УДК 519.6, 519.85

ПРЯМОДВОЙСТВЕННЫЕ МЕТОДЫ ВНУТРЕННЕЙ ТОЧКИ В ЗАДАЧАХ ЛИНЕЙНОГО, КВАДРАТИЧНОГО И ПОЛУОПРЕДЕЛЁННОГО ПРОГРАММИРОВАНИЯ

Кулагин Никита Андреевич[1], Булякова Ирина Александровна[2]

[1]Студент;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: n.kulagin.it@gmail.com.

[2]Старший преподаватель;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: buljakova@mail.ru.

В работе рассматриваются три класса выпуклых задач оптимизации – линейное, квадратичное и полуопределённое программирование, затем выводятся и анализируются три варианта метода внутренней точки типа «предиктор-корректор» для решения данных задач. В заключении проводятся численные эксперименты, подтверждающие теоретические выкладки.

Ключевые слова: методы внутренней точки, методы оптимизации, выпуклая оптимизация, метод Ньютона, условия Каруша-Куна-Таккера.

Для цитирования:
Кулагин Н. А., Булякова И. А. Прямодвойственные методы внутренней точки в задачах линейного, квадратичного и полуопределённого программирования // Системный анализ в науке и образовании: сетевое научное издание. 2023. № 4. С. 13-34. EDN: ZOETZG. URL : https://sanse.ru/index.php/sanse/article/view/595.

PRIMAL-DUAL INTERIOR POINT METHODS IN LINEAR, QUADRATIC AND SEMIDEFINITE PROGRAMMING PROBLEMS

Kulagin Nikita A.[1], Bulyakova Irina A.[2]

[1]Student;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: n.kulagin.it@gmail.com.

[2]Senior Lecturer at the Department of System Analysis and Management;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: buljakova@mail.ru.

In this paper we consider three classes of convex optimization problems – linear, quadratic and semidefinite programming problems, then we derive and analyze a «predictor-corrector» interior point algorithm for each class of problems. Finally, we conduct numerical experiments to verify theoretical results.

Keywords: Primal-Dual Interior point methods, IPM, numerical optimization, convex optimization, dual problem, Newton's method, Karush-Kuhn-Tucker conditions.

For citation:
Kulagin N. A., Bulyakova I. A. Primal-Dual Interior point methods in linear, quadratic and semidefinite programming problems. *System analysis in science and education*, 2023;(4):13-34 (in Russ). EDN: ZOETZG. Available from: https://sanse.ru/index.php/sanse/article/view/595.

Введение

Современные прикладные задачи экономики, комбинаторики, теории вероятности, теории управления и задачи о робастном проектировании различного рода систем путём подбора их параметров зачастую могут быть сформулированы как задачи оптимизации.

В экономике наиболее распространены линейные и квадратичные задачи за счёт их простоты, так как более сложные экономические модели трудно применимы на практике и требует продвинутых знаний математики. Задачи комбинаторики, теории управления и теории вероятностей зачастую сводятся к задачам полуопределённого программирования, так как требуется найти некоторую матрицу, удовлетворяющую определённым ограничениями – матрица ковариации, матрица инцидентности, матрицы, определяющие устойчивость динамических систем.

На данный момент для решения задач линейного и квадратичного программирования разработаны различные методы. Задачи линейного программирования изначально решались симплекс-методом. Для задач квадратичного программирования было предложено множество различных методов их решения. Общим является то, что эти классы задач покрываются одним методом – методом внутренней точки, применимым также и к задачам полуопределённого программирования, оказавшимся на порядок сложнее и для которых до сих пор не разработано эффективных алгоритмов, способных их решать, кроме методов внутренней точки.

Определение и свойства задач оптимизации

В начале следует дать определение каждой задаче оптимизации, которые будем рассматривать. Один из способов записать задачу оптимизации – указать целевую функцию и указать функциональные ограничения, которые должны быть соблюдены аргументом функции.

$$\begin{gathered}\min_x f_0(x) \\ f_i(x) \le 0, i = 1, \dots, m \\ h_j(x) = 0, j = 1, \dots, p\end{gathered}$$

Если в данной задаче функции $f_i(x), i = 1, \dots, m$ выпуклы и функции $h_j(x)$ аффинны, то множество векторов $x \in \mathbb{R}^n$, удовлетворяющих ограничениям, называется допустимым множеством является выпуклым. Если также функция $f_0(x)$ выпукла, то задача называется выпуклой. В этом случае любая точка локального минимума функции $f_0(x)$ на выпуклом допустимом множестве является точкой глобального минимума на этом множестве.

Для условных задач, в которых ограничения имеют вид равенств, но не неравенств, разработан метод множителей Лагранжа. Данный метод предлагает вместо исходной условной задачи решать безусловную задачу с большим количеством переменных. Функция Лагранжа определяется как $\mathcal{L}(x, \lambda) = f_0(x) + \sum_{j=1}^{p} \lambda_j h_j(x)$. Метод множителей Лагранжа ищет минимум этой функции, необходимым условием экстремума которого является равенство нулю производных.

$$\mathcal{L}'_x(x, \lambda) = \nabla f_0(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) = 0$$
$$\mathcal{L}'_\lambda(x, \lambda) = h_j(x) = 0, j = 1, \dots, p$$

Во многом по аналогии с методом множителей Лагранжа, для условных задач с ограничениями-неравенствами были разработаны т. н. условия Каруша-Куна-Таккера, являющиеся необходимыми (и достаточными, в случае выпуклой задачи) условиями экстремума. Аналогично составляется модифицированная функция Лагранжа, имеющая вид

$$\mathcal{L}(x,s,\lambda) = f_0(x) + \sum_{i=1}^{m} s_i f_i(x) + \sum_{j=1}^{p} \lambda_j h_j(x)$$

Переменные x называются «прямыми» или исходными, а переменные (s,λ) – двойственными или множителями Лагранжа. Говорят, что прямодвойственная пара векторов $x \in \mathbb{R}^n$ и $(s,\lambda) \in \mathbb{R}^m \times \mathbb{R}^p$ удовлетворяет условиями Каруша-Куна-Таккера, если для неё выполнены следующие условия

$$f_i(x) \le 0, i = 1, \dots, m$$
$$h_j(x) = 0, j = 1, \dots, p$$
$$s \ge 0$$
$$s_i f_i(x) = 0, i = 1, \dots, m$$
$$\nabla \mathcal{L}_x(x,s,\lambda) = \nabla f_0(x) + \sum_{i=1}^{m} s_i \nabla f_i(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) = 0$$

Последнее и предпоследнее условия называются условиями стационарности и дополняющей нежёсткости соответственно. По сути, это система нелинейных уравнений и неравенств. В силу их достаточности для выпуклых задач, можно предложить искать решение этой системы вместо решения исходной задачи.

Если из условия стационарности удаётся выразить переменную $x^\star(s,\lambda)$ как функцию от (s,λ) и являющуюся точкой минимума функции Лагранжа, то её можно подставить обратно в функцию Лагранжа и получить т. н. двойственную задачу

$$\max_{s,\lambda} g(s,\lambda) = \max_{s,\lambda} \inf_{x} \mathcal{L}(x,s,\lambda) = \max_{s,\lambda} \mathcal{L}(x^\star(s,\lambda),s,\lambda)$$
$$\lambda \ge 0$$

Из определения двойственной функции (целевой функции в двойственной задаче) следует базовое неравенство $g(s,\lambda) \le \mathcal{L}(x,s,\lambda) \le f_0(x)$, если переменные (x,s,λ) удовлетворяют ограничениям в прямой и двойственной задаче. Это поясняет, почему двойственную функцию нужно максимизировать. Двойственная задача всегда является вогнутой задачей (выпуклой после умножения целевой функции на -1) так как двойственная функция является инфимумом вогнутых (аффинных) по (s,λ) функций и следовательно сама является вогнутой.

Задача линейного программирования

Задача линейного программирования, как следует из названия, содержит линейные ограничения и линейную целевую функцию. В литературе принято записывать задачи в стандартной форме, для задачи линейного программирования стандартная форма выглядит следующим образом (остальные варианты записи могут быть приведены к данному)

$$\min_{x} c^\top x$$
$$Ax = b$$
$$x \ge 0$$

Задача задана относительно переменной $x \in \mathbb{R}^n$с точностью до матрицы $A \in \mathbb{R}^{m \times n}$и вектора правой части $b \in \mathbb{R}^m$системы линейных уравнений. В данном случае $f_i(x) = -x_i, i = 1, \dots, n$ выпуклы, ограничения-равенства $h_j(x) = a_i^\top x - b_i$ – аффинные. Значит это выпуклая задача.

Условия Каруша-Куна-Таккера для задачи линейного программирования в стандартной форме выглядят следующим образом

$$x \ge 0$$
$$Ax = b$$
$$s \ge 0$$
$$s_i x_i = 0, i = 1, \dots, n$$

$$A^\top \lambda + s = c$$

В данном случае видно, что система оказалось гораздо более простой, чем в общем случае. Это система линейных уравнений и линейных неравенств с единственным нелинейным условием $s_i x_i = 0$.

Задача квадратичного программирования

Задача квадратичного программирования включает в себя задачи линейного программирования как частный случай. Единственным принципиальным отличием данного класса задач от линейных является квадратичная целевая функция.

$$\begin{gathered} \min_x \frac{1}{2} x^\top Q x + c^\top x \\ Ax = b \\ Bx \leq d \end{gathered}$$

Задача квадратичного программирования также имеет линейные ограничения-равенства и линейные ограничения-неравенства (хоть и не тривиальные, как в линейном программировании, а заданные матрицей B и вектором правой части d). Единственным неявным ограничением, налагаемым на целевую функцию, является ограничение на симметричность и положительную полуопределённость матрицы коэффициентов Q (пишут $Q \succcurlyeq 0$). Если это так, то целевая функция (а значит и вся задача) является выпуклой, в противном случае – невыпуклой. Невыпуклые задачи квадратичного программирования также имеют большое количество приложений, однако является на порядок сложнее и требуют отдельного подхода к решению, поэтому их в работе не рассматриваем.

Условия Каруша-Куна-Таккера для этого класса задач имеют вид

$$\begin{gathered} Ax = b \\ Bx \leq d \\ s \geq 0 \\ s_i\left(B_i^\top x - d_i\right) = 0 \\ Qx + c + A^\top \lambda + B^\top s = 0 \end{gathered}$$

Задача полуопределённого программирования

Задача полуопределённого программирования, в отличии от предыдущих задач, задаётся относительно матрицы $X \in \mathbb{R}^{n \times n}$, на которую накладываются линейные ограничения-равенства, ограничения на симметричность и положительную полуопределённость (т. е. неотрицательность собственных чисел матрицы), а оптимизируется линейная функция. С помощью записи $\langle A, B \rangle = tr(A^\top B)$ будем обозначать стандартное скалярное произведение на пространстве матриц (след от произведения $A^\top B$). Тогда задача полуопределённого программирования примет вид

$$\begin{gathered} \min_X \ \langle C, X \rangle \\ \langle A_i, X \rangle = b_i, i = 1, \dots, m \\ X \succcurlyeq 0 \end{gathered}$$

Задача задана относительно матрицы X с точностью до симметричных матриц A_i, C и коэффициентов b_i. Несмотря на кажущуюся простоту в виде линейной целевой функции и линейных ограничений-равенств, этот класс задач включает в себя задачи линейного и квадратичного программирования как частный случай.

Условия Каруша-Куна-Таккера и функция Лагранжа для этой задачи формируются относительно скалярного произведения $\langle A, B \rangle$ на пространстве матриц для ограничения на положительную полуопределённость. Функция Лагранжа является аффинной по матрице X и имеет вид

$$\mathcal{L}(X, \lambda, S) = \langle C, X \rangle - \sum_{i=1}^{m} \lambda_i \langle A_i, X \rangle + \lambda^\top b - \langle S, X \rangle = \langle C - \sum_{i=1}^{m} \lambda_i A_i - S, X \rangle + \ \mu^\top b$$

Условия Каруша-Куна-Таккера являются линейными, как и в случае задачи линейного программирования, за исключением условия дополняющей нежёсткости, и примут вид

$$\begin{gathered}\langle A_i, X\rangle = b_i \\ X \succcurlyeq 0 \\ \sum_{i=1}^{m} \lambda_i A_i + S = C \\ S \succcurlyeq 0 \\ XS = 0\end{gathered}$$

В этом контексте «линейное» условие означает линейное уравнение либо линейное матричное неравенство вида $S \succcurlyeq 0$, несмотря на то, что последнее задаёт выпуклую область с нелинейной (полиномиальной) границей.

Метод внутренней точки в задаче линейного программирования

Существует три принципиальных вида методов внутренней точки – классические (прямые) методы, метод потенциалов и прямодвойственные методы. Классический вариант на данный момент устарел и проигрывает по скорости работы прямодвойственным методам, а метод потенциалов крайне сильно отличается от прямодвойственных методов и более похож на классический вариант, поэтому будем рассматривать прямодвойственные методы внутренней точки так как они могут быть получены из схожих соображений для всех трёх классов задач и являются эффективными.

Существует множество интерпретаций принципа работы методов внутренней точки, для прямодвойственных методов – это попытка решения условий Каруша-Куна-Таккера с помощью метода Ньютона. Введём вектор-функцию $F(x, \lambda, s)$

$$F(x, \lambda, s) = \begin{pmatrix} A^{\top}\lambda + s - c \\ Ax - b \\ XSe \end{pmatrix}$$

Здесь под XS подразумевается произведение диагональных матриц, составленных из компонент векторов x и s. Вектор $e = (1,1,\dots,1)$ – вектор из единиц. Из определения функции $F(x, \lambda, s)$ следует, что если мы нашли прямодвойственную пару (x, λ, s)такую, что она является корнем функции $F(x, \lambda, s)$и $x \geq 0, s \geq 0$, то мы нашли набор векторов, удовлетворяющих условиям Каруша-Куна-Таккера и следовательно решили задачу линейного программирования.

Прямодвойственные методы внутренней точки пытаются решать систему нелинейных уравнений $F(x, \lambda, s) = 0$, поддерживая на каждой итерации инвариант $x > 0, s > 0$. Поэтому они и называются методами внутренней точки, т. к. итерации всё время находятся строго внутри допустимого множества (если рассматривать линейные неравенства), а не на границе, как это делает симплекс-метод. Слово «прямодвойственной» отсылает к тому, что метод в том числе решает прямую и двойственную задачу одновременно (или изменяет прямые и двойственные переменные одновременно).

Для поиска корня функции $F(x, \lambda, s)$можно применить метод Ньютона, для этого необходимо выписать матрицу Якоби функции $F(x, \lambda, s)$ и решить линейное уравнение относительно приращений $(\Delta x, \Delta\lambda, \Delta s)$, после чего обновить переменные, перейдя на следующую итерацию $(x_{k+1}, \lambda_{k+1}, s_{k+1}) = (x_k, \lambda_k, s_k) + \alpha_k(\Delta x_k, \Delta\lambda_k, \Delta s_k)$ с некоторым шагом $\alpha_k \in (0; 1]$. Система линейных уравнений, решаемая в данном случае, будет иметь вид

$$J(x, \lambda, s)\begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -F(x, \lambda, s) \Leftrightarrow \begin{pmatrix} 0 & A^{\top} & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -\begin{pmatrix} A^{\top}\lambda + s - c \\ Ax - b \\ XSe \end{pmatrix}$$

Также для корректной работы алгоритма необходимо ввести критерий остановки. В данном случае удачным выбором будет критерий μ, оценивающий близость к точке, удовлетворяющей условиям дополняющей нежёсткости

$$\mu = \frac{x^\top s}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i s_i$$

На практике данная итеративная схема позволяет делать лишь маленькие шаги ($\alpha_k \ll 1$) перед тем, как нарушить ограничения $x_k > 0, s_k > 0$. Такой шаг является шагом метода Ньютона и называется шагом аффинного шкалирования (*affine scaling*) в контексте метода внутренней точки. Решением этой проблемы является подход, когда мы пытаемся не сразу достичь выполнения условия дополняющей нежёсткости $x_i s_i = 0$, а лишь уменьшить значение попарного произведения до значения $x_i s_i = \sigma\mu$, где $\sigma \in [0; 1]$ – параметр центрирования, при $\sigma < 1$ следующая точка будет иметь меньшее значение критерия μ. Линейные уравнения в данном случае будут иметь аналогичный вид, единственное изменение будет в последнем уравнении

$$\begin{pmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -\begin{pmatrix} A^\top\lambda + s - c \\ Ax - b \\ XSe - \sigma\mu e \end{pmatrix}$$

Центральным путём называется множество C, элементы которого удовлетворяют релаксированным условиями Каруша-Куна-Таккера с некоторым параметром $t > 0$

$$C = \{(x, \lambda, s): |A^\top\lambda + s = c, Ax = b, x_i s_i = t, x > 0, s > 0\}$$

Значение $\sigma = 1$ не позволяет совершить какой бы то ни было прогресс в решении, так как мы не приближаемся к точке, удовлетворяющей условию $x_i s_i = 0$, но шаг, полученный при таком σ, направлен в сторону центрального пути и делает возможным больший прогресс на будущих итерациях. С другой стороны, $\sigma = 0$ приводит к слишком агрессивным итерациям, быстро нарушающим условия $x > 0, s > 0$ и также не имеющим большого прогресса, если большинство компонент векторов x, s уже близки к нулю. Таким образом, параметр σ_k следует выбирать на каждой итерации в виде некоторого промежуточного значения. На рисунке 1 представлена проекция центрального пути в пространство переменной x, в виде многогранника представлены ограничения в задаче линейного программирования.

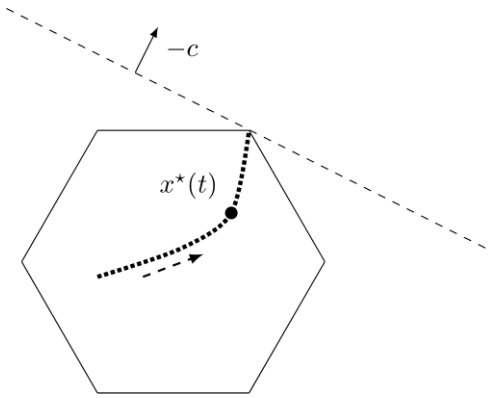

Рис. 1. Центральный путь $x^\star(t)$

На практике распространение получили алгоритмы типа «предиктор-корректор». Пусть $(\Delta x^{aff}, \Delta\lambda^{aff}, \Delta s^{aff})$ – обновления переменных при шаге аффинного шкалирования (т. е. решение указанной системы линейных уравнений при $\sigma = 0$). Тогда рассмотрим произведение переменных $x_i s_i$ на следующем шаге, причём шаг сделаем полным (т. е. $\alpha_k = 1$), получим

$$\left(x_i + \Delta x_i^{aff}\right) \cdot \left(s_i + \Delta s_i^{aff}\right) = x_i s_i + x_i \Delta s_i^{aff} + s_i \Delta x_i^{aff} + \Delta x_i^{aff} \Delta s_i^{aff} = \Delta x_i^{aff} \Delta s_i^{aff}$$

Равенство нулю остального выражения следует из последнего уравнения системы линейных уравнений при $\sigma = 0$. Таким образом, мы получили, что произведение переменных на следующем шаге будет равно не 0, а некоторому числу $\Delta x_i^{aff} \Delta s_i^{aff}$. Чтобы исправить этот недостаток, можно решить аналогичную систему линейных уравнений, которая бы минимизировала значение произведения

$\Delta x_i^{aff} \Delta s_i^{aff}$. Данный шаг называется шагом типа «корректор» (а шаг аффинного шкалирования – соответственно шагом типа «предиктор») и он является решением похожей системы линейных уравнений с аналогичными диагональными матрицами $\Delta X^{aff} \Delta S^{aff}$

$$\begin{pmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{corr} \\ \Delta \lambda^{corr} \\ \Delta s^{corr} \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \Delta X^{aff} \Delta S^{aff} e \end{pmatrix}$$

Обновление всех переменных с помощью суммы двух решений системы линейных уравнений $\left(\Delta x^{aff}, \Delta \lambda^{aff}, \Delta s^{aff}\right) + \left(\Delta x^{corr}, \Delta \lambda^{corr}, \Delta s^{corr}\right)$ позволяет ускорить сходимость метода. На рисунке 2 представлена геометрическая интерпретация этого подхода.

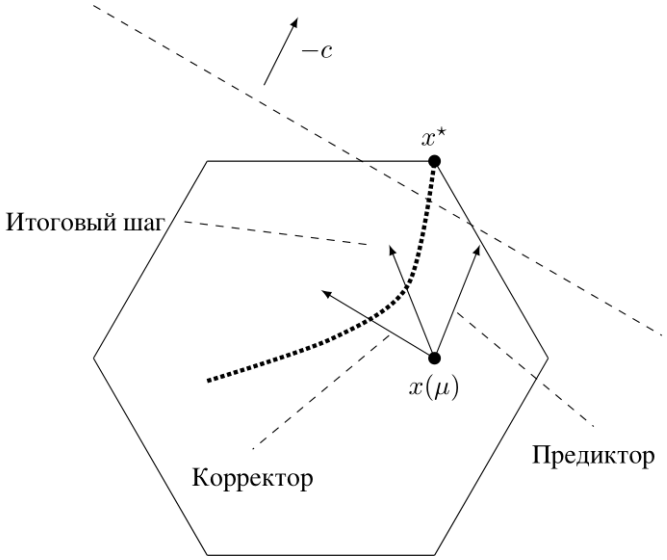

Рис. 2. Принцип работы метода внутренней точки типа «предиктор-корректор»

На основе этих двух шагов можно предложить эвристику для выбора параметра σ_k. Если шаг «предиктор» достигает достаточного прогресса в решении условий Каруша-Куна-Таккера, то параметр σ_k можно уменьшать, иначе – увеличивать. При большом значении σ_k на следующей итерации возможен больший прогресс в решении задачи.

Вычислив направление аффинного шкалирования $\left(\Delta x^{aff}, \Delta \lambda^{aff}, \Delta s^{aff}\right)$, можно вычислить максимальные величины шагов вдоль этого направления по прямым и двойственным переменным

$$\alpha_{aff,max}^p = min\left(1, \min_{i:\Delta x_i^{aff}<0} -\frac{x_i}{\Delta x_i^{aff}}\right)$$

$$\alpha_{aff,max}^d = min\left(1, \min_{i:\Delta s_i^{aff}<0} -\frac{s_i}{\Delta s_i^{aff}}\right)$$

После этого можем вычислить значение критерия остановки, если бы мы совершили эту итерацию

$$\mu_{aff} = \frac{1}{n} \cdot \left(x + \alpha_{aff}^p \Delta x^{aff}\right)^\top \left(s + \alpha_{aff}^d \Delta s^{aff}\right)$$

Параметр σ можно выбрать как степенную функцию от отношения критерия остановки на текущей и на итерации типа «предиктор», что совпадает с ранее предложенным вариантом выбора σ

$$\sigma = \left(\frac{\mu_{aff}}{\mu}\right)^3$$

Практические реализации решают две системы линейных уравнений на каждой итерации. Первая система уравнений – это система для вычисления направления аффинного шкалирования $(\Delta x^{aff}, \Delta \lambda^{aff}, \Delta s^{aff})$, после чего вычисляются параметры μ, μ_{aff}, σ и решается система, в которую были агрегированы корректирующие слагаемые

$$\begin{pmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = -\begin{pmatrix} A^\top\lambda + s - c \\ Ax - b \\ XSe + \Delta X^{aff}\Delta S^{aff}e - \sigma\mu e \end{pmatrix}$$

После чего выбираются новые величины шага по прямым и двойственным переменным $\alpha^p_{k,max}, \alpha^d_{k,max}$(аналогично формулам для шагов аффинного шкалирования, но используя направление $(\Delta x, \Delta\lambda, \Delta s)$) и совершается шаг по направлению, определяемому как решение агрегированной системы. Так как эта система и система для определения шага аффинного шкалирования имеют одинаковую матрицу коэффициентов, то достаточно вычислить LU-разложение данной матрицы один раз, воспользовавшись им дважды и решив вторую систему за $O((2n+m)^2)$ арифметических операций, вместо $O((2n+m)^3)$.

Величину итогового шага на практике также можно выбирать не как максимально допустимые величины шагов $\alpha^p_{k,max}, \alpha^d_{k,max}$, а слегка уменьшать их значения, таким образом оставляя некоторый зазор и не упираясь в границу допустимых значений переменных x, s. Можно выбрать некоторую последовательность $\eta_k \in [0.9,1.0)$, стремящуюся к 1, и выбирать значения итоговых шагов так

$$\alpha^p_k = min\big(1, \eta_k\alpha^p_{k,max}\big)$$
$$\alpha^d_k = min(1, \eta_k\alpha^d_{k,max})$$

Также несложно показать, что если направление $(\Delta x_k, \Delta\lambda_k, \Delta s_k)$ удовлетворяет указанной выше системе и мы определим невязки $r^b_k = Ax_k - b, r^c_k = A^\top\lambda_k + s_k - c$, т. е. выполнены равенства $A\Delta x_k = -(Ax_k - b)$ и $A^\top\Delta\lambda_k + \Delta s_k = -(A^\top\lambda_k + s_k - c)$, то вектора невязок удовлетворяют соотношениям $r^b_{k+1} = \big(1-\alpha^p_k\big)r^b_k$ и $r^c_{k+1} = \big(1-\alpha^d_k\big)r^c_k$. Таким образом, вектора невязок стремятся к нулевым векторам при $k \to \infty$.

Итоговая итерация имеет следующий вид

$$x_{k+1} = x_k + \alpha^p_k\Delta x_k$$
$$(\lambda_{k+1}, s_{k+1}) = (\lambda_k, s_k) + \alpha^d_k(\Delta\lambda_k, \Delta s_k)$$

Выбор начального приближения (x^0, λ^0, s^0) крайне важен для данного алгоритма и неудачный выбор может приводить к расходимости. Идеальным выбором могла бы быть тройка векторов, удовлетворяющая всем линейным ограничениям-равенствам и имеющая положительные компоненты в векторах (x, s), однако на практике такая тройка неизвестна заранее. Вместо этого можно предложить простую эвристику по выбору начального приближения [2, с. 410-411].

Метод внутренней точки в задаче квадратичного программирования

Метод внутренней точки для квадратичного программирования во многом аналогичен методу для задач линейного программирования. Рассмотрим задачу квадратичного программирования в стандартной форме

$$min\frac{1}{2}x^\top Qx + c^\top x$$
$$Ax = b$$
$$Bx \le d$$

Введём дополнительную переменную y и преобразуем ограничение-неравенство в равенство, получим следующую задачу относительно переменных (x, y)

$$\min_{x,y}\frac{1}{2}x^\top Qx + c^\top x$$
$$Ax = b$$
$$Bx + y = d$$
$$y \ge 0$$

Условия Каруша-Куна-Таккера для этой задачи имеют вид

$$Ax = b$$
$$Bx + y = d$$
$$y \geq 0$$
$$s \geq 0$$
$$s_i y_i = 0, i = 1, \dots, m$$
$$Qx + c + A^\top \lambda + B^\top s = 0$$

Поступим по аналогии с линейным программированием и будем искать корень вектор-функции

$$F(x, s, \lambda, y) = \begin{pmatrix} Qx + c + A^\top \lambda + B^\top s \\ Bx + y - d \\ Ax - b \\ \mathcal{Y}Se - \sigma\mu e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Критерий остановки μ определим аналогично $\mu = \frac{y^\top s}{m}$, где m – число строк матрицы B. Матрицы $\mathcal{Y}, S$ – диагональные матрицы, составленные из векторов y, s. Применив метод Ньютона к этой системе, получим систему линейных уравнений

$$J(x, s, \lambda, y)\begin{pmatrix} \Delta x \\ \Delta s \\ \Delta\lambda \\ \Delta y \end{pmatrix} = -F(x, s, \lambda, y) \Leftrightarrow \begin{pmatrix} Q & B^\top & A^\top & 0 \\ B & 0 & 0 & I \\ A & 0 & 0 & 0 \\ 0 & \mathcal{Y} & 0 & S \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta s \\ \Delta\lambda \\ \Delta y \end{pmatrix} = -\begin{pmatrix} Qx + c + A^\top \lambda + B^\top s \\ Bx + y - d \\ Ax - b \\ \mathcal{Y}Se - \sigma\mu e \end{pmatrix}$$

Однако выбор величин шага α^p, α^d по прямым и двойственным переменным в случае квадратичного программирования более сложен. По аналогии с линейным программированием, проанализируем как изменяются вектора невязок от итерации к итерации. Введём обозначения

$$(x^+, y^+) = (x + \alpha^p \Delta x, y + \alpha^p \Delta y)$$
$$(s^+, \lambda^+) = (s + \alpha^d \Delta s, \lambda + a^d \Delta\lambda)$$
$$r_d^+ = Qx^+ + c + A^\top \lambda^+ + B^\top s^+$$
$$r_p^+ = \begin{pmatrix} Bx^+ + y^+ - d \\ Ax^+ - b \end{pmatrix}$$

Тогда можно показать, что вектора невязок на следующей итерации удовлетворяют соотношениям

$$r_d^+ = (1 - \alpha^d) r_d + (\alpha^p - \alpha^d) Q\Delta x$$
$$r_p^+ = (1 - \alpha^p) r_p$$

Таким образом, если $\alpha^p \neq \alpha^d$, то невязка по двойственным переменным r_d может, вообще говоря, увеличиваться и приводить к расхождению алгоритма, а если величина шага по прямым и двойственным переменным совпадает, то обе невязки линейно убывают на каждой итерации. Поэтому будем рассматривать случай, когда величина шага является общей для обоих переменных.

Хорошим вариантом выбора величины шага может быть вычисление двух величин, где параметр t регулирует близость полученного шага $y + \alpha\Delta y$ к вектору с нулевыми компонентами

$$\alpha_t^p = max\{\alpha \in (0; 1] | y + \alpha\Delta y \geq (1 - t)y\}$$
$$\alpha_t^d = max\{\alpha \in (0; 1] | s + \alpha\Delta s \geq (1 - t)s\}$$

После этого можно вычислить общую величину шага $\alpha = min(\alpha_t^p, \alpha_t^d)$.

Таким образом, метод внутренней точки типа «предиктор-корректор» для задач квадратичного программирования сначала вычисляет направление аффинного шкалирования, решая систему линейных уравнений с $\sigma = 0$

$$\begin{pmatrix} Q & B^\top & A^\top & 0 \\ B & 0 & 0 & I \\ A & 0 & 0 & 0 \\ 0 & \mathcal{Y} & 0 & S \end{pmatrix}\begin{pmatrix} \Delta x^{aff} \\ \Delta s^{aff} \\ \Delta\lambda^{aff} \\ \Delta y^{aff} \end{pmatrix} = -\begin{pmatrix} Qx + c + A^\top \lambda + B^\top s \\ Bx + y - d \\ Ax - b \\ \mathcal{Y}Se - \sigma\mu e \end{pmatrix}$$

Затем вычисляются величины

$$\mu = \frac{y^\top s}{m}$$
$$\alpha_{aff} = max\{\alpha \in (0;1] | (s,y) + \alpha(\Delta s^{aff}, \Delta y^{aff}) \geq 0\}$$
$$\mu_{aff} = \frac{\left(y + \alpha_{aff}\Delta y^{aff}\right)^\top \left(s + \alpha_{aff}\Delta s^{aff}\right)}{m}$$
$$\sigma = \left(\frac{\mu_{aff}}{\mu}\right)^3$$

После этого вычисляется направление типа «корректор» с поправками, агрегированными в правую часть системы линейных уравнений

$$\begin{pmatrix} Q & B^\top & A^\top & 0 \\ B & 0 & 0 & I \\ A & 0 & 0 & 0 \\ 0 & \mathcal{Y} & 0 & S \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta\lambda \\ \Delta y \end{pmatrix} = -\begin{pmatrix} Qx + c + A^\top\lambda + B^\top s \\ Bx + y - d \\ Ax - b \\ \mathcal{Y}Se + \Delta\mathcal{Y}^{aff}\Delta S^{aff}e - \sigma\mu e \end{pmatrix}$$

В конце вычисляется общий шаг, используя решение этой системы, и совершается обновление всех переменных

$$\alpha = min\left(\alpha_t^p, \alpha_t^d\right)$$
$$(x_{k+1}, s_{k+1}, \lambda_{k+1}, y_{k+1}) = (x_k, s_k, \lambda_k, y_k) + \alpha(\Delta x_k, \Delta s_k, \Delta\lambda_k, \Delta y_k)$$

Число t_k можно выбирать в виде некоторой предопределённой последовательности вещественных чисел, таких, что $t_k \to 1$ при $k \to \infty$. В данном случае аналогично имеем две системы линейных уравнений с одинаковыми матрицами, поэтому рекомендуется посчитать LU-разложение один раз и с помощью него решить вторую систему за меньшее количество арифметических операций.

Выбор хорошего начального приближения также достаточно важен для данного алгоритма. Можно воспользоваться следующей эвристикой – получить начальное приближение $(\hat{x}, \hat{s}, \hat{\lambda}, \hat{y})$ от пользователя (или инициализировать случайными числами, положительными для s, y), после чего вычислить направление аффинного шкалирования и положить начальное приближение равным $(x, \lambda) = \left(\hat{x}, \hat{\lambda}\right), s = max\left(1, \left|\hat{s} + \Delta s^{aff}\right|\right), y = max\left(1, \left|\hat{y} + \Delta y^{aff}\right|\right)$, где модуль и максимум применяется покомпонентно.

Методы внутренней точки в задачах полуопределённого программирования

Прямодвойственные методы внутренней точки для задач полуопределённого программирования хоть и похожи идейно, имеют достаточно сильные отличия от аналогичных методов для линейного и квадратичного программирования. Рассмотрим задачу полуопределённого программирования в стандартной форме

$$\min_X \langle C, X\rangle$$
$$\langle A_i, X\rangle = b_i, i = 1, \dots, m$$
$$X \succcurlyeq 0$$

Ограничение $X \succcurlyeq 0$ говорит, что матрица X должна быть симметрична и положительно полуопределёна, а матрицы C, A_i предполагаются заданными симметричными матрицами. Однако с помощью симметричных матриц нельзя задать линейные ограничения на симметричность матрицы X, так как симметричность следовала бы из ортогональности матрицы X всем кососимметричным матрицам, т. е. некоторые матрицы A_i должны были бы быть кососимметричными. Таким образом метод внутренней точки должен некоторым образом сам поддерживать симметричность этих матриц и решать этот вопрос отдельно. Для этого есть несколько подходов, мы рассмотрим метод использующий т. н. направления Нестерова-Тода (или точку шкалирования).

Для определения понятия точки шкалирования необходимо ввести понятие барьера. Барьером на конусе K называется такая выпуклая дифференцируемая функция $F(x)$, что $F(x) \to +\infty$ при $x \to \partial K$, где ∂K – граница множества (конуса) K. Множество K называется конусом, если для любого $x \in K$ и

для любого числа $\lambda \geq 0$ верно, что $\lambda x \in K$. Двойственный конус к конусу K называется выпуклый конус

$$K^* = \{s \mid \langle s, x\rangle \geq 0 \ \forall x \in K\}$$

Множество симметричных положительно полуопределённых матриц является самодвойственным выпуклым конусом, самодвойственность означает, что $K^* = K$. На этом конусе существует в некотором смысле «канонический» барьер $F(X) = -\ln det(X)$, можно показать, что на данном множестве данная функция выпукла и при $X \to \partial K$ имеет место $F(X) \to +\infty$. Также имеет место понятия автошкалированных барьеров и конусов, которые мы здесь не приводим, но важным следствием определения является тот факт, что для любого $X \in int\ K$ и для любого $S \in int\ K^*$, существует единственная точка шкалирования $W \in int\ K$, удовлетворяющей уравнению

$$\nabla^2 F(W)X = S$$

Здесь под $int\ K$ подразумевается внутренность конуса K, а $\nabla^2 F(W)$ – матрица (или тензор) вторых производных барьера F. Также можно определить двойственную функцию $F_*(S)$ – автошкалированный барьер для K^*

$$F_*(S) = sup\{-\langle S, X\rangle - F(X) \mid X \in K\}$$

$F_*(S)$ обладает свойством, что для неё выполнено аналогичное свойство и существует двойственная точка шкалирования $T = -\nabla F(W) \in int\ K^*$, удовлетворяющая соотношениям

$$\nabla^2 F_*(T) = \left(\nabla^2 F(W)\right)^{-1}$$
$$\nabla^2 F_*(T)S = X$$

Для $F(X) = -\ln det(X)$ можно показать, что $\nabla F(X) = -X^{-1}$, а $\nabla^2 F(W)$ – это линейный оператор, действующий по следующему принципу

$$\nabla^2 F(W)X = W^{-1}XW^{-1}$$

Соответственно, получим уравнение, из которого необходимо выразить точку шкалирования

$$\nabla^2 F(W)X = W^{-1}XW^{-1} = S$$
$$\left(X^{\frac{1}{2}}W^{-1}X^{\frac{1}{2}}\right)^2 = X^{\frac{1}{2}}SX^{\frac{1}{2}}$$
$$W = X^{\frac{1}{2}}\left(X^{\frac{1}{2}}SX^{\frac{1}{2}}\right)^{-\frac{1}{2}}X^{\frac{1}{2}}$$

Аналогичным образом, двойственную точку шкалирования T можно вычислить воспользовавшись соотношением для обратного оператора и определением точки шкалирования

$$\nabla^2 F_*(T)S = \left(\nabla^2 F(W)\right)^{-1}S = WSW = X$$
$$W = S^{-\frac{1}{2}}\left(S^{\frac{1}{2}}XS^{\frac{1}{2}}\right)^{\frac{1}{2}}S^{-\frac{1}{2}}$$

Наконец, рассмотрим релаксированные условия Каруша-Куна-Таккера для задачи полуопределённого программирования с $\mu > 0$

$$\langle A_i, X\rangle = b_i, i = 1, \dots, m$$
$$\sum_{i=1}^{m} y_i A_i + S = C$$
$$XS = \mu I$$

В данном случае, параметр μ можно выбирать как скалярное произведение, аналогично введём параметр центрирования $\sigma \in [0; 1]$

$$\mu = \frac{\langle X, S\rangle}{n}$$

Аналогичным образом составим векторно-матрично-значную функцию, корень которой необходимо найти, и выпишем систему линейных уравнений уравнения метода Ньютона

$$\begin{gathered}\langle A_i, \Delta X\rangle = -(\langle A_i, X\rangle - b_i)\\ \sum_{i=1}^{m}\Delta y_i A_i + \Delta S = -\left(\sum_{i=1}^{m} y_i A_i + S - C\right)\\ \Delta X S + X\Delta S = -(XS - \sigma\mu I)\end{gathered}$$

Сразу становится видно, что направления $(\Delta X, \Delta y, \Delta S)$, получаемые из этой системы после применения метода Ньютона, вообще говоря, не обязаны быть симметричными матрицами, а именно матрица ΔS гарантировано симметрична в силу второго уравнения и симметричности матриц A_i, C, а вот матрица ΔX уже не обязана быть симметричной и соответственно матрица $X + \Delta X$ тоже. Для решения этой проблемы можно ввести симметризующую функцию $H_P(M)$

$$H_P(M) = \frac{1}{2}(PMP^{-1} + P^{-\top}M^{\top}P^{\top})$$

Данная функция параметризуется некоторой матрицей P. Также можно показать, что если P обратима, а матрица M подобна симметричной положительно определённой матрице, тогда

$$H_P(M) = \mu I \Leftrightarrow M = \mu I$$

Это означает, что последнее уравнение в условиях Каруша-Куна-Таккера $XS = \mu I$ можно заменить на уравнение $H_P(XS) = \mu I$, так как XS подобна $S^{\frac{1}{2}}XS^{\frac{1}{2}}$

Соответственно последнее уравнение в системе Ньютона может быть заменено на эквивалентное

$$H_P(\Delta X S + X\Delta S) = -(H_P(XS) - \sigma\mu I)$$

Данная система имеет т. н. кронекеровскую структуру, так как содержит произведения матриц коэффициентов и неизвестных матриц в разном порядке. Програмное обеспечение для решения систем линейных уравнений исторически поддерживает только системы вида $Ax = b$, т. е. в векторном формате с заданной матрицей A и вектором правой части b, поэтому было бы удобно привести нашу систему к аналогичному виду. Для этого будет удобно ввести линейный оператор симметричной векторизации $svec$ и симметричное произведение Кронекера $\otimes_s$

Симметричная векторизация отображает симметричную $n \times n$ матрицу в вектор размерности $n(n+1)/2$, фактор $\sqrt{2}$ позволяет $svec$ быть изометрией, т. е. отображением, сохраняющим расстояния между симметричными матрицами $\mathbb{S}^n$ и их образами.

$$\begin{gathered}svec : \mathbb{S}^n \to \mathbb{R}^{n(n+1)/2}\\ svec(H) = \left(h_{11}, \sqrt{2}h_{12}, \dots, \sqrt{2}h_{n1}, h_{22}, \sqrt{2}h_{32}, \dots, \sqrt{2}h_{n2}, \dots, h_{nn}\right)^{\top}\end{gathered}$$

Симметричное произведение Кронекера двух $n \times n$ (не обязательно симметричных) матриц определяется неявно как матрица размера $n(n+1)/2$, отображающая вектор $svec(H)$, где $H \in \mathbb{S}^n$, по правилу

$$(G \otimes_s K)svec(H) = \frac{1}{2}svec(GHK^{\top} + KHG^{\top})$$

Очевидно, что отображение $svec$ обратимо, будем обозначать обратное к нему отображение как $smat : \mathbb{R}^{n(n+1)/2} \to \mathbb{S}^n$. Используя эти операции и симметризовав последнее уравнение, запишем систему Ньютона в компактной форме, где I – единичная матрица размеров $n(n+1)/2$

$$\begin{pmatrix} 0 & \mathcal{A} & 0\\ \mathcal{A}^{\top} & 0 & I\\ 0 & E & F\end{pmatrix}\begin{pmatrix}\Delta y\\ svec(\Delta X)\\ svec(\Delta S)\end{pmatrix} = \begin{pmatrix} r_p\\ svec(R_d)\\ svec(R_c)\end{pmatrix}$$

$$r_p = b - \mathcal{A}svec(X)$$

$$R_d = C - S - \sum_{i=1}^{m} y_i A_i$$

$$R_c = \sigma\mu I - H_P(XS)$$

$$E = P \otimes_s P^{-\top} S$$
$$F = PX \otimes_s P^{-T}$$
$$\mathcal{A}^\top = (svec(A_1) \quad \dots \quad svec(A_m))$$

Направлением Нестерова-Тода называется решение $(\Delta X, \Delta y, \Delta S)$ системы линейных уравнений метода Ньютона, модифицированной домножением на точку шкалирования W и однородным первым и вторым уравнением (т. е. предполагается, что тройка (X, y, S) удовлетворяет ограничениям-равенствам)

$$\langle A_i, \Delta X \rangle = 0, i = 1, \dots, m$$
$$\sum_{i=1}^{m} \Delta y_i A_i + \Delta S = 0$$
$$W^{-1} \Delta X \, W^{-1} + \Delta S = \sigma \mu X^{-1} - S$$

Оказывается, что если в системе для направлений Нестерова-Тодда опустить предположение и выполнении ограничений и заменить нули в правой части на r_p, R_d соответственно, то при выборе матрицы P такой, что она обратима и $P^\top P = W^{-1}$ решения системы Нестерова-Тодда и решения системы с симметричным произведением Кронекера совпадают [3]. Выбор такой P не единственен, далее будет предложен вариант как это делать.

Вычисление матрицы W

Технически, вычислять матрицу W можно по ранее приведённым формулам, но это не эффективно. Рассмотрим положительно определённые симметричные матрицы X, S и их разложения Холецкого, а также сингулярное разложение матрицы $R^\top L$

$$X = LL^\top, S = RR^\top, R^\top L = UDV^\top$$

Определим матрицу $Q = L^{-1} X^{\frac{1}{2}}$, где $X^{\frac{1}{2}}$ обозначает матричный корень, несложно показать, используя выше указанные определения, что Q – ортогональная матрица. Далее получим спектральное разложение $VD^2V^\top$ матрицы $L^\top RR^\top L = L^\top SL$, пользуясь тем, что $R^\top L = UDV^\top$

$$X^{\frac{1}{2}} S X^{\frac{1}{2}} = Q^\top (L^\top R)(R^\top L) Q = (Q^\top V) D^2 (V^\top Q)$$
$$L^\top SL = VD^2V^\top$$

Так как произведение ортогональных матриц $Q^\top V$ также ортогональная матрица, тогда

$$\left(X^{\frac{1}{2}} S X^{\frac{1}{2}} \right)^{-\frac{1}{2}} = (Q^\top V) D^{-1} (V^\top Q)$$

Отсюда легко получается выражение для W, учитывая, что нужно также домножить слева и справа на $X^{\frac{1}{2}} = LQ = Q^\top L^\top$

$$W = LVD^{-1}V^\top L^\top = GG^\top, G = LVD^{-\frac{1}{2}}$$
$$G^{-1} = D^{\frac{1}{2}} V^\top L^{-1}$$

Также можно заметить, что $G^{-\top} G^{-1} = W^{-1}$, поэтому $P = G^{-1}$ удовлетворяет соотношению $P^\top P = W^{-1}$ и поэтому это один из способов выбрать матрицу P.

Решение системы Ньютона для SDP

Заметим, что при $P = G^{-1}$имеют место два равенства

$$G^\top SG = G^{-1} X G^\top = D$$

Таким образом $P^{-\top} S P^{-1} = PXP^\top$ это одна и та же диагональная матрица и они коммутируют. Также заметим, что при $P = G^{-1}$ выражение $H_P(XS)$ принимает вид

$$H_P(XS) = \frac{1}{2}(G^{-1}XSG + G^{\top}SXG^{-\top}) = D^2$$

Далее подставим данное P в систему, описанную ранее, и подставим вместо $H_P(XS) = D^2$, получим систему

$$\begin{pmatrix} 0 & \mathcal{A} & 0 \\ \mathcal{A}^{\top} & 0 & I \\ 0 & E & F \end{pmatrix}\begin{pmatrix} \Delta y \\ svec(\Delta X) \\ svec(\Delta S) \end{pmatrix} = \begin{pmatrix} r_p \\ svec(R_d) \\ svec(R_c) \end{pmatrix}$$

$$r_p = b - \mathcal{A}svec(X)$$

$$R_d = C - S - \sum_{i=1}^{m} y_i A_i\,, \qquad R_c = \sigma\mu I - D^2$$

$$E = G^{-1} \otimes_s G^{\top}S, \qquad F = G^{-1}X \otimes_s G^{\top}$$

Хоть данная система и является системой линейных уравнений, она не может быть решена напрямую так как матрицы E, F заданы неявно через симметричное произведение Кронекера. Однако воспользовавшись блочным методом Гаусса, эту систему можно свести к серии более простых систем.

$$(\mathcal{A}E^{-1}F\mathcal{A}^{\top})\Delta y = r_p + \mathcal{A}E^{-1}Fsvec(R_d) - \mathcal{A}E^{-1}svec(R_c)$$

В данном случае матрица E^{-1}сложна в получении, но реально требуется только матрица $E^{-1}F$ и вектор $E^{-1}svec(R_c)$. Можно показать, что данные матрица и вектор выражаются как

$$E^{-1}F = W \otimes_s W, E^{-1}svec(R_c) = svec(\sigma\mu S^{-1} - X)$$

Свойства симметричного произведения Кронекера также позволяют вычислить разложение Холецкого для $W \otimes_s W = HH^{\top}$

$$W \otimes_s W = (G \otimes_s G)(G \otimes_s G)^{\top}, H = G \otimes_s G$$

Поэтому уравнение на Δy можно записать в следующем виде, где X_r это $n \times n$ симметричная матрица, такая, что $\mathcal{A}svec(X_r) = r_p = b - \mathcal{A}svec(X)$

$$\mathcal{A}H(\mathcal{A}H)^{\top}\Delta y = \mathcal{A}H(H^{\top}svec(R_d) + H^{-1}[svec(X_r) - E^{-1}svec(R_c)])$$

Если матрица $\mathcal{A}$ полного ранга (матрицы A_i исходной задачи линейно независимы), то матрица X_r всегда существует. В случае когда матрица X удовлетворяет линейным ограничениям исходной задачи, в качестве X_r можно брать нулевую матрицу. Далее несложно заметить, что система линейных уравнений на Δy крайне похожа на нормальные уравнения в задаче наименьших квадратов и это действительно так. Несложно заметить, что эта система эквивалента задаче наименьших квадратов

$$min\ \|B^{\top}\Delta y - h\|_2^2$$

Здесь матрица $B^{\top}$и вектор h имеют вид

$$B^{\top} = H^{\top}\mathcal{A}^{\top} = [svec(G^{\top}A_1G) \quad \dots \quad svec(G^{\top}A_mG)]$$
$$h = svec(G^{\top}R_dG + G^{-1}X_rG^{-\top} - \sigma\mu D^{-1} + D)$$

Решение Δy может быть получено с помощью QR разложения матрицы $B^{\top}$. После нахождения Δy, приращение ΔS можно найти с помощью подстановки решения Δy в исходные уравнения

$$\Delta S = R_d - smat(\mathcal{A}^{\top}\Delta y)$$

Приращение ΔX можно получить двумя способами, можно подставить $\Delta y, \Delta S$ в исходную систему уравнений и выразить ΔX

$$\Delta X = smat\big(E^{-1}svec(R_c) - E^{-1}Fsvec(\Delta S)\big)$$

Можно показать, что второй способ вычисления ΔX – с помощью вычисления невязки в задаче наименьших квадратов на Δy [3].

$$\Delta X = -H(h - B^{\top}\Delta y) + X_r = -G\big(smat(h - B^{\top}\Delta y)\big)G^{\top} + X_r$$

Вычисление шага типа «корректор»

До этого был рассмотрен только способ решения основной системы, определяющей шаг типа «предиктор», если подставить значение $\sigma = 0$. Теперь предположим, что у нас уже есть вычисленные по ранее описанным формулам значения приращений для матриц X, S и обозначим их $\delta X, \delta S$ соответственно. Рассмотрим линеаризацию условия дополняющей нежёсткости и при этом не будем отбрасывать квадратичные слагаемые, вместо них подставим найденные величины $\delta X, \delta S$

$$H_P(\Delta X S + X \Delta S) = \sigma \mu I - H_P(XS + \delta X \delta S)$$

Тогда получим аналогичное уравнение на $\Delta X, \Delta S$

$$Esvec(\Delta X) + Fsvec(\Delta S) = svec(R_s)$$
$$R_s = R_c + R_q = R_c - H_P(\delta X \delta S)$$

В качестве матрицы P в данном случае можно выбрать, как и раньше, матрицу $P = G^{-1}$. Повторив аналогичные выкладки, можно показать, что для нахождения решения $(\Delta y, \Delta X, \Delta S)$ системы с поправками $(\delta X, \delta S)$ достаточно в выражении для h заменить $-\sigma \mu D^{-1} + D$ на $-\sigma \mu D^{-1} + D - R_{NT}$, где поправка R_{NT} определяется как

$$svec(R_{NT}) = H^{-1}E^{-1}svec\left(-H_{G^{-1}}(\delta X \delta S)\right)$$

Таким образом $svec(R_{NT})$ – решение системы линейных уравнений

$$EHsvec(R_{NT}) = svec\left(-H_{G^{-1}}(\delta X \delta S)\right)$$

Поскольку $H = G \otimes_s G$ и $E = G^{-1} \otimes G^{\top} S$ и так как $G^{\top} S G = D$ – диагональная матрица, то данное уравнение можно упростить до известного уравнения Ляпунова и затем записать явное решение

$$(G^{\top} S G) R_{NT} + R_{NT}(G^{\top} S G) = -G^{-1}(\delta X \delta S) G - G^{\top}(\delta S \delta X) G^{-\top}$$

$$R_{NT} = -\frac{1}{2} D^{-1}(G^{-1}(\delta X \delta S) G + G^{\top}(\delta S \delta X) G^{-\top})$$

Описание алгоритма типа «предиктор-корректор»

Описав способ решения исходной системы линейных уравнений и способ вычисления поправки, можно описать итоговый алгоритм. На вход подаётся тройка (X_0, y_0, S_0) с положительно определёнными симметричными матрицами (X_0, S_0) (можно брать единичными или единичными, умноженными на положительное число), а также число $\tau \in (0; 1)$. Далее для $k = 0,1, \ldots$ повторять

1. Вычислить параметр $\mu_k = \frac{1}{n} \cdot \langle X_k, S_k \rangle$
2. Вычислить приращения $(\delta X_k, \delta y_k, \delta S_k)$, решив основную систему линейных уравнений при $\sigma = 0$, пользуясь ранее выведенными формулами
3. Вычислить длины шагов $\alpha_p = min\left(1, -\frac{\tau}{\lambda_{min}(X_k^{-1} \delta X_k)}\right), \alpha_d = min\left(1, -\frac{\tau}{\lambda_{min}(S_k^{-1} \delta S_k)}\right)$
4. Вычислить параметр центрирования $\sigma_k = \frac{\left[\langle X_k + \alpha_p \delta X_k, S_k + \alpha_d \delta S_k \rangle\right]^2}{[\langle X_k, S_k \rangle]^2}$
5. Определив μ_k, σ_k, решить систему линейных уравнений с поправкой $\delta X_k \delta S_k$ и найти $(\Delta X_k, \Delta y_k, \Delta S_k)$
6. Вычислить α_p, α_d для направлений $\Delta X_k, \Delta S_k$ по ранее указанным формулам
7. Обновить переменные $(X_{k+1}, y_{k+1}, S_{k+1}) = (X_k + \alpha_p \Delta X_k, y_k + \alpha_d \Delta y_k, S_k + \alpha_d \Delta S_k)$
8. Обновить $k := k + 1$, перейти к шагу 1

Замечания

- Параметр τ обычно выбирается близким к единице, например $\tau = 0.98$.
- Задачи SDP чувствительны к малым значениям σ, поэтому в них рекомендуется установить нижнюю границу для значений σ_k. Например, по формуле $\widehat{\sigma_k} = \begin{cases} max(0.05, \sigma_k), \text{если } \alpha_p + \alpha_d \geq 1.8 \\ max(0.1, \sigma_k), \text{если } 1.4 \leq \alpha_p + \alpha_d < 1.8 \\ max(0.2, \sigma_k), \text{если } \alpha_p + \alpha_d < 1.4 \end{cases}$
- Решение двух задач наименьших квадратов на $\delta y, \Delta y$ будет дешевле, если сохранить QR разложение матрицы $B^\top$ и воспользоваться им на шаге «корректор».
- Алгоритм останавливается, если $\sigma_k > 1$ или шаги α_p, α_d становятся меньше 10^{-8}.

Примеры задач оптимизации

Задача минимизации спектрального радиуса

Задача минимизации спектрального радиуса встречается в задачах теории управления, когда необходимо определить, является ли устойчивой линейная динамическая система (или, что то же самое, сходится ли траектория к нулю)

$$u_{n+1} = A(x)u_n$$

$$A(x) = A_0 + \sum_{k=1}^{m} x_k A_k$$

Эта последовательность сходится, если спектральный радиус матрицы $\rho\big(A(x)\big) < 1$, соответственно получим задачу выпуклой оптимизации с негладкой целевой функцией, зависящей от $x \in \mathbb{R}^m$

$$min\, \rho\big(A(x)\big) = min \left\| A_0 + \sum_{k=1}^{m} x_k A_k \right\|_2$$

Эта задача эквивалентна задаче полуопределённого программирования с переменными $(t, x, S) \in \mathbb{R} \times \mathbb{R}^m \times \mathbb{S}^n$

$$min\, t$$

$$S = A_0 + \sum_{k=1}^{m} x_k A_k$$

$$\begin{pmatrix} tI & S \\ S^\top & tI \end{pmatrix} \succcurlyeq 0$$

Можно показать, что двойственная задача к данной будет задачей полуопределённого программирования в стандартной форме, для которой применим метод внутренней точки. Решив двойственную задачу, получим набор двойственных (относительно двойственной задачи) переменных $(y_1, \dots, y_{k+1}) = (t, x_1, \dots, x_k)$, являющихся решением исходной задачи. Соответственно, если $y_1 = t < 1$, то ответ на вопрос о сходимости последовательность u_n будет положительным и отрицательным иначе.

Возьмём следующие матрицы в качестве входных данных и запустим метод внутренней точки для полуопределённого программирования

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} A_1 = \begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix} A_2 = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

Обозначим $J_k = \begin{pmatrix} t_k I & S_k \\ S_k^\top & t_k I \end{pmatrix}$, под матрицей F_k будем понимать некоторую матрицу из двойственной задачи, изменяемую на каждой итерации. Ниже на рисунке 3 изображены динамики изменений норм невязок ограничений и условие дополняющей нежёсткости $\langle F_k, J_k \rangle$, а также его аффинный прогноз,

если бы метод не делал итераций типа «корректор». На рисунке 4 изображена двойственная функция (считающаяся исходной) $\langle C, F_k \rangle$ для некоторой матрицы коэффициентов C, а также функция из исходной задачи $-t_k$. Видно, что почти на всех итерациях выполняется неравенство $-t_k \leq \langle C, F_k \rangle$ по свойствам двойственной функции, за исключением первой. Это вызвано тем, что метод стартует из точки, не удовлетворяющей ограничениями-равенствам, это можно заметить по ненулевым нормам невязок на рисунке 3 на первых двух итерациях.

В результате было получено решение $(t, x_1, x_2) = (0.948683, -1.8, 1.7)$, т. е. спектральный радиус может быть уменьшен до значения $0.948683 < 1$ при указанных значениях x_1, x_2.

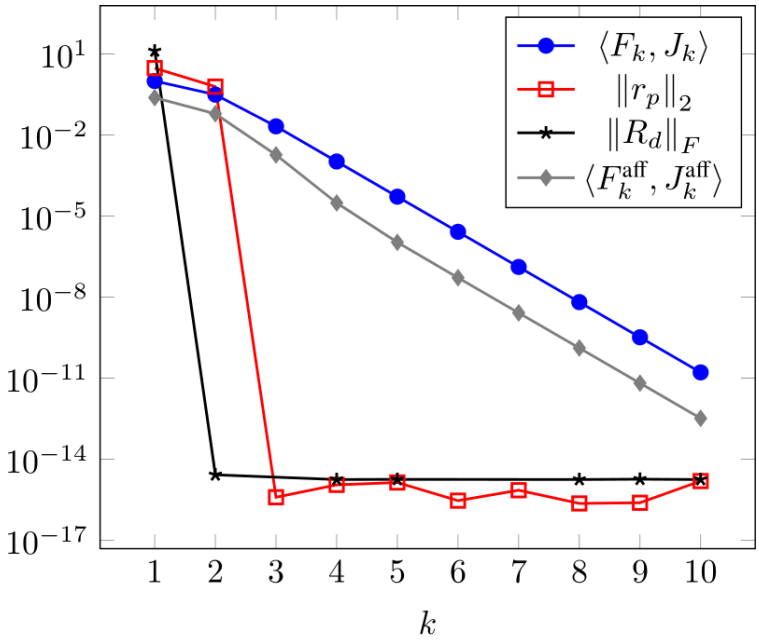

Рис. 3. Изменение норм ограничений и условия дополняющей нежёсткости

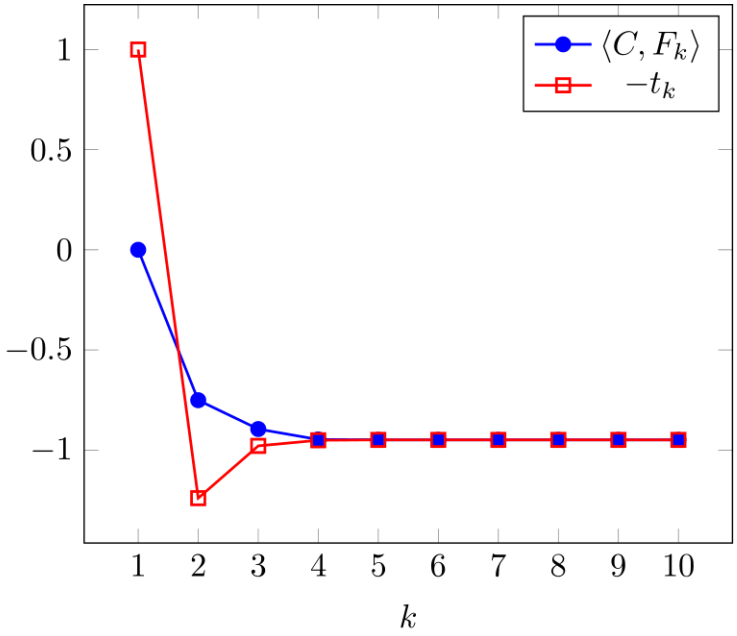

Рис. 4. Изменение прямой и двойственной функции

Задача о тестировании студентов (Educational Testing Problem)

Данная задача ставит перед собой цель оценить коэффициент ρ, определяющий надёжность или точность экзаменационного теста [1]. Нижняя оценка на данный коэффициент является оптимальным

значением задачи полуопределённого программирования с заданной симметричной положительно полуопределённой матрицей A

$$max\ 1^{\top}d = \sum_{k=1}^{n} d_k$$

$$A - diag(d) \succcurlyeq 0$$

Аналогично предыдущему примеру, необходимо построить двойственную задачу к данной и решить её методом внутренней точки. Сгенерируем случайную симметричную положительно определённую матрицу A и запустим на этой задаче в размерности $n = 3$ метод внутренней точки. В качестве решения получим набор двойственных переменных, связанных с переменными d как $(d_1, d_2, d_3) = (-y_1, -y_2, -y_3) = (1.381, 1.55802, 1.60382)$, таким образом оптимальное значение исходной задачи равно $\sum_k d_k = 4.54284$. На рисунках 5 и 6 приведены аналогичные ранее представленным графикам сходимости, где матрица $J_k = A - diag(d_k)$, а матрица F_k – матрица переменных из двойственной задачи.

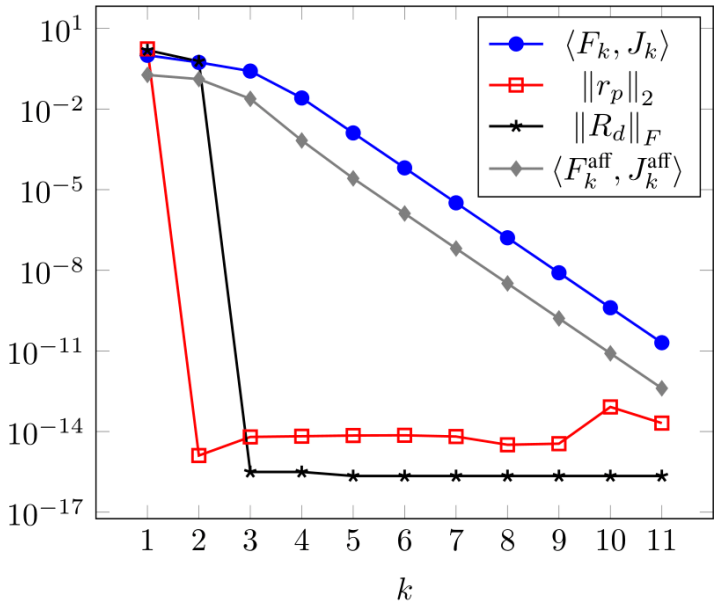

Рис. 5. Сходимость метода внутренней точки в задаче о тестировании

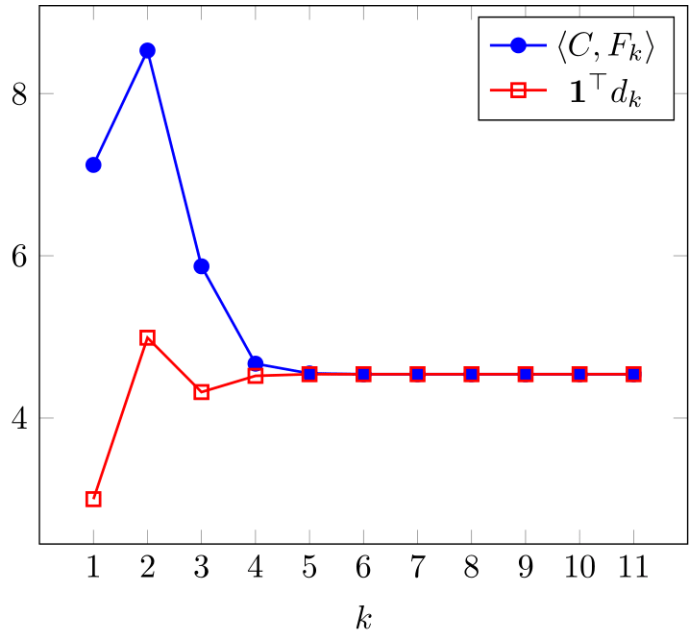

Рис. 6. Изменение исходной и двойственной функции в задаче о тестировании

В данном случае можно было получить в том числе и аналитическое решение $d^\star = 1 \cdot n\lambda_{min}(A)$, где 1 – единичный вектор, после чего вычислить оптимальное значение.

Задача Basis Pursuit

В данной задаче необходимо найти хорошее приближение x, которое бы решало систему линейных уравнений $Ax = y$ таким образом, что x содержит как можно больше нулевых компонент. Эта задача формируется как задача выпуклой негладкой оптимизации

$$min\ \|x\|_1$$
$$Ax = y$$

Сделав замену переменных $x = x^+ - x^-$ и добавив ограничение $x^+, x^- \geq 0$, можно избавиться от модуля в целевой функции и получить задачу линейного программирования в стандартной форме

$$min\ (1^\top \quad 1^\top)\begin{pmatrix} x^+ \\ x^- \end{pmatrix}$$
$$(A \quad -A)\begin{pmatrix} x^+ \\ x^- \end{pmatrix} = b$$
$$(x^+ \quad x^-) \geq 0$$

Предположим, что нам дан некоторый сигнал $y(t)$, а точнее его значения $y(t_i)$ в моменты времени $t_i, i = 0, \dots, m-1$. Тогда рассмотрим задачу аппроксимации этого сигнала с помощью линейной комбинации синусов и косинусов с заданными частотами

$$y(t_i) \approx \sum_{k=0}^{n-1} a_k\, sin\big((k+1)t_i\big) + b_k\, cos(kt_i) = \hat{y}(t_i), i = 0, \dots, m-1$$

Причём, заранее известно, что сигнал является разреженным, т. е. большая часть коэффициентов a_k, b_k не существенны и могут быть положены равными нулю, в итоге требуется найти все эти коэффициенты. Тогда имеет смысл скомпоновать их в вектор неизвестных $x = (a_0, \dots, a_{n-1}, b_0, \dots, b_{n-1}) \in \mathbb{R}^{2n}$ и минимизировать его первую норму $\|x\|_1$ при ограничении $Ax = y$, где $y = \big(y(t_0), \dots, y(t_{m-1})\big)$, а матрица A имеет вид

$$A = \begin{pmatrix} sin(t_0) & sin(2t_0) & \cdots & sin(nt_0) & cos(0t_0) & cos(t_0) & \cdots & cos\big((n-1)t_0\big) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ sin(t_{m-1}) & sin(2t_{m-1}) & \cdots & sin(nt_{m-1}) & cos(0t_{m-1}) & cos(t_{m-1}) & \cdots & cos\big((n-1)t_{m-1}\big) \end{pmatrix}$$

Решим соответствующую задачу линейного программирования методом внутренней точки, найдя переменные (x^+, x^-), после чего восстановим из него реальные значения коэффициентов по формуле $x = x^+ - x^-$. В качестве примера была взята задача на отрезке $t \in [0; 14]$, в котором было вычислено 30 значений сигнала $y(t)$ на равномерной сетке, сигнал $y(t)$ имел вид

$$y(t) = sin(t) + cos(2t) + cos(sin(t)) + sin(t) \cdot cos(t)$$

Задача решалась с параметрами $m = 30, n = 100$, всего необходимо найти $2n = 200$ коэффициентов, из которых всего 13 оказались ненулевыми (коэффициент считался нулевым, если он имел значение по модулю меньше, чем 10^{-10}). При $m = 30$ и $n = 100$ сигнал $\hat{y}(t)$ становится визуально неотличим от сигнала $y(t)$, поэтому график сравнения не приводим.

На рисунке 7 приведён аналогичный график условия дополняющей нежёсткости и норм невязок (в соответствии с обозначениями, использовавшимися в разделе о линейном программировании).

Обозначив $c^\top = (1^\top \quad 1^\top), \hat{x} = (x^+ \quad x^-)^\top$ (т. е. целевую функцию в соответствующей задаче линейного программирования) и обозначив $g(\lambda, s)$ двойственную к ней, на рисунке 8 изображён график, иллюстрирующий неравенство $g(\lambda, s) \leq c^\top \hat{x}$

На рисунке 9 приведены изменения значений величин шага по прямым и двойственным переменным, а также изменение параметра σ.

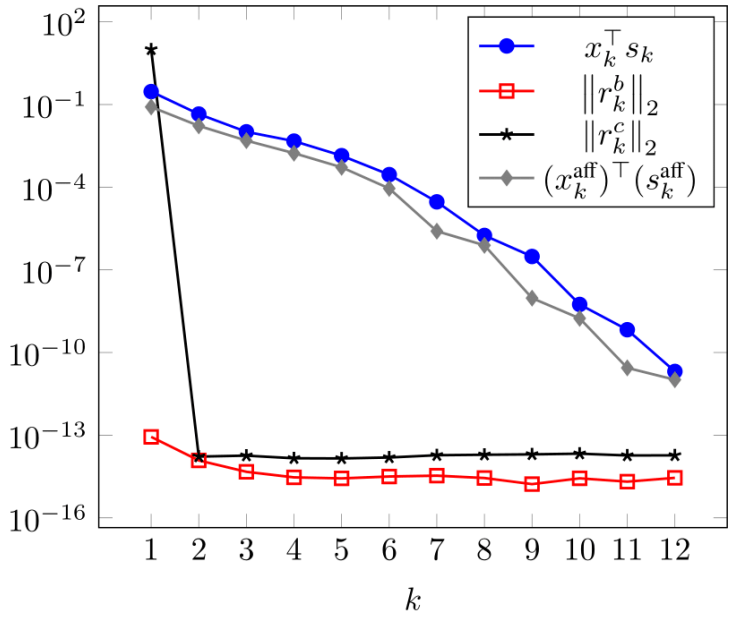

Рис. 7. Сходимость метода внутренней точки в задаче Basis Pursuit

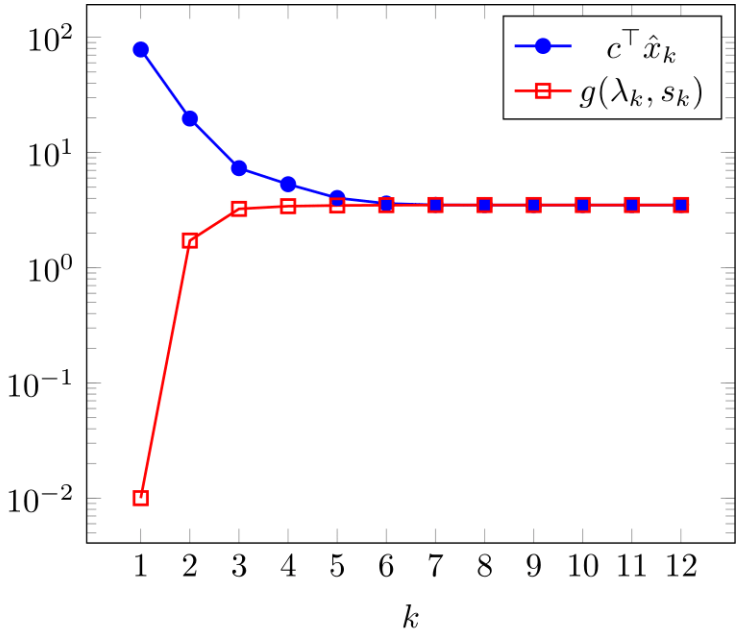

Рис. 8. Сходимость значений прямой и двойственной функции в задаче Basis Pursuit

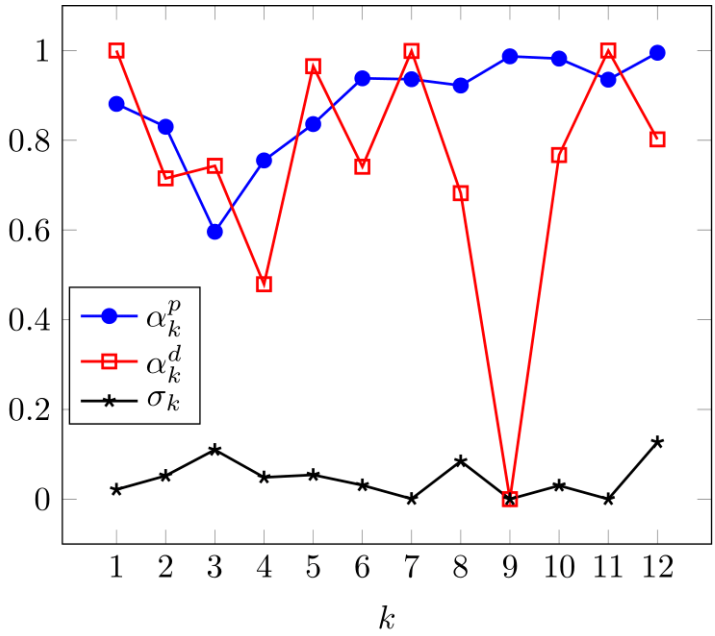

Рис. 9. Динамика изменения величин шага и параметра центрирования в задаче Basis Pursuit

Заключение

В работе были рассмотрены прямодвойственные методы внутренней точки типа «предиктор-корректор» для задач линейного, квадратичного и полуопределённого программирования, а также примеры задач, в которых данные методы применимы. Также был сделан обзор такого важного понятия, как точка шкалирования W на примере задачи полуопределённого программирования, хотя понятно, что из определения можно вывести аналогичный результат и для задач линейного и квадратичного программирования.

Несмотря на то, что для решения каждой задачи понадобилось малое (в пределах 20) количество итераций, каждая итерация достаточно дорого стоит и это не позволяет решать задачи большой размерности ($n \geq 10^6$ для линейного, $10^4 - 10^5$ для квадратичного и 10^3 для полуопределённого программирования), если только в них не будет учитываться дополнительная структура (например, разреженность матриц).

Также, для данного класса методов слабо разработаны критерии остановки и теория сходимости. Большинство практических реализаций включают в себя в том числе и эвристики, позволяющие определить, что метод расходится. Прямодвойственные методы, вообще говоря, расходятся, если задача не имеет решения (множество допустимых решений пусто или неограничено в направлении убывания целевой функции), но могут расходиться в том числе и для «корректных» задач при неудачном выборе начального приближения.

Тем не менее, указанный класс методов на данный момент является единственным универсальным способом для решения задач выпуклой оптимизации, где допустимое множество является симметричным выпуклым конусом. Разработаны также и методы внутренней точки для невыпуклой оптимизации, хотя они на порядок сложнее в анализе и включают гораздо больше эвристик из-за недостаточной развитости теории в этой области.

Список источников

1. Fletcher, R. A Nonlinear Programming Problem in Statistics (Educational Testing) // SIAM Journal on Scientific and Statistical Computing. 1981. Vol. 2 (3). Pp. 257-267. DOI:10.1137/0902021.
2. Nocedal J., Wright S. J. Numerical Optimization. 2006. New York, USA: Springer. DOI:10.1007/978-0-387-40065-5.
3. Todd M., Toh K.-C., Tütüncü R. On the Nesterov-Todd Direction in Semidefinite Programming // SIAM Journal on Optimization.1997. Vol. 8(3). DOI: 10.1137/S105262349630060X.