



Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики  
Кафедра вычислительных технологий и моделирования

Петров Сергей Владимирович

**Проекционные методы решения линейных систем на  
основе случайного выбора подпространств**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Научный руководитель:**  
чл.-корр. РАН, д.ф.-м.н., профессор  
Е.Е. Тыртышников

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Рандомизированные методы</b>	<b>4</b>
2.1	$\epsilon$ - вложения . . . . .	4
2.2	Лемма Джонсона-Линденштраусса . . . . .	5
2.3	Быстрое преобразование JL . . . . .	6
2.4	Рандомизированный метод наименьших квадратов . . . . .	6
2.5	Быстрое вычисление матриц Грама . . . . .	7
<b>3</b>	<b>Задача MIMO-downlink</b>	<b>8</b>
3.1	Введение в задачу MIMO-downlink . . . . .	8
3.2	Известные эвристические алгоритмы . . . . .	9
3.3	Применение рандомизированного вычисления матриц Грама . . . . .	9
3.4	Постановка общей задачи . . . . .	10
3.5	Существующие исследования этой задачи . . . . .	11
3.6	Подзадача fixed-QoS . . . . .	12
3.7	Решение задачи fixed-QoS методом внутренней точки . . . . .	14
3.8	Модификация метода и аппроксимация допустимой области . . . . .	18
3.9	Метод ветвей и границ для общей задачи оптимизации . . . . .	19
<b>4</b>	<b>Заключение</b>	<b>22</b>
<b>5</b>	<b>Список литературы</b>	<b>23</b>
	<b>Список литературы</b>	<b>23</b>
<b>6</b>	<b>Приложения</b>	<b>24</b>
6.1	Таблица вычислительных ошибок в рандомизированном методе наименьших квадратов с различными размерами задачи и проекции. . . . .	24
6.2	Графики распределения нормы ошибки (относительной) для рандомизированного вычисления матриц Грама в задаче D-MIMO . . . . .	25
6.3	Графические иллюстрации области допустимой производительности при различных каналах и уровнях шума . . . . .	26
6.4	Сходимость метода ветвей и границ . . . . .	29
6.5	Графические иллюстрации аппроксимации области допустимой производительности с помощью модификации метода внутренней точки . . . . .	29

# 1 Введение

Современные математические задачи часто сталкиваются с проблемой высокой размерности и при классическом точном подходе требуют решения линейных систем, не разрешимых за разумное время даже на современных вычислительных системах. В связи с этим получила развитие область рандомизированных проекционных методов линейной алгебры, идея которых в сведении задач к значительно более простым с точки зрения вычислений, но дающих решение исходной сложной задачи с оценимой вероятностью и точностью. В данной работе сформулированы опорные утверждения рандомизированных методов линейной алгебры, а также рассмотрены некоторые рандомизированные алгоритмы для решения задач наименьших квадратов и быстрого приближенного умножения матриц, а так же аспекты их практической реализации.

Одна из задач, к которой был адаптирован и применен рандомизированный алгоритм умножения матриц - задача о связи по модели MIMO-downlink. В данной работе рассмотрены некоторые известные практические алгоритмы, использующие вычисление матриц Грама. После проведения экспериментов с быстрым умножением матриц применительно к этим алгоритмам, была подробно рассмотрена общая задача оптимизации MIMO-downlink и методы выпуклой и монотонной оптимизации, применимые к ней.

## 2 Рандомизированные методы

### 2.1 $\epsilon$ - вложения

Различные рандомизированные методы линейной алгебры обычно предполагают переход к пространству меньшей размерности с целью ускорения вычислений, но предполагают малую ошибку в точности по сравнению с полноценным алгоритмом в большом пространстве. Для достижения этого результата, требуется введение значительных ограничений на выбор подпространства, на которое производится проекция (или, что то же самое, на оператор проектирования). Введем такие ограничения:

**Определение 1** *Оператор  $A$  является  $\epsilon$  - вложением для пространства  $V$ , если  $\forall x \in V$  выполнено*

$$\|x\|_2^2(1 - \epsilon) \leq \|Ax\|_2^2 \leq \|x\|_2^2(1 + \epsilon) \quad (1)$$

Требование приближенного сохранения нормы всех векторов пространства является достаточно сильным и сразу может гарантировать ряд других свойств оператора. Например:

**Утверждение 1**  *$\epsilon$  - вложения сохраняют и скалярные произведения векторов, но с аддитивной ошибкой.*

Доказательство.

$$\forall x, y \in V : \|x\| = \|y\| = 1, 4(Ax, Ay) = \|Ax + Ay\|_2^2 - \|Ax - Ay\|_2^2$$

$$\|x + y\|_2^2(1 - \epsilon) - \|x - y\|_2^2(1 + \epsilon) \leq 4(Ax, Ay) \leq \|x + y\|_2^2(1 + \epsilon) - \|x - y\|_2^2(1 - \epsilon)$$

$$4(x, y) + 2\epsilon(\|x\|_2^2 + \|y\|_2^2) \leq 4(Ax, Ay) \leq 4(x, y) + 2\epsilon(\|x\|_2^2 + \|y\|_2^2)$$

$$(x, y) - \epsilon \leq (Ax, Ay) \leq (x, y) + \epsilon.$$

Можно доказать и более сложные утверждения, например о произведении матриц:

**Утверждение 2**  *$A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times m}, m < n. J \in \mathbb{R}^{r \times n}$  -  $\epsilon$ -вложение для столбцов  $B$  и строк  $A$ . Тогда*

$$\|AJ^T JB - AB\|_2 \leq \epsilon \|A\|_2 \|B\|_2. \quad (2)$$

Доказательство можно построить на сингулярном разложении матриц  $A$  и  $B$ :

$$\|AJ^T JB - AB\|_2 = \|U_A \Sigma_A (V_A J^T J U_B - V_A U_B) \Sigma_B V_B\|_2 \leq \quad (3)$$

$$\leq \|\Sigma_A\|_2 \|V_A J^T J U_B - V_A U_B\|_2 \|\Sigma_B\|_2 \leq \epsilon \|A\|_2 \|B\|_2. \quad (4)$$

Однако, далеко не очевидно, как можно получить  $\epsilon$  - вложение для произвольного пространства. На этот вопрос дают ответ случайные матрицы.

## 2.2 Лемма Джонсона-Линденштраусса

Рассмотрим одно из отправных утверждений рандомизированных методов линейной алгебры.

**Теорема 1 (Johnson-Lindenstrauss Lemma)**  $\forall \epsilon > 0, \forall n$  точек из пространства  $\mathbb{R}^d$  существует линейное отображение  $A: \mathbb{R}^d \rightarrow \mathbb{R}^k$ , такое что для каждой точки  $x$  из данного множества выполнено соотношение

$$\|x\|_2^2(1 - \epsilon) \leq \|Ax\|_2^2 \leq \|x\|_2^2(1 + \epsilon), \quad (5)$$

причем  $k$  имеет порядок  $O(\frac{\log n}{\epsilon^2})$ .

Известные доказательства этой леммы предполагают использование матрицы  $A$ , каждый элемент которой распределен независимо от остальных элементов по одному из следующих распределений:

- Нормальное распределение с дисперсией 1 и матожиданием 0
- Дискретное распределение вида

$$\begin{cases} Z = 1, Pr = 1/2 \\ Z = -1, Pr = 1/2 \end{cases}$$

Матрица затем нормируется по количеству строк и показывается, что вероятность получения таким образом матрицы, удовлетворяющей лемме, больше нуля. Такие доказательства, например, есть в [1, 5].

Это утверждение сохраняет норму лишь конечного числа точек, но само по себе интересно, а так же позволяет доказать и более общее утверждение:

**Теорема 2 (Subspace Johnson-Lindenstrauss Lemma)**  $\forall \epsilon, \forall V$  - линейного подпространства размерности  $n \in \mathbb{R}^d$  существует линейное отображение  $A: \mathbb{R}^d \rightarrow \mathbb{R}^k$ , такое что для каждой точки  $x$  из  $V$  выполнено соотношение

$$\|x\|_2^2(1 - \epsilon) \leq \|Ax\|_2^2 \leq \|x\|_2^2(1 + \epsilon), \quad (6)$$

причем  $k$  имеет порядок  $O(\frac{n \log n}{\epsilon^2})$ .

Доказательство последней теоремы в [7] идейно заключается в следующем. Нормированием рассматриваются лишь вектора  $V$  с единичной сферы. Затем, произведение отрезков  $[-1, 1]$  по каждой размерности покрывается равномерной сеткой, и заказывается отображение из Теоремы 1, сохраняющее с точностью до  $\epsilon$  норму каждого вектора сетки, а затем выводится оценка на изменение нормы произвольного вектора с единичной сферы. При этом число точек сетки экспоненциально зависит от размерности, поэтому при логарифмировании (по Т.1) появляется дополнительный множитель  $n$ .

Таким образом, можно показать, что матрица, распределенная случайно, нормально или дискретно, и имеющая размерность  $O(\frac{n \log n}{\epsilon^2})$  с высокой вероятностью даст  $\epsilon$ -вложение для любого подпространства.

Следует заметить, что переходить от размерности  $m$  к  $O(\frac{n \log n}{\epsilon^2})$  имеет смысл только в случае  $m \gg n$ . Однако, даже в этом случае проектирование на такое подпространство будет соответствовать умножению матрицы  $O(\frac{n \log n}{\epsilon^2}) \times m$  на матрицу  $m \times n$ , и сложность этого умножения асимптотически больше, чем  $O(mn^2)$ . Практически такие алгоритмы для вытянутых матриц заведомо неинтересны. Поэтому, для того, чтобы  $\epsilon$ -вложение можно было эффективно применить, нужны более быстрые алгоритмы JL-преобразования.

## 2.3 Быстрое преобразование JL

Для улучшения асимптотической скорости преобразования JL можно использовать разреженные матрицы со случайно распределенной позицией и величиной ненулевых элементов. Такая техника введена в [4]. Показано, что в качестве ускоренного преобразования JL можно использовать произведение трех матриц:

$$\Phi = [SparseJL] [Hadamard] [Diagonal] \quad (7)$$

SparseJL - разреженная матрица, каждый элемент которой не равен нулю с вероятностью  $q$ , а значение всех ненулевых элементов имеет независимое (друг с другом) нормальное распределение. Hadamard - матрица Адамара (будем считать, что размерность исходного пространства - степень двойки), Diagonal - диагональная матрица, каждый элемент которой выбирается как  $\pm 1$  с равной вероятностью.

**Теорема 3 (Лемма о быстром преобразовании JL)** *Фиксируем множество  $X$  из  $n$  векторов в  $\mathbb{R}^d$ ,  $\epsilon < 1$ . Пусть  $\Phi$  выбирается, как описано выше. Тогда существует такое  $q$ , что с вероятностью не менее  $2/3$ , для всех  $x \in X$*

$$\|x\|_2^2(1 - \epsilon) \leq \|\Phi x\|_2^2 \leq \|x\|_2^2(1 + \epsilon), \quad (8)$$

*а машинное выполнение преобразования  $\Phi: \mathbb{R}^d \rightarrow \mathbb{R}^k$  требует  $O(d \log d + \min(d\epsilon^{-2} \log(n), \epsilon^{-2} \log(n)^3))$  операций.*

На практике, непосредственное применение это теоремы неудобно, так как существующие оценки на константу в  $O(\dots)$  достаточно велики и обычно не позволяют получить выигрыш в производительности. Удобнее выбирать размерность преобразования JL и плотность матрицы эвристически, чтобы гарантировать ускорение, а затем рассматривать практически полученную величину ошибки какого-либо алгоритма (либо отклонение нормы векторов при проектировании). Доказательства аналога такой теоремы для дискретного заполнения матрицы элементами  $\pm 1$ , неизвестны, но такое построение на практике эффективнее в случаях, когда умножение элементов вычислительно сложнее сложения (например, для комплексных чисел).

## 2.4 Рандомизированный метод наименьших квадратов

Исходя из специфики преобразования JL для подпространств, его можно применить лишь к задачам с вытянутыми матрицами,  $m \gg n$ . Одной из таких задач является задача наименьших квадратов. Пусть  $m \gg n$ ,  $A \in \mathbb{R}^{m \times n}$ , и решается задача

$$\|Ax - b\|_2 \rightarrow \min; \quad (9)$$

Применив проектирование этой задачи с помощью JL-преобразования  $J$ , можно перейти к задаче с меньшим числом уравнений

$$\|JAx - Jb\|_2 \rightarrow \min. \quad (10)$$

Пусть  $x_{ls} = A^+b, x'_{ls} = (JA)^+Jb$ . Тогда выполнено следующее утверждение (оно доказано в [7]).

**Теорема 4** Пусть  $J$  -  $\epsilon$ -вложение, размера  $\mathbb{R}^{t \times n}$ , где  $t = O(\frac{n \log(n)}{\epsilon})$ ; Тогда выполнено

$$\|Ax'_{ls} - b\|_2 \leq (1 + \epsilon)\|Ax_{ls} - b\|_2 \quad (11)$$

В случае  $t = O(\frac{n \log(n)}{\epsilon^2})$  выполнено

$$\|x'_{ls} - x_{ls}\|_2 \leq \frac{\epsilon}{\sigma_{\min}(A)}\|Ax_{ls} - b\|_2. \quad (12)$$

Таким образом, можно применить какой-нибудь из способов JL - преобразования (речь о различных распределениях) для генерации  $J$ , и условия теоремы будут выполнены с высокой вероятностью.

На основе последней теоремы реализован рандомизированный метод наименьших квадратов и его параллельная версия, проведены численные эксперименты. Тесты проводились на задаче наименьших квадратов, возникающих при приближении  $f(x) = \sin(1/x)$  на равномерной сетке по  $[-1,1]$  полиномами Чебышева. Таблицы результатов см. в приложении 6.1.

## 2.5 Быстрое вычисление матриц Грама

Другой возможной задачей для вытянутых матриц, позволяющей применить ускоренный рандомизированный алгоритм, является вычисление матриц Грама. Простейший алгоритм для проведения ускоренных вычислений напрямую использует утверждение 2: ускорение умножения можно получить заменой исходного вычисления на вычисление матрицы Грама для JL-преобразованной матрицы.

Практическое применение быстрого вычисления матрицы Грама было проведено для одного из алгоритмов задачи ММО-downlink теории связи. Подробно об этой задаче в следующей главе.

## 3 Задача MIMO-downlink

### 3.1 Введение в задачу MIMO-downlink

Задача MIMO-downlink - задача о передаче данных  $m$  пользователям с помощью  $n$  антенн в соответствии с некоторой моделью, и обеспечения хорошего качества сигнала для всех пользователей, по возможности. Рассмотрим заданную модель связи. Пусть  $x \in \mathbb{C}^m$  - вектор комплексных данных,  $i$ -ый элемент которого мы хотим передать  $i$ -му пользователю. Процесс передачи информации описывается последовательным умножением на две матрицы. Первая матрица,  $P \in \mathbb{C}^{n \times m}$  - матрица прекодирования, после умножения на которой мы получим вектор размерности  $n$ , каждый элемент которого будет соответствовать тому, что на самом деле (физически) будет отправлено соответствующей антенной. Затем, после отправления вектор умножается на матрицу  $H \in \mathbb{C}^{m \times n}$  (канальную матрицу) и получается вектор размера  $m$  из данных, которые реально пришли пользователям.

Таким образом,

$$x' = \begin{bmatrix} \vec{h}_1 \\ \vec{h}_2 \\ \vdots \\ \vec{h}_m \end{bmatrix} [\vec{p}_1 \quad \vec{p}_2 \quad \dots \quad \vec{p}_m] x$$

где  $p_i \in \mathbb{C}^n$  - столбец матрицы прекодирования, определяющий *направление луча* для  $i$ -го пользователя (*beamforming vector*),  $h_i \in \mathbb{C}^n$  - строка матрицы канала, определяющая канал  $i$ -го пользователя.

Рассмотрим данные, полученные  $i$ -м пользователем.

$$x'_i = (\vec{h}_i \vec{p}_i) x_i + \sum_{j \neq i} (\vec{h}_i \vec{p}_j) x_j$$

Часть данных  $(\vec{h}_i \vec{p}_i) x_i$  желаемая и соответствует отправленному  $x_i$ . Однако пользователю придет и другая часть сигнала,  $\sum_{j \neq i} (\vec{h}_i \vec{p}_j) x_j$ , нежелательная, которая отправлялась другим пользователям. Соотношение между желательной и нежелательной частью сигнала в среднем (по поданным данным) и определяет качество сигнала. Введем формальное определение.

**Определение 2 (*Signal-to-Interference-and-Noise Ratio для  $k$ -го пользователя*)**

$$SINR_k := \frac{|(\vec{h}_k, \vec{p}_k)|^2}{\sigma_k^2 + \sum_{j \neq k} |(\vec{h}_k, \vec{p}_j)|^2} \quad (13)$$

В выражении выше  $\sigma_k \in \mathbb{R}$  - константа, соответствующая по смыслу уровню внешнего шума (не от интерференции с другими пользователями) для  $k$ -го пользователя. Таким образом определенный  $SINR_k$  соответствует среднему качеству сигнала  $k$ -го пользователя.

В приведенной модели матрица  $H$  определяется физикой среды и считается фиксированной в данный момент времени, а матрица  $P$  должна быть выбрана с целью оптимизации SINR-ов с ограничением по мощности на антеннах, что математически определяется как ограничение на норму каждой *строки* матрицы  $P$ . Таким образом, MIMO-downlink предполагает многокритериальную оптимизацию всех  $m$  SINR-ов при ограничениях на мощность каждой антенны, которые описываются, как ограничения на вторую норму каждой строки матрицы прекодирования.



### 3.2 Известные эвристические алгоритмы

Рассмотрим практические подходы к выбору матрицы  $P$ , имеющие низкую вычислительную сложность и обеспечивающие "неплохое" качество сигнала, являющиеся эвристическими, но дающие оптимальную матрицу прекодирования в определенных случаях:

Пусть  $m < n$  (пользователей меньше, чем антенн). В таком случае можно полностью устранить интерференцию для каждого пользователя, выбрав столбцы  $P$  как биортогональную систему со строками  $H$ ; такой подход в литературе называется *Zero-Force Beamforming* [2]. При таком подходе

$$SINR_k = \frac{|\vec{h}_k \vec{p}_k|^2}{\sigma_k^2}$$

откуда понятно (и это доказано в [2]), что этот алгоритм стремится к наилучшему в случае внешнего уровня шума, стремящегося к нулю. Соответствующая матрица  $P$  определяется тогда как  $(H^*H)^+H^*$ . Эту матрицу можно использовать как матрицу прекодирования и в случае  $m > n$ , но она потеряет свой физический смысл.

Другой практический подход рассматривает альтернативную величину  $SLNR$  вместо  $SINR$ :

**Определение 3** (*Signal-to-Leakage-and-Noise Ratio (SLNR) для k-го пользователя*)

$$SLNR_k := \frac{|(\vec{h}_k, \vec{p}_k)|^2}{\sigma_k^2 + \sum_{j \neq k} |(\vec{h}_j, \vec{p}_k)|^2} \quad (14)$$

Эта величина для каждого пользователя характеризует соотношение желаемого сигнала для некоего пользователя ( $k$ -го) к шуму и вкладу этого желаемого для  $k$ -го пользователя сигнала в интерференцию других пользователей. Так как  $SLNR_k$ , в отличие от  $SINR_k$ , зависит только от одного столбца матрицы прекодирования, оптимизирующие  $SLNR_k$  столбцы  $p_k$  можно найти аналитически. Можно показать, что с точностью до константы  $SLNR_k$  максимизируется вектором ([2])

$$p_k := (I + HDH^*)^{-1}h_k$$

где  $D$  - диагональная матрица. Можно так же показать, что этот алгоритм выбора матрицы прекодирования стремится к матрице ZF-алгоритма при низком уровне шума ([2]).

Таким образом, в известных практических подходах к решению задачи D-MIMO присутствует вычисление матрицы Грама  $H^*H$ . В случае количества пользователей, значительно превосходящих число антенн, такое вычисление можно провести рандомизированно.

### 3.3 Применение рандомизированного вычисления матриц Грама

С целью упрощения вычислений стратегия D-MIMO (*Distributed MIMO*) заключается в том, что пользователей и антенны обычно бьют на группы и сортируют по близости друг к другу. Иными словами, пользователи и антенны обычно можно перегруппировать таким образом, что матрицу канала можно рассматривать как блочную, причем элементы диагональных блоков будут иметь значительно большие модули, чем элементы остальных блоков. Группа антенн, соответствующих одному блоку (номера столбцов) обычно называется 'cell'.

Эту специфику можно использовать и для рандомизированного умножения. Так, при достаточно большом количестве пользователей (на рассматриваемом примере их 5632, а антенн 24) умножение можно провести блочно, причем каждый блок, соответствующий делению антенн и пользователей на 'cell-ы', также будет вытянутым. Затем, разные блоки можно перемножать рандомизированно с разными параметрами. Так, например, умножение двух диагональных блоков можно провести точно (таких блоков мало), а умножение двух внедиагональных - приближенно с малой размерностью JL-проектирования.

Алгоритмы JL - преобразования для комплексных матриц не имеют известных доказательств. Однако, использование разреженных JL матриц с элементами  $\pm 1$  сулило большую выгоду в производительности, так как умножение комплексных чисел затратнее их сложения, поэтому я реализовал его с целью проверить возможность умножения комплексных матриц экспериментально. Результаты дали относительную ошибку не хуже, чем для вещественного случая. Графики для полученных относительных ошибок при различных способах умножения представлены в приложении 6.2.

Однако, к сожалению, случай такого большого количества пользователей достаточно редкий, и, кроме того, обычно вместо поиска матрицы прекодирования для всех пользователей сразу, обычно пользователей разбивают на группы по временным интервалам передачи данных.

### 3.4 Постановка общей задачи

Применение быстрого вычисления матрицы Грама для ускорения одного из стандартных алгоритмов D-MIMO позволяет получить существенное ускорение, но лишь в ограниченном круге задач, в котором число пользователей очень велико, что бывает редко. Рассмотрим другие, оптимизационные подходы к выбору матрицы прекодирования.

Для постановки общей задачи оптимизации MIMO-downlink введем выражение, описывающее суммарное качество сигнала у всех пользователей одновременно, и будем его оптимизировать. Качество сигнала одного пользователя (*user-performance function*) определяем следующим образом:

$$g_k = \lg(1 + SINR_k)$$

Общую задачу оптимизации MIMO-downlink определим как максимизацию линейного функционала от  $g_i$ :

$$f = \sum_k \frac{B_k}{r_k} g_k \rightarrow \max$$

при условиях

$$g_k = \lg(1 + SINR_k)$$

$$SINR_k := \frac{|\vec{h}_k \vec{p}_k|^2}{\sigma_k^2 + \sum_{j \neq k} |\vec{h}_k \vec{p}_j|^2}$$

$$\sum_k |(\vec{p}_k)_l|^2 \leq P_l, l = 1 \dots n, k = 1 \dots m$$

где последнее неравенство - ограничение по мощности каждой антенны (ограничение второй нормы каждой строки матрицы прекодирования),  $B_k$  - заданный физический коэффициент, определяющий ширину полосы, а  $r_k$  - усреднение user-performance  $g_k$  соответствующего

пользователя по нескольким предыдущим временным интервалам. Его тоже можно считать константой в рамках решения задачи для текущего временного интервала. Он вводится для того, чтобы пользователи, обладающие плохим каналом (зашумленные, или расположенные далеко от антенн) не были полностью проигнорированы при оптимизации, что достигается путем увеличения соответствующего коэффициента линейного функционала  $f$ .

Заметим, что наличие шума  $\sigma_k$  в задаче крайне существенно. Ранее уже было показано, что в случае малого шума алгоритм ZF дает отличное качество сигнала всем пользователям, и в дальнейшем влияние шума на возможное качество сигнала будет проиллюстрировано графически.

### 3.5 Существующие исследования этой задачи

Поставленная задача оптимизации не имеет известных эффективных методов, гарантирующих быструю сходимость к точному решению. Однако, известно несколько свойств этой задачи.

**Определение 4** *Допустимое множество производительности - множество всех точек  $(g_1, \dots, g_m)$ , таких что качество сигнала  $g_i$  достижимо для каждого пользователя одновременно при некоторой матрице  $P$ , удовлетворяющей условиям на мощность.*

После введения такого определения общую задачу оптимизации D-MIMO можно геометрически интерпретировать как задачу нахождения наивысшей точки касания допустимого множества производительности гиперплоскостью с заданной нормалью.

Область производительности можно считать нормальной, в том смысле что максимум линейного функционала с положительными коэффициентами не может достигаться на точке, меньшей по всем координатам, чем другая допустимая точка.

Граница допустимой области производительности  $R$  является границей Парето.

Таким образом, введение линейного функционала с положительными коэффициентами как цель оптимизации определяет оптимальную точку на границе Парето.



Рис. 1: Допустимая область производительности двух пользователей

Для решения общей задачи оптимизации D-MIMO требуется, по крайней мере, инструмент для работы с областью  $R$ . Способы ее аналитического представления известны лишь

для тривиальных канальных матриц, и, как мы увидим далее, она может быть весьма нетривиальной и существенно зависит от показателей внешнего шума и интерференции.

Проверка принадлежности некоторой точки допустимой области производительности называется задачей *Fixed Quality of Service* (fixed-QoS), так как она проверяется возможность обеспечения каждого пользователя неким заданным качеством связи. Известно, что для этой задачи существует выпуклая переформулировка, и, соответственно, она может быть решена эффективно методом внутренней точки.

### 3.6 Подзадача fixed-QoS

Задача fixedQoS определяется следующим образом ([2]):

Найти  $\vec{p}_1, \vec{p}_2, \dots, \vec{p}_m \in \mathbb{C}^n$  (матрицу прекодирования), такие что выполнены следующие ограничения:

- Мощностные ограничения:  $\sum_k |(\vec{p}_k)_l|^2 \leq P_l, l = 1 \dots n$
- Ограничения качества:  $SINR_k \geq \gamma_k$ . ("Качество не хуже, чем запрошенное качество  $\gamma_k$ ".)

где  $\gamma_k$  - требуемое качество сигнала - задано, или констатировать, что множество, удовлетворяющее всем ограничениям, пусто ( $\vec{p}_1, \vec{p}_2, \dots, \vec{p}_m \in \mathbb{C}^n$  не существуют).

Покажем, что наложенные ограничения образуют выпуклое множество.

Перепишем неравенства  $SINR_k \geq \gamma_k$  с учетом определения SINR:

$$\frac{|\vec{h}_k \vec{p}_k|^2}{\sigma_k^2 + \sum_{j \neq k} |\vec{h}_k \vec{p}_j|^2} \geq \gamma_k$$

$$\vec{h}_k \vec{p}_k \geq \gamma_k (\sigma_k^2 + \sum_{j \neq k} |\vec{h}_k \vec{p}_j|^2)$$

$$(1 + \gamma_k) \vec{h}_k \vec{p}_k \geq \gamma_k (\sigma_k^2 + \sum_j |\vec{h}_k \vec{p}_j|^2)$$

$$\sqrt{\frac{1 + \gamma_k}{\gamma_k}} \operatorname{Re}(h_k^H p_k) \geq \left\| \begin{array}{c} h_k^H p_1 \\ h_k^H p_2 \\ \vdots \\ h_k^H p_m \\ \sigma_k \end{array} \right\|, k = 1 \dots m$$

Для применения методов выпуклой оптимизации требуется функционал, который необходимо оптимизировать; его можно ввести с помощью новой переменной  $\beta$ , которая будет характеризовать часть мощностей, реально использованных текущей матрицей прекодирования, и минимизировать эту часть. Тогда можно записать следующую задачу выпуклой оптимизации:

$$z := \begin{bmatrix} Re(p_1) \\ Im(p_1) \\ Re(p_2) \\ Im(p_2) \\ \vdots \\ Re(p_m) \\ Im(p_m) \\ \beta \end{bmatrix}$$

$$f(z) = f(\vec{p}_1, \vec{p}_2, \dots, \vec{p}_m, \beta) = \beta \rightarrow \min \quad (15)$$

при условиях

$$1. \sum_k |(\vec{p}_k)_l|^2 \leq \beta^2 P_l, l = 1 \dots n$$

$$2. \frac{1+\gamma_k}{\gamma_k} (Re(h_k^H p_k))^2 \geq \left\| \begin{bmatrix} h_k^H p_1 \\ h_k^H p_2 \\ \vdots \\ h_k^H p_m \\ \sigma_k \end{bmatrix} \right\|^2, k = 1 \dots m$$

$$3. Re(h_k^H p_k) \geq 0, k = 1 \dots m$$

$$4. Im(h_k^H p_k) = 0, k = 1 \dots m.$$

И если результирующий  $\min f \leq 1$ , можно заключить, что комбинация SINR-ов достижима, иначе - недостижима.

Ограничение на качество (2,3) можно переписать в стандартном виде конуса второго порядка, таким образом показав, что оно выпукло. В самом деле, условие имеет следующий вид [2]:

$$c_k z \geq \|A_k z + b_k\| \quad (16)$$

где:

$$\bullet c_k := \sqrt{\frac{1+\gamma_k}{\gamma_k}} [0 \dots 0, Re(h_k), Im(h_k), 0 \dots 0, 0]$$

$$\bullet b_k := \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma_k \end{bmatrix}$$

$$\bullet A_k := \begin{bmatrix} V_k & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & V_k & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$V_k := \begin{bmatrix} Re(h_k) & Im(h_k) \\ -Im(h_k) & Re(h_k) \end{bmatrix}$$

Аналогичный вид для ограничения по мощности:

$$d_l z \geq \|Q_l z\|, l = 1 \dots n \quad (17)$$

где:

- $Q_l := \begin{bmatrix} \vec{e}_l & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \dots & \vec{e}_l & 0 \\ 0 & \dots & \dots & 0 \end{bmatrix}$
- $d_l := \sqrt{P_l} [0 \dots 0, 1]$

Таким образом, показано, что задача fixedQoS сводится к задаче выпуклой оптимизации.

### 3.7 Решение задачи fixed-QoS методом внутренней точки

Решение задачи выпуклой оптимизации для решения задачи fixed-QoS было реализовано программно с помощью метода внутренней точки [6]. Опишем примененный метод подробнее.

Задача

$$f(x) \rightarrow \min, \quad (18)$$

при условиях

$$f_i(x) \leq 0, i = 1 \dots m, Ax = b$$

где  $f, f_1, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$  выпуклы и непрерывно дифференцируемы,  $A \in \mathbb{R}^{p \times n}$ ,  $rank(A) = p < n$ , является выпуклой задачей оптимизации с ограничениями в виде неравенств и аффинными ограничениями в виде равенств.

Для таких выпуклых задач вводится функция Лагранжа

$$L(x, \lambda, v) := f(x) + \sum_i \lambda_i f_i(x) + v^T (Ax - b)$$

$$g(\lambda, v) = \inf_x L(x, \lambda, v)$$

Задача оптимизации

$$g(\lambda, v) \rightarrow \max, \text{ при условиях } \lambda_i \geq 0, i = 1 \dots m$$

называется задачей, двойственной к исходной.

В предположениях существования допустимой точки, для которой все неравенства выполнены строго, оптимальное решение прямой задачи  $x^*$  и двойственной задачи  $\lambda^*, v^*$  необходимо связаны условиями Куна-Таккера

$$\begin{aligned} Ax^* &= b, f_i(x^*) \leq 0, i = 1 \dots m; \\ \lambda_i^* &\geq 0, i = 1 \dots m; \\ \nabla f(x) + \sum_i \lambda_i^* \nabla f_i(x^*) + A^T v^* &= 0; \end{aligned}$$

$$\lambda_i^* f_i(x^*) = 0, i = 1 \dots m;$$

Простейшая интерпретация вариации метода внутренней точки заключается в следующем: производится модификация условий Куна-Таккера к выражениям:

$$\begin{aligned} Ax &= b; \\ \nabla f(x) + \sum_i \lambda_i \nabla f_i(x) + A^T v &= 0; \\ -\lambda_i f_i(x) &= \frac{1}{t}, i = 1 \dots m; \end{aligned}$$

где  $t$  - константа-параметр. Метод внутренней точки тогда можно интерпретировать как прямое решение модифицированных условий Куна-Таккера с помощью итераций Ньютона.

Введем невязку по компонентам системы

$$r = \begin{bmatrix} \nabla f(x) + \sum_i \lambda_i \nabla f_i(x) + A^T v \\ -\lambda_i f_i(x) - \frac{1}{t} \\ Ax - b \end{bmatrix} \quad (19)$$

и рассмотрим шаг Ньютона для решения нелинейной системы

$$r(x, \lambda, v) = 0 \quad (20)$$

Для этой системы запишем линейную модель

$$r(y + \Delta y) \approx r(y) + Dr(y)\Delta y = 0$$

$$\Delta y = -Dr(y)^{-1}r(y)$$

Получим следующую систему:

$$\begin{bmatrix} \nabla^2 f(x) + \sum_i \lambda_i \nabla^2 f_i(x) & DF(x)^T & A^T \\ -diag(\vec{\lambda})DF(x) & -diag(F(x)) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \lambda \\ \Delta v \end{bmatrix} = -r$$

где

$$F(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, DF(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}.$$

Алгоритм примененного метода внутренней точки [6]:

На вход поступает некоторая начальная точка  $x$ , удовлетворяющая всем ограничениям, а так же набор  $\lambda > 0, \nu > 0$ .

1. Определяется  $t = const \frac{m}{-F(x)^T \vec{\lambda}}$
2. Вычисляется направление поиска  $y_{pd}$
3. Линейным поиском находится шаг  $s > 0$ , не выводящий за пределы ограничений в виде неравенств, и происходит добавка  $y = y + s\Delta y_{pd}$ .

Алгоритм останавливается по падению невязки  $r$  до заданного малого числа.

Однако, для выполнения требований метода внутренней точки необходимо выбрать стартовую точку. Дополнив неизвестные еще одной координатой  $\beta$ , мы теперь можем задать ее изначально достаточно большой, что эквивалентно наличию неограниченной мощности на антеннах. Но даже при неограниченной мощности не каждый набор SINR-ов достижим.

Это можно увидеть на следующем примере:

$$H = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Если два пользователя имеют полностью одинаковые каналы, соотношение желаемого сигнала к шуму от интерференции не может быть одновременно  $\geq 1$ . Этот пример - искусственный, однако для корректной работы функции fixed-QoS с данными, лежащими близко к границе допустимой области, требуется надежный метод получения стартовой точки, который не будет ухудшать практическую сходимость метода внутренней точки в целом.

Для решения проблемы поиска стартовой точки было найдено решение с помощью еще одной модификации условий задачи выпуклой оптимизации. Добавим  $\beta$  не только в ограничения мощности, но и в ограничения на качество:

1.  $\sum_k |(\vec{p}_k)_l|^2 \leq \beta^2 P_l, l = 1 \dots n$
2.  $\frac{1+\gamma_k}{\gamma_k} (\text{Re}(h_k^H p_k) + (\beta - 1))^2 \geq \left\| \begin{bmatrix} h_k^H p_1 \\ h_k^H p_2 \\ \vdots \\ h_k^H p_m \\ \sigma_k \end{bmatrix} \right\|^2, k = 1 \dots m$
3.  $\text{Re}(h_k^H p_k) + (\beta - 1) \geq 0, k = 1 \dots m$
4.  $\text{Im}(h_k^H p_k) = 0, k = 1 \dots m.$

Нетрудно видеть, что при  $\beta = 1$  мы получаем исходные условия на качество и на мощность, из чего следует, что если  $\min \beta > 1$ , то запрошенный набор SINR-ов недостижим. Если же  $\min \beta < 1$ , то можно видеть, что решена задача с более жесткими неравенствами и меньшей доступной мощностью  $\Rightarrow$  и запрошенное качество достижимо. Таким образом,  $\min \beta \leq 1$  все так же является критерием принадлежности точки допустимой области производительности. Модифицированные условия на качество также допускают запись в виде конуса 2 порядка.

Однако, для получения начальной точки достаточно положить  $v_i = h_i$ , а  $\beta$  выбрать достаточно большим, чтобы условия 1-3 были выполнены. Кроме того, такая точка сразу будет удовлетворять условию 4 и являться неплохим начальным приближением в том смысле, что каждый пользователь будет иметь большую желаемую компоненту сигнала.

На каждой итерации метода внутренней точки требуется решить линейную систему большого размера. Так, для  $m = 100$  пользователей,  $n = 64$  антенн необходимо найти вещественную и мнимую часть матрицы прекодирования размера  $m \times n$ , таким образом размер системы  $\approx 13000$ . Так как метод внутренней точки для решения задачи оптимизации D-MIMO рассматривается лишь как инструмент проверки принадлежности допустимому множеству, который необходимо применить большое количество раз, необходимо решать линейные системы с большой скоростью. Этого можно добиться, используя структуру матрицы системы.



Для рассмотрения структуры матрицы проведем дифференцирование аналитически. (Определения векторов и матриц можно посмотреть в сведениях к выпуклой задаче.)

- $f(z) = \beta^2 \rightarrow \min$
- $\nabla f(z) = [0 \dots 0, 2\beta]^T$
- $\nabla^2 f(z) = \text{diag}(0 \dots 0, 2)$
- $f_l(z) = \|Q_l z\|^2 - (d_l z)^2 \leq 0, l = 1 \dots n$
- $g_k(z) = \|A_k z + b_k\|^2 - (c_k z + u_k)^2 \leq 0, k = 1 \dots m$
- $\nabla f_l(z) = 2Q_l^T Q_l z - 2d_l^T (d_l z) = 2\text{diag}(\vec{e}_l, \vec{e}_l, \dots, \vec{e}_l, -P_l)z$
- $\nabla g_k(z) = 2A_k^T (A_k z + b_k) - 2c_k^T (c_k z + u_k) = 2A_k^T A_k z - 2c_k^T (c_k z + u_k)$
- $\nabla^2 f_l(z) = 2Q_l^T Q_l - 2d_l^T d_l = 2\text{diag}(\vec{e}_l, \vec{e}_l, \dots, \vec{e}_l, -P_l)$
- $\nabla^2 g_k(z) = 2A_k^T A_k - 2c_k^T c_k$

А вся матрица будет иметь вид

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

где

- $A \in \mathbb{R}^{2mn \times 2mn}$  - основная часть матрицы, соответствует неизвестным компонентам матрицы прекодирования. Путем перестановок строк и столбцов ее можно привести к блочно-диагональной матрице с блоками размера  $\in \mathbb{R}^{2n \times 2n}$ .
- $B = C^T, B \in \mathbb{R}^{2mn \times (3m+n)}$ , она является плотной (содержит градиенты функций - ограничений на качество, которые являются плотными: качество сигнала любого пользователя существенно зависит от каждого элемента матрицы прекодирования).
- $D = D^T \in \mathbb{R}^{(3m+n) \times (3m+n)}$ , малый квадратный блок.

Структура основного блока матрицы позволяет вычислить  $A^{-1}$  очень эффективно, поэтому для решения всей системы будет использовать блочное исключение. Перейдем от системы к эквивалентной ей:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

→

$$\begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b - CA^{-1}a \end{bmatrix}$$

Решение всей системы с помощью блочного исключения будет состоять из факторизации  $A$ , решения системы с матрицей  $A$  и матрицей правых частей  $B$ , умножения результата на  $C$  для получения нижнего квадратного блока, а затем решения блочно-треугольной системы.

$A$  является блочно-диагональной, причем блоки являются симметричными, но вообще говоря не положительно определенными. Так, по смыслу задачи желаемая компонента сигнала, которая предполагается доминирующей, вычитается во всех ограничениях на качество. Поэтому для факторизации  $A$  и решения с ней систем я использовал  $LDL^T$  факторизацию с помощью стандартных функций `dsytrf`, `dsytrs` из библиотеки LAPACK.

Общая сложность решения системы оценивается так:

- Факторизация основного квадратного блока  $A$ :  $\approx \frac{8}{3}n^3m$
- Решение системы с прямоугольным блоком  $B$  как правой частью с помощью полученной факторизации:  $\approx 2n^2(2n + m)m$
- Умножение двух прямоугольных блоков  $C(A^{-1}B)$ :  $\approx 2(2n + m)^2mn$

Остальные части решения системы имеют меньший порядок сложности. Таким образом общая сложность одной итерации метода внутренней точки при  $m = n$ :  $\approx 27m^4$ . Это отличная сложность, с учетом того, что система имеет порядок  $\approx m^2$ .

Вычислительная сложность метода внутренней точки для задачи fixed-QoS определяется так же и количеством итераций, за которое невязка падает достаточно сильно. В моей реализации требуемое падение невязки  $10^{-8}$ . Количество итераций в таком случае на практике обычно 8-25 и существенно зависит от запрошенной точки. Так, например, если для трех пользователей и некоторого канала проверить достижимость SINR-ов (1,1,1) и SINR-ов (1000, 0.001, 0.001), для второго случая потребуется втрое больше итераций метода внутренней точки.

Практически также установлено, что если записать условия на качество без квадратов, то требуемое количество итераций метода внутренней точки упадет вдвое. Однако, в таком случае к блочной структуре основного блока матрицы будет добавлена добавка ранга  $m$ , и эффективное решение с помощью многократных  $LDL^T$  факторизаций более не будет возможным, но такая матрица все еще будет допускать быстрое умножение. Попытка применить метод сопряженных градиентов для такой постановки задачи не дала выгоды, т.к. преимущество, полученное от уменьшения числа итераций метода внутренней точки было полностью нивелировано за счет большого числа итераций метода сопряженных градиентов, и общая производительность оказалась существенно ниже, чем для постановки задачи с квадратами.

Полученное мной программное решение задачи fixedQoS позволило построить графики допустимой области производительности для практических трехмерных случаев (с помощью линейного поиска, на каждой точке равномерной сетке по двум координатам). Графики примеров допустимых областей производительности представлены в приложении 6.3.

### 3.8 Модификация метода и аппроксимация допустимой области

Изначально параметр  $\beta$  вводился по двум причинам - был необходим функционал для модификации и начальная точка. Для получения начальной точки параметр был встроен и в неравенства, отвечающие за ограничения по качеству, и, таким образом, в случае результирующего  $\min \beta < 1$  полученная матрица прекодирования гарантировала даже лучшее качество

сигнала, чем было изначально запрошено. Небольшое изменение в алгоритме позволяет существенно приблизить заданную точку к границе достижимой области производительности. В случае  $\min \beta < 1$  на одной из итераций метода внутренней точки текущий  $\beta < 1$ . Как только это произойдет, можно считать, что получена начальная точка для решения выпуклой задачи с классическим ограничением на мощность:

$$f = \beta \rightarrow \min$$

$$1. \sum_k |(\vec{p}_k)_l|^2 \leq P_l, l = 1 \dots n$$

$$2. \frac{1+\gamma_k}{\gamma_k} (\operatorname{Re}(h_k^H p_k) + (\beta - 1))^2 \geq \left\| \begin{array}{c} h_k^H p_1 \\ h_k^H p_2 \\ \vdots \\ h_k^H p_m \\ \sigma_k \end{array} \right\|^2, k = 1 \dots m$$

$$3. \operatorname{Re}(h_k^H p_k) + (\beta - 1) \geq 0, k = 1 \dots m$$

$$4. \operatorname{Im}(h_k^H p_k) = 0, k = 1 \dots m.$$

Таким образом, минимизация теперь будет соответствовать улучшению качества сигнала (заметим, это не аналог оптимизации функционала общей задачи оптимизации MIMO-downlink, которая не сводится к выпуклой задаче). Направление улучшения может быть параметризовано введением коэффициентов при  $\beta - 1$  в ограничения на качество сигнала, зависящих от пользователя, но оно не является прямой линией на допустимой области производительности, так как область производительности определяется в логарифмических координатах. Однако, она может быть использована для улучшения нижних оценок на значения максимума в методе ветвей и границ, о котором пойдет речь в следующем пункте. Кроме того, учитывая гладкость области на практике, можно использовать полученные точки для аппроксимации области достижимой производительности линейной комбинацией базисных функций методом наименьших квадратов. Об этом см. в приложении 6.5.

### 3.9 Метод ветвей и границ для общей задачи оптимизации

Получив инструмент для определения допустимой области  $R$ , имеем возможность перейти к решению общей задачи оптимизации.

Для поиска оптимальной точки можно применить метод ветвей и границ (*Branch-Reduce-bound method*) [3].

Была использована готовая программная реализация этого метода с применением AVL-деревьев.

Идея метода заключается в итерационном покрытии границы области достижимой производительности  $R$   $m$ -мерными параллелепипедами (боксами), которые будут сжиматься к оптимальной точке. Каждый параллелепипед будет содержать достижимую точку внутри себя и некую верхнюю оценку на возможное значение функционала всюду внутри себя. Начать можно, например, с полного покрытия всей области одним параллелепипедом.

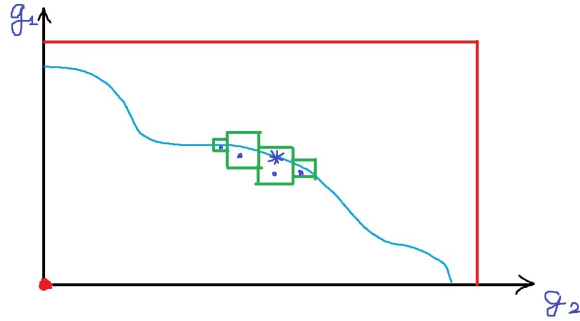


Рис. 2: Покрытие границы параллелепипеда-ми

Каждая итерация метода состоит из трех частей. Первая часть предполагает разбиение одного из параллелепипедов вдоль его самой длинной стороны на два. Достижимая точка для нижнего параллелепипеда проектируется на него, если она выше полученного параллелепипеда. Также для нижнего параллелепипеда проверяется, не дает ли его новая вершина лучшей верхней оценки, чем имелась ранее.

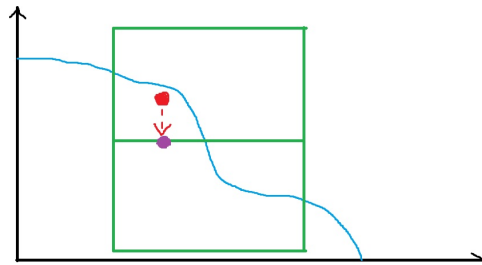


Рис. 3: Разбиение бокса

Вторая часть заключается в сжатии параллелепипеда на основе имеющихся оценок на оптимум. Так, имея нижнюю оценку на оптимальное значение функционала (максимум значения функционала по всем достижимым точкам всех боксов), можно найти точку вдоль каждого ребра бокса, ниже которой искать бессмысленно - значения функционала будут слишком малыми. В случае линейного функционала для координат таких точек имеется явное аналитическое выражение. Тогда, возможно, появится возможность обрезать бокс.

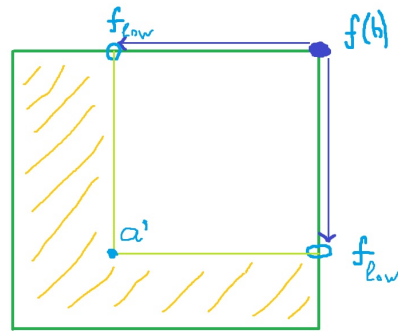


Рис. 4: Сужение бокса

Третья часть заключается в проверке, может ли оптимальная точка находиться в боксе, после чего улучшается верхняя и нижняя оценка на значения функционала внутри бокса.

Если локальная верхняя оценка ниже глобальной нижней оценки на значение оптимума, то бокс бесполезен и может быть отброшен (он ниже границы.)

Если достижимая точка после сужения или разбиения бокса вне бокса и ниже его, то необходимо проверить, достижима ли хотя бы нижняя вершина бокса. Если это не так, бокс целиком выше границы и может быть отброшен.

Если бокс все еще не отброшен, то он может иметь пересечение с границей и содержать оптимум. Тогда для улучшения локальных оценок используется линейный поиск по диагонали бокса.

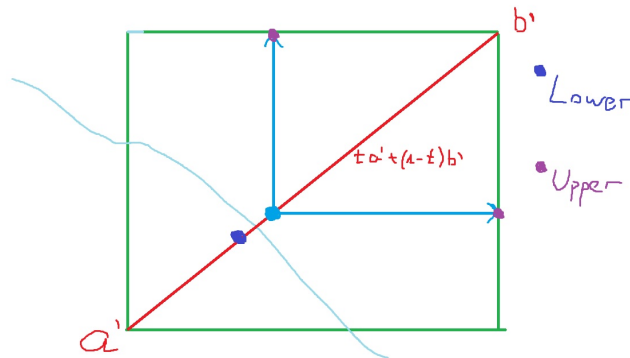


Рис. 5: Линейный поиск

Я подключил свою реализацию метода внутренней точки к готовой программной реализации этого метода и смог провести практическое решение общей задачи оптимизации на размерностях вплоть до 10. Практическое время и количество итераций метода представлено в приложении 6.4. Кроме того, синими стрелками отмечены оптимальные значения функционала на областях достижимой производительности в приложении 6.3. Зелеными стрелками там же обозначены точки, полученные алгоритмом Zero-force - для сравнения.

## 4 Заключение

В данной работе рассмотрены две задачи, решаемые рандомизированными методами линейной алгебры: задача наименьших квадратов и быстрого умножения матриц. Для обеих задач проведены программные эксперименты с различными способами быстрого JL - преобразования. Рандомизированное вычисление матрицы Грама применено на прикладной задаче D-MIMO, а так же оптимизировано для нее с учетом структуры матрицы канала.

Далее, были изучены известные алгоритмы для общей задачи оптимизации MIMO-downlink, метод внутренней точки для подзадачи fixed-QoS был реализован программно, решение внутренней линейной системы метода значительно оптимизировано с использованием структуры матрицы. На основе решения подзадачи fixed-QoS были графически построены области допустимой производительности для различных каналов и уровней шума, а исходя из графиков получены идеи по аппроксимации области. Предложена модификация постановки задачи fixed-QoS, позволяющая получать улучшенные оценки для метода ветвей и границ и представляющая аналог линейного поиска, работающий за время одного запуска решения задачи fixed-QoS. С помощью этого аналога линейного поиска построены аппроксимации допустимой области производительности.

Различные модификации и оптимизации метода внутренней точки, таким образом, позволили получать за разумное время решение общей задачи оптимизации D-MIMO для больших размерностей вплоть до 10.

## 5 Список литературы

### Список литературы

- [1] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66:671–687, 2003.
- [2] Eduard Jorswieck Emil Bjornson. *Optimal resource allocation in coordinated multi-cell systems*. Now Publishers, 2013.
- [3] Phan Thien Thach Hoang Tuy, Fais Al-Khayyal. Monotonic optimization: branch and cut methods.
- [4] Bernard Chazelle Niar Ailon. The fast johnson-lindenstrauss transform and approximate nearest neighbours. *SIAM Journal on Scientific Computing*, 35(3):A1511–A1536, 2013.
- [5] Anupam Gupta Sanjoy Dasgupta. An elementary proof of a theorem of johnson and lindenstrauss. *Wiley Periodicals*, 2002.
- [6] Lieven Vanderberghe Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2004.
- [7] Anastasios Zouzias. Randomized primitives for linear algebra and applications. *University of Toronto*, 2013.

## 6 Приложения

### 6.1 Таблица вычислительных ошибок в рандомизированном методе наименьших квадратов с различными размерами задачи и проекции.

Для тестирования рандомизированного метода наименьших квадратов была создана искусственная задача наименьших квадратов, заключающаяся в приближении функции приближении  $f(x) = \sin(1/x)$  на  $[-1,1]$  полиномами Чебышева с минимизацией второй нормы ошибки аппроксимации на равномерной сетке по  $[-1, 1]$ . Число точек сетки много превосходит число базисных функций.  $\epsilon$  задавалось мной довольно большим, а размерность проекции выбиралась как размерность проекции, требуемая теоремами о JL, деленная на 40.

В следующих таблицах видно, что норма невязки и ошибки даже при много сниженной размерности проекции значительно меньше  $\epsilon$  по относительной величине, то есть требуемые в теоремах о JL размерности могут быть завышенными на практике. Матрицы JL генерировались разреженными, как в теореме 3. Проведены эксперименты с нормальным и дискретным распределением элементов матрицы JL, а так же с преобразованием Адамара и без него.

Метод	Норма невязки	Норма ошибки
Матрица $262144 \times 2048$ , $\epsilon = 0.4$ , $r = 39038$ , $q = 0.01489$		
Точное решение (MKL, dgels)	0.14853	Норма решения: 1.1917
Базис из единиц, без м. Адамара	0.15256	0.05917
Базис из единиц, с м. Адамара	0.15283	0.07186
Гауссовский базис, без м. Адамара	0.15276	0.07013
Гауссовский базис, с. м Адамара	0.15307	0.07031
Матрица $262144 \times 2048$ , $\epsilon = 0.2$ , $r = 78076$ , $q = 0.01489$		
Точное решение (MKL, dgels)	0.14853	Норма решения: 1.1917
Базис из единиц, без м. Адамара	0.15043	0.04481
Базис из единиц, с м. Адамара	0.15074	0.04882
Гауссовский базис, без м. Адамара	0.15062	0.04388
Гауссовский базис, с. м Адамара	0.15104	0.04916

Метод	Норма невязки
Матрица $2097152 \times 2048$ , $\epsilon = 0.4$ , $r = 39038$ , $q = 0.000745$	
Базис из единиц, без м. Адамара	0.15284
Базис из единиц, с м. Адамара	0.15272
Гауссовский базис, без м. Адамара	0.15252
Гауссовский базис, с. м Адамара	0.15256
Матрица $2097152 \times 2048$ , $\epsilon = 0.2$ , $r = 78076$ , $q = 0.000745$	
Базис из единиц, без м. Адамара	0.15042
Базис из единиц, с м. Адамара	0.15051
Гауссовский базис, без м. Адамара	0.15049
Гауссовский базис, с. м Адамара	0.15070



## 6.2 Графики распределения нормы ошибки (относительной) для рандомизированного вычисления матриц Грама в задаче D-MIMO

В этой секции рассмотрены практические результаты для рандомизированного вычисления матриц Грама для задачи D-MIMO (см. 3.3). Эксперименты проводились на матрице  $5632 \times 24$ . Рассмотрены различные стратегии выбора блоков для приближенного умножения. Использовался только базис из  $\pm 1$ , так как сложение комплексных чисел быстрее их умножения. Сложность умножения в сравнении с классическим умножением - см. в заголовках картинок.

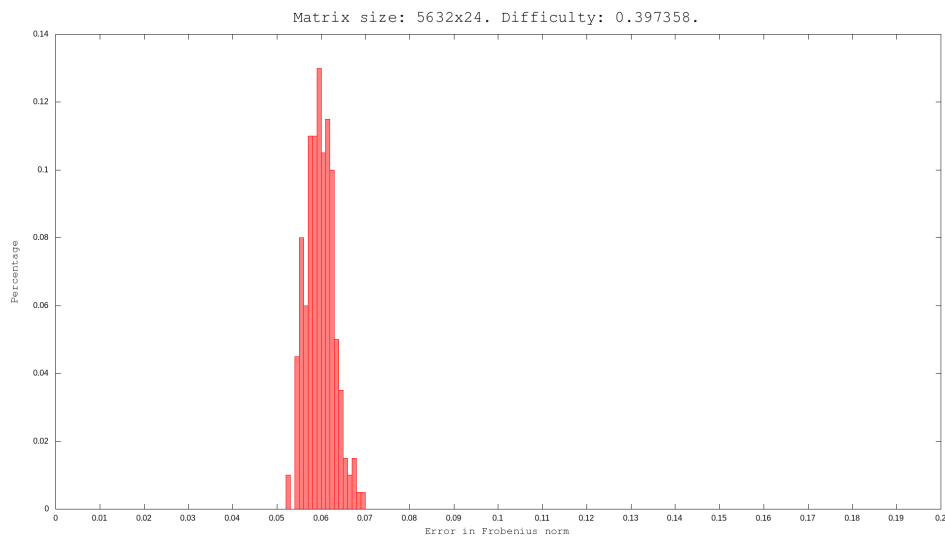


Рис. 6: Приближенное умножение всех блоков

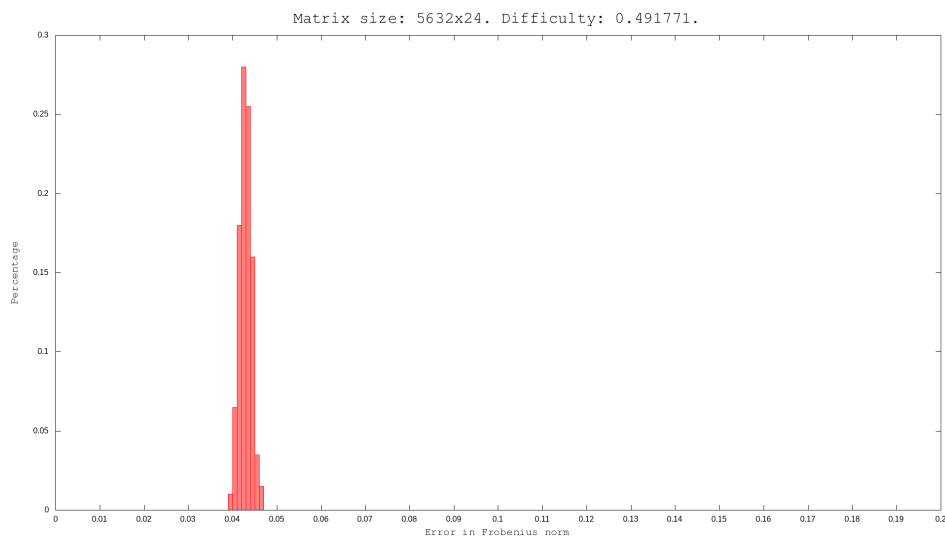


Рис. 7: Умножение диагональных блоков между собой выполнено точно

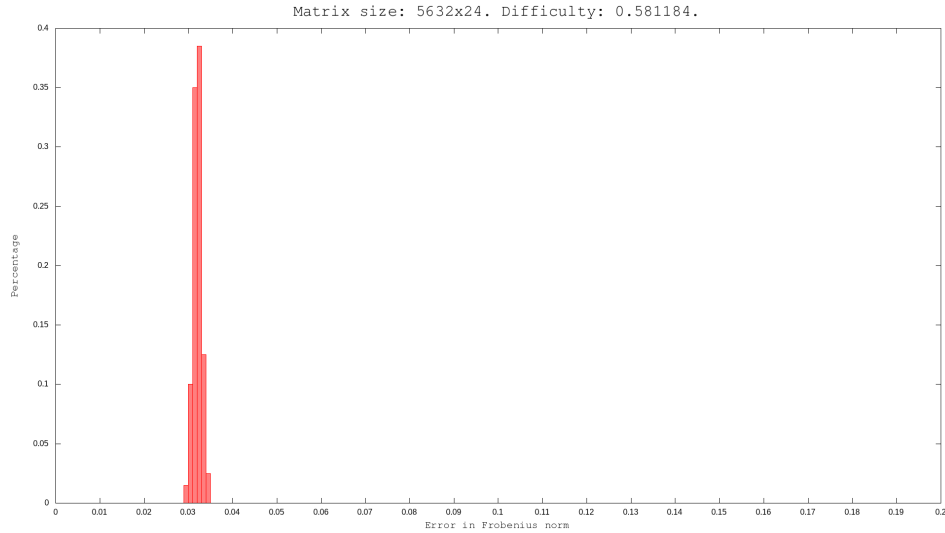


Рис. 8: Приближенно перемножаются только внедиагональные блоки между собой

### 6.3 Графические иллюстрации области допустимой производительности при различных каналах и уровнях шума

В этой секции приведены графические иллюстрации возможной формы допустимой области для трех пользователей. Они получены введением равномерной сетки по осям  $x$ ,  $y$ , и проведением линейных поисков пересечения вертикальных лучей с границей допустимой области. Также, на графиках отмечены оптимальные значения функционала (синие стрелки), и значения, полученные алгоритмом Zero-Force (зеленые стрелки, для сравнения).

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

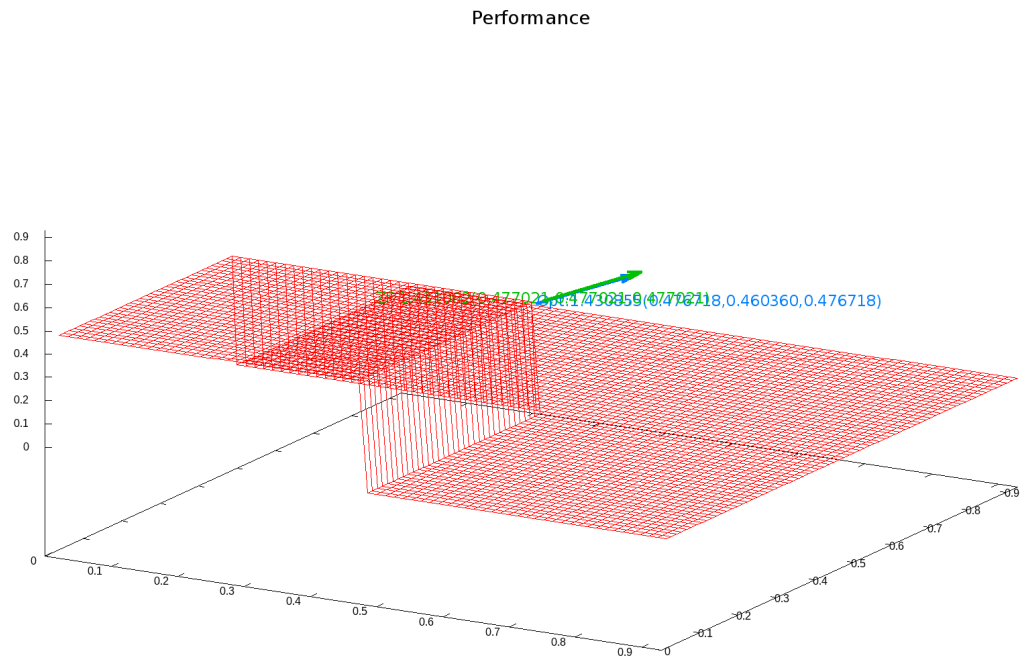


Рис. 9: Ортогональный канал

$$H = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}$$

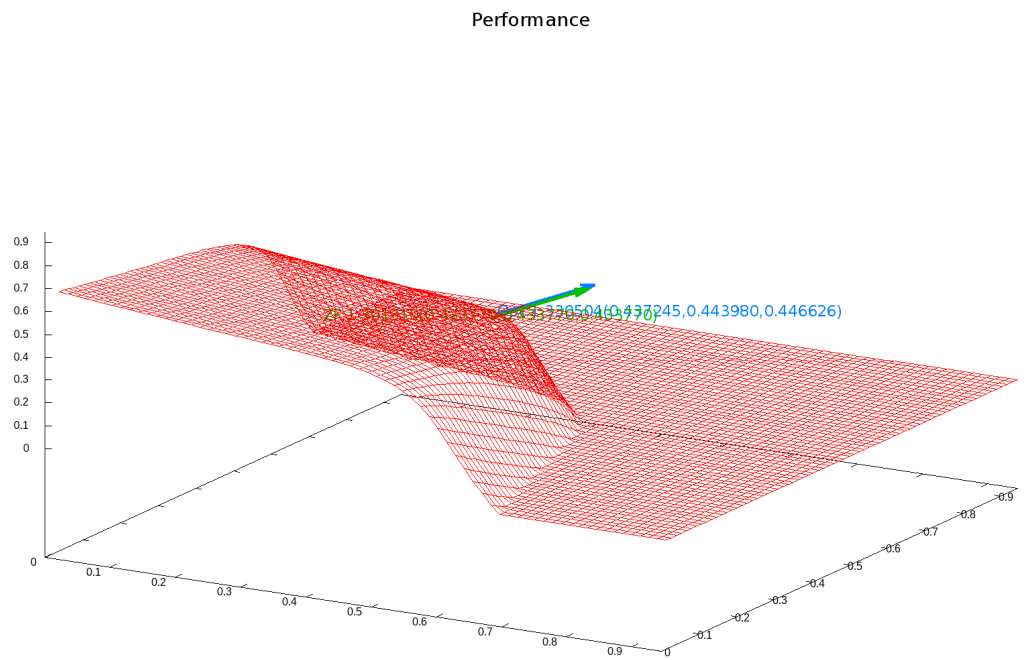


Рис. 10: Малая интерференция

$$H = [random \in [0, 1]]$$

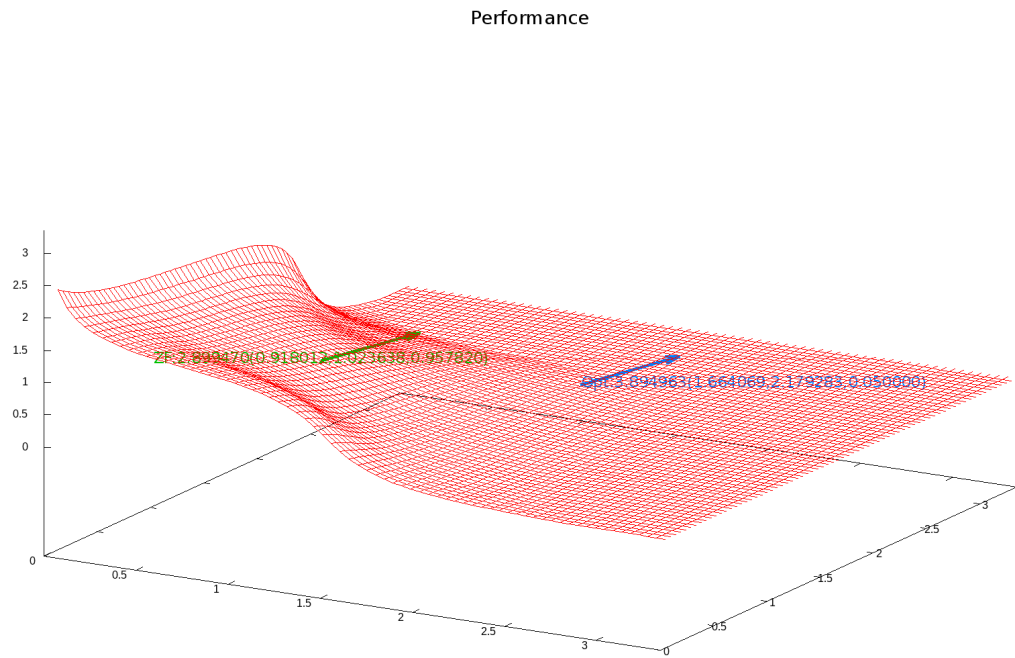


Рис. 11: Случайный канал

$$H = \begin{bmatrix} 1 & 0.7 & 0.7 \\ 0.7 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{bmatrix}$$

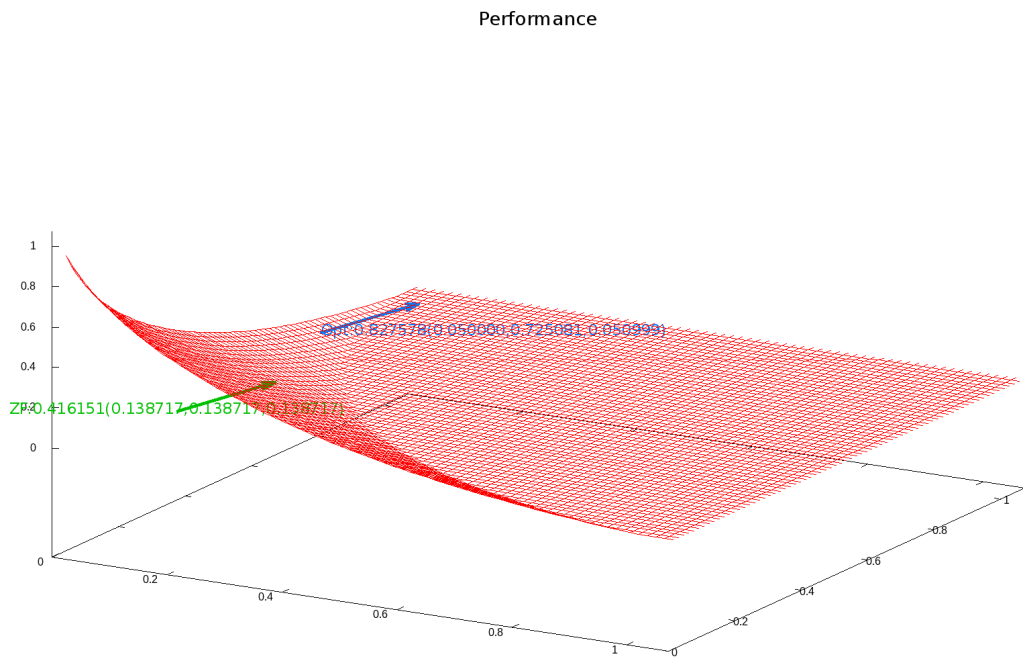


Рис. 12: Значительная интерференция

## 6.4 Сходимость метода ветвей и границ

Вычисления проводились последовательно на одном процессоре. Вместо линейного поиска в методе BRB для ускорения применялась модификация метода внутренней точки из 3.8, что сокращало число требуемых вызовов fixedQoS до 2-3 для одной стадии *Bound* метода ветвей границ. Подробнее об этой модификации в следующей секции.

Размерность задачи	Итераций BRB	Вызовов fixedQoS	Общее время
5	72	148	0.09 sec
8	975	1924	4 sec
10	4298	7963	29 sec

Таблица 1: Таблица требуемого числа итераций BRB и вызовов fixedQoS для решения общей задачи оптимизации D-MIMO с точностью  $10^{-3}$  в случае канала с низкой интерференцией.

## 6.5 Графические иллюстрации аппроксимации области допустимой производительности с помощью модификации метода внутренней точки

В этой секции рассматриваются численные эксперименты по аппроксимации границы допустимой области базисом из тригонометрических функций в полярных координатах (значение функции, задающей поверхность - расстояние от начала координат до пересечения луча, заданного двумя углами в полярных координатами, с границей допустимой области). Аппроксимация производилась методом наименьших квадратов. Точки на поверхности допустимой области были получены следующим алгоритмом. Был найден шар, целиком лежащий в допустимой области. Затем, на границе шара случайно было набрано фиксированное число точек (500 для численных экспериментов в этой секции). К каждой из них применялся модифицированный метод внутренней точки из 3.8 и генерировал новую, улучшенную точку в допустимой области. Если эта точка не была близко к границе (если точка, все координаты которой больше на некое малое  $\delta$ , все еще достижима), к новой точке повторно применялось улучшение с помощью метода внутренней точки и т.д. Такой алгоритм эффективнее линейного поиска по лучу, так как на практике требует меньше вызовов метода внутренней точки.

На следующих двух иллюстрациях можно видеть распределение полученных точек на границе достижимой области: (фиолетовые стрелки).

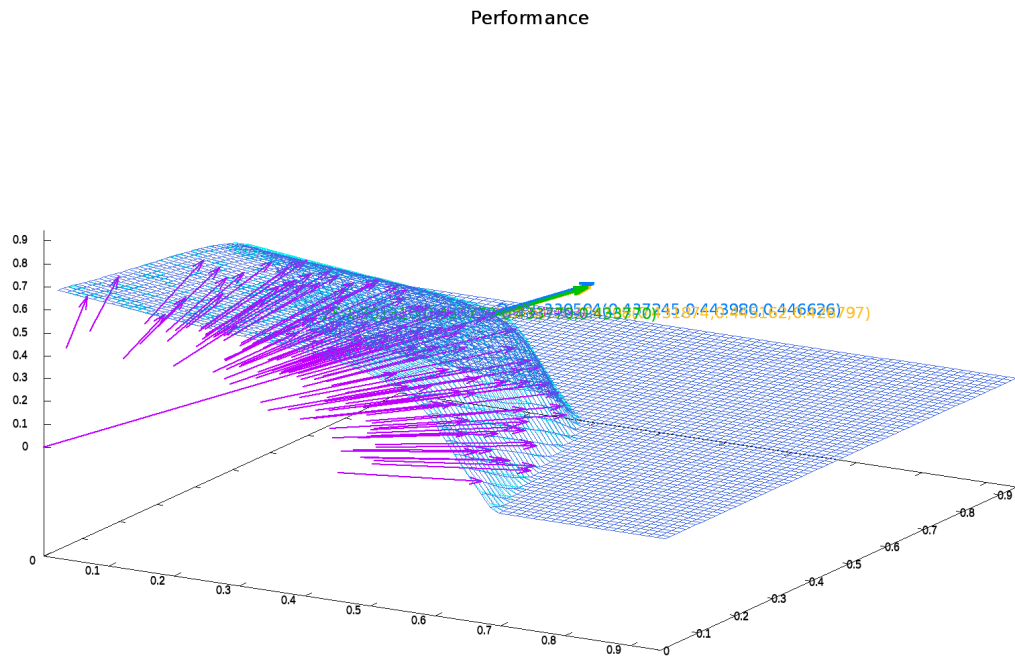


Рис. 13: Низкая интерференция

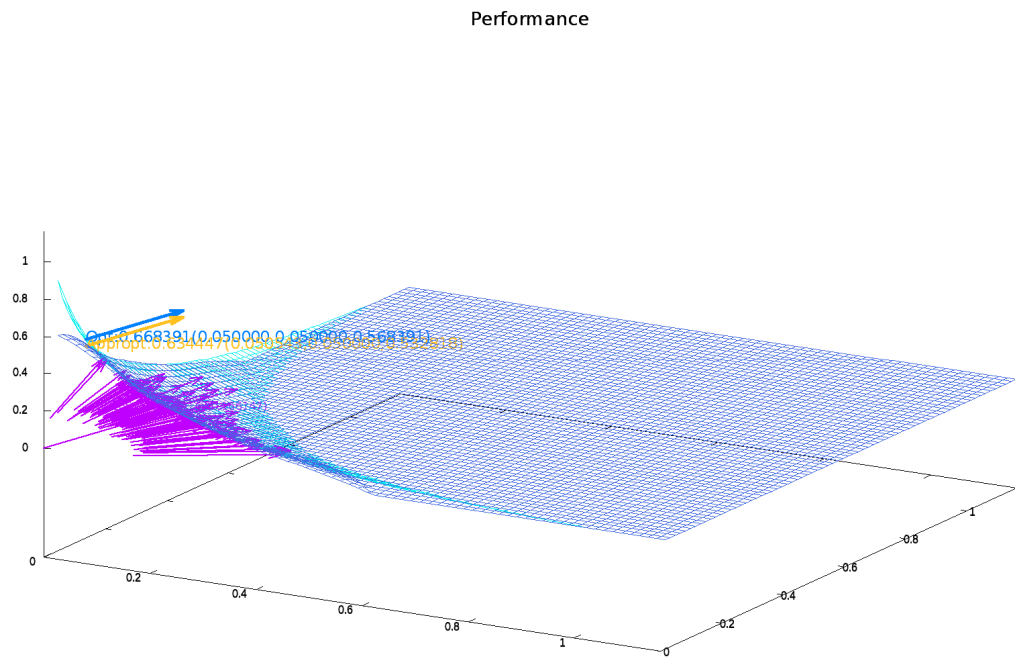


Рис. 14: Одинаковые каналы

Ниже проиллюстрировано сравнение настоящей области с ее аппроксимацией, а так же приведен график ошибки аппроксимации для двух различных возможных канальных матриц.

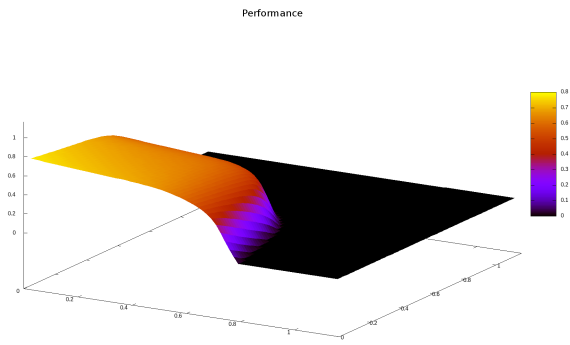


Рис. 15: Канал с низкой интерференцией

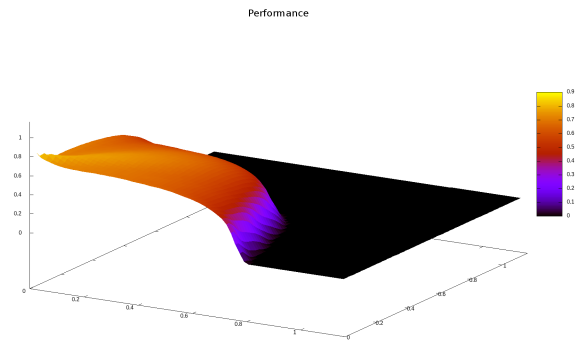


Рис. 16: Аппроксимация, 12 функций по каждой размерности

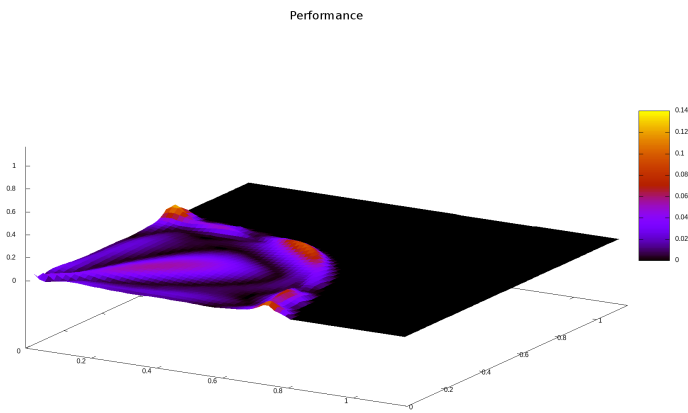


Рис. 17: График ошибки по оси Z

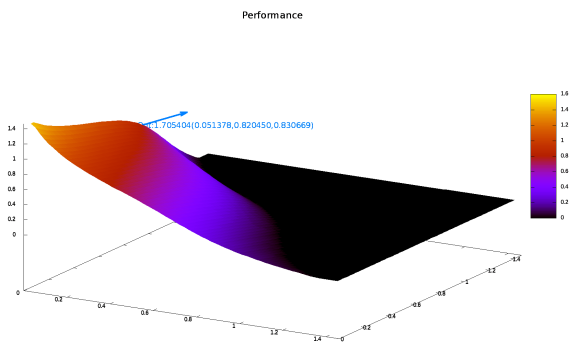


Рис. 18: Случайный канал, высокая интерференция

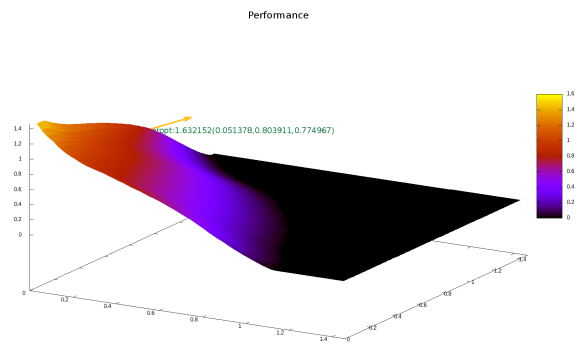


Рис. 19: Аппроксимация, 8 функций по каждой размерности

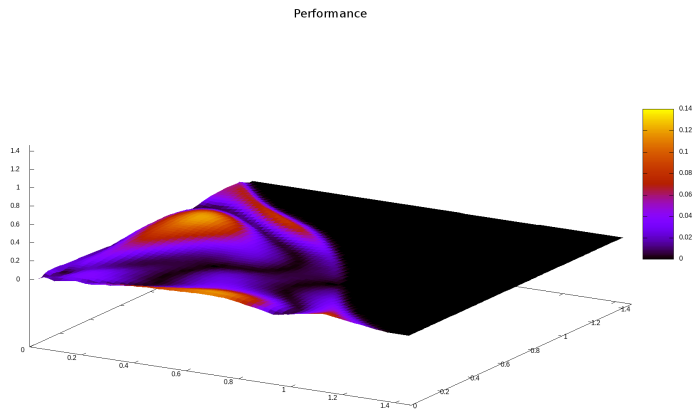


Рис. 20: График ошибки по оси Z