1. Краткий обзор алгоритмов внутренних точек в линейном программировании

1.1. Теоретические основы линейной оптимизации

Линейная оптимизация [2], [14] (поиск экстремума линейной функции при ограничениях в форме линейных неравенств) - один из важнейших разделов математики как в плане теоретических исследований, так и в плане практических приложений. Как математическая дисциплина линейная оптимизация ведет отсчет с основополагающей работы Л.В.Канторовича [24], каждый из разделов которой посвящен моделированию конкретной экономической задачи. В этой же работе был представлен метод разрешающих множителей - прообраз разработанного в 1947 году Дж.Данцигом симплекс-метода [6]. Благодаря простоте реализации и высоким скоростным характеристикам модификации симплекс-метода стали наиболее распространенным способом решения задач линейного программирования.

В то же время, симплекс-метод является далеко не единственным таким способом. В частности, статьями [8], [9], [13] было порождено альтернативное направление - алгоритмы внутренних точек. Их название связано с тем, что, в отличие от симплекс-метода, перебирающего угловые точки многогранника допустимых решений, вычислительный процесс в алгоритмах внутренних точек происходит в относительной внутренности допустимого множества. Кроме того, вырабатываемая последовательность приближений сходится к относительно внутренней точке множества оптимальных решений.

За рубежом повышенный интерес к алгоритмам внутренних точек возник в 80-х годах и был обусловлен созданием полиномиальных алгоритмов для задач линейного программирования. Понятие полиномиальной разрешимости класса задач играет основную роль в теории сложности. Если

алгоритм способен решить любую задачу из исследуемого класса за время, выражаемое в виде некоторого полинома от ее размерности, то алгоритм имеет полиномиальную сложность, а сам класс задач называется полиномиально разрешимым.

В работе Л.Г.Хачияна [34] на основе использующейся в выпуклом программировании техники построения последовательности сокращающихся в объеме множеств [36] было показано, что линейное программирование относится к классу полиномиально разрешимых задач. Несмотря на то, что алгоритм, исследуемый Хачияном, на практике показал невысокую скорость сходимости, результат Хачияна был крайне важен в теоретическом плане и способствовал последующим исследованиям в этой области. Кроме того, на основе используемой в [36] техники впоследствии был разработан оригинальный класс полиномиальных алгоритмов погружения-отсечения [1], в последние годы существенно повысивших свою эффективность [30].

В 1984 году Н.Кармаркаром был создан первый полиномиальный алгоритм внутренних точек [51] для задачи линейного программирования. Хотя анонсированное Кармаркаром утверждение, что программная реализация его метода решает практические и тестовые задачи линейного программирования быстрее, чем имеющиеся программные реализации симплекс-метода, в результате проведенных рядом исследователей экспериментов не подтвердилось [55], [70], статья вызвала огромный интерес к алгоритмам внутренних точек, следствием которого было более 2000 работ, посвященных данному направлению, выполненных за последующие 15 лет учеными всего мира. Следует отметить, что зарубежные ученые часто воспроизводили и переоткрывали сделанное в России И.И.Дикиным, Ю.Г.Евтушенко, В.И.Зоркальцевым, В.Г.Жаданом и другими в 70-80 годах.

Прежде чем более подробно остановиться на основных направлениях методов внутренних точек, введем ряд обозначений, приведем некоторые

факты теории двойственности и сделаем предположения относительно рассматриваемых в дальнейшем задач.

Пара взаимно-двойственных задач линейного программирования

В первых двух главах диссертационной работы будет рассматриваться пара взаимно-двойственных задач линейного программирования следующего вида:

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} \to \min_{\mathbf{x}\in\mathbf{X}},\ \mathbf{X} = \left\{\mathbf{x} \in \mathbf{R}^{\mathbf{n}}:\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\right\}, \tag{1}$$

$$\mathbf{b}^{\mathbf{T}}\mathbf{u} \to \max_{\mathbf{u}\in\mathbf{U}},\ \mathbf{U} = \left\{\mathbf{u} \in \mathbf{R}^{\mathbf{m}}:\ \mathbf{g}(\mathbf{u}) \equiv \mathbf{c} - \mathbf{A}^{\mathbf{T}}\mathbf{u} \geq \mathbf{0}\right\}, \tag{2}$$

где $\mathbf{c} \in \mathbf{R}^{\mathbf{n}}$, $\mathbf{b} \in \mathbf{R}^{\mathbf{m}}$, $\mathbf{A}$ - матрица размерности $m \times n$, $rank\,\mathbf{A} = m$. Здесь $\mathbf{X}$, $\mathbf{U}$ - множества допустимых решений задач (1)-(2). Множества оптимальных решений этих задач обозначим

$$\overline{\mathbf{X}} = \operatorname{Arg}\min_{\mathbf{x}\in\mathbf{X}} \mathbf{c}^{\mathbf{T}}\mathbf{x}, \quad \overline{\mathbf{U}} = \operatorname{Arg}\max_{\mathbf{u}\in\mathbf{U}} \mathbf{b}^{\mathbf{T}}\mathbf{u}.$$

Введем функцию от векторов $\mathbf{x} \in \mathbf{R}^{\mathbf{n}}$, $\mathbf{u} \in \mathbf{R}^{\mathbf{m}}$

$$f(\mathbf{x},\mathbf{u}) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u}.$$

Отметим, что для любых $\mathbf{x} \in \mathbf{R}^{\mathbf{n}}$, $\mathbf{u} \in \mathbf{R}^{\mathbf{m}}$ при условии $\mathbf{A}\mathbf{x} = \mathbf{b}$ выполняется равенство

$$f(\mathbf{x},\mathbf{u}) = \sum_{j=1}^{n} x_j g_j(\mathbf{u}).$$

Действительно,

$$\mathbf{x}^{\mathbf{T}}\mathbf{g}(\mathbf{u}) = \mathbf{x}^{\mathbf{T}}\left(\mathbf{c} - \mathbf{A}^{\mathbf{T}}\mathbf{u}\right) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \left(\mathbf{A}^{\mathbf{T}}\mathbf{u}\right)^{\mathbf{T}}\mathbf{x} = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{u}^{\mathbf{T}}\mathbf{A}\mathbf{x} = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u}.$$

Основные факты теории двойственности

Для задач (1), (2) возможна одна из следующих четырех ситуаций:

1. $\mathbf{X} = \varnothing$, $\mathbf{U} = \varnothing$. Тогда $\overline{\mathbf{X}} = \varnothing$, $\overline{\mathbf{U}} = \varnothing$.

2. $\mathbf{X} = \varnothing$, $\mathbf{U} \neq \varnothing$. Тогда $\overline{\mathbf{X}} = \varnothing$, $\overline{\mathbf{U}} = \varnothing$, целевая функция задачи (2) неограниченна сверху на множестве допустимых решений этой задачи.

3. $\mathbf{X} \neq \varnothing$, $\mathbf{U} = \varnothing$. Тогда $\overline{\mathbf{X}} = \varnothing$, $\overline{\mathbf{U}} = \varnothing$, целевая функция задачи (1) неограниченна снизу на множестве допустимых решений этой задачи.

4. $\mathbf{X} \neq \varnothing$, $\mathbf{U} \neq \varnothing$. Тогда $\overline{\mathbf{X}} \neq \varnothing$, $\overline{\mathbf{U}} \neq \varnothing$. Для любых допустимых решений $\mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}$ выполняется неравенство

$$f(\mathbf{x}, \mathbf{u}) \geq 0.$$

Для любых оптимальных решений $\overline{\mathbf{x}} \in \overline{\mathbf{X}}, \overline{\mathbf{u}} \in \overline{\mathbf{U}}$ выполняется равенство

$$f(\overline{\mathbf{x}}, \overline{\mathbf{u}}) = 0,$$

которое означает совпадение для оптимальных решений значений целевых функций взаимно-двойственных задач

$$\mathbf{c}^{\mathbf{T}}\overline{\mathbf{x}} = \mathbf{b}^{\mathbf{T}}\overline{\mathbf{u}}$$

и выполнение условий дополняющей нежесткости

$$\overline{x}_j g_j(\overline{\mathbf{u}}) = 0, \ \ j = 1, \ldots, n.$$

Кроме того, существуют $\tilde{\mathbf{x}} \in \overline{\mathbf{X}}$, $\tilde{\mathbf{u}} \in \overline{\mathbf{U}}$, для которых условия дополняющей нежесткости выполняются в строгой форме

$$\tilde{x}_j g_j(\tilde{\mathbf{u}}) = 0, \ \ \max\left\{\tilde{x}_j, g_j(\tilde{\mathbf{u}})\right\} > 0, \ \ j = 1, \ldots, n.$$

Приведенные факты теории двойственности, кроме последнего, получили обоснование в работе [73] Дж. фон Неймана. Последнее утверждение было впервые доказано в [47] и приобрело название теоремы Гольдмана-Таккера.

Самосопряженная задача линейного программирования

Из приведенных фактов теории двойственности следует, что пара задач (1)-(2) равносильна следующей задаче линейного программирования

$$f(\mathbf{x}, \mathbf{u}) \to \min_{\mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}}. \qquad (3)$$

Поскольку двойственная к задаче (3) задача совпадает с ней самой, будем называть ее самосопряженной.

Предположение 1

Пусть для задачи (1) существуют допустимые решения, для которых все ограничения-неравенства выполняются в строгой форме, то есть имеются векторы $\mathbf{x} \in \mathbf{X}$, такие что $\mathbf{x} > \mathbf{0}$. Здесь и далее неравенство $\mathbf{x} > \mathbf{0}$ для вектора $\mathbf{x}$ будет обозначать, что все его компоненты положительные.

Следствием данного предположения является то, что относительно внутренними точками [29] множества допустимых решений задачи (1) являются векторы, для которых ограничения-неравенства выполняются в строгой форме:

$$ri\,\mathbf{X} = \left\{ \mathbf{x} \in \mathbf{R}^{\mathbf{n}},\ \mathbf{Ax} = \mathbf{b},\ \mathbf{x} > \mathbf{0} \right\}.$$

Из предположения 1 также следует, что если непусто множество $\overline{\mathbf{U}}$ оптимальных решений задачи (2), то все компоненты вектора $\mathbf{g}(\mathbf{u})$ при $\mathbf{u} \in \overline{\mathbf{U}}$ будут [21] ограниченными.

Предположение 2

Пусть для задачи (2) существуют допустимые решения, для которых все ограничения-неравенства выполняются в строгой форме, то есть имеются векторы $\mathbf{u} \in \mathbf{U}$, такие что $\mathbf{g}(\mathbf{u}) > \mathbf{0}$.

Из предположения 2 следует, что относительно внутренними точками множества допустимых решений задачи (2) являются векторы, для которых ограничения-неравенства выполняются в строгой форме:

$$ri\,\mathbf{U} = \left\{ \mathbf{u} \in \mathbf{R}^{\mathbf{m}},\ \mathbf{g}(\mathbf{u}) > \mathbf{0} \right\}.$$

Также предположение 2 влечет ограниченность множества $\overline{\mathbf{X}}$ оптимальных решений задачи (1), при условии, что оно является непустым.

Для некоторых из рассматриваемых ниже алгоритмов (в частности, для прямых и двойственных аффинно-масштабирующих) требуется выполнение только одного из приведенных предположений. Для других (к которым относится большинство полиномиальных) - обязательно должны выполняться и предположение 1, и предположение 2. Подробнее об этом будет сказано при описании ввода в допустимую область.

1.2. Аффинно-масштабирующие алгоритмы

Истоки алгоритмов внутренних точек

Истоки алгоритмов внутренних точек восходят к предложенной в 1965 году Л.В.Канторовичем методике оценки множителей Лагранжа ограничений задачи при неоптимальном плане методом наименьших квадратов. Подробно данная методика описана в [10].

По одному из ее вариантов для допустимого, но неоптимального решения $\mathbf{x^k} \in ri\,\mathbf{X}$ задачи (1), вектор оценок определяется по формуле

$$\mathbf{u^k} = \arg\min_{\mathbf{u}\in\mathbf{R^m}} \sum_{j=1}^{n} d_j^k \left(g_j(\mathbf{u})\right)^2, \qquad (4)$$

где

$$d_j^k = \left(x_j^k\right)^2, \ \ j = 1,\ldots,n. \qquad (5)$$

На основе этого правила И.И.Дикиным [8] в 1967 году был разработан первый из алгоритмов внутренних точек для задачи линейного программирования, в котором решение итеративно улучшается по правилу

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \lambda_k \Delta\mathbf{x^k}, \quad k = 1,2,\ldots, \qquad (6)$$

где $\Delta\mathbf{x^k}$ - направление корректировки, вычисляемое по формуле

$$\Delta x_j^k = -d_j^k g_j\left(\mathbf{u^k}\right), \ \ j = 1,\ldots,n, \qquad (7)$$

а λ_k - шаг корректировки, определяемый равенством

$$\lambda_k = 1 \Big/ \sqrt{\sum_{j=1}^{n} d_j^k \left(g_j\left(\mathbf{u^k}\right)\right)^2}\,. \qquad (8)$$

Геометрически вектор $\mathbf{x^{k+1}}$ в данном алгоритме определяется как точка минимума целевой функции задачи (1) на вписанном в $\mathbf{X}$ эллипсоиде с центром в $\mathbf{x^k}$. Действительно, вектор $\mathbf{x^{k+1}}$ является решением задачи

$$\mathbf{c^T x} \to \min, \quad \mathbf{A}\left(\mathbf{x} - \mathbf{x^k}\right) = \mathbf{0}, \quad \sum_{j=1}^{n} \left(\left(x_j - x_j^k\right) \Big/ x_j^k \right)^2 \le 1.$$

Обоснование сходимости последовательности векторов $\mathbf{x^k}$ к точке из $ri\,\overline{\mathbf{X}}$ со скоростью геометрической прогрессии при предположении о невырожденности задачи (1) (то есть когда невырождены все ее базисные планы) было опубликовано И.И.Дикиным в [9].

Семейство прямых аффинно-масштабирующих алгоритмов

В.И.Зоркальцевым в [16] было предложено изменение алгоритма (4)-(8), базирующееся на идее движения вдоль направления корректировки $\Delta\mathbf{x^k}$ не до границы эллипсоида, а на часть пути, равную γ, от точки $\mathbf{x^k}$ до границы положительного ортанта. В этом случае шаг корректировки вычисляется не в соответствии с (8), а следующим образом:

$$\lambda_k = \gamma \min_{j:\Delta x_j^k < 0} \left(- x_j^k \Big/ \Delta x_j^k \right), \quad \gamma \in \left(0;\, 2/3\right). \qquad (9)$$

Во второй половине 80-х годов после ряда публикаций, в частности, [40] и [72], где данный алгоритм вводился как модификация алгоритма Кармаркара, за ним закрепилось название affine-scaling method. В 1991 году была получено [71] его полное теоретическое обоснование без предположения о невырожденности задачи.

Данные ограничения, накладываемые на γ, являются необходимыми для обоснования алгоритма в случае отсутствия предположения о невырожденности задачи, хотя при использовании алгоритма на практике (где

вырожденные задачи встречаются достаточно редко) в программно-вычислительных комплексах рекомендуются к применению и применяются [37], [59], [64] большие величины: $\gamma = 0.95$, $\gamma = 0.99$ и даже изменяющиеся по итерациям и асимптотически стремящиеся к единице величины γ^k.

В [56] был приведен контрпример, демонстрирующий, что аффинно-масштабирующий алгоритм на вырожденных задачах может не сходиться к оптимальному решению при $\gamma = 0.999$ из-за так называемого зигзагообразного движения. В [67] показано, что подобная ситуация может произойти при любом $\gamma \in (0.9107; 1)$.

Наряду с (5), используются и другие способы задания весовых коэффициентов d_j^k. При этом величины $1/d_j^k$ можно интерпретировать как штрафы, сдерживающие выход переменных на ближайшие границы области допустимых решений. Действительно, направление корректировки $\Delta\mathbf{x}^{\mathbf{k}}$ является решением задачи

$$\mathbf{c}^{\mathbf{T}}\Delta\mathbf{x} + \frac{1}{2}\sum_{j=1}^{n}\frac{\Delta x_j^2}{d_j^k} \to \min_{\Delta\mathbf{x}\in\mathbf{R}^{\mathbf{n}}}, \quad \mathbf{A}\Delta\mathbf{x} = \mathbf{0}. \tag{10}$$

В [17] был предложен аксиоматический подход к определению коэффициентов d_j^k. Чтобы обеспечить сходимость алгоритма к оптимальному решению, необходимо потребовать выполнения неравенств

$$\overline{\sigma}\left(x_j^k\right) \geq d_j^k \geq \underline{\sigma}\left(x_j^k\right), \ j = 1,\ldots,n,$$

где $\overline{\sigma}$ и $\underline{\sigma}$ - функции, удовлетворяющие условиям

$$\overline{\sigma}(\alpha) \geq \underline{\sigma}(\alpha) > 0, \ \forall\alpha > 0, \tag{11}$$

$$\overline{\sigma}(\alpha) \leq M\alpha, \ \forall\alpha \in (0;\varepsilon) \tag{12}$$

при некоторых $\varepsilon > 0$, $M > 0$.

В частности, коэффициенты d_j^k могут зависеть только от значений x_j^k. Так, например, они заданы в следующем обобщении правила (5), где в целях сужения диапазона весовых коэффициентов вводится величина $N > 0$:

$$d_j^k = \min\left\{\left(x_j^k\right)^p, N\right\},\ \ j = 1,\ldots,n,\ p \geq 1.$$

В этом случае коэффициент γ, фигурирующий в формуле (9) вычисления шага корректировки, может принимать значения

$$\gamma \in \left(0;\, 2/\left(p+1\right)\right).$$

Не является обязательным наличие конкретных выражений для функций $\overline{\sigma}$ и $\underline{\sigma}$. Достаточно иметь доказательства существования таких функций и выполнения для них свойств (11)-(12). В уже упоминавшейся работе [17] было, в частности, предложено применять при $k > 1$ весовые коэффициенты, использующие вычисленные на предыдущей итерации векторы множителей Лагранжа ограничений-равенств вспомогательной задачи:

$$d_j^k = x_j^k \Big/ \max\left\{\varepsilon, g_j\left(\mathbf{u}^{\mathbf{k-1}}\right)\right\},\ \ j = 1,\ldots,n,\ \ \varepsilon > 0.$$

Такое правило позволяет уменьшить сильно негативное влияние погрешностей вычислений, особенно в финальной стадии вычислительного процесса, и увеличивает численную устойчивость алгоритма. Одной из целей экспериментального исследования является практическая проверка скорости сходимости данного алгоритма.

Алгоритмы данного типа рассматривались и другими исследователями, в частности, Ю.Г.Евтушенко и В.Н.Жаданом [11], [12], а также нашли практические приложения [18], [23], [32], где показали свою высокую эффективность.

Ввод в допустимую область

Для прямых аффинно-масштабирующих алгоритмов требуется выполнение предположения 1. Более того, необходимо иметь стартовую точку

$\mathbf{x} \in \mathbf{X}$, для которой ограничения-неравенства выполняются в строгой форме. Поскольку такая точка не всегда является априори известной, то с помощью незначительных модификаций алгоритма возможно совмещение процессов оптимизации и ввода в допустимую область. Вычислительный процесс при этом начинается с любого вектора $\mathbf{x^1} > \mathbf{0}$. На каждой итерации вычисляется вектор невязок балансовых ограничений-равенств $\mathbf{r^k} = \mathbf{b} - \mathbf{A}\mathbf{x^k}$. Для определения направления корректировки вместо задачи (10) решается следующая:

$$\mathbf{c^T}\Delta\mathbf{x} + \frac{1}{2}\sum_{j=1}^{n}\frac{\Delta x_j^2}{d_j^k} \to \min_{\Delta\mathbf{x}\in\mathbf{R^n}}, \quad \mathbf{A}\Delta\mathbf{x} = \mathbf{r^k}. \tag{13}$$

В остальном алгоритм остается неизменным. Невязки ограничений-равенств сокращаются по итерациям по правилу

$$\mathbf{r^{k+1}} = (1 - \lambda_k)\mathbf{r^k}.$$

Поэтому при $\mathbf{x} \notin \mathbf{X}$ шаг корректировки λ_k должен ограничиваться сверху единицей: если $\mathbf{r^k} \neq \mathbf{0}$, то следует пересчет:

$$\lambda_k := \min\{1, \lambda_k\}.$$

Если вычислительный процесс сходится к вектору $\tilde{\mathbf{x}} \in ri\,\mathbf{X}$, некоторые компоненты $\tilde{x}_j$ которого равны нулю, это означает, что $x_j = 0$ при любых $\mathbf{x} \in \mathbf{X}$. Таким образом, переменную x_j можно без ущерба исключить из задачи (1).

Решение задач с интервальными ограничениями на переменные

Рассмотренные алгоритмы пригодны и для решения задачи, в которой вместо условия $\mathbf{x} \geq \mathbf{0}$ используется более общее условие $\mathbf{x} \in [\underline{\mathbf{x}}; \overline{\mathbf{x}}]$ при заданных векторах ограничений $\overline{\mathbf{x}}, \underline{\mathbf{x}} \in \mathbf{R^n}$, причем $\overline{\mathbf{x}} > \underline{\mathbf{x}}$. При этом допускается, что некоторые компоненты вектора переменных **x** могут быть неограниченны сверху или снизу, то есть возможно, что некоторые $\underline{x}_j = -\infty$, а некоторые

$\bar{x}_j = +\infty$. Алгоритм вырабатывает последовательность векторов $\mathbf{x^k} \in (\underline{\mathbf{x}}; \bar{\mathbf{x}})$ в соответствии с вышезаписанными правилами итеративного перехода с тем отличием, что весовые коэффициенты вычисляются по формуле

$$d_j^k = \min\left\{N, \left(x_j^k - \underline{x}_j\right)^2, \left(\bar{x}_j - x_j^k\right)^2\right\}, \ j = 1,\ldots,n,$$

а шаг корректировки - по формуле

$$\lambda_k = \min\left\{\delta_k, \ \gamma \min_{j:\Delta x_j^k<0} \frac{\underline{x}_j - x_j^k}{\Delta x_j^k}, \ \gamma \min_{j:\Delta x_j^k>0} \frac{\bar{x}_j - x_j^k}{\Delta x_j^k}\right\}, \ j = 1,\ldots,n,$$

где $\delta_k = 1$, если $\mathbf{Ax^k} \neq \mathbf{b}$ и $\delta_k = \infty$, если $\mathbf{Ax^k} = \mathbf{b}$, $\gamma \in (0; 2/3)$.

Решение систем линейных уравнений и неравенств

Отдельный интерес представляет задача ввода в допустимую область. При этом целевая функция не учитывается, то есть вектор **c** полагается равным нулю. Необходимо либо отыскать допустимое решение, либо максимально быстро идентифицировать факт несовместности следующей системы линейных уравнений и неравенств:

$$\mathbf{Ax} = \mathbf{b}, \quad \bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}. \tag{14}$$

Система (14) сводится к задаче линейного программирования с помощью введения дополнительной переменной. Пусть имеется некоторая стартовая точка $\mathbf{x^1}$, удовлетворяющая в строгой форме ограничениям-неравенствам:

$$\bar{x}_j > x_j^1 > \underline{x}_j. \tag{15}$$

В частности, если мы обладаем дополнительной полезной информацией, например, полученной из предыдущих расчетов, ее можно учесть при формировании стартовой точки.

Нас будет интересовать случай, когда $\mathbf{Ax^1} \neq \mathbf{b}$, то есть стартовая точка не является допустимой для системы (14). Введем вектор невязок ограничений-равенств

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^{\mathbf{1}}. \qquad (16)$$

Рассмотрим задачу линейного программирования, переменными в которой выступают компоненты вектора **x** и дополгительная переменная α:

$$\alpha \to \min, \quad \mathbf{A}\mathbf{x} + \alpha\,\mathbf{r} = \mathbf{b}, \quad \overline{\mathbf{x}} \ge \mathbf{x} \ge \underline{\mathbf{x}}, \quad \alpha \ge 0. \qquad (17)$$

Из (15), (16) следует, что значения $\mathbf{x} = \mathbf{x}^{\mathbf{1}}$, $\alpha = 1$ составляют относительно внутреннюю точку множества допустимых решений задачи (17), то есть эта задача имеет допустимые решения, и мы располагаем одним из них.

Несложно убедиться, что если для оптимального решения задачи (17) $\alpha = 0$, то соответствующий вектор **x** является допустимым решением системы (14). Если же оптимальное значение α положительно, то система (14) несовместна.

Продемонстрированный прием сведения системы уравнений и неравенств к задаче линейного программирования, предложенный В.И.Зоркальцевым в [16], является базовым при создании алгоритмов внутренних точек для решения систем линейных уравнений и неравенств.

Быстрая идентификация случая несовместности

Одной из наиболее существенных проблем долгое время являлась невозможность быстрой идентификации случая несовместности ограничений системы (14). В ранее использовавшихся алгоритмах для этого, как правило, требовался большой объем вычислений, сопоставимый с необходимым для получения допустимой точки в случае совместности ограничений. В то же время данная задача очень актуальна в связи с тем, что необходимость решения систем линейных уравнений и неравенств часто возникает при итеративной линеаризации [28] существенно нелинейных моделей. При этом система (14) представляет собой линеаризованную подзадачу, нередко оказывающуюся несовместной, в первую очередь, по причине погрешностей линеаризации. В этом случае нецелесообразно решать ее до конца, поскольку

это может оказаться достаточно трудоемким процессом, - можно лишь перейти в новую точку, где заново провести линеаризацию.

В.И.Зоркальцевым [3] на основе теоремы Фаркаша об альтернативных неравенствах [41] был построен и обоснован критерий несовместности системы неравенств. Для системы (14) он запишется следующим образом:

Теорема 1

Система уравнений и неравенств (14) несовместна в том и только в том случае, если существует вектор $\mathbf{u} \in \mathbf{R}^{\mathbf{m}}$, при котором

$$\varphi(\mathbf{u}) > 0,$$

где

$$\varphi(\mathbf{u}) = \mathbf{b}^{\mathbf{T}}\mathbf{u} - \overline{\mathbf{x}}^{\mathbf{T}}\left(\mathbf{A}^{\mathbf{T}}\mathbf{u}\right)_{+} - \underline{\mathbf{x}}^{\mathbf{T}}\left(\mathbf{A}^{\mathbf{T}}\mathbf{u}\right)_{-}.$$

Здесь и далее $(\mathbf{v}_{+})_j = \max\{0, v_j\}$, $(\mathbf{v}\_)_j = \min\{0, v_j\}$.

Одной из целей проводимого исследования является проверка того, насколько эффективно с помощью данного критерия можно идентифицировать несовместность на практике.

Двойственные алгоритмы

В процессе решения вспомогательной задачи (13) на каждой итерации вырабатываются двойственные оценки $\mathbf{u}^{\mathbf{k}}$, которые, хотя и немонотонно, сходятся к оптимальному решению задачи (2). Причем имеются теоретические результаты, показывающие, что двойственные оценки сходятся быстрее, чем переменные прямой задачи. В этой связи особенный интерес представляет использование двойственных алгоритмов, в которых последовательность приближений $\mathbf{x}^{\mathbf{k}}$ быстрее, хотя и немонотонно, сходится к оптимальному решению задачи (1). Одной из целей работы можно считать практическую проверку того, действительно ли двойственные аффинно-

масштабирующие алгоритмы являются более эффективными, чем прямые, в целях получения решения прямой задачи.

Двойственные алгоритмы аффинно-масштабирующего метода были введены в [17] и, независимо, в [37] и [59]. Вычислительный процесс в них начинается с произвольных векторов $\mathbf{u^1}$ и $\mathbf{g^1} > \mathbf{0}$. На каждой итерации вычисляется вектор невязок ограничений-равенств $\mathbf{r^k} = \mathbf{c} - \mathbf{A^T u^k} - \mathbf{g^k}$. Направления корректировки переменных $\Delta\mathbf{u^k}$ и $\Delta\mathbf{g^k}$ находятся путем решения следующей задачи:

$$-\mathbf{b^T}\Delta\mathbf{u} + \frac{1}{2}\sum_{j=1}^{n}\frac{\left(\Delta g_j\right)^2}{d_j^k} \to \min_{\Delta\mathbf{u}\in\mathbf{R^m},\Delta\mathbf{g}\in\mathbf{R^n}}, \quad \Delta\mathbf{g} + \mathbf{A^T}\Delta\mathbf{u} = \mathbf{r^k}, \tag{18}$$

где d_j^k - весовые коэффициенты, которые могут, в частности, вычисляться как

$$d_j^k = \min\left\{\left(g_j^k\right)^p, N\right\}, \ j = 1,\ldots,n, \ p \geq 1.$$

Шаг корректировки вычисляется аналогично (9):

$$\lambda_k = \gamma \min_{j:\Delta g_j^k < 0}\left(-g_j^k \,/\, \Delta g_j^k\right), \quad \gamma \in \left(0;\, 2/(p+1)\right).$$

Поскольку невязки ограничений-равенств сокращаются по итерациям, как и в случае прямого алгоритма, по правилу

$$\mathbf{r^k} = \left(1 - \lambda_k\right)\mathbf{r^k},$$

то на фазе ввода в допустимую область шаг корректировки λ_k должен ограничиваться сверху единицей: если $\mathbf{r^k} \neq \mathbf{0}$, то следует пересчет:

$$\lambda_k := \min\left\{1, \lambda_k\right\}.$$

Итеративный переход осуществляется по правилам

$$\mathbf{u^{k+1}} = \mathbf{u^k} + \lambda_k \Delta\mathbf{u^k}, \ \mathbf{g^{k+1}} = \mathbf{g^k} + \lambda_k \Delta\mathbf{g^k}.$$

Множители Лагранжа $\mathbf{x^k}$ ограничений-равенств задачи (18) могут служить в качестве приближений к решению задачи (1).

В качестве другого способа вычисления весовых коэффициентов можно предложить следующий, использующий значения множителей Лагранжа с предыдущей итерации:

$$d_j^k = g_j^k \Big/ \max\left\{\varepsilon, x_j^{k-1}\right\},\ j = 1,\ldots,n,\ \varepsilon > 0,\ k > 1.$$

В приведенном алгоритме совмещены ввод в допустимую область с оптимизацией. Если вычислительный процесс сходится к вектору $\tilde{\mathbf{u}} \in ri\,\mathbf{U}$, такому что некоторые компоненты вектора $g_j(\tilde{\mathbf{u}})$ равны нулю, это означает, что $g_j(\mathbf{u}) = 0$ при любых $\mathbf{u} \in \mathbf{U}$. Таким образом, переменную $g_j(\mathbf{u})$ можно без ущерба исключить из задачи (2).

Вычислительная сложность одной итерации

Наиболее сложной в вычислительном отношении проблемой как в вышеизложенных аффинно-масштабирующих алгоритмах, так и в алгоритмах, рассматриваемых далее, является решение на каждой итерации системы линейных уравнений с симметричной положительно определенной матрицей вида $\mathbf{AD_kA^T}$ размерности $m \times m$, где по итерациям изменяется диагональная матрица $\mathbf{D_k} = diag\left\{d_j^k\right\}$. В частности, в прямых аффинно-масштабирующих алгоритмах такая проблема возникает при нахождении из (4) вектора двойственных оценок $\mathbf{u^k}$:

$$\mathbf{u^k} = \left(\mathbf{AD_kA^T}\right)^{-1}\mathbf{AD_kc}.$$

Для обращения матрицы можно, в частности, использовать метод квадратного корня [7], для чего требуется $m^3/2$ арифметических операций. Все остальные действия в пределах итерации требуют не более $O\left(m^2\right)$ операций. Поэтому теоретическую сложность одной итерации можно оценить как $O\left(m^3\right)$.

Примерно одинаковый объем вычислений на одной итерации позволяет сопоставлять алгоритмы не по времени получения решения (что существенно зависит от мощности компьютера, а также от конкретной программной реализации), а по числу необходимых для этого итераций. При этом ускорение процедуры обращения матрицы $\mathbf{AD_k A^T}$ является самостоятельной задачей, решение которой существенно ускорит решение исходной пары задач (1)-(2).

В то же время до сих пор остается открытым вопрос о сходимости аффинно-масштабирующих алгоритмов за число итераций, выражающееся в виде полинома от размерности задачи, в частности, из-за отсутствия для них величины, характеризующей близость к оптимуму, которая гарантированно изменяется в фиксированной пропорции.

1.3. Полиномиальные алгоритмы

Алгоритмы уменьшения потенциала

Первым подходом, обеспечивающим полиномиальность алгоритма, является подход связанный с так называемой потенциальной функцией. Потенциальная функция была введена Кармаркаром [51] для обоснования полиномиальности своего алгоритма. Хотя алгоритм Кармаркара основывался на технике проективных преобразований, впоследствии, начиная с [63], алгоритмы уменьшения потенциала (potential reduction algorithms) выделились в самостоятельное направление алгоритмов внутренних точек. Истоки алгоритмов уменьшения потенциала восходят к “методу центров” П.Хьюарда [49] решения задач нелинейного программирования.

Пусть z^* - оптимальное значение целевых функций задач (1) и (2). Прямая потенциальная функция имеет вид

$$f_1(\mathbf{x}, z) = q \ln\left(\mathbf{c^T x} - z\right) - \sum_{j=1}^{n} \ln x_j \,, \qquad (19)$$

где $z \le z^*$ - нижняя граница для оптимального значения целевой функции задачи (1), $q \ge n$ - параметр потенциальной функции. Первое слагаемое потенциальной функции - это умноженный на q логарифм невязки двойственности, а второе - логарифмическая барьерная функция, предназначенная для того, чтобы "отталкивать" решение задачи от границы допустимой области. Заметим, что если имеется допустимое решение $\mathbf{u} \in \mathbf{U}$ двойственной задачи, то можно принять $z = \mathbf{b}^{\mathbf{T}}\mathbf{u}$.

Идея прямого алгоритма уменьшения потенциала состоит в получении, начиная с исходного приближения $\mathbf{x}^{\mathbf{1}} \in ri\,\mathbf{X}$ и нижней границы z^1, последовательности векторов $\mathbf{x}^{\mathbf{k}}$ и нижних границ z^k, таких что на каждой итерации значение потенциальной функции гарантированно уменьшается на величину $\delta > 0$. При этом через k итераций получим

$$\ln\left(\mathbf{c}^{\mathbf{T}}\mathbf{x}^{\mathbf{k+1}} - z^{k+1}\right) \le \frac{f_1\left(\mathbf{x}^{\mathbf{1}}, z^1\right)}{q} - \frac{k\delta}{q} + \frac{n}{q}\ln\left(\frac{\sum x_j^{k+1}}{n}\right).$$

Отсюда выводится, что верхняя оценка числа итераций, необходимых для получения решения, для которого невязка двойственности не превышает ε, равна

$$k = \frac{q}{\delta}\left(\ln\frac{\mathbf{c}^{\mathbf{T}}\mathbf{x}^{\mathbf{1}} - z^1}{\varepsilon} + C\right),$$

где константа C зависит от данных задачи и от выбора начальной точки.

Существует значительное число модификаций методов уменьшения потенциала. Некоторые из них используют вышеописанную потенциальную функцию (19), базовым здесь считается описанный в [48] алгоритм. В [74] был разработан подход, основанный на применении функции

$$f_2(\mathbf{x}, \mathbf{u}) = q\ln\left(\mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u}\right) - \sum_{j=1}^{n}\ln x_j - \sum_{j=1}^{n}\ln g_j(\mathbf{u}),$$

которая впоследствии получила название симметричной прямо-двойственной функции Танабе-Тодда-Йе. Это позволило уменьшить теоретическую слож-

ность алгоритма до $O\left(\sqrt{n}L\right)$. L (здесь и в дальнейшем) - объем входных данных задачи, который находится по формуле

$$L = \ln(1+\Delta) + \ln\left(1+\max_{j=1,\ldots,n}|c_j|\right) + \ln\left(1+\max_{i=1,\ldots,m}|b_i|\right) + \ln(m+n),$$

где Δ - максимальное абсолютное значение определителя любой квадратной подматрицы матрицы **A**.

Инициализация алгоритма

Одной из существенных проблем является инициализация алгоритма. Возможны различные пути ее решения. Первый состоит в решении вместо (1) расширенной задачи

$$\hat{\mathbf{c}}^{\mathbf{T}}\hat{\mathbf{x}} \to \min, \quad \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{b}, \quad \mathbf{e}^{\mathbf{T}}\hat{\mathbf{x}} \le M, \quad \hat{\mathbf{x}} \ge \mathbf{0}. \tag{20}$$

Здесь $\hat{\mathbf{x}} \in \mathbf{R}^{\mathbf{n+1}}$, $\hat{\mathbf{A}} = \left(\mathbf{A}, \mathbf{b} - \mathbf{A}\mathbf{e}\right)$, $\hat{\mathbf{c}}^{\mathbf{T}} = \left(\mathbf{c}^{\mathbf{T}}, M\right)$.

Известно [61], что расширенная задача при $M = 2^{O(L)}$ эквивалентна (1). Эквивалентность понимается в том смысле, что если для оптимального решения задачи (20) $\hat{x}_{n+1} > 0$, то задача (1) несовместна, если $\mathbf{e}^{\mathbf{T}}\hat{\mathbf{x}} = M$, то в задаче (1) неограниченна целевая функция. В противном случае оптимальные решения совпадают. Для задачи (20) имеется допустимая стартовая точка

$$\hat{\mathbf{x}}^{\mathbf{1}} = \mathbf{e}, \; z^1 = -2^{O(L)},$$

с которой может начинаться вычислительный процесс. Однако в связи с очень большим значением величины M задача (20) является малопривлекательной с вычислительной точки зрения.

На практике большее распространение получили так называемые двухфазные алгоритмы. В них также рассматривается расширенная задача, эквивалентная (1):

$$\hat{\mathbf{c}}^{\mathbf{T}}\hat{\mathbf{x}} \to \min, \quad \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{b}, \quad \mathbf{d}^{\mathbf{T}}\hat{\mathbf{x}} = 0, \quad \hat{\mathbf{x}} \ge \mathbf{0}. \tag{21}$$

Здесь $\hat{\mathbf{x}} \in \mathbf{R}^{\mathbf{n+1}}$, $\hat{\mathbf{A}} = \left(\mathbf{A}, \mathbf{b} - \mathbf{A}\mathbf{x}^{\mathbf{1}}\right)$, $\hat{\mathbf{c}} = \begin{pmatrix} \mathbf{c} \\ 0 \end{pmatrix}$, $\mathbf{d} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$.

Вычислительный процесс может начинаться с любого вектора $\mathbf{x}^{\mathbf{1}}$ со всеми положительными компонентами. Выполнение ограничений-равенств $\mathbf{A}\mathbf{x}^{\mathbf{1}} = \mathbf{b}$ не требуется. Для $\hat{x}_j^1 = x_j^1$, $j = 1,\ldots,n$, $\hat{x}_{n+1}^1 = 1$ выполняются все ограничения задачи (21) кроме $\mathbf{d}^{\mathbf{T}}\hat{\mathbf{x}} = 0$. Идея алгоритма состоит в одновременной минимизации прямой потенциальной функции $f_1(\hat{\mathbf{x}}, z)$, записанной для расширенной задачи, и потенциальной функции ввода в допустимую область

$$f_3(\hat{\mathbf{x}}) = q \ln\left(\mathbf{d}^{\mathbf{T}}\hat{\mathbf{x}}\right) - \sum_{j=1}^{n+1} \ln\left(\hat{x}_j\right).$$

Подобные алгоритмы были разработаны, в частности, К.Анстрайхером [38] и М.Тоддом [68]. На основе этого же подхода Р.Фройнд разработал [43] алгоритм, использующий единую потенциальную функцию.

Еще один подход связан с методами “сдвинутого барьера”, первый из которых был также разработан Р.Фройндом [42]. Предлагается начинать вычислительный процесс с вектора $\mathbf{x}^{\mathbf{1}}$, у которого могут быть отрицательные компоненты, но для которого выполняются ограничения-равенства $\mathbf{A}\mathbf{x}^{\mathbf{1}} = \mathbf{b}$. Прямая потенциальная функция заменяется на следующую:

$$f_4(\mathbf{x}, z) = q \ln\left(\mathbf{c}^{\mathbf{T}}\mathbf{x} - z\right) - \sum_{j=1}^{n} \ln\left(x_j + h_j\left(\mathbf{c}^{\mathbf{T}}\mathbf{x} - z\right)\right).$$

Здесь вектор “сдвига” $\mathbf{h} > \mathbf{0}$ определяется из условия $\mathbf{x}^{\mathbf{1}} + \left(\mathbf{c}^{\mathbf{T}}\mathbf{x}^{\mathbf{1}} - z^1\right)\mathbf{h} > \mathbf{0}$.

Исследования методов уменьшения потенциала не ограничиваются упомянутыми работами. Всесторонние обзоры алгоритмов этого направления представлены К.Анстрайхером в [39] и М.Тоддом в [69].

В то же время алгоритмы уменьшения потенциала, хотя и обладают хорошими теоретическими оценками, на практике уступают алгоритмам, относящимся к следующему, наиболее перспективному направлению алго-

ритмов внутренних точек - алгоритмам оптимизации в конусе центрального пути (path-following algorithms).

Алгоритмы центрального пути

Истоки алгоритмов центрального пути восходят к идее К.Фриша [45] включения в целевую функцию штрафных слагаемых в виде логарифма ограничений-неравенств с параметром, монотонно уменьшающимся до нуля.

Рассматривается задача минимизации логарифмической барьерной функции на множестве допустимых решений пары задач (1)-(2)

$$f_\mu(\mathbf{x},\mathbf{u}) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u} - \mu\sum_{j=1}^{n}\ln\left(x_j g_j(\mathbf{u})\right) \to \min_{\mathbf{x}\in\mathbf{X},\mathbf{u}\in\mathbf{U}}. \qquad (22)$$

Ее точным решением при любом $\mu > 0$ является пара векторов $\mathbf{x}(\mu) \in ri\,\mathbf{X}$, $\mathbf{u}(\mu) \in ri\,\mathbf{U}$, для которой выполняется следующее условие:

$$x_j(\mu)g_j(\mathbf{u}(\mu)) = \mu,\ j = 1,\ldots,n.$$

Множество таких пар векторов $\mathbf{x}(\mu)$, $\mathbf{u}(\mu)$ при всех $\mu > 0$ образует центральный путь. Скаляр μ называется параметром центрального пути. Точки центрального пути при $\mu \to 0$ сходятся [21] к оптимальным решениям задач (1)-(2), а, точнее, к единственной точке $\mathbf{x}(0), \mathbf{u}(0)$, которой можно доопределить центральный путь при $\mu = 0$:

$$\mathbf{x}(0) = \arg\max_{\mathbf{x}\in ri\,\overline{\mathbf{X}}} \sum_{j\in J(\overline{\mathbf{X}})} \ln x_j\,, \qquad (23)$$

$$\mathbf{u}(0) = \arg\max_{\mathbf{u}\in ri\,\overline{\mathbf{U}}} \sum_{j\in J_0(\overline{\mathbf{X}})} \ln g_j(\mathbf{u}). \qquad (24)$$

Здесь $J(\overline{\mathbf{X}})$ - множество номеров компонент вектора $\mathbf{x}$ из $ri\,\overline{\mathbf{X}}$, имеющих положительные значения, а $J_0(\overline{\mathbf{X}})$ - множество номеров положительных компонент вектора $\mathbf{g}(\mathbf{u})$ при $\mathbf{u} \in ri\,\overline{\mathbf{U}}$.

В качестве другой интерпретации отметим, что точки центрального пути являются наиболее удаленными от границ множества допустимых

решений задачи (3) среди имеющих одно и то же значение целевой функции, равное $n\mu$. Действительно, пара $\mathbf{x}(\mu), \mathbf{u}(\mu)$ является решением задачи

$$\sum_{j=1}^{n} \ln x_j + \sum_{j=1}^{n} \ln g_j(\mathbf{u}) \to \max_{\mathbf{x}\in\mathbf{X},\mathbf{u}\in\mathbf{U}}, \quad \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u} = n\mu. \tag{25}$$

Удаленность здесь измеряется суммой логарифмов. Возможны и другие способы. В частности, можно определять расстояние от границ допустимого множества, исходя из чебышевской нормы, - тогда пара $\mathbf{x}(\mu), \mathbf{u}(\mu)$ определяется как решение задачи

$$\min_{j=1,\ldots,n} \min\left\{x_j, g_j(\mathbf{u})\right\} \to \max_{\mathbf{x}\in\mathbf{X},\mathbf{u}\in\mathbf{U}}, \quad \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u} = n\mu.$$

Чтобы отличать способ (25) определения точек центрального пути от других, иногда центральный путь также называют путем аналитических центров.

В контексте нелинейной оптимизации центральный путь и его свойства были всесторонне рассмотрены А.Фиакко и Г.Мак-Кормиком [33] в рамках исследования методов внутренних штрафных функций. Появление алгоритма Кармаркара [51] подтолкнуло исследователей к активному применению функций логарифмического барьера в задачах линейного программирования. Первой из таких работ стала работа [46], в которой П.Гиллом и соавторами, в частности, отмечалось сходство направлений корректировки переменных в алгоритме Кармаркара и при использовании подхода, связанного с логарифмическими барьерными функциями.

Нужно также отметить, что следование вдоль центрального пути не является прерогативой алгоритмов центрального пути. В частности, в алгоритме Ренегара [63], являющемся классическим алгоритмом уменьшения потенциала, вычислительный процесс происходит вдоль той же траектории.

Поскольку практически невозможно определить точные значения векторов $\mathbf{x}(\mu)$ и $\mathbf{u}(\mu)$ при заданном μ, то при конструировании реальных алгоритмов решения пары задач (1)-(2) можно использовать только

приближения к точкам центрального пути. Причем, чтобы гарантировать сходимость к оптимальным решениям, точность такого приближения должна нарастать с уменьшением значения параметра μ.

На основе исследований Н.Меджиддо [57] М.Коджимой, Ш.Мицуно и А.Йошисом была предложено [54] использовать расширение множества точек центрального пути, которое будем называть конусом центрального пути. Он состоит из всех пар векторов $(\mathbf{x},\mathbf{u})$, таких что $\mathbf{x} \in ri\,\mathbf{X}$, $\mathbf{u} \in ri\,\mathbf{U}$ и существует $\mu > 0$, при котором выполняется неравенство

$$\Phi_2(\mathbf{x},\mathbf{u},\mu) \le \theta\mu, \qquad (26)$$

где

$$\Phi_2(\mathbf{x},\mathbf{u},\mu) = \sum_{j=1}^{n} \frac{1}{\mu}\left(\mu - x_j g_j(\mathbf{u})\right)^2 .$$

Здесь θ - заданный неотрицательный параметр. Величину $\sqrt{\theta}$ можно интерпретировать как радиус конуса центрального пути. В частном случае, при $\theta = 0$, конус центрального пути совпадает с самим центральным путем. Положительные значения радиуса конуса центрального пути означают возможность отклонения от центрального пути по мере возрастания параметра μ.

Данное множество названо конусом центрального пути в связи с тем, что оно является конусом в пространстве переменных $\mathbf{z} = \mathbf{x} \otimes \mathbf{g}(\mathbf{u})$. Здесь знак $\otimes$ обозначает покомпонентное перемножение векторов. Действительно, из принадлежности вектора **z** конусу центрального пути, следует, что вектор $\lambda\mathbf{z}$ также принадлежит ему, поскольку $\lambda\mathbf{z}$ вместе со значением параметра центрального пути $\lambda\mu$ удовлетворяют неравенству (26).

Из выполнения условия (26) следует справедливость следующих неравенств:

$$\left(1-\sqrt{\theta}\right)\mu \le x_j g_j(\mathbf{u}) \le \left(1+\sqrt{\theta}\right)\mu, \;\; j = 1,\ldots,n . \qquad (27)$$

Неравенства (27) показывают, что $x_j g_j(\mathbf{u}) \to 0$ для всех $j = 1,\ldots,n$ при $\mu \to 0$, откуда следует, что если пары векторов $(\mathbf{x^k}, \mathbf{u^k})$ при $k = 1,2,3,\ldots$ принадлежат конусу центрального пути при соответствующих значениях μ^k, а $\lim\limits_{k\to\infty} \mu^k = 0$, то последовательности $\{\mathbf{x^k}\}$ и $\{\mathbf{u^k}\}$ сходятся [21] при $k \to \infty$ к оптимальным решениям задач (1)-(2), а конкретно к векторам $\mathbf{x}(0)$ и $\mathbf{u}(0)$, определенным в (23)-(24).

Если $\theta \in (0;1)$, а именно такие значения используются в рассматриваемых алгоритмах, то из (27), в частности, следует, что $x_j g_j(\mathbf{u}) > 0$ для всех $j = 1,\ldots,n$, и чтобы показать, что $\mathbf{x} \in \mathbf{X}$, а $\mathbf{u} \in \mathbf{U}$, необходимо проверить только выполнение равенства $\mathbf{Ax} = \mathbf{b}$ и положительность компонент одного из векторов: $\mathbf{x}$ или $\mathbf{g}(\mathbf{u})$.

Суть алгоритмов центрального пути состоит в том, что вырабатываются последовательности пар векторов $(\mathbf{x^k}, \mathbf{u^k})$, принадлежащих конусу центрального пути, и соответствующих значений μ^k, таких что для $\mathbf{x^k}, \mathbf{u^k}, \mu^k$ справедливо условие (26). При этом для некоторого $\beta > 0$ выполняется неравенство

$$\mu^{k+1} \le (1 - \beta/\sqrt{n})\mu^k . \qquad (28)$$

что обеспечивает [60] получение решения для пары задач (1)-(2) за $O(\sqrt{n}L)$ итераций.

Классическими алгоритмами данного направления являются, в частности, алгоритм, разработанный Коджимой, Мицуно и Йошисом [53], в котором значение β может быть не меньше, чем 0.125, а также алгоритм Р.Монтейро и И.Адлера [60], где максимальное значение β равно 0.35.

Поскольку задача (22) распадается на две формально несвязанные задачи - относительно вектора переменных $\mathbf{x}$

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} - \mu\sum_{j=1}^{n}\ln x_j \to \min,\ \mathbf{A}\mathbf{x} = \mathbf{b} \qquad (29)$$

и относительно вектора **u**

$$\mathbf{b}^{\mathbf{T}}\mathbf{u} + \mu\sum_{j=1}^{n}\ln g_j \to \max,\ \ \mathbf{g} + \mathbf{A}^{\mathbf{T}}\mathbf{u} = \mathbf{c}, \qquad (30)$$

получаем два подкласса алгоритмов: прямые и двойственные.

Прямые алгоритмы могут быть проинтерпретированы как процедуры приближенного решения методом Ньютона задачи (29) при итеративно уменьшающемся в соответствии с (28) значением параметра μ. Двойственные можно построить симметрично им на основе решения методом Ньютона задачи (30). Также существуют объединяющие в себе оба подхода прямо-двойственные алгоритмы.

Инициализация алгоритма

Одной из существенных проблем для алгоритмов центрального пути так же (и еще в большей степени), как и для алгоритмов уменьшения потенциала, является проблема инициализации алгоритма. Для данного класса алгоритмов стартовая точка должна быть не только относительно внутренней точкой множества допустимых решений пары задач (1)-(2), но и принадлежать конусу центрального пути, то есть для нее должно при некотором μ выполняться (26).

Первым подходом к инициализации алгоритма является подход, изложенный Монтейро и Адлером в [60]. Он заключается в решении расширенной задачи размерности $(m+1)\times(n+2)$ и двойственной к ней. Расширенная пара задач, эквивалентная (1)-(2), формируется в соответствии со следующими правилами:

$$\mathbf{A} = \left(\begin{array}{ccc|cc} a_{11} & \ldots & a_{1n} & 0 & b_1 - \lambda\sum a_{1j} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{mn} & 0 & b_m - \lambda\sum a_{mj} \\ \hline \alpha - c_1 & \ldots & \alpha - c_n & \alpha & 0 \end{array}\right), \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \ldots \\ b_m \\ K_b \end{pmatrix},$$

$$\mathbf{c}^{\mathbf{T}} = \begin{pmatrix} c_1 & \ldots & c_n & 0 & \alpha\lambda \end{pmatrix},$$

где $K_b = \alpha\lambda(n+1) - \lambda\sum c_j$, величины α и λ выбираются, исходя из условий

$$\lambda = 2^{2L},\ \alpha = 2^{4L} = \lambda^2, \qquad (31)$$

Имеется точка, принадлежащая конусу центрального пути и, следовательно, могущая рассматриваться в качестве стартовой:

$$\mathbf{x^1} = (\lambda, \lambda, \ldots, \lambda, 1),\ \mathbf{u^1} = (0,0,\ldots,0,-1),\ \mathbf{g^1} = (\alpha, \alpha, \ldots, \alpha, \alpha\lambda),\ \mu^1 = \alpha\lambda\,.$$

При выборе α и λ по правилам (31) возможны следующие случаи: либо оптимальные решения расширенной задачи таковы, что $\bar{x}_{n+2} = 0$ и $\bar{g}_{n+1} = 0$, тогда решения исходной и расширенной пар задач совпадают, либо $\bar{x}_{n+2} \neq 0$, тогда несовместна система ограничений исходной задачи, либо $\bar{g}_{n+1} \neq 0$, тогда неограниченна ее целевая функция.

Недостатком данного способа инициализации алгоритмов является сильная “перекошенность” расширенной задачи ввиду очень больших значений величин α и λ. На практике несколько сократить подобный эффект можно, присваивая им меньшие значения, что и было сделано при решении тестовых задач. Тем не менее существуют и другие подходы к инициализации.

Например, в [75] Йе, Тодд и Мицуно предложили и обосновали подход, где также производится переход от исходной задачи (1) к расширенной задаче, для которой известна точка центрального пути. Переход основан на следующей последовательности преобразований. Перенумеруем переменные задачи (1) таким образом, чтобы первые m столбцов матрицы **A** были линейно независимыми. Введем множества индексов $I = \{1,\ldots,m\}$ и $J = \{m+1,\ldots,n\}$. Таким образом,

$$\mathbf{A} = \left(\mathbf{A_I A_J}\right), \quad \mathbf{x} = \begin{pmatrix} \mathbf{x_I} \\ \mathbf{x_J} \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} \mathbf{c_I} \\ \mathbf{c_J} \end{pmatrix}.$$

Ограничение $\mathbf{Ax} = \mathbf{b}$ можно теперь переписать в виде $\mathbf{A_I x_I} + \mathbf{A_J x_J} = \mathbf{b}$, откуда

$$\mathbf{x_I} = \mathbf{A_I^{-1}}\left(\mathbf{b} - \mathbf{A_J x_J}\right). \tag{32}$$

Следовательно, целевая функция перепишется в виде

$$\mathbf{c^T x} = \mathbf{c_I^T x_I} + \mathbf{c_J^T x_J} = \mathbf{c_I^T A_I^{-1}}\left(\mathbf{b} - \mathbf{A_J x_J}\right) + \mathbf{c_J^T x_J} = \mathbf{c_I^T A_I^{-1} b} + \left(\mathbf{c_J} - \mathbf{A_J^T A_I^{-T} c_I}\right)^{\mathbf{T}} \mathbf{x_J}.$$

Опуская константу в целевой функции и преобразовав неравенство $\mathbf{x_I} \geq \mathbf{0}$, получим задачу, эквивалентную (1) и записанную в канонической форме

$$\left(\mathbf{c_J} - \mathbf{A_J^T A_I^{-T} c_I}\right)^{\mathbf{T}} \mathbf{x_J} \to \min, \quad -\mathbf{A_I^{-1} A_J x_J} \geq -\mathbf{A_I^{-1} b}, \quad \mathbf{x_J} \geq \mathbf{0}. \tag{33}$$

От (33) перейдем к кососимметрической задаче вида

$$\mathbf{q^T}\xi \to \min, \ \mathbf{M}\xi \geq -\mathbf{q}, \ \xi \geq \mathbf{0}. \tag{34}$$

M и **q** задаются следующим образом:

$$\mathbf{M} = \begin{pmatrix} \mathbf{0_{mm}} & -\mathbf{A_I^{-1} A_J} & \mathbf{A_I^{-1} b} & \overline{\mathbf{b}} \\ \left(\mathbf{A_I^{-1} A_J}\right)^{\mathbf{T}} & \mathbf{0_{(n-m)(n-m)}} & \mathbf{c_J} - \mathbf{A_J^T A_I^{-T} c_I} & \overline{\mathbf{c}} \\ \left(-\mathbf{A_I^{-1} b}\right)^{\mathbf{T}} & \left(\mathbf{A_J^T A_I^{-T} c_I} - \mathbf{c_J}\right)^{\mathbf{T}} & 0 & \beta \\ -\overline{\mathbf{b}}^{\mathbf{T}} & -\overline{\mathbf{c}}^{\mathbf{T}} & -\beta & 0 \end{pmatrix}, \ \mathbf{q} = \begin{pmatrix} \mathbf{0_m} \\ \mathbf{0_{n-m}} \\ 0 \\ n+2 \end{pmatrix},$$

где

$$\overline{\mathbf{b}} = \mathbf{e_m} - \mathbf{A_I^{-1} b} + \mathbf{A_I^{-1} A_J e_{n-m}},$$

$$\overline{\mathbf{c}} = \mathbf{e_{n-m}} - \mathbf{c_J} + \mathbf{A_J^T A_I^{-T} c_I} - \left(\mathbf{A_I^{-1} A_J}\right)^{\mathbf{T}} \mathbf{e_{n-m}},$$

$$\beta = 1 - \left(\mathbf{A_I^{-1} b}\right)^{\mathbf{T}} \mathbf{e_m} + \left(\mathbf{c_J} - \mathbf{A_J^T A_I^{-T} c_I}\right)^{\mathbf{T}} \mathbf{e_{n-m}}.$$

Здесь и ниже $\mathbf{e_m}$ будет обозначать вектор из единиц размерности m, $\mathbf{0_m}$ - нулевой вектор размерности m, $\mathbf{E_{mm}}$ - единичную матрицу размерности $m \times m$, а $\mathbf{0_{mm}}$ - матрицу, состоящую из нулей размерности $m \times m$.

Сведем кососимметрическую задачу (34) снова к стандартной форме:

$$\widetilde{\mathbf{c}}^{\mathbf{T}}\widetilde{\mathbf{x}} \to \min,\ \widetilde{\mathbf{A}}\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}},\ \widetilde{\mathbf{x}} \ge \mathbf{0}, \tag{35}$$

Двойственная к ней задача запишется в виде

$$\widetilde{\mathbf{b}}^{\mathbf{T}}\widetilde{\mathbf{u}} \to \max,\ \mathbf{g}\left(\widetilde{\mathbf{u}}\right) \equiv \widetilde{\mathbf{c}} - \widetilde{\mathbf{A}}^{\mathbf{T}}\widetilde{\mathbf{u}} \ge \mathbf{0}. \tag{36}$$

Здесь матрица $\widetilde{\mathbf{A}}$ и векторы $\widetilde{\mathbf{b}}$ и $\widetilde{\mathbf{c}}$ задаются в соответствии с формулами

$$\widetilde{\mathbf{A}} = \left(\mathbf{M},\ -\mathbf{E}_{(\mathbf{n}+\mathbf{2})(\mathbf{n}+\mathbf{2})}\right),\quad \widetilde{\mathbf{b}} = -\mathbf{q},\quad \widetilde{\mathbf{c}} = \begin{pmatrix} \mathbf{q} \\ \mathbf{0}_{\mathbf{n}+\mathbf{2}} \end{pmatrix}$$

Для задач (35)-(36) имеется априори известная пара векторов

$$\widetilde{\mathbf{x}}^{\mathbf{1}} = \mathbf{e}_{\mathbf{2}(\mathbf{n}+\mathbf{2})},\ \widetilde{\mathbf{u}}^{\mathbf{1}} = \mathbf{e}_{\mathbf{n}+\mathbf{2}},$$

принадлежащая центральному пути. Действительно, нетрудно проверить, что $\widetilde{\mathbf{g}}^{\mathbf{1}} = \mathbf{g}\left(\widetilde{\mathbf{u}}^{\mathbf{1}}\right) = \mathbf{e}_{\mathbf{2}(\mathbf{n}+\mathbf{2})}$. Соответствующее значение параметра центрального пути равно $\widetilde{\mu}^1 = 1$.

В [75] показано, что если для оптимального решения задачи (35) $\widetilde{x}_{n+1} = 0$, то одна из задач (1) или (2) является несовместной. Если же полученное решение $\widetilde{x}_{n+1}$ строго положительно, то оптимальные значения базисных переменных находятся как $\bar{x}_j = \widetilde{x}_{m+j} / \widetilde{x}_{n+1}$, $j = 1,\ldots,n-m$. Остальные переменные вычисляются по (32).

Очевидным недостатком описанной процедуры является увеличение числа переменных и ограничений. Вместо решения задачи размерности $m \times n$ приходится оперировать задачей размерности $(n+2) \times 2(n+2)$. Тем не менее данный способ достаточно эффективен и на практике, поскольку, в нем, в отличие от предыдущего, матрицы расширенной задачи не становятся плохо обусловленными. Кроме того, в [65] предложена модификация алгоритма, позволяющая организовать вычислительный процесс таким образом, что вместо непосредственного решения расширенной задачи на каждой итерации требуется решать три системы линейных уравнений с матрицами размерности $(m+2) \times (m+2)$.

Еще одним способом снятия проблемы инициализации являются так называемые алгоритмы, работающие в недопустимой области (infeasible-interior-point algorithms), активно разрабатываемые в последние годы. Их суть состоит в том, что вырабатываются последовательности векторов $\mathbf{x}^k, \mathbf{u}^k, \mathbf{g}^k$, для которых на каждой итерации выполняются в строгой форме ограничения-неравенства $\mathbf{x}^k > \mathbf{0}$ и $\mathbf{g}^k > \mathbf{0}$, но не выполняются ограничения-равенства $\mathbf{A}\mathbf{x}^k = \mathbf{b}$ и, для некоторых алгоритмов, $\mathbf{g}^k + \mathbf{A}^T\mathbf{u}^k = \mathbf{c}$.

Классическим для данного класса считается алгоритм [52], разработанный в 1993 году Коджимой, Меджиддо и Мицуно. Он сам, а особенно его последующие модификации (в частности, [62]), использующие техники приближенного вычисления направлений корректировки, оказались очень эффективными на практике. Несмотря на это, подобные алгоритмы долгое время не поддавались теоретическому анализу. Для них не удавалось получить не только полиномиальную оценку максимального объема вычислений, но даже и обоснование сходимости.

Существенным сдвигом в этом направлении явилась работа Р.Фройнда, Ф.Жарра и Ш.Мицуно [44], в которой была предложена такая модификация алгоритма [52], использующая приближенные вычисления, для которой удалось получить обоснование сходимости.

Подытожим вышесказанное. Первой задачей является сопоставление различных вариантов алгоритмов на задачах линейного программирования и на системах линейных уравнений и неравенств.

Среди аффинно-масштабирующих алгоритмов будем рассматривать два прямых (исторически первый и хорошо зарекомендовавший себя на практике вариант, а также вариант, использующий множители Лагранжа, вычисленные на предыдущей итерации, который уменьшает негативное влияние погрешностей вычислений в финальной стадии вычислительного

процесса), два двойственных, - которые предполагаются более предпочтительными для получения решения прямой задачи, и прямо-двойственный симметричный вариант алгоритма [19].

Среди полиномиальных алгоритмов остановимся на алгоритмах оптимизации в конусе центрального пути, как на наиболее перспективных с практической точки зрения. Ранее полиномиальные алгоритмы уступали по скорости лучшим из использующихся на практике, поскольку ради получения полиномиальных оценок приходится жертвовать вычислительными удобствами. Однако в последнее время благодаря разработке специальных процедур, ускоряющих сходимость, появилась надежда, что алгоритмы, обладающие “хорошей” гарантированной оценкой, смогут показать и высокую скорость решения практических задач.

При этом для алгоритмов центрального пути особенно актуальной остается проблема их инициализации. Одним из способов ее решения является создание алгоритмов, обладающих аналогичными свойствами, но не предъявляющих жестких требований к стартовому приближению.

Еще одной важной задачей, которой уделим особое внимание в диссертационной работе, является разработка специальных модификаций алгоритмов внутренних точек для решения систем линейных уравнений и неравенств. Данная задача как встречается на этапе ввода в допустимую область при решении задач линейного программирования, так и представляет самостоятельный интерес, в том числе, для решения нелинейных систем на базе итеративной линеаризации. Особо рассмотрим системы линейных уравнений и неравенств с интервальными ограничениями на переменные. Учет полезных свойств таких систем может существенно ускорить вычислительный процесс, в частности, решить проблему быстрой идентификации случая несовместности.