

О НЕКОТОРЫХ ПОСЛЕДНИХ ДОСТИЖЕНИЯХ В ОПТИМИЗАЦИИ

Гольштейн Е. Г., Немировский А. С., Нестеров Ю. Е.

(Москва)

В статье приводится обзор некоторых новых направлений в теории выпуклой оптимизации, связанных с «нежесткими» постановками оптимизационных задач, декомпозицией и полиномиальными алгоритмами.

1. ВВЕДЕНИЕ

Математическое программирование служит основным аппаратом формализации и решения разнообразных оптимизационных задач, возникающих на практике, и в первую очередь — задач экономического происхождения. Численные методы оптимизации — один из наиболее популярных математических инструментов, используемых в экономическом моделировании, причем тенденции в развитии теории численной оптимизации во многом определяются потребностями экономических приложений. Так, резкое усложнение прикладных задач, среди которых нередко задачи с десятками тысяч переменных и тысячами ограничений, стимулировало особый интерес к вопросам, связанным с синтезом эффективных численных методов.

Цель настоящей статьи — описать некоторые достижения в этой области. Она ни в коей мере не претендует на полный обзор; отбор материала определяется вкусами авторов и их научными интересами. Мы, однако, берем на себя смелость утверждать, что обсуждаемая проблематика принадлежит к «множеству точек роста» математического программирования, хотя и не исчерпывает этого множества. Здесь рассматривается, во-первых, важная для практики проблематика, связанная с неточностью исходных данных, и развивается подход, позволяющий с помощью соответствующей модификации постановки обеспечить устойчивость решений задач с неточно известными данными; во-вторых, описывается новый, весьма общий декомпозиционный подход, позволяющий децентрализовать обмен информацией и собственно вычислительный процесс при решении больших задач; в-третьих, дается обзор достижений по синтезу теоретически эффективных (полиномиальных) численных методов выпуклого программирования, которые могут претендовать на роль «закрывающего звена» в схеме «постановка большой задачи — расчленение на подзадачи меньшего размера — численное решение подзадач», по которой решаются по-настоящему сложные прикладные задачи.

2. НЕЖЕСТКАЯ ПОСТАНОВКА ОПТИМИЗАЦИОННЫХ ЗАДАЧ

Совершенно естественно желание исследователей, использующих аппарат оптимизации для анализа практических задач, иметь дело с корректными постановками оптимизационных задач, т. е. с такими задачами, решение которых незначительно изменяется при малых изменениях исходных данных. Вместе с тем, как хорошо известно, многие важные классы оптимизационных задач (например, задачи линейного программирования) не обладают этим важным свойством. Можно ли освободиться от этого недостатка оптимизационного подхода и всегда иметь дело с корректно поставленными задачами? Один из возможных подходов к положительному решению этого вопроса связан с так называемой нежесткой постановкой оптимизационной задачи [1]. Остановимся на этом подходе применительно к линейному программированию.

Рассмотрим задачу

$$\langle c, x \rangle \rightarrow \max, Ax \leq b, x \geq 0, \quad (1)$$

и двойственную к ней

$$\langle b, y \rangle \rightarrow \min, yA \geq c, y \geq 0, \quad (2)$$

где c, x — n -векторы; b, y — m -векторы; A — $m \times n$ -матрица. Поскольку в дальнейшем нас будет интересовать зависимость этих задач лишь от векторов b и c , составляющих вместе вектор $d = (c, b)$, то введем для (1) и (2) обозначения $\mathcal{P}(d)$ и $\tilde{\mathcal{P}}(d)$. Пусть $X^*(d)$ и $Y^*(d)$ множества оптимальных точек (решений) задач $\mathcal{P}(d)$ и $\tilde{\mathcal{P}}(d)$ соответственно, $Z^*(d) = X^*(d) \times Y^*(d)$. Если множество $Z^*(d^0)$ состоит более чем из одной точки, то при любом $\delta > 0$ не существует такой непрерывной вектор-функции $z^*(d)$, $|d - d^0| \leq \delta$, что $z^*(d) \in Z^*(d)$ при всяком d из шара $\{d: |d - d^0| \leq \delta\}$.

Положение меняется при переходе от (1), (2) к их нежестким аналогам. Мотивация введения таких аналогов основана на следующих соображениях. На практике целевой вектор c и вектор ресурсов b , как правило, бывают известны лишь с определенной степенью точности. Пусть $\varepsilon > 0$ измеряет точность задания вектора d ; имея вектор d , мы располагаем лишь информацией о том, что его истинное значение $\bar{d}$ удовлетворяет неравенству $|d - \bar{d}| \leq \varepsilon$. В этих предположениях вполне допустимо под решением задач (1) и (2) понимать любую точку $z^*(d') = (x^*(d'), y^*(d')) \in Z^*(d')$, где $|d - d'| \leq \varepsilon$. Учитывая это, введем задачи $\mathcal{P}_\varepsilon(d)$ и $\tilde{\mathcal{P}}_\varepsilon(d)$, множества $X_\varepsilon^*(d)$ и $Y_\varepsilon^*(d)$ решений которых задаются соответственно соотношениями

$$X_\varepsilon^*(d) = \bigcup_{|d - d'| \leq \varepsilon} X^*(d'), \quad Y_\varepsilon^*(d) = \bigcup_{|d - d'| \leq \varepsilon} Y^*(d');$$

задачи $\mathcal{P}_\varepsilon(d)$ и $\tilde{\mathcal{P}}_\varepsilon(d)$ естественно назвать нежесткими аналогами задач $\mathcal{P}(d)$ и $\tilde{\mathcal{P}}(d)$.

В отличие от задач $\mathcal{P}(d)$ и $\tilde{\mathcal{P}}(d)$ их нежесткие аналоги всегда допускают построение решений, непрерывно зависящих от векторного параметра d . Один из способов формирования подобных решений (естественно, далеко не единственный) основан на регуляризации функции Лагранжа $L(x, y) = \langle c, x \rangle - yAx + \langle b, y \rangle$ пары двойственных задач (1), (2). Задавшись параметром $\gamma > 0$, введем регуляцию L_γ функции L , положив

$$L_\gamma(x, y) = L(x, y) - \frac{1}{2} \gamma x^2 + \frac{1}{2} \gamma y^2. \quad (3)$$

Функция (3) порождает пару двойственных задач выпуклого программирования вида

$$\langle c, x \rangle - \frac{1}{2} \gamma x^2 - \frac{1}{2\gamma} |Ax - b|_+^2 \rightarrow \max, \quad x \geq 0, \quad (4)$$

$$\langle b, y \rangle + \frac{1}{2} \gamma y^2 + \frac{1}{2\gamma} |yA - c|_+^2 \rightarrow \min, \quad y \geq 0, \quad (5)$$

являющихся, очевидно, результатом одновременного применения к задачам (1), (2) методов штрафных функций с параметром штрафа $1/2\gamma$ и метода регуляризации с параметром $\gamma/2$.

Обозначим через $z_\gamma(d) = (x_\gamma(d), y_\gamma(d))$ единственную седловую точку функции L_γ , составляющие $x_\gamma(d)$ и $y_\gamma(d)$ которой — решения задач (4) и (5) соответственно. Связь между задачами $\mathcal{P}_\varepsilon(d)$, $\tilde{\mathcal{P}}_\varepsilon(d)$ и парой задач (4), (5) определяется следующим простым предложением: для произвольных $\varepsilon > 0$ и компакта D , принадлежащего множеству разрешимости задач (1), (2), найдется такой параметр $\gamma = \gamma(\varepsilon)$, что $z_\gamma(d) \in Z_\varepsilon^*(d) = X_\varepsilon^*(d) Y_\varepsilon^*(d)$, $d \in D$, т. е. $z_\gamma(d)$ при любом $d \in D$ составлено из решений $x_\gamma(d)$ и $y_\gamma(d)$ нежестких аналогов задач $\mathcal{P}(d)$ и $\tilde{\mathcal{P}}(d)$, определяемых точностью ε .

Основное свойство вектор-функции $z_\gamma(d)$, оправдывающее ее выбор в качестве решения задач $\mathcal{P}_\varepsilon(d)$, $\tilde{\mathcal{P}}_\varepsilon(d)$, таково: для любых d^1, d^2 имеет

место неравенство

$$\langle \bar{z}_\gamma(d^1) - \bar{z}_\gamma(d^2), d^1 - d^2 \rangle \geq \gamma |\bar{z}_\gamma(d^1) - \bar{z}_\gamma(d^2)|^2, \quad (6)$$

где $\bar{z}_\gamma(d) = (x_\gamma(d), -y_\gamma(d))$.

Из (6), в частности, вытекает непрерывность вектор-функции $z_\gamma(d)$, которая удовлетворяет условию Липшица с постоянной $1/\gamma$: $|z_\gamma(d^1) - z_\gamma(d^2)| \leq \gamma^{-1} |d^1 - d^2|$.

Таким образом, при переходе от жестких постановок оптимизационных задач к их нежестким аналогам эти задачи приобретают важное для практических приложений свойство корректности.

3. ДЕКОМПОЗИЦИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

Неравенство (6) не только гарантирует непрерывность вектор-функции $z_\gamma(d)$, но и обеспечивает возможность реализации различных процессов управления решениями задач (3), (4) за счет воздействий на векторы c и b . В частности, это неравенство лежит в основе доказательства сходимости итеративных декомпозиционных алгоритмов, связанных с последовательным пересчетом ресурсных и целевых векторов отдельных блоков, на которые разбита оптимизационная задача.

Для иллюстрации соответствующего декомпозиционного подхода ограничимся линейным программированием и способом разбиения исходной задачи (1) на подзадачи, определяемым представлением матрицы A в виде совокупности прямоугольных подматриц. Разобьем множество $I = \{1, \dots, m\}$ номеров ограничений задачи (1) на непересекающиеся подмножества $I(q)$, $q=1, \dots, Q$. Аналогично представим множество $J = \{1, \dots, n\}$ переменных этой задачи в виде совокупности непересекающихся подмножеств $J(r)$, $r=1, \dots, R$. Обозначим через $b(q)$ и $c(r)$ векторы, составленные из компонент b_i , $i \in I(q)$, и c_j , $j \in J(r)$, соответственно, а под $A(q, r)$ будем понимать матрицу с элементами $a_{i,j}$, где $i \in I(q)$, $j \in J(r)$.

Декомпозиционный подход к решению нежестких аналогов задач (1), (2) состоит в следующем. Сначала по заданному $\varepsilon > 0$ подбирают такой параметр $\gamma > 0$, что решения задач (4), (5) являются одновременно решениями задач $\mathcal{P}_\varepsilon(d)$, $\tilde{\mathcal{P}}_\varepsilon(d)$. Далее используется итеративный декомпозиционный процесс решения задач (4), (5) с выбранным параметром γ . Перед началом очередного (скажем, k -го) шага этого процесса мы располагаем представлениями целевого вектора c и вектора ограничений b

$$\sum_{q=1}^Q c^{qk} = c, \quad \sum_{r=1}^R b^{rk} = b, \quad (7)$$

с помощью которых образуется QR задач-блоков вида

$$\begin{aligned} & \langle c^{qk}(r), x(r) \rangle - \frac{1}{2Q} \gamma (x(r))^2 - \\ & - \frac{R}{2\gamma} |A(q, r)x(r) - b^{rk}(q)|_+^2 \rightarrow \max, \quad x(r) \geq 0, \end{aligned} \quad (8)$$

где $q=1, \dots, Q$, $r=1, \dots, R$. Заметим, что задача (8) связана с нежестким аналогом задачи

$$\langle c^{qk}(r), x(r) \rangle \rightarrow \max, \quad A(q, r)x(r) \leq b^{rk}(q), \quad x(r) \geq 0.$$

Основная вычислительная работа k -го шага связана с независимым и приближенным решением QR задач (8). Пусть $x^{qk}(r)$ — приближенное с точностью $\sigma_k \geq 0$ (по функционалу) решение задачи (8), а $y^{rk}(q) = \frac{R}{\gamma} |A(q, r)x^{qk}(r) - b^{rk}(q)|_+$. Векторы $x^{qk}(r)$ и $y^{rk}(q)$ используются для пересчета представлений (7) с помощью формул

$$c^{q, k+1} = c^{qk} - \frac{\gamma}{Q} (x^{qk} - x^k), \quad b^{r, k+1} = b^{rk} + \frac{\gamma}{R} (y^{rk} - y^k), \quad (9)$$

где

$$x^k = Q^{-1} \sum_{q=1}^Q x^{qk}, \quad y^k = R^{-1} \sum_{r=1}^R y^{rk}. \quad (10)$$

В предположении достаточно быстрого стремления к нулю точности решения задач (8) $\left(\sum_{k=1}^{\infty} \sigma_k^{1/2} < \infty \right)$ последовательности $\{x^k\}$ и $\{y^k\}$ сходятся к решениям задачи (4) и (5), или, что то же самое, к решениям задач $\mathcal{P}_\varepsilon(d)$ и $\tilde{\mathcal{P}}_\varepsilon(d)$.

При построении регуляризации (3) функции Лагранжа можно вместо x^2 и y^2 использовать функции $(x-u)^2$ и $(y-v)^2$, вектор $w = (u, v)$ при этом естественно называть центром регуляризации.

Если описанный выше декомпозиционный процесс (9), (10) решения задач $\mathcal{P}_\varepsilon(d)$ и $\tilde{\mathcal{P}}_\varepsilon(d)$, в котором параметр γ необходимо выбирать тем меньшим, чем выше точность ε , дополнить соответствующим пересчетом центра регуляризации, то он может быть использован для решения задач (1) и (2). В этом случае параметр $\gamma > 0$ можно выбирать произвольным образом, а векторы u и v , пересчитываемые от шага к шагу, естественно интерпретировать как прогнозы решений задач (1) и (2) (например, в качестве u и v на k -м шаге можно использовать x^k и y^k из (10)). Для любой разрешимой задачи линейного программирования (1) модифицированный процесс вырабатывает последовательности, сходящиеся к некоторым решениям задач (1) и (2).

Изложение декомпозиционного метода решения оптимизационных задач в нежесткой постановке содержится в [1], его аналог для жестких постановок описан в [2]. Декомпозиционный подход допускает более общие, чем описанные выше, способы расчленения задачи на подзадачи [3].

4. ПОЛИНОМИАЛЬНЫЕ АЛГОРИТМЫ И ПОЛИНОМИАЛЬНО РАЗРЕШИМЫЕ ЗАДАЧИ

Полиномиальность некоторого алгоритма и полиномиальная разрешимость некоторой массовой задачи — ключевые понятия теории алгоритмической сложности вычислений. Содержательно эти понятия можно описать так. «Массовая задача» — это семейство задач некоторой фиксированной структуры; каждая индивидуальная задача выделяется в этом семействе конечным набором числовых параметров — «коэффициентов». Например, массовой задачей является общая задача линейного программирования $c^T x \rightarrow \min | x \in R^n, Ax \leq b$ (A — $m \times n$ -матрица); идентифицирующие параметры здесь — это n , m и коэффициенты матрицы условий A , вектора правых частей b и целевого функционала c . Другими примерами является задача квадратичного программирования при линейных ограничениях, задача квадратичного программирования при квадратичных ограничениях, задача геометрического программирования и т. п. Будем рассматривать лишь такие массовые задачи, у которых, как и в перечисленных примерах, решения индивидуальных представителей естественным образом отождествляются с конечномерными векторами. С теоретической точки зрения полезно ограничиться задачами, допускающими «вполне финитное» описание, для чего на идентифицирующие параметры накладывается условие рациональности. При этом набор идентифицирующих индивидуальную задачу параметров становится конечным словом некоторого алфавита; длина L этого слова («общее число битов, необходимое для записи всех коэффициентов задачи») называется *размером* индивидуальной задачи. Теперь наметим определение понятия «полиномиальный алгоритм». Для нужд последующего изложения достаточно следующего описания: *строго полиномиальный* алгоритм решения массовой задачи — это алгоритм, перерабатывающий каждый набор идентифицирующих индивидуальную задачу ко-

эффициентов в набор координат ее решения с помощью конечного числа M арифметических операций над коэффициентами; при этом M может зависеть от решаемой задачи, но не должно превосходить некоторого многочлена $p(L)$ от размера задачи. Что касается арифметических операций, то алгоритму не разрешается выполнять такие операции абсолютно точно, но считается возможным «заказать» не более чем полиномиальное по L число $q(L)$ знаков до и после запятой, удерживаемое при всех вычислениях. Массовая задача называется полиномиально разрешимой, если она допускает строго полиномиальный разрешающий алгоритм.

Согласно установившейся в теории сложности вычислений приблизительно четверть века назад традиции (мотивировка которой увела бы нас слишком далеко от предмета последующего изложения), полиномиальность алгоритма считается теоретическим эквивалентом его «вычислительной эффективности», а полиномиальная разрешимость массовой задачи — синонимом ее «эффективной» разрешимости.

На момент формирования понятия «полиномиальная разрешимость» это свойство было установлено у весьма небольшого числа массовых задач, рассматриваемых в математическом программировании (наиболее общая из них — это задача о максимальном потоке минимальной стоимости). Так, уже вопрос о полиномиальной разрешимости общей задачи линейного программирования был открыт. Как известно, на практике задачи ЛП успешно решают симплекс-методом, этот последний, как было выяснено давно, не является полиномиальным: для всех известных версий симплекс-метода были построены примеры серий задач ЛП с целыми коэффициентами порядка 1 и растущими m и n , на которых число итераций метода росло как экспонента от $(m+n)$. Первый полиномиальный алгоритм для решения задач ЛП — алгоритм эллипсоидов — был предложен лишь в 1979 г. Л. Г. Хачияном [4].

5. ПОЛИНОМИАЛЬНЫЕ АЛГОРИТМЫ В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

1. Алгоритм эллипсоидов. В основе алгоритма эллипсоидов решения общей задачи линейного программирования лежал впервые предложенный в 1976 г. в [5] и независимо переоткрытый в [6] метод эллипсоидов решения негладких выпуклых задач, который в своем базовом варианте предназначен для приближенного решения задачи

$$f(x) \rightarrow \min_{x \in R^n, \|x\| \leq R}, \quad (11)$$

порожденной (произвольной) выпуклой функцией f ; при любом $\varepsilon \in (0, 1)$ метод эллипсоидов находит ε -решение указанной задачи, т. е. точку x_ε , $\|x_\varepsilon\| \leq R$, такую, что $f(x_\varepsilon) - \min\{f(x) \mid \|x\| \leq R\} \leq \varepsilon \mathcal{V}_R(f)$, где $\mathcal{V}_R(f) = \max\{f(x) \mid \|x\| \leq R\} - \min\{f(x) \mid \|x\| \leq R\}$ — «размах» функции f на допустимом множестве, за не более чем полиномиальное по n и $\ln 1/\varepsilon$ число итераций (конкретно, за $O(n^2 \ln(1/\varepsilon))$ итераций), на каждой из которых требуется однократно вычислить значение f и опорный к f функционал в некоторой рекуррентно строящейся точке и, сверх этого, осуществить не более чем полином по n (конкретно, $O(n^2)$) операций точной вещественной арифметики. Л. Г. Хачиян заметил, что этого достаточно для построения строго полиномиального алгоритма решения задачи ЛП. Он показал, что по задаче $\mathcal{P}$ линейного программирования с n переменными и $m > n$ ограничениями-неравенствами, имеющей длину входа L , можно построить выпуклую задачу $\mathcal{F}$ вида (11) с $R = 2^{O(L)}$, такую, что по $2^{-O(L)}$ -решению $\mathcal{F}$ можно выяснить совместность $\mathcal{P}$ и, в случае совместности, построить за $O(m^2 n)$ операций точное решение $\mathcal{P}$. Отыскание $2^{-O(L)}$ -решения $\mathcal{F}$ методом эллипсоидов требует $O(n^2 L)$ итераций; трудоемкость каждой из них имеет порядок $O(n^2) + M$ операций, где M — число операций, требуемых для однократного вычисления значения и субградиента целевого функционала задачи $\mathcal{F}$; в данном контексте, как оказывается, $M = O(mn)$. Таким образом, для решения задачи $\mathcal{P}$ достаточно $O(n^3 mL)$ операций точной вещественной ариф-

метики. Последний шаг состоит в том, чтобы показать, что точные вычисления можно заменить приближенными с удерживанием $O(nL)$ знаков до и после запятой.

Обратим внимание на некоторые особенности описанной схемы. Прежде всего, ядром алгоритма эллипсоидов послужил «бесконечношаговый» алгоритм выпуклой оптимизации, основанный на точной вещественной арифметике; единственное свойство «ядра», по существу использованное в алгоритме Хачияна, — это полиномиальная (по n , m и $\ln 1/\epsilon$) оценка числа итераций, необходимых для получения ϵ -решения, при полиномиальной по тем же параметрам арифметической стоимости каждой итерации. Точно так же обстоит дело и со всеми остальными известными к настоящему моменту строго полиномиальными алгоритмами (а также построены не только для задач ЛП, но и для задач выпуклой квадратичной минимизации при линейных ограничениях). Каждый из таких алгоритмов сводит точное решение исходной задачи к отысканию ϵ -решения (при подходе $\epsilon > 0$) некоторой другой выпуклой задачи; это последнее отыскивается приближенным методом, арифметическая стоимость которого (в операциях точной вещественной арифметики) полиномиальна по размерности (числу переменных n и ограничений m) исходной задачи и по величине $\ln 1/\epsilon$ — логарифму требуемой точности. При этом по-настоящему нетривиальным является лишь вопрос о форме задачи, к которой редуцируется исходная, и о построении приближенного метода решения этой новой задачи, позволяющего получить — при любом $\epsilon > 0$ — ее ϵ -решение за не более чем полином от размерности исходной задачи и $\ln 1/\epsilon$ операций точной вещественной арифметики. Что же касается проблем, связанных с превращением приближенного метода решения новой задачи в точный метод решения исходной (выбор подходящего ϵ , способ пересчета ϵ -решения новой задачи в точное решение исходной, замена точных арифметических операций приближенными), то они во всех известных случаях решаются точно так же, как и в алгоритме эллипсоидов, и представляют теперь некоторое «общее место» технического характера.

Представляется целесообразным поэтому несколько расширить понятие полиномиального алгоритма и понимать под ним итеративную процедуру решения некоторой массовой задачи выпуклого программирования, отыскивающую приближенное ϵ -решение (при любом положительном ϵ) любого индивидуального представителя данной массовой задачи с арифметической стоимостью, мажорируемой полиномом от $\ln 1/\epsilon$ и «размерности» решаемой задачи (сказанное не претендует на роль формального определения; в некоторых случаях, в частности, удобно слегка модифицировать определения ϵ -решений и размерности).

2. Метод Кармаркара. После доказательства полиномиальной разрешимости задачи линейного программирования с помощью метода эллипсоидов появилось несколько других приближенных методов, с помощью которых можно получить тот же результат. Однако теперь целью является не только улучшение теоретических оценок, но и построение методов, способных на практике конкурировать с симплекс-методом. Метод эллипсоидов для этого не годится: дело в том, что «экспоненциально сложные» для симплекс-метода задачи, как показывает вычислительный опыт, на практике не встречаются; общепринятая эмпирическая оценка числа итераций симплекс-метода линейна по размерности, что приводит к кубической по размерности «экспериментально наблюдаемой» арифметической стоимости этого метода. В методе эллипсоидов как теоретическая, так, похоже, и эмпирическая зависимость, имеют порядок 4. Поэтому на задачах большого размера метод эллипсоидов безнадежно отстает от симплексного.

Большие надежды связаны с предложенным в 1984 г. Н. Кармаркаром [7] новым полиномиальным алгоритмом решения линейных задач, к описанию которого мы и перейдем.

Метод Кармаркара непосредственно применим к задачам линейного программирования в следующей форме

$$\langle c, x \rangle \rightarrow \min | Ax = 0, x \in \Delta, \quad (12)$$

где Δ — стандартный симплекс $\{x \geq 0, \langle e_n, x \rangle = n^{-1}\}$, $e_n = (n^{-1}, \dots, n^{-1})^T$; A — $m \times n$ -матрица. При этом предполагается, что

$$Ae_n = 0, \quad (13)$$

$$\langle c, x^* \rangle = 0 \quad (14)$$

(x^* — решение задачи). Отметим, что при наличии априорных оценок на норму прямого и двойственного решения любая задача линейного программирования может быть сведена к виду (12) при условиях (13), (14).

Метод работает следующим образом. К i -му шагу имеется очередная допустимая точка $x_{i-1} > 0$ ($x_0 = e_n$; эта точка допустима в силу (13)). Новое приближение x_i определяется следующим образом. Пусть $D_{i-1} = \text{diag}\{x_{i-1}\}$; вычислим ортогональную проекцию c_{i-1} вектора $D_{i-1}c$ на линейное многообразие $\{AD_{i-1}u = 0, e_n^T u = n^{-1}\}$, и пусть $y_i = e_n - \gamma_i \|c_{i-1}\|_2^{-1} \cdot c_{i-1}$ (о выборе положительного шагового множителя γ_i будет сказано ниже; пока заметим лишь, что этот выбор обеспечивает положительность y_i). Тогда $x_i = (n \langle y_i, D_{i-1}e_n \rangle)^{-1} D_{i-1}y_i$ (легко понять, что x_i — положительная допустимая точка задачи). Отметим, что вычисление x_i требует порядка $m^2 n$ арифметических операций.

В базовом варианте алгоритма γ_i можно выбирать фиксированным (зависящим только от n); подходящий выбор такого рода влечет оценки

$$c^T x_i \leq (\exp\{1\}/2)^{-i/n} c^T e_n. \quad (15)$$

Можно также выбирать γ_i из условия минимума функции Кармаркара $K_i(\gamma) = \langle D_{i-1}c, e_n - \gamma c_{i-1} \rangle / V(e_n - \gamma c_{i-1})$, где $V(x) = \prod_{i=1}^n x^i$, на множестве положительных γ , для которых $e_n > \gamma c_{i-1}$. Для этой «версии с большим шагом» также действует оценка (15).

Оценка (15) показывает, что обеспечение относительной погрешности $\varepsilon_i \leq \varepsilon$ требует порядка $n \ln 1/\varepsilon$ итераций по $O(m^2 n)$ арифметических операций каждая. Таким образом, теоретически базовая версия метода Кармаркара не лучше, чем метод эллипсоидов.

Н. Кармаркар показал, что, используя некоторое модифицированное приближенное проектирование и надлежащим образом организовав вычисления, можно уменьшить общее число операций до величины порядка $m^2 n^{1.5} L$ (приемы такого рода в дальнейшем, не вдаваясь в детали их реализации, будем называть кармаркаровским ускорением; подробности можно найти в [7–10]). Тем самым Н. Кармаркару удалось добиться на полпорядка лучшей, чем у метода эллипсоидов, гарантированной оценки трудоемкости.

Если бы алгоритм Кармаркара работал на реальных задачах не лучше, чем по своей теоретической оценке, надежды на его успешную конкуренцию симплекс-методу вряд ли продержались сколько-нибудь заметное время. Однако оказалось, что этот алгоритм (по крайней мере в версии с большим шагом) работает существенно лучше, чем можно судить по гарантированной оценке наихудшего случая. Сам Н. Кармаркар анонсировал существенное превосходство своего метода перед симплексным; имеются сообщения, согласно которым фактическое число итераций метода при заданной точности растет с ростом размерности не пропорционально n , а гораздо медленнее — как $\ln n$. В пользу этого можно привести и некоторые правдоподобные рассуждения. Дело в том, что для «версии с большим шагом» более тщательный анализ (см. [11]) дает оценку (для простоты ограничимся методом без ускорения)

$$\varepsilon_i \leq \exp\{-n^{-1}(1 - \ln 2) \sum_{j=1}^i \|\xi_j\|_2^2 \|\xi_j\|_\infty^{-2}\}, \quad (16)$$

в которой ξ_i — некоторые n -мерные векторы (определяемые данными задачи); очевидно, всегда $\|\xi_j\|_2^2 \|\xi_j\|_\infty^{-2} \geq 1$, и оценка (15) отвечает самому неблагоприятному с точки зрения скорости сходимости случаю $\|\xi_j\|_2^2 /$

$\|\xi_j\|_\infty^2 = 1$ при всех j . Между тем «типичное» значение указанного отношения (отвечающее, скажем, случайному равномерно распределенному на единичной сфере направлению ξ) равно не 1, а величине порядка $n \ln^{-1} n$, так что, если бы ξ_j были случайны, (16) действительно дало бы $\epsilon_i \leq \exp \{-0(i/\ln n)\}$.

Вопрос о практической эффективности алгоритма Кармаркара и его преимуществах в этом отношении перед симплекс-методом еще не решен; тем не менее, появление этого алгоритма весьма сильно стимулировало работу в области синтеза новых полиномиальных алгоритмов для линейного программирования.

С практической точки зрения некоторым недостатком конструкции Кармаркара являются условия (13), (14); имеется много работ, в которых алгоритм модифицируется так, чтобы избавиться от одного или обоих этих допущений. Например, в [11] описан использующий идеи метода Кармаркара алгоритм решения задачи ЛП в форме (12), свободный от ограничений (13)—(14). Оценка скорости сходимости алгоритма такова. Пусть множество M_0 допустимых планов задачи пересекается с относительной внутренностью симплекса, и пусть

$$z^* = \operatorname{argmax} \{V(x) \mid x \in M_0\},$$

так что величина $\theta^* = [V(z^*)/V(e_n)]^{1/n}$ положительна. Тогда для некоторой последовательности n -мерных векторов $\xi_i \neq 0$ при всех i справедлива импликация (ср. (16))

$$\begin{aligned} v_i &\equiv \exp \left\{ -n^{-1} (1 - \ln 2) \sum_{j=1}^i \|\xi_j\|_2^2 / \|\xi_j\|_\infty^2 \right\} < \theta^* \Rightarrow \\ &\Rightarrow (c^T x_i - c^*) \leq (\theta^*)^{-1} v_i (c^T z^* - c^*), \end{aligned}$$

где x_i — приближение, сформированное методом после i итераций (оно является допустимым планом задачи); c^* — оптимальное значение целевого функционала. Трудоемкость итерации такая же, как и в базовой версии алгоритма Кармаркара; сохраняются и возможности кармаркаровского ускорения.

6. МЕТОДЫ ОТСЛЕЖИВАНИЯ ТРАЕКТОРИИ

1. Предварительные замечания. В методе эллипсоидов число итераций, необходимое для получения ϵ -решения, пропорционально квадрату размерности, а в методе Кармаркара — самой размерности. В самое последнее время предложены (впервые — Дж. Ренегаром, 1897 [12]) и интенсивно разрабатываются методы, в которых указанная величина пропорциональна квадратному корню из размерности. Такие методы, получившие в зарубежной литературе название «path-following» («методы отслеживания траектории»), в отличие от предшествующих им полиномиальных алгоритмов внешне выглядят вполне традиционно — они представляют собой методы внутренних штрафных функций, использующие (отнюдь не впервые) барьеры логарифмического типа (отсюда — другое название методов: «методы внутренней точки»). Способы обоснования этих методов, однако, отнюдь не традиционны и основаны на эксплуатации не замечавшихся ранее свойств логарифмических барьеров.

В этом разделе мы, придерживаясь намеченного в [13, 14] и развитого в [15—18] общего подхода, на содержательном уровне опишем идеи, лежащие в основе полиномиальных методов внутренней точки решения линейных и ряда других массовых задач выпуклого программирования.

2. Общая схема «path-following» методов. Одна из вполне традиционных схем решения экстремальной задачи

$$f(x) \rightarrow \min \mid x \in G \subset R^n \quad (17)$$

такова. Задаче сопоставляется семейство функций $F_t(x)$, зависящее от меняющегося в интервале Δ вещественного параметра t , такое, что траектория минимумов $x^*(t) = \operatorname{argmin} F_t(x)$ сходится к решению x^* задачи,

когда t сходится к надлежащему концу t^* интервала Δ . Семейство строится так, чтобы для некоторого значения t_0 параметра точка $x^*(t_0)$ была известна. После этого «отслеживают траекторию $x^*(t)$ » вдоль некоторой последовательности $\{t_i\}$ значений параметра, начинающейся в t_0 и сходящейся к t^* : строят аппроксимации $x(t_i)$ точек $x^*(t_i)$ именно так. Имея хорошее приближение $x(t_{i-1})$ к $x^*(t_{i-1})$, изменяют параметр ($t_{i-1} \rightarrow t_i$), так, чтобы $x(t_{i-1})$ осталось не слишком плохим приближением к новой точке $x^*(t_i)$ отслеживаемой траектории; после этого, делая из $x(t_{i-1})$ один или несколько шагов подходящего итеративного метода решения задачи $F_{t_i}(\cdot) \rightarrow \min$, получают из «не слишком плохого» приближения $x(t_{i-1})$ к точке $x^*(t_i)$ «хорошее» приближение $x(t_i)$. Одной из наиболее популярных версий такого подхода является метод барьеров, порождаемый внутренней штрафной функцией — барьером — F для G (гладкой на $\text{int } G$ функцией, уходящей на бесконечность при подходе к ∂G изнутри области); соответствующее семейство есть $F_t(x) = t f(x) + F(x)$, причем здесь $\Delta = (0, \infty)$, $t^* = +\infty$. Другая версия — это метод центров, который нам потребуется в следующей форме: $F_t(x) = \xi \ln(1/(t - f(x))) + F(x)$, где F — как и выше, барьер для G , ξ — положительная постоянная; здесь $\Delta = (f^*, +\infty)$, $t^* = f^*$ (f^* — оптимальное значение в (17)).

В «path — following» методах (далее — PF-методы) семейства $F_t(x)$ состоят из выпуклых (несобственных) функций x , гладких и обладающих невырожденным гессианом по x на $\text{Dom } F_t(\cdot)$. В качестве процедуры $x(t_{i-1}) \rightarrow x(t_i)$ «возвращения к отслеживаемой траектории» используется метод Ньютона. Последнее требует, в частности, строить семейства из функций, на которых метод Ньютона «хорошо себя ведет», и мы начнем с описания таких функций.

3. Самосогласованные функции и метод Ньютона. Традиционные результаты о методе Ньютона устанавливают его квадратичную сходимость в окрестности минимума, если только гессиан Φ'' минимизируемой функции Φ локально липшицев и невырожден. Заметим, что и градиент, и гессиан — «метрические» понятия: они определяются не только самой функцией, но и «внешней» для функции евклидовой структурой, которой снабжено пространство переменных. Хотя сам факт невырожденности и липшицевости гессиана не зависит от выбора евклидовой структуры, его числовые характеристики — и, следовательно, описание области, в которой имеет место квадратичная сходимость, — от этой структуры зависят, в то время как траектория метода Ньютона, как известно, от этой структуры не зависит. Получается странная картина: сам метод инвариантен по отношению к евклидовой структуре пространства переменных, а описание его свойств (например, окрестности минимума, в которой метод сходится квадратично) «привязано» к совершенно произвольной евклидовой структуре. Можно, однако, заметить, что сама Φ задает локальную евклидову структуру (своим вторым дифференциалом $D^2\Phi(x)[\cdot, \cdot]$), и можно потребовать липшицевости второй производной Φ в этой структуре (т. е. выполнения неравенства вида $|D^3\Phi(x) \cdot [h, q, q]| \leq L(x) (D^2\Phi(x)[h, h])^{1/2} D^2\Phi(x)[q, q] \forall q, h \in R^n$; $L(x)$ и есть локальная константа Липшица второй производной в заданной ею локальной евклидовой структуре). С этой точки зрения самое сильное предположение — это «липшицевость с фиксированной постоянной» — $L(x) \equiv L$. Удовлетворяющие этому предположению функции будем называть самосогласованными. Разумеется, нормировкой Φ можно сделать L наперед заданным числом; удобно считать $L=2$. Теперь перейдем от наводящих соображений к точному определению.

Определение 1. Пусть H — открытое выпуклое подмножество R^n и $\Phi \in C^3$ — гладкая выпуклая функция на H . Φ называется стандартно самосогласованной на H , если для любых $x \in H, h \in R^n$

$$|D^3\Phi(x)[h, h, h]| \leq 2\{D^2\Phi(x)[h, h]\}^{3/2}. \quad (18)$$

Полезно заметить, что конечная сумма стандартно самосогласованных на областях H_i функций Φ_i стандартно самосогласована на $\bigcup_i H_i$; функция $\Phi^*(y) = \Phi(A(y))$, получающаяся из стандартно самосогласо-

ванной на H функции Φ аффинной заменой переменных $x=A(y)$, $A: R^h \rightarrow R^n$, стандартно самосогласована на $H^*=A^{-1}(H)$.

Поведение метода Ньютона на стандартно самосогласованных функциях удобно описывать в терминах ньютоновских приращений точек траектории метода. Пусть Φ стандартно самосогласована на H и $x \in H$. Ньютоновским приращением $\lambda(x, \Phi)$ функции Φ в точке x будем называть норму производной $D\Phi(x)$ в порожденной Φ локальной метрике

$$\lambda(x, \Phi) = \sup \{ |D\Phi(x)[h]| \mid h \in R^n, D^2\Phi(x)[h, h] \leq 1 \}$$

(возможно, что $D^2\Phi(x)$ вырождена, а $D\Phi(x)$ имеет ненулевое сужение на нулевое подпространство этой формы); в этом случае, по определению, $\lambda(x, \Phi) = \infty$. Можно доказать, что либо $\lambda(x, \Phi) = \infty$ при всех $x \in H$, либо $\lambda(x, \Phi)$ конечна и непрерывна всюду на H). В случае, когда $D^2\Phi$ невырождено в некоторой (а тогда, как оказывается, и во всякой) точке, ньютоновское приращение можно записать в виде

$$\lambda^2(x, \Phi) = (\Phi'(x))^T (\Phi''(x))^{-1} \Phi'(x),$$

где Φ' , Φ'' — градиент и гессиан Φ относительно (какой угодно) евклидовой структуры R^n . Мотивировка названия «ньютоновское приращение» связана с тем, что $\lambda^2(x, \Phi)/2$ есть в точности величина, на которую (в квадратичном относительно x приближении) убывает Φ на одном шаге метода Ньютона.

Основной для дальнейшего результат таков.

Предложение 1. Пусть Φ стандартно самосогласована на H и имеет H своей естественной областью определения (т. е. стремится к ∞ вдоль любой последовательности точек H , сходящейся к граничной точке H). Функция Φ достигает на H минимума тогда и только тогда, когда множество $\{x \mid \lambda(x, \Phi) < 1\}$ непусто. Далее, если $x_0 \in H$ таково, что $\lambda(x_0, \Phi) < \lambda_* \equiv 2-3^{1/2}$, то траектория стартующего из x_0 метода Ньютона

$$x_{i+1} \in \text{Arg min} \{ D\Phi(x_i)[y - x_i] + \frac{1}{2} D^2\Phi(x_i)[y - x_i, y - x_i] \mid y \in R^n \}$$

корректно определена, лежит в H и удовлетворяет соотношению

$$\lambda_{i+1} \equiv \lambda(x_{i+1}, \Phi) \leq \lambda_i^2 / (1 - \lambda_i)^2 \leq \lambda_i / 2.$$

Кроме того, при некоторой абсолютной постоянной C для всех i

$$\Phi(x_i) - \min_H \Phi \leq C \lambda_i^2.$$

Мы видим, что и «область квадратичной сходимости» метода Ньютона на стандартно самосогласованных функциях, и поведение метода в этой области допускают эффективное описание в терминах ньютоновских приращений.

4. Самосогласованные барьеры. Метод барьеров. Семейства, с которыми работают все известные полиномиальные РF-методы, образованы самосогласованными (после несущественной нормировки — даже стандартно самосогласованными) функциями. Для обоснования полиномиальности этого, однако, недостаточно — на каждом шаге РF-метода мы имеем дело со своей функцией семейства, и важно, как эти функции согласованы между собой. Можно ввести несложно формулируемое и легко проверяемое понятие «самосогласованного семейства» и построить на его основе достаточно прозрачную единую теорию [15], объясняющую предложенные в литературе (и некоторые новые) полиномиальные РF-методы, но мы не будем здесь этого делать. Ограничимся рассмотрением конкретного — барьерного — семейства $F_t(x) = t\Phi(x) + F(x)$, $t > 0$, притом не с любой Φ , а с выпуклой квадратичной. Свойства такого семейства определяются в основном барьером F ; оказывается, что ключевым свойством барьера, ответственным за возможность полиномиальной реализации соответствующего РF-метода, является свойство, которое фиксирует следующее.

Определение 2. Функция F называется самосогласованным барьером для выпуклого замкнутого множества $G \subset R^n$, $\text{int } G \neq \emptyset$, если F — стандартно самосогласованная функция на $\text{int } G$, имеющая $\text{int } G$ своей есте-

ственной областью определения и такая, что величина

$$\vartheta(F) = \max \{ \sup \{ \lambda^2(x, F) \mid x \in \text{int } G \}, 1 \}$$

конечна; $\vartheta(F)$ называется параметром самосогласованного барьера F .

Основной для дальнейшего результат о барьерном семействе, порожденном самосогласованным барьером, таков (мы формулируем его в самом лаконичном, но не в самом общем виде).

Предложение 2. Пусть G — выпуклое замкнутое ограниченное множество с непустой внутренностью в R^n , f — выпуклая квадратичная форма на R^n , F — самосогласованный барьер для G с параметром $\vartheta(F)$ и $F_t(x) = tf(x) + F(x)$, $t > 0$. Пусть, далее, $(x_0, t_0) \in \text{int } G \times (0, \infty)$ таково, что

$$\lambda(x_0, F_{t_0}(x_0)) \leq 0,1.$$

Рассмотрим процесс

$$t_{i+1} = t_i \exp \{ \pm 0,1 \vartheta^{-1/2}(F) \}, \quad x_{i+1} = x_i - (F''_{t_{i+1}}(x_i))^{-1} F'_{t_{i+1}}(x_i)$$

(F' , F'' — градиент и гессиан F_t по x относительно какой-нибудь евклидовой структуры R^n). Указанный процесс корректно определен (все x_i лежат в $\text{int } G$), и при всех i выполнены неравенства

$$\lambda(x_i, F_{t_i}(x_i)) \leq 0,1, \quad (20)$$

$$f(x_i) - \min_G f \leq 2\vartheta(F)/t_i. \quad (21)$$

Мы видим, что, имея достаточно близкую к траектории $x^*(t)$ минимумов барьерного семейства (конкретно, удовлетворяющую (18)) точку (x_0, t_0) и увеличивая на шаге значение параметра t в $(1 + O(\vartheta^{-1/2}(F)))$ раз, мы обеспечим сходимость по функционалу к решению задачи минимизации f на G со скоростью геометрической прогрессии со знаменателем $(1 - O(\vartheta^{-1/2}(F)))$. Что касается выбора подходящей стартовой точки (x_0, t_0) , то он не представляет принципиальных затруднений. В самом деле, пусть мы имеем в своем распоряжении некоторую внутреннюю точку w множества G . Положим $g(x) = -DF(w)[x-w]$ и введем семейство $F_t^*(x) = tg(x) + F(x)$. Точка $(x_0 = w, t_0 = 1)$ по выбору g в точности лежит на траектории $x^{**}(t)$ минимумов этого семейства; поэтому из нее можно начать процесс отслеживания указанной траектории при $t \rightarrow 0$ (т. е. действовать, как описано в предложении, выбирая знак — в формуле преобразования текущего значения параметра). При $t \rightarrow 0$ траектория $x^{**}(t)$ сходится, очевидно, к точке $x(F)$ минимума F на $\text{int } G$ — к F -центру области G . Имеем $\lambda(x(F), F) = 0$; но тогда и $\lambda(x, F_t) \leq 0,1$ для всех достаточно близких к $x(F)$ точек x и всех достаточно близких к 0 значений t . В частности, если в качестве x_0 выбрать получающуюся при отслеживании траектории x^{**} точку, отвечающую достаточно малому значению t , а затем взять за t_0 достаточно малое положительное число, то (18) будет заведомо удовлетворенно. Эти соображения можно сделать строгими и конструктивными, получив в результате двухэтапный метод барьеров [14], скорость сходимости которого устанавливает следующее.

Предложение 3. Пусть f, F, G — те же, что и в предложении 2, а w — внутренняя точка G . При любом $\epsilon \in (0, 1)$ стартовый из w двухэтапный метод барьеров строит ϵ -решение задачи минимизации f на G за не более чем $O(\vartheta^{1/2}(F) \ln(2\vartheta(F)/(\epsilon\delta)))$ итераций, каждая из которых состоит в выполнении одного шага метода Ньютона в применении к функции вида $f \Phi(\cdot) + F(\cdot)$, где в зависимости от итерации Φ есть либо определенная выше линейная форма g , либо сама f . Здесь $\delta = 1 - \rho_{x(F)}(w)$, $\rho_{x(F)}(\cdot)$ — функция Минковского для G с полюсом в точке $x(F)$.

5. Полиномиальные PF-методы в линейном программировании. Из предыдущего следует, что мы можем строить сходящиеся со скоростью геометрической прогрессии методы минимизации выпуклых квадратичных форм на тех выпуклых компактах, для которых умеем строить самосогласованные барьеры. Оказывается, что каждое выпуклое замкнутое тело в R^n допускает самосогласованный барьер с параметром $O(n)$ (здесь и далее константы во всех $O(\cdot)$ — абсолютные). Однако конст-

рукция «универсального барьера» в общем случае доставляет весьма сложно вычислимые функции, так что для построения эффективных методов нужно научиться строить «простые» барьеры хотя бы не для всех, но для многих достаточно важных типов тел G . На помощь приходят два простых соображения (немедленно вытекающие из определения 2): прежде всего, если F — самосогласованный барьер с параметром $\vartheta(F)$ для $G \subset R^n$ и $A: R^k \rightarrow R^n$ — аффинное отображение, образ которого пересекает внутренность G , то $F^* \equiv F(A(\cdot))$ — самосогласованный барьер для $G^* = A^{-1}(G)$, причем $\vartheta(F^*) \leq \vartheta(F)$. Далее, если F_i — самосогласованные барьеры для тел $G_i \subset R^n$, $1 \leq i \leq k$, и пересечение G этих тел имеет непустую внутренность, то $F \equiv \sum F_i$ есть самосогласованный барьер для G , причем $\vartheta(F) \leq \sum \vartheta(F_i)$. Далее, функция $\ln 1/x$, как легко видеть, является самосогласованным с параметром 1 барьером для неотрицательного луча вещественной оси. Отсюда в силу предшествующих замечаний следует, что многогранник G с непустой внутренностью, выделенный системой m линейных неравенств вида $\{a_i^T x \leq b_i | 1 \leq i \leq m\}$, допускает самосогласованный барьер

$$F(x) = \sum_{i=1}^m \ln(1/(b_i - a_i^T(x))) \quad (22)$$

с параметром m , следовательно, минимизацию выпуклой квадратичной (и тем более — линейной) формы на G — т. е. задачу ЛП и, более общо, выпуклого квадратичного программирования с линейными ограничениями, — можно осуществить методом этого пункта, обеспечив оценку трудоемкости отыскания ϵ -решения в $O(m^{1/2} \ln(2m/(\delta\epsilon)))$ итераций по $O(mn^2)$ операций каждая (последнее есть очевидная оценка трудоемкости шага метода Ньютона для функции, получающейся квадратичным возмущением функции (22)). Здесь δ — описанная выше «степень близости к границе G » стартовой точки $w \in \text{int } G$ (априорное знание такой точки, как и компактность G , предполагаются). Однако «независимое» от шага к шагу решение линейных систем, определяющих ньютоновские смещения, не является наилучшим способом действий; указанные системы, как показывает более детальный анализ, на смежных итерациях близки друг к другу, что позволяет реализовать кармаркаровское ускорение — рекуррентную схему пересчета соответствующих обратных матриц (более точно, достаточно хороших, с точки зрения сохранения скорости сходимости, приближений к этим обратным матрицам), для которой средняя по итерациям трудоемкость шага Ньютона оказывается величиной порядка $O(m^{1/2}n^2 + mn)$ операций вместо исходного $O(mn^2)$ операций. Соответственно, трудоемкость отыскания ϵ -решения задачи ЛП (и даже квадратичной задачи на R^n с m линейными ограничениями — неравенствами) может быть доведена до $O((mn^2 + m^{3/2}n) \ln(2m/(\delta\epsilon)))$ операций, т. е. сделана кубической по размерности задачи. Такого рода оценки для задач ЛП являются на сегодня рекордными. Укажем, что метод с такой оценкой был впервые предложен П. Вадьей [8], которому удалось ввести кармаркаровское ускорение в незадолго перед тем предложенный Дж. Ренегаром [12] первый «современный» полиномиальный RF-метод. Заметим, что это был не метод барьеров, а метод центров; речь шла о задаче ЛП, и отслеживалась траектория минимумов семейства $F_t(x) = m \ln(1/(t - f_0(x))) + F(x)$, где f_0 — (линейный) целевой функционал задачи ЛП, а F — барьер (22) для допустимого множества задачи. Ясно, что траектория минимумов этого семейства лишь параметризацией отличается от траектории барьерного семейства $tf_0(x) + F(x)$, так что метод центров и метод барьеров, отвечающие одним и тем же f_0 и F , отслеживают одну и ту же траекторию, хотя и используют для этого разные формулы. Оценки скорости сходимости обоих методов одинаковы, с точностью до абсолютных постоянных. Следует заметить, что большинство публикаций по полиномиальным RF-методам, видимо, по чисто историческим причинам связаны со схемой метода центров; пред-

ставляющийся более прозрачным метод барьеров был независимо предложен К. Гонзагой [9] (для ЛП) и Ю. Нестеровым [14, 17] (для ЛП и квадратичных задач с линейными ограничениями).

6. **Нелинейные задачи.** Может показаться, что возможности намеченного выше подхода к конструированию полиномиальных РF-методов исчерпываются задачами квадратичной минимизации. Однако ясно, что условно-экстремальную выпуклую задачу вида

$$f_0(x) \rightarrow \min | f_i(x) \leq 0, \quad 1 \leq i \leq m, \quad (23)$$

можно многими способами свести к задаче минимизации линейной формы на выпуклом множестве G , скомбинированном из надграфиков функций f_i ; соответственно вопрос о применимости описанных выше методов к (23) есть вопрос о том, как построить самосогласованный барьер для результирующего G . В [16, 18] явно указаны барьеры для большого числа «типичных» для выпуклой оптимизации областей и развито своего рода «исчисление барьеров», позволяющее строить самосогласованные барьеры для «сложных» областей. Разработанный в упомянутых работах подход позволяет, в частности, строить полиномиальные алгоритмы для таких выпуклых задач вида (23), как квадратичные с квадратичными ограничениями, задачи геометрического программирования, аппроксимации в p -нормах, минимизации матричных норм, минимизации на конусе симметричных неотрицательно определенных матриц, задачи отыскания экстремальных эллипсоидов.

ЛИТЕРАТУРА

1. Вересков А. И., Гольштейн Е. Г. О задаче линейного программирования в размытой постановке // Экономика и мат. методы. 1986. Т. XXII. Вып. 6.
2. Гольштейн Е. Г. Метод декомпозиции задач линейного и выпуклого программирования // Экономика и мат. методы. 1985. Т. XXI. Вып. 6.
3. Гольштейн Е. Г. Общий подход к декомпозиции оптимизационных задач // Изв. АН СССР. Техн. кибернетика. 1987. Вып. 1.
4. Хачиян Л. Г. Полиномиальный алгоритм в линейном программировании // Докл. АН СССР. 1979. Т. 244. Вып. 5.
5. Юдин Д. Б., Немировский А. С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач // Экономика и мат. методы. 1976. Т. XII. Вып. 2.
6. Шор Н. З. Метод отсекающего с растяжением пространства для решения задач выпуклого программирования // Кибернетика. 1977. Вып. 1.
7. Karmarkar N. A new Polynomial-time Algorithm for Linear Programming // Combinatorica. 1984. V. 4.
8. Vaidya P. An Algorithm for Linear Programming which Requires $O((m+n)n^2 + (m+n)^{1.5}nL)$ Arithmetic Operations // AT&T Bell Laboratories, Murray Hill. N. J., 1987.
9. Gonzaga C. C. An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ Operations. Tech. Report. Dept. of Electrical Engineering and Computer Sciences, Berkeley, California, 1987.
10. Monteiro R. C. I. Adler An $O(n^3)$ Primal-dual Interior Point Algorithm for Linear Programming. Report ORC 87-4, Operations Research Center, Dept. of Operations Research, University of California, Berkeley, 1987.
11. Немировский А. С. Об одном алгоритме типа Кармаркара // Изв. АН СССР. Техн. кибернетика. 1987. Вып. 1.
12. Renegar J. A Polynomial-time Algorithm, Based on Newton's Method, for Linear Programming // Math. Prog. 1988. V. 40.
13. Нестеров Ю. Е. Метод линейного программирования с кубической трудоемкостью // Экономика и мат. методы. 1988. Т. XXIV. Вып. 1.
14. Нестеров Ю. Е. Полиномиальные методы в линейном и квадратичном программировании // Изв. АН СССР. Техн. кибернетика. 1988. Вып. 3.
15. Нестеров Ю. Е. Полиномиальные итеративные методы линейного и квадратичного программирования // Вопросы кибернетики. М.: Наука, 1988.
16. Нестеров Ю. Е., Немировский А. С. Полиномиальные методы барьеров в выпуклом программировании // Экономика и мат. методы. 1988. Т. XXIV. Вып. 6.
17. Нестеров Ю. Е. Двойственные полиномиальные алгоритмы в линейном программировании // Кибернетика. 1989. Вып. 1.
18. Нестеров Ю. Е., Немировский А. С. Самосогласованные функции и полиномиальные алгоритмы в выпуклом программировании. М.: ЦЭМИ АН СССР, 1989.

Поступила в редакцию
23 VI 1989