u(w)$– соответствующий вектор Куна – Таккера.

Как уже отмечалось, $u(w) = -\omega'(w)$. Поэтому $\omega''(0) = -u'_w(0)$ и, значит, матрица $\omega''(0)$ строго положительно определена, что доказывает выполнение условий теоремы 8.6. Для успешной работы приведенного алгоритма существенно, насколько хорошо и эффективно будет решаться вспомогательная задача. Как и в методе модифицированных функций Лагранжа, здесь возникают трудности двух видов.

В (8.27) минимизируемая функция имеет разрывы вторых производных. Поэтому желательно, чтобы, по крайней мере в окрестности решения, функция этих разрывов не имела. Вторая трудность связана со степенью вырожденности этой функции.

Рассмотрим, как ведет себя функция

$$\sum_{i=0}^{m} (f_i(x) - y_k{}^i)_+^2,$$

когда k достаточно велико, т.е. $y_k \approx \lambda_* e - \alpha e_1$. Тогда

$$y_k^0 \approx f_0(x_*) - \alpha / \sqrt{1 + \|u\|^2},$$
$$y_k^i \approx -\alpha u^i / \sqrt{1 + \|u\|^2},\ i = 1, \ldots, m.$$

Поэтому при больших k вспомогательная задача сводится к минимизации функции типа

$$\left(f_0(x) - f_0(x_*) + \frac{\alpha}{\sqrt{1 + \|u\|^2}}\right)_+^2 + \sum_{i=1}^{m} \left(f_i(x) + \frac{\alpha u^i}{\sqrt{1 + \|u\|^2}}\right)_+^2.$$

Если $\alpha > 0$, $u^i > 0$ для $i \in I_0$, то при x из некоторой окрестности x_* эта последняя функция приобретает вид

$$\left(f_0(x) - f_0(x_*) + \frac{\alpha}{\sqrt{1 + \|u\|^2}}\right)^2 + \sum_{i \in I_0} \left(f_i(x) + \frac{\alpha u^i}{\sqrt{1 + \|u\|^2}}\right)^2,$$

т.е. является гладкой функцией. Вычислим ее матрицу вторых производных в точке x_*. Простой подсчет показывает, что эта матрица равна

$$A = \frac{2\alpha}{\sqrt{1 + \|u\|^2}}\, L''_{xx}(x_*, u) + 2 \sum_{i \in I_0} f'_i(x_*)(f'_i(x_*))^*, \qquad (8.32)$$

где $f_i'(x_*)$ – вектор-столбец, $(f_i'(x_*))^*$ – вектор-строка.

Приведенный выше алгоритм представляет собой внешний цикл решения задачи. Он включает в себя внутренний цикл, связанный с решением задачи минимизации (8.27). Если множество X имеет простую структуру, в частности, если $X = \mathbf{R}^n$, то для решения задачи (8.27) можно использовать хорошо разработанные стандартные алгоритмы.

Из того, что матрица L''_{xx} строго положительно определена, и выражения (8.32) следует, что матрица A строго положительно определена, что является существенным условием высокой скорости сходимости всех алгоритмов безусловной оптимизации. При этом строгая положительная определенность матрицы A обусловлена тем фактом, что $\alpha > 0$. Из предыдущего изложения теперь ясно, что требования качества сходимости внешнего и внутреннего циклов приводят относительно α к противоречивым требованиям: для быстрой сходимости внешнего цикла необходимо α уменьшить до нуля, для быстрой сходимости внутреннего цикла α должно быть достаточно большим. Напомним, что ускорение сходимости может быть достигнуто также за счет умножения всех функций $f_i(x)$, $i = 1, \ldots, m$, на $K > 0$. По-видимому, невозможно дать общие рекомендации по выбору $\alpha > 0$ и $K > 1$. Однако разумное варьирование этих величин обеспечивает в целом быстро сходящийся процесс.

§ 9. АЛГОРИТМЫ И ПРИМЕРЫ РАСЧЕТОВ

1. Метод линеаризации. В § 4 были отмечены некоторые вычислительные аспекты метода линеаризации. Здесь даны дополнительные рекомендации, касающиеся практической стороны вопроса, приведены примеры расчетов.

При записи алгоритма метода линеаризации для практической реализации удобно задачу минимизации (4.1) сформулировать в следующем виде.

Минимизировать $f_0(x)$, $x \in E^n$, при ограничениях

$$f_i(x) \leqslant 0, \quad i = 1, \ldots, m,$$

$$f_i(x) = 0, \quad i = m+1, \ldots, m+l. \tag{9.1}$$

Обозначим

$$F(x) = \max\{0, f_1(x), \ldots, f_m(x), |f_{m+1}(x)|, \ldots, |f_{m+l}(x)|\},$$

$$I_\delta^-(x) = \{i: f_i(x) \geqslant F(x) - \delta, \quad i = 1, \ldots, m\},$$

$$I_\delta^0(x) = \{i: |f_i(x)| \geqslant F(x) - \delta, \quad i = m+1, \ldots, m+l\},$$

$$\Phi_N(x) = f_0(x) + NF(x), \quad \|p\|^2 = \sum_{i=1}^{n} (p_i)^2.$$

Метод линеаризации порождает две итерационные последовательности: $\{x_k\}$, $k = 1, 2, \ldots$, сходящуюся к решению задачи (9.1) x_*, и $\{u_k\}$, $k = 1, 2, \ldots$, сходящуюся к множителям Лагранжа u_*. Последовательность $\{u_k\}$ порождается решением задачи, двойственной к (4.4).

Запишем эту задачу в виде, удобном для реализации на ЭВМ. Раскроем норму в выражении (4.21) и приведем его к виду

$$\varphi(u) = -\tfrac{1}{2}(Au, u) + (d, u) + C. \tag{9.2}$$

Здесь матрица A симметрична и состоит из элементов

$$\{a_{ij}\} \doteq \{ f_i'(x) f_j'^{*}(x) \},\ i, j \in I_\delta^-(x) \cup I_\delta^0(x),$$

i-я компонента вектора d равна $-f_0'(x) f_i'^{*}(x) + f_i(x),\ i \in I_\delta^-(x) \cup I_\delta^0(x)$, а $C = \frac{1}{2} f_0'(x) f_0'^{*}(x)$. Получим задачу квадратичного программирования

$$\max_u \{ \varphi(u):\ u^i \geqslant 0,\ i \in I_\delta^-(x) \}. \tag{9.3}$$

Эту задачу можно решать методом сопряженных градиентов, описанным в п. 8 § 3, или любым другим конечным методом для решения задач квадратичного программирования.

Опишем теперь алгоритм метода линеаризации. Пусть выбраны произвольное начальное приближение x_0, точность решения исходной задачи (9.1) ϵ_1, точность решения (9.3) ϵ_2 и величины $N_0 > 0$, $\delta_0 > 0$, $0 < \epsilon < 1$.

Общий шаг алгоритма. Пусть точка x_k и числа N_k, δ_k построены.

1. Строим множества $I_\delta^-(x_k)$, $I_\delta^0(x_k)$ и задачу (9.3).

2. Решаем задачу (9.3). Ее решение — множители Лагранжа u_k^i, $i \in I_{\delta_k}^-(x_k) \cup I_{\delta_k}^0(x_k)$. Если задача (9.3) несовместна, то полагаем

$$x_{k+1} = x_k,\ \delta_{k+1} = \delta_k / 2,\ N_{k+1} = N_k$$

и возвращаемся к 1.

3. Если решение задачи (9.3) получено и

$$N_k \geqslant \sum_{i \in I_{\delta_k}^-(x_k)} u_k^i + \sum_{i \in I_{\delta_k}^0(x_k)} |u_k^i|,$$

то $N_{k+1} = N_k$.

В противном случае

$$N_{k+1} = 2\left[\sum_{i \in I_{\delta_k}^-(x_k)} u_k^i + \sum_{i \in I_{\delta_k}^0(x_k)} |u_k^i| \right].$$

4. Вычисляем вектор направления движения

$$p_k = -f_0'(x_k) - \sum_{i \in I_{\delta_k}^-(x_k)} u_k^i f_i'(x_k) - \sum_{i \in I_{\delta_k}^0(x_k)} u_k^i f_i'(x_k). \tag{9.4}$$

5. Если

$$\| p_k \|^2 \leqslant \epsilon_1, \tag{9.5}$$

переходим к 7.

6. Если (9.5) не выполняется, то полагаем

$$x_{k+1} = x_k + \alpha_k p_k,\ \delta_{k+1} = \delta_k,$$

где шаг α_k выбирается равным $1/2^{q_0}$, а q_0 — первое из целых чисел $q = 0, 1, \ldots$, для которого выполняется соотношение

$$\Phi_{N_{k+1}}\left(x_k + \frac{1}{2^q} p_k\right) \leqslant \Phi_{N_{k+1}}(x_k) - \frac{1}{2^q} \epsilon \| p_k \|^2. \tag{9.6}$$

Возвращаемся к 1.

7. Выход.

Из опыта практической работы можно сделать вывод, что δ_0 следует выбирать достаточно большим, чтобы охватить сразу все ограничения. Однако здесь необходимо учитывать еще фактор разрешимости возникающей при этом задачи (9.3). Если задача разрешима, то учитываются все ограничения сразу, быстро выделяются активные и $u(x_k) \to u_*, x_k \to x_*$. В противном случае нужно дробить δ и при каждом дроблении начинать работу сначала. Если δ_0 велико, то это приводит к значительным затратам времени.

Хорошо зарекомендовал себя выбор δ_0 в виде

$$\delta_0 = F(x_0) + \epsilon^1,$$

где $\epsilon^1 > 0$– некоторое число. При таком выборе δ_0 сразу учитываются ограничения, которые сильно нарушены, и точка x_k быстрее начинает удовлетворять всем ограничениям задачи. В качестве ϵ^1 можно использовать ϵ, фигурирующее в (9.6). Однако лучше подбирать его, руководствуясь конкретной задачей или классом задач.

Пусть среди ограничений (9.1) имеются простые ограничения на координаты вектора x – типа

$$a^j \leqslant x^j \leqslant b^j, \quad j = 1, \ldots, n, \tag{9.7}$$

где a^j, b^j – произвольные числа. В реальных практических задачах такие ограничения вводятся из физических соображений и их нарушение часто приводит к различного рода аварийным ситуациям. В начале итерационного процесса метода линеаризации, когда множество $I_\delta(x)$ (4.3) еще далеко от истинного множества активных ограничений $I_0(x_*) = I_*$, возможен выход точки x_k за ограничения. Поэтому при наличии ограничений типа (9.7) рекомендуется заранее включать в $I_\delta(x)$ все ограничения такого вида, а на оставшиеся ограничения распространить аппарат построения множества $I_\delta(x)$.

Несмотря на симметричность матрицы A в задаче (9.3), позволяющей хранить в памяти ЭВМ лишь половину этой матрицы, все же значительная часть памяти ЭВМ уходит на ее хранение. Из вида элементов матрицы A ясно, что при наличии ограничений типа (9.7) она приобретает блочную структуру. Все блоки, за исключением одного, состоят из таких элементов, что для их определения нет необходимости заставлять ЭВМ вычислять скалярные произведения $(f_i'(x), f_j'(x))$. Это дает возможность хранить в памяти ЭВМ лишь незначительную часть этой матрицы или вообще не хранить ее. Если вектор x имеет большую размерность и ограничения (9.7) наложены на все его компоненты, это приводит к значительной экономии памяти.

Сделаем еще замечание относительно соотношения между величинами ϵ_1 и ϵ_2, указанными в алгоритме. Точность решения задачи (9.3) ϵ_2 следует задавать выше точности решения основной задачи ϵ_1 на один или несколько порядков, так как недостаточно точное решение задачи (9.3) может не привести к результатам.

2. **Ускоренный метод линеаризации.** В отличие от алгоритма, изложенного в § 6 для ускоренного метода, здесь приводится алгоритм, может быть, менее обозримый, но более близкий к практическому использованию.

Предполагается, что читатель знаком с п.1 этого параграфа, поэтому при изложении алгоритма некоторые детали, например, связанные с выбором параметров N_k и δ_k, здесь опускаются. Введем множество

$$I(x) = \{ i \in I_\delta^-(x) \cup I_\delta^0(x):\ f_i(x) = 0 \}.$$

При описании алгоритма ускоренного метода в § 6 было показано, что итерационная последовательность $\{x_k\}$ разбивается на отрезки, в которых переход от x_k к x_{k+1} происходит либо по формуле

$$x_{k+1} = x_k + y_k, \tag{9.8}$$

где y_k – решение системы уравнений

$$\begin{aligned} &A(x_k, h_k)y + (f_0'(x_k))^* + \sum_{i \in I(x_k)} v^i (f_i'(x_k))^* = 0, \\ &f_i'(x_k)y + f_i(x_k) = 0,\ \ i \in I(x_k), \end{aligned} \tag{9.9}$$

либо по формуле метода линеаризации с выбором шага согласно формуле (9.6). При этом шаг из x_k в x_{k+1} осуществляется по формуле (9.8) только в том случае, если выполнено условие

$$\| p(x_{k+1}) \| \leqslant \gamma \| p_k \|, \tag{9.10}$$

где γ, $0 < \gamma < 1$, – то же, что в § 6.

Определим некоторую величину $\mathcal{Y}$ принимающую значения:

$$\mathcal{Y} = \begin{cases} 1, \text{ если система (9.9) разрешима,} \\ 2, \text{ если система (9.9) разрешима и выполнено условие (9.10),} \\ 0, \text{ если система (9.9) неразрешима либо не выполнено (9.10).} \end{cases}$$

Опишем алгоритм ускоренного метода. Пусть выбраны $N > 0$, $\delta > 0$, начальное приближение x_0, а также числа $0 < \gamma < 1$, $0 < \epsilon < 1$, $0 < q_0 < 1$, $h > 0$, ϵ_1, ϵ_2. Положим $C_0 = +\infty$, $\alpha_0 = +\infty$, $\mathcal{Y} = 0$.

(Под величиной $+\infty$ понимается достаточно большое число, допустимое разрядной сеткой ЭВМ, на которой происходит реализация алгоритма.)

Общий шаг алгоритма. Пусть x_k, C_k и q_k уже построены.

1. Решаем задачу (9.3) при $x = x_k$. Получим $u_k^i = u^i(x_k)$, $i \in I_{\delta_k}^-(x) \cup I_{\delta_k}^0(x_k)$. Вычисляем вектор $p_k = p(x_k)$.

2. Если $\mathcal{Y} = 1$, переходим к 8.

3. Проверяем, не является ли точка x_k решением. Если $\| p_k \|^2 \leqslant \epsilon_1$, то переходим к 10.

4. Если $\mathcal{Y} = 2$, то переходим к 7.

5. Если $\alpha_{k-1} \| p_k \| > q_k$, то $q_{k+1} = q_k$ и идем на 9.

6. Если $\| p_k \| > C_k$, то полагаем $q_{k+1} = q_k$ и переходим к 9.

7. Полагаем $h_k = \min\{h, \| p_k \|\}$, решаем систему уравнений (9.9) относительно $\mathcal{Y}$. Если система не имеет решения, то полагаем $q_{k+1} = q_k$, $C_{k+1} = \gamma \| p_k \|$, $\mathcal{Y} = 0$ и переходим к 9. В противном случае полагаем

$$\bar{x} = x_k + y_k,\ \ \mathcal{Y} = 1 \tag{9.11}$$

и возвращаемся к 1 на вычисление $p(\bar{x})$.

8. Если

$$\| p(\bar{x}) \| \leqslant \gamma \| p_k \|, \tag{9.12}$$

то полагаем

$$x_{k+1} = \bar{x}, \quad C_{k+1} = \gamma \| p_k \|, \quad q_{k+1} = q_k, \quad \mathcal{Y} = 2$$

и переходим к 3. В противном случае полагаем

$$C_{k+1} = \gamma \| p_k \|, \quad q_{k+1} = q_k/2, \quad \mathcal{Y} = 0$$

и переходим к следующему шагу.

9. Дробим $\alpha = 1$ путем деления пополам до выполнения неравенства

$$f_0(x_k + \alpha p_k) + NF(x_k + \alpha p_k) \leqslant f_0(x_k) + NF(x_k) - \alpha\epsilon \| p_k \|^2. \tag{9.13}$$

Полагаем

$$x_{k+1} = x_k + \alpha p_k \tag{9.14}$$

и переходим к 1.

10. Выход.

Из построенного алгоритма видно, что в начале итерационного процесса, когда точка x_k еще очень далека от x_*, работает только метод линеаризации. После первого выполнения условия $\alpha_{k-1} \| p_k \| \leqslant q_k$, $0 < q_k < 1$, $q_k \to 0$, включается в работу проверка на выполнение условий для работы по формуле (9.11). Все теоретические выкладки, проведенные в § 6 и обеспечивающие движение по формуле (9.11), справедливы лишь в локальной окрестности точки x_*. Поэтому введение в алгоритм параметра q_k для практических целей вполне оправдано, так как иначе будет выполняться много лишней работы. При выборе q_0 можно руководствоваться, например, заданной точностью решения исходной задачи ϵ_1. Так, при расчетах, в которых заданная точность решения была указана в интервале $10^{-4} - 10^{-7}$, параметр q_0 полагался равным 0.1, и при этом были получены хорошие результаты.

С другой стороны, из алгоритма видно, что после первого удачного шага по формуле (9.11) следующий шаг снова происходит по этой формуле; естественно, при этом проверяется выполнение условия (9.12).

В проведенных расчетах переход от формул (9.11) к формулам (9.14) и наоборот происходил в основном в начале включения в работу формул (9.11). Завершение итерационного процесса происходило по формулам (9.11). Величина γ бралась близкой к единице. Хорошая сходимость была получена даже при $\gamma = 1$.

Для решения системы уравнений (9.9) можно использовать любой метод, предназначенный для решения систем линейных алгебраических уравнений, например метод Гаусса.

3. Примеры расчетов. Приведем некоторые примеры задач, решенных описанным в пп. 1, 2 этого параграфа алгоритмами.

Пример 1. Минимизировать

$$f_0(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4$$

при ограничениях

$$f_1(x) = x_1 + x_2^2 + x_3^3 = 2 + 3\sqrt{2}, \quad f_2(x) = x_2 - x_3^2 + x_4 = -2 + 2\sqrt{2},$$

$$f_3(x) = x_1 x_5 = 2.$$

Таблица 1

Метод	x_0	x_*	$f(x_*)$	$\|p\|$
Линеаризации	x_{0_1}	x_{*_1}	44.02	$1.24\cdot10^{-4}$
	x_{0_2}	x_{*_2}	27.87	$2.33\cdot10^{-5}$
	x_{0_3}	x_{*_3}	607.04	$1.74\cdot10^{-3}$
	x_{0_4}	x_{*_4}	0.0293	$7.74\cdot10^{-6}$
Ускоренный	x_{0_1}	x_{*_1}	44.02	$5.12\cdot10^{-5}$
	x_{0_2}	x_{*_2}	27.87	$1.27\cdot10^{-5}$
	x_{0_3}	x_{*_3}	607.04	$1.88\cdot10^{-6}$
	x_{0_4}	x_{*_4}	0.0293	$4.04\cdot10^{-5}$

Таким образом, нужно минимизировать нелинейную функцию пяти переменных при трех нелинейных ограничениях типа равенств. Эта задача решалась с различными начальными значениями методом линеаризации и ускоренным методом.

В качестве начальных были выбраны точки

$x_{0_1} = (-1, 3, -0.5, -2, -3)$, $x_{0_2} = (-1, 2, 1, -2, -2)$,
$x_{0_3} = (-2, -2, -2, -2, -2)$, $x_{0_4} = (1, 1, 1, 1, 1)$.

Соответственно этим начальным точкам получены точки решения x_*:

$x_{*_1} = (-0.7034, 2.636, -0.09636, -1.798, -2.843)$,

$x_{*_2} = (-1.273, 2.410, 1.195, -0.1542, -1.571)$,

$x_{*_3} = (-2.791, -3.004, 0.2054, 3.875, -0.7166)$,

$x_{*_4} = (1.117, 1.220, 1.538, 1.973, 1.791)$.

В табл. 1 приведены результаты счета обоих методов, по которым можно сделать сравнительные характеристики этих методов.

Заданы величины $q_0 = 0.1$, $\gamma = 1$. Под одним вычислением функций понимается одно вычисление минимизируемой функции $f_0(x)$ и функций $f_i(x)$, $i \in I_\delta(x)$.

Из табл. 1 видно, что ускоренный метод особенно эффективен, когда метод линеаризации движется с маленьким шагом, так как на дробление шага уходит много затрат ресурсов. Каждое дробление требует вычисления функций для проверки условия (9.13). В случае движения из точки x_{0_4} метод линеаризации оказался достаточно эффективным.

В ускоренном методе основные затраты по вычислению функций уходят на построение матрицы $A(x, h)$. Чтобы один раз построить матрицу $A(x, h)$ порядка $n \times n$ нужно $n(1 + 2n)$ раз обратиться к вычислению функций. Так, в описанном выше примере при движении из точки x_{0_1} ускоренным методом 220 раз понадобилось вычислить значения функций, чтобы обеспечить четыре раза построение матрицы $A(x, h)$ порядка 5×5, в то время как для обеспечения 19-суммарного числа итераций метода понадобилось лишь 19 раз обратиться к вычислению функций.

Общее число итераций	Число итер. по (9.11)	Число лишних итер. по (9.11)	Число вычисл. функц.	Число вычисл. градиентов	Шаг на посл. итерации
481	–	–	3516	482	$1.3 \cdot 10^{-13}$
399	–	–	2623	400	$9.09 \cdot 10^{-13}$
605	–	–	6506	606	$9.09 \cdot 10^{-13}$
21	–	–	49	22	1
19	4	0	239	20	1
29	3	0	258	30	1
112	8	5	1313	118	1
11	3	0	176	12	1

Из этого становится очевидной целесообразность введения в алгоритм параметра q_k, обеспечивающего движение вначале процесса по чистому методу линеаризации и уменьшающего вероятность лишних обращений к вычислениям, реализующим движение по формуле (9.11), когда она еще неправомочна.

Из табл. 1 видно также, что затраты на вычисление градиентов от функций в ускоренном методе во всех случаях значительно меньше, чем в методе линеаризации.

Рассмотрим еще один пример минимизации нелинейной функции с двумя нелинейными ограничениями [43]. Одно ограничение – неравенство, второе – равенство.

Пример 2. Найти минимум

$$f_0(x) = 0.5(x_1 + x_2)^2 + 50(x_2 - x_1)^2 + x_3^2$$

при ограничениях

$$f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + (x_3 - 1)^2 - 1.5 \leqslant 0,$$

$$f_2(x) = \sin(x_1 + x_2) - x_3 = 0.$$

Начальная точка $x_0 = (-1, 4, 5)$. Минимум в точке $x_* = (0.229, 0.229$ $0.442)$ и значение $f_0(x_*) = 0.3$. Эта задача решалась методом линеаризации и ускоренным методом. Для достижения точности по $\|p\|$, равной 10^{-5}, по методу линеаризации потребовалось 740 итераций, по ускоренному методу 27 итераций.

При $\gamma = 1$, $q_0 = 0.1$ три раза произошел выход на (9.11). Первый раз переход с (9.14) на (9.11) был преждевременным. Два последних шага движение шло по (9.11).

Поскольку разница в числе итераций по этим методам очень большая, понятно, насколько эффективным здесь оказался ускоренный метод.

Особенность следующего примера [18] состоит в том, что матрица вторых производных от функции Лагранжа по переменной x имеет отрицательное собственное значение -136, а все положительные собственные значения меньше 2, т.е. она не является положительно определенной.

Пример 3. Целевая функция

$$f_0(x) = e^{x_1 x_2 x_3 x_4 x_5} - 0.5(x_1^3 + x_2^3 + 1)^2$$

Ограничения-равенства:

$$f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,$$

$$f_2(x) = x_2 x_3 - 5x_4 x_5 = 0,$$

$$f_3(x) = x_1^3 + x_2^3 + 1 = 0.$$

Начальная точка $x_0 = (-2, 2, 2, -1, -1)$. Точка минимума $x_* = (-1.717, 1.596, 1.827, -0.7636, -0.7636)$. Значение целевой функции в x_* равно 0.05395.

Для достижения точности по $\| p \|$, равной 10^{-4}, методу линеаризации потребовалось 32 итерации, ускоренному методу 6 итераций, из них последние 4 итерации процесс двигался по формуле (9.11).

Приведем еще примеры задач, решенных методом линеаризации и его модификацией, которые подробно описаны и проиллюстрированы в [45, с. 416–424]. Рассматривался ряд задач, имеющих одну и ту же целевую функцию, но различный набор ограничений-неравенств. Целевая функция зависит от двух переменных и на заданном интервале значений этих переменных имеет один пик и одну седловую точку. Пик целевой функции находится в точке с координатами $x_1 = 81.154841$, $x_2 = 69.135588$; $f(x_*) = 61.9059345$. Условный максимум находится в точке с координатами $x_1 = 75.000000$, $x_2 = 65.000000$; значение функции в этой точке равно 58.903436.

Целевая функция имеет вид

$$f(x) = B_1 + B_2 x_1 + B_3 x_1^2 + B_4 x_1^3 + B_5 x_1^4 + B_6 x_2 +$$
$$+ B_7 x_1 x_2 + B_8 x_1^2 x_2 + B_9 x_1^3 x_2 + B_{10} x_1^4 x_2 +$$
$$+ B_{11} x_2^2 + B_{12} x_2^3 + B_{13} x_2^4 + B_{14}(1/(x_2 + 1)) +$$
$$+ B_{15} x_1^2 x_2^2 + B_{16} x_1^3 x_2^2 + B_{17} x_1^3 x_2^3 + B_{18} x_1 x_2^2 +$$
$$+ B_{19} x_1 x_2^3 + B_{20} e^{0.0005 x_1 x_2},$$

где

$B_1 = 75.1963666677$, $B_{11} = 0.2564581253$,
$B_2 = -3.8112755343$, $B_{12} = -0.0034604030$,
$B_3 = 0.1269366345$, $B_{13} = 0.0000135139$,
$B_4 = -0.0020567665$, $B_{14} = -28.1064434908$,
$B_5 = 0.0000103450$, $B_{15} = -0.0000052375$,
$B_6 = -6.8306567613$, $B_{16} = -0.0000000063$,
$B_7 = 0.0302344793$, $B_{17} = 0.0000000007$,
$B_8 = -0.0012813448$, $B_{18} = 0.0003405462$,
$B_9 = 0.0000352599$, $B_{19} = -0.0000016638$
$B_{10} = -0.0000002266$, $B_{20} = -2.8673112392$.

Таблица 2

Номер задачи	Номера ограничений	Номер задачи	Номера ограничений
I	1, 2, 3, 4, 5	IV	5, 6, 7, 8, 9, 10
II	5, 6, 7, 8	V	5, 6, 7, 8, 9, 11, 12
III	5, 6, 7, 8, 9		

Набор ограничений-неравенств, которые использовались для построения конкретных задач, таков:

$f_1(x) = -x_1 \leqslant 0$, $f_2(x) = -x_2 \leqslant 0$,

$f_3(x) = x_1 - 95 \leqslant 0$, $f_4(x) = x_2 - 75 \leqslant 0$,

$f_5(x) = 700 - x_1 x_2 \leqslant 0$, $f_6(x) = x_1 - 75 \leqslant 0$,

$f_7(x) = x_2 - 65 \leqslant 0$, $f_8(x) = 5(x_1/25)^2 - x_2 \leqslant 0$,

$f_9(x) = 5(x_1 - 55) - (x_2 - 50)^2 \leqslant 0$, $f_{10}(x) = 54 - x_i \leqslant 0$,

$f_{11}(x) = (x_1 - 45) - (3/2)(x_2 - 45) \leqslant 0$,

$f_{12}(x) = (4/25)(x_2 - 40) - x_1 + 35 \leqslant 0$.

В табл. 2 приведены варианты номеров ограничений, с помощью которых было конкретизировано пять задач. Во всех задачах требовалось максимизировать $f_0(x)$ при этих ограничениях.

При решении указанных задач чистый метод линеаризации оказался достаточно эффективным, поэтому в табл. 3 приводятся результаты расчетов по методу линеаризации.

При решении этих задач метод линеаризации почти на всех итерациях работал с шагом, равным единице. Поэтому разница в числе итераций по ускоренному методу здесь была не столь разительна, как в примерах, описанных выше, когда шаг метода линеаризации очень мал. Например, при движении из недопустимой точки $x_0 = (95, 10)$ в первой задаче по ускоренному методу решение получено за 56 итераций, причем только последняя итерация была осуществлена по формуле (9.11), а 55 итераций процесс шел с учетом формул метода линеаризации.

Таблица 3

Номер задачи	Начальная точка x_0	x_*	$f_0(x_*)$	$\|p\|$	Число итераций
I	95, 10	81.1548, 69.1356	61.9059	$5.47 \cdot 10^{-6}$	74
II	31, 48	75, 65	58.9034	$2.50 \cdot 10^{-9}$	22
III	31, 48	75, 65	58.9034	$4.03 \cdot 10^{-9}$	47
IV	68.8, 31.2	75, 65	58.9034	$5.06 \cdot 10^{-19}$	51
V	68.8, 31.2	75, 65	58.9034	$6.06 \cdot 10^{-17}$	35

Большое количество задач, описанных в [45], также было решено описанными методами. Характер поведения методов соответствует картине, описанной на предыдущих примерах.

Мы здесь не будем останавливаться на перечислении этих примеров, а дадим пример расчета практической задачи. Эта задача возникает из нужд народного хозяйства и связана с рациональным планированием комплексного использования водных ресурсов бассейна реки Днепр [22]. Задача имеет 136 ограничений, из них 112 простых ограничений на координаты и 24 нелинейных ограничения типа равенств.

Сформулируем задачу. Максимизировать

$$f_0(x) = \sum_{i=1}^{12} (-2.188 + 19.95x_{24+i} + 0.07656x_i) +$$

$$+ \sum_{i=1}^{12} (11.56 - 24.89x_{36+i} - 0.7135x_{12+i} + 2.155x_{36+i}x_{12+i})$$

при ограничениях

$$f_1(x) \div f_{24}(x): \quad 51.2 \leqslant x_i \leqslant 51.4,$$

$$f_{25}(x) \div f_{48}(x): \quad 15 \leqslant x_{12+i} \leqslant 16.1,$$

$$f_{49}(x) \div f_{72}(x): \quad 0.4 \leqslant x_{24+i} \leqslant 4.6,$$

$$f_{73}(x) \div f_{96}(x): \quad 0.5 \leqslant x_{36+i} \leqslant 4.8,$$

$$f_{97}(x) \div f_{104}(x): \quad 0 \leqslant x_{48+j} \leqslant 0.7,$$

$$f_{105}(x) \div f_{112}(x): \quad 0 \leqslant x_{52+j} \leqslant 0.7,$$

$$f_{112+j}(x) = W(x_j, x_{24+j}) + c_j - 2.68x_{24+j} - W(x_{j-1}, x_{24+j-1}) = 0,$$

$$f_{116+j}(x) = W(x_{4+j}, x_{28+j}) + c_{4+j} -$$

$$- 2.68x_{28+j} - W(x_{3+j}, x_{27+j}) - 2.68x_{48+j} = 0,$$

$$f_{120+j}(x) = W(x_{8+j}, x_{32+j}) + c_{8+j} -$$

$$- 2.68x_{32+j} - W(x_{7+j}, x_{31+j}) = 0,$$

$$f_{124+k}(x) = W(x_{12+k}, x_{36+k}) + c_{12+k} -$$

$$- 2.68(x_{24+k} + x_{36+k}) - W(x_{11+k}, x_{35+k}) = 0,$$

$$f_{128+k}(x) = W(x_{16+k}, x_{40+k}) + c_{16+k} -$$

$$- 2.68(x_{28+k} + x_{40+k}) - W(x_{15+k}, x_{39+k}) - 2.68x_{52+k} = 0,$$

$$f_{132+k}(x) = W(x_{20+k}, x_{44+k}) + c_{20+k} -$$

$$- 2.68(x_{32+k} + x_{44+k}) - W(x_{19+k}, x_{43+k}) = 0, \quad i = \overline{1, 12}, \quad j = k = \overline{1, 4}.$$

Таблица 4

		a_1	a_2	a_3	a_4	a_5
j	$\overline{1,4}$	34.547	−0.55878	8.05339	−0.02252	−0.29316
k	$\overline{1,4}$	20.923	−4.22088	1.42061	−0.41040	−0.15082

Таблица 4 (окончание)

		a_6	a_7	a_8	a_9	a_{10}
j	$\overline{1,4}$	−0.013521	0.00042	0.00267	0.000281	0.0000032
k	$\overline{1,4}$	0	−0.00826	0.00404	0.000168	−0.000038

Таблица 5

i	1	2	3	4	5	6
c_i	5.61	4.68	1.62	1.8	2.13	2.1
c_{12+i}	−0.02	−0.01	−0.16	−0.47	−0.75	−0.94

Таблица 5 (окончание)

i	7	8	9	10	11	12
c_i	1.99	2.02	2.14	2.15	2.36	2.63
c_{12+i}	−0.93	−0.99	−0.42	−0.07	0.04	−0.06

Таблица 6

i	1	2	3	4	5	6
x_{*_i}	51.40	51.40	51.40	51.40	51.40	51.38
$x_{*_{12+i}}$	16.10	16.10	16.10	16.10	16.10	16.03
$x_{*_{24+i}}$	2.009	1.751	0.6233	0.6709	0.7928	0.7871
$x_{*_{36+i}}$	1.502	1.740	0.5835	0.5000	0.5096	0.5000
$x_{*_{48+i}}$	$-1.49\cdot10^{-7}$	$-2.7\cdot10^{-7}$	$1.95\cdot10^{-7}$	$1.71\cdot10^{-8}$	$1.12\cdot10^{-7}$	$1.71\cdot10^{-7}$

Таблица 6 (окончание)

i	7	8	9	10	11	12
x_{*_i}	51.38	51.37	51.20	51.20	51.20	51.20
$x_{*_{12}+i}$	15.92	15.80	15.89	15.64	15.00	15.00
$x_{*_{24}+i}$	0.7431	0.7551	0.8245	0.8026	0.8792	0.9796
$x_{*_{36}+i}$	0.5000	0.5000	0.5852	1.002	1.482	0.9712
$x_{*_{48}+i}$	$-7.84\cdot10^{-8}$	$-1.74\cdot10$				

где

$$W(x, y) = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 +$$
$$+ a_7x^3 + a_8x^2y + a_9xy^2 + a_{10}y^3$$

При $j = k = 1$ $W(x_0, x_{24}) = W(50.82,2)$, $W(x_{12}, x_{36}) = W(15.5, 2.3)$. Значения коэффициентов a_i, $1 \leqslant i \leqslant 10$, для $W(x, y)$ приведены в табл. 4, значения c_i, $1 \leqslant i \leqslant 24$, – в табл. 5.

Задача решалась методом линеаризации. Начиная из точки x_0, за 13 итераций была получена точка x_* при $\| p \| = 8.139\cdot10^{-4}$. Точка x_0 следующая: $x_{1_0} \div x_{12_0} = 51.35$, $x_{13_0} \div x_{24_0} = 15.5$, $x_{25_0} \div x_{36_0} = 2.5$, $x_{37_0} \div x_{48_0} = 2.6$, $x_{49_0} \div x_{56_0} = 0.3$. Значения координат точки x_* приведены в табл. 6. Значение целевой функции в точке x_*: $f_0(x_*) = 36.4138$. Итерационный процесс метода линеаризации двигался с шагом $\alpha_k = 1$, $1 \leqslant k \leqslant 13$. Все расчеты, приводимые здесь, проводились на ЭВМ БЭСМ-6.

БИБЛИОГРАФИЧЕСКИЙ КОММЕНТАРИЙ

Так как литература, посвященная необходимым условиям экстремума, теории двойственности в выпуклом программировании, минимаксным задачам и численным методам решения различных экстремальных задач, насчитывает тысячи наименований, то нет никакой возможности здесь дать какой-либо сравнительный анализ. Поэтому ограничимся лишь указаниями на книги и статьи, упомянутые в тексте и имеющие самое прямое отношение к предмету, рассматриваемому в этой книге.

При изложении в § 2 теории необходимых условий экстремума и двойственности в выпуклом программировании за основу брались книги А.Д. Иоффе и В.М. Тихомирова [7], Б.Н. Пшеничного [23, 28], Р.Т. Рокафеллара [33], И. Экланда, Р. Темама [47]. Теоретический анализ методов штрафных функций подробно изложен в недавно вышедшей книге К. Гросмана и А.А. Каплана [2], где можно найти дальнейшие ссылки на литературу. Однако при исследовании условий, когда точки минимума штрафной функции совпадают с решениями исходной задачи, в п. 7 § 2 за основу была взята статья Ф. Кларка [9].

Задача квадратичного программирования – основная при использовании метода линеаризации. От того, насколько быстро и экономно с точки зрения вычислений и требуемой памяти ЭВМ она решается, зависит эффективность метода, особенно при решении задач большого объема. В связи с этим за основу в § 3 был взят метод решения, обобщающий симплекс-метод линейного программирования (см. Дж. Данциг [3]). При этом мы обращаем внимание на необходимость экономной организации вычислений при работе с разреженными матрицами, для чего использовались подходы, развитые Б.А. Муртагом и М.А. Саундерсом [11], а также мультипликативное представление обратных матриц. Такое представление не единственно возможное и не всегда самое оптимальное. Подробнее с этим можно познакомиться в книге Р. Тьюарсона [36]. Помимо изложенного в § 3 существует и много других методов решения задач квадратичного программирования, которые можно найти в статье В.А. Даугавет [5], в книгах Г. Кюнци и В. Крелле [10], Б.Н. Пшеничного и Ю.М. Данилина [28], в сборнике [46]. Следует сказать, что большинство из этих методов переносятся на задачу минимизации произвольной выпуклой функции при линейных ограничениях, как это показывается в уже упомянутых работах [11, 46] и статье Б.Н. Пшеничного и И.Ф. Ганжелы [27]. Подробнейшее изложение методов сопряженных направлений применительно к минимизации квадратичных функций имеется в [28] и книге Е. Полака [19].

Метод линеаризации в рассматриваемом в книге виде впервые был изложен в статье Б.Н. Пшеничного [24], а применительно к решению равенств и неравенств – в [25]. Его модификации, отличающиеся вспомогательной задачей, рассматривают У.М. Гарсиа-Паломарес, О.Л. Мангасариан [1], С.М. Робинсон [31, 32], С.П. Хан [39]. Правда, вначале ими рассматривалась лишь локальная сходимость. Позднее в статьях [40–42] С.П. Хан рассмотрел и вопросы глобальной сходимости. Этой же проблематике посвящены статьи В.М. Панина [13–16] и М. Пауэлла [17, 18]. Следует отметить, что в упомянутых работах за счет использования во вспомогательной задаче вторых производных функции Лагранжа (в явном или неявном виде) достигалась сверхлинейная скорость сходимости. Однако при этом по существу требовалась строгая положительная определенность матрицы вторых производных функции Лагранжа. Этот недостаток был преодолен в работе Б.Н. Пшеничного и Л.А. Соболенко [30].

Решение минимаксных задач методом линеаризации (другие методы ее решения рассмотрены в книге В.Ф. Демьянова и В.Н. Малоземова [6]) исследовалось в книге

Б.Н. Пшеничного и Ю.М. Данилина [28] и в статье Б.Н. Пшеничного [26]. То, что для чебышевских точек метод линеаризации обеспечивает квадратичную сходимость, показано в работе В.А. Даугавет и В.Н. Малоземова [4]. Решение различных обобщенных задач минимакса методами, являющимися модификациями метода линеаризации, рассматривалось К.Кивислом [8], В.М. Паниным [13, 14], С.П. Ханом [42], Р. Флетчером [38].

Все расчеты по методу линеаризации, приводимые в § 9, проводила Л.А. Соболенко. При этом решались как примеры из литературы, так и многие реальные задачи. Весь этот материал невозможно отразить в книге, и поэтому в § 9 вошли лишь такие примеры, которые показались автору в том или ином смысле характерными. Хотя трудно провести точное сравнение ввиду отсутствия стандарта при отображении результатов расчетов в литературе, однако можно с уверенностью сказать, что метод линеаризации, или его ускоренные модификации, показывает результаты по количеству вычислений входящих в задачу функций, не худшие, чем любой другой метод.

ЛИТЕРАТУРА

1. Гарсия-Паломарес У.М., Мангасариан О.Л. (Garcia-Palomares U., Mangasarian O.) Superlinearly convergent quasi-newton methods for nonlinearly constrained optimization problems. – Math. Program., 1976, 11, p. 1 – 13.
2. Гросман К., Каплан А.А. Нелинейное программирование на основе безусловной оптимизации. – Новосибирск: Наука, 1981.
3. Данциг Дж.Б. Линейное программирование, его приложения и обобщения. – М.: Прогресс, 1966.
4. Даугавет В.А., Малоземов В.Н. Квадратичная скорость сходимости одного метода линеаризации для решения дискретных минимаксных задач. – ЖВМ и МФ, 1981, 21, № 4, с. 835 – 843.
5. Даугавет В.А. Модификация метода Вулфа. – ЖВМ и МФ, 1981, 21, № 2, с. 504 – 508.
6. Демьянов В.Ф., Малоземов В.Н. Введение в минимакс. – М.: Наука, 1972.
7. Иоффе А.Д., Тихомиров В.М. Теория экстремальных задач. – М.: Наука, 1974.
8. Кивиел К. (Kiwiel K.) A globally convergent quadratic approximation for inequality constraned minimax problems. – IIASA, 1981, CP-81-9, p. 1 – 24.
9. Кларк Ф. (Clark F.) A new approach to Lagrange multipliers. – Math. of operations research, 1976, 1, № 2, p. 165 – 174.
10. Кюнци Г., Крелле В. Нелинейное программирование. – М.: Сов. радио, 1965.
11. Муртаг Б.А., Саундерс М.А. (Murtag B., Saunders M.) Large Scale linearly constrained optimization. – Math. Program., 1978, 14, p. 41 – 72.
12. Островский А.М. Решение уравнений и систем уравнений. – М.: ИЛ, 1963.
13. Панин В.М. Метод линеаризации для задачи дискретного минимакса. – Кибернетика, 1980, № 3, с. 86 – 90.
14. Панин В.М. Метод линеаризации для задачи непрерывного минимакса. – Кибернетика, 1981, № 2, с. 75 – 78.
15. Панин В.М. О некоторых методах решения задач выпуклого программирования. – ЖВМ и МФ, 1981, 21, № 2, с. 315 – 328.
16. Панин В.М. Глобальная сходимость демпфированного метода Ньютона в задачах выпуклого программирования. – ДАН СССР, 1982, 261, № 4, с. 811 – 814.
17. Пауэлл М. (Powell M.) The convergence of variable matric methods for nonlinearly constrained optimization calculations. – Depart. of Appl. Math. and Theoret. Phys., 1977.
18. Пауэлл М. (Powell M.) Algorithms for nonlinear constraints that use Lagrangian functions. – Math. Program., 1978, 14, p. 224 – 248.
19. Полак Е. Численные методы оптимизации. Единый подход. – М.: Мир, 1974.
20. Поляк Б.Т., Третьяков Н.В. Об одном итерационном методе линейного программирования и его экономической интерпретации. – Эконом. и мат. методы 1972, VIII, № 5, с. 740 – 751.
21. Поляк Б.Т., Третьяков Н.В. Метод штрафных оценок для задач на условный экстремум. – ЖВМ и МФ, 1973, 13, № 1, с. 34 – 46.
22. Попков Н.В., Тулупчук Ю.М., Хилюк Л.Ф. Задача трехкритериального управления водохозяйственным комплексом днепровского водохранилища. – Автоматика, 1978, № 2, с. 44 – 53.

23. Пшеничный Б.Н. Необходимые условия экстремума. – М.: Наука, 1969.
24. Пшеничный Б.Н. Алгоритм для общей задачи математического программирования. – Кибернетика, 1970, № 5, с. 120–125.
25. Пшеничный Б.Н. Метод Ньютона для решения систем равенств и неравенств – Мат. заметки, 1970, 8, № 5, с. 635–640.
26. Пшеничный Б.Н. (Pshenishniy B.N.) Nonsmooth optimization and nonlinear programming.– In:Nonsmooth optimization. Pergamon Press, 1978.
27. Пшеничный Б.Н., Ганжела И.Ф. Алгоритм для решения задачи выпуклого программирования при линейных ограничениях. – Кибернетика, 1970, № 3, с. 81–85.
28. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. – М.: Наука, 1975.
29. Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи. – М.: Наука, 1980.
30. Пшеничный Б.Н., Соболенко Л.А. Ускорение сходимости метода линеаризации для задачи условной минимизации. – ЖВМ и МФ, 1980, 20, № 3, с. 605–614.
31. Робинсон С.М. (Robinson S.) A quadratical convergent algorithm for general nonlinear programming. – Problem Math. Program., 1972, 3, p. 145–156.
32. Робинсон С.М. (Robinson S.) Perturbed Kuhn – Tucker points and rates of convergence for a class of nonlinear programming algorithm. – Math. Program., 1974, 4, p. 1–16.
33. Рокафеллар Р.Т. Выпуклый анализ. – М.: Мир, 1973.
34. Рокафеллар Р.Т. (Rockafellar R.) Augmented Lagrangians and Applications of the Proximol Point Algorithm in convex programming. – Math. of operations Research, 1976, 1, № 2, p. 97–115.
35. Третьяков Н.В. Метод штрафных оценок для задач выпуклого программирования. – Эконом. и мат. методы, 1973, IX, № 3, с. 525–540.
36. Тьюарсон Р. Разреженные матрицы. – М.: Мир, 1977.
37. Федоренко Р.П. Приближенное решение задач оптимального управления. – М.: Наука, 1978.
38. Флетчер Р. (Fletcher R.) Second order corrections for nondifferentiable optimization. – Num. Analysis Report, 1981, 50, p. 1–30.
39. Хан С.П. (Han S.) Superlinearly convergent variable metric algorithms for general nonlinear programming problems. – Math. program., 1976, 11, p. 263–282.
40. Хан С.П. (Han S.) A global convergent method for nonlinear programming. – J. of optimization th. and Appl., 1977, 22, p. 297–309.
41. Хан С.П. (Han S.) Dual variable metric method for constrained optimization. – SIAM J. on Control and Optimiz., 1977, 15, p. 546–565.
42. Хан С.П. (Han S.) Variable metric method for minimizing a class of nondifferentiable functions. – Math. Program., 1981, 20, p. 1–13.
43. Хенкин Э.И., Волынский М.З. Поисковый алгоритм решения общей задачи математического программирования. – ЖВМ и МФ, 1976, 10, № 1, с. 61–71.
44. Хестенс М.Р. (Hesteines M.) Multiplier and Gradient Methods. – J. Optimiz. Th. Appl., 1969, 4, p. 303–320.
45. Химмельблау Д. Прикладное нелинейное программирование. – М.: Мир, 1975.
46. Численные методы условной оптимизации: Сб. статей / Под ред. Ф. Гилла, У. Мюррея. – М.: Мир, 1977.
47. Экланд И., Темам Р. Выпуклый анализ и вариационные проблемы. – М.: Мир, 1979.