

2. Алгоритмы оптимизации в конусе скошенного пути для решения пары взаимно-двойственных задач линейного программирования

В данной главе для решения взаимно-двойственных задач (1.1)-(1.2) предлагается и обосновывается класс алгоритмов оптимизации в конусе “скошенного пути”, развивающий идеи алгоритмов оптимизации в конусе пути аналитических центров [15], [20], [21], [22]. Ключевой отличительной особенностью предлагаемых алгоритмов является то, что вычислительный процесс в них может начинаться с любых относительно внутренних точек множеств допустимых решений этих задач. Данный факт создает особенно широкие перспективы использования алгоритмов “скошенного пути” для решения систем нелинейных уравнений и неравенств на базе итеративной линеаризации. Кроме того, сами по себе они являются полезным обобщением, важным как с теоретической, так и с практической точки зрения.

Идейно алгоритмы “скошенного пути” наиболее близки к введенным в [50] Б.Янсенем, К.Русом, Т.Терлаки и Ж.-Ф. Виалом методам следования цели (target-following methods).

Алгоритмы центрального пути будем рассматривать как частный случай предлагаемых алгоритмов. Соответственно, новые процедуры, позволяющие повысить скорость сходимости, обобщим сразу на случай алгоритмов “скошенного пути”. В то же время экспериментальное исследование, результаты которого также приводятся в данной главе, будет проведено и на алгоритмах центрального пути, чтобы продемонстрировать рост эффективности, вызываемый введенными процедурами.

Прежде чем перейти непосредственно к изложению алгоритмов, сформулируем и докажем несколько вспомогательных утверждений. Приводимые утверждения можно считать известными фактами. Их доказательства приведены для полноты изложения.

2.1. Несколько вспомогательных утверждений

Лемма 1

Для любого заданного n -мерного вектора $\mathbf{t} > \mathbf{0}$ безусловный минимум функции

$$\varphi_{\mathbf{t}}(\mathbf{z}) = \sum_{j=1}^n (z_j - t_j \ln z_j)$$

единственен и достигается при

$$z_j = t_j, \quad j = 1, \dots, n. \quad (1)$$

Доказательство

Функция $\varphi_{\mathbf{t}}(\mathbf{z})$ является сепарабельной и строго выпуклой. Получим (1), приравняв к нулю ее частные производные

$$\frac{\partial \varphi_{\mathbf{t}}(\mathbf{z})}{\partial z_j} = 1 - \frac{t_j}{z_j}, \quad j = 1, \dots, n.$$

Лемма 2

Для любого заданного n -мерного вектора $\mathbf{t} > \mathbf{0}$ в задаче

$$\sum_{j=1}^n t_j \ln z_j \rightarrow \max, \quad \sum_{j=1}^n z_j = \sum_{j=1}^n t_j, \quad \mathbf{z} \geq \mathbf{0}$$

существует и единственно оптимальное решение - вектор с компонентами (1).

Доказательство

Умножим целевую функцию на -1 и добавим в нее неизменную по условиям-равенствам величину $\sum z_j$. Получим, что данная задача равносильна следующей:

$$\varphi_{\mathbf{t}}(\mathbf{z}) \rightarrow \min, \quad \sum_{j=1}^n z_j = \sum_{j=1}^n t_j, \quad \mathbf{z} \geq \mathbf{0}. \quad (2)$$

Безусловный минимум функции $\varphi_{\mathbf{t}}(\mathbf{z})$ по лемме 1 единственен и достигается в точке (1). А она удовлетворяет ограничениям задачи (2).

Теорема 2

Пусть для задач линейного программирования (1.1)-(1.2) выполняются предположения 1 и 2. Тогда для любого заданного вектора $\mathbf{t} > \mathbf{0}$ существует и единственная пара векторов $(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$, такая что выполняются условия

$$\mathbf{x}(\mathbf{t}) \in \mathbf{X}, \mathbf{u}(\mathbf{t}) \in \mathbf{U}, x_j(\mathbf{t})g_j(\mathbf{u}(\mathbf{t})) = t_j, j = 1, \dots, n. \quad (3)$$

Пара $(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ является решением задачи минимизации логарифмической барьерной функции при ограничениях задач (1.1)-(1.2):

$$\mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{u} - \sum_{j=1}^n t_j \ln(x_j g_j(\mathbf{u})) \rightarrow \min_{\mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}} \quad (4)$$

Доказательство

Из предположений 1 и 2 следует, что задача (4) имеет допустимое решение с конечным значением целевой функции.

Используя вытекающее из условия $\mathbf{Ax} = \mathbf{b}$ равенство

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

получаем, что задача (4) эквивалентна следующей:

$$\sum_{j=1}^n x_j g_j(\mathbf{u}) - \sum_{j=1}^n t_j \ln(x_j g_j(\mathbf{u})) \rightarrow \min_{\mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}}.$$

Положив $z_j = x_j g_j(\mathbf{u})$ и используя лемму 1, видим, что целевая функции задачи (4) ограничена снизу значением $\sum t_j - \sum t_j \ln t_j$. Поэтому задача (4) имеет оптимальное решение, которое мы обозначим $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$. Поскольку при стремлении каких-либо компонент векторов \mathbf{x} или $\mathbf{g}(\mathbf{u})$ к нулю целевая функция неограниченно возрастает, для оптимальных решений задачи (4) справедливы неравенства $\bar{\mathbf{x}} > \mathbf{0}$ и $\bar{\mathbf{g}} = \mathbf{g}(\bar{\mathbf{u}}) > \mathbf{0}$.

Перепишем задачу (4), опустив ограничения-неравенства, заведомо выполняющиеся в строгой форме на оптимальном решении $(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{g}})$:

$$\mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{u} - \sum_{j=1}^n t_j \ln x_j - \sum_{j=1}^n t_j \ln g_j \rightarrow \min, \quad (5)$$

$$\mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0}, \quad (6)$$

$$\mathbf{c} - \mathbf{A}^T \mathbf{u} - \mathbf{g} = \mathbf{0}. \quad (7)$$

Обозначим $\tilde{\mathbf{u}}, \tilde{\mathbf{x}}$ множители Лагранжа ограничений (6) и (7). Запишем условия оптимальности Лагранжа:

$$c_j - t_j / \bar{x}_j = (\mathbf{A}^T \tilde{\mathbf{u}})_j, \quad j = 1, \dots, n,$$

$$-t_j / \bar{g}_j = -\tilde{x}_j, \quad j = 1, \dots, n,$$

$$-\mathbf{b} = -\mathbf{A} \tilde{\mathbf{x}}.$$

Обозначив $\tilde{\mathbf{g}} = \mathbf{c} - \mathbf{A}^T \tilde{\mathbf{u}}$, перепишем условия в следующем виде:

$$\bar{x}_j \tilde{g}_j = t_j, \quad j = 1, \dots, n, \quad (8)$$

$$\tilde{x}_j \bar{g}_j = t_j, \quad j = 1, \dots, n, \quad (9)$$

$$\mathbf{A} \tilde{\mathbf{x}} = \mathbf{b}.$$

Поскольку $\bar{\mathbf{x}} > \mathbf{0}$ и $\mathbf{t} > \mathbf{0}$, из (8) следует, что $\tilde{\mathbf{g}} > \mathbf{0}$. Следовательно, $\tilde{\mathbf{u}} \in \mathbf{U}$.

Таким образом, существует конкретная пара векторов $(\bar{\mathbf{x}}, \tilde{\mathbf{u}})$, для которой выполняются условия (3).

По лемме 1 набор векторов $(\bar{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{\mathbf{g}})$ доставляет безусловный минимум функции (5). Поскольку для него выполняются ограничения (6)-(7), то он будет и оптимальным решением задачи (5)-(7).

Любая пара векторов, удовлетворяющая условиям (3), является оптимальным решением задачи (5)-(7), поскольку, подставляя условие $x_j(\mathbf{t})g_j(\mathbf{u}(\mathbf{t})) = t_j, \quad j = 1, \dots, n$ в целевую функцию, получаем ее значение равным $\sum t_j - \sum t_j \ln t_j$. Задача (5)-(7) состоит в минимизации выпуклой функции при линейных ограничениях. Причем целевая функция строго выпукла по переменным x_j и g_j . Значит, по векторам \mathbf{x} и \mathbf{g} задача имеет единственное решение. То есть вектор $\bar{\mathbf{g}}$ совпадает с $\tilde{\mathbf{g}}$. Тогда из (9) следует,

что вектор $\tilde{\mathbf{x}}$ совпадает с $\bar{\mathbf{x}}$. Поскольку $\text{rank } \mathbf{A} = m$, а $m \leq n$, то каждому заданному вектору $\bar{\mathbf{g}}$ может соответствовать не более одного вектора $\bar{\mathbf{u}}$, для которого выполняется условие $\bar{\mathbf{g}} = \mathbf{c} - \mathbf{A}^T \bar{\mathbf{u}}$. То есть вектор $\bar{\mathbf{u}}$ совпадает с $\tilde{\mathbf{u}}$. Таким образом, решение единственно. Теорема доказана.

Следствие

Пара векторов $(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ является также оптимальным решением задачи

$$\sum_{j=1}^n t_j \ln x_j + \sum_{j=1}^n t_j \ln g_j(\mathbf{u}) \rightarrow \max_{\mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}}, \quad \mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{u} = \sum_{j=1}^n t_j. \quad (10)$$

Доказательство

Пара $(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ является допустимым решением задачи (10). Действительно,

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

а остальные ограничения выписаны непосредственно в (3). Приведя задачу (10) к виду

$$\sum_{j=1}^n t_j \ln z_j \rightarrow \max, \quad \sum_{j=1}^n z_j = \sum_{j=1}^n t_j, \quad z_j = x_j g_j(\mathbf{u}), \quad \mathbf{x} \in \mathbf{X}, \mathbf{u} \in \mathbf{U}$$

и используя лемму 2, получаем, что пара векторов $(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ является ее оптимальным решением.

2.2. Описание и теоретическое исследование

алгоритмов оптимизации в конусе скошенного пути

Скошенный путь

Пусть задан некоторый иницирующий вектор $\mathbf{t} > \mathbf{0}$. По теореме 2 для любой скалярной величины $\mu > 0$, называемой в данном контексте параметром скошенного пути, существует и единственная пара векторов $(\mathbf{x}(\mu, \mathbf{t}), \mathbf{u}(\mu, \mathbf{t}))$, удовлетворяющая условиям

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

Данная пара векторов называется точкой скошенного пути. Множество пар векторов $(\mathbf{x}(\mu, \mathbf{t}), \mathbf{u}(\mu, \mathbf{t}))$ при всех $\mu > 0$ образует скошенный путь, иницируемый вектором \mathbf{t} . Исходя из данного определения, скошенный путь можно интерпретировать как множество всех допустимых решений задач (1.1)-(1.2), для которых компоненты вектора невязок двойственности пропорциональны соответствующим компонентам иницирующего вектора.

Разные иницирующие векторы порождают разные скошенные пути. В частности, единичный вектор $\mathbf{t} = \mathbf{e}$ (равно как и вектор $\mathbf{t} = \lambda \mathbf{e}$ при любом положительном λ) иницирует путь аналитических центров. Поэтому алгоритмы центрального пути можно считать частным случаем алгоритмов, предлагаемых здесь.

Каждый скошенный путь можно характеризовать коэффициентом скошенности, который показывает отношение среднего значения компонент иницирующего вектора к его минимальной компоненте

$$\gamma = \bar{t}/t_{\min},$$

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

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

Точки каждого из скошенных путей при $\mu \rightarrow 0$ сходятся к оптимальным решениям задач (1.1)-(1.2). Точнее, каждый скошенный путь сходится к единственной точке $(\mathbf{x}(0, \mathbf{t}), \mathbf{u}(0, \mathbf{t}))$, которой можно доопределить его при $\mu = 0$:

$$\mathbf{x}(0, \mathbf{t}) = \arg \max_{\mathbf{x} \in \text{ri } \bar{\mathbf{X}}} \sum_{j \in J(\bar{\mathbf{X}})} t_j \ln x_j,$$

$$\mathbf{u}(0, \mathbf{t}) = \arg \max_{\mathbf{u} \in ri \bar{\mathbf{U}}} \sum_{j \in J_0(\bar{\mathbf{X}})} t_j \ln g_j(\mathbf{u}).$$

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

Алгоритмы скошенного пути развивают идеи включения в целевые функции задач (1.1)-(1.2) логарифмических штрафов за приближение к границам множеств допустимых решений. Действительно, вектор $\mathbf{x}(\mu, \mathbf{t})$ является решением задачи

$$\mathbf{c}^T \mathbf{x} - \mu \sum_{j=1}^n t_j \ln x_j \rightarrow \min, \quad \mathbf{A} \mathbf{x} = \mathbf{b}. \quad (11)$$

Вектор $\mathbf{u}(\mu, \mathbf{t})$ состоит из множителей Лагранжа ее ограничений. Также $\mathbf{u}(\mu, \mathbf{t})$ можно определить как решение задачи

$$\mathbf{b}^T \mathbf{u} + \mu \sum_{j=1}^n t_j \ln g_j \rightarrow \max, \quad \mathbf{g} + \mathbf{A}^T \mathbf{u} = \mathbf{c}. \quad (12)$$

$\mathbf{x}(\mu, \mathbf{t})$ составляет вектор множителей Лагранжа ограничений.

Точки скошенного пути можно также интерпретировать как наиболее удаленные от границ множества допустимых решений задачи (1.3) среди имеющих одно и то же значение целевой функции, равное $\mu \sum t_j$. Удаленность здесь измеряется взвешенной суммой логарифмов. Действительно, по теореме 2 пара векторов $(\mathbf{x}(\mu, \mathbf{t}), \mathbf{u}(\mu, \mathbf{t}))$ является решением задачи

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

Конус скошенного пути

Далее считаем зафиксированным вектор $\mathbf{t} > \mathbf{0}$. Рассмотрим расширение скошенного пути, инициированного этим вектором, которое назовем конусом

скошенного пути. Он состоит из всех пар векторов (\mathbf{x}, \mathbf{u}) , таких что $\mathbf{x} \in ri \mathbf{X}$, $\mathbf{u} \in ri \mathbf{U}$ и существует $\mu > 0$, при котором выполняется неравенство

$$\Phi_2(\mathbf{x}, \mathbf{u}, \mu) \leq \theta \mu t_{\min}, \quad (13)$$

где

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

Здесь θ - заданный неотрицательный параметр. Величину $\sqrt{\theta}$ можно интерпретировать как радиус конуса скошенного пути.

Следует отметить, что в частном случае при $\mathbf{t} = \mathbf{e}$ функция $\Phi_2(\mathbf{x}, \mathbf{u}, \mu)$ совпадает с определенной в главе 1 при рассмотрении алгоритмов центрального пути. Этим объясняется неизменность обозначений, хотя новая функция не эквивалентна введенной в предыдущей главе, а является ее обобщением.

Выполнение условия (13) влечет за собой справедливость неравенств

$$(1 - \sqrt{\theta})\mu t_j \leq x_j g_j(\mathbf{u}) \leq (1 + \sqrt{\theta})\mu t_j, \quad j = 1, \dots, n, \quad (14)$$

из которых следует, что $x_j g_j(\mathbf{u}) \rightarrow 0$ для всех $j = 1, \dots, n$ при $\mu \rightarrow 0$. Значит, если пары векторов $(\mathbf{x}^k, \mathbf{u}^k)$ при $k = 1, 2, \dots$ принадлежат конусу скошенного пути при соответствующих значениях μ^k , а $\lim_{k \rightarrow \infty} \mu^k = 0$, то последовательности $\{\mathbf{x}^k\}$ и $\{\mathbf{u}^k\}$ сходятся при $k \rightarrow \infty$ к оптимальным решениям задач (1.1)-(1.2), а именно к определенным выше векторам $\mathbf{x}(0, \mathbf{t})$ и $\mathbf{u}(0, \mathbf{t})$.

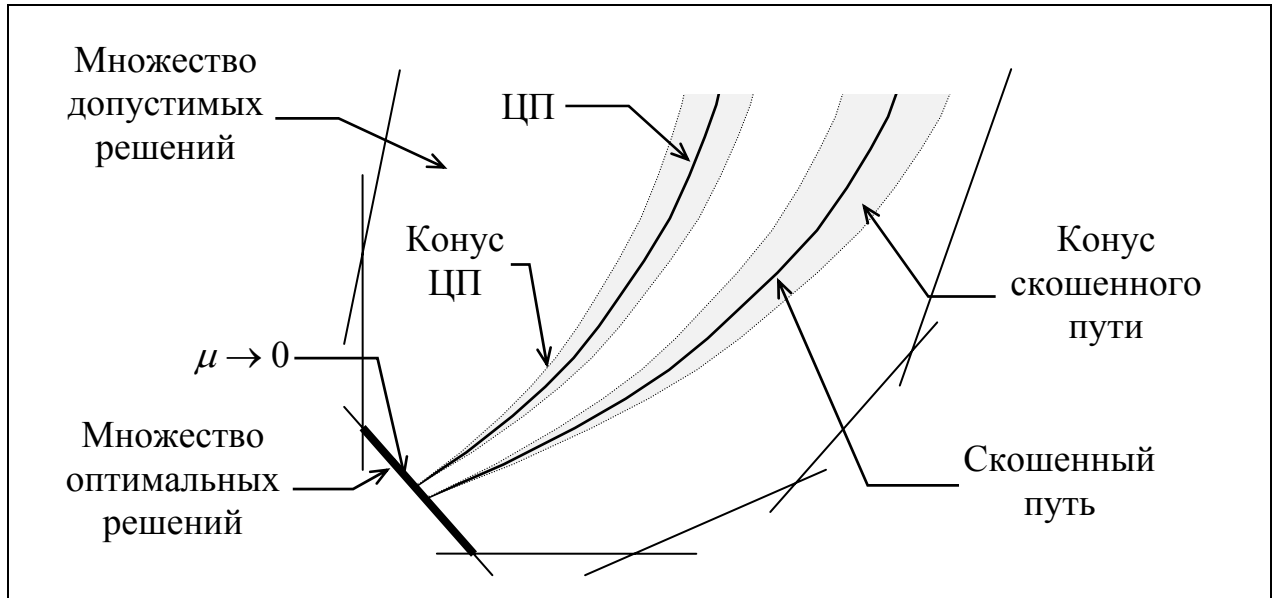
Попутно отметим, что из (14) и определения коэффициента скошенности γ вытекает

$$\sum_{j=1}^n x_j g_j(\mathbf{u}) \leq (1 + \sqrt{\theta})\mu \sum_{j=1}^n t_j = (1 + \sqrt{\theta})n\gamma\mu t_{\min}.$$

Следовательно, для того, чтобы показать, что невязка двойственности $\sum x_j g_j(\mathbf{u})$ не превышает ε , достаточно выполнения условия

$$\mu t_{\min} \leq \frac{\varepsilon}{(1 + \sqrt{\theta})n\gamma}.$$

Схематически траекторию алгоритмов скошенного пути покажем на следующем рисунке:



Как и для алгоритмов оптимизации в конусе центрального пути, далее будем считать, что $\theta \in (0;1)$. В этом случае из неравенства (13) следует, что $x_j g_j(\mathbf{u}) > 0$, и, чтобы показать, что $\mathbf{x} \in ri\mathbf{X}$, а $\mathbf{u} \in ri\mathbf{U}$, необходимо проверить только выполнение равенства $\mathbf{Ax} = \mathbf{b}$ и положительность компонент одного из векторов: \mathbf{x} или $\mathbf{g}(\mathbf{u})$.

Все рассматриваемые ниже алгоритмы вырабатывают последовательности точек $(\mathbf{x}^k, \mathbf{u}^k)$, принадлежащих конусу скошенного пути, и значений μ^k , таких что для $\mathbf{x}^k, \mathbf{u}^k, \mu^k$ справедливо (13). При этом для некоторого $\beta > 0$ выполняется неравенство

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

Из этого следует [60] получение оптимального решения пары задач (1.1)-(1.2) за $O(\sqrt{n}L)$ итераций.

Все алгоритмы предполагают наличие стартового приближения, принадлежащего конусу скошенного пути. То есть при $k=1$ должны выполняться условия

$$\mathbf{A}\mathbf{x}^k = \mathbf{b}, \mathbf{x}^k > \mathbf{0}, \mathbf{g}(\mathbf{u}^k) > \mathbf{0}, \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^{k+1} t_{\min}. \quad (15)$$

По сути для алгоритмов скошенного пути это означает необходимость наличия любой относительно внутренней точки $(\mathbf{x}^1, \mathbf{u}^1)$ множества допустимых решений, поскольку, выбрав иницирующий вектор с компонентами $t_j = x_j^1 g_j(\mathbf{u}^1)$, $j = 1, \dots, n$, мы получим, что точка $(\mathbf{x}^1, \mathbf{u}^1)$ принадлежит данному скошенному пути.

Алгоритм А

Перейдем непосредственно к изложению алгоритмов. Начнем с наиболее простого варианта, итеративный переход в котором осуществляется по правилам

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{u} \in \mathbf{R}^m} \Phi_2(\mathbf{x}^k, \mathbf{u}, \mu^k), \quad (16)$$

$$x_j^{k+1} = 2x_j^k - \frac{1}{\mu^k t_j} (x_j^k)^2 g_j(\mathbf{u}^{k+1}), \quad j = 1, \dots, n, \quad (17)$$

$$\mu^{k+1} = \left(1 - \frac{\sqrt{\theta(1-\theta)n\gamma - \theta}}{n\gamma - \theta} \right) \mu^k. \quad (18)$$

Решением задачи (16) является вектор

$$\mathbf{u}^{k+1} = (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T)^{-1} (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} - \mathbf{b}). \quad (19)$$

Здесь и далее $\mathbf{X}_k = \text{diag}\{x_j^k\}$, $\mathbf{M}_k = \text{diag}\{\mu^k t_j\}$.

Идейной основой перехода (17) является использование квадратичной аппроксимации задачи (11) в точке $\mathbf{x} = \mathbf{x}^k$, $\mu = \mu^k$. Действительно, решая методом Лагранжа задачу

$$\mathbf{c}^T \Delta \mathbf{x} - \mu^k \sum_{j=1}^n t_j \left(\frac{\Delta x_j}{x_j^k} - \frac{1}{2} \left(\frac{\Delta x_j}{x_j^k} \right)^2 \right) \rightarrow \min, \quad (20)$$

$$\mathbf{A} \Delta \mathbf{x} = \mathbf{0}, \quad (21)$$

получим

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

где \mathbf{u}^{k+1} - вектор множителей Лагранжа ограничений (21), который вычисляется по формуле (19).

Теорема 3

Для алгоритма (16)-(18) из выполнения условий (15) на k -итерации следует выполнение условий (15) на $(k+1)$ -итерации.

Доказательство

Начнем доказательство с проверки балансовых ограничений-равенств.

Используя выражения (17) и (19) для \mathbf{x}^{k+1} и \mathbf{u}^{k+1} , получим

$$\begin{aligned} \mathbf{A} \mathbf{x}^{k+1} &= 2 \mathbf{A} \mathbf{x}^k - \frac{1}{\mu^k} \mathbf{A} \mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} + \\ &+ \frac{1}{\mu^k} \mathbf{A} \mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T \left(\mathbf{A} \mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T \right)^{-1} \left(\mathbf{A} \mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} - \mu^k \mathbf{b} \right) = \mathbf{b}. \end{aligned}$$

Принимая во внимание (16),

$$\Phi_2(\mathbf{x}^k, \mathbf{u}^{k+1}, \mu^k) \leq \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}.$$

Поскольку $\theta \in (0;1)$, из этого следует, что $x_j^k g_j(\mathbf{u}^{k+1}) > 0$, $j = 1, \dots, n$. Учитывая $\mathbf{x}^k > \mathbf{0}$, получаем, что $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$.

На дальнейшем этапе доказательства ключевую роль играет следующее выводимое из (17) равенство:

$$\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) = \frac{1}{\mu^k t_j} \left(\mu^k t_j - x_j^k g_j(\mathbf{u}^{k+1}) \right)^2, \quad j = 1, \dots, n. \quad (22)$$

Из него следует

$$\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \geq 0, \quad j = 1, \dots, n. \quad (23)$$

Также, проведя ряд преобразований, получим

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^k) \leq \theta^2 \mu^k t_{\min}. \quad (24)$$

Действительно,

$$\begin{aligned} \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^k) &= \sum_{j=1}^n \frac{1}{\mu^k t_j} \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^2 \leq \\ &\leq \frac{1}{\mu^k t_{\min}} \sum_{j=1}^n \frac{1}{(\mu^k t_j)^2} \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^4 \leq \\ &\leq \frac{1}{\mu^k t_{\min}} \left(\sum_{j=1}^n \frac{1}{\mu^k t_j} \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^2 \right)^2 = \\ &= \frac{1}{\mu^k t_{\min}} \left(\Phi_2(\mathbf{x}^k, \mathbf{u}^{k+1}, \mu^k) \right)^2 \leq \frac{1}{\mu^k t_{\min}} \left(\Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \right)^2 \leq \theta^2 \mu^k t_{\min}. \end{aligned}$$

Определим, при каких значениях α будет выполняться неравенство $\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, (1-\alpha)\mu^k) \leq \theta(1-\alpha)\mu^k t_{\min}$. На основе выражения (22) и неравенств (23)-(24), получим

$$\begin{aligned} \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, (1-\alpha)\mu^k) &= \sum_{j=1}^n \frac{1}{(1-\alpha)\mu^k t_j} \left((1-\alpha)\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^2 = \\ &= \frac{1}{1-\alpha} \sum_{j=1}^n \frac{1}{\mu^k t_j} \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) - \alpha \mu^k t_j \right)^2 \leq \\ &\leq \frac{1}{1-\alpha} \left(\sum_{j=1}^n \frac{1}{\mu^k t_j} \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^2 - \right. \\ &\quad \left. - 2\alpha \sum_{j=1}^n \left(\mu^k t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right) + \alpha^2 n \gamma \mu^k t_{\min} \right) \leq \\ &\leq \frac{1}{1-\alpha} \left(\theta^2 \mu^k t_{\min} + \alpha^2 n \gamma \mu^k t_{\min} \right). \end{aligned}$$

Решим квадратное относительно α неравенство

$$\theta^2 \mu^k t_{\min} + \alpha^2 n \gamma \mu^k t_{\min} \leq \theta(1-\alpha)^2 \mu^k t_{\min},$$

$$(n\gamma - \theta)\alpha^2 + 2\theta\alpha + \theta^2 - \theta \leq 0,$$

$$\frac{-\theta - \sqrt{\theta^2 - (n\gamma - \theta)(\theta^2 - \theta)}}{n\gamma - \theta} \leq \alpha \leq \frac{-\theta + \sqrt{\theta^2 - (n\gamma - \theta)(\theta^2 - \theta)}}{n\gamma - \theta},$$

$$\frac{-\theta - \sqrt{n\gamma\theta(1-\theta) + \theta^3}}{n\gamma - \theta} \leq \alpha \leq \frac{-\theta + \sqrt{n\gamma\theta(1-\theta) + \theta^3}}{n\gamma - \theta}.$$

Отсюда очевидно, что при переходе (18) выполняется условие

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1}. \quad (25)$$

Неравенство (25), а также выполнение условия $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$ влекут за собой справедливость неравенства $\mathbf{x}^{k+1} > \mathbf{0}$. Теорема доказана.

Алгоритм В

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 (18), а путем решения вспомогательной задачи

$$\mu \rightarrow \min_{\mu \geq 0}, \quad \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu) \leq \theta \mu t_{\min}. \quad (26)$$

Оптимальное значение здесь достигается при выполнении точного равенства

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) = \theta \mu^{k+1} t_{\min}. \quad (27)$$

Таким образом, задача (26) сводится к определению корня квадратного уравнения. Решение нетрудно выводится аналитически и имеет вид

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

Такое правило дает гарантированно не худшую сходимость, чем алгоритм А. Действительно, μ^{k+1} , вычисляемое по формуле (18), является допустимым, но не оптимальным решением задачи (26).

Легко проверить также, что заново вычисленная точка будет принадлежать конусу скошенного пути:

$$\mathbf{Ax}^{k+1} = \mathbf{b}, \quad \mathbf{x}^{k+1} > \mathbf{0}, \quad \mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}, \quad \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min}.$$

Первые три условия выполняются, исходя из соображений, используемых при обосновании алгоритма А, а четвертое - исходя из равенства (27).

Гарантированные оценки сокращения параметра скошенного пути

Начиная с алгоритма С, значение параметра скошенного пути μ будет гарантированно уменьшаться на каждой итерации $k > 1$ в соответствии с неравенством

$$\mu^{k+1} \leq \left(1 - \sqrt{\frac{\theta(1-\theta)}{n\gamma - \theta}} \right) \mu^k.$$

Поскольку при любых натуральных n и $\theta \in (0;1)$

$$1 - \sqrt{\frac{\theta(1-\theta)}{n\gamma - \theta}} < 1 - \sqrt{\frac{\theta(1-\theta)}{n\gamma}} < 1 - \frac{\sqrt{\theta(1-\theta)n\gamma - \theta}}{n\gamma - \theta}, \quad (28)$$

последующие алгоритмы С, D, E и их модификации сходятся гарантированно быстрее, чем алгоритм А.

Кроме того, неравенства (28) показывают, что выражение, стоящее в центральной их части, можно использовать в качестве оценки показателя скорости сходимости алгоритмов.

Наилучшее значение оценки достигается, когда максимально значение величины $\theta(1-\theta)$, то есть при $\theta = 0.5$. В этом случае

$$\sqrt{\frac{\theta(1-\theta)}{n\gamma}} = \frac{0.5}{\sqrt{n\gamma}}.$$

Отсюда получаем, что в предлагаемых алгоритмах уменьшение параметра скошенного пути может осуществляться в соответствии с неравенством

$$\mu^{k+1} < \left(1 - 0.5/\sqrt{n\gamma} \right) \mu^k.$$

Это как превосходит оценки, выполняющиеся для алгоритмов следования цели [50], так и в частном случае при $\gamma = 1$ является наилучшей оценкой для класса алгоритмов оптимизации в конусе центрального пути.

Алгоритм С

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

$$\mathbf{u}^k(\lambda) = \arg \min_{\mathbf{u} \in \mathbb{R}^m} \Phi_2(\mathbf{x}^k, \mathbf{u}, \lambda \mu^k), \quad (29)$$

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

$$\begin{aligned} \mathbf{u}^k(1) &= (\mathbf{A}\mathbf{X}_k^2\mathbf{M}_k^{-1}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{X}_k^2\mathbf{M}_k^{-1}\mathbf{c} - \mathbf{b}), \\ \mathbf{u}^k(0) &= (\mathbf{A}\mathbf{X}_k^2\mathbf{M}_k^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}_k^2\mathbf{M}_k^{-1}\mathbf{c}. \end{aligned}$$

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

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_2(\mathbf{x}^k, \mathbf{u}^k(\lambda), \lambda \mu^k) \leq \theta \lambda \mu^k t_{\min}. \quad (30)$$

При этом ее ограничение выполняется в виде строгого равенства, то есть задача (30) сводится к вычислению минимального из корней квадратного уравнения и решается аналитически [21]. Итеративный переход осуществляем по формулам

$$\mathbf{u}^{k+1} = \mathbf{u}^k(\lambda^k), \quad \mu^{k+1} = \lambda^k \mu^k, \quad x_j^{k+1} = 2x_j^k - \frac{1}{\mu^{k+1} t_j} (x_j^k)^2 g_j(\mathbf{u}^{k+1}). \quad (31)$$

Теорема 4

Для алгоритма (29)-(31) из выполнения условий

$$\mathbf{x}^k \in \text{ri } \mathbf{X}, \quad \mathbf{u}^k \in \text{ri } \mathbf{U}, \quad \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}$$

следует выполнение следующих соотношений:

$$\mathbf{x}^{k+1} \in ri \mathbf{X}, \mathbf{u}^{k+1} \in ri \mathbf{U}, \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min}.$$

При этом

$$\mu^{k+1} \leq \mu^k,$$

а при $k > 1$ выполняется неравенство

$$\mu^{k+1} \leq \left(1 - \sqrt{\frac{\theta(1-\theta)}{n\gamma - \theta}}\right) \mu^k. \quad (32)$$

Доказательство

Начнем доказательство с проверки сохранения балансовых ограничений-равенств. Явное выражение для вектор-функции $\mathbf{u}^k(\lambda)$ можно записать в следующем виде:

$$\mathbf{u}^k(\lambda) = (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T)^{-1} (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} - \lambda \mathbf{b}). \quad (33)$$

$$\begin{aligned} \mathbf{A}\mathbf{x}^{k+1} &\stackrel{(31)}{=} 2\mathbf{A}\mathbf{x}^k - \frac{1}{\lambda^k} \mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} (\mathbf{c} - \mathbf{A}^T \mathbf{u}^{k+1}) \stackrel{(33)}{=} \\ &= 2\mathbf{b} - \frac{1}{\lambda^k} \mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} + \frac{1}{\lambda^k} \mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{A}^T)^{-1} (\mathbf{A}\mathbf{X}_k^2 \mathbf{M}_k^{-1} \mathbf{c} - \lambda^k \mathbf{b}) = \mathbf{b}. \end{aligned}$$

Решение задачи (30) достигается при выполнении ее ограничения как строгого равенства, следовательно

$$\Phi_2(\mathbf{x}^k, \mathbf{u}^{k+1}, \mu^{k+1}) = \theta \mu^{k+1} t_{\min}. \quad (34)$$

Убедимся в выполнении ограничений задачи (30) при $\lambda = 1$:

$$\Phi_2(\mathbf{x}^k, \mathbf{u}^k(1), \mu^k) \stackrel{(29)}{\leq} \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}.$$

Таким образом, решение задачи (30) достигается при $\lambda \leq 1$. Отсюда следует справедливость неравенства

$$\mu^{k+1} \leq \mu^k.$$

Поскольку по условию $\mathbf{x}^k > \mathbf{0}$, то из (34) следует, что $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$.

Ключевым на дальнейших шагах доказательства является следующее соотношение, выводимое из (31):

$$\begin{aligned}\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) &= \mu^{k+1}t_j - 2x_j^k g_j(\mathbf{u}^{k+1}) + \frac{1}{\mu^{k+1}t_j} (x_j^k)^2 (g_j(\mathbf{u}^{k+1}))^2, \\ \mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) &= \frac{1}{\mu^{k+1}t_j} (\mu^{k+1}t_j - x_j^k g_j(\mathbf{u}^{k+1}))^2.\end{aligned}\quad (35)$$

Из этого равенства, а также из (34) следует

$$\sum_{j=1}^n (\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1})) = \theta \mu^{k+1}t_{\min}. \quad (36)$$

Теперь можем показать справедливость неравенства

$$\begin{aligned}\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) &\leq \theta \mu^{k+1}t_{\min}. \\ \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) &= \sum_{j=1}^n \frac{1}{\mu^{k+1}t_j} (\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}))^2 \stackrel{(35)}{\leq} \\ &\leq \frac{1}{\mu^{k+1}t_{\min}} \sum_{j=1}^n \frac{1}{(\mu^{k+1}t_j)^2} (\mu^{k+1}t_j - x_j^k g_j(\mathbf{u}^{k+1}))^4 \leq \\ &\leq \frac{1}{\mu^{k+1}t_{\min}} \left(\sum_{j=1}^n \frac{1}{\mu^{k+1}t_j} (\mu^{k+1}t_j - x_j^k g_j(\mathbf{u}^{k+1}))^2 \right)^2 = \theta^2 \mu^{k+1}t_{\min} < \theta \mu^{k+1}t_{\min}.\end{aligned}$$

Отсюда, учитывая, что $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$, получаем $\mathbf{x}^{k+1} > \mathbf{0}$.

Таким образом, остается доказать только неравенство (32). Пусть $z_j^{k+1} = x_j^{k+1}g_j(\mathbf{u}^{k+1})$. Найдем, при каких α выполняется неравенство

$$\sum_{j=1}^n \frac{1}{\mu^{k+1}t_j(1-\alpha)} (\mu^{k+1}t_j - z_j^{k+1} - \alpha \mu^{k+1}t_j)^2 \leq \theta(1-\alpha)\mu^{k+1}t_{\min}. \quad (37)$$

Раскроем скобки, выполним ряд преобразований, получим:

$$\begin{aligned}&\left(\sum_{j=1}^n \frac{1}{\mu^{k+1}t_j} (\mu^{k+1}t_j - z_j^{k+1})^2 - 2\alpha \sum_{j=1}^n (\mu^{k+1}t_j - z_j^{k+1}) + \alpha^2 \sum_{j=1}^n \mu^{k+1}t_j \right) / (1-\alpha) \stackrel{(36)}{\leq} \\ &\leq (\theta^2 \mu^{k+1}t_{\min} - 2\alpha\theta \mu^{k+1}t_{\min} + \alpha^2 n \mu^{k+1}t_{\min}) / (1-\alpha).\end{aligned}$$

Решаем относительно α неравенство

$$\left(\theta^2 \mu^{k+1} t_{\min} - 2\alpha\theta \mu^{k+1} t_{\min} + \alpha^2 n\gamma \mu^{k+1} t_{\min}\right) / (1-\alpha) \leq \theta(1-\alpha) \mu^{k+1} t_{\min}.$$

Умножим обе части неравенства на $\frac{1-\alpha}{\mu^{k+1} t_{\min}}$:

$$\theta^2 - 2\alpha\theta + \alpha^2 n\gamma \leq \theta(1 - 2\alpha + \alpha^2),$$

$$\alpha^2 (n\gamma - \theta) \leq \theta - \theta^2.$$

Отсюда получим, что неравенство справедливо при $\alpha \leq \sqrt{\theta}\sqrt{1-\theta}/\sqrt{n\gamma-\theta}$.

Следовательно, при $\alpha = \sqrt{\theta}\sqrt{1-\theta}/\sqrt{n\gamma-\theta}$ выполняется и условие (37),

то есть

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, (1 - \sqrt{\theta}\sqrt{1-\theta}/\sqrt{n\gamma-\theta})\mu^{k+1}) \leq \theta(1 - \sqrt{\theta}\sqrt{1-\theta}/\sqrt{n\gamma-\theta})\mu^{k+1}t_{\min}.$$

Значит, на $(k+2)$ -итерации можно гарантированно продвинуться в соответствии с оценкой (32), даже не меняя значений переменных. Следовательно, начиная со второй итерации, выполняются все условия теоремы. Теорема доказана.

Алгоритм D

Данный алгоритм является зеркальным отражением алгоритма C. Его идейной основой является решение методом Ньютона задачи максимизации двойственной логарифмической барьерной функции (12), поэтому алгоритм можно назвать двойственным.

Квадратичная аппроксимация задачи (12) в точке $\mathbf{u} = \mathbf{u}^k$, $\mu = \mu^k$ выглядит следующим образом:

$$\mathbf{b}^T \Delta \mathbf{u} + \mu^k \sum_{j=1}^n t_j \left(\frac{\Delta g_j}{g_j(\mathbf{u}^k)} - \frac{1}{2} \left(\frac{\Delta g_j}{g_j(\mathbf{u}^k)} \right)^2 \right) \rightarrow \max_{\Delta \mathbf{u}}, \quad \Delta \mathbf{g} = -\mathbf{A}^T \Delta \mathbf{u}. \quad (38)$$

Решение задачи (38) находится по формулам

$$\Delta \mathbf{u}^k = \frac{1}{\mu^k} \mathbf{r}^k = \frac{1}{\mu^k} (\mathbf{A} \mathbf{G}_k^{-2} \mathbf{T} \mathbf{A}^T)^{-1} (\mathbf{b} - \mu^k \mathbf{A} \mathbf{G}_k^{-1} \mathbf{t}),$$

где, как и в дальнейшем, $\mathbf{G}_k = \text{diag}\{g_j(\mathbf{u}^k)\}$, $\mathbf{T} = \text{diag}\{t_j\}$.

Вектор \mathbf{x}^{k+1} состоит из множителей Лагранжа ограничений-равенств из (38) и вычисляется по правилу

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

Можно попутно отметить, что он является одновременно решением задачи

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

что также подтверждает полную симметричность прямого и двойственного алгоритмов.

Перейдем непосредственно к описанию алгоритма D. Так же, как и в алгоритме C, будем использовать параметризацию. Закрепим вектор \mathbf{u}^k переменных двойственной задачи и введем линейную вектор-функцию

$$\mathbf{x}^k(\lambda) = \arg \min_{\mathbf{x} \in \mathbf{R}^n: \mathbf{A} \mathbf{x} = \mathbf{b}} \Phi_2(\mathbf{x}, \mathbf{u}^k, \lambda \mu^k). \quad (39)$$

На основе стандартной техники получим, что

$$\mathbf{x}^k(\lambda) = \mathbf{G}_k^{-2} \mathbf{T} \mathbf{A}^T \mathbf{r}^k(\lambda) + \lambda \mu^k \mathbf{G}_k^{-1} \mathbf{t}, \quad (40)$$

где

$$\mathbf{r}^k(\lambda) = (\mathbf{A} \mathbf{G}_k^{-2} \mathbf{T} \mathbf{A}^T)^{-1} (\mathbf{b} - \lambda \mu^k \mathbf{A} \mathbf{G}_k^{-1} \mathbf{t}). \quad (41)$$

λ^k находим как решение задачи

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_2(\mathbf{x}^k(\lambda), \mathbf{u}^k, \lambda \mu^k) \leq \theta \lambda \mu^k t_{\min}. \quad (42)$$

После чего осуществляем итеративный переход:

$$\mathbf{x}^{k+1} = \mathbf{x}^k(\lambda^k), \quad \mu^{k+1} = \lambda^k \mu^k, \quad \mathbf{r}^{k+1} = \mathbf{r}^k(\lambda^k), \quad u_i^{k+1} = u_i^k + \frac{1}{\mu^{k+1}} r_i^{k+1}. \quad (43)$$

Теорема 5

Для алгоритма (39)-(43) из выполнения условий

$$\mathbf{x}^k \in ri \mathbf{X}, \mathbf{u}^k \in ri \mathbf{U}, \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}$$

следует справедливость соотношений

$$\mathbf{x}^{k+1} \in ri \mathbf{X}, \mathbf{u}^{k+1} \in ri \mathbf{U}, \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min}.$$

При этом $\mu^{k+1} \leq \mu^k$, а при $k > 1$ выполняется неравенство (32).

Доказательство

Начнем доказательство с проверки сохранения балансовых ограничений-равенств. При любом λ , в том числе при λ^k , выполняется равенство

$$\begin{aligned} \mathbf{A}\mathbf{x}^k(\lambda) &= \mathbf{A}\mathbf{G}_k^{-2}\mathbf{T}\mathbf{A}^T\mathbf{r}^k(\lambda) + \lambda\mu^k\mathbf{A}\mathbf{G}_k^{-1}\mathbf{t} = \\ &= \mathbf{A}\mathbf{G}_k^{-2}\mathbf{T}\mathbf{A}^T(\mathbf{A}\mathbf{G}_k^{-2}\mathbf{T}\mathbf{A}^T)^{-1}(\mathbf{b} - \lambda\mu^k\mathbf{A}\mathbf{G}_k^{-1}\mathbf{t}) + \lambda\mu^k\mathbf{A}\mathbf{G}_k^{-1}\mathbf{t} = \mathbf{b}. \end{aligned}$$

Следовательно, $\mathbf{A}\mathbf{x}^{k+1} = \mathbf{b}$.

Убедимся в выполнении ограничений задачи (42) при $\lambda = 1$:

$$\Phi_2(\mathbf{x}^k(1), \mathbf{u}^k, \mu^k) \stackrel{(39)}{\leq} \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}.$$

Таким образом, решение задачи (42) достигается при $\lambda \leq 1$. Отсюда следует справедливость неравенства

$$\mu^{k+1} \leq \mu^k.$$

Решение задачи (42) достигается при выполнении ее ограничения в виде строгого равенства:

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^k, \mu^{k+1}) = \theta \mu^{k+1} t_{\min}. \quad (44)$$

Поскольку по условию $\mathbf{g}(\mathbf{u}^k) > \mathbf{0}$, то из (44) следует, что $\mathbf{x}^{k+1} > \mathbf{0}$.

Дальнейшие шаги доказательства связаны со следующим равенством:

$$\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) = \frac{1}{\mu^{k+1} t_j} \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^k) \right)^2. \quad (45)$$

Его доказательство осуществляется на основе равенства (40), которое при подстановке $\lambda = \lambda^k$ примет вид

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

Умножив обе его части слева на $\mathbf{T}^{-1} \mathbf{G}_k^2$, получим

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

Отсюда

$$\left(\mathbf{A}^T \mathbf{r}^{k+1} \right)_j = \frac{\left(g_j(\mathbf{u}^k) \right)^2 x_j^{k+1}}{t_j} - \mu^{k+1} g_j(\mathbf{u}^k). \quad (46)$$

Далее

$$\begin{aligned} \mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) &= \mu^{k+1} t_j - x_j^{k+1} \left(\mathbf{c} - \mathbf{A}^T \mathbf{u}^{k+1} \right)_j \stackrel{(43)}{=} \\ &= \mu^{k+1} t_j - x_j^{k+1} \left(\mathbf{c} - \mathbf{A}^T \mathbf{u}^k \right)_j + \frac{1}{\mu^{k+1}} x_j^{k+1} \left(\mathbf{A}^T \mathbf{r}^{k+1} \right)_j \stackrel{(46)}{=} \\ &= \mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^k) + \frac{1}{\mu^{k+1}} x_j^{k+1} \left(\frac{\left(g_j(\mathbf{u}^k) \right)^2 x_j^{k+1}}{t_j} - \mu^{k+1} g_j(\mathbf{u}^k) \right) = \\ &= \frac{1}{\mu^{k+1} t_j} \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^k) \right)^2. \end{aligned}$$

Равенство (45) доказано.

Из (44) и (45) следует, что для алгоритма D

$$\sum_{j=1}^n \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right) = \theta \mu^{k+1} t_{\min}. \quad (47)$$

Теперь можем показать справедливость неравенства

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min}.$$

$$\begin{aligned}
\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) &= \sum_{j=1}^n \frac{1}{\mu^{k+1} t_j} \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^{k+1}) \right)^2 \stackrel{(45)}{\leq} \\
&\leq \frac{1}{\mu^{k+1} t_{\min}} \sum_{j=1}^n \frac{1}{(\mu^{k+1} t_j)^2} \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^k) \right)^4 \leq \\
&\leq \frac{1}{\mu^{k+1} t_{\min}} \left(\sum_{j=1}^n \frac{1}{\mu^{k+1} t_j} \left(\mu^{k+1} t_j - x_j^{k+1} g_j(\mathbf{u}^k) \right)^2 \right)^2 \stackrel{(44)}{=} \theta^2 \mu^{k+1} t_{\min} < \theta \mu^{k+1} t_{\min}.
\end{aligned}$$

Отсюда, учитывая $\mathbf{x}^{k+1} > \mathbf{0}$, получаем $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$.

Таким образом, остается доказать только неравенство (32). Пусть $z_j^{k+1} = x_j^{k+1} g_j(\mathbf{u}^{k+1})$. Найдем, при каких α выполняется неравенство

$$\sum_{j=1}^n \frac{1}{\mu^{k+1} t_j (1-\alpha)} \left(\mu^{k+1} t_j - z_j^{k+1} - \alpha \mu^{k+1} t_j \right)^2 \leq \theta (1-\alpha) \mu^{k+1} t_{\min}. \quad (48)$$

Раскроем скобки, выполним ряд преобразований, получим:

$$\begin{aligned}
&\left(\sum_{j=1}^n \frac{1}{\mu^{k+1} t_j} \left(\mu^{k+1} t_j - z_j^{k+1} \right)^2 - 2\alpha \sum_{j=1}^n \left(\mu^{k+1} t_j - z_j^{k+1} \right) + \alpha^2 \sum_{j=1}^n \mu^{k+1} t_j \right) / (1-\alpha) \stackrel{(47)}{\leq} \\
&\leq \left(\theta^2 \mu^{k+1} t_{\min} - 2\alpha \theta \mu^{k+1} t_{\min} + \alpha^2 n \gamma \mu^{k+1} t_{\min} \right) / (1-\alpha).
\end{aligned}$$

Решаем относительно α неравенство

$$\left(\theta^2 \mu^{k+1} t_{\min} - 2\alpha \theta \mu^{k+1} t_{\min} + \alpha^2 n \gamma \mu^{k+1} t_{\min} \right) / (1-\alpha) \leq \theta (1-\alpha) \mu^{k+1} t_{\min}.$$

Умножим обе части неравенства на $\frac{1-\alpha}{\mu^{k+1} t_{\min}}$:

$$\theta^2 - 2\alpha \theta + \alpha^2 n \gamma \leq \theta (1 - 2\alpha + \alpha^2),$$

$$\alpha^2 (n \gamma - \theta) \leq \theta - \theta^2.$$

Отсюда получим, что неравенство справедливо при $\alpha \leq \sqrt{\theta} \sqrt{1-\theta} / \sqrt{n \gamma - \theta}$.

Следовательно, при $\alpha = \sqrt{\theta} \sqrt{1-\theta} / \sqrt{n \gamma - \theta}$ выполняется и условие (48),

то есть

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, (1 - \sqrt{\theta} \sqrt{1-\theta} / \sqrt{n \gamma - \theta}) \mu^{k+1}) \leq \theta (1 - \sqrt{\theta} \sqrt{1-\theta} / \sqrt{n \gamma - \theta}) \mu^{k+1} t_{\min}.$$

Значит, начиная со второй итерации, выполняются все условия теоремы. Теорема доказана.

Алгоритм Е

На каждой итерации возможно объединение шагов прямого и двойственного алгоритмов. То есть сначала на основе формул (29), (30) прямого алгоритма пересчитываются значения переменных двойственной задачи и параметра скошенного пути, а затем на основе формул (39), (42) двойственного алгоритма пересчитываются значения переменных прямой задачи и, заново, параметра скошенного пути. Такой алгоритм назовем прямо-двойственным и обозначим буквой Е.

Алгоритмы С₄ и D₄

Дальнейшее развитие алгоритмов связано с использованием более высоких степеней при решении вспомогательных задач (30) и (42). Можно надеяться на более быструю сходимость алгоритмов С₄ и D₄, поскольку применение более высоких степеней позволяет сильнее уменьшать значение параметра центрального пути μ на каждой итерации. То есть, стартуя с одного и того же приближения $(\mathbf{x}^k, \mathbf{u}^k)$ и одного и того же значения μ^k , разные алгоритмы приводят в разные точки, в которых значения параметра центрального пути распределяются согласно неравенствам

$$\mu_A^{k+1} > \mu_C^{k+1} \geq \mu_{C_4}^{k+1}, \quad \mu_A^{k+1} > \mu_D^{k+1} \geq \mu_{D_4}^{k+1}. \quad (49)$$

Здесь индексами А, С, С₄, D, D₄ обозначены значения μ^{k+1} , вычисляемые алгоритмами, соответствующими этим индексам.

Конечно, скорости сходимости алгоритмов не обязательно должны соответствовать приведенным неравенствам (49). Они справедливы 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. Одной из целей экспериментального исследования будет проверка того, имеет ли место ускорение на тестовых задачах.

Введем функцию, являющуюся аналогом используемой в алгоритмах A-D функции $\Phi_2(\mathbf{x}, \mathbf{u}, \mu)$:

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

Отличие алгоритма C_4 от алгоритма C заключается в том, что величину λ^k , показывающую степень уменьшения параметра μ находим не по правилу (30), а как решение следующей задачи:

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_4(\mathbf{x}^k, \mathbf{u}^k(\lambda), \lambda \mu^k) \leq \theta^2 \lambda^2 (\mu^k)^2 (t_{\min})^2. \quad (50)$$

Аналогично, в алгоритме D_4 вместо задачи (42) решаем задачу

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_4(\mathbf{x}^k(\lambda), \mathbf{u}^k, \lambda \mu^k) \leq \theta^2 \lambda^2 (\mu^k)^2 (t_{\min})^2. \quad (51)$$

В остальном вычислительный процесс обоих алгоритмов остается прежним.

Теорема 6

Для алгоритма (29), (50), (31) из выполнения условий

$$\mathbf{x}^k \in ri \mathbf{X}, \quad \mathbf{u}^k \in ri \mathbf{U}, \quad \Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k) \leq \theta \mu^k t_{\min}$$

следует справедливость соотношений

$$\mathbf{x}^{k+1} \in ri \mathbf{X}, \quad \mathbf{u}^{k+1} \in ri \mathbf{U}, \quad \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min}.$$

При этом $\mu^{k+1} \leq \mu^k$, а при $k > 1$ выполняется неравенство (32).

Доказательство

Доказательство выполнения балансовых ограничений-равенств

$\mathbf{Ax}^{k+1} = \mathbf{b}$ полностью совпадает с доказательством для алгоритма C .

Решение задачи (50) достигается при выполнении ее ограничения как строгого равенства, следовательно

$$\Phi_4(\mathbf{x}^k, \mathbf{u}^{k+1}, \mu^{k+1}) = \theta^2 (\mu^{k+1})^2 (t_{\min})^2. \quad (52)$$

Убедимся в выполнении ограничений задачи (50) при $\lambda = 1$. Основой здесь является очевидное соотношение, справедливое для любых векторов $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{u} \in \mathbf{R}^m$ и скаляра μ

$$\Phi_4(\mathbf{x}, \mathbf{u}, \mu) \leq (\Phi_2(\mathbf{x}, \mathbf{u}, \mu))^2.$$

Действительно, сумма квадратов неотрицательных величин не превосходит квадрат их суммы:

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

Отсюда получим:

$$\Phi_4(\mathbf{x}^k, \mathbf{u}^k(1), \mu^k) \leq (\Phi_2(\mathbf{x}^k, \mathbf{u}^k(1), \mu^k))^2 \stackrel{(29)}{\leq} (\Phi_2(\mathbf{x}^k, \mathbf{u}^k, \mu^k))^2 \leq \theta^2 (\mu^k)^2 (t_{\min})^2.$$

Таким образом, решение задачи (50) достигается при $\lambda \leq 1$. Следовательно, выполняется неравенство

$$\mu^{k+1} \leq \mu^k.$$

Поскольку $\theta \in (0;1)$, из равенства (52) следует, что $x_j^k g_j(\mathbf{u}^{k+1}) > 0$ для любого $j = 1, \dots, n$. По условию алгоритма $\mathbf{x}^k > \mathbf{0}$, отсюда справедливым является неравенство $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$.

Ключевым на дальнейших шагах доказательства является следующее соотношение, выводимое из (31):

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

$$\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) = \frac{1}{\mu^{k+1}t_j} \left(\mu^{k+1}t_j - x_j^k g_j(\mathbf{u}^{k+1}) \right)^2, \quad (53)$$

откуда следует

$$\left(\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) \right)^2 = \frac{1}{\left(\mu^{k+1} \right)^2 \left(t_j \right)^2} \left(\mu^{k+1}t_j - x_j^k g_j(\mathbf{u}^{k+1}) \right)^4. \quad (54)$$

Из (54) на основе равенства (52) выводится

$$\sum_{j=1}^n \left(\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) \right)^2 = \theta^2 \left(\mu^{k+1} \right)^2 \left(t_{\min} \right)^2, \quad (55)$$

что влечет

$$\begin{aligned} \Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) &\stackrel{def}{\leq} \frac{1}{\mu^{k+1}t_{\min}} \sum_{j=1}^n \left(\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) \right)^2 \stackrel{(55)}{=} \\ &= \theta^2 \mu^{k+1}t_{\min} < \theta \mu^{k+1}t_{\min}. \end{aligned}$$

Отсюда, учитывая $\mathbf{g}(\mathbf{u}^{k+1}) > \mathbf{0}$, получаем $\mathbf{x}^{k+1} > \mathbf{0}$.

На заключительном этапе, как обычно, нужно доказать неравенство (32). При доказательстве будем использовать выводимое из (55) свойство

$$\sum_{j=1}^n \left(\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) \right) \geq \theta \mu^{k+1}t_{\min}. \quad (56)$$

Действительно, $\sum a_j \geq \sqrt{\sum a_j^2}$ при $a_j \geq 0$, а, учитывая соотношение (53),

$$\mu^{k+1}t_j - x_j^{k+1}g_j(\mathbf{u}^{k+1}) \geq 0 \text{ для любого } j = 1, \dots, n.$$

Пусть $z_j^{k+1} = x_j^{k+1}g_j(\mathbf{u}^{k+1})$. Найдем, при каких α выполняется неравенство

$$\sum_{j=1}^n \frac{1}{\mu^{k+1}t_j(1-\alpha)} \left(\mu^{k+1}t_j - z_j^{k+1} - \alpha \mu^{k+1}t_j \right)^2 \leq \theta(1-\alpha)\mu^{k+1}t_{\min}.$$

Раскроем скобки, выполним ряд преобразований, получим:

$$\begin{aligned} &\left(\sum_{j=1}^n \frac{1}{\mu^{k+1}t_j} \left(\mu^{k+1}t_j - z_j^{k+1} \right)^2 - 2\alpha \sum_{j=1}^n \left(\mu^{k+1}t_j - z_j^{k+1} \right) + \alpha^2 \sum_{j=1}^n \mu^{k+1}t_j \right) / (1-\alpha) \stackrel{(56)}{\leq} \\ &\leq \left(\theta^2 \mu^{k+1}t_{\min} - 2\alpha\theta \mu^{k+1}t_{\min} + \alpha^2 n \mu^{k+1}t_{\min} \right) / (1-\alpha). \end{aligned}$$

Получили неравенство, точно совпадающее с неравенством из алгоритма С. Завершение доказательства идентично описанному для алгоритма С. Теорема доказана.

Обоснование алгоритма D_4 осуществляется аналогично.

Использование более высоких степеней при решении вспомогательных задач

Поскольку эксперименты показали, что применение четвертой степени при решении вспомогательных задач (30) и (42), 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

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

В прямом алгоритме C_p вместо задачи (30) будем решать следующую:

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_p(\mathbf{x}^k, \mathbf{u}^k(\lambda), \lambda \mu^k) \leq \theta^{p/2} \lambda^{p/2} (\mu^k)^{p/2} (t_{\min})^{p/2}. \quad (57)$$

Вычислительный процесс двойственного алгоритма D_p отличается от вычислительного процесса алгоритма D тем, что задача (42) заменяется на задачу

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \Phi_p(\mathbf{x}^k(\lambda), \mathbf{u}^k, \lambda \mu^k) \leq \theta^{p/2} \lambda^{p/2} (\mu^k)^{p/2} (t_{\min})^{p/2}. \quad (58)$$

Отдельный интерес здесь представляет использование p , стремящегося к бесконечности. В этом случае задача (57) запишется следующим образом:

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \max_{j=1, \dots, n} |\lambda \mu^k t_j - x_j^k g_j(\mathbf{u}^k(\lambda))| \leq \sqrt{\theta} \lambda \mu^k t_{\min}, \quad (59)$$

а задача (58) - в виде

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \max_{j=1, \dots, n} |\lambda \mu^k t_j - x_j^k(\lambda) g_j(\mathbf{u}^k)| \leq \sqrt{\theta} \lambda \mu^k t_{\min}. \quad (60)$$

В связи с возможностью существенных расхождений точек минимума функции $\Phi_p(\mathbf{x}, \mathbf{u}, \mu)$, используемой при решении вспомогательных задач (57) и (58), и функции $\Phi_2(\mathbf{x}, \mathbf{u}, \mu)$, используемой при нахождении параметрических вектор-функций $\mathbf{u}^k(\lambda)$ и $\mathbf{x}^k(\lambda)$ для алгоритмов с $p > 4$ не удается гарантировать принадлежность новой точки конусу скошенного пути, а именно, справедливость неравенства

$$\Phi_2(\mathbf{x}^{k+1}, \mathbf{u}^{k+1}, \mu^{k+1}) \leq \theta \mu^{k+1} t_{\min},$$

а также уменьшение параметра скошенного пути в соответствии с (32).

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

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

или для найденного λ^k выполняется неравенство

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

то заново осуществляется поиск λ^k на основе $p = 4$ и выполняется итеративный переход. Поскольку нет необходимости заново вычислять параметрическую функцию $\mathbf{u}^k(\lambda)$ или $\mathbf{x}^k(\lambda)$, 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

$$\mathbf{A}\mathbf{x}^k = \mathbf{b}, \mathbf{x}^k > \mathbf{0}, \mathbf{g}(\mathbf{u}^k) > \mathbf{0}, \max_{j=1, \dots, n} \left| \mu^k t_j - x_j^k g_j(\mathbf{u}^k) \right| \leq \sqrt{\theta} \mu^k t_{\min}. \quad (61)$$

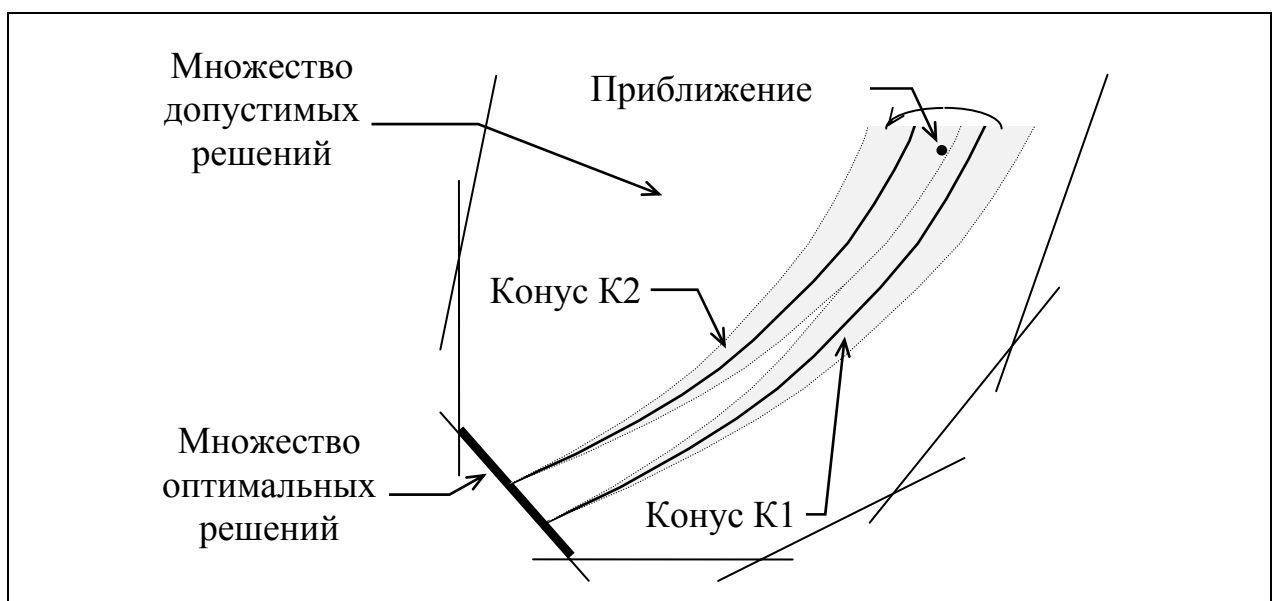
Возврат осуществляется только в случае, если какое-то из условий (61) нарушается. Однако подобные ситуации, если и возможны, то крайне редки, и в процессе экспериментального исследования не возникали ни разу.

Для подобной модификации алгоритмов (29),(59),(31), обозначенного C_p , и (39)-(41), (60), (43), обозначенного D_p , нельзя гарантировать уменьшения параметра μ в соответствии с неравенством (32). Однако на практике алгоритмы C_p и D_p при $p > 4$ могут успешно конкурировать с вариантами, предложенными ранее, и эффективно решать практические задачи, демонстрируя при этом существенно более высокую скорость сходимости.

Процедура уменьшения коэффициента скошенности

Меньшие значения коэффициента скошенности пути способствуют ускорению сходимости вычислительного процесса. Наиболее предпочтительным является движение в конусе пути аналитических центров. В то же время значения компонент стартового приближения \mathbf{x}^1 и $\mathbf{g}(\mathbf{u}^1)$ могут существенно (даже на порядки) отличаться друг от друга, из чего следует существенный разброс значений компонент иницирующего вектора $\mathbf{t} = \mathbf{x}^1 \otimes \mathbf{g}(\mathbf{u}^1)$.

В связи с этим очень полезной в вычислительном плане является процедура перехода по итерациям из конуса одного скошенного пути в конус другого, обладающего меньшим коэффициентом скошенности. Схематично такой переход можно изобразить на рисунке:



Выполнив итеративный переход в конусе K1 и получив новую пару векторов $(\mathbf{x}^k, \mathbf{u}^k)$, принадлежащую одновременно конусу K2 скошенного пути, обладающего меньшим коэффициентом скошенности, следующую итерацию осуществляем в конусе K2.

Наиболее простая процедура перехода в конус скошенного пути, обладающего меньшим коэффициентом скошенности, состоит в том, что будем уменьшать те компоненты иницирующего вектора, которые превышают значения $x_j^k g_j(\mathbf{u}^k)$, одновременно увеличивая минимальные его компоненты на максимально возможную величину при условии того, что пара векторов $(\mathbf{x}^k, \mathbf{u}^k)$ по-прежнему должна принадлежать конусу скошенного пути. То есть будем искать новый иницирующий вектор $\tilde{\mathbf{t}}$ путем решения следующей задачи:

$$\Delta \rightarrow \max, \quad \sum_{j=1}^n \frac{1}{\tilde{t}_j(\Delta)} \left(\tilde{t}_j(\Delta) - x_j^k g_j(\mathbf{u}^k) \right)^2 \leq \theta (\mu^k t_{\min} + \Delta), \quad (62)$$

где

$$\tilde{t}_j(\Delta) = \max \left\{ \mu^k t_{\min} + \Delta, \min \left\{ \mu^k t_j, x_j^k g_j(\mathbf{u}^k) \right\} \right\}. \quad (63)$$

Новое значение параметра μ равняется единице. Также справедливо

$$\tilde{t}_{\min} = \mu^k t_{\min} + \Delta.$$

Задача (62) может решаться приближенно. Поскольку объем требуемых вычислений не очень велик, то можно без особых затрат осуществлять подобную процедуру на каждой итерации. В то же время несмотря на ее простоту, процедура является высокоэффективной, что показало экспериментальное исследование, к рассмотрению результатов которого мы и переходим.

2.3. Экспериментальное исследование алгоритмов

Исследование будет состоять из нескольких экспериментов. Во-первых, проверим практическую эффективность процедур, ускоряющих сходимость алгоритмов, в первую очередь, процедуры параметризации. Здесь же определим, насколько полезным на практике является расширение конуса путем увеличения его радиуса, а также путем использования более высоких степеней p в решении вспомогательных задач. Второй эксперимент призван показать зависимость объема вычислений от размерности задачи. В целях чистоты эксперимента и нивелирования влияния побочных факторов первые два эксперимента будут осуществляться на алгоритмах центрального пути - частном случае предложенных алгоритмов. В то же время выявленные тенденции можно распространить и на весь класс алгоритмов скошенного пути. Третий эксперимент посвящен практической проверке непосредственно алгоритмов скошенного пути.

Тестовые задачи

Приведем тестовые задачи, на которых будет осуществляться экспериментальное исследование алгоритмов. Задачи 1-3 представлены матрицей \mathbf{A} и векторами \mathbf{b} и \mathbf{c} , а задачи 4-5 записаны в виде (1.2).

Задача 1 (1×2)

$$\mathbf{A} = (1 \ 1), \quad \mathbf{b} = (1), \quad \mathbf{c}^T = (1 \ 2).$$

Задача 2 (2×4)

$$\mathbf{A} = \begin{pmatrix} 5 & 3 & 1 & 0 \\ 3 & 2 & 0 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 480 \\ 300 \end{pmatrix}, \quad \mathbf{c}^T = (-1.2 \ -1 \ 0 \ 0).$$

Задача 3 (7×16)

$$\mathbf{A} = \begin{pmatrix} 15 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 25 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 25 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 & 0 & 17 & 0 & 0 & 0 & 45 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 300 \\ 200 \\ 1000 \\ 500 \\ 50 \\ 20 \\ 30 \end{pmatrix},$$

$$\mathbf{c}^T = (15 \ 20 \ 25 \ 40 \ 70 \ 28 \ 15 \ 45 \ 40 \ 70 \ 40 \ 65 \ 0 \ 0 \ 0 \ 0).$$

Задача 4 ($m \times 2m$)

$$u_m \rightarrow \max,$$

$$\begin{cases} u_1 \in [-1; 1], \\ u_i \in [-1 - u_{i-1}; 1 + u_{i-1}], \quad i = 2, \dots, m. \end{cases}$$

Решение: $\bar{u}_i = i$, $i = 1, \dots, m$. Задача 4 решалась при $m = 18$ и $m = 99$.

Задача 5 (Задача Данцига) ($m \times 2m$)

$$u_m \rightarrow \max,$$

$$\begin{cases} u_1 \in [-1; 1], \\ u_i \in [-2^{2i-2} - 2u_{i-1}; 2^{2i-2} + 2u_{i-1}], \quad i = 2, \dots, m. \end{cases}$$

Решение: $\bar{u}_1 = 1$, $\bar{u}_i = 2\bar{u}_{i-1} + 2^{2i-2}$, $i = 2, \dots, m$. $\bar{\mathbf{u}} = (1, 6, 28, 120, 496, \dots)$.

Отметим, что применение симплекс-метода для решения задачи Данцига [27] может привести к перебору всех 2^m вершин многогранника допустимых решений.

Случайно сгенерированные задачи ($m \times n$)

Часть задач была сгенерирована случайным образом с помощью датчика случайных чисел. Во всех этих задачах матрица \mathbf{A} является сильно заполненной, число нулевых элементов в ней не превышает 10 процентов. В то же время при формировании стартовой точки достаточно велик оказывается разброс значений переменных. Несколько подробнее об этом скажем при описании третьего эксперимента.

Инициализация алгоритмов и завершение вычислительного процесса

Проблему инициализации алгоритмов центрального пути будем решать с помощью метода, изложенного Монтейро и Адлером в [60], суть которого заключается в решении расширенной задачи размерности $(m+1) \times (n+2)$ и двойственной к ней, для которых априори известной является точка центрального пути. Способ формирования пары расширенных задач был описан в первой главе диссертационной работы.

Для инициализации алгоритмов скошенного пути достаточно отыскать любые относительно внутренние точки множеств допустимых решений задач (1.1) и (1.2). Ввод в допустимую область будем осуществлять на основе аффинно-масштабирующих алгоритмов, что также описывалось в первой главе диссертационной работы. Вычислительный процесс будет начинаться с единичных векторов $\mathbf{x}^1 = \mathbf{e}$ и $\mathbf{g}^1 = \mathbf{e}$. Целевые функции задач (1.1)-(1.2) во время фазы ввода в допустимую область учитываться не будут.

Критерием останова послужит достижение невязкой двойственности $\sum x_j g_j(\mathbf{u})$ значения $5 \cdot 10^{-6}$ во всех задачах, кроме задачи Данцига размерности 19×38 , для которой условие завершения выглядит $\sum x_j g_j(\mathbf{u}) = 0.001$.

Перейдем непосредственно к экспериментам.

Эксперимент 1. Проверка эффективности процедур, ускоряющих сходимость алгоритмов

На тестовых задачах 1-5 сопоставим предложенные в предыдущем разделе прямые алгоритмы А, В, С, C_4 , C_{16} и C_∞ , а также прямо-двойственный алгоритм Е и его аналог E_4 .

Важным является практическое использование полиномиальных алгоритмов, в частности, предложенных выше. Поэтому интересно сравнить их скорость сходимости со скоростью сходимости вариантов аффинно-

масштабирующего метода, не являющегося полиномиальным, но хорошо зарекомендовавшего себя на практике и, как показали ряд исследований [37], [58], [59], сопоставимого по скорости с симплекс-методом. Сравнение будем проводить с двумя вариантами прямого аффинно-масштабирующего метода, отличающихся друг от друга способом вычисления весовых коэффициентов d_j^k . В первом из них, классическом варианте [10], обозначенном как алгоритм F₁, их расчет осуществляется по формуле

$$d_j^k = (x_j^k)^2, \quad j = 1, \dots, n,$$

а во втором (алгоритм F₂), предложенном В.И.Зоркальцевым в [17] в целях уменьшения влияния погрешностей вычислений на заключительном этапе вычислительного процесса, - по формуле

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

Значение параметра γ в приводимых расчетах примем равным 0.6.

В таблицу 1 занесем число итераций, необходимых для решения каждым алгоритмом каждой из приведенных тестовых задач. Все исследуемые алгоритмы, кроме прямо-двойственных, имеют приблизительно одинаковый объем вычислений на каждой итерации - наибольшую сложность представляет обращение некоторой симметричной положительно определенной матрицы размерности $m \times m$, что требует не более $m^3/2$ арифметических операций. Поэтому данные о числе итераций могут быть использованы в качестве сравнительной характеристики объема вычислений. На каждой итерации прямо-двойственных алгоритмов используется два обращения матрицы. Следовательно, итерацию этих алгоритмов можно приближенно считать в два раза более трудоемкой.

Итеративный процесс для всех алгоритмов начинался с одной точки, полученной в соответствии с процедурой инициализации. Чтобы сократить негативный эффект плохой обусловленности расширенных задач, принима-

лись меньшие значения величин λ , а именно 1, 256, 256, 1.2^{18} , 10^6 , 10^6 , 1.2^{100} соответственно для каждой из 7 задач.

Величина θ для алгоритма А принималась равной 0.5. Для остальных алгоритмов через знак дроби в таблицу запишем данные о числе итераций, соответствующие $\theta = 0.5$ и $\theta = 0.9$. Для задачи 4 размерности 100×200 значение θ равнялось 0.9 для всех алгоритмов, кроме А.

Размерности, записанные в таблице, соответствуют расширенным задачам.

Таблица 1. Число итераций, необходимое для решения задач различными методами

Алг.	Задача 1 2×4	Задача 2 3×6	Данцига 6×12	Задача 3 8×18	Данцига 19×38	Задача 4 19×38	Задача 4 100×200
А	90	218	278	258	730	805	2174
В	38/30	113/87	158/119	154/117	459/342	507/380	1093
С	28/20	50/37	85/63	82/59	260/189	240/170	501
Е	16/12	42/33	57/44	55/43	157/120	123/95	193
С ₄	25/17	45/29	62/45	50/42	119/101	110/79	142
Е ₄	14/11	34/27	39/30	34/26	82/64	79/61	95
С ₁₆	24/12	44/29	53/39	44/33	99/75	84/53	74
С _∞	23/12	43/26	52/39	41/31	93/73	78/49	63
F ₁	16	35	36	30	62	66	84
F ₂	16	37	40	33	69	68	100

Из таблицы видно, что, несмотря на отсутствие хорошей стартовой точки и необходимость решения расширенной плохо обусловленной задачи, лучшие из предложенных алгоритмов даже в таком виде становятся сопоставимыми по скорости сходимости с методами, успешно применяющимися на практике. В немалой степени повышает их перспективность тот факт, что они обладают верхней оценкой объема вычислений, гарантирующей получение оптимального решения за полиномиальное время. При этом алго-

ритмы на практике работают существенно лучше имеющейся гарантированной оценки, что показывает значительное (до десятков раз) превосходство наиболее эффективных из них над алгоритмом А, действующим по теоретической оценке.

Интересным результатом экспериментального исследования является то, что, хотя наилучшие оценки достигаются при значении $\theta=0.5$, поскольку при этом максимально значение величины $\theta(1-\theta)$, на практике наиболее эффективны алгоритмы со значением θ , близким к 0.9. Вообще, как показало исследование, перспективными являются более широкие аппроксимации конуса центрального пути. Причем, это может достигаться как увеличением его радиуса $\sqrt{\theta}$, так и использованием более высоких степеней в процессе решения вспомогательной задачи.

Алгоритмы, использующие степени $p>4$, как отмечалось в предыдущем разделе, не являются обоснованными. В то же время, на описанных здесь задачах (включая задачу Данцига), а также на ряде случайно сгенерированных задач, ситуация невыполнения на одной из итераций ослабленных условий (62) либо неравенства (32), не была выявлена, а алгоритмы продемонстрировали высокую скорость сходимости. Наиболее эффективным из обоснованных алгоритмов можно считать алгоритм С₄.

Оба варианта аффинно-масштабирующего метода можно считать близкими по скорости. Поэтому на задачах, где предполагается возникновение проблем, связанных с негативным влиянием погрешностей вычислений, рекомендуется использовать алгоритм F₂. Также представляется возможным, сделав некоторое количество итераций по правилам алгоритма F₁, сходящегося несколько быстрее, на заключительной стадии вычислительного процесса сменить правила вычисления весовых коэффициентов на новые, используя преимущества алгоритма F₂.

Эксперимент 2. Зависимость объема вычислений от размерности задачи

Как видно из таблицы 1, на задачах большей размерности эффект использования процедур, ускоряющих сходимость, увеличивается. В связи с этим интерес может представлять исследование зависимости объема вычислений от размерности задачи.

В целях исключения остальных факторов будем сопоставлять число итераций, требующихся для получения решения одной и той же задачи, а именно задачи 4 различной размерности. Значения величины λ , используемой при формировании расширенной задачи, будем выбирать равным 1.2^n , $\theta = 0.9$. Приведем в таблице результаты по наиболее эффективному из обоснованных алгоритмов C_4 и алгоритму, использующему чебышевскую норму, C_∞ . Размерности, записанные в таблице, соответствуют расширенным задачам.

Таблица 2. Число итераций, необходимое для решения задачи 4 различной размерности

Алг.	20 × 40	30 × 60	40 × 80	50 × 100	70 × 140	100 × 200	150 × 300
C_4	49	60	71	83	107	142	196
C_∞	36	39	43	48	53	63	80

Дальнейшее увеличение размерности задач ведет к еще большим значениям величины λ и сложностям при численной реализации алгоритмов. Здесь могут оказаться очень полезными предложенные алгоритмы оптимизации в конусе скошенного пути. Несмотря на то, что значение коэффициента скошенности пути, в конусе которого начинается вычислительный процесс, может быть очень большим, благодаря процедуре (62)-(63), его значение по итерациям быстро уменьшается. При этом доминирующим фактором становится то, что решаются задачи с исходной, а не расширенной, плохо обусловленной матрицей A .

Эксперимент 3. Практическая проверка алгоритмов скошенного пути

Как уже говорилось, вычислительный процесс алгоритмов скошенного пути, состоит из двух фаз: ввода в область допустимых решений и непосредственно оптимизации. При этом на всех рассмотренных задачах получение допустимого стартового приближения, как по переменным прямой, так и по переменным двойственной задачи, требовало не более 5 итераций. В таблицу внесем суммарное число итераций, необходимое для получения оптимального решения, начиная с точки $\mathbf{x}^1 = \mathbf{e}$, $\mathbf{g}^1 = \mathbf{e}$.

Сопоставление будем проводить на вариантах алгоритма C_4 , наиболее эффективного из обоснованных. В первом из них, C_4 , формируется расширенная задача и осуществляется процесс оптимизации в конусе центрального пути, а во втором, C_4^C - ищется любая допустимая точка, и начинается процесс оптимизации в конусе скошенного пути.

Для тестирования сгенерируем наборы случайных задач различной размерности (по 5 каждой). В таблицу внесем среднее количество итераций, требующееся методам для их решения. Также для сведения запишем среднее значение коэффициента скошенности в стартовой точке, а также среднее его значение после 40 итераций. Параметры алгоритмов оставим прежними: $\theta = 0.9$, $\lambda = 1.2^n$.

Таблица 3. Среднее число итераций, необходимое для решения случайно сгенерированных задач

Алгоритмы	20×40	50×100	100×200	300×1000
C_4 (центральный)	63.6	112.8		
C_4^C (скошен.)	64.6	84.0	98.8	194.0
Начальн. знач. γ	688.59136	4063.85557	7256.78106	14601.30272
Знач. γ на 40-итер.	1.04021	2.00803	2.97036	21.91051

Несмотря на то, что задачи были сгенерированы специальным образом так, чтобы в стартовой точке наблюдался существенный разброс значений переменных (а это должно было оказать негативное влияние), эксперимент показал высокую вычислительную эффективность алгоритма скошенного пути. В первую очередь, этот результат достигнут благодаря процедуре сокращения коэффициента скошенности. Таблица наглядно показывает, что уже после 40 итераций значение γ невелико. В дальнейшем траектория алгоритма продолжает приближаться к центральному пути.

Алгоритм может решать задачи большой размерности, не испытывая при этом затруднений, связанных с плохой обусловленностью обрабатываемых матриц. В частности, видим, что алгоритм успешно решил случайно сгенерированные задачи размерности 300×1000 , а при такой размерности даже заниженные значения величины $\lambda = 1.2^n$ не позволяют решить задачу алгоритмом центрального пути с использованием стартового приближения, сформированного на основе процедуры, изложенной у Монтейро и Адлера.

Из результатов эксперимента выделим также, что алгоритм скошенного пути нашел оптимальное решение задачи 4 размерности 100×200 за 67 итераций, а размерности 400×800 - за 95 итераций.

Подведем итоги. Последовательно вводимые усовершенствования (от исходного алгоритма А до наиболее эффективных вариантов, использующих параметризацию, а также более высокие степени p при решении вспомогательных задач), хотя и не изменили верхние оценки объема необходимых для решения задач вычислений, привели к значительному сокращению времени расчетов на практике. Можно считать, что полиномиальные алгоритмы приблизились по скорости к наилучшим из применяющихся на практике алгоритмам.

Вместе с тем наибольшей сложностью для алгоритмов центрального пути остается проблема поиска стартового приближения. Одним из способов решения этой проблемы является использование алгоритмов оптимизации в конусе скошенного пути, вычислительный процесс в которых может начинаться с любых относительно внутренних точек множеств допустимых решений. Благодаря эффективной процедуре уменьшения коэффициента скошенности, алгоритмы скошенного пути демонстрируют высокую скорость в решении практических задач, в том числе, большой размерности.

При этом остается проблема ввода в допустимую область. Для инициализации алгоритмов скошенного пути необходимо получить два вектора: $\mathbf{x} \in ri\mathbf{X}$ и $\mathbf{u} \in ri\mathbf{U}$, то есть нужно реализовать две процедуры ввода в допустимую область - для прямой задачи (1.1) и для двойственной задачи (1.2). По этому признаку алгоритмы скошенного пути уступают как прямым, так и двойственным аффинно-масштабирующим алгоритмам, для которых достаточно получить начальное допустимое приближение только одной из задач.

В то же время есть задачи, где начальное приближение может быть априори известным. Более того, для некоторых задач может иметься стартовая точка, принадлежащая центральному пути. Перейдем к рассмотрению систем уравнений и неравенств с двухсторонними ограничениями на переменные.