Введение

При решении многих задач из различных прикладных областей широкое применение находят методы линейной оптимизации. В частности, к задачам линейного программирования и тесно связанным с ними системам линейных уравнений и неравенств сводятся многие существенно нелинейные модели, при реализации которых используется итеративная линеаризация.

Как математическая дисциплина линейная оптимизация ведет отсчет с основополагающей работы Л.В. Канторовича [1], 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 [2]. 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, в 60–70-х годах зародилось альтернативное направление – методы внутренних точек, первый из которых был опубликован в 1967 году И.И. Дикиным [3]. Их название связано с тем, что, в отличие от симплекс-метода, перебирающего угловые точки многогранника допустимых решений, вычислительный процесс в методах внутренних точек происходит в относительной внутренности допустимого множества. Схематически траектории обоих классов алгоритмов изображены на рис.1.

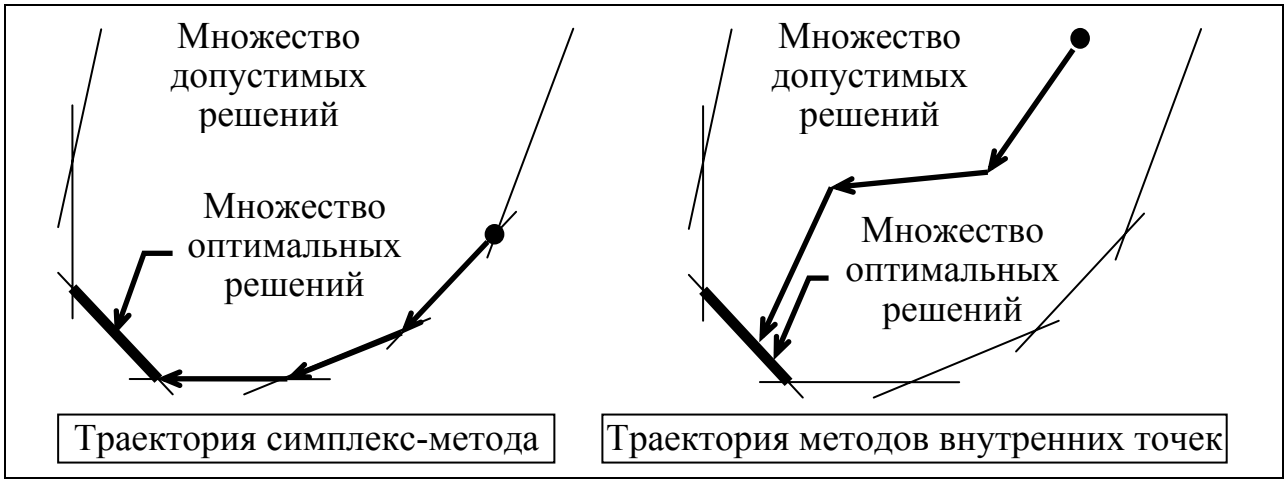


Рис.1. Траектории симплекс-метода и методов внутренних точек

Повышенный интерес к методам внутренних точек (результатом которого стали более 2000 опубликованных статей) в мировой науке возник вслед за созданием в 1984 году Н. Кармаркаром [4] первого алгоритма данного класса, обладающего полиномиальными гарантированными оценками максимального объема вычислений, необходимых для решения задачи.

1. Теоретические основы линейной оптимизации

Пара взаимно-двойственных задач линейного программирования

Рассмотрим пару 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

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

Задача (1) обычно называется прямой, а задача (2) – двойственной. К данным формам с помощью введения дополнительных переменных и ограничений можно привести любую задачу. Действительно, задачи минимизации и максимизации переходят друг в друга при умножении целевой функции на –1. Ограничения-равенства и ограничения-неравенства также легко преобразуются друг в друга с помощью несложных преобразований:

$$(Ax)_i \leq b_i \quad \Leftrightarrow \quad (Ax)_i + x_{n+1} = b_i,\ x_{n+1} \geq 0,$$

$$(Ax)_i \geq b_i \quad \Leftrightarrow \quad (Ax)_i - x_{n+1} = b_i,\ x_{n+1} \geq 0,$$

$$(Ax)_i = b_i \quad \Leftrightarrow \quad (Ax)_i \leq b_i,\ (Ax)_i \geq b_i.$$

Однако, поскольку при этом происходит увеличение размерности задачи, иногда лучше построить двойственную задачу непосредственно к исходной. В табл.1 продемонстрировано, как строится двойственная задача в произвольном случае.

Таблица 1. Соответствие прямой и двойственной задачи

	Прямая задача	**Двойственная задача**
Целевая функция	$\mathbf{c}^{\mathbf{T}}\mathbf{x} \to \min$	$\mathbf{b}^{\mathbf{T}}\mathbf{u} \to \max$
Ограничения 1	$(Ax)_i = b_i$ $(Ax)_i \geq b_i$ $(Ax)_i \leq b_i$	u_i – свободная переменная $u_i \geq 0$ $u_i \leq 0$
Ограничения 2	x_j – свободная переменная $x_j \geq 0$ $x_j \leq 0$	$c_j - (A^T u)_j = 0$ $c_j - (A^T u)_j \geq 0$ $c_j - (A^T u)_j \leq 0$

В то же время, если не оговаривается особо, нами будет решаться пара задач (1), (2).

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

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

Для любых $\mathbf{x} \in \mathbf{R}^{\mathbf{n}}$, $\mathbf{u} \in \mathbf{R}^{\mathbf{m}}$ при условии $\mathbf{Ax} = \mathbf{b}$ выполняется равенство

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

Действительно,

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

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.

Пусть для задач (1), (2) существуют допустимые решения, для которых все ограничения-неравенства выполняются в строгой форме, то есть имеются векторы $\mathbf{x} \in \mathbf{X}$, такие что $\mathbf{x} > \mathbf{0}$, и векторы $\mathbf{u} \in \mathbf{U}$, такие что $\mathbf{g}(\mathbf{u}) > \mathbf{0}$. Здесь и далее неравенство $\mathbf{x} > \mathbf{0}$ для вектора **x** будет обозначать, что все его компоненты положительные.

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

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

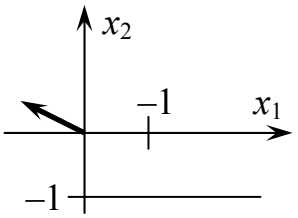
Приведенные факты теории двойственности, кроме последнего, получили обоснование в работе [5] Дж. фон Неймана. Последнее утверждение было впервые доказано в [6] и приобрело название теоремы Гольдмана-Таккера.

Продемонстрируем на численных примерах все 4 случая:

Пример 1.

$$\begin{cases} -2x_1 + x_2 \to \min, \\ x_2 = -1, \\ x_1 \ge 0,\ x_2 \ge 0. \end{cases} \qquad \begin{cases} -u \to \max, \\ -2 \ge 0, \\ 1 - u \ge 0. \end{cases}$$

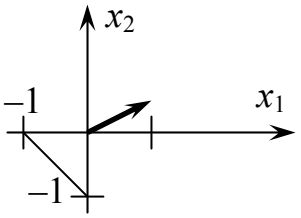
$\mathbf{X} = \varnothing,\ \overline{\mathbf{X}} = \varnothing.$ $\mathbf{U} = \varnothing,\ \overline{\mathbf{U}} = \varnothing.$



Пример 2.

$$\begin{cases} 2x_1 + x_2 \to \min, \\ x_1 + x_2 = -1, \\ x_1 \ge 0,\ x_2 \ge 0. \end{cases} \qquad \begin{cases} -u \to \max, \\ 2 - u \ge 0, \\ 1 - u \ge 0. \end{cases}$$

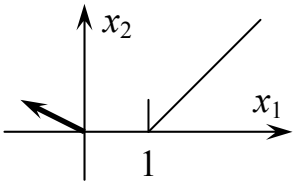
$\mathbf{X} = \varnothing,\ \overline{\mathbf{X}} = \varnothing.$ $\mathbf{U} \ne \varnothing,\ \overline{\mathbf{U}} = \varnothing.$



Пример 3.

$$\begin{cases} -2x_1 + x_2 \to \min, \\ x_1 - x_2 = -1, \\ x_1 \ge 0,\ x_2 \ge 0 \end{cases} \qquad \begin{cases} -u \to \max, \\ -2 - u \ge 0, \\ 1 + u \ge 0. \end{cases}$$

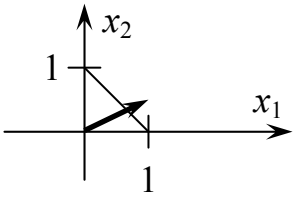
$\mathbf{X} \ne \varnothing,\ \overline{\mathbf{X}} = \varnothing.$ $\mathbf{U} = \varnothing,\ \overline{\mathbf{U}} = \varnothing.$



Пример 4.

$$\begin{cases} 2x_1 + x_2 \to \min, \\ x_1 + x_2 = 1, \\ x_1 \ge 0,\ x_2 \ge 0. \end{cases} \qquad \begin{cases} u \to \max, \\ 2 - u \ge 0, \\ 1 - u \ge 0. \end{cases}$$

$\mathbf{X} \ne \varnothing,\ \overline{\mathbf{X}} \ne \varnothing,\ \overline{\mathbf{x}}^{\mathbf{T}} = (0;1).$ $\mathbf{U} \ne \varnothing,\ \overline{\mathbf{U}} \ne \varnothing,\ \overline{\mathbf{u}} = (1).$



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

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

где

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

На основе этого правила И.И. Дикиным [3] в 1967 году был разработан первый из алгоритмов внутренних точек для задачи линейного программирования, в котором решение итеративно улучшается по правилу

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \lambda^k \Delta\mathbf{x^k}, \quad k = 1,2,\ldots, \tag{6}$$

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

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

а λ^k – шаг корректировки, определяемый равенством

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

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

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

Прямые аффинно-масштабирующие алгоритмы

В.И. Зоркальцевым в [7] 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 $\gamma \in (0;2/3)$, от точки $\mathbf{x^k}$ до границы положительного ортанта.

Во второй половине 80-х годов после ряда публикаций, где данный алгоритм вводился как модификация алгоритма Кармаркара, за ним закрепилось название affine-scaling method. В 1991 году японским математиком Т. Цучия была получено его полное теоретическое обоснование без предположения о невырожденности задачи.

Поскольку направление корректировки $\Delta\mathbf{x^k}$ является решением задачи

$$\mathbf{c^T}\left(\mathbf{x^k} + \Delta\mathbf{x}\right) + \frac{1}{2}\sum_{j=1}^{n}\frac{\Delta x_j^2}{d_j^k} \to \min_{\Delta\mathbf{x}\in\mathbf{R^n}}, \quad \mathbf{A}\left(\mathbf{x^k} + \Delta\mathbf{x}\right) = \mathbf{b}$$

или

$$\mathbf{c^T}\Delta\mathbf{x} + \frac{1}{2}\sum_{j=1}^{n}\frac{\Delta x_j^2}{d_j^k} \to \min_{\Delta\mathbf{x}\in\mathbf{R^n}}, \quad \mathbf{A}\Delta\mathbf{x} = \mathbf{0}. \tag{9}$$

то величины $1\big/d_j^k$ можно интерпретировать как штрафы, сдерживающие выход переменных на ближайшие границы области допустимых решений. Действительно, обозначив $\mathbf{D_k} = diag\left\{d_j^k\right\}$ и применив метод Лагранжа, получаем:

$$L(\Delta\mathbf{x},\mathbf{u}) = \mathbf{c^T}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x^T}\mathbf{D_k^{-1}}\Delta\mathbf{x} - \mathbf{u^T}\mathbf{A}\Delta\mathbf{x} \to \min_{\mathbf{x},\mathbf{u}}, \quad \mathbf{A}\Delta\mathbf{x} = \mathbf{0}.$$

$$\mathbf{c} + \mathbf{D_k^{-1}}\Delta\mathbf{x} - \mathbf{A^T}\mathbf{u} = \mathbf{0}, \quad \Delta\mathbf{x^k} = -\mathbf{D_k}\left(\mathbf{c} - \mathbf{A^T}\mathbf{u}\right),$$

что совпадает с формулой (7).

Вектор двойственных оценок $\mathbf{u^k}$ легко отыскать из равенства

$\mathbf{A\Delta x^k = AD_k A^T u - AD_k c = 0}$.

Таким образом, получим следующую последовательность действий, выполняемых на каждой итерации $k = 1,2,...$:

$$\mathbf{u^k} = \left(\mathbf{AD_k A^T}\right)^{-1}\mathbf{AD_k c}, \tag{10}$$

$$\Delta x_j^k = -d_j^k g_j\left(\mathbf{u^k}\right),\ j = 1,...,n, \tag{11}$$

$$\lambda^k = \gamma \min_{j:\Delta x_j^k < 0}\left(-x_j^k / \Delta x_j^k\right),\ \gamma \in (0;2/3), \tag{12}$$

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \lambda^k \Delta\mathbf{x^k}. \tag{13}$$

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 большие величины: $\gamma = 0{,}95$, $\gamma = 0{,}99$ и даже изменяющиеся по итерациям и асимптотически стремящиеся к единице величины γ^k.

Однако в 1997 году В. Маскаренас привел 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. Спустя 2 года было показано, что подобная ситуация может произойти при любом $\gamma \in (0{,}9107;1)$.

Способы задания весовых коэффициентов

Наряду с (5), используются и другие способы задания весовых коэффициентов d_j^k. Для сокращения негативных эффектов, связанных с большими значениями штрафов в финальной стадии вычислительного процесса, можно использовать алгоритм с весовыми коэффициентами

$$d_j^k = x_j^k,\ j = 1,...,n. \tag{14}$$

а также алгоритм с весовыми коэффициентами, вычисляемыми в соответствии со следующим обобщением правил (5) и (14):

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

Величина $N > 0$ здесь вводится в целях сужения диапазона весовых коэффициентов. В этом случае коэффициент γ, фигурирующий в формуле (12) вычисления шага корректировки, может принимать значения

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

В [8] 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

$$\overline{\sigma}(\alpha) \geq \underline{\sigma}(\alpha) > 0,\ \forall \alpha > 0, \tag{15}$$

$$\overline{\sigma}(\alpha) \leq M\alpha,\ \forall \alpha \in (0;\varepsilon) \tag{16}$$

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

В частности, коэффициенты d_j^k могут зависеть только от значений x_j^k.

Не является обязательным наличие конкретных выражений для функций $\overline{\sigma}$ и $\underline{\sigma}$. Достаточно иметь доказательства существования таких функций и выполнения для них свойств (15), (16). В уже упоминавшейся работе [8] было, в частности, предложено применять при $k > 1$ весовые коэффициенты, использующие вычисленные на предыдущей итерации векторы множителей Лагранжа ограничений-равенств вспомогательной задачи:

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

Такое правило, как и правило (14), позволяет уменьшить сильно негативное влияние погрешностей вычислений и увеличивает численную устойчивость алгоритма. В то же время, этот способ, по сравнению с (14) обладает существенно более быстрой сходимостью.

Рассмотрим одну вычислительную итерацию аффинно-масштабирующего алгоритма на примере 4. Вычислительный процесс начнем со стартового приближения $\left(\mathbf{x^1}\right)^{\mathbf{T}} = (0{,}5;\ 0{,}5)$. Тогда вектор весов будет равен $\left(\mathbf{d^1}\right)^{\mathbf{T}} = (0{,}25;\ 0{,}25)$.

$$2\Delta x_1 + \Delta x_2 + \frac{1}{2}\left(\frac{\Delta x_1^2}{0{,}25} + \frac{\Delta x_2^2}{0{,}25}\right) \to \min,\ \Delta x_1 + \Delta x_2 = 0.$$

$$\begin{array}{l} 2 + 4\Delta x_1 = u, \\ 1 + 4\Delta x_2 = u. \end{array} \Rightarrow \begin{array}{l} \Delta x_1 = -0{,}25(2-u), \\ \Delta x_2 = -0{,}25(1-u). \end{array}$$

$$-0{,}25(2-u) - 0{,}25(1-u) = 0,\ 0{,}5u - 0{,}75 = 0,\ \boxed{u^1 = 1{,}5}.$$

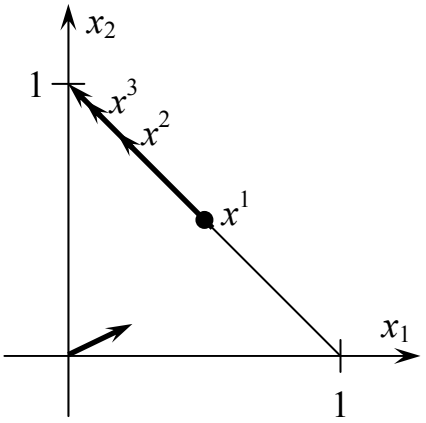
$$\Delta x_1^1 = -0{,}25(2-1{,}5) = -0{,}125,\ \Delta x_2^1 = -0{,}25(1-1{,}5) = 0{,}125.$$

$$\lambda^1 = \frac{2}{3}\left(-\frac{0{,}5}{-0{,}125}\right) = \frac{8}{3},\ \boxed{x_1^2 = 0{,}5 - \frac{8}{3}0{,}125 = \frac{1}{6}},\ \boxed{x_2^2 = 0{,}5 + \frac{8}{3}0{,}125 = \frac{5}{6}}.$$

В матричной форме нахождение двойственных оценок по правилу (10) выглядит следующим образом:

$$(1\ 1)\begin{pmatrix} 0{,}25 & 0 \\ 0 & 0{,}25 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} u = (1\ 1)\begin{pmatrix} 0{,}25 & 0 \\ 0 & 0{,}25 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix},\ 0{,}5u = 0{,}75.$$

Продемонстрируем работу алгоритма графически. Для рассматриваемого примера, начиная с $\mathbf{x^1}$, на каждой итерации происходит сокращение расстояния до точки оптимума в пропорции $1-\gamma$. В общем случае, алгоритм обеспечивает сходимость к точке оптимума $\bar{\mathbf{x}}$ со скоростью геометрической прогрессии:



$$\lim_{k\to\infty}\frac{\left\|\mathbf{x^{k+1}}-\bar{\mathbf{x}}\right\|}{\left\|\mathbf{x^{k}}-\bar{\mathbf{x}}\right\|}=1-\gamma\,.$$

Ввод в допустимую область

Если не имеется допустимого стартового приближения, вычислительный процесс можно начать с произвольной точки $\mathbf{x^1}>\mathbf{0}$. На каждой итерации вычисляется вектор невязок балансовых ограничений-равенств $\mathbf{r^k}=\mathbf{b}-\mathbf{Ax^k}$. Для определения направления корректировки вместо задачи (9) решается следующая:

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

Вектор двойственных переменных теперь вычисляется по формуле

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

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

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

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:

$$\lambda^k:=\min\left\{1,\lambda^k\right\}.$$

Выполним одну итерацию алгоритма на примере 4, начиная со стартового приближения $\left(\mathbf{x^1}\right)^{\mathbf{T}}=(1;1)$. Тогда вектор весов будет равен $\left(\mathbf{d^1}\right)^{\mathbf{T}}=(1;1)$.

$$\mathbf{r^1}=1-(1+1)=-1,$$

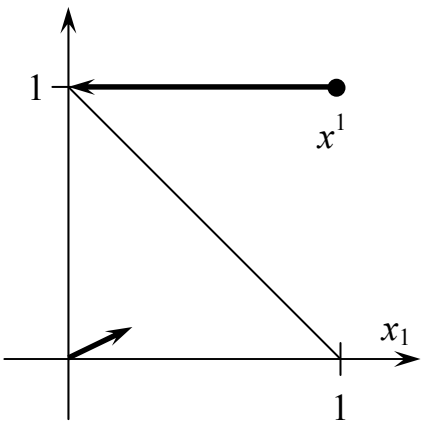
$$(1\ \ 1)\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix}u=(1\ \ 1)\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix}-1\,,$$

$$2u=3-1=2,\ \ u^1=1,$$

$$\Delta x_1^1=-1(2-1)=-1,$$

$$\Delta x_2^1=-1(1-1)=0\,.$$

Также можно изобразить работу алгоритма графически:



Двойственные аффинно-масштабирующие алгоритмы

В процессе решения вспомогательной задачи (9) на каждой итерации вырабатываются двойственные оценки $\mathbf{u^k}$, 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 связи с этим особенный интерес представляет использование двойственных алгоритмов, в которых последовательность приближений $\mathbf{x^k}$ быстрее, хотя и немонотонно, сходится к оптимальному решению задачи (1).

Двойственные аффинно-масштабирующие алгоритмы были введены в [8] и, независимо, в конце 80-х годов, рядом иностранных математиков. 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:

$$-\mathbf{b^T}\Delta\mathbf{u} + \frac{1}{2}\sum_{j=1}^{n}\frac{(\Delta g_j)^2}{d_j^k} \to \min_{\Delta\mathbf{u}\in\mathbf{R^m},\,\Delta\mathbf{g}\in\mathbf{R^n}}, \quad \Delta\mathbf{g} + \mathbf{A^T}\Delta\mathbf{u} = \mathbf{r^k}, \qquad (17)$$

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

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

Получим решение задачи (17):

$$-\mathbf{b^T}\Delta\mathbf{u} + \frac{1}{2}\left(\mathbf{r^k} - \mathbf{A^T}\Delta\mathbf{u}\right)^{\mathbf{T}}\mathbf{D_k^{-1}}\left(\mathbf{r^k} - \mathbf{A^T}\Delta\mathbf{u}\right) \to \min_{\Delta\mathbf{u}\in\mathbf{R^m}},$$

$$-\mathbf{b} - \mathbf{A}\mathbf{D_k^{-1}}\left(\mathbf{r^k} - \mathbf{A^T}\Delta\mathbf{u}\right) = \mathbf{0},$$

$$\Delta\mathbf{u^k} = \left(\mathbf{A}\mathbf{D_k^{-1}}\mathbf{A^T}\right)^{-1}\left(\mathbf{A}\mathbf{D_k^{-1}}\mathbf{r^k} + \mathbf{b}\right), \;\; \Delta\mathbf{g^k} = \mathbf{r^k} - \mathbf{A^T}\Delta\mathbf{u}.$$

Шаг корректировки вычисляется аналогично (12):

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

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 $\mathbf{r^k} \ne \mathbf{0}$, то следует пересчет:

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

Итеративный переход осуществляется по правилам

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

Множители Лагранжа $\mathbf{x^k}$ ограничений-равенств задачи (17) 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:

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

3. Полиномиальные алгоритмы

Исторический экскурс в полиномиальные алгоритмы

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.

Симплекс-метод, наиболее распространенный метод решения задач линейного программирования, не является полиномиальным. В частности, его применение на задаче Данцига

$$u_m \to \max,$$

$$\begin{cases} u_1 \in [-1;\ 1], \\ u_i \in \left[-2^{2i-2} - 2u_{i-1};\ 2^{2i-2} + 2u_{i-1}\right],\ i = 2,\ldots,m, \end{cases}$$

решение которой представляется в виде

$$\overline{u}_1 = 1,\ \overline{u}_i = 2\overline{u}_{i-1} + 2^{2i-2},\ i = 2,\ldots,m,$$

то есть

$$\overline{\mathbf{u}} = (1,6,28,120,496,...),$$

приводит к перебору всех 2^m вершин многогранника допустимых решений. Можно заметить, что число перебираемых вершин (то есть число итераций симплекс-метода) уже для задачи с 30 переменными превышает миллиард, что делает проблематичным ее решение симплекс-методом даже на самой современной технике.

Альтернативные способы перебора вершин не решают возникшую проблему – каков бы ни был алгоритм, можно придумать задачу, в которой оптимальной окажется именно последняя из перебираемых вершин.

В то же время в 1980 году в работе Л.Г. Хачияна на основе разработанной Н.З. Шором для выпуклого программирования техники построения последовательности сокращающихся в объеме множеств 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 этой техники впоследствии был разработан оригинальный класс полиномиальных алгоритмов погружения-отсечения, в последние годы существенно повысивших свою эффективность.

В 1984 году Н. Кармаркаром был создан первый полиномиальный алгоритм внутренних точек [4] 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, статья вызвала огромный интерес к алгоритмам внутренних точек. 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.

Полиномиальные алгоритмы внутренних точек строятся на основе использования логарифмических штрафных функций. Первый подход связан с так называемой потенциальной функцией. Потенциальная функция была введена Кармаркаром для обоснования полиномиальности своего алгоритма. Хотя алгоритм Кармаркара основывался на технике проективных преобразований, впоследствии алгоритмы уменьшения потенциала (potential reduction algorithms) выделились в самостоятельное направление алгоритмов внутренних точек.

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

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

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 $\mathbf{x}^{\mathbf{1}} \in \mathbf{X}$ 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, некоторые симметричную прямо-двойственную функцию Танабе-Тодда-Йе.

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

Использование последней позволило уменьшить теоретическую сложность 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 $\mathbf{A}$.

В то же время алгоритмы этого направления, хотя и обладают хорошими теоретическими оценками, на практике уступают алгоритмам, относящимся к наиболее перспективному направлению алгоритмов внутренних точек – алгоритмам оптимизации в конусе центрального пути (path-following algorithms).

Идейная основа и общая схема алгоритмов центрального пути

Истоки алгоритмов центрального пути восходят к идее К. Фриша 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)

$$f_\mu(\mathbf{x},\mathbf{u}) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{u} - \mu\sum_{j=1}^{n}\ln\left(x_j g_j(\mathbf{u})\right) \to \min_{\mathbf{x}\in\mathbf{X},\,\mathbf{u}\in\mathbf{U}}. \qquad (18)$$

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

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

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 к оптимальным решениям задач (1), (2).

Продемонстрируем поиск точки центрального пути для примера 4. Выпишем задачу минимизации логарифмической барьерной функции:

$$f_\mu(\mathbf{x},\mathbf{u}) = 2x_1 + x_2 - u - \mu\ln x_1 - \mu\ln x_2 - \mu\ln(2-u) - \mu\ln(1-u) \to \min,$$

$$x_1 + x_2 = 1.$$

Обозначим множитель Лагранжа ограничения-равенства $\tilde{u}$ и выпишем условия оптимальности Лагранжа:

$$\begin{cases} 2-\mu/x_1=\widetilde{u}, \\ 1-\mu/x_2=\widetilde{u}, \\ -1+\dfrac{\mu}{2-u}+\dfrac{\mu}{1-u}=0, \\ x_1+x_2=1. \end{cases} \Leftrightarrow \begin{cases} x_1(2-\widetilde{u})=\mu, \\ x_2(1-\widetilde{u})=\mu, \\ \dfrac{\mu}{2-u}+\dfrac{\mu}{1-u}=1, \\ x_1+x_2=1. \end{cases} \quad x_1=\frac{\mu}{2-\widetilde{u}}, \quad x_2=\frac{\mu}{1-\widetilde{u}}.$$

$$\begin{cases} \dfrac{\mu}{2-u}+\dfrac{\mu}{1-u}=1, \\ \dfrac{\mu}{2-\widetilde{u}}+\dfrac{\mu}{1-\widetilde{u}}=1. \end{cases} \Rightarrow u=\widetilde{u}, \Rightarrow \begin{cases} x_1 g_1(u)=\mu, \\ x_2 g_2(u)=\mu. \end{cases}$$

Для произвольного значения параметра μ можно найти точку центрального пути. Продемонстрируем решение этой задачи для $\mu=1$:

$$\frac{1}{2-u}+\frac{1}{1-u}=1, \quad \frac{1-u+2-u-2+3u-u^2}{(2-u)(1-u)}=0, \quad u^2-u-1=0.$$

$$u=\frac{1\pm\sqrt{5}}{2}, \; x_1=\frac{1}{2-\dfrac{1-\sqrt{5}}{2}}=\frac{2}{3+\sqrt{5}}=\frac{3-\sqrt{5}}{2}, \; x_2=\frac{1}{1-\dfrac{1-\sqrt{5}}{2}}=\frac{2}{1+\sqrt{5}}=\frac{\sqrt{5}-1}{2}.$$

При подстановке корня $u=(1+\sqrt{5})/2$ не будет выполнено условие $1-u>0$.

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

$$\sum_{j=1}^{n}\ln x_j+\sum_{j=1}^{n}\ln g_j(\mathbf{u})\to\max_{\mathbf{x}\in\mathbf{X},\,\mathbf{u}\in\mathbf{U}}, \; \mathbf{c}^{\mathbf{T}}\mathbf{x}-\mathbf{b}^{\mathbf{T}}\mathbf{u}=n\mu. \qquad (19)$$

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

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

Чтобы отличать способ (19) 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 [9] в рамках исследования методов внутренних штрафных функций. Появление алгоритма Кармаркара подтолкнуло исследователей к активному применению функций логарифмического барьера в задачах линейного программирования.

Поскольку для задач большой размерности 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.

В 1989 году М. Коджимой, Ш. Мицуно и А. Йошисом [10] было предложено использовать расширение множества точек центрального пути, которое будем называть конусом центрального пути. Он состоит из всех пар векторов $(\mathbf{x},\mathbf{u})$, таких что $\mathbf{x}\in\mathbf{X}$, $\mathbf{u}\in\mathbf{U}$ и существует $\mu>0$, при котором выполняется неравенство

$$\Phi_2(\mathbf{x},\mathbf{u},\mu)\le\theta\mu, \qquad (20)$$

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 $\mathbf{z}$ 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 (20).

Из выполнения условия (20) следует справедливость следующих неравенств:

$$\left(1-\sqrt{\theta}\right)\mu\le x_j g_j(\mathbf{u})\le\left(1+\sqrt{\theta}\right)\mu,\ \ j=1,\ldots,n. \qquad (21)$$

Неравенства (21) 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 $(\mathbf{x}^{\mathbf{k}},\mathbf{u}^{\mathbf{k}})$ 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 $\{\mathbf{x}^{\mathbf{k}}\}$ и $\{\mathbf{u}^{\mathbf{k}}\}$ сходятся при $k\to\infty$ к оптимальным решениям задач (1), (2).

Если $\theta\in(0;1)$, а именно такие значения используются в рассматриваемых алгоритмах, то из (21), 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 $\left(\mathbf{x^k}, \mathbf{u^k}\right)$, принадлежащих конусу центрального пути, и соответствующих значений μ^k, таких что для $\mathbf{x^k}, \mathbf{u^k}, \mu^k$ справедливо условие (20). При этом для некоторого $\beta > 0$ выполняется неравенство

$$\mu^{k+1} \le \left(1 - \beta/\sqrt{n}\right)\mu^k . \qquad (22)$$

что обеспечивает получение решения для пары задач (1), (2) за $O\left(\sqrt{n}L\right)$ итераций.

Классическими алгоритмами данного направления являются, в частности, алгоритм, разработанный М. Коджимой, Ш. Мицуно и А. Йошисом [10], в котором значение β может быть не меньше, чем 0,125, а также алгоритм Р. Монтейро и И. Адлера [11], где максимальное значение β равно 0,35.

Поскольку задача (18) распадается на две формально несвязанные задачи – относительно вектора переменных **x**

$$\mathbf{c^T x} - \mu \sum_{j=1}^{n} \ln x_j \to \min, \quad \mathbf{Ax = b} \qquad (23)$$

и относительно вектора **u**

$$\mathbf{b^T u} + \mu \sum_{j=1}^{n} \ln g_j \to \max, \quad \mathbf{g + A^T u = c}, \qquad (24)$$

получаем два подкласса алгоритмов: прямые и двойственные.

Прямые алгоритмы могут быть проинтерпретированы как процедуры приближенного решения методом Ньютона задачи (23) при итеративно уменьшающемся в соответствии с (22) значением параметра μ. Двойственные можно построить симметрично им на основе решения методом Ньютона задачи (24). Также существуют объединяющие в себе оба подхода прямо-двойственные алгоритмы.

Простейший вариант алгоритма центрального пути

Изложение алгоритмов начнем с наиболее простого варианта прямого алгоритма, итеративный переход в котором осуществляется по правилам

$$\mathbf{u^{k+1}} = \arg \min_{\mathbf{u} \in \mathbf{R^m}} \Phi_2\left(\mathbf{x^k}, \mathbf{u}, \mu^k\right) = \left(\mathbf{A X_k^2 A^T}\right)^{-1}\left(\mathbf{A X_k^2 C} - \mu^k \mathbf{b}\right), \qquad (25)$$

$$x_j^{k+1} = 2x_j^k - \frac{1}{\mu^k}\left(x_j^k\right)^2 g_j\left(\mathbf{u^{k+1}}\right), \quad j = 1,\ldots,n, \qquad (26)$$

$$\mu^{k+1} = \left(1 - \frac{\sqrt{\theta(1-\theta)}}{\sqrt{n-\theta}}\right)\mu^k . \qquad (27)$$

В формуле (25) использовано обозначение $\mathbf{X_k} = diag\left\{x_j^k\right\}$.

Идейной основой перехода (26) является применение квадратичной аппроксимации задачи (23) в точке $\mathbf{x} = \mathbf{x^k}$, $\mu = \mu^k$. Действительно, решая методом Лагранжа задачу

$$\mathbf{c}^{\mathbf{T}}\Delta\mathbf{x} - \mu^k \sum_{j=1}^{n}\left(\frac{\Delta x_j}{x_j^k} - \frac{1}{2}\left(\frac{\Delta x_j}{x_j^k}\right)^2\right) \to \min, \qquad (28)$$

$$\mathbf{A}\Delta\mathbf{x} = \mathbf{0}, \qquad (29)$$

получим

$$\Delta x_j^k = x_j^k - \frac{1}{\mu^k}\left(x_j^k\right)^2\left(\mathbf{c} - \mathbf{A}^{\mathbf{T}}\mathbf{u}^{\mathbf{k+1}}\right)_j,$$

где $\mathbf{u}^{\mathbf{k+1}}$ – вектор множителей Лагранжа ограничений (29), который вычисляется по формуле (25).

Из принадлежности точки $\left(\mathbf{x}^{\mathbf{k}}, \mathbf{u}^{\mathbf{k}}, \mu^k\right)$ конусу центрального пути следует, что точка $\left(\mathbf{x}^{\mathbf{k+1}}, \mathbf{u}^{\mathbf{k+1}}, \mu^{k+1}\right)$, полученная по формулам (25)–(27), также будет принадлежать конусу центрального пути.

Максимальное гарантированное сокращение параметра μ, означающее самую быструю скорость сходимости алгоритма, достигается, когда максимально значение величины $\theta(1-\theta)$, то есть при $\theta = 0{,}5$. В этом случае

$$\sqrt{\frac{\theta(1-\theta)}{n-\theta}} > \sqrt{\frac{\theta(1-\theta)}{n}} = \frac{0{,}5}{\sqrt{n}}.$$

Отсюда получаем, что в предлагаемом алгоритме (как и в последующих модификациях) уменьшение параметра скошенного пути осуществляется в соответствии с неравенством

$$\mu^{k+1} < \left(1 - 0{,}5/\sqrt{n}\right)\mu^k.$$

Рассмотрим итерацию алгоритма (25)–(27) для примера 4. Вычислительный процесс начнем с точки центрального пути $\left(\mathbf{x}^{\mathbf{1}}\right)^{\mathbf{T}} = (1/3;\, 2/3)$, $u^1 = 0$, $\left(\mathbf{g}^{\mathbf{1}}\right)^{\mathbf{T}} = (2-0;\, 1-0) = (2;\, 1)$, $\mu^1 = 2/3$.

$$(1\ \ 1)\begin{pmatrix}(1/3)^2 & 0\\ 0 & (2/3)^2\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix} u = (1\ \ 1)\begin{pmatrix}(1/3)^2 & 0\\ 0 & (2/3)^2\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix} - 2/3 * 1,$$

$$5/9\, u = 6/9 - 2/3 = 0, \quad u^2 = 0,$$

$$\mathbf{g}^{\mathbf{2}} = \begin{pmatrix}2-0\\1-0\end{pmatrix} = \begin{pmatrix}2\\1\end{pmatrix}, \quad \mathbf{x}^{\mathbf{2}} = \begin{pmatrix}2 * 1/3 - 3/2 * (1/3)^2 * 2\\ 2 * 2/3 - 3/2 * (2/3)^2 * 1\end{pmatrix} = \begin{pmatrix}1/3\\2/3\end{pmatrix},$$

$$\mu^2 = 0{,}8619 * 2/3 = \boxed{0{,}5746}.$$

Процедуры, ускоряющие вычислительный процесс

Разумным представляется на каждой итерации в максимальной степени уменьшать значение параметра скошенного пути. Наиболее простой модификацией алгоритма в этом направлении будет нахождение значения μ^{k+1} не по правилу (27), а путем решения вспомогательной задачи

$$\mu \to \min_{\mu \ge 0}, \quad \Phi_2\left(\mathbf{x}^{\mathbf{k+1}}, \mathbf{u}^{\mathbf{k+1}}, \mu\right) \le \theta\mu. \qquad (30)$$

Оптимальное значение здесь достигается при выполнении точного равенства

$$\Phi_2\left(\mathbf{x^{k+1}}, \mathbf{u^{k+1}}, \mu^{k+1}\right) = \theta\mu^{k+1}.$$

Таким образом, задача (30) сводится к определению наименьшего корня квадратного уравнения. Решение легко выводится аналитически и имеет вид

$$\mu^{k+1} = \frac{\sum_{j=1}^{n} x_j^{k+1} g_j\left(\mathbf{u^{k+1}}\right) - \sqrt{\left(\sum_{j=1}^{n} x_j^{k+1} g_j\left(\mathbf{u^{k+1}}\right)\right)^2 - (n-\theta)\sum_{j=1}^{n}\left(x_j^k\right)^2 g_j^2\left(\mathbf{u^{k+1}}\right)}}{n-\theta}.$$

Такое правило дает гарантированно не худшую сходимость, чем алгоритм с правилом пересчета (27). Действительно, μ^{k+1}, вычисляемое по формуле (27), является допустимым, но не оптимальным решением задачи (30).

Для примера 4 задача (30) будет иметь следующий вид:

$$\mu \to \min, \quad (\mu - 1/3 * 2)^2 + (\mu - 2/3 * 1)^2 \le 0{,}5\mu^2,$$

$$1{,}5\mu^2 - 8/3\,\mu + 8/9 \le 0, \quad \mu \in [12/27;\, 36/27], \quad \mu^2 = 12/27 = \boxed{0{,}4444}.$$

Дальнейшее усовершенствование алгоритмов связано с введением параметрических функций. Вместо фиксированного значения μ^k будем использовать априори неопределенный параметр μ. Закрепляется вектор $\mathbf{x^k}$ переменных прямой задачи и вводится линейная вектор-функция

$$\mathbf{u^k}(\lambda) = \arg\min_{\mathbf{u}\in\mathbf{R^m}} \Phi_2\left(\mathbf{x^k}, \mathbf{u}, \lambda\mu^k\right).$$

Для ее построения нужно вычислить два вектора, например, при $\lambda = 1$ и при $\lambda = 0$:

$$\mathbf{u^k}(1) = \left(\mathbf{A}\mathbf{X_k^2}\mathbf{A^T}\right)^{-1}\left(\mathbf{A}\mathbf{X_k^2}\mathbf{c} - \mu^k\,\mathbf{b}\right),$$

$$\mathbf{u^k}(0) = \left(\mathbf{A}\mathbf{X_k^2}\mathbf{A^T}\right)^{-1}\mathbf{A}\mathbf{X_k^2}\mathbf{c}.$$

Функция примет вид

$$\mathbf{u^k}(\lambda) = \lambda\,\mathbf{u^k}(1) + (1-\lambda)\mathbf{u^k}(0).$$

Обозначим через λ^k оптимальное решение задачи

$$\lambda \to \min_{\lambda\ge 0}, \quad \Phi_2\left(\mathbf{x^k}, \mathbf{u^k}(\lambda), \lambda\mu^k\right) \le \theta\lambda\mu^k. \qquad (31)$$

При этом ее ограничение выполняется в виде строгого равенства, то есть задача (31) сводится к вычислению корня квадратного уравнения и решается аналитически. Итеративный переход осуществляем по формулам

$$\mathbf{u^{k+1}} = \mathbf{u^k}\left(\lambda^k\right), \quad \mu^{k+1} = \lambda^k\mu^k, \quad x_j^{k+1} = 2x_j^k - \frac{1}{\mu^{k+1}}\left(x_j^k\right)^2 g_j\left(\mathbf{u^{k+1}}\right).$$

Проведем вычисления для примера 4:

$$u^1(1) = 9/5 * (6/9 - 2/3) = 0, \quad u^1(0) = 9/5 * 6/9 = 6/5,$$

$$u^1(\lambda) = \lambda * 0 + (1-\lambda) * 6/5 = 6/5 - 6/5\,\lambda,$$

$$g^1(\lambda) = \begin{pmatrix} 2 - (6/5 - 6/5\,\lambda) \\ 1 - (6/5 - 6/5\,\lambda) \end{pmatrix} = \begin{pmatrix} 6/5\,\lambda + 4/5 \\ 6/5\,\lambda - 1/5 \end{pmatrix}.$$

Задача (31) примет вид

$$\lambda \to \min, \quad \left(\frac{2}{3}\lambda - \frac{1}{3}\left(\frac{6}{5}\lambda + \frac{4}{5}\right)\right)^2 + \left(\frac{2}{3}\lambda - \frac{2}{3}\left(\frac{6}{5}\lambda - \frac{1}{5}\right)\right)^2 \le 0{,}5\left(\frac{2}{3}\lambda\right)^2,$$

$$\left(\frac{4\lambda - 4}{15}\right)^2 + \left(\frac{2 - 2\lambda}{15}\right)^2 \le \frac{2}{9}\lambda^2, \quad 3\lambda^2 + 4\lambda - 2 \ge 0,$$

$$\lambda \in \left(-\infty; \frac{-2-\sqrt{10}}{3}\right) \bigcup \left(\frac{-2+\sqrt{10}}{3}; +\infty\right).$$

Минимальное положительное значение составляет $\lambda = \left(-2 + \sqrt{10}\right)/3 \approx 0{,}3874$,

$\mu^2 = 0{,}3874 * 2/3 = \boxed{0{,}2583}$.

Видим, что с помощью использования параметризации можно сократить значение параметра центрального пути в существенно большей пропорции, чем это гарантируется оценкой (27).

Двойственные алгоритмы центрального пути

Можно построить алгоритмы, симметричные рассмотренным. Их идейной основой является решение методом Ньютона задачи максимизации двойственной логарифмической барьерной функции (24).

Квадратичная аппроксимация задачи (24) в точке $\mathbf{u} = \mathbf{u}^{\mathbf{k}}$, $\mu = \mu^k$ выглядит следующим образом:

$$\mathbf{b}^{\mathbf{T}}\Delta\mathbf{u} + \mu^k \sum_{j=1}^{n}\left(\frac{\Delta g_j}{g_j\left(\mathbf{u}^{\mathbf{k}}\right)} - \frac{1}{2}\left(\frac{\Delta g_j}{g_j\left(\mathbf{u}^{\mathbf{k}}\right)}\right)^2\right) \to \max_{\Delta\mathbf{u}}, \quad \Delta\mathbf{g} = -\mathbf{A}^{\mathbf{T}}\Delta\mathbf{u}. \qquad (32)$$

Решение задачи (32) находится по формулам

$$\Delta\mathbf{u}^{\mathbf{k}} = \frac{1}{\mu^k}\mathbf{r}^{\mathbf{k}} = \frac{1}{\mu^k}\left(\mathbf{A}\mathbf{G}_{\mathbf{k}}^{-\mathbf{2}}\mathbf{A}^{\mathbf{T}}\right)^{-\mathbf{1}}\left(\mathbf{b} - \mu^k\mathbf{A}\mathbf{G}_{\mathbf{k}}^{-\mathbf{1}}\mathbf{e}\right),$$

где, как и в дальнейшем, $\mathbf{G}_{\mathbf{k}} = diag\left\{g_j\left(\mathbf{u}^{\mathbf{k}}\right)\right\}$.

Вектор $\mathbf{x}^{\mathbf{k+1}}$ состоит из множителей Лагранжа ограничений-равенств из (32) и вычисляется по правилу

$$\mathbf{x}^{\mathbf{k+1}} = \mathbf{G}_{\mathbf{k}}^{-\mathbf{2}}\mathbf{A}^{\mathbf{T}}\mathbf{r}^{\mathbf{k}} + \mu^k\mathbf{G}_{\mathbf{k}}^{-\mathbf{1}}\mathbf{e}.$$

Можно попутно отметить, что он является одновременно решением задачи

$$\Phi_2\left(\mathbf{x}, \mathbf{u}^{\mathbf{k}}, \mu^k\right) \to \min_{\mathbf{x}\in\mathbf{R}^{\mathbf{n}}}, \quad \mathbf{A}\mathbf{x} = \mathbf{b},$$

что также подтверждает полную симметричность прямого и двойственного алгоритмов.

Так же, как и в прямых алгоритмах, можно использовать параметризацию. Закрепим вектор $\mathbf{u}^{\mathbf{k}}$ переменных двойственной задачи и введем линейную вектор-функцию

$$\mathbf{x}^{\mathbf{k}}(\lambda) = \arg\min_{\mathbf{x}\in\mathbf{R}^{\mathbf{n}}:\mathbf{A}\mathbf{x}=\mathbf{b}} \Phi_2\left(\mathbf{x}, \mathbf{u}^{\mathbf{k}}, \lambda\mu^k\right). \qquad (33)$$

На основе стандартной техники получим, что

$$\mathbf{x^k}(\lambda) = \mathbf{G_k^{-2}} \mathbf{A^T} \mathbf{r^k}(\lambda) + \lambda \mu^k \mathbf{G_k^{-1}} \mathbf{e}, \tag{34}$$

где

$$\mathbf{r^k}(\lambda) = \left(\mathbf{A G_k^{-2}} \mathbf{A^T}\right)^{-1} \left(\mathbf{b} - \lambda \mu^k \mathbf{A G_k^{-1}} \mathbf{e}\right). \tag{35}$$

λ^k находим как решение задачи

$$\lambda \to \min_{\lambda \geq 0}, \quad \Phi_2\left(\mathbf{x^k}(\lambda), \mathbf{u^k}, \lambda \mu^k\right) \leq \theta \lambda \mu^k. \tag{36}$$

После чего осуществляем итеративный переход:

$$\mathbf{x^{k+1}} = \mathbf{x^k}\left(\lambda^k\right), \quad \mu^{k+1} = \lambda^k \mu^k, \quad \mathbf{r^{k+1}} = \mathbf{r^k}\left(\lambda^k\right), \quad u_i^{k+1} = u_i^k + \frac{1}{\mu^{k+1}} r_i^{k+1}. \tag{37}$$

Из принадлежности точки $\left(\mathbf{x^k}, \mathbf{u^k}, \mu^k\right)$ конусу центрального пути следует, что точка $\left(\mathbf{x^{k+1}}, \mathbf{u^{k+1}}, \mu^{k+1}\right)$, полученная по формулам (33)–(37), также будет принадлежать конусу центрального пути. При этом параметр μ уменьшается не менее, чем в гарантированной пропорции (27).

Использование более высоких степеней при решении вспомогательной задачи

Дальнейшее развитие алгоритмов центрального пути связано с использованием более высоких степеней при решении вспомогательных задач (31) и (36). Введем функцию

$$\Phi_4(\mathbf{x}, \mathbf{u}, \mu) = \sum_{j=1}^{n} \frac{1}{\mu^2} \left(\mu - x_j g_j(\mathbf{u})\right)^4.$$

Отличие модифицированного прямого алгоритма от исходного заключается в том, что величину λ^k, показывающую степень уменьшения параметра μ находим не по правилу (31), а как решение следующей задачи:

$$\lambda \to \min_{\lambda \geq 0}, \quad \Phi_4\left(\mathbf{x^k}, \mathbf{u^k}(\lambda), \lambda \mu^k\right) \leq \theta^2 \lambda^2 \left(\mu^k\right)^2. \tag{38}$$

В модифицированном двойственном алгоритме вместо (36) решается задача

$$\lambda \to \min_{\lambda \geq 0}, \quad \Phi_4\left(\mathbf{x^k}(\lambda), \mathbf{u^k}, \lambda \mu^k\right) \leq \theta^2 \lambda^2 \left(\mu^k\right)^2. \tag{39}$$

В остальном вычислительный процесс обоих алгоритмов остается прежним.

Поскольку применение четвертой степени позволяет сильнее уменьшать значение параметра центрального пути μ на каждой итерации, можно надеяться на более быструю сходимость алгоритма. Конечно, скорость сходимости алгоритмов, использующих правила (38) и (39), выше скорости сходимости исходных вариантов алгоритмов только при одном и том же начальном приближении на данной итерации. Поскольку вычислительные процессы для разных алгоритмов приводят к разным траекториям, то локальный выигрыш на одной итерации не означает гарантированного ускорения процесса в целом. Однако экспериментальное исследование [12] показало, что на практических задачах алгоритмы с (38) и (39) действительно работает быстрее.

Поскольку эксперименты показали, что применение четвертой степени при решении вспомогательных задач, кроме гарантированно не худших оценок уменьшения параметра μ в пределах одной итерации, дает существенное ускорение вычислительного процесса на практике, была осуществлена попытка использования еще более высоких степеней.

Рассмотрим функцию

$$\Phi_p(\mathbf{x},\mathbf{u},\mu)=\sum_{j=1}^{n}\frac{1}{\mu^{p/2}}\left(\mu-x_j g_j(\mathbf{u})\right)^p,\quad p=8,16,...$$

Вместо задачи (31) будем решать следующую:

$$\lambda\to\min_{\lambda\ge 0},\quad \Phi_p\left(\mathbf{x}^{\mathbf{k}},\mathbf{u}^{\mathbf{k}}(\lambda),\lambda\mu^k\right)\le\theta^{p/2}\lambda^{p/2}\left(\mu^k\right)^{p/2}. \qquad (40)$$

Отдельный интерес здесь представляет использование p, стремящегося к бесконечности. В этом случае задача (40) запишется следующим образом:

$$\lambda\to\min_{\lambda\ge 0},\quad \max_{j=1,...,n}\left|\lambda\mu^k-x_j^k g_j\left(\mathbf{u}^{\mathbf{k}}(\lambda)\right)\right|\le\sqrt{\theta}\lambda\mu^k.$$

Симметричные аналоги можно построить и применительно к двойственным алгоритмам.

В связи с возможностью существенных расхождений точек минимума функции $\Phi_p(\mathbf{x},\mathbf{u},\mu)$ и функции $\Phi_2(\mathbf{x},\mathbf{u},\mu)$, использующейся при нахождении параметрической вектор-функции $\mathbf{u}^{\mathbf{k}}(\lambda)$ для алгоритмов с $p>4$, не удается гарантировать принадлежность новой точки конусу скошенного пути, а именно, справедливость неравенства

$$\Phi_2\left(\mathbf{x}^{\mathbf{k+1}},\mathbf{u}^{\mathbf{k+1}},\mu^{k+1}\right)\le\theta\mu^{k+1},$$

а также гарантированного уменьшение параметра скошенного пути.

Защититься от «выпадения» из конуса скошенного пути для алгоритмов с $p>4$ можно с помощью следующей проверки: если

$$\Phi_2\left(\mathbf{x}^{\mathbf{k+1}},\mathbf{u}^{\mathbf{k+1}},\mu^{k+1}\right)>\theta\mu^{k+1}$$

или для найденного λ^k выполняется неравенство

$$\lambda^k>1-\frac{\sqrt{\theta}\sqrt{1-\theta}}{\sqrt{n-\theta}},$$

то заново осуществляется поиск λ^k на основе $p=4$ и выполняется итеративный переход. Поскольку нет необходимости заново вычислять параметрическую функцию $\mathbf{u}^{\mathbf{k}}(\lambda)$, то данная процедура не увеличивает алгоритмическую сложность вычислительного процесса.

В то же время самостоятельный интерес представляют алгоритмы, в которых обязательное нахождение в конусе скошенного пути на всех итерациях заменяется более слабыми условиями

$$\mathbf{A}\mathbf{x}^{\mathbf{k}}=\mathbf{b},\ \mathbf{x}^{\mathbf{k}}>\mathbf{0},\ \mathbf{g}\left(\mathbf{u}^{\mathbf{k}}\right)>\mathbf{0},\quad \max_{j=1,...,n}\left|\mu^k-x_j^k g_j\left(\mathbf{u}^{\mathbf{k}}\right)\right|\le\sqrt{\theta}\mu^k. \qquad (41)$$

Возврат осуществляется только в случае, если какое-то из условий (41) нарушается. Однако подобные ситуации, если и возможны, то крайне редки, и в процессе экспериментального исследования [12] не возникали ни разу.

Проблема инициализации алгоритмов. Два способа ее решения путем перехода к расширенным задачам специального вида

Одной из существенных проблем для алгоритмов центрального пути 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 (20).

Первым подходом к инициализации алгоритма является подход, изложенный Р. Монтейро и И. Адлером в [11]. 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:

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

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

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

$$\lambda=2^{2L},\ \alpha=2^{4L}=\lambda^2, \qquad (42)$$

L (здесь и в дальнейшем) – объем входных данных задачи, который находится по формуле

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

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

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

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

При выборе α и λ по правилам (42) 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. Даже для примера 4 получим следующие значения:

$L = \ln 2 + \ln 3 + \ln 2 + \ln(1+2) = \ln 36 \approx 3{,}58$,

$\lambda = 2^{\ln 36} \approx 144$, $\alpha = \lambda^2 \approx 20653$, $\alpha\lambda \approx 2968111$, $K_b \approx 8903902$.

На практике несколько сократить подобный эффект можно, присваивая константам α и λ меньшие значения, что и было сделано при решении тестовых задач. Тем не менее, существуют и другие подходы к инициализации.

Например, в 1994 году 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,

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

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

$$\mathbf{x_I} = \mathbf{A_I^{-1}}(\mathbf{b} - \mathbf{A_J x_J}). \qquad (43)$$

Следовательно, целевая функция перепишется в виде

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

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

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

Отсюда перейдем к кососимметрической задаче вида

$$\mathbf{q^T}\xi \to \min,\ \ \mathbf{M}\xi \ge -\mathbf{q},\ \ \xi \ge \mathbf{0}. \qquad (44)$$

M и **q** задаются следующим образом:

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

где

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

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

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

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 (44) снова к стандартной форме:

$$\tilde{\mathbf{c}}^{\mathbf{T}} \tilde{\mathbf{x}} \to \min, \ \tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \ \tilde{\mathbf{x}} \ge \mathbf{0}, \qquad (45)$$

Двойственная к ней задача запишется в виде

$$\tilde{\mathbf{b}}^{\mathbf{T}} \tilde{\mathbf{u}} \to \max, \ \mathbf{g}\left(\tilde{\mathbf{u}}\right) \equiv \tilde{\mathbf{c}} - \tilde{\mathbf{A}}^{\mathbf{T}} \tilde{\mathbf{u}} \ge \mathbf{0}. \qquad (46)$$

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

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

Для задач (45), (46) имеется априори известная пара векторов

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

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

Если для оптимального решения задачи (45) $\tilde{x}_{n+1} = 0$, то одна из задач (1) или (2) является несовместной. Если же полученное решение $\tilde{x}_{n+1}$ строго положительно, то оптимальные значения свободных переменных находятся как $\bar{x}_j = \tilde{x}_{m+j} / \tilde{x}_{n+1}$, $j = 1,...,n-m$. Базисные переменные вычисляются по (43).

Продемонстрируем эту процедуру на следующем численном примере:

Пример 5.

$$\begin{cases} x_1 + x_2 + x_3 \to \min, \\ x_1 - x_2 = 1, \quad x_3 = 1, \\ x_1 \ge 0, \ x_2 \ge 0, \ x_3 \ge 0. \end{cases} \quad \mathbf{c}^{\mathbf{T}} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \quad \begin{matrix} m = 2, \\ n - m = 1, \\ n = 3. \end{matrix}$$

Линейно независимыми являются, например, первый и третий столбцы матрицы **A**. Выразим соответствующие базисные переменные x_1 и x_3 через оставшуюся свободную переменную x_2:

$$\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 + x_2 \\ 1 \end{pmatrix} \ge \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Подставим эти значения в целевую функцию, также учтем только что выписанные ограничения:

$$\left(1 + x_2\right) + x_2 + 1 = 2x_2 + 2 \to \min,$$

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} x_2 \ge \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \ x_2 \ge 0.$$

Сформируем кососимметрическую задачу вида (44):

$$\mathbf{M}=\begin{pmatrix} \begin{pmatrix}0&0\\0&0\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}1\\1\end{pmatrix} & \begin{pmatrix}-1\\0\end{pmatrix} \\ \begin{pmatrix}-1&0\end{pmatrix} & 0 & 2 & 0 \\ \begin{pmatrix}-1&-1\end{pmatrix} & -2 & 0 & 5 \\ \begin{pmatrix}1&0\end{pmatrix} & 0 & -5 & 0 \end{pmatrix}, \qquad q=\begin{pmatrix}\begin{pmatrix}0\\0\end{pmatrix}\\0\\0\\5\end{pmatrix}.$$

В матрице **M** по диагонали стоят нулевые элементы и матрицы. Далее будут располагаться матрица коэффициентов при свободных переменных $\begin{pmatrix}1\\0\end{pmatrix}$ и умноженный на –1 вектор правых частей ограничений $\begin{pmatrix}1\\1\end{pmatrix}$. В следующей строке находится вектор коэффициентов целевой функции. Для данной задачи он состоит из одного числа 2. Последний столбец составляется как разность единицы и суммы элементов каждой строки. Вектор **q** является нулевым, кроме последней компоненты $n+m=5$.

После сведения к стандартной форме получим задачу размерности 5×10, для которой известна точка центрального пути – единичные векторы $\tilde{\mathbf{x}}^{\mathbf{1}}$ и $\tilde{\mathbf{g}}^{\mathbf{1}}$ совместно со значением $\mu^1=1$. Решением данной задачи будет вектор $\tilde{\mathbf{x}}^{\mathbf{T}}=(0\ \ 0\ \ 0\ \ 4/5\ \ 0\ \ 4/5\ \ 4/5\ \ 8/5\ \ 0\ \ 1)$. Поскольку $\tilde{x}_{n+1}=\tilde{x}_4=4/5\neq 0$, то исходная задача является совместной. Оптимальное значение свободной переменной x_2 равно $\bar{x}_2=\tilde{x}_3/\tilde{x}_4=0$. Отсюда $\bar{x}_1=1+0=1,\ \bar{x}_3=1$. Задача решена.

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 1997 году К. Русом, Т. Терлаки, Ж.-Ф. Виалом была 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 $(m+2)\times(m+2)$.

Алгоритмы скошенного пути

Есть и принципиально другой способ решить проблему инициализации для полиномиальных алгоритмов – использовать предложенный и обоснованный в [12] класс алгоритмов оптимизации в конусе «скошенного пути». Вычислительный процесс в них может начинаться с любых относительно внутренних точек множеств допустимых решений задач (1), (2).

В [12] доказано, что для любого заданного вектора $\mathbf{t}>\mathbf{0}$ существует и единственна такая пара векторов $\mathbf{x}(\mathbf{t}),\mathbf{u}(\mathbf{t})$, что выполняются условия

$$\mathbf{x}(\mathbf{t})\in\mathbf{X},\ \ \mathbf{u}(\mathbf{t})\in\mathbf{U},\ \ x_j(\mathbf{t})g_j(\mathbf{u}(\mathbf{t}))=t_j,\ \ j=1,\ldots,n.$$

Скошенным путем, инициируемым вектором $\mathbf{t}>\mathbf{0}$, назовем множество

пар векторов $\mathbf{x}(\mu\mathbf{t}), \mathbf{u}(\mu\mathbf{t})$ при всех $\mu > 0$. В частности, вектор $\mathbf{t} = \mathbf{e}$ (равно как и $\mathbf{t} = \lambda\mathbf{e}$ при любом $\lambda > 0$) инициирует центральный путь.

Каждый скошенный путь характеризуется коэффициентом скошенности $\gamma = \bar{t}/t_{\min}$,

где $\bar{t} = \frac{1}{n}\sum_{j=1}^{n} t_j$, $t_{\min} = \min_{j=1,\ldots,n} t_j$.

Поскольку для пути аналитических центров все компоненты инициирующего вектора равны между собой, его коэффициент скошенности равен единице. Для всех остальных скошенных путей он больше единицы.

Все предлагаемые алгоритмы вырабатывают последовательности пар векторов $\left(\mathbf{x}^{\mathbf{k}}, \mathbf{u}^{\mathbf{k}}\right)$ и соответствующих значений μ^k, для которых при заданном $\theta \in (0;1)$ выполняются условия принадлежности конусу скошенного пути:

$$\mathbf{x}^{\mathbf{k}} \in \mathbf{X}, \mathbf{u}^{\mathbf{k}} \in \mathbf{U},$$

$$\widetilde{\Phi}_2\left(\mathbf{x}^{\mathbf{k}}, \mathbf{u}^{\mathbf{k}}, \mu^k\right) \equiv \sum_{j=1}^{n} \frac{1}{\mu^k t_j}\left(\mu^k t_j - x_j^k g_j\left(\mathbf{u}^{\mathbf{k}}\right)\right)^2 \le \theta \mu^k t_{\min}.$$

Схематически траектория алгоритмов скошенного пути показана на рис. 2.

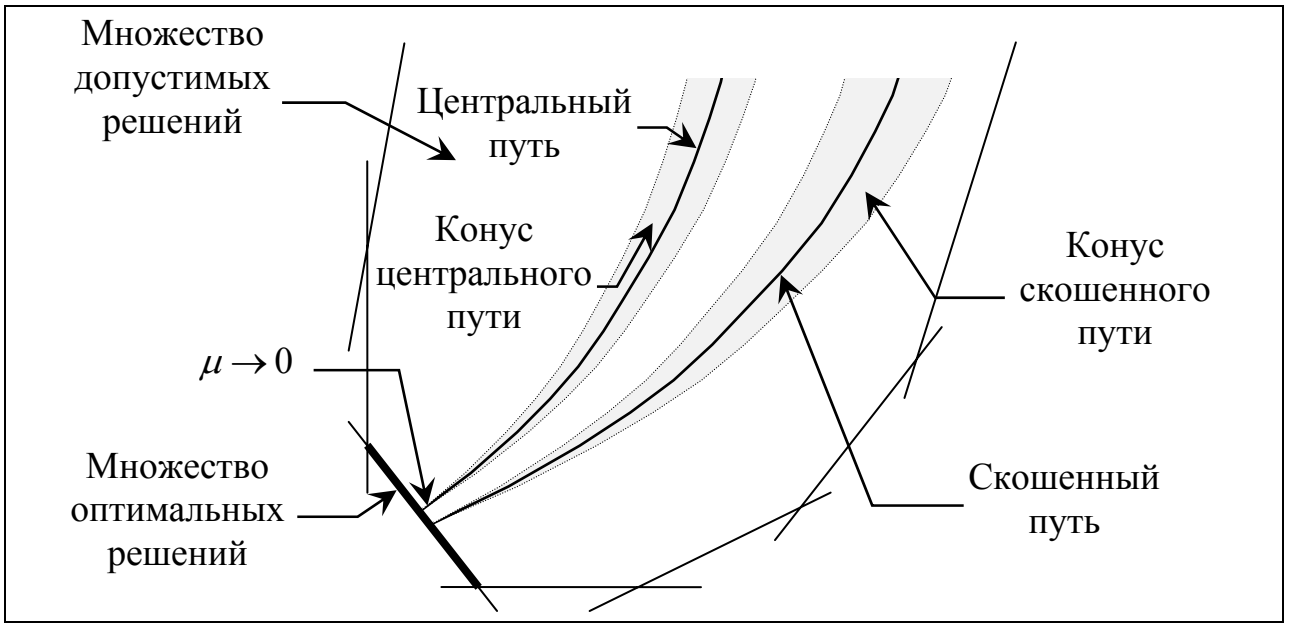


Рис.2. Траектория алгоритмов скошенного пути

Поскольку любая пара векторов $\left(\mathbf{x}^{\mathbf{1}}, \mathbf{u}^{\mathbf{1}}\right)$, лежащих в относительной внутренности допустимого множества, принадлежит скошенному пути, инициируемому вектором с компонентами $t_j = x_j^1 g_j\left(\mathbf{u}^{\mathbf{1}}\right)$, то проблема инициализации для алгоритмов скошенного пути существенно ослабляется.

Формулы алгоритмов скошенного пути похожи на формулы алгоритмов центрального пути. В частности, в прямых алгоритмах закрепляется вектор $\mathbf{x}^{\mathbf{k}}$ переменных прямой задачи и вводится линейная вектор-функция

$$\mathbf{u^k}(\lambda) = \arg\min_{\mathbf{u}\in\mathbf{R^m}} \widetilde{\Phi}_2\left(\mathbf{x^k}, \mathbf{u}, \lambda\mu^k\right) = \left(\mathbf{AX_k^2M_k^{-1}A^T}\right)^{-1}\left(\mathbf{AX_k^2M_k^{-1}c} - \lambda\,\mathbf{b}\right),$$

где $\mathbf{M_k} = diag\left\{\mu^k t_j\right\}$.

λ^k находится как решение задачи

$$\lambda \to \min_{\lambda\ge 0}, \quad \widetilde{\Phi}_p\left(\mathbf{x^k}, \mathbf{u^k}(\lambda), \lambda\mu^k\right) \le \theta^{p/2}\lambda^{p/2}\left(\mu^k\right)^{p/2}\left(t_{\min}\right)^{p/2},$$

где

$$\widetilde{\Phi}_p(\mathbf{x}, \mathbf{u}, \mu) = \sum_{j=1}^{n} \frac{1}{\mu^{p/2}\left(t_j\right)^{p/2}}\left(\mu t_j - x_j g_j(\mathbf{u})\right)^p,$$

Итеративный переход осуществляется по формулам

$$\mathbf{u^{k+1}} = \mathbf{u^k}\left(\lambda^k\right),\ \mu^{k+1} = \lambda^k\mu^k,\ x_j^{k+1} = 2x_j^k - \frac{1}{\mu^{k+1}t_j}\left(x_j^k\right)^2 g_j\left(\mathbf{u^{k+1}}\right).$$

В двойственных алгоритмах вычислительный процесс осуществляется по правилам

$$\mathbf{x^k}(\lambda) = \arg\min_{\mathbf{x}\in\mathbf{R^n}:\,\mathbf{Ax}=\mathbf{b}} \widetilde{\Phi}_2\left(\mathbf{x}, \mathbf{u^k}, \lambda\mu^k\right) = \mathbf{G_k^{-2}TA^T r^k}(\lambda) + \lambda\mu^k\mathbf{G_k^{-1}t}.$$

где

$$\mathbf{r^k}(\lambda) = \left(\mathbf{AG_k^{-2}TA^T}\right)^{-1}\left(\mathbf{b} - \lambda\mu^k\mathbf{AG_k^{-1}t}\right),$$

$$\mathbf{G_k} = diag\left\{g_j\left(\mathbf{u^k}\right)\right\},\ \mathbf{T} = diag\left\{t_j\right\}.$$

λ^k определяется как решение задачи

$$\lambda \to \min_{\lambda\ge 0}, \quad \widetilde{\Phi}_p\left(\mathbf{x^k}(\lambda), \mathbf{u^k}, \lambda\mu^k\right) \le \theta^{p/2}\lambda^{p/2}\left(\mu^k\right)^{p/2}\left(t_{\min}\right)^{p/2}.$$

После чего происходит итеративный переход:

$$\mathbf{x^{k+1}} = \mathbf{x^k}\left(\lambda^k\right),\ \mu^{k+1} = \lambda^k\mu^k,\ \mathbf{r^{k+1}} = \mathbf{r^k}\left(\lambda^k\right),\ u_i^{k+1} = u_i^k + \frac{1}{\mu^{k+1}}r_i^{k+1}.$$

В [12] представлено теоретическое обоснование прямых и двойственных алгоритмов скошенного пути с $p = 2$ и $p = 4$. Сходимость обеспечивается гарантированным сокращением параметра μ в соответствии с неравенством

$$\mu^{k+1} < \left(1 - 0{,}5/\sqrt{n\gamma}\right)\mu^k.$$

Процедура сокращения коэффициента скошенности

Поскольку меньшие значения коэффициента скошенности способствуют ускорению сходимости вычислительного процесса, предлагается осуществлять по итерациям переход из конуса скошенного пути K1 в конус другого – K2, обладающего меньшим коэффициентом скошенности. Схематично такой переход изображен на рис. 3. Здесь полученное на *k*-итерации приближение ($\mathbf{x^k}$, $\mathbf{u^k}$) принадлежит обоим конусам – K1 и K2, при этом коэффициент скошенности пути, для которого построен конус K1, меньше аналогичного для конуса K2.

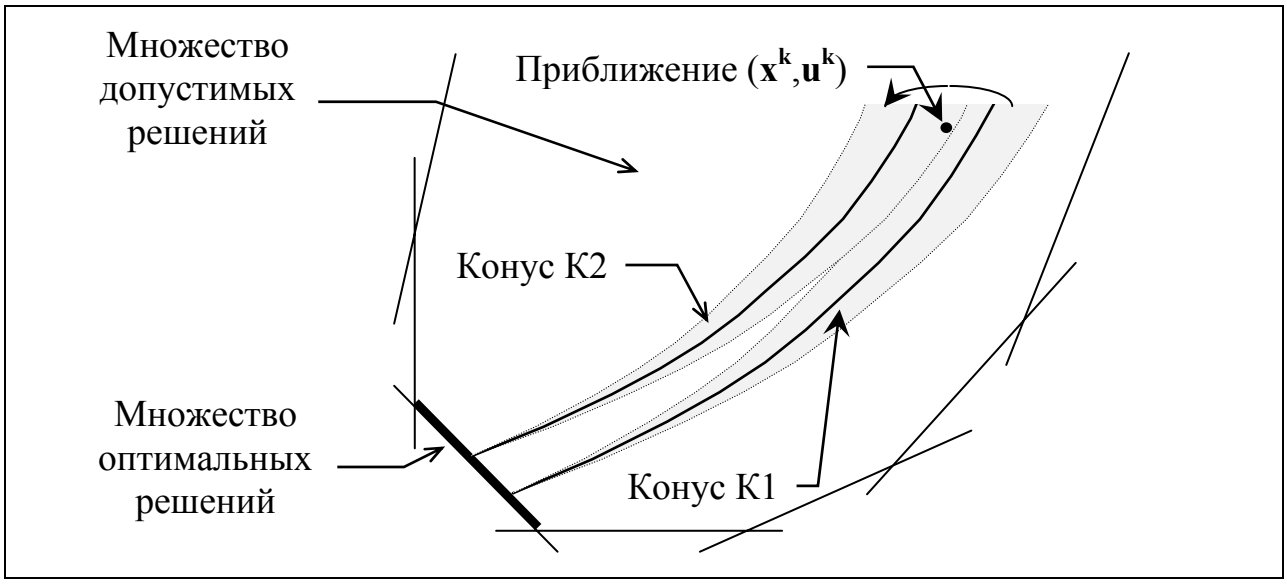


Рис.3. Процедура сокращения коэффициента скошенности

Наиболее простая процедура сокращения коэффициента скошенности состоит в уменьшении тех компонент инициирующего вектора, значения которых превышают $x_j^k g_j\left(\mathbf{u}^{\mathbf{k}}\right)\big/\mu^k$, и одновременном увеличении его минимальной компоненты на максимально возможную величину при условии неувеличения среднего значения и невыхода из конуса скошенного пути.

Можно проиллюстрировать эту процедуру следующей картинкой. На рис.4 горизонтальной чертой помечены значения $x_j^k g_j\left(\mathbf{u}^{\mathbf{k}}\right)\big/\mu^k$, белым кружком старые значения компонент вектора **t**, черным кружком – новые значения этих компонент.

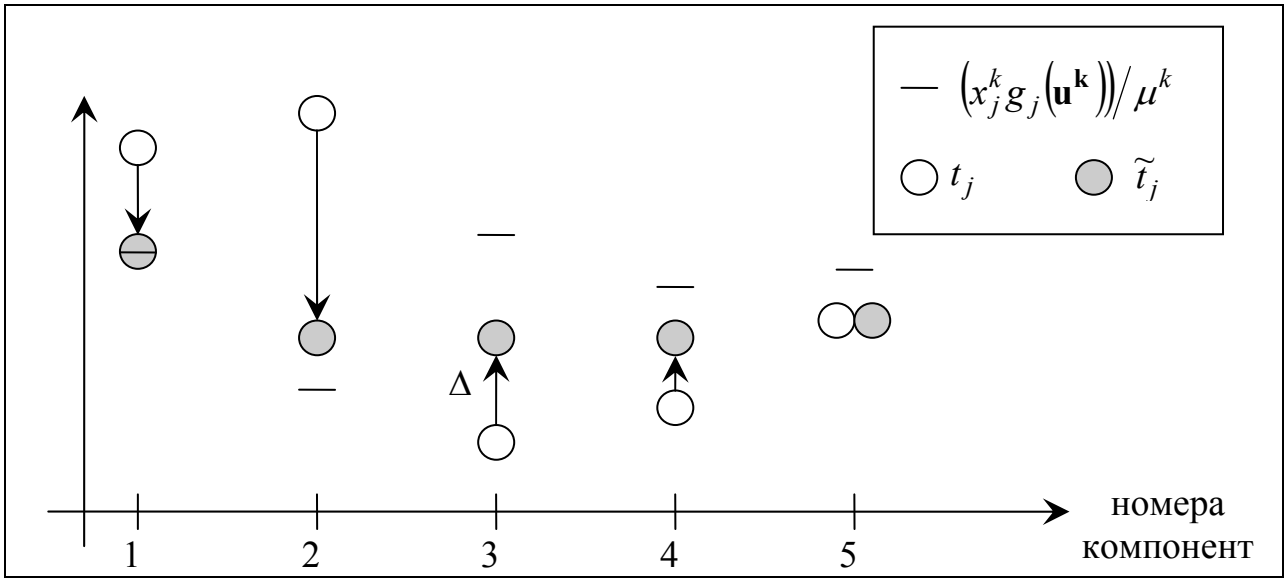


Рис.4. Геометрическая интерпретация
процедуры сокращения коэффициента скошенности

Формализуем процедуру. На каждой итерации после получения новых приближений $\mathbf{x^k}$ и $\mathbf{u^k}$, а также нового значения μ^k будем искать новый инициирующий вектор $\tilde{\mathbf{t}} = \tilde{\mathbf{t}}(\Delta)$ путем решения следующей задачи:

$$\Delta \to \max,$$

$$\sum_{j=1}^{n} \frac{1}{\mu^k \tilde{t}_j(\Delta)} \left(\mu^k \tilde{t}_j(\Delta) - x_j^k g_j\left(\mathbf{u^k}\right)\right)^2 \le \theta \mu^k \left(t_{\min} + \Delta\right),$$

$$\sum_{j=1}^{n} \tilde{t}_j(\Delta) \le \sum_{j=1}^{n} t_j(\Delta),$$

$$\tilde{t}_j(\Delta) = \max\left\{t_{\min} + \Delta,\ \min\left\{t_j, \left(x_j^k g_j\left(\mathbf{u^k}\right)\right) / \mu^k\right\}\right\}.$$

Комбинированные алгоритмы

Несмотря на имеющиеся различия (различные способы исключения ограничений-неравенств; наличие полиномиальных оценок максимального объема вычислений у алгоритмов центрального пути и отсутствие таковых у аффинно-масштабирующих алгоритмов; необходимость формирования особого стартового приближения для алгоритмов центрального пути и т.д.), нетрудно заметить, что формулы алгоритмов довольно близки.

Комбинированные алгоритмы [13] объединяют с помощью несложной вычислительной процедуры идеи, использующиеся обоими классами алгоритмов. Итеративный процесс комбинированного алгоритма начинается с произвольной стартовой точки $\mathbf{x^1} > \mathbf{0}$. На каждой итерации $k = 1,2,\ldots$ ищется вектор невязок балансовых ограничений-равенств

$$\mathbf{r^k} = \mathbf{b} - \mathbf{A}\mathbf{x^k}.$$

Итеративный переход осуществляется по правилу

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \lambda^k \Delta \mathbf{x^k},$$

где $\Delta \mathbf{x^k}$ – направление корректировки, а λ^k – шаг корректировки, задаваемый из соображений невыхода переменных прямой задачи за пределы внутренности допустимой области.

Для нахождения направления корректировки $\Delta \mathbf{x^k}$ решается следующая параметрическая задача минимизации:

$$\mathbf{c^T}\Delta \mathbf{x} - \mu \sum_{j=1}^{n} \frac{\Delta x_j}{x_j^k} + \frac{1}{2} \sum_{j=1}^{n} \frac{\Delta x_j^2}{\left(x_j^k\right)^2} \to \min_{\Delta \mathbf{x} \in \mathbf{R^n}}, \quad \mathbf{A}\Delta \mathbf{x} = \mathbf{r^k}. \tag{47}$$

Заметим, что если значение параметра μ равно нулю, то формулы задачи (47) полностью совпадают с формулами аффинно-масштабирующего алгоритма. При $\mu = 1$ задача (47) дает такое же направление, как и задача, решаемая в алгоритме центрального пути, за исключением того, что теперь не требуется иметь труднополучаемого стартового приближения.

Задача (47) решается методом множителей Лагранжа. Двойственные оценки $\mathbf{u}^{\mathbf{k+1}}$ ограничений-равенств $\mathbf{A}\Delta\mathbf{x} = \mathbf{r}^{\mathbf{k}}$ находятся по формуле

$$\mathbf{u}^{\mathbf{k+1}}(\mu) = \left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{A}^{\mathbf{T}}\right)^{-1}\left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{c} + \mathbf{r}^{\mathbf{k}} - \mu\mathbf{b}\right).$$

Направление корректировки вычисляется следующим образом:

$$\Delta\mathbf{x}^{\mathbf{k}}(\mu) = \mu\mathbf{x}^{\mathbf{k}} - \mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{g}\left(\mathbf{u}^{\mathbf{k+1}}(\mu)\right).$$

Легко обнаружить, что функции $\mathbf{u}^{\mathbf{k+1}}(\mu)$ и $\Delta\mathbf{x}^{\mathbf{k}}(\mu)$ являются линейными относительно μ. Таким образом, можно отыскать 2 вектора

$$\mathbf{u}^{\mathbf{k+1}}(0) = \left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{A}^{\mathbf{T}}\right)^{-1}\left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{c} + \mathbf{r}^{\mathbf{k}}\right)$$

и

$$\mathbf{u}^{\mathbf{k+1}}(1) = \left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{A}^{\mathbf{T}}\right)^{-1}\left(\mathbf{A}\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{c} + \mathbf{r}^{\mathbf{k}} - \mathbf{b}\right),$$

что принципиально не сложнее исходной задачи, поскольку здесь, по сути, решается одна и та же система линейных уравнений с 2 правыми частями, и искомая функция $\mathbf{u}^{\mathbf{k+1}}(\mu)$ будет иметь вид

$$\mathbf{u}^{\mathbf{k+1}}(\mu) = (1-\mu)\mathbf{u}^{\mathbf{k+1}}(0) + \mu\mathbf{u}^{\mathbf{k+1}}(1).$$

Аналогично, итоговое направление корректировки $\Delta\mathbf{x}^{\mathbf{k}}(\mu)$ будет линейной комбинацией

$$\Delta\mathbf{x}^{\mathbf{k}}(\mu) = (1-\mu)\Delta\mathbf{x}^{\mathbf{k}}(0) + \mu\Delta\mathbf{x}^{\mathbf{k}}(1)$$

аффинно-масштабирующего

$$\Delta\mathbf{x}^{\mathbf{k}}(0) = -\mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{g}\left(\mathbf{u}^{\mathbf{k+1}}(0)\right)$$

и центрирующего

$$\Delta\mathbf{x}^{\mathbf{k}}(1) = \mathbf{x}^{\mathbf{k}} - \mathbf{X}_{\mathbf{k}}^{\mathbf{2}}\mathbf{g}\left(\mathbf{u}^{\mathbf{k+1}}(1)\right)$$

направлений. При этом для каждого направления легко найти максимальный шаг, с которым можно производить корректировку, оставаясь внутри допустимой области по ограничениям-неравенствам:

$$\lambda^k(\mu) = \gamma \min_{j:\Delta x_j^k(\mu)<0}\left(-x_j^k \,/\, \Delta x_j^k(\mu)\right), \quad \gamma \in (0;1).$$

Таким образом, путем небольшого усложнения алгоритма мы получили целое множество направлений корректировки $\Delta\mathbf{X} = \left\{\Delta\mathbf{x}^{\mathbf{k}}(\mu):\ \mu \in [0;1]\right\}$. Из этого множества возможных направлений нужно выбрать наиболее предпочтительное в целях скорейшего решения задачи.

Работа алгоритма разделяется на 2 этапа. На первом – необходимо максимально быстро найти допустимую точку, удовлетворяющую равенствам $\mathbf{A}\mathbf{x} = \mathbf{b}$. Поскольку уменьшение невязки ограничений-равенств осуществляется в соответствии с формулой

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

то на первом этапе в качестве критерия можно использовать задачу

$$\lambda^k(\mu) \to \max_{\mu \in [0;1]}. \qquad (48)$$

На втором этапе необходимо максимально быстро сокращать невязку двойственности (разницу значений целевых функций прямой и двойственной задачи), равную

$$\mathbf{c^T x} - \mathbf{b^T u} = \sum_{j=1}^{n} x_j g_j(\mathbf{u}).$$

Таким образом, критерием второго этапа будет следующая задача:

$$\sum_{j=1}^{n} \left(x_j^k + \lambda^k(\mu)\Delta x_j^k(\mu)\right) g_j\left(\mathbf{u^{k+1}}(\mu)\right) \to \min_{\mu \in [0;1]}. \qquad (49)$$

И на первом, и на втором этапе необязательно решать задачи (48) и (49) точно. Достаточно вычислить критерии для нескольких точек на интервале $\mu \in [0;1]$, найти наилучшее значение μ^k, отыскать направление $\Delta\mathbf{x^k}\left(\mu^k\right)$ и шаг $\lambda^k\left(\mu^k\right)$ корректировки и осуществить итеративный переход

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \lambda^k\left(\mu^k\right)\Delta\mathbf{x^k}.$$

Нельзя гарантировать, что локальный выигрыш на одной итерации ведет и к ускорению всего вычислительного процесса относительно фиксированного случая $\mu = 0$, соответствующего формулам аффинно-масштабирующего алгоритма. Действительно, вычислительные процессы для разных значений μ приводят к различным траекториям. Однако экспериментальное исследование [13] показало, что на практических задачах действительно наблюдается ускорение.

4. Решение систем линейных неравенств

Метод Фурье-Черникова

При реализации многих прикладных задач возникает необходимость решения систем линейных неравенств относительно $\mathbf{x} \in \mathbf{R^n}$ вида

$$\mathbf{Ax} - \mathbf{b} \le \mathbf{0}, \qquad (50)$$

где $\mathbf{A} \in \mathbf{R^{m \times n}}$, $\mathbf{b} \in \mathbf{R^m}$.

Если неравенство имеет противоположный знак, достаточно умножить его на –1. Все равенства из системы (50) исключены с соответствующим сокращением числа переменных.

Аналогом метода Гаусса для систем линейных неравенств будет метод последовательного исключения переменных Фурье-Черникова [14]. Обозначим $\mathbf{a_j}$ – j-строку матрицы **A**. Введем функции

$$l_j(\mathbf{x}) = \mathbf{a_j x} - b_j, \ j = 1,...,m.$$

Таким образом, система (50) представима в следующем виде:

$$l_j(\mathbf{x}) \le 0, \ j = 1,...,m. \qquad (51)$$

Обозначим J_+, J_- и J_0 – множества номеров неравенств с положительным, отрицательным и нулевым коэффициентом при x_1:

$$J_+ = \{j : a_{j1} > 0\}, \quad J_- = \{j : a_{j1} < 0\}, \quad J_0 = \{j : a_{j1} = 0\}.$$

Исключим переменную x_1 из системы (51), разделив все неравенства из J_+ и J_- на $|a_{j1}|$. Получим:

$$\begin{cases} x_1 + \widetilde{l}_j(\widetilde{\mathbf{x}}) \le 0, & j \in J_+, \\ -x_1 + \widetilde{l}_i(\widetilde{\mathbf{x}}) \le 0, & i \in J_-, \\ \widetilde{l}_k(\widetilde{\mathbf{x}}) \le 0, & k \in J_+, \end{cases}$$

$$\widetilde{l}_j(\widetilde{\mathbf{x}}) = \frac{1}{a_{j1}} l_j(\mathbf{x}) - x_1, \ j \in J_+, \ \widetilde{l}_i(\widetilde{\mathbf{x}}) = \frac{1}{|a_{i1}|} l_i(\mathbf{x}) + x_1, \ i \in J_-, \ \widetilde{l}_k(\widetilde{\mathbf{x}}) = l_k(\mathbf{x}), \ k \in J_0.$$

Здесь $\widetilde{\mathbf{x}}^{\mathbf{T}} = (x_2, x_3, ..., x_n)$.

Отсюда,

$$\widetilde{l}_i(\widetilde{\mathbf{x}}) \le x_1 \le -\widetilde{l}_j(\widetilde{\mathbf{x}}), \ i \in J_-, \ j \in J_+, \ \widetilde{l}_k(\widetilde{\mathbf{x}}) \le 0, \ k \in J_0$$

или, после исключения переменной x_1,

$$\widetilde{l}_i(\widetilde{\mathbf{x}}) + \widetilde{l}_j(\widetilde{\mathbf{x}}), \ i \in J_-, \ j \in J_+, \ \widetilde{l}_k(\widetilde{\mathbf{x}}) \le 0, \ k \in J_0.$$

Аналогичным способом последовательно исключаем переменные x_2, $x_3, \ldots, x_{n-1}$. В итоге останется система с единственной переменной x_n. Находим любое ее допустимое решение, подставляем его в систему с x_{n-1}, и т.д.

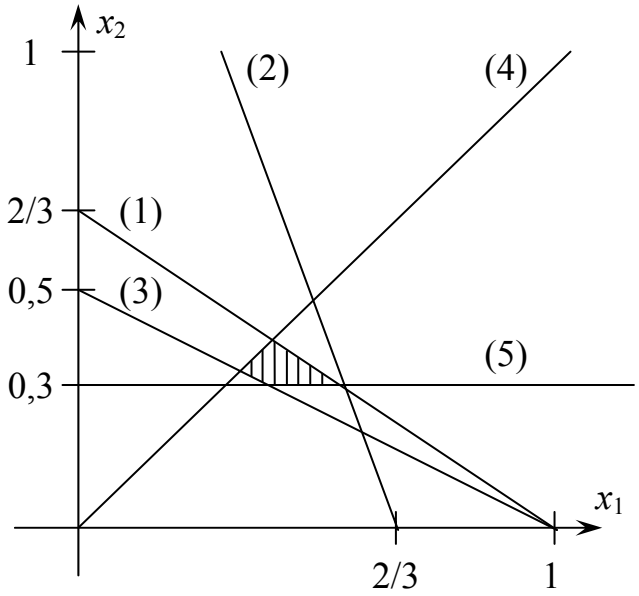
Рассмотрим численный пример:

Пример 6.

$$\begin{cases} 2x_1 + 3x_2 - 2 \le 0, & (1) \\ 3x_1 + x_2 - 2 \le 0, & (2) \\ -x_1 - 2x_2 + 1 \le 0, & (3) \\ -x_1 + x_2 \le 0, & (4) \\ -x_2 + 0{,}3 \le 0. & (5) \end{cases}$$

Графически допустимая область изображена на рисунке справа. Решим задачу методом Фурье-Черникова:

$$\begin{cases} \left.\begin{array}{l} x_1 + 1{,}5x_2 - 1 \le 0 \\ x_1 + 1/3\, x_2 - 2/3 \le 0 \end{array}\right\} J_+ \\ \left.\begin{array}{l} -x_1 - 2x_2 + 1 \le 0 \\ -x_1 + x_2 \le 0 \end{array}\right\} J_- \\ \left.\begin{array}{l} -x_2 + 0{,}3 \le 0 \end{array}\right\} J_0 \end{cases}$$



$$\begin{cases} x_1 \le -1{,}5x_2 + 1 \\ x_1 \le -1/3x_2 + 2/3 \\ x_1 \ge -2x_2 + 1 \\ x_1 \ge x_2 \\ -x_2 + 0{,}3 \le 0 \end{cases} \Rightarrow \begin{cases} -2x_2 + 1 + 1{,}5x_2 - 1 \le 0 \\ -2x_2 + 1 + 1/3x_2 - 2/3 \le 0 \\ x_2 + 1{,}5x_2 - 1 \le 0 \\ x_2 + 1/3x_2 - 2/3 \le 0 \\ -x_2 + 0{,}3 \le 0 \end{cases} \Rightarrow \begin{cases} -0{,}5x_2 \le 0 \\ -5/3x_2 + 1/3 \le 0 \\ 2{,}5x_2 - 1 \le 0 \\ 4/3x_2 - 2/3 \le 0 \\ -x_2 + 0{,}3 \le 0 \end{cases} \Rightarrow \begin{cases} x_2 \ge 0 \\ x_2 \ge 0{,}2 \\ x_2 \le 0{,}4 \\ x_2 \le 0{,}5 \\ x_2 \ge 0{,}3 \end{cases}$$

Отсюда получаем, что $x_2 \in [0{,}3;\ 0{,}4]$. Выберем произвольное значение, например, $x_2 = 0{,}35$ и вернемся к системе с переменной x_1:

$$\begin{cases} x_1 \le -1{,}5 * 0{,}35 + 1 \\ x_1 \le -1/3 * 0{,}35 + 2/3 \\ x_1 \ge -2 * 0{,}35 + 1 \\ x_1 \ge 0{,}35 \end{cases} \Rightarrow \begin{cases} x_1 \le 0{,}475 \\ x_1 \le 0{,}55 \\ x_1 \ge 0{,}3 \\ x_1 \ge 0{,}35 \end{cases} \Rightarrow x_1 \in [0{,}35;\ 0{,}475]$$

Например, решением примера 6 будет точка $x_1 = 0{,}4$; $x_2 = 0{,}35$.

Главный недостаток рассмотренного метода – очень быстрый рост объема вычислений, необходимых для получения решения, с ростом числа переменных и неравенств в решаемой системе. Без использования сверток и алгоритмов, исключающих неравенства-следствия, даже задача с несколькими десятками переменных и таким же числом неравенств не подлежит решению с использованием современной техники. В то же время существует метод, позволяющий эффективно решать системы линейных неравенств.

Метод сведения к задаче линейного программирования с одной дополнительной переменной

Пусть необходимо найти решение следующей системы неравенств:

$$\mathbf{Ax} = \mathbf{b},\ \mathbf{x} \ge \mathbf{0} \qquad (52)$$

Систему (50) можно привести к данному виду введением дополнительных переменных с поправкой на неотрицательность только части переменных. Задача (52) сходна с задачей (1). Отличие ее состоит в отсутствии целевой функции. Итеративный процесс начинается с произвольного вектора $\mathbf{x^1} > \mathbf{0}$. Вычисляется вектор невязок ограничений-равенств

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax^1}$$

и решается задача минимизации дополнительной переменной α:

$$\alpha \to \min,$$

$$\mathbf{Ax} + \alpha\,\mathbf{r} = \mathbf{b},\quad \mathbf{x} \ge \mathbf{0},\quad \alpha \ge 0.$$

Пара $(\mathbf{x^1};\ \alpha^1 = 1)$ является допустимым стартовым приближением. Если оптимальное решение данной задачи $\overline{\alpha} = 1$, то несовместна система $\mathbf{Ax} = \mathbf{b}$. Данный факт выявляется методом внутренних точек на первой итерации. Если оптимальное решение $\overline{\alpha} \in (0;\ 1)$, несовместной является система (52). Метод внутренних точек идентифицирует этот случай за конечное число итера-

ций. И, наконец, если $\overline{\alpha}=0$, то вектор $\overline{\mathbf{x}}$ – относительно внутренняя точка множества решений системы (52).

В простейшем варианте алгоритма, как и при решении задач (1), (2), итеративный переход осуществляется по правилу

$$\mathbf{x^{k+1}}=\mathbf{x^{k}}+\lambda^{k}\Delta\mathbf{x^{k}}.$$

Направление корректировки $\Delta\mathbf{x^{k}}$ вычисляется по формуле

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

где

$$\mathbf{u^{k}}=\alpha^{k}\left(\mathbf{AD_{k}A^{T}}\right)^{-1}\mathbf{r},\ \ \mathbf{g}\left(\mathbf{u^{k}}\right)=-\mathbf{A^{T}u^{k}}.$$

Шаг корректировки по-прежнему обеспечивает нахождение в положительном ортанте и ограничен сверху единицей:

$$\lambda^{k}=\min\left\{1;\ \gamma\min_{j:\Delta x_j^k<0}\left(-x_j^k/\Delta x_j^k\right)\right\},\ \ \gamma\in(0;2/3).$$

При этом минимизируемая переменная уменьшается в пропорции

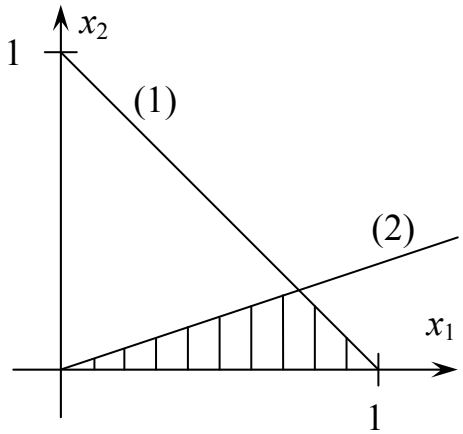
$$\alpha^{k+1}=\left(1-\lambda^{k}\right)\alpha^{k}.$$

Отметим, что формулы здесь близки к формулам аффинно-масштабирующего алгоритма при недопустимом стартовом приближении. Отличие заключается в отсутствии целевой функции $\mathbf{c^{T}x}$.

Рассмотрим численный пример:

Пример 7.

$$\begin{cases}x_1+x_2\le 1, & (1)\\ x_1-3x_2\ge 0, & (2)\\ x_1\ge 0,\quad x_2\ge 0.\end{cases}$$



Данную систему, изображенную на рисунке справа, сведем к виду (52) введением двух дополнительных переменных:

$$\begin{cases}x_1+x_2+x_3=1,\\ x_1-3x_2-x_4\ge 0,\\ x_1\ge 0,\quad x_2\ge 0,\quad x_3\ge 0,\quad x_4\ge 0.\end{cases}$$

Выполним итерацию метода внутренних точек:

$$\left(\mathbf{x^{1}}\right)^{\mathbf{T}}=(1\ 1\ 1\ 1),\ \ \alpha^{1}=1,\ \ \mathbf{r}=\begin{pmatrix}1-1-1-1=-2\\ 0-1+3+1=3\end{pmatrix},\ \ \left(\mathbf{d^{1}}\right)^{\mathbf{T}}=(1\ 1\ 1\ 1),$$

$$\begin{pmatrix}1 & 1 & 1 & 0\\ 1 & -3 & 0 & -1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -3\\ 1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix}=\begin{pmatrix}-2\\ 3\end{pmatrix},$$

$$\begin{pmatrix} 3 & -2 \\ -2 & 11 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}=\begin{pmatrix} -2 \\ 3 \end{pmatrix},\ u_2=1{,}5u_1+1,\ -2u_1+16{,}5u_1+11=3,\ 14{,}5u_1=-8,$$

$$\mathbf{u^1}=\begin{pmatrix} -16/29 \\ 5/29 \end{pmatrix},\quad \mathbf{g(u^1)}=\begin{pmatrix} 11/29 \\ 31/29 \\ 16/29 \\ 5/29 \end{pmatrix},\quad \Delta\mathbf{x^1}=\begin{pmatrix} -11/29 \\ -31/29 \\ -16/29 \\ -5/29 \end{pmatrix}.$$

Шаг корректировки

$$\lambda^1=\min\left\{1;\ \frac{2}{3}*\frac{29}{11};\ \frac{2}{3}*\frac{29}{31};\ \frac{2}{3}*\frac{29}{16};\ \frac{2}{3}*\frac{29}{5};\right\}=\frac{58}{93}<1,$$

поэтому допустимое решение еще не найдено. Перейдем ко второй итерации:

$$\mathbf{x^2}=\begin{pmatrix} 1-58/93*11/29 \\ 1-58/93*31/29 \\ 1-58/93*16/29 \\ 1-58/93*5/29 \end{pmatrix}=\begin{pmatrix} 71/93 \\ 1/3 \\ 61/93 \\ 83/93 \end{pmatrix},\ \alpha^2=1-\lambda^1=\frac{35}{93}.$$

На второй итерации допустимое решение системы будет получено.

Случай интервальных ограничений на переменные. Быстрая идентификация несовместности системы

Более важные изменения алгоритма, существенно увеличивающие его эффективность, произойдут в случае интервальных ограничений на переменные. Пусть требуется найти решение системы

$$\mathbf{Ax}=\mathbf{b},\quad \underline{\mathbf{x}}\le\mathbf{x}\le\overline{\mathbf{x}}, \tag{53}$$

где $\underline{\mathbf{x}}<\overline{\mathbf{x}}$, то есть $\underline{x}_j<\overline{x}_j$ для всех $j=1,\ldots,n$.

Итеративный процесс начинается с произвольного вектора $\underline{\mathbf{x}}<\mathbf{x^1}<\overline{\mathbf{x}}$. При отсутствии дополнительной априорной информации в качестве стартовой точки часто берется середина интервала

$$\mathbf{x^1}=(\underline{\mathbf{x}}+\overline{\mathbf{x}})/2.$$

Первым отличием будет способ вычисления весовых коэффициентов. В частности, можно использовать квадрат расстояния до ближайшей границы:

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

И главной особенностью будет возможность быстрой идентификации несовместности системы неравенств (53) на основе варианта теоремы Фаркаша об альтернативных неравенствах. Система (53) несовместна в том и только в том случае, если для некоторого вектора $\mathbf{u}\in\mathbf{R^m}$ функция

$$\psi(\mathbf{u})=\mathbf{b^T u}-\overline{\mathbf{x}}^{\mathbf{T}}(\mathbf{A^T u})_+-\underline{\mathbf{x}}^{\mathbf{T}}(\mathbf{A^T u})_- \tag{54}$$

принимает положительное значение. Здесь $(\cdot)_+$ и $(\cdot)_-$ – положительная и отрицательная срезки компонент векторов, то есть для любого $\mathbf{x}\in\mathbf{R^n}$

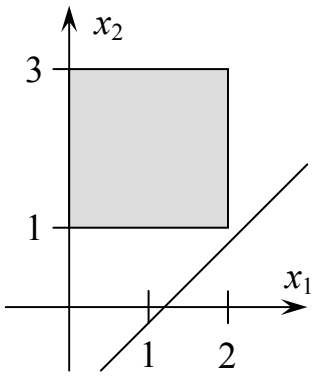
$$(x_+)_j=\max\{0;x_j\},\ (x_-)_j=\min\{0;x_j\},\ j=1,\ldots,n.$$

В качестве вектора **u** можно использовать вычисляемый на каждой итерации вектор двойственных переменных $\mathbf{u^k}$. Известно, что если система (53) несовместна, то через конечное число итераций будет выполнено неравенство $\psi(\mathbf{u^k}) > 0$. При этом, как показали эксперименты [12], выявление случая несовместности с помощью критерия (54) происходит часто на первой же итерации метода внутренних точек.

Приведем пример:

Пример 8.

$$\begin{cases} x_1 - x_2 = 1{,}1, \\ x_1 \in [0;\ 2],\ \ x_2 \in [1;\ 3]. \end{cases}$$



На рисунке справа видим, что допустимых решений данная система не имеет. Выполним итерацию метода внутренних точек:

$$\left(\mathbf{x^1}\right)^{\mathbf{T}} = (1\ \ 2),\ \ \alpha^1 = 1,\ \ \mathbf{r} = 1{,}1 - 1 - (-2) = 2{,}1,$$

$$\left(\mathbf{d^1}\right)^{\mathbf{T}} = (1\ \ 1),$$

$$(1\ \ -1)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} u = 2{,}1\ ,\ \ 2u = 2{,}1,\ \ u^1 = 1{,}05,$$

$$\psi(u^1) = 1{,}1 * 1{,}05 - (2\ \ 3)\begin{pmatrix} 1{,}05 \\ 0 \end{pmatrix} - (0\ \ 1)\begin{pmatrix} 0 \\ -1{,}05 \end{pmatrix} = 1{,}1 * 1{,}05 - 1{,}05 = 0{,}105 > 0.$$

Видим, что идентификация несовместности системы произошла на первой же итерации, при том, что небольшой корректировкой условий задачи (например, уменьшением правой части ограничения-равенства на 0,1) можно добиться того, что множество допустимых решений становится непустым.

5. Обращение симметричной положительно определенной матрицы

Наиболее сложной в вычислительном отношении задачей на каждой итерации алгоритма внутренних точек является обращение симметричной положительно определенной матрицы $\mathbf{AD_k A^T}$, где $\mathbf{D_k}$ – меняющаяся по итерациям диагональная матрица весовых коэффициентов. Для решения этой задачи требуется порядка n^3 вычислительных операций (метод Крамера, в котором требуется вы вычисление определителей, неприменим для решения практических задач).

Метод Гаусса

Наиболее распространенным на практике методом решения систем уравнений вида

$$\widetilde{\mathbf{A}}\,\mathbf{u} = \widetilde{\mathbf{b}} \qquad (55)$$

является метод последовательного исключения переменных – метод Гаусса. Метод Гаусса основывается на следующих утверждениях:

1. Системы уравнений, полученные в результате перестановки уравнений; умножения уравнения на любое число, отличное от нуля; прибавления одного уравнения, умноженного на любое число, к другому уравнению, являются эквивалентными, то есть имеют одни и те же решения.

2. Преобразованиям уравнений системы, очевидно, соответствуют элементарные преобразования строк расширенной матрицы системы $\left(\widetilde{\mathbf{A}} \mid \widetilde{\mathbf{b}}\right)$: перестановка строк; умножение всех элементов строки на любое число, отличное от нуля; прибавление элементов одной строки, умноженных на одно и то же число, к соответствующим элементам другой строки.

Чтобы решить систему m уравнений с m неизвестными, нужно расширенную матрицу системы с помощью элементарных преобразований строк привести к такому виду, чтобы основная матрица системы (стоящая слева от черты) была приведена к единичной, тогда на месте столбца свободных членов (справа от черты) окажется решение системы.

Рассмотрим следующий пример:

Пример 9.

$$\begin{pmatrix} 4 & 2 \\ 2 & 10 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 14 \\ 16 \end{pmatrix}.$$

Выполним последовательные преобразования расширенной матрицы:

$$\left(\begin{array}{cc|c} 4 & 2 & 14 \\ 2 & 10 & 16 \end{array}\right) \sim \left(\begin{array}{cc|c} 1 & 0{,}5 & 3{,}5 \\ 1 & 5 & 8 \end{array}\right) \sim \left(\begin{array}{cc|c} 1 & 0{,}5 & 3{,}5 \\ 0 & 4{,}5 & 4{,}5 \end{array}\right) \sim \left(\begin{array}{cc|c} 1 & 0{,}5 & 3{,}5 \\ 0 & 1 & 1 \end{array}\right) \sim \left(\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 1 \end{array}\right).$$

Получили решение системы $u_1 = 3$, $u_2 = 1$.

Метод Халецкого

Для получения решения методом Гаусса требуется порядка m^3 арифметических операций. Метод Халецкого (метод квадратного корня), использующий дополнительную информацию о том, что решается система с симметричной положительно определенной матрицей, работает примерно вдвое быстрее.

Суть метода состоит в представлении матрицы $\widetilde{\mathbf{A}}$ в виде произведения нижней треугольной и верхней треугольной матриц $\widetilde{\mathbf{A}} = \mathbf{R}\mathbf{R}^{\mathbf{T}}$:

$$\widetilde{\mathbf{A}} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{12} & a_{22} & a_{23} & \dots & a_{2m} \\ a_{13} & a_{23} & a_{33} & \dots & a_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{1m} & a_{2m} & a_{3m} & \dots & a_{mm} \end{pmatrix} = \begin{pmatrix} r_{11} & & & & \\ r_{12} & r_{22} & & & \\ r_{13} & r_{23} & r_{33} & & \\ \dots & \dots & \dots & \dots & \\ r_{1m} & r_{2m} & r_{3m} & \dots & r_{mm} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1m} \\ & r_{22} & r_{23} & \dots & r_{2m} \\ & & r_{33} & \dots & r_{3m} \\ & & & \dots & \dots \\ & & & & a_{mm} \end{pmatrix}.$$

Отсюда последовательно найдем

$$a_{11} = r_{11}^2 \qquad \Rightarrow r_{11} = \sqrt{a_{11}},$$
$$a_{12} = r_{11}r_{12}, ..., a_{1m} = r_{11}r_{1m} \qquad \Rightarrow r_{12} = a_{12}/r_{11}, ..., r_{1m} = a_{1m}/r_{11},$$
$$a_{22} = r_{12}^2 + r_{22}^2 \qquad \Rightarrow r_{22} = \sqrt{a_{22} - r_{12}^2},$$
$$a_{23} = r_{12}r_{13} + r_{22}r_{23}, ..., a_{2m} = r_{12}r_{1m} + r_{22}r_{2m} \Rightarrow r_{23} = \frac{a_{23} - r_{12}r_{13}}{r_{22}}, ..., r_{2m} = \frac{a_{2m} - r_{12}r_{1m}}{r_{22}}.$$

Аналогично находятся все остальные элементы матрицы **R**. Поскольку

$$r_{1k}^2 + r_{2k}^2 + ... + r_{kk}^2 = a_{kk},\ k = 1,...,m,$$

можно утверждать, что метод обладает высокой вычислительной устойчивостью, что является еще одним плюсом метода Халецкого.

Далее для решения системы (55), что эквивалентно $\mathbf{R}\mathbf{R}^{\mathbf{T}}\mathbf{u} = \widetilde{\mathbf{b}}$, потребуется последовательно решить 2 системы с треугольными матрицами:

$$\mathbf{R}\mathbf{y} = \widetilde{\mathbf{b}},\ \ \mathbf{R}^{\mathbf{T}}\mathbf{u} = \mathbf{y}.$$

Первая из них (используется прямой ход) имеет решение

$$y_1 = b_1/r_{11},\ \ y_i = \left(b_i - \sum_{k=1}^{i-1} r_{ki}y_k\right)\Big/ r_{ii},\ \ i = 2,...,m,$$

а вторая (используется обратный ход) –

$$u_m = y_m/r_{mm},\ \ u_i = \left(y_i - \sum_{k=i+1}^{m} r_{ik}u_k\right)\Big/ r_{ii},\ \ i = m-1,...,1.$$

Отыщем решение системы из примера 9. Сначала нужно найти разложение Халецкого:

$$\begin{pmatrix} 4 & 2 \\ 2 & 10 \end{pmatrix} = \begin{pmatrix} r_{11} & 0 \\ r_{12} & r_{22} \end{pmatrix}\begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}.$$

$$r_{11}^2 = 4,\ r_{11} = 2, \quad r_{11}r_{12} = 2,\ r_{12} = 1, \quad r_{12}^2 + r_{22}^2 = 10,\ 1 + r_{22}^2 = 10,\ r_{22} = 3.$$

Таким образом,

$$\begin{pmatrix} 4 & 2 \\ 2 & 10 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}.$$

Далее,

$$\begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 14 \\ 16 \end{pmatrix},\ \ 2y_1 = 14,\ \ y_1 = 7,\ \ 7 + 3y_2 = 16,\ \ y_2 = 3.$$

$$\begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 3 \end{pmatrix},\ \ 3u_2 = 3,\ \ u_2 = 1,\ \ 2u_1 + u_2 = 7,\ \ u_1 = 3.$$

Метод сопряженных направлений

Альтернативой точным методам, типичными представителями которых является метод Гаусса и метод Халецкого, является использование итеративных методов, одним из которых является метод сопряженных направлений.

Задача (55) эквивалентна задаче безусловной минимизации квадратичной функции

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathbf{T}}\widetilde{\mathbf{A}}\mathbf{u} - \widetilde{\mathbf{b}}^{\mathbf{T}}\mathbf{u}. \qquad (56)$$

Начиная с некоторого произвольно выбранного приближения $\mathbf{u}^{\mathbf{0}}$, находим на каждой итерации $t = 1,2,\ldots$ метода сопряженных направлений вектор

$$\Delta\mathbf{u}^{\mathbf{t}} = \nabla f\left(\mathbf{u}^{\mathbf{t-1}}\right) - \beta_t \Delta\mathbf{u}^{\mathbf{t-1}} = \widetilde{\mathbf{A}}\mathbf{u}^{\mathbf{t-1}} - \widetilde{\mathbf{b}} - \beta_t \Delta\mathbf{u}^{\mathbf{t-1}}, \qquad (57)$$

где способ определения шага β_t задает тот или иной вариант метода. Например, на первой итерации $\beta_1 = 0$, далее

$$\beta_t = \frac{\left(\widetilde{\mathbf{A}}\mathbf{u}^{\mathbf{t-1}} - \widetilde{\mathbf{b}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t-1}}}{\left(\Delta\mathbf{u}^{\mathbf{t-1}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t-1}}}. \qquad (58)$$

Итеративный переход осуществляется по формуле

$$\mathbf{u}^{\mathbf{t}} = \mathbf{u}^{\mathbf{t-1}} + \alpha_t \Delta\mathbf{u}^{\mathbf{t}}, \qquad (59)$$

где шаг α_t находим как решение задачи одномерной минимизации функции $f\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\,\Delta\mathbf{u}^{\mathbf{t}}\right)$:

$$f\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\Delta\mathbf{u}^{\mathbf{t}}\right) = \frac{1}{2}\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\Delta\mathbf{u}^{\mathbf{t}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\Delta\mathbf{u}^{\mathbf{t}}\right) - \widetilde{\mathbf{b}}^{\mathbf{T}}\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\Delta\mathbf{u}^{\mathbf{t}}\right) =$$

$$= \frac{1}{2}\left(\mathbf{u}^{\mathbf{t-1}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\mathbf{u}^{\mathbf{t-1}} + \alpha\left(\mathbf{u}^{\mathbf{t-1}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t}} + \frac{1}{2}\alpha^2\left(\Delta\mathbf{u}^{\mathbf{t}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t}} - \widetilde{\mathbf{b}}^{\mathbf{T}}\mathbf{u}^{\mathbf{t-1}} - \alpha\mathbf{r}^{\mathbf{T}}\Delta\mathbf{u}^{\mathbf{t}} \to \min_{\alpha}.$$

Найдем конкретное выражение для α_t, приравняв производную к нулю:

$$\frac{\partial f\left(\mathbf{u}^{\mathbf{t-1}} + \alpha\Delta\mathbf{u}^{\mathbf{t}}\right)}{\partial\alpha} = \left(\mathbf{u}^{\mathbf{t-1}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t}} + \alpha\left(\Delta\mathbf{u}^{\mathbf{t}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t}} - \widetilde{\mathbf{b}}^{\mathbf{T}}\Delta\mathbf{u}^{\mathbf{t}} = 0,$$

$$\alpha_t = -\frac{\left(\widetilde{\mathbf{A}}\mathbf{u}^{\mathbf{t-1}} - \widetilde{\mathbf{b}}\right)^{\mathbf{T}}\Delta\mathbf{u}^{\mathbf{t}}}{\left(\Delta\mathbf{u}^{\mathbf{t}}\right)^{\mathbf{T}}\widetilde{\mathbf{A}}\Delta\mathbf{u}^{\mathbf{t}}}. \qquad (60)$$

Известно, что метод сопряженных направлений (57)–(60), начиная с произвольно выбранного $\mathbf{u}^{\mathbf{0}}$, получает точку безусловного минимума $\mathbf{u}^{*}$ квадратичной функции (56) не более, чем за m итераций. Вектор $\mathbf{u}^{*}$ удовлетворяет условию (55). В то же время специфика алгоритмов внутренних точек позволяет решать задачу (55) приближенно. Действительно, на первых итерациях достаточно искать направление корректировки $\Delta\mathbf{x}^{\mathbf{k}}$, используя вектор $\mathbf{u}^{\mathbf{k}}$, для которого $\mathbf{A}\mathbf{D}_{\mathbf{k}}\mathbf{A}^{\mathbf{T}}\mathbf{u} = \widetilde{\mathbf{r}}^{\mathbf{k}} \neq \mathbf{r}^{\mathbf{k}}$, а на финальной стадии вычислительного процесса, где важна высокая точность, диагональная матрица весовых коэффициентов $\mathbf{D}_{\mathbf{k}}$ изменяется по итерациям крайне незначительно. Таким образом, в методе сопряженных направлений имеется хорошее стартовое приближение $\mathbf{u}^{\mathbf{0}} = \mathbf{u}^{\mathbf{k-1}}$, полученное как решение задачи (55) на предыдущей итерации алгоритма внутренних точек.

Дополнительным преимуществом метода сопряженных направлений относительно метода Халецкого является то, что формулы (57)–(60) легко адаптируются к работе с разреженными матрицами, что позволяет существенно повысить максимальную размерность решаемых задач.

Техника «частичного обновления»

Большинство приемов, сокращающих вычислительную сложность алгоритмов внутренних точек, базируется на том, что при решении вспомогательной задачи

$$\mathbf{A}\mathbf{D}_\mathbf{k}\mathbf{A}^\mathbf{T} = \tilde{\mathbf{b}}$$

по итерациям меняется только диагональная матрица весовых коэффициентов $\mathbf{D}_\mathbf{k} = diag\left\{d_j^k\right\}$, и эти изменения не столь велики. В частности, Н. Кармаркаром была предложена техника «частичного обновления», позволяющая уменьшить теоретическую сложность алгоритмов внутренних точек в $\sqrt{n}$ раз.

Для упрощения предположим, что на некоторой итерации матрица $\mathbf{D}_\mathbf{k}$ была единичной, а на следующей – стала равной $\mathbf{D}$. Обозначим *j*-столбец матрицы **A** за $\mathbf{a}_\mathbf{j}$. Тогда

$$\mathbf{A}\mathbf{D}\mathbf{A}^\mathbf{T} = \sum_{j=1}^{n} d_j \mathbf{a}_\mathbf{j}\mathbf{a}_\mathbf{j}^\mathbf{T} = \sum_{j=1}^{n} \mathbf{a}_\mathbf{j}\mathbf{a}_\mathbf{j}^\mathbf{T} + \sum_{j=1}^{n}\left(d_j - 1\right)\mathbf{a}_\mathbf{j}\mathbf{a}_\mathbf{j}^\mathbf{T} = \mathbf{A}\mathbf{A}^\mathbf{T} + \sum_{j=1}^{n}\left(d_j - 1\right)\mathbf{a}_\mathbf{j}\mathbf{a}_\mathbf{j}^\mathbf{T}.$$

Рассмотрим гипотетическую ситуацию, когда $d_j = 1$ для всех $j \neq 1$. В этом случае $\mathbf{A}\mathbf{D}\mathbf{A}^\mathbf{T}$ представляет собой одноранговую модификацию $\mathbf{A}\mathbf{A}^\mathbf{T}$:

$$\mathbf{A}\mathbf{D}\mathbf{A}^\mathbf{T} = \mathbf{A}\mathbf{A}^\mathbf{T} + \left(d_1 - 1\right)\mathbf{a}_\mathbf{1}\mathbf{a}_\mathbf{1}^\mathbf{T}.$$

По формуле Шермана-Моррисона получаем

$$\left(\mathbf{A}\mathbf{D}\mathbf{A}^\mathbf{T}\right)^{-\mathbf{1}} = \left(\mathbf{A}\mathbf{A}^\mathbf{T}\right)^{-\mathbf{1}} - \left(d_1 - 1\right)\frac{\left(\mathbf{A}\mathbf{A}^\mathbf{T}\right)^{-\mathbf{1}}\mathbf{a}_\mathbf{1}\mathbf{a}_\mathbf{1}^\mathbf{T}\left(\mathbf{A}\mathbf{A}^\mathbf{T}\right)^{-\mathbf{1}}}{1 + \left(d_1 - 1\right)\mathbf{a}_\mathbf{1}^\mathbf{T}\left(\mathbf{A}\mathbf{A}^\mathbf{T}\right)^{-\mathbf{1}}\mathbf{a}_\mathbf{1}}.$$

Данное преобразование требует порядка $O\left(n^2\right)$ арифметических операций.

В общем случае, когда все компоненты d_j отличны от единицы, обращение матрицы $\mathbf{A}\mathbf{D}\mathbf{A}^\mathbf{T}$ потребует *n* подобных одноранговых преобразований, то есть сложность одной итерации по-прежнему составляет $O\left(n^3\right)$.

Идея техники частичного обновления состоит в том, что производятся только одноранговые преобразования, соответствующие компонентам *j*, для которых величина $\left|d_j - 1\right|$ превосходит некоторое пороговое значение. Фактически на каждой итерации матрица весовых коэффициентов $\mathbf{D}_\mathbf{k}$ заменяется на ее аппроксимацию $\tilde{\mathbf{D}}_\mathbf{k}$.

Анализ алгоритмов внутренних точек, использующих технику частичного обновления должен состоять из двух дополнительных процедур. Во-

первых, необходимо показать, что при использовании модифицированной матрицы весовых коэффициентов алгоритм сохраняет свойства исходного алгоритма, в частности, полиномиальность. Второй желательной процедурой является подсчет максимального количества необходимых одноранговых преобразований.

Подробное исследование вычислительных аспектов техники частичного обновления провел Д. Шанно в работе [15]. К сожалению, несмотря на серьезное сокращение оценки максимального объема вычислений, применение частичного обновления на практике, как показали экспериментальные исследования, не дает пока существенных положительных результатов.

6. Примеры практического приложения алгоритмов внутренних точек

Методы внутренних точек к настоящему времени превратились из красивой теоретической конструкции в полноценный инструмент, позволяющий решать практические задачи большой размерности. Большинство распространенных программных продуктов, среди которых можно выделить CPlex (http://maximal-usa.com), BPMPD (http://sztaki.hu), MOSEK (http://mosek.com), и HOPDM (http://www.maths.ed.ac.uk/~gondzio/software/hopdm.html), позволяют решать задачи линейного программирования алгоритмами внутренних точек. При этом последние демонстрируют свою высокую эффективность как на задачах с заполненными матрицами, так и на задачах с блочной структурой и высокой степенью разреженности, а также на плохо обусловленных задачах.

Необходимость решения задач линейного программирования и систем линейных уравнений и неравенств часто возникает в связи с реализацией нелинейных моделей из различных прикладных областей на основе методов последовательной линеаризации. Многие такие модели сводятся к решению относительно вектора $\mathbf{z}$ системы вида

$$\mathbf{W}(\mathbf{z}) = \mathbf{0}, \quad \overline{\mathbf{z}} \geq \mathbf{z} \geq \underline{\mathbf{z}}. \tag{61}$$

Здесь $\overline{\mathbf{z}}$ и $\underline{\mathbf{z}}$ – заданные векторы ограничений, $\mathbf{W}(\mathbf{z})$ – нелинейная дифференцируемая вектор-функция.

Задача поиска допустимых режимов электроэнергетических систем

Одной из задач, представимых в виде (61), является задача [16] поиска допустимых режимов электроэнергетических систем (ЭЭС). Приведем соотношения, раскрывающие содержание ее условий.

Ограничения-равенства:

В соответствии с законами Кирхгофа для каждого i-го узла сети выполняются следующие соотношения

$$\sum_{j \neq i} P_{ij} + P_{Hi} - P_{Gi} = 0, \quad \sum_{j \neq i} Q_{ij} + Q_{Hi} - Q_{Gi} = 0, \quad i = 1, \ldots, n,$$

где P_{ij}, Q_{ij} – перетоки активной и реактивной мощности в ветви, соединяющей узлы i и j; P_{Hi}, Q_{Hi} – мощности активной и реактивной нагрузки в i-ом узле, значения которых принимаются постоянными; P_{Gi}, Q_{Gi} – мощности активной и реактивной генерации в i-ом узле.

Соотношения для определения значений перетоков активной и реактивной мощности в ветви $(i-j)$ содержат комплексные величины напряжения в i-узле (U_{ia}, U_{ir}), комплексные коэффициенты трансформации (K_{ija}, K_{ijr}) и (K_{jia}, K_{jir}), включенные соответственно у i-го и j-го конца ветви; продольные проводимости ветви (y_{ija}, y_{ijr}) и поперечные проводимости у i-го и j-го конца ветви (y_{iija}, y_{iijr}) и (y_{ijja}, y_{ijjr}):

$$
\begin{aligned}
P_{ij} &= K_{ij}^2 U_i^2 \left(y_{ija} + y_{iija}\right) - \\
&- \left(K_{ija}K_{jia} + K_{ijr}K_{jir}\right)\left(\left(U_{ia}U_{ja} + U_{ir}U_{jr}\right)y_{ija} - \left(U_{ia}U_{jr} - U_{ir}U_{ja}\right)y_{ijr}\right) + \\
&+ \left(K_{ija}K_{jir} - K_{ijr}K_{jia}\right)\left(\left(U_{ia}U_{jr} - U_{ir}U_{ja}\right)y_{ija} + \left(U_{ia}U_{ja} + U_{ir}U_{jr}\right)y_{ijr}\right), \\
Q_{ij} &= K_{ij}^2 U_i^2 \left(y_{ijr} + y_{iijr}\right) - \\
&- \left(K_{ija}K_{jia} + K_{ijr}K_{jir}\right)\left(\left(U_{ia}U_{jr} - U_{ir}U_{ja}\right)y_{ija} + \left(U_{ia}U_{ja} + U_{ir}U_{jr}\right)y_{ijr}\right) - \\
&- \left(K_{ija}K_{jir} - K_{ijr}K_{jia}\right)\left(\left(U_{ia}U_{ja} + U_{ir}U_{jr}\right)y_{ija} - \left(U_{ia}U_{jr} - U_{ir}U_{ja}\right)y_{ijr}\right).
\end{aligned}
$$

В приведенных формулах U_i – модуль напряжения в i-узле, K_{ij} – модуль коэффициента трансформации, включенного у i-го конца ветви $(i-j)$.

Ограничения-неравенства:

Для всех узлов должны выполняться условия

$$\overline{U}_i \ge U_i \ge \underline{U}_i, \quad \overline{P}_{Gi} \ge P_{Gi} \ge \underline{P}_{Gi}, \quad \overline{Q}_{Gi} \ge Q_{Gi} \ge \underline{Q}_{Gi},$$

а для всех ветвей – условия

$$\overline{P}_{ij} \ge P_{ij} \ge \underline{P}_{ij}, \quad \overline{K}_{ij} \ge K_{ij} \ge \underline{K}_{ij}.$$

Здесь $\overline{U}_i, \underline{U}_i, \overline{P}_{Gi}, \underline{P}_{Gi}, \overline{Q}_{Gi}, \underline{Q}_{Gi}, \overline{P}_{ij}, \underline{P}_{ij}, \overline{K}_{ij}, \underline{K}_{ij}$ – заданные максимальные и минимальные значения каждого из параметров режима.

Все приведенные параметры режима образуют вектор **z**. Вектор **z** разбивается на два вектора: независимых переменных **x** (вектор регулируемых параметров) и зависимых переменных **y** (базис уравнений). Разбиение производится таким образом, что при заданном **x** система уравнений $\mathbf{W}(\mathbf{x},\mathbf{y}) = \mathbf{0}$ разрешима относительно вектора **y**, то есть можно получить неявную функцию $\mathbf{y}(\mathbf{x})$, такую что $\mathbf{W}(\mathbf{x},\mathbf{y}(\mathbf{x})) \equiv \mathbf{0}$.

Существуют различные процедуры выбора состава компонент вектора **x**, использование которых приводит соответственно к различным наборам регулируемых параметров. Не любой базис обеспечивает получение решения. В

связи с этим логическим продолжением проблемы выбора базиса является проблема смены базиса.

Итогом разбиения переменных является переход к так называемой второй форме математического описания задачи

$$\mathbf{W}(\mathbf{x},\mathbf{y})=\mathbf{0},\quad \mathbf{x}\ge\mathbf{x}\ge\underline{\mathbf{x}},\quad \overline{\mathbf{y}}\ge\mathbf{y}\ge\underline{\mathbf{y}}. \tag{62}$$

Алгоритм решения задачи (62) основан на сочетании метода приведенного градиента и последовательной линеаризации. На каждой итерации $t=0,1,2,\ldots$ (эту итерацию назовем «глобальной», чтобы отличать от «локальных» итераций, возникающих при решении линеаризованной задачи) выполняем [16] следующие действия:

1. Для заданного допустимого по ограничениям-неравенствам вектора $\mathbf{x}^{\mathbf{t}}\in\left[\underline{\mathbf{x}};\overline{\mathbf{x}}\right]$ находим вектор $\mathbf{y}^{\mathbf{t}}$, удовлетворяющий условию $\mathbf{W}\left(\mathbf{x}^{\mathbf{t}},\mathbf{y}^{\mathbf{t}}\right)=\mathbf{0}$.

2. Проверяем, удалось ли получить допустимый режим. Если справедливо условие $\mathbf{y}^{\mathbf{t}}\in\left[\underline{\mathbf{y}};\overline{\mathbf{y}}\right]$, то завершаем вычислительный процесс.

3. Определяем состав ограничений, учитываемых на данной итерации. Номера этих ограничений образуют множество $\mathbf{I}_{\mathbf{v}}^{\mathbf{t}}$. В него обязательно входят номера i, для которых компоненты y_i^t не удовлетворяют ограничениям-неравенствам: $y_i^t\notin\left[\underline{y}_i;\overline{y}_i\right]$. Также сюда могут входить номера i, для которых компоненты y_i^t в некотором смысле близки к границам. В частности, в одном из вариантов алгоритма в множество $\mathbf{I}_{\mathbf{v}}^{\mathbf{t}}$ также включались номера i, такие что $y_i^t\notin\left[\underline{y}_i+\Delta_i;\overline{y}_i-\Delta_i\right]$, где Δ_i – заданные положительные величины, в другом – такие что $y_i^{t-1}\notin\left[\underline{y}_i;\overline{y}_i\right]$.

Данный пункт алгоритма совместно с выбором хорошего базиса позволяют существенно уменьшить размерность исходной задачи. В то же время следует отметить, что рассматриваемая схема алгоритма может быть использована и при отсутствии исключения части ограничений.

4. Пусть $\mathbf{I}_{\mathbf{v}}^{\mathbf{t}}=\{1,\ldots,m\}$. Далее будем считать, что вектор-функция $\mathbf{W}$ состоит только из компонент с номерами из $\mathbf{I}_{\mathbf{v}}^{\mathbf{t}}$. Полагаем $\mathbf{B}^{\mathbf{t}}=\nabla_{\mathbf{y}}\mathbf{W}\left(\mathbf{x}^{\mathbf{t}},\mathbf{y}^{\mathbf{t}}\right)$, $\mathbf{C}^{\mathbf{t}}=\nabla_{\mathbf{x}}\mathbf{W}\left(\mathbf{x}^{\mathbf{t}},\mathbf{y}^{\mathbf{t}}\right)$. Линеаризуем условие $\mathbf{W}(\mathbf{x},\mathbf{y})=\mathbf{0}$ в точке $\left(\mathbf{x}^{\mathbf{t}},\mathbf{y}^{\mathbf{t}}\right)$:

$$\mathbf{W}\left(\mathbf{x}^{\mathbf{t}},\mathbf{y}^{\mathbf{t}}\right)+\mathbf{B}^{\mathbf{t}}\left(\mathbf{y}-\mathbf{y}^{\mathbf{t}}\right)+\mathbf{C}^{\mathbf{t}}\left(\mathbf{x}-\mathbf{x}^{\mathbf{t}}\right)=\mathbf{0}.$$

Раскрывая скобки, получаем

$$\mathbf{y}-\hat{\mathbf{y}}^{\mathbf{t}}=\mathbf{A}^{\mathbf{t}}\mathbf{x},$$

где $\mathbf{A}^{\mathbf{t}}=-\left(\mathbf{B}^{\mathbf{t}}\right)^{-1}\mathbf{C}^{\mathbf{t}}$, $\hat{\mathbf{y}}^{\mathbf{t}}=\mathbf{y}^{\mathbf{t}}+\left(\mathbf{B}^{\mathbf{t}}\right)^{-1}\mathbf{C}^{\mathbf{t}}\mathbf{x}^{\mathbf{t}}$.

5. Решаем точно или приближенно задачу

$$\alpha\to\min,\quad \mathbf{y}=\mathbf{A}^{\mathbf{t}}\mathbf{x}+\hat{\mathbf{y}}^{\mathbf{t}},\quad \overline{\mathbf{x}}\ge\mathbf{x}\ge\underline{\mathbf{x}},\quad \overline{\mathbf{y}}\ge\mathbf{y}+\alpha\mathbf{r}^{\mathbf{t}}\ge\underline{\mathbf{y}},\quad \alpha\ge 0. \tag{63}$$

относительно переменных $\mathbf{x}$, $\mathbf{y}$ и α. Здесь $r_i^t=\left(\overline{y}_i+\underline{y}_i\right)/2-y_i^t$, $i=1,\ldots,m$.

Поскольку набор $\left(\mathbf{x}^{\mathbf{t}}, \mathbf{y}^{\mathbf{t}}, \alpha = 1\right)$ является допустимым, данная задача всегда имеет решение при некотором $\alpha \in [0;1]$. Параметр α здесь является показателем степени несовместности линеаризованной системы. Если оптимальное значение α равняется нулю, то система

$$\overline{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \overline{\mathbf{y}} \geq \mathbf{A}^{\mathbf{t}}\mathbf{x} + \hat{\mathbf{y}}^{\mathbf{t}} \geq \underline{\mathbf{y}}$$

совместна. Чем ближе оптимальное значение α к единице, тем больше будут нарушаться ограничения. Обозначим решение задачи (63) $\tilde{\mathbf{x}}^{\mathbf{t}}$, $\tilde{\mathbf{y}}^{\mathbf{t}}$.

6. Осуществляем итеративный переход по правилам:

$$\mathbf{x}^{\mathbf{t+1}} = \mathbf{x}^{\mathbf{t}} + \rho_t\left(\tilde{\mathbf{x}}^{\mathbf{t}} - \mathbf{x}^{\mathbf{t}}\right), \quad \mathbf{y}^{\mathbf{t+1}} = \mathbf{y}^{\mathbf{t}} + \rho_t\left(\tilde{\mathbf{y}}^{\mathbf{t}} - \mathbf{y}^{\mathbf{t}}\right),$$

где шаг ρ_t выбирается, исходя из двух требований. Во-первых, необходимо, чтобы векторы $\mathbf{x}^{\mathbf{t+1}}$ и $\mathbf{y}^{\mathbf{t+1}}$ находились соответственно в интервалах $\left[\underline{\mathbf{x}}; \overline{\mathbf{x}}\right]$ и $\left[\underline{\mathbf{y}}; \overline{\mathbf{y}}\right]$. Это достигается введением условия $\rho_t \leq 1$. Во-вторых, должна минимизироваться (точно или приближенно) функция

$$\phi(\rho) = \left\| \mathbf{W}\left(\mathbf{x}^{\mathbf{t}} + \rho\left(\tilde{\mathbf{x}}^{\mathbf{t}} - \mathbf{x}^{\mathbf{t}}\right), \mathbf{y}^{\mathbf{t}} + \rho\left(\tilde{\mathbf{y}}^{\mathbf{t}} - \mathbf{y}^{\mathbf{t}}\right)\right)\right\|.$$

Наиболее вычислительно емким является пункт 5. И именно здесь высокую эффективность демонстрируют алгоритмы внутренних точек. Помимо традиционных преимуществ, следует отметить удобство их адаптации к задаче в нестандартной постановке. Кроме того, в некоторых случаях необязательно или даже нецелесообразно решать точно линеаризованную подзадачу (63). Может оказаться достаточным получить некое улучшающее решение, соответствующее некоторой величине α, являющейся верхней оценкой ее оптимального значения. То есть вариантом пункта 5 алгоритма может быть решение системы уравнений и неравенств

$$\mathbf{y} = \mathbf{A}^{\mathbf{t}}\mathbf{x} + \tilde{\mathbf{y}}^{\mathbf{t}}, \quad \overline{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \overline{\mathbf{y}} \geq \mathbf{y} + \alpha\,\mathbf{r}^{\mathbf{t}} \geq \underline{\mathbf{y}}$$

при задаваемом итеративно увеличиваемом $\alpha \in [0;1]$. Для данной модификации алгоритма особенно важную роль играет максимально быстрая идентификация несовместности системы линейных уравнений и неравенств обеспечиваемая методами внутренних точек на основе критерия, аналогичного критерию (54).

Задача идентификации состояния электроэнергетических систем

Задача идентификации состояния некоторой (в частности, электроэнергетической) системы состоит в том, чтобы по ряду измерений найти значения параметров системы, удовлетворяющие всем ее ограничениям и при этом минимально отклоняющиеся от их наблюдаемых значений.

В то же время, ошибки измерений (особенно грубые, возникающие при выходе из строя измерительных приборов, из-за помех в каналах передачи данных или из-за неверных действий операторов) могут стать [17] источником существенного расхождения между истинными и полученными в соот-

ветствии с моделью значениями параметров. Это может привести к принятию неверных решений. Одним из способов уменьшения влияния «плохих данных» является применение в качестве целевой функции критерия Хьюбера.

Рассмотрим модель в линеаризованной постановке. Пусть имеются результаты наблюдений $\hat{x}_i$, $i = 1,...,n$, а также заданы матрица **A** размерности $m \times n$, вектор $\mathbf{b} \in \mathbf{R^m}$ и векторы ограничений $\underline{\mathbf{x}}, \overline{\mathbf{x}} \in \mathbf{R^n}$, $\underline{\mathbf{x}} < \overline{\mathbf{x}}$. Необходимо найти значения параметров, удовлетворяющие условиям

$$\mathbf{Ax} = \mathbf{b}, \quad \underline{\mathbf{x}} \le \mathbf{x} \le \overline{\mathbf{x}},$$

при которых достигает минимума следующая функция:

$$\sum_{j=1}^{n} f_j\left(\frac{x_j - \hat{x}_j}{\sigma_j}\right),$$

где

$$f_j(a) = \begin{cases} 0{,}5a^2, & a \le \overline{a}, \\ \overline{a}a - 0{,}5\overline{a}^2, & a > \overline{a}. \end{cases}$$

Здесь σ_j – коэффициенты, характеризующие погрешности измерений соответствующих параметров.

Данная задача выпуклого квадратичного программирования также была эффективно решена с помощью алгоритмов внутренних точек.

Литература

1. Канторович. Л.В. Математические методы организации и планирования производства. Л.: ЛГУ, 1939, 68с.

2. Данциг Дж. Линейное программирование, его применения и обобщения. М.: Прогресс, 1966, 600с.

3. Дикин И.И. Итеративное решение задач линейного и квадратичного программирования // Доклады АН СССР, 1967, том 174, с.747–748.

4. Karmarkar N. A new polynomial-time algorithm for linear programming // Combinatorica, 1984, №4, pp.373–395.

5. Von Neumann J. Discussion of a maximization problem. Manuscript, Institute for Advanced Studies, Princeton University, Princeton, NJ 08544, USA, 1947.

6. Goldman A., Tucker A. Theory of linear programming // Annals of mathematical studies, 1956, №38, pp.53–97.

7. Зоркальцев В.И. Итеративный алгоритм решения задачи линейного программирования. В кн. Алгоритмы и программы решения задач линейной алгебры и математического программирования. Иркутск, СЭИ СО АН СССР, 1978, с.77–89.

8. Зоркальцев В.И. Метод относительно внутренних точек. Сыктывкар: Коми филиал АН СССР, 1986, 18с.

9. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной оптимизации. М.: Мир, 1972, 240с.

10. Kojima M., Mizuno S., Yoshise A. A polynomial-time algorithm for a class of linear complementarity problems // Mathematical programming, 1989, №44, pp.1–26.

11. Monteiro R., Adler I. Interior path-following primal-dual algorithms. Part 1: Linear programming // Mathematical programming, 1989, №44, pp.27–41.

12. Филатов А.Ю. Развитие алгоритмов внутренних точек и их приложение к системам неравенств. Диссертация на соискание уч. степ. канд. физ.-мат. наук. Иркутск, 2001, 123с.

13. Зоркальцев В.И., Филатов А.Ю. Комбинированные алгоритмы для решения задач линейного программирования и систем линейных неравенств // Оптимизация, управление, интеллект, 2005, №1(9), с.84-93.

14. Черников С.Н. Линейные неравенства. М.: Наука, 1968, 488с.

15. Shanno D. Computing Karmarkar's projection quickly // Mathematical programming, 1988, №41, pp.61-71.

16. Мурашко Н.А., Охорзин Ю.А., Крумм Л.А. и др. Анализ и управление установившимися состояниями электроэнергетических систем. Новосибирск: Наука, 1987, 240с.

17. Гамм А.З., Колосок И.Н. Обнаружение грубых ошибок телеизмерений в электроэнергетических системах. Новосибирск: Наука, 2000, 152с.

Содержание