

Санкт-Петербургский государственный университет

Прикладная математика и информатика

Исследование операций и принятие решений в задачах оптимизации, управления и  
экономики

Кольцов Максим Алексеевич

МЕТОДЫ ПОСТРОЕНИЯ МИНИМАЛЬНОГО ЭЛЛИПСОИДА

Бакалаврская работа

Научный руководитель:

д. ф.-м. н., профессор В. Н. Малозёмов

Рецензент:

к. ф.-м. н., научный сотрудник

М. В. Долгополик

Санкт-Петербург

2016

Saint Petersburg State University  
Applied Mathematics and Computer Science  
Operation Research and Decision Making in Optimisation, Control and Economics  
Problems

Koltsov Maxim Alekseevich

METHODS OF BUILDING MINIMAL ELLIPSOID

Bachelor's Thesis

Scientific Supervisor:  
Professor V. N. Malozemov

Reviewer:  
Research fellow M. V. Dolgopolik

Saint Petersburg  
2016

# Оглавление

<b>Введение</b> . . . . .	4
<b>Глава 1. Постановка задачи</b> . . . . .	5
<b>Глава 2. Задача Сильвестра</b> . . . . .	6
2.1. Постановка задачи . . . . .	6
2.2. Общий вид задачи квадратичного программирования и двойственной к ней . . . . .	7
2.3. Двойственная задача для задачи Сильвестра и её решение . . . . .	8
2.4. Решение задачи в среде MATLAB . . . . .	9
<b>Глава 3. Алгоритм Шора</b> . . . . .	12
3.1. Оператор сжатия пространства . . . . .	12
3.2. Итеративный алгоритм построения минимального эллипсоида . . . . .	14
3.3. Практическое испытание алгоритма . . . . .	18
<b>Глава 4. Алгоритм Хачияна</b> . . . . .	20
4.1. Вспомогательные сведения. . . . .	20
4.2. Вывод двойственной задачи для задачи о минимальном эллипсоиде. . . . .	21
4.3. Алгоритм Хачияна. . . . .	24
<b>Глава 5. Комбинированный алгоритм</b> . . . . .	27
<b>Глава 6. Сводное сравнение методов</b> . . . . .	30
<b>Заключение</b> . . . . .	32
<b>Список литературы</b> . . . . .	33

## Введение

В данной работе рассматривается задача построения эллипсоида минимального объёма, содержащего заданное множество точек в  $\mathbb{R}^n$ . Как её частный случай также рассмотрена задача Сильвестра — задача построения минимального шара.

Решение задачи Сильвестра получается с помощью двойственной задачи квадратичного программирования. Оно реализовано в MATLAB. Продемонстрированы результаты работы программы на тестовых данных.

Для задачи о минимальном эллипсоиде описано три алгоритма, дающих приближённое решение. Все три алгоритма используют вспомогательную задачу в пространстве  $\mathbb{R}^{n+1}$ , которая позволяет искать эллипсоид с фиксированным центром. Представлен вывод формул, по которым восстанавливается решение исходной задачи.

Первый алгоритм, алгоритм Шора, использует оператор сжатия пространства. Оператор сжатия рассмотрен подробно и его работа проиллюстрирована на нескольких примерах с шарами и эллипсами на плоскости.

Вторым был рассмотрен алгоритм Хачияна. Этот алгоритм решает двойственную по Лагранжу задачу, по решению которой однозначно восстанавливается решение исходной. Подробно представлен вывод двойственной задачи и самого алгоритма.

На основе алгоритма Хачияна разработан и описан комбинированный алгоритм, с помощью которого решается двойственная задача. Используется градиентный метод без точного линейного поиска и идея активного множества.

Все три метода реализованы в MATLAB и протестированы на модельных задачах. Приведены сводные таблицы результатов, в которых представлены данные о точности вычислений и затраченном времени. Продемонстрировано превосходство комбинированного метода во всех тестовых случаях.

## Глава 1

## Постановка задачи

Основной темой работы является задача о построении минимального эллипсоида: дано множество точек  $a_i \in \mathbb{R}^n, i \in 1 : m$ , требуется найти эллипсоид минимального объёма, который содержит все точки. Для существования решения будем предполагать, что аффинная оболочка множества  $\{a_i\}$  совпадает с  $\mathbb{R}^n$ .

Эллипсоид будем задавать таким образом:

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid \langle M(x - c), x - c \rangle \leq 1\},$$

где  $M$  — симметричная положительно-определённая матрица,  $c$  — центр эллипсоида. Объём вычисляется по формуле

$$\text{vol}(\mathcal{E}) = \omega_n (\det M)^{-1/2},$$

где  $\omega_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)}$  — объём единичного шара в  $n$ -мерном пространстве.

Пренебрегая множителем  $\omega_n$  и переходя к логарифму целевой функции, задачу о минимальном эллипсоиде можно формально поставить как

$$\begin{aligned} f(M, c) &:= -\ln \det M \rightarrow \min \\ \langle M(a_i - c), a_i - c \rangle &\leq 1, \quad i \in 1 : m \\ M &\succ 0, \end{aligned} \tag{1.1}$$

где запись  $M \succ 0$  означает положительную определённость матрицы  $M$ .

Основным пространством задачи является пространство симметричных положительно определённых матриц, на котором задана векторная норма — корень из суммы квадратов элементов матрицы. Таким образом, в задаче (1.1) имеется  $\frac{n(n+1)}{2} + n$  независимых переменных. Отдельную сложность представляет то, что в задаче требуется найти и матрицу эллипсоида, и его центр. В работе будет представлен способ, позволяющий искать только матрицу при фиксированном центре, решая задачу с  $\frac{(n+1)(n+2)}{2}$  независимыми переменными. Также будет представлен способ перехода к задаче, в которой содержится  $m$  независимых переменных.

В работе будет рассмотрено несколько алгоритмов, решающих задачу (1.1) или эквивалентные ей.

## Глава 2

## Задача Сильвестра

## 2.1. Постановка задачи

Рассмотрим предварительно *задачу Сильвестра* — по заданному набору точек  $a_i \in \mathbb{R}^n$ ,  $i \in M = 1 : m$ , найти наименьший по объёму шар, их содержащий. Чтобы формально поставить задачу, введем переменную  $x \in \mathbb{R}^n$  — центр искомого шара. Очевидно, что для того, чтобы шар  $B_{\mathbb{R}^n}(x, r)$  содержал все заданные точки, необходимо, чтобы его радиус  $r$  удовлетворял следующему условию:

$$\forall i \in M \quad \|a_i - x\| \leq r$$

То есть, в задаче Сильвестра требуется найти точку минимума функции  $\max_{i \in M} \{\|a_i - x\|\}$  на  $\mathbb{R}^n$ , при этом радиус искомого минимального шара будет равен значению функции в этой точке. Заметим, что выражение под знаком максимума можно заменить на  $1/2\|a_i - x\|^2$ , что не меняет точку минимума. Теперь можно сказать так: решение задачи Сильвестра — это точка  $x_*$ , являющаяся решением экстремальной задачи

$$\varphi(x) := \max_{i \in M} \{1/2\|a_i - x\|^2\} \rightarrow \min_{x \in \mathbb{R}^n} \quad , \quad (2.1)$$

а радиус соответствующего минимального шара равен  $\sqrt{2\varphi(x_*)}$ . Доказательство существования и единственности такого решения имеется в книге [1].

Задача 2.1 сводится к задаче квадратичного программирования. Раскроем по определению нормы:

$$\varphi(x) = 1/2\|x\|^2 + \max_{i \in M} \{-\langle a_i, x \rangle + 1/2\|a_i\|^2\}$$

Теперь выражение под максимумом линейно по  $x$ , а значит сам максимум можно обозначить за  $t$  и ввести набор ограничений. То есть, исходная задача эквивалентна задаче квадратичного программирования

$$\begin{aligned} & \frac{1}{2} \langle Ex, x \rangle + t \rightarrow \min \\ & \langle a_i, x \rangle + t \geq \frac{1}{2} \|a_i\|^2, \quad i \in M \end{aligned}$$

## 2.2. Общий вид задачи квадратичного программирования и двойственной к ней

Чтобы двигаться дальше, необходимо сначала рассмотреть понятия задачи квадратичного программирования и двойственной задачи. Задача квадратичного программирования — это экстремальная задача вида

$$\begin{aligned}
 Q(x) &= \frac{1}{2} \langle Dx, x \rangle + \langle c, x \rangle \rightarrow \min \\
 A[M_1, N] \times x[N] &\geq b[M_1] \\
 A[M_2, N] \times x[N] &= b[M_2] \\
 x[N_1] &\geq \mathbb{O}[N_1]
 \end{aligned}$$

где  $D = D[N, N]$  — симметричная неотрицательно-определённая матрица. Как и в случае линейного программирования, можно ввести двойственную задачу и получить теорему, связывающую её с исходной задачей — первую теорему двойственности. При этом двойственная задача выглядит так:

$$\begin{aligned}
 -\frac{1}{2} \langle Dv, v \rangle + \langle b, u \rangle &\rightarrow \max \\
 -v[N] \times D[N, N_1] + u[M] \times A[M, N_1] &\leq c[N_1] \\
 -v[N] \times D[N, N_2] + u[M] \times A[M, N_2] &= c[N_2] \\
 u[M_1] &\geq \mathbb{O}[M_1]
 \end{aligned}$$

Для удобства записи двойственной задачи можно использовать таблицу

	$c$	
$v$	$-D$	$-\frac{1}{2}Dv$
$u$	$A$	$b$

Тогда целевая функция получается скалярным умножением первого и последнего столбца таблицы, ограничения — первого и второго, а правые части ограничений стоят в первой строке, причём неравенствами являются ограничения, соответствующие знаковым ограничениям исходной задачи. Знаковые ограничения накладываются на двойственные переменные (компоненты вектора  $u$ ), соответствующие исходным ограничениям-неравенствам.

### 2.3. Двойственная задача для задачи Сильвестра и её решение

Вернёмся к задаче Сильвестра. Из векторов-строк  $a_i$  составим матрицу  $A_0$  и введём стандартные для квадратичной задачи обозначения: матрицу квадратичной формы  $D$ , вектор коэффициентов линейной формы  $c$ , матрицу ограничений  $A$  и столбец правых частей  $b$ . Получаем следующее:

$$D = \begin{pmatrix} & & & 0 \\ & E[N, N] & & \vdots \\ & & & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \quad c = (0 \quad \dots \quad 0 \quad 1) \quad A = \begin{pmatrix} & 1 \\ A_0 & \vdots \\ & 1 \end{pmatrix} \quad b[i] = \frac{1}{2} \|a_i\|^2$$

Переменной в такой задаче является пара  $y = (x, t)$ , а в двойственной вектор  $z = ((v, s), u)$ . Чтобы записать двойственную задачу, воспользуемся таблицей

	0	...	0	1	
$v[N]$				0	
	$-E[N, N]$		$\vdots$	$-\frac{1}{2}v[N]$	
$s$	0	...	0	0	0
$u$		$A_0$	$\vdots$	1	$b$
				1	

Тогда по описанному общему алгоритму имеем

$$g(z) := -\frac{1}{2} \langle v, v \rangle + \langle b, u \rangle \rightarrow \max$$

$$-v + uA_0 = 0$$

$$\sum_{i \in M} u[i] = 1$$

$$u[M] \geq \mathbb{O}[M]$$

Видно, что здесь можно исключить вектор  $v$  и получить задачу минимизации на стандартном симплексе

$$h(u) := \frac{1}{2} \|uA_0\|^2 - \langle b, u \rangle \rightarrow \min$$

$$\sum_{i \in M} u[i] = 1$$

$$u[M] \geq \mathbb{O}[M]$$



Для последней задачи существует оптимальный план  $u_*$ . В [1] доказано, что в таком случае вектор  $x_* = u_* A_0$  является решением задачи Сильвестра. Так как два оптимальных плана пары двойственных задач имеют одно и то же значение целевой функции, то  $\varphi(x_*) = -h(u_*)$  (на последнем шаге целевая функция была умножена на  $-1$ , чтобы перейти от задачи максимизации к задаче минимизации). Значит, радиус искомого минимального шара равен  $\sqrt{-2h(u_*)}$ .

## 2.4. Решение задачи в среде MATLAB

Для удобства решения задачи Сильвестра была написана функция `sylvester`, принимающая единственный аргумент — матрицу  $A$ , строки которой образованы исходным множеством точек. Алгоритм, реализованный в функции, следующий:

1. Вычислить матрицу  $D = AA^T$  и вектор  $b$ , соответствующие двойственной задаче.
2. Вызвать встроенную функцию `quadprog` для решения двойственной задачи и получить её решение — вектор  $u_*$ , и значение целевой функции  $f$  на нём.
3. Вычислить радиус минимального шара по формуле  $r = \sqrt{-2f}$ .

Написанная функция работает с пространствами любых размерностей, но для наглядности она была проверена на размерности 2. Для этого генерировались случайные множества точек, лежащих внутри и на границе круга  $(x - 4)^2 + (y - 3)^2 \leq 4$  и для них решалась задача Сильвестра. Одно из таких решений с количеством точек 100 представлено на рис. 2.1.

Стоит отметить, что уже при попытке решить задачу для 200 точек, мы получаем ошибку:

```
Maximum number of iterations exceeded; increase options.MaxIter.
To continue solving the problem with the current solution as the
starting point, set x0 = x before calling quadprog.
```

Это связано с тем, что по умолчанию количество итераций, которые делает функция `quadprog`, ограничено числом 200. Чтобы решить эту проблему, можно поднять ограничение. При этом оказывается, что количество итераций зависит от количества точек примерно линейно (см рис. 2.2).

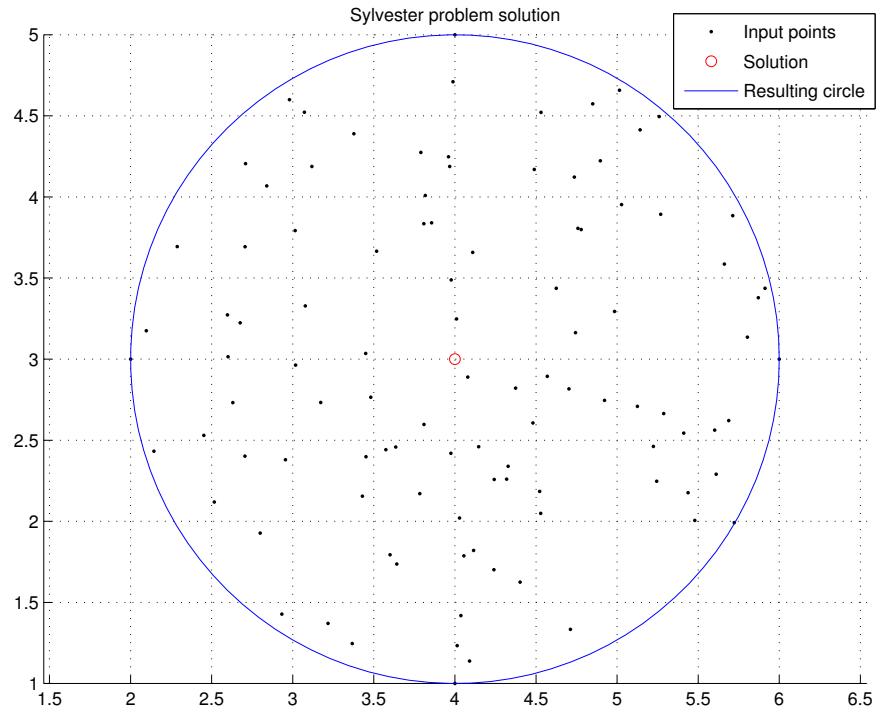


Рис. 2.1. Решение задачи Сильвестра для  $m = 100$

Такой результат можно объяснить тем, что по умолчанию MATLAB выбирает для решения этой задачи алгоритм «active-set». Если же задать использование алгоритма «interior-point-convex», пригодного для решения выпуклых задач, то для 100 точек требуется 6 итераций, а для 500 — всего 9.

Также можно протестировать программу в размерности 3. Для этого было сгенерировано 100 случайных точек внутри и на границе стандартного куба  $[-1, 1]^3$ . На этом наборе точек программа выдает ожидаемый ответ — центр шара близок к 0, а его радиус — к  $\sqrt{3}$ .

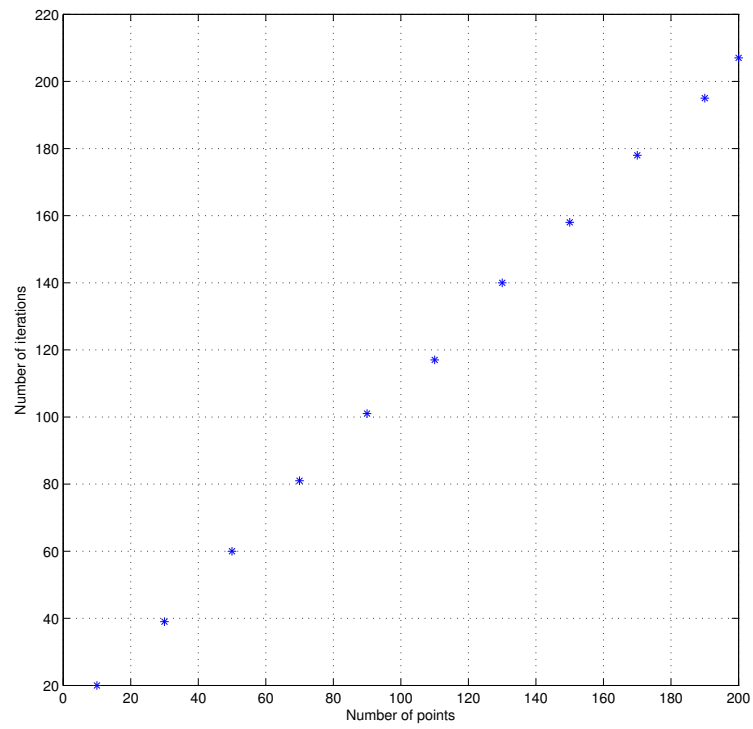


Рис. 2.2. Зависимость числа итераций от числа точек

## Глава 3

## Алгоритм Шора

Перейдём теперь к собственно задаче о минимальном эллипсоиде. В отличие от задачи Сильвестра, которую можно решить за конечное число шагов с помощью квадратичного программирования, эту задачу удаётся решить лишь приближённо итеративными методами. В этом разделе будет рассмотрен алгоритм Шора, строящий приближение к минимальному эллипсоиду.

## 3.1. Оператор сжатия пространства

Алгоритм построения минимального эллипсоида использует оператор сжатия пространства, поэтому рассмотрим сначала его свойства.

**ОПРЕДЕЛЕНИЕ.** Пусть  $\xi \in \mathbb{R}^n$ ,  $\|\xi\| = 1$  и  $\gamma \in (0, 1)$ . Тогда оператор сжатия  $R_\gamma$  определяется формулой

$$R_\gamma = E - (1 - \gamma)\xi\xi^T$$

Здесь  $\xi\xi^T$  — это матрица проектирования на прямую  $x = \lambda\xi$ ,  $\lambda \in \mathbb{R}$ . Вычисление значения  $R_\gamma$  на конкретном векторе  $y \in \mathbb{R}^n$  позволяет понять, как действует этот оператор:

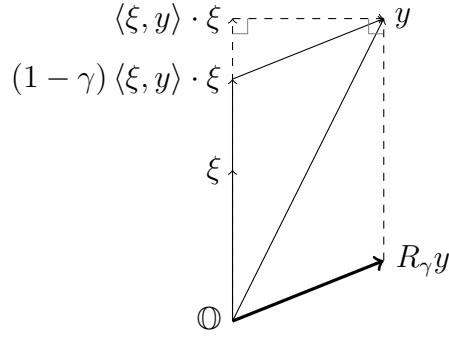
$$R_\gamma y = (E - (1 - \gamma)\xi\xi^T)y = y - (1 - \gamma)\xi\xi^T y = y - (1 - \gamma) \cdot \langle \xi, y \rangle \cdot \xi$$

Из этого равенства и определения  $R_\gamma$  очевидны следующие свойства:

1.  $R_\gamma$  — симметричная матрица
2.  $R_\gamma \xi = \gamma \xi$
3.  $R_\gamma p = p$ , если  $\langle \xi, p \rangle = 0$

Значит,  $R_\gamma$  имеет собственное число  $\lambda_0 = \gamma$  кратности 1 с собственным вектором  $\xi$  и собственное число  $\lambda_1 = 1$  кратности  $n - 1$  с собственным подпространством, ортогональным  $\xi$ . Если  $y = \alpha\xi + p$ , где  $\langle \xi, p \rangle = 0$ , то  $R_\gamma y = \gamma\alpha\xi + p$ .

Кроме этого, матрица  $R_\gamma$  положительно определена при всех  $\gamma \in (0, 1)$ .

Рис. 3.1. Оператор сжатия с  $\gamma = 0.2$ 

**ЛЕММА 1.** Оператор  $R_\gamma$  обратим и обратный к нему оператор задаётся формулой

$$R_\gamma^{-1} = E + \frac{1 - \gamma}{\gamma} \xi \xi^T$$

*Доказательство.* Вычислим произведение  $R_\gamma \cdot R_\gamma^{-1}$ :

$$\begin{aligned} & (E - (1 - \gamma) \xi \xi^T) \left( E + \frac{1 - \gamma}{\gamma} \xi \xi^T \right) = \\ & = E + \frac{1 - \gamma}{\gamma} \xi \xi^T - (1 - \gamma) \xi \xi^T - \frac{(1 - \gamma)^2}{\gamma} \underbrace{\xi \xi^T \xi \xi^T}_{=1} = \\ & = E + \frac{1 - \gamma}{\gamma} (\xi \xi^T - \gamma \xi \xi^T - (1 - \gamma) \xi \xi^T) = E \end{aligned}$$

□

К обратному оператору применимы те же рассуждения о собственных числах и собственных векторах, в частности, вектор  $\xi$  собственный с собственным числом  $\frac{1}{\gamma}$ .

Теперь поймём, как оператор сжатия действует на шары и эллипсоиды в  $\mathbb{R}^n$ . Пусть, для начала, имеется шар с центром в точке  $c$  и радиусом  $r$ , который задан уравнением

$$\langle x - c, x - c \rangle \leq r^2$$

Подействуем на него оператором  $R_\gamma$ , получив новые точки  $y = R_\gamma x$  и новый центр  $a = R_\gamma c$ . Так как  $R_\gamma$  обратим, то  $x = R_\gamma^{-1} y$  и  $c = R_\gamma^{-1} a$ . Подставим эти выражения в уравнение шара

$$\begin{aligned} & \langle x - c, x - c \rangle \leq r^2 \\ & \langle R_\gamma^{-1}(y - a), R_\gamma^{-1}(y - a) \rangle \leq r^2 \\ & \langle (R_\gamma^{-2})(y - a), y - a \rangle \leq r^2 \\ & \left\langle \frac{R_\gamma^{-2}}{r^2}(y - a), y - a \right\rangle \leq 1 \end{aligned}$$

Получилось уравнение эллипсоида с центром в точке  $a = R_\gamma c$  и матрицей  $M := \frac{R_\gamma^{-2}}{r^2}$ . Как было установлено выше, матрица  $M$  имеет собственный вектор  $\xi$ , которому соответствует собственное число  $\frac{1}{r^2\gamma^2}$ , а остальные собственные векторы ортогональны  $\xi$  и имеют собственные числа  $\frac{1}{r^2}$ . Таким образом, оператор сжатия превращает шар радиуса  $r$  в эллипсоид, одной из полуосей которого является  $\xi$  с длиной  $r\gamma$ , а длины остальных полуосей равны  $r$  — шар «сплющивается» в направлении  $\xi$ .

Разобравшись с шаром, подействуем теперь оператором  $R_\gamma$  на эллипсоид, заданный уравнением

$$\langle M(x - c), x - c \rangle \leq 1$$

Положим аналогично  $x = R_\gamma^{-1}y$ ,  $c = R_\gamma^{-1}a$  и получим новое уравнение эллипсоида в виде

$$\langle R_\gamma^{-1}MR_\gamma^{-1}(y - a), y - a \rangle \leq 1$$

Наглядное представление о работе оператора сжатия даёт рис. 3.2.

### 3.2. Итеративный алгоритм построения минимального эллипсоида

Для задачи о минимальном эллипсоиде известен итеративный алгоритм, строящий последовательные приближения к ответу. Этот алгоритм, описанный Н.З. Шором в статье [2], основан на достаточно простых идеях.

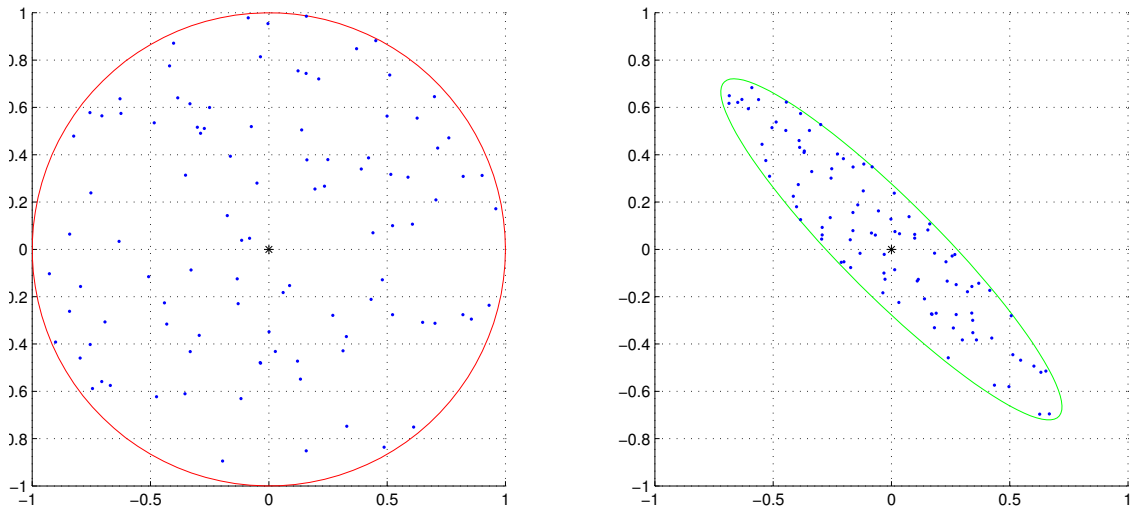
Введём вспомогательную задачу: вложим множество точек  $a_i$  в гиперплоскость  $x_{n+1} = 1$  пространства  $\mathbb{R}^{n+1}$ . Обозначим  $q_i = \begin{pmatrix} a_i \\ 1 \end{pmatrix}$ . В новом пространстве будем искать минимальный эллипсоид с фиксированным центром в нуле. В статье [2] утверждается, что сечение такого эллипсоида плоскостью  $x_{n+1} = 1$  и будет решением исходной задачи (см рис. 3.3).

Поставим формально вспомогательную задачу в терминах матрицы  $\tilde{B}$  порядка  $n + 1$ :

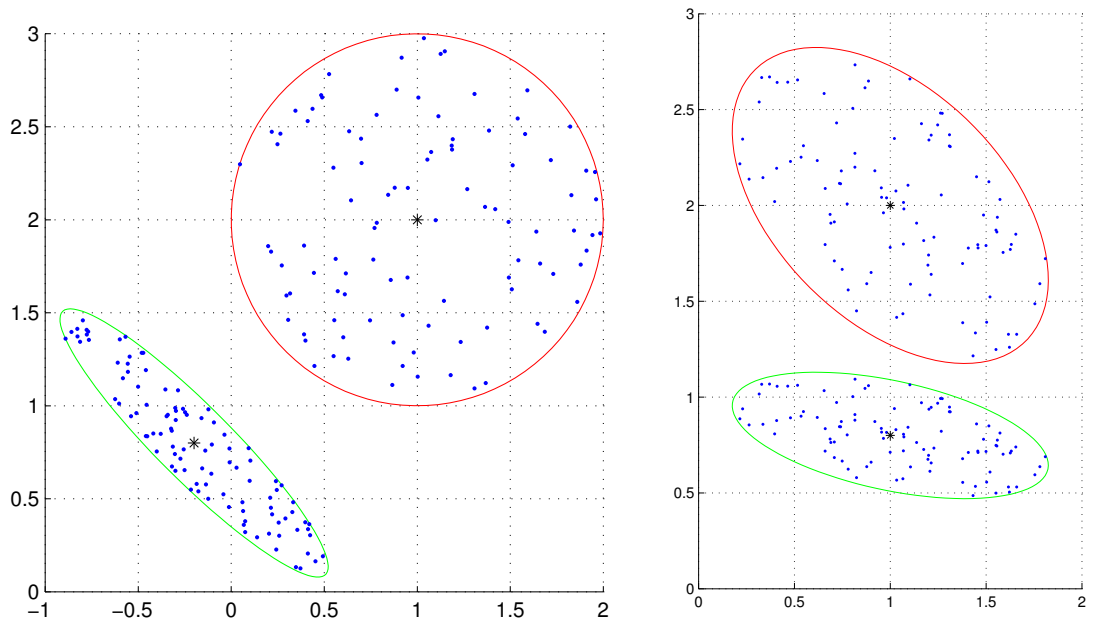
$$\begin{aligned} f(\tilde{B}) &:= -\ln \det \tilde{B} \rightarrow \min \\ \langle \tilde{B}q_i, q_i \rangle &\leq 1, \quad i \in 1 : m \\ \tilde{B} &\succ 0. \end{aligned} \tag{3.1}$$

Рассмотрим вопрос построения сечения подробнее. Итак, пусть

$$\tilde{\mathcal{E}} = \left\{ \tilde{x} \in \mathbb{R}^{n+1} \mid \langle \tilde{B}\tilde{x}, \tilde{x} \rangle \leq 1 \right\}$$



(а) Простейший случай — сжатие единичного круга в направлении  $(1, 1)$  с  $\gamma = 0.2$



(b) Сжатие круга в направлении  $(1, 1)$  с  $\gamma = 0.2$

(с) Сжатие эллипса в направлении  $(0, 1)$  с  $\gamma = 0.4$

Рис. 3.2. Действие оператора сжатия на множество, выделенное красной линией. Результат выделен зелёным цветом

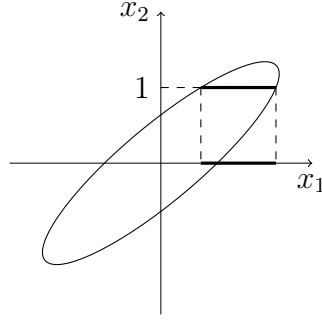


Рис. 3.3. Сечение эллипса с центром в нуле прямой  $x_2 = 1$

— минимальный эллипсоид, содержащий точки  $q_i$ ,  $i \in 1 : m$ . Возьмём две различные точки  $q_{i_0}$  и  $q_{i_1}$ . Отметим, что

$$\langle \tilde{B}(q_{i_0} + q_{i_1}), q_{i_0} + q_{i_1} \rangle + \langle \tilde{B}(q_{i_0} - q_{i_1}), q_{i_0} - q_{i_1} \rangle = 2 \langle \tilde{B}q_{i_0}, q_{i_0} \rangle + 2 \langle \tilde{B}q_{i_1}, q_{i_1} \rangle \quad (3.2)$$

Обозначим

$$\tilde{x}_0 = \frac{1}{2}(q_{i_0} + q_{i_1}) = \begin{pmatrix} x_0 \\ 1 \end{pmatrix}, \quad \tilde{x}_1 = \frac{1}{2}(q_{i_0} - q_{i_1}) = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}$$

Поделив равенство (3.2) на 4, получим

$$\langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle = \frac{1}{2} \langle \tilde{B}q_{i_0}, q_{i_0} \rangle + \frac{1}{2} \langle \tilde{B}q_{i_1}, q_{i_1} \rangle - \langle \tilde{B}\tilde{x}_1, \tilde{B}\tilde{x}_1 \rangle$$

По определению точки  $q_{i_0}$  и  $q_{i_1}$  принадлежат  $\tilde{\mathcal{E}}$  и  $\tilde{x}_1 \neq \mathbb{O}$ , поэтому

$$\langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle < 1 \quad (3.3)$$

Представим матрицу  $\tilde{B}$  в виде

$$\tilde{B} = \begin{pmatrix} B & b \\ b^T & \hat{b} \end{pmatrix}$$

где  $B$  — главный минор порядка  $n$ . Очевидно, что  $B$  симметрична и положительно определена (по критерию Сильвестра). Аккуратно раскроем скобки в (3.3):

$$\begin{aligned} \langle \tilde{B}\tilde{x}_0, \tilde{x}_0 \rangle &= \left\langle \begin{pmatrix} Bx_0 + b \\ \langle b, x_0 \rangle + \hat{b} \end{pmatrix}, \begin{pmatrix} x_0 \\ 1 \end{pmatrix} \right\rangle = \langle Bx_0 + b, x_0 \rangle + \langle b, x_0 \rangle + \hat{b} = \\ &= \langle B(x_0 + B^{-1}b), x_0 + B^{-1}b \rangle - \langle B(x_0 + B^{-1}b), B^{-1}b \rangle + \langle b, x_0 \rangle + \hat{b} = \\ &= \langle B(x_0 + B^{-1}b), x_0 + B^{-1}b \rangle - \langle B^{-1}b, b \rangle + \hat{b} < 1 \end{aligned} \quad (3.4)$$

**ЛЕММА 2.** *Справедливы неравенства*

$$0 < -\langle B^{-1}b, b \rangle + \hat{b} < 1$$



*Доказательство.* Пусть, во-первых,  $\tilde{x} = \begin{pmatrix} -B^{-1}b \\ 1 \end{pmatrix} \neq \mathbb{O}$ . Матрица  $\tilde{B}$  положительно определена, так что  $\langle \tilde{B}\tilde{x}, \tilde{x} \rangle > 0$ . Рассуждения, аналогичные (3.4), приводят к равенству  $\langle \tilde{B}\tilde{x}, \tilde{x} \rangle = -\langle B^{-1}b, b \rangle + \hat{b}$ , что доказывает левое неравенство.

Для доказательства правого неравенства заметим, что в неравенстве (3.4)  $\langle B(x + B^{-1}b), x + B^{-1}b \rangle \geq 0$  в силу положительной определённости  $B$ . Тогда имеем

$$1 > \langle B(x + B^{-1}b), x + B^{-1}b \rangle - \langle B^{-1}b, b \rangle + \hat{b} \geq -\langle B^{-1}b, b \rangle + \hat{b}$$

Неравенство доказано. □

Теперь последнее неравенство из (3.4) поделим на положительное число  $1 + \langle B^{-1}b, b \rangle - \hat{b}$ , введём обозначения

$$c := -B^{-1}b, \quad M := B / (1 + \langle B^{-1}b, b \rangle - \hat{b}) \quad (3.5)$$

и окончательно получим

$$\langle M(x - c), x - c \rangle \leq 1 \text{ — уравнение минимального эллипсоида в } \mathbb{R}^n$$

Алгоритм решения задачи (3.1) устроен следующим образом. Текущее множество точек  $q_i^{(k)}$  хранится в виде столбцов матрицы  $X_k$ , текущая матрица последовательности операторов сжатия обозначается  $A_k$ . Изначально  $X_0$  состоит из исходных точек  $q_i \in \mathbb{R}^{n+1}$ , а  $A_0 = E_{n+1}$ . Один шаг работы алгоритма состоит из нескольких простых действий:

- Найти среди векторов  $q_i^{(k)}$  вектор  $q_{i_k}^{(k)}$  с максимальной нормой  $\tau_{k+1} = \|q_{i_k}^{(k)}\|$
- Вычислить единичный вектор направления сжатия  $\xi_{k+1} = \frac{q_{i_0}^{(k)}}{\tau_{k+1}}$
- Построить оператор сжатия  $R_{k+1}$  вдоль вектора  $\xi_{k+1}$  с коэффициентом  $\alpha_{k+1}$
- Вычислить новые точки  $q_i^{(k+1)}$ , умножив матрицу  $R_{k+1}$  на матрицу  $X_k$ , и сохранить результат в  $X_{k+1}$
- Добавить матрицу  $R_{k+1}$  к последовательности операторов сжатия, умножив её на матрицу  $A_k$ , и сохранить результат в  $A_{k+1}$

Коэффициенты  $\alpha_k$  в соответствии со статьёй [2] выбираются из условий

$$\alpha_k = 1 - \beta_k, \quad \beta_k > 0, \quad \beta_k \xrightarrow[k \rightarrow \infty]{} 0, \quad \sum_{k=1}^{\infty} \beta_k = \infty$$

Пусть вычисления остановлены после шага  $m$ . Тогда из описания алгоритма следует, что

$$X_m = R_m \cdot R_{m-1} \cdots R_1 \cdot X_0 = A_m \cdot X_0$$

Также известно, что точки  $q_i^{(m)}$  лежат внутри сферы с радиусом  $\tau_m$  и центром в начале координат. С учётом этих двух фактов можно получить матрицу искомого эллипсоида:

$$\begin{aligned} \langle q_i^{(m)}, q_i^{(m)} \rangle &\leq \tau_m^2 \\ \langle A_m q_i, A_m q_i \rangle &\leq \tau_m^2 \\ \langle A_m^T A_m q_i, q_i \rangle &\leq \tau_m^2 \\ \left\langle \frac{A_m^T A_m}{\tau_m^2} q_i, q_i \right\rangle &\leq 1 \end{aligned}$$

Матрица  $\tilde{B} := A_m^T A_m / \tau_m^2$  симметрична, положительно определена и является (приближённым) решением вспомогательной задачи о минимальном эллипсоиде. Окончательное решение основной задачи получается по формулам (3.5).

### 3.3. Практическое испытание алгоритма

Алгоритм был реализован в среде MATLAB и испытан на тестовых данных. Вычисления производились в два этапа: пока нормы  $\tau_k$  меняются больше, чем на  $\varepsilon_1$ , использовался постоянный коэффициент  $\beta_k = 0.2$ . На втором этапе использовались убывающие коэффициенты  $\beta_k = 1/(k+1)$ , пока нормы  $\tau_k$  меняются больше, чем на  $\varepsilon_2$ . В качестве тестовых данных было сгенерировано множество случайных точек внутри эллипсоида с центром в точке  $(1, 2)$  с полуосями  $2, 1$ , который вытянут в направлении  $(-1, 1)$ . Результаты вычисления при различных  $\varepsilon_1$  и  $\varepsilon_2$  приведены в таблице 3.1. В столбце «точность» содержится относительная ошибка  $\delta$ , то есть

$$\delta = \frac{|(\det M)^{-1/2} - (\det M^*)^{-1/2}|}{(\det M^*)^{-1/2}},$$

где  $M^*$  — известная матрица минимального эллипсоида.

Как видно из таблицы, при  $\varepsilon_1 = 10^{-6}$  и  $\varepsilon_2 = 10^{-10}$  достигается наилучшее время работы, но наихудшая точность. Подобрать параметры  $\varepsilon_1, \varepsilon_2$  так, чтобы обеспечить и высокую точность, и низкое время работы, не удалось.

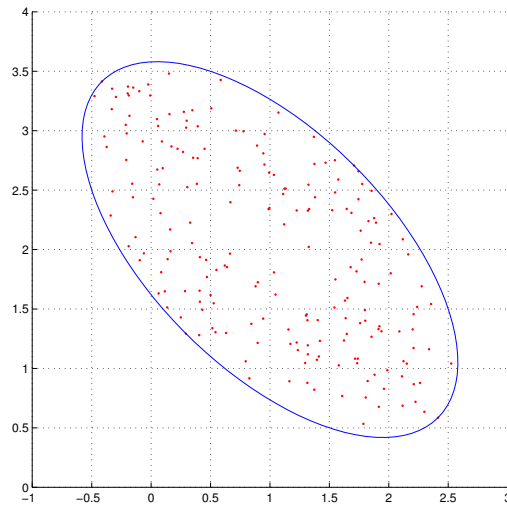


Рис. 3.4. Тестовое множество точек. Синим отмечена граница минимального эллипсоида

Таблица 3.1. Результаты алгоритма Шора при разных  $\varepsilon_1$  и  $\varepsilon_2$ .

$\varepsilon_1$	$\varepsilon_2$	точность	число итераций	время (с)
$10^{-5}$	$10^{-10}$	$7 \cdot 10^{-5}$	21990	2.4
$10^{-4}$	$10^{-9}$	$8 \cdot 10^{-5}$	20591	2.3
$10^{-6}$	$10^{-10}$	$4 \cdot 10^{-4}$	4121	0.4
$10^{-4}$	$10^{-10}$	$2 \cdot 10^{-5}$	100000	21.2

Кроме того, в таблице 3.2 приведены результаты вычисления при  $\varepsilon_1 = 10^{-6}$ ,  $\varepsilon_2 = 10^{-10}$  и различных комбинациях  $n$  и  $m$ .

Таблица 3.2. Результаты алгоритма Шора при разных  $n$  и  $m$ .

$n$	$m$	число итераций	точность	время (с)
2	104	4121	$4 \cdot 10^{-4}$	0.2
2	504	4118	$5 \cdot 10^{-4}$	0.2
5	510	16729	$8 \cdot 10^{-4}$	1.1
10	1020	3533	$3 \cdot 10^{-3}$	0.6

## Глава 4

## Алгоритм Хачияна

Как видно из таблиц, алгоритм Шора не достигает высоких результатов по точности полученного эллипсоида. Но, несмотря на это, в нём содержится удачная идея — решение задачи в расширенном пространстве (3.1).

В этой главе будет рассматриваться ещё один алгоритм решения расширенной задачи. Решение исходной задачи будет получаться по формулам (3.5).

## 4.1. Вспомогательные сведения.

Пусть  $q$  — вектор, а  $M$  — симметричная матрица соответствующей размерности. Найдём другое представление для  $q^T M q$ , используя индексную технику:

$$\begin{aligned} q^T M q &= \langle M q, q \rangle = \left\langle \sum_{k=1}^n M[1:n, k] q[k], q \right\rangle = \sum_{k=1}^n q[k] \langle M[1:n, k], q \rangle = \\ &= \sum_{k=1}^n q[k] \left( \sum_{j=1}^n M[j, k] q[j] \right) = \sum_{k=1}^n \sum_{j=1}^n M[j, k] (q[j] q[k]) = \\ &= \sum_{k=1}^n M[j, k] (q q^T)[j, k] = \text{tr}(M \cdot q q^T). \end{aligned} \quad (4.1)$$

Нам понадобится ещё одно равенство, называемое «лемма об определителе матрицы» в англоязычной литературе [3]. Пусть  $A$  — обратимая матрица,  $u, v$  — векторы. Тогда

$$\det(A + uv^T) = (1 + v^T A^{-1} u) \cdot \det A. \quad (4.2)$$

Прежде чем проверить это равенство, заметим, что  $\det(A + uv^T) = \det A \cdot \det(E + (A^{-1}u)v^T)$ . Следовательно, достаточно проверить только случай  $A = E$ . В этом случае необходимое равенство следует из свойств определителя и равенства

$$\begin{pmatrix} E & 0 \\ v^T & 1 \end{pmatrix} \cdot \begin{pmatrix} E + uv^T & u \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} E & 0 \\ -v^T & 1 \end{pmatrix} = \begin{pmatrix} E & u \\ 0 & 1 + v^T u \end{pmatrix}.$$

Действительно, определитель матрицы справа равен  $1 + v^T u$ , а определитель средней матрицы в левой части равен  $\det(E + uv^T)$ . Кроме того, определители остальных матриц равны 1. Значит, равенство доказано.

Рассмотрим теперь общую задачу выпуклого программирования с ограничениями:

$$\begin{aligned} f(x) &\rightarrow \min \\ h_i(x) &\leq 0, \quad i \in 1 : m, \\ x &\in D, \end{aligned}$$

где  $D$  – открытое выпуклое множество. Введём неотрицательные числа  $\lambda_i$ , соответствующие ограничениям задачи. Тогда *функцией Лагранжа* [4] этой задачи называется функция

$$\mathcal{L}(x, \lambda) := f(x) + \sum_{i=1}^m \lambda_i h_i(x),$$

а числа  $\lambda_i$  называются *множителями Лагранжа*.

Используя функцию Лагранжа, можно записать двойственную по Лагранжу задачу:

$$\begin{aligned} g(\lambda) &:= \inf_{x \in D} \mathcal{L}(x, \lambda) \rightarrow \max \\ \lambda_i &\geq 0, \quad i \in 1 : m. \end{aligned}$$

Для любой пары векторов  $x$  и  $\lambda$ , удовлетворяющих прямой и двойственной задачам соответственно, верно  $f(x) \geq g(\lambda)$ . Отсюда, если  $x^*$  и  $\lambda^*$  – решения пары двойственных задач, то

$$f(x^*) \geq g(\lambda^*).$$

## 4.2. Вывод двойственной задачи для задачи о минимальном эллипсоиде.

Попробуем применить двойственность Лагранжа к задаче (3.1). Введём множители Лагранжа  $p_i \geq 0$ , соответствующие ограничениям  $\langle \tilde{B}q_i, q_i \rangle - 1 \leq 0$ , а ограничение  $M \succ 0$  будем считать нефункциональным и включим в область определения  $\mathcal{L}$ . Тогда функция Лагранжа записывается как

$$\mathcal{L}(\tilde{B}, p) := -\ln \det \tilde{B} + \sum_{i=1}^m p_i (q_i^T \tilde{B} q_i - 1).$$

Или, с учётом формулы (4.1),

$$\mathcal{L}(\tilde{B}, p) := -\ln \det \tilde{B} + \sum_{i=1}^m p_i \operatorname{tr}(q_i q_i^T \cdot \tilde{B}) - \sum_{i=1}^m p_i.$$

Здесь и далее будем считать, что  $\operatorname{dom} \mathcal{L} = \{(\tilde{B}, p) \mid \tilde{B} \succ 0, p \in \mathbb{R}^m\}$ .

По определению целевая функция двойственной к (3.1) задачи равна

$$g(p) := \inf_{\tilde{B} \succ 0} \mathcal{L}(\tilde{B}, p) = \inf_{\tilde{B} \succ 0} \left[ -\ln \det \tilde{B} + \sum_{i=1}^m p_i \operatorname{tr}(q_i q_i^\top \cdot \tilde{B}) - \sum_{i=1}^m p_i \right].$$

Для того, чтобы упростить это выражение, найдём инфимум аналитически. Известно (см. [4]), что функции  $-\ln \det \tilde{B}$  и  $\operatorname{tr}(q_i q_i^\top \cdot \tilde{B})$  выпуклы по  $\tilde{B}$  на открытом множестве положительно определённых матриц. Значит, инфимум достигается в точке, где производная по  $\tilde{B}$  выражения под ним равна нулю. Возьмём эту производную:

$$\frac{\partial \mathcal{L}(\tilde{B}, p)}{\partial \tilde{B}} = -\tilde{B}^{-1} + \sum_{i=1}^m p_i q_i q_i^\top = -\tilde{B}^{-1} + QPQ^\top,$$

где  $Q$  — матрица, составленная из точек  $q_i$ , а  $P = \operatorname{diag}(p)$ . Для дальнейших рассуждений необходимо, чтобы матрица  $QPQ^\top$  была положительно определена. Проверим, так ли это.

Зафиксируем произвольный ненулевой вектор  $x \in \mathbb{R}^{n+1}$ . По определению, для положительной определённости  $QPQ^\top$  необходимо, чтобы  $\langle QPQ^\top x, x \rangle > 0$ . Распишем скалярное произведение:

$$\left\langle \sum_{i=1}^m p_i q_i q_i^\top \cdot x, x \right\rangle = \sum_{i=1}^m p_i \langle q_i q_i^\top \cdot x, x \rangle = \sum_{i=1}^m p_i \langle q_i, x \rangle^2$$

Так как среди точек  $q_i$  содержится аффинно независимое множество, то вектор  $x$  не может быть ортогонален сразу всем векторам  $q_i$ . Значит, если для всех  $i$   $p_i > 0$ , то матрица  $QPQ^\top$  заведомо положительно определена. На самом деле, достаточно, чтобы числа  $p_i$  были положительны на индексах  $i$ , соответствующих аффинно независимому множеству точек  $q_i$ . Будем теперь рассматривать только такие  $p$ , что  $QPQ^\top \succ 0$ . В следующей части будет описан алгоритм, на каждом шаге которого выполнено условие  $p > 0$ .

Таким образом, единственным решением уравнения  $\frac{\partial \mathcal{L}(\tilde{B}, p)}{\partial \tilde{B}} = 0$  при фиксированном  $p$  является матрица  $\tilde{B} = (QPQ^\top)^{-1}$ . Подставим это значение в  $\mathcal{L}(\tilde{B}, p)$  и получим выражение для целевой функции двойственной задачи  $g(p)$ :

$$g(p) = \ln \det QPQ^\top + \sum_{i=1}^m p_i \operatorname{tr} [q_i^\top q_i \cdot (QPQ^\top)^{-1}] - \sum_{i=1}^m p_i.$$

Обозначим  $A = QPQ^\top$  и распишем выражение под первой суммой:

$$\sum_{i=1}^m p_i \operatorname{tr} [q_i q_i^\top \cdot (QPQ^\top)^{-1}] = \operatorname{tr} \left( \sum_{i=1}^m p_i q_i q_i^\top \right) A^{-1} = \operatorname{tr} AA^{-1} = n + 1 \quad (4.3)$$

В итоге имеем

$$g(p) = \ln \det QPQ^\top - \sum_{i=1}^m p_i + n + 1 \quad (4.4)$$

Значит, двойственная задача имеет вид

$$g(p) := \ln \det QPQ^\top - \sum_{i=1}^m p_i + n + 1 \rightarrow \max \quad (4.5)$$

$$p_i \geq 0, \quad i \in 1 : m.$$

Чтобы ещё упростить вид целевой функции, запишем для этой задачи условия Куна-Таккера. Чтобы сделать это, посчитаем сначала производную  $\ln \det QPQ^\top$ , аккуратно применив правило цепочки.

$$\frac{\partial}{\partial p_i} \ln \det QPQ^\top = \text{tr} \left( \frac{\partial \ln \det QPQ^\top}{\partial QPQ^\top} \cdot \frac{\partial QPQ^\top}{\partial p_i} \right) = \text{tr} \left( (QPQ^\top)^{-1} \cdot q_i q_i^\top \right) = q_i^\top (QPQ^\top)^{-1} q_i$$

Следовательно, условия Куна-Таккера для (4.5) с множителями  $\lambda_i$ , соответствующими ограничениям  $p_i \geq 0$ , выглядят как

$$q_i^\top (QPQ^\top)^{-1} q_i - 1 + \lambda_i = 0$$

$$\lambda_i p_i = 0$$

$$\lambda_i \geq 0$$

Избавимся от переменных  $\lambda_i$ :

$$q_i^\top (QPQ^\top)^{-1} q_i \leq 1 \quad (4.6)$$

$$p_i (1 - q_i^\top (QPQ^\top)^{-1} q_i) = 0 \quad (4.7)$$

Просуммировав (4.7) по  $i$  и воспользовавшись ещё раз формулой (4.3) получаем

$$\sum_{i=1}^m p_i = n + 1.$$

Теперь двойственную задачу можно записать как

$$g(p) := \ln \det QPQ^\top \rightarrow \max$$

$$1^\top p = n + 1$$

$$p \geq 0.$$

С помощью замены переменных  $u := \frac{p}{n+1}$ ,  $U := \text{diag}(u) = \frac{1}{n+1} P$  получаем итоговый вид двойственной задачи

$$\widehat{g}(u) := \ln \det QUQ^\top \rightarrow \max$$

$$1^\top u = 1 \quad (4.8)$$

$$u \geq 0.$$

Таким образом, как и в задаче Сильвестра, мы свели нахождение минимального эллипсоида к максимизации функции на стандартном симплексе в  $\mathbb{R}^m$ .

Пусть найдено решение двойственной задачи  $u^*$ . Из того, что в точке  $u^*$  достигается максимум функции  $\hat{g}$ , следует, что  $\hat{g}(u^*) > -\infty$ , а значит  $QU^*Q^\top$  положительно определена. Положим  $p^* = (n+1)u^*$ ,  $\tilde{B}^* = (n+1) \cdot (QU^*Q^\top)^{-1}$ . В силу (4.6) эта матрица будет удовлетворять ограничениям задачи (3.1). Очевидно, что выполняется равенство  $f(\tilde{B}^*) = g(p^*)$ . В силу неравенства между целевыми функциями пары двойственных задач, для любой матрицы  $\tilde{B}$ , удовлетворяющей ограничениям, верно  $f(\tilde{B}) \geq g(p^*) = f(\tilde{B}^*)$ , следовательно,  $\tilde{B}^*$  — решение задачи (3.1).

### 4.3. Алгоритм Хачияна.

В статье [5] описан простой алгоритм, решающий двойственную задачу (4.8). Приведём его вывод.

Во-первых, заметим, что формула (4.6) для изменённой задачи (4.8) имеет вид

$$q_i^\top (QUQ^\top)^{-1} q_i \leq n+1$$

Как было вычислено выше,  $\hat{g}_j(u) := \frac{\partial}{\partial u_j} \hat{g}(u) = q_j^\top (QUQ^\top)^{-1} q_j$ . Значит, если  $u^*$  — решение, то

$$\hat{g}_j(u^*) \leq n+1, \quad \forall j \in 1:m. \quad (4.9)$$

Пусть на  $k$ -м шаге имеется вектор  $u_k$ . Найдём, с какой ошибкой  $u_k$  удовлетворяет условию (4.9). То есть, найдём такое минимальное число  $\varepsilon$ , что

$$\hat{g}_j(u_k) \leq (1+\varepsilon)(n+1), \quad \forall j \in 1:m.$$

Пусть  $R \subset 1:m$  — множество индексов, на которых неравенство обращается в равенство. Тогда для  $j \notin R$

$$\hat{g}_j(u_k) < (1+\varepsilon)(n+1) = \hat{g}_r(u_k), \quad \forall r \in R.$$

Выберем какой-нибудь  $r \in R$ . Из формулы видно, что  $\hat{g}_r(u_k) = \max \{\hat{g}_j(u_k) | j \in 1:m\}$  и

$$\varepsilon = \frac{\hat{g}_r(u_k) - (n+1)}{n+1}.$$

Хачиян предлагает следующее: в качестве  $u_{k+1}$  выбрать точку из отрезка  $[u_k, e_r]$ , где  $e_r$  —  $r$ -й орт  $\mathbb{R}^m$ , максимизирующую  $\hat{g}(u_{k+1})$ . То есть, выбрать число  $\alpha \in (0,1)$  и положить

$$u_{k+1} = (1-\alpha)u_k + \alpha e_r.$$



Покажем, как найти такое  $\alpha$ . Для этого распишем  $\widehat{g}(u_{k+1})$ :

$$\begin{aligned}\widehat{g}(u_{k+1}) &= \ln \det [Q((1-\alpha)U_k + \alpha E_r)Q^\top] = \ln \det [(1-\alpha)QU_kQ^\top + \alpha q_r q_r^\top] = \\ &= \ln \left[ (1-\alpha)^{n+1} \det \left( QU_kQ^\top + \frac{\alpha}{1-\alpha} q_r q_r^\top \right) \right].\end{aligned}$$

Применим формулу (4.2):

$$\det \left( QU_kQ^\top + \frac{\alpha}{1-\alpha} q_r q_r^\top \right) = \left( 1 + \frac{\alpha}{1-\alpha} q_r^\top (QU_kQ^\top)^{-1} q_r \right) \det QU_kQ^\top.$$

Таким образом,

$$\begin{aligned}\widehat{g}(u_{k+1}) &= \ln \left[ (1-\alpha)^{n+1} \left( 1 + \frac{\alpha}{1-\alpha} q_r^\top (QU_kQ^\top)^{-1} q_r \right) \det QU_kQ^\top \right] = \\ &= \ln [(1-\alpha)^n (1 + \alpha(\widehat{g}_r(u_k) - 1)) \det QU_kQ^\top] = \\ &= n \ln(1-\alpha) + \ln(1 + \alpha(\widehat{g}_r(u_k) - 1)) + \ln \det QU_kQ^\top.\end{aligned}$$

Возьмём производную по  $\alpha$ :

$$\frac{d\widehat{g}(u_{k+1})}{d\alpha} = -\frac{n}{1-\alpha} + \frac{\widehat{g}_r(u_k) - 1}{1 + \alpha(\widehat{g}_r(u_k) - 1)}.$$

Приравнивая нулю и решая относительно  $\alpha$ , получаем единственное решение

$$\alpha = \frac{\widehat{g}_r(u_k) - (n+1)}{(n+1)(\widehat{g}_r(u_k) - 1)}.$$

Так как  $\widehat{g}_r(u_k) = (1+\varepsilon)(n+1) > n+1 > 1$ , то  $\alpha > 0$ . Кроме того,

$$\alpha < 1 \Leftrightarrow \widehat{g}_r(u_k) - n - 1 < n \cdot \widehat{g}_r(u_k) - n + \widehat{g}_r(u_k) - 1 \Leftrightarrow \widehat{g}_r(u_k) > 0.$$

Значит,  $\alpha$  действительно лежит в  $(0, 1)$ . Так как  $\widehat{g}(u_{k+1})$  выпукла по  $\alpha$ ,  $u_{k+1}$  — искомая точка максимума на отрезке  $(0, 1)$ .

В качестве начального приближения  $u_0$  можно взять вектор  $(1, \dots, 1)/m$ . Так как  $u_0 > 0$ , то  $QU_0Q^\top \succ 0$  и целевая функция  $\widehat{g}(u_0)$  корректно определена. Кроме того, на каждом шаге  $\alpha_k < 1$ , значит если  $u_k > 0$ , то и  $u_{k+1} > 0$ . Таким образом, алгоритм корректный.

Опишем ещё раз последовательность действий на  $k$ -м шаге алгоритма:

- Вычислить  $\nabla \widehat{g}(u_k)$
- Найти максимальную компоненту градиента  $\widehat{g}_r(u_k)$

- Вычислить  $\varepsilon = \frac{\widehat{g}_r(u_k) - (n+1)}{n+1}$
- Если  $\varepsilon$  меньше требуемой погрешности  $\varepsilon_0$ , закончить вычисления. В противном случае, вычислить  $\alpha := \frac{\widehat{g}_r(u_k) - (n+1)}{(n+1)(\widehat{g}_r(u_k) - 1)}$
- Положить  $u_{k+1} := (1 - \alpha)u_k + \alpha e_r$  и перейти на следующий шаг

Кроме того, заметим, что на каждом шаге матрица  $QUQ^\top$  умножается на  $1 - \alpha$  и к ней прибавляется одноранговая матрица  $\alpha q_r q_r^\top$ . В таком случае нет необходимости каждый раз заново вычислять обратную матрицу, а можно применить формулу обновления (см. [5]):

$$(QU_{k+1}Q^\top)^{-1} = \left(1 + \frac{\varepsilon}{n \cdot (1 + \varepsilon)}\right) (QU_kQ^\top)^{-1} - \frac{\varepsilon}{n \cdot (1 + \varepsilon)^2} b_k b_k^\top,$$

где  $b_k := (QU_kQ^\top)^{-1} q_r$ .

Были проведены численные эксперименты при  $\varepsilon_0 = 10^{-5}$  и различных размерностях основного пространства  $n$  и различном количестве двойственных переменных  $m$  (то есть количестве точек  $q_i$ ). Результаты приведены в таблице 4.1.

Таблица 4.1. Численные результаты алгоритма Хачияна

$n$	$m$	число итераций	точность	время (с)
2	104	199993	$7 \cdot 10^{-6}$	17.2
2	504	199994	$7 \cdot 10^{-6}$	78.8
5	510	499990	$1.4 \cdot 10^{-5}$	261.3

Если в алгоритме критерий останова по  $\varepsilon$  (по точности выполнения ограничений) заменить на останов по внутренней сходимости, то число итераций и время работы сократится, но относительная точность не изменится. Отметим, что даже в случае с четырьмя точками в  $\mathbb{R}^2$ , одна из которых лежит внутри треугольника, образованного тремя другими, алгоритм делает порядка 80 тысяч шагов и работает около 4 секунд.

## Глава 5

## Комбинированный алгоритм

Как было продемонстрировано в предыдущей главе, применение одного только алгоритма Хачияна не приносит желаемых результатов из-за низкой точности и большого времени работы. Поэтому были предприняты попытки изменить алгоритм так, чтобы он работал быстрее и точнее.

В ходе численных экспериментов была обнаружена следующая закономерность — в начале своей работы алгоритм делает достаточно большие шаги  $\alpha$ , быстро получая эллипсоид, близкий к искомому. Затем же алгоритм начинает делать очень короткие шаги, с  $\alpha \approx 10^{-5}$ , постепенно увеличивая эллипсоид и приближаясь к решению. Это связано с тем, что на каждом шаге алгоритм производит изменение в направлении только одной координаты, как бы «подтягивая» эллипсоид к точке, которая дальше всего от него находится. Таким образом, если текущий эллипсоид уже достаточно близок к оптимальному, то алгоритму приходится поочерёдно немного подтягивать его к точкам, лежащим вне.

На основе этого наблюдения и статьи [6] возникла идея улучшения алгоритма. Во-первых, в 1948 году Джон доказал, что минимальный эллипсоид определяется не более  $(n^2 + 3n)/2$  точками, где  $n$  — размерность пространства. То есть, в нашем случае только  $l = ((n + 1)^2 + 3(n + 1))/2$  точек из исходных  $m$  определяют искомый эллипсоид. Кроме того, из формулы (4.6) и её эквивалента для вектора  $u$  видно, что те точки  $a_i$ , для которых  $\hat{g}_i(u) \geq n + 1$ , лежат на границе или вне эллипсоида, задаваемого  $u$ . Значит, если их количество не превосходит  $l$ , можно предположить, что только эти точки и задают эллипсоид, который мы ищем. Будем дальше работать уже только с ними. Для корректности задания целевой функции на этих точках необходимо потребовать, чтобы они так же содержали аффинно независимое множество. Таким образом, в «активное» множество всегда будет входить не более  $l$  и не менее  $n + 1$  точек.

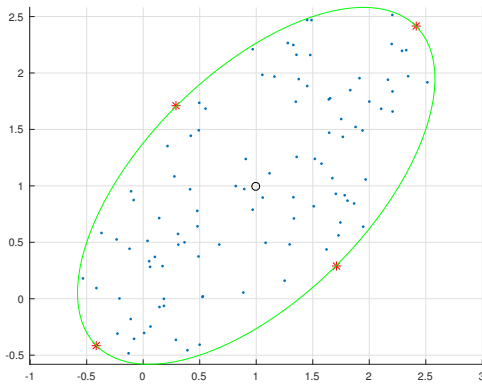
Во-вторых, раз увеличение эллипсоида в направлении только одной точки приводит к очень маленьким шагам, разумно попытаться увеличивать его сразу в нескольких направлениях. В первую очередь для этого был испытан обычный градиентный метод наискорейшего подъёма. В процедуре линейного поиска использовано условие Армихо. В качестве направления подъёма выбирается направление градиента, спроецированное

на стандартный симплекс по алгоритму, описанному в [7]. Кроме того, линейный поиск останавливается при достижении какой-либо компонентой вектора нуля. Таким образом обеспечивается выполнение ограничений двойственной задачи на каждом шаге градиентного метода. Критерий останова используется тот же, что и в алгоритме Хачияна.

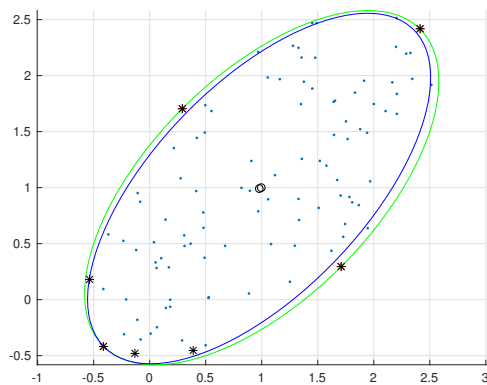
На рисунке 5.1 наглядно представлена работа такого комбинированного подхода, а подробные результаты его применения приведены в таблице 5.1. Здесь столбцы «алгоритм Хачияна» и «градиентный метод» содержат количество шагов соответствующих алгоритмов, в столбце «активные точки» содержится число точек в активном множестве после останова алгоритма Хачияна, а столбец «точность» имеет тот же смысл, что и в предыдущей таблице. Для вычислений использовалась погрешность  $\varepsilon_0 = 10^{-5}$ .

Таблица 5.1. Численные результаты комбинированного алгоритма

$n$	$m$	алгоритм Хачияна	активные точки	градиентный метод	точность	время (с)
2	104	18	7	40	$2 \cdot 10^{-9}$	0.05
2	504	123	9	87	$1.5 \cdot 10^{-8}$	0.18
5	510	92	26	75	$1.5 \cdot 10^{-7}$	0.2
10	1020	80	74	217	$3.6 \cdot 10^{-6}$	0.9



(a) Тестовое множество и известный минимальный эллипсоид (зелёный)



(b) Активное множество и построенный алгоритмом Хачияна эллипсоид (синий)

Рис. 5.1. Иллюстрация работы комбинированного алгоритма при  $n = 2$ ,  $m = 104$ .

К сожалению, в некоторых случаях алгоритм закликивается или выдает неверный ответ. Такое происходит, например, при  $n = 30$ ,  $m = 560$  — в этом случае градиентный метод начиная с 1890-го шага начинает делать циклические шаги, не имея возможности никак остановиться. При этом достигается относительная точность только  $1 \cdot 10^{-4}$ . Это

может быть связано с недостатками процедуры линейного поиска, а также с общей неприменимостью простых градиентных методов вблизи решения.

В дальнейшем планируется применить метод сопряженных градиентов с условием Вулфа, а так же реализовать изменение активного множества между шагами градиентного метода.

## Глава 6

## Сводное сравнение методов

Приведём ещё раз таблицы сравнения трёх методов: алгоритма Шора, алгоритма Хачияна с остановом по внутренней сходимости и комбинированного метода.

Таблица 6.1. Алгоритм Шора

$n$	$m$	число итераций	точность	время (с)
2	104	4121	$4 \cdot 10^{-4}$	0.2
2	504	4118	$5 \cdot 10^{-4}$	0.2
5	510	16729	$8 \cdot 10^{-4}$	1.1
10	1020	3533	$3 \cdot 10^{-3}$	0.6

Таблица 6.2. Алгоритм Хачияна

$n$	$m$	число итераций	точность	время (с)
2	104	86594	$1.8 \cdot 10^{-5}$	6.2
2	504	86595	$1.7 \cdot 10^{-6}$	36.5
5	510	94854	$8 \cdot 10^{-5}$	56.2
10	1020	97451	$2.8 \cdot 10^{-3}$	343.5

Таблица 6.3. Комбинированный алгоритм

$n$	$m$	число итераций	точность	время (с)
2	104	40	$2 \cdot 10^{-9}$	0.05
2	504	87	$1.5 \cdot 10^{-8}$	0.18
5	510	75	$1.5 \cdot 10^{-7}$	0.2
10	1020	217	$3.6 \cdot 10^{-6}$	0.9

Также все три алгоритма были запущены на множестве, минимальным эллипсоидом которого является круг, и их результаты сравнены с результатом программы для решения задачи Сильвестра из первой главы. Данные приведены в таблицах 6.4 и 6.5.

Таблица 6.4. Сравнение алгоритмов — точность

$n$	$m$	задача Сильвестра	алгоритм Шора	алгоритм Хачияна	комбинированный
2	104	$1.4 \cdot 10^{-11}$	$4 \cdot 10^{-4}$	$1.7 \cdot 10^{-5}$	$7.5 \cdot 10^{-9}$
2	504	$1.8 \cdot 10^{-4}$	$3.7 \cdot 10^{-4}$	$1.7 \cdot 10^{-5}$	$1.7 \cdot 10^{-9}$
5	510	$6.6 \cdot 10^{-5}$	$6.6 \cdot 10^{-4}$	$7.9 \cdot 10^{-5}$	$2.8 \cdot 10^{-7}$
10	1020	$1.8 \cdot 10^{-5}$	$1.7 \cdot 10^{-3}$	$2.8 \cdot 10^{-4}$	$9 \cdot 10^{-7}$

Таблица 6.5. Сравнение алгоритмов — время работы (в секундах)

$n$	$m$	задача Сильвестра	алгоритм Шора	алгоритм Хачияна	комбинированный
2	104	0.2	0.3	6.5	0.3
2	504	1.4	0.5	51	0.3
5	510	1.4	1.4	100	0.2
10	1020	8.2	0.7	398	1.4

## Заключение

В рамках данной работы проведён анализ решения задачи о минимальном эллипсоиде с применением теории двойственности по Лагранжу. Рассмотрен также частный случай — задача Сильвестра, для которой получено решение с помощью двойственности в квадратичном программировании.

Одним из результатов работы является разработка комбинированного алгоритма, использующего идею повышения размерности из алгоритма Шора, идею двойственности по Лагранжу из алгоритма Хачияна и градиентный метод без точного линейного поиска.

Все алгоритмы реализованы в MATLAB и проверены на множестве тестовых примеров. Приведены сводные таблицы точности и времени работы всех методов, выполнено сравнение их производительности на задаче Сильвестра, для которой имеется точное решение с помощью квадратичного программирования.

Предварительные результаты были опубликованы в [8, 9, 10, 11].



## Список литературы

1. Гавурин М. К., Малозёмов В. Н. Экстремальные задачи с линейными ограничениями. Л.: Изд-во ЛГУ, 1984. С. 176.
2. Шор Н. З., Стеценко С. И. Алгоритм последовательного сжатия пространства для построения описанного эллипсоида минимального объёма // Исследование методов решения экстремальных задач, ИК АН УССР, Киев. 1990. С. 25–29.
3. Brookes M. The Matrix Reference Manual. URL: <http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html>.
4. Boyd S., Vandenberghe L. Convex Optimization. New York, NY, USA: Cambridge University Press, 2004.
5. Khachiyan L. G. Rounding of Polytopes in the Real Number Model of Computation // Mathematics of Operations Research. 1996. Vol. 21, no. 2. P. 307–320.
6. Sun P., Freund R. M. Computation of Minimum-Volume Covering Ellipsoids // Operations Research. 2004. Vol. 52, no. 5. P. 690–706.
7. Малозёмов В. Н. Проектирование точки на подпространство и на стандартный симплекс. Семинар «DHA & CAGD». Избранные доклады. 28.02.2013. URL: <http://dha.spb.ru/rep13.shtml#0228>.
8. Кольцов М. А. Решение задачи Сильвестра в MATLAB. Семинар «CNSA & NDO». Избранные доклады. 26.02.2015. URL: <http://www.apmath.spbu.ru/cnsa/rep15.shtml#0226>.
9. Кольцов М. А. Построение минимального эллипсоида. Семинар «CNSA & NDO». Избранные доклады. 14.05.2015. URL: <http://www.apmath.spbu.ru/cnsa/rep15.shtml#0514>.
10. Кольцов М. А. Построение минимального эллипсоида: алгоритм Хачияна. Семинар «CNSA & NDO». Избранные доклады. 21.04.2016. URL: <http://www.apmath.spbu.ru/cnsa/rep16.shtml#0421>.
11. Кольцов М. А. Построение минимального эллипсоида // Процессы управления и устойчивость. Т. 3 (19). 2016. (принята в печать).