

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Сорокин Владимир Николаевич

**РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ
РЕШЕНИЯ МНОГОМЕРНЫХ МИНИМАКСНЫХ
ЗАДАЧ ТРОПИЧЕСКОЙ ОПТИМИЗАЦИИ**

Специальность 01.01.07 —

вычислительная математика

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физ.-мат. наук, доцент
Кривулин Николай Кимович

Санкт-Петербург — 2018

Оглавление

| | |
|---|-----------|
| Введение | 4 |
| 1 Задача с псевдоквадратичной целевой функцией и линейными ограничениями | 15 |
| 1.1 Элементы тропической математики | 16 |
| 1.1.1 Идемпотентное полуполе | 16 |
| 1.1.2 Алгебра матриц | 17 |
| 1.2 Линейные неравенства и их решения | 21 |
| 1.3 Задачи тропической оптимизации | 22 |
| 1.4 Задача оптимизации с ограничениями | 23 |
| 1.5 Оценка вычислительной сложности | 30 |
| 1.6 Численные примеры | 32 |
| 1.6.1 Задача без ограничений | 32 |
| 1.6.2 Задачи с ограничениями | 33 |
| 2 Практическое применение методов тропической оптимизации для планирование работ по дезактивации | 40 |
| 2.1 Задача ликвидатора | 41 |
| 2.2 Решение задачи ликвидатора | 44 |
| 2.3 Численный пример | 45 |
| 3 Задача псевдочebyшевской аппроксимации в тропическом векторном пространстве | 52 |
| 3.1 Задачи тропической оптимизации | 53 |
| 3.2 Предварительный анализ задачи | 55 |
| 3.3 Разрежение матрицы задачи | 57 |
| 3.4 Полное решение задачи | 61 |
| 3.5 Процедуры построения полного решения | 65 |

| | | |
|----------|--|------------|
| 3.6 | Алгоритм нахождения множества нижних границ | 68 |
| 3.7 | Численный пример | 69 |
| 4 | Метод построения множества всех решений тропического линейного векторного неравенства | 75 |
| 4.1 | Метод нахождения решений неравенства | 76 |
| 4.1.1 | Предварительная подготовка | 76 |
| 4.1.2 | Общая идея | 77 |
| 4.2 | Применение в задачах оптимизации | 78 |
| 4.2.1 | Линейная задача оптимизации | 79 |
| 4.2.2 | Варианты работы с нелинейными задачами | 79 |
| 4.3 | Пример использования метода | 80 |
| | Заключение | 82 |
| | Список литературы | 86 |
| | Список рисунков | 99 |
| A | Программная реализация задачи с псевдоквадратичной целевой функцией | 100 |
| B | Программная реализация расширенной задачи псевдочевышевской аппроксимации | 110 |
| B | Схема поиска решений неравенства | 120 |

Введение

Актуальность темы.

Тропическая (идемпотентная) математика представляет собой раздел прикладной математики, который изучает теорию полуколец и полуполей с идемпотентным сложением, а также связанные с ними вычислительные задачи. За последнее время тропическая математика стала одной из быстро развивающихся ветвей математической науки, роль которой в качестве инструмента эффективного решения задач в экономике, технике, управлении и других сферах человеческой деятельности постоянно растет. Сильная взаимосвязь с нечеткой и булевой логикой, дискретной математикой и методами оптимизации обеспечили ей надежные позиции в задачах оптимального управления и принятия решений.

Существует несколько основных причин для объяснения возросшего интереса к тропической математике. Во-первых, многие упомянутые задачи, в которых целевая функция в обычной математике является нелинейной и негладкой (и соответственно сложной для многих стандартных методов анализа), становятся линейными при переходе на язык идемпотентной алгебры. После этого решение таких задач может быть получено путем вычисления собственных чисел и векторов матриц или решения линейных неравенств и уравнений. Во-вторых, многие вычислительные процедуры линейной алгебры, такие, например, как алгоритм Якоби и метод Гаусса-Зейделя имеют свои идемпотентные аналоги, которые позволяют строить эффективные вычислительные алгоритмы тропической математики. При этом открываются новые возможности для анализа таких задач, что во многих случаях приводит к упрощению как процедур их численного решения, так и интерпретации полученных результатов.

Помимо этого, эволюция многих динамических систем, играющих важную роль в практических задачах (например, системы и сети с очередями), может быть представлена в терминах линейных уравнений идемпотентной алгебры,

что обеспечивает основу для разработки новых подходов к численному (имитационному) моделированию таких систем.

Тропическая математика имеет большое количество приложений к задачам оптимизации, включая задачи размещения, принятия решений, сетевого планирования и многим другим. Значительная часть этих задач может быть решена с использованием точных конечношаговых вычислительных методов, таких как методы линейного и смешанного целочисленного линейного программирования и т.п. В этих методах применяются итерационные процедуры, с помощью которых можно численно получить одно из решений, если решение существует, либо убедиться в отсутствии решений.

В отличие от решений с помощью указанных процедур, решения на основе методов тропической оптимизации во многих случаях предоставляют возможность нахождения всего множества решений в явном виде в простой матрично-векторной форме, удобной как для аналитического исследования множества решений, так и для создания алгоритмов численного решения. Полученное таким образом множество решений может быть напрямую использовано в других задачах в качестве области допустимых значений, обеспечивая композицию решений различных задач. Прямые явные решения позволяют проводить дальнейшее исследование множества решений математическими методами, изучать влияние дополнительных ограничений, точно определять трудоемкость нахождения решений, а также строить экономичные процедуры для последовательных и параллельных вычислений. Эти решения обычно представляют значительный интерес, что делает тему настоящей работы, направленной на разработку, обоснование и исследование эффективности прямых точных методов решения задач тропической оптимизации и их приложений, весьма актуальной.

Степень разработанности темы.

Понятие полукольца, которое является центральным понятием тропической математики, по-видимому, как указывается в работах [1, 2], впервые было введено Г. Вандивером в 1934 году в статье [3], хотя неявное использование присутствует уже в книге Р. Дедекинда [4] 1894 года и некоторых других работах. В дальнейшем на протяжении длительного времени полукольца и их различные аспекты исследовались многими математиками в связи с большим количеством теоретических и прикладных вычислительных задач, в которых эти полукольца

возникали. Однако начало наиболее активного развития тропической математики относится к 1950-60 годам, вскоре после публикаций работ С. К. Клини [5] в 1956, Р. А. Канингхейм-Грина [6] в 1962 и Н. Н. Воробьева [7–9] в 1963, 1967 и 1970 годах.

Если обратиться к истории развития тропической математики в России, то статья [8] была одной из первых на этапе ее становления. Появлению этой статьи предшествовали как различные теоретические работы (например по теории структур [10] и \mathcal{K} -пространств [11]), так и многочисленные возникшие практические задачи вычислительной математики. Как указывает в ней Н. Н. Воробьев: «...к построению такой теории приводит и необходимость обобщения ряда конкретных фактов, встречающихся в различных прикладных математических теориях. Достаточно сослаться на статьи А. Г. Лунца [12] и Н. Г. Поварова [13] по теории релейно-контактных схем, а также А. Шимбела [14], Р. Беллмана и У. Каруша [15] и др., исследовавших вопросы путей в графе.» ([8], стр. 42) Помимо проектирования контактных схем, подобные вычислительные задачи в том или ином виде возникли также во многих других областях народного хозяйства (см. например работу Л. В. Канторовича [16] 1942 года). В частности, «идеи экстремального гармонического анализа... в духе динамического программирования разбирает И. В. Романовский [17, 18]» ([8], стр. 42).

Помимо работ Н. Н. Воробьева стоит отметить статьи А. А. Корбута [19, 20], в которых проводятся дальнейшие исследование введенных в этих работах «экстремальных векторных пространств».

Различные аспекты идемпотентной математики изучались также С. Н. Самборским. В частности, в работе [21] рассматривается вопрос существования нетривиального спектра эндоморфизма над идемпотентным полумодулем, а также исследуется асимптотическое поведение при итерациях и сходимость «рядов Неймана», проявляющихся при решении уравнений $y = Ay \oplus f$.

Идемпотентный анализ в том смысле, в котором он понимается сейчас, был разработан научным коллективом академика В. П. Маслова [22–28] в восьмидесятых годах XX века в Москве [29].

В настоящее время в тропической математике существуют различные направления развития. Помимо применения чисто алгебраических методов, как, например, в работах Н. К. Кривулина [30–32] и С. Н. Сергеева в [33–35], суще-

ствуется и геометрический подход, связанный с изучением тропической геометрии, применяемый Г. Б. Михалкиным [36–38] и М. Э. Казаряном [39].

Существует широкий класс задач оптимизации, в которых целевая функция и ограничения выражаются при помощи операций максимума и минимума, а также арифметических операций. К этому классу относятся, например, некоторые задачи размещения [40–43] и сетевого планирования [6, 44–46]. Решение таких задач часто сопряжено с определенными трудностями, которые могут быть связаны, в частности, с нелинейностью и негладкостью целевой функции и ограничений.

Во многих случаях решение подобных задач можно упростить путем их представления на языке тропической математики и использования ее результатов [47–49]. Тропическая (идемпотентная) математика охватывает область, связанную с изучением теории полуколец с идемпотентным сложением и ее приложениями [22, 32, 40, 45, 50, 51]. Одним из направлений развития этой области является разработка вычислительных методов и алгоритмов решения задач оптимизации, которые могут быть сформулированы в терминах тропической математики (задач тропической оптимизации).

Изучению задач тропической оптимизации посвящен ряд исследований, опубликованных за последние несколько десятилетий. К числу таких публикаций относятся ранние работы [6, 44, 45], которые положили начало развитию этого направления, а также недавние работы [31, 42, 46, 52–58].

Существует класс задач оптимизации, которые формулируются в терминах тропической математики, состоят в минимизации нелинейных функционалов, и могут иметь ограничения в виде линейных векторных неравенств [30]. Решение некоторых таких задач опирается на экстремальное свойство спектрального радиуса матрицы и связано с его вычислением [31, 52, 54, 59].

Также имеется ряд практических задач (см., например, [8, 44, 60–62]), которые сводятся к наилучшему приближенному решению в смысле метрики Чебышева векторного уравнения $\mathbf{Ax} = \mathbf{p}$, где \mathbf{A} и \mathbf{p} обозначают заданные матрицу и вектор, \mathbf{x} – неизвестный вектор, а произведение матрицы на вектор понимается в смысле тропической математики.

Проблема чебышевской аппроксимации для решения рассматриваемого уравнения сводится к задаче нахождения векторов \mathbf{x} , на которых достигается

ся минимум в задаче

$$\min (\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{p}^- \mathbf{Ax},$$

где матричные и векторные операции понимаются в смысле идемпотентной алгебры.

Исследованию задачи посвящен ряд работ, опубликованных в различное время, включая работы [40, 44, 60–62]. Представленные в этих работах результаты обычно сводятся к нахождению одного из решений и не позволяют определить все множество решений задачи. Поэтому представляет интерес разработка методов получения всех решений в явном виде в компактной векторной форме и построение вычислительных процедур поиска всех решений.

Цели и задачи работы. Целью диссертации является исследование ряда задач тропической оптимизации для получение их полного решения в явном виде, разработка эффективных методов для численного нахождения соответствующих решений, а также реализация этих методов при решении прикладных задач, возникающих при математическом моделировании задач сетевого планирования. Для достижения указанной цели необходимо было сформулировать и решить следующие задачи:

1. Решить задачу с псевдоквадратичной целевой функцией и линейными ограничениями на множестве допустимых значений с использованием аппарата тропической математики для нахождения полного решения в явном виде в простой аналитической форме и представлении его в форме теоремы.
2. Реализовать численный метод, позволяющий найти решение этой задачи за полиномиальное по размерности задачи время.
3. Разработать математическую модель задачи планирования мероприятий по ликвидации последствий аварии с радиоактивным загрязнением местности, для решения которой может быть использована доказанная теорема и разработанный вычислительный метод.
4. Использовать аппарат идемпотентной математики и технику разрежения матриц для нахождения полного множества решений расширенной задачи псевдочebyшевской аппроксимации (которая при некоторых условиях

становится чебышевской) в тропическом векторном пространстве, а также реализовать численный метод получения этого множества.

5. Исследовать тропическое линейное векторное неравенство и разработать метод нахождения множества всех его решений.

Соответствие диссертации паспорту специальности. Содержание диссертационного исследования соответствует следующим пунктам паспорта специальности 01.01.07 – «Вычислительная математика»: создание алгоритмов численного решения задач алгебры, анализа, дифференциальных и интегральных уравнений, математической физики, теории вероятностей и статистики, типичных для приложений математики к различным областям науки и техники (пункт 1); разработка теории численных методов, анализ и обоснование алгоритмов, вопросы повышения их эффективности (пункт 2); реализация численных методов в решении прикладных задач, возникающих при математическом моделировании естественнонаучных и научно-технических проблем, соответствие выбранных алгоритмов специфике рассматриваемых задач (пункт 4).

Научная новизна. В диссертации обобщен ряд задач тропической оптимизации с псевдоквадратичной целевой функцией, решение которых получается применением разработанного точного конечношагового численного метода, в котором количество шагов известно заранее, а шаги представляют собой применение простых матрично-векторных операций. Полученные результаты обеспечивают дальнейшее развитие теории численных методов задач тропической оптимизации.

Найдено новое применение полученных результатов в реализации численных методов в решении прикладной задачи планирования операции по ликвидации последствий радиационной аварии при наличии ограничений на сроки выполнения работ.

Впервые получено полное решение задачи с псевдочебышевской метрикой, для которой был разработан точный численный метод получения множества всех решений. Предложен конечношаговый алгоритм, который используется для построения общего решения, представленного в компактной матрично-векторной форме.

Все основные результаты, выносимые на защиту, являются новыми и получены автором лично или при его определяющем участии.

Теоретическая и практическая ценность работы. Результаты работы имеют теоретическую ценность и могут быть использованы для решения реальных задач сетевого планирования. В частности, как было показано в главе 2, с их помощью можно оптимальным образом наметить план действий по ликвидации последствий чрезвычайной ситуации антропогенной природы.

Результатом работы стало получение полного решения для двух задач тропической оптимизации, которые могут быть использованы в комбинации с другими задачами и ограничениями. Матрично-векторная форма решений предполагает естественное распараллеливание вычислений.

Для задачи с псевдоквадратичной целевой функцией и линейными ограничениями решение записано в простой аналитической форме в удобном виде, что делает возможным проведение дальнейшего аналитического исследования математическими методами.

Методология и методы исследования. В работе применяются инструменты линейной алгебры, общей теории чисел, теории экстремальных задач, математического моделирования, а также методы оптимизации, теории сложности вычислений, компьютерного моделирования, построения математических моделей сложных систем и идемпотентной математики. Программирование велось на языке высокого уровня (R).

Положения, выносимые на защиту:

- Полностью решена задача с псевдоквадратичной целевой функцией и линейными ограничениями, решение получено в явном виде в простой аналитической форме с использованием матрично-векторных операций.
- Разработан точный конечношаговый численный метод построения решения этой задачи с полиномиальной по размерности задачи сложностью, где все шаги представляют собой выполнение простых матрично-векторных операций.
- Представлена математическая модель задачи сетевого планирования мероприятий по ликвидации чрезвычайной ситуации, которая решается путем применения разработанного численного метода.

- Решена расширенная задача псевдочebyшевской аппроксимации в тропическом векторном пространстве с использованием разрежения матрицы задачи. Разработан точный численный метод нахождения множества всех решений задачи, а также процедуры, позволяющие уменьшить вычислительную сложность этого метода. Предложен конечношаговый алгоритм, который используется для построения общего решения, представленного в компактной векторной форме.
- Получены результаты исследования линейного векторного неравенства, построена схема нахождения множества всех решений неравенства. Предложены варианты использования схемы как в линейной, так и в нелинейной задачах оптимизации в случаях, когда присутствуют ограничения на множество допустимых значений в форме рассматриваемого неравенства.

Степень достоверности изложенных в работе теоретических результатов обеспечивается их строгим математическим доказательством. Кроме того, достоверность результатов подтверждается сравнением с ранее известными результатами. В работе приводятся полные доказательства для теорем, доказанных диссертантом; для прочих использованных утверждений приведены ссылки на доказательства.

Апробация результатов. Результаты данной работы докладывались на международной конференции International Scientific Conference «Mathematical Modeling» (Borovets, Bulgaria – 2017); на I Международной научно-практической конференции «Теоретические и прикладные вопросы комплексной безопасности» (Санкт-Петербург, Россия – 2018); на семинарах кафедры статистического моделирования Санкт-Петербургского государственного университета, семинаре «Стохастическая оптимизация в информатике» СПбГУ и семинаре СПбГУ и СПб ЭМИ по тропической математике и смежным вопросам.

Результаты работы использовались при создании рабочего прототипа на хакатоне «EdHack: Chatbots and AI», проводившегося в рамках Международной конференции по новым образовательным технологиям EdCrunch (Москва, Россия – 2016).

Результаты диссертационной работы были получены при поддержке грантов Российского гуманитарного научного фонда РГНФ (№16-02-00059 – «Развитие

моделей и методов тропической математики в прикладных задачах экономики и управления» и №13-02-00338 – «Модели и методы тропической математики в прикладных задачах экономики и управления»), а также гранта Российского фонда фундаментальных исследований РФФИ №18-010-00723А – «Разработка моделей и методов тропической математики для прикладных задач экономики и управления».

Публикации. Основные результаты работы представлены в 2 печатных работах [63,64], опубликованных в рецензируемых научных изданиях, рекомендованных ВАК при Минобрнауки России, переводы которых [65,66] индексируются в международных библиографических базах Scopus и Web Of Science.

Всего по результатам диссертации автором опубликовано 5 печатных работ [63, 64, 67–69].

В совместных работах с Кривулиным Н. К. [63, 64, 67, 69] соискателю принадлежит формулировка и доказательства теорем о решении задачи с псевдоквадратичной целевой функцией и линейными ограничениями, а также расширенной задачи псевдочebyшевской аппроксимации в тропическом векторном пространстве, разработка алгоритмов и программных средств, проведение вычислительных экспериментов для верификации полученных результатов, соавтору принадлежат постановки задач и выбор методов решения.

Структура работы. Диссертационная работа состоит из введения, четырех глав, заключения и трех приложений. Полный объем диссертации составляет 123 страницы машинописного текста. Список литературы содержит 103 наименования.

Во введении обосновывается актуальность темы диссертации, приводится обзор соответствующей литературы, определены цели и задачи работы, аргументируется их научная ценность.

В главе 1 рассматривается задача с псевдоквадратичной целевой функцией и линейными ограничениями и разработан метод ее решения. В § 1.1 описываются основные понятия и результаты тропической математики, на которые опираются решения, представленные в остальной части работы. В § 1.2 приведены решения базовых векторных неравенств идемпотентной алгебры. В § 1.3 приведены известные задачи тропической математики, решение которых опирается на экстремальное свойство спектрального радиуса матрицы и связано с

его вычислением. В § 1.4 формулируется и решается обобщенная задача с псевдоквадратичной целевой функцией и линейными ограничениями, решение которой формулируется в виде теоремы и является основным результатом данной главы. Это решение представляет собой точный численный метод с конечным количеством шагов. В § 1.5 проводится оценка вычислительной сложности представленного численного метода и приводится схема вычислений, при которой решение находится за полиномиальное по размерности задачи время. В § 1.6 демонстрируются численные примеры решения вариантов задачи с различными ограничениями и приводится наглядная графическая иллюстрация.

В главе 2 рассматривается решение задачи планирования операции по ликвидации последствий радиационной аварии с помощью методов идемпотентной алгебры, которое получено путем применения теоремы, сформулированной и доказанной в предыдущей главе. В § 2.1 формулируется задача ликвидатора, которая заключается в составлении плана работ по ликвидации последствий аварии. Далее строится математическая модель, которая может быть использована для решения задачи различными способами, например, методами линейного программирования. Затем в § 2.2 задача формулируется в терминах идемпотентной математики, что позволяет применить для ее решения разработанный в главе 1 точный численный метод и получить решение в явном виде. В § 2.3 подробно рассматривается наглядный численный пример.

В главе 3 приводится решение расширенной задачи псевдочebyшевской аппроксимации в тропическом векторном пространстве. В § 3.1 проводится обзор известных результатов в данной области, из которого, в частности, следует, что в имеющихся исследованиях было получено только частичное решение задачи аппроксимации. В § 3.2 проводится анализ расширенной задачи псевдочebyшевской аппроксимации, вычисляется минимум целевой функции, предъявляется одно из решений и строится система неравенств, которая определяет множество всех решений. Далее в § 3.3 вводится понятие разреженной матрицы задачи, которая получается из исходной матрицы путём обнуления всех элементов, которые меньше некоторого порогового значения. Показывается, что замена матрицы задачи на разреженную не изменяет минимальное значение целевой функции и множество решений. В § 3.4 представлено полное решение расширенной задачи аппроксимации в виде семейства решений, которое задается множеством

матриц, полученных из разреженной матрицы задачи. Для этого сначала осуществляется расширение множества решений, а затем представляется полное решение расширенной задачи аппроксимации в виде семейства решений, которое задается множеством матриц, полученных из разреженной матрицы задачи путем дальнейшего обнуления ее элементов. В качестве прямого следствия этого результата получено полное решение задачи чебышевской аппроксимации в тропическом векторном пространстве. Перебор всевозможных матриц для построения подмножеств семейства решений задачи в соответствии с полученным результатом (для получения расчетной формулы) может представлять определенные трудности. Кроме того, некоторые семейства решений могут содержаться в других семействах и поэтому при записи общего решения могут быть отброшены. В § 3.5 описываются процедуры, позволяющие во многих случаях сократить число подмножеств, которые необходимо учесть при построении общего решения. В § 3.6 рассматривается алгоритм, необходимый для нахождения расчетной формулы. Численный пример, демонстрирующий решение задачи с применением описанных процедур, представлен в § 3.7.

В главе 4 проведено исследование тропического линейного векторного неравенства. Так как в явном виде множество всех решений для этой задачи получить не удастся ввиду сложности данного множества, то в § 4.1 предлагается метод, с помощью которого можно получить все решения алгоритмически. В § 4.2 рассматриваются возможности применения предложенного метода в линейной и нелинейной задачах тропической оптимизации, а в § 4.3 демонстрируется численный пример использования метода.

В заключении представлены основные результаты работы, а также рекомендации и предложения по дальнейшему проведению исследований по теме работы.

В приложении А представлено описание процедур и компьютерный код методов, применявшихся для расчетов в главах 1 и 2.

В приложении Б приводится компьютерный код методов, использованный для исследования расширенной задачи псевдочебышевской аппроксимации главы 3, а также для вычислений в примерах.

В приложении В представлено описание предлагаемой схемы решения тропического линейного векторного неравенства главы 4.

Глава 1

Задача с псевдоквадратичной целевой функцией и линейными ограничениями

Одна из задач оптимизации, которая рассматривалась еще в работе [6], связана с минимизацией функции, определенной на множестве вещественных векторов. В терминах тропического полукольца $\mathbb{R}_{\max,+}$, где максимум выступает в роли сложения, а арифметическое сложение в роли умножения, эта задача приобретает форму

$$\min \mathbf{x}^- \mathbf{A} \mathbf{x},$$

где \mathbf{A} – квадратная матрица, \mathbf{x} – неизвестный вектор, \mathbf{x}^- – вектор, полученный при помощи мультипликативно сопряженного транспонирования \mathbf{x} , а матрично-векторные операции определены аналогично стандартным с заменой обычных покомпонентных операций сложения и умножения на тропические.

Известно (см., например, [6]), что минимум в задаче совпадает с тропическим спектральным радиусом матрицы \mathbf{A} и достигается на любом собственном векторе, соответствующем этому радиусу. Полное решение задачи, которое оказывается шире, чем множество собственных векторов \mathbf{A} , найдено в работах [31, 52, 59].

В настоящей главе рассматривается дальнейшее обобщение задачи, в котором целевая функция имеет более сложную форму и введены дополнительные ограничения. Сначала представлен краткий обзор основных понятий и результатов тропической математики, необходимых для последующего анализа и решения задачи. Затем формулируется новая задача оптимизации и находится ее полное решение в явном виде в компактной векторной форме и проводит-

ся оценка вычислительной сложности. Приводятся числовые примеры решения задач на множестве двумерных векторов, и представлена графическая иллюстрация решений на плоскости в декартовой системе координат.

1.1 Элементы тропической математики

В этом разделе приводятся основные понятия и результаты тропической математики [32], на которые опираются решения, представленные в остальной части работы. Дополнительные детали и подробное изложение различных аспектов теории и методов тропической математики можно найти, например, в работах [22, 40, 50].

1.1.1 Идемпотентное полуполе

Рассмотрим набор $\langle \mathbb{X}, \oplus, \odot, \mathbb{0}, \mathbb{1} \rangle$, где \mathbb{X} – непустое множество, на котором определены операции сложения \oplus и умножения \odot . По сложению \mathbb{X} является идемпотентным коммутативным моноидом с нейтральным элементом $\mathbb{0}$ (нулем). По умножению множество $\mathbb{X} \setminus \{\mathbb{0}\}$ образует коммутативную группу с нейтральным элементом $\mathbb{1}$ (единицей).

Сложение идемпотентно: для любого $x \in \mathbb{X}$ выполняется $x \oplus x = x$. Идемпотентность сложения индуцирует частичный порядок на \mathbb{X} так, что $x \leq y$ тогда и только тогда, когда $x \oplus y = y$. Отсюда, в частности, следует, что неравенство $x \oplus y \leq z$ равносильно неравенствам $x \leq z$ и $y \leq z$, а сумма $x \oplus y$ не меньше любого из слагаемых: $x \leq x \oplus y$ и $y \leq x \oplus y$. Кроме того, операции \oplus и \odot монотонны в смысле указанного порядка по каждому аргументу, из чего следует, что неравенство $x \leq y$ влечет за собой неравенства $x \oplus z \leq y \oplus z$ и $x \odot z \leq y \odot z$ для любых $z \in \mathbb{X}$. Будем предполагать, что определенный таким образом частичный порядок может быть продолжен до полного порядка на \mathbb{X} .

Умножение дистрибутивно относительно сложения и обратимо для всех элементов, кроме $\mathbb{0}$: для любого $x \in \mathbb{X} \setminus \{\mathbb{0}\}$ существует обратный x^{-1} такой, что $x \odot x^{-1} = \mathbb{1}$. Операция обращения антитонна: это значит, что из неравенства $x \leq y$ вытекает неравенство $x^{-1} \geq y^{-1}$ для ненулевых x и y .

Естественным образом можно задать целые степени для любого ненулевого x и натурального p :

$$x^0 = \mathbb{1}, \quad x^p = x^{p-1} \odot x, \quad x^{-p} = (x^{-1})^p.$$

Учитывая, что множество \mathbb{X} не является группой по сложению, но его ненулевые элементы образуют группу по умножению, такая структура обычно называется идемпотентным полуполем. Будем считать, что уравнение $x^m = a$ разрешимо для любых $a \in \mathbb{X}$ и натуральных m , то есть, что введенная операция возведения в целую степень может быть распространена на случай рационального показателя степени.

Далее будем опускать знак умножения для упрощения записи. Знаки отношений и оператор \min будут пониматься в смысле порядка на \mathbb{X} , описанного выше.

Примерами идемпотентных полуполей на множестве вещественных чисел являются

$$\mathbb{R}_{\max,+} = \langle \mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0 \rangle,$$

$$\mathbb{R}_{\min,+} = \langle \mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0 \rangle,$$

$$\mathbb{R}_{\max,\times} = \langle \mathbb{R}_+ \cup \{0\}, \max, \times, 0, 1 \rangle,$$

$$\mathbb{R}_{\min,\times} = \langle \mathbb{R}_+ \cup \{+\infty\}, \min, \times, +\infty, 1 \rangle,$$

где $\mathbb{R}_+ = \{x \in \mathbb{R} | x > 0\}$. Дополнительные примеры могут быть найдены, в частности, в [22, 70–74].

Рассмотрим полуполе $\mathbb{R}_{\max,+}$. В нем роль нуля играет $-\infty$, а единицы — 0 . Для любого $x \in \mathbb{R}$ существует обратный по умножению x^{-1} , который равен противоположному числу $-x$ в обычной арифметике. Степень x^y определена для любых $x, y \in \mathbb{R}$ и соответствует арифметическому произведению xy . Порожденный сложением порядок на $\mathbb{R}_{\max,+}$ совпадает с естественным линейным порядком на \mathbb{R} .

1.1.2 Алгебра матриц

Обозначим через $\mathbb{X}^{m \times n}$ множество матриц над \mathbb{X} , состоящих из m строк и n столбцов. Матрица, все элементы которой равны $\mathbb{0}$, считается нулевой. Матри-

ца, у которой нет нулевых строк (столбцов) называется регулярной по строкам (столбцам).

Операции сложения и умножения матриц вводятся аналогично операциям в стандартной алгебре с заменой соответствующих покомпонентных операций на \oplus и \odot . Пусть $\mathbf{A} = (a_{ij})$, $\mathbf{B} = (b_{ij})$ и $\mathbf{C} = (c_{ij})$ — матрицы подходящего размера, а x — скаляр. Тогда

$$\{\mathbf{A} \oplus \mathbf{B}\}_{ij} = a_{ij} \oplus b_{ij}, \quad \{\mathbf{AC}\}_{ij} = \bigoplus_{k=1}^n a_{ik} c_{kj}, \quad \{x\mathbf{A}\}_{ij} = xa_{ij}.$$

Более подробную информацию о матрично-векторных операциях идемпотентной алгебры можно найти в работах [75–78].

Транспонирование матрицы \mathbf{A} обозначается через \mathbf{A}^T .

Мультипликативно сопряженным транспонированием матрицы $\mathbf{A} = (a_{ij})$ будем называть преобразование, при котором она трансформируется в транспонированную матрицу $\mathbf{A}^- = (a_{ij}^-)$ с элементами $a_{ij}^- = a_{ji}^{-1}$, если $a_{ji} \neq 0$ и $a_{ij}^- = 0$ в противном случае.

Рассмотрим квадратные матрицы из $\mathcal{X}^{n \times n}$. Обозначим через \mathbf{I} единичную матрицу, на главной диагонали которой стоят $\mathbb{1}$, а вне ее — 0 . Для любой квадратной матрицы \mathbf{A} и натурального p определим степень

$$\mathbf{A}^0 = \mathbf{I}, \quad \mathbf{A}^p = \mathbf{A}^{p-1} \mathbf{A}.$$

След квадратной матрицы $\mathbf{A} = (a_{ij})$ вычисляется по формуле

$$\text{tr } \mathbf{A} = a_{11} \oplus \cdots \oplus a_{nn}.$$

Для любых двух матриц \mathbf{A} и \mathbf{B} , а также скаляра x из определения следа вытекают равенства

$$\text{tr}(\mathbf{A} \oplus \mathbf{B}) = \text{tr } \mathbf{A} \oplus \text{tr } \mathbf{B}, \quad \text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}), \quad \text{tr}(x\mathbf{A}) = x \text{tr } \mathbf{A}.$$

Биномиальное тождество [52] для матриц \mathbf{A} и \mathbf{B} из $\mathbb{X}^{n \times n}$, и натуральной степени m имеет следующий вид

$$(\mathbf{A} \oplus \mathbf{B})^m = \bigoplus_{k=1}^m \bigoplus_{i_0 + \dots + i_k = m-k} \mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k} \oplus \mathbf{B}^m.$$

Для проверки этого утверждения достаточно заметить, что при раскрытии скобок будут получаться всевозможные произведения из m сомножителей, k из которых равны \mathbf{A} , $m - k$ равны \mathbf{B} (при $k = m$ получим \mathbf{A}^m).

Нетрудно проверить справедливость тождества [52]

$$\bigoplus_{k=1}^m (\mathbf{A} \oplus \mathbf{B})^k = \bigoplus_{k=1}^m \bigoplus_{0 \leq i_0 + \dots + i_k \leq m-k} \mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k} \oplus \bigoplus_{k=1}^m \mathbf{B}^k. \quad (1.1)$$

Здесь вначале были сгруппированы в одну сумму все слагаемые по степеням матрицы \mathbf{A} , а затем добавлена сумма степеней \mathbf{B} .

Как обычно, матрица, состоящая из одного столбца или строки, образует вектор. Если не оговорено иначе, будем рассматривать векторы как вектор-столбцы. Множество вектор-столбцов размерности n с элементами из \mathbb{X} обозначается \mathbb{X}^n . Вектор считается регулярным, если у него отсутствуют нулевые компоненты.

Мультипликативно сопряженным транспонированием вектора $\mathbf{x} = (x_j)$ называется преобразование, при котором \mathbf{x} трансформируется в вектор-строку $\mathbf{x}^- = (x_j^-)$ с элементами $x_j^- = x_j^{-1}$, если $x_j \neq 0$ и $x_j^- = 0$ в противном случае.

Для регулярных векторов \mathbf{x} и \mathbf{y} выполняется следующее соотношение [32]:

$$\mathbf{x} \mathbf{y}^- \geq (\mathbf{x}^- \mathbf{y})^{-1} \mathbf{I}. \quad (1.2)$$

Отсюда следует, что если вектор \mathbf{x} является регулярным, то верно следующее неравенство:

$$\mathbf{x} \mathbf{x}^- \geq \mathbf{I}. \quad (1.3)$$

Для ненулевого вектора \mathbf{x} справедливо равенство [52, 79, 80]

$$\mathbf{x}^- \mathbf{x} = \mathbb{1}.$$

Для любых регулярных векторов $\mathbf{x}, \mathbf{y} \in \mathbb{X}^n$ определим функцию расстояния следующим образом:

$$d(\mathbf{x}, \mathbf{y}) = \mathbf{x}^- \mathbf{y} \oplus \mathbf{y}^- \mathbf{x}.$$

В дальнейшем будем называть такое расстояние псевдочebyшевской метрикой.

Заметим, что в рамках полуполя $\mathbb{R}_{\max,+}$ это выражение принимает вид

$$d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq n} |x_i - y_i|,$$

что соответствует метрике Чебышева.

Скаляр λ является собственным числом матрицы \mathbf{A} , если существует ненулевой вектор \mathbf{x} такой, что

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

Максимальное собственное число называется спектральным радиусом матрицы \mathbf{A} и вычисляется по формуле

$$\lambda = \bigoplus_{m=1}^n \text{tr}^{1/m}(\mathbf{A}^m).$$

Вектор, все компоненты которого равны $\mathbb{1}$, обозначается как $\mathbf{1} = (\mathbb{1}, \dots, \mathbb{1})^T$.

Выпуклой линейной комбинацией векторов $\mathbf{x}_1, \dots, \mathbf{x}_k$ называется выражение вида

$$u_1 \mathbf{x}_1 \oplus \dots \oplus u_k \mathbf{x}_k,$$

где числа u_1, \dots, u_k удовлетворяют условию

$$u_1 \oplus \dots \oplus u_k = \mathbb{1}.$$

Это условие означает, что $u_i \leq \mathbb{1}$ для всех индексов $i = 1, \dots, k$, и по крайней мере для одного i выполняется равенство $u_i = \mathbb{1}$.

1.2 Линейные неравенства и их решения

Приведем решения некоторых линейных неравенств, которые будут использованы ниже. Сначала предположим, что заданы матрица $\mathbf{A} \in \mathbb{X}^{m \times n}$ и регулярный вектор $\mathbf{d} \in \mathbb{X}^m$. Требуется найти все векторы $\mathbf{x} \in \mathbb{X}^n$, удовлетворяющие неравенству

$$\mathbf{Ax} \leq \mathbf{d}. \quad (1.4)$$

Решение задачи обеспечивается следующим утверждением, доказательство которого приводится, например, в работах [32, 59].

Лемма 1. *Для любой регулярной по столбцам матрицы \mathbf{A} и регулярного вектора \mathbf{d} , все решения неравенства (1.4) имеют вид*

$$\mathbf{x} \leq (\mathbf{d}^- \mathbf{A})^-.$$

Пусть теперь заданы матрица $\mathbf{A} \in \mathbb{X}^{n \times n}$ и вектор $\mathbf{b} \in \mathbb{X}^n$. Необходимо найти все регулярные векторы \mathbf{x} , для которых выполняется неравенство

$$\mathbf{Ax} \oplus \mathbf{b} \leq \mathbf{x}. \quad (1.5)$$

Чтобы записать решение задачи, сначала введем функцию, которая ставит в соответствие любой матрице $\mathbf{A} \in \mathbb{X}^{n \times n}$ скаляр по правилу

$$\text{Tr}(\mathbf{A}) = \text{tr} \mathbf{A} \oplus \dots \oplus \text{tr} \mathbf{A}^n.$$

При условии, что $\text{Tr}(\mathbf{A}) \leq \mathbf{1}$, введем оператор, известный также как «звезда Клини», который сопоставляет матрице \mathbf{A} матрицу

$$\mathbf{A}^* = \mathbf{I} \oplus \mathbf{A} \oplus \dots \oplus \mathbf{A}^{n-1}.$$

Решение неравенства (1.5) получено в [31] в следующей форме.

Теорема 1. *Для любой матрицы \mathbf{A} и вектора \mathbf{b} справедливы следующие утверждения:*

1. *Если $\text{Tr}(\mathbf{A}) \leq \mathbf{1}$, то все регулярные решения неравенства (1.5) имеют вид $\mathbf{x} = \mathbf{A}^* \mathbf{u}$, где \mathbf{u} – регулярный вектор такой, что $\mathbf{u} \geq \mathbf{b}$.*

2. Если $\text{Tr}(\mathbf{A}) > 1$, то регулярных решений не существует.

1.3 Задачи тропической оптимизации

Теперь рассмотрим ряд задач оптимизации, которые формулируются в терминах тропической математики, состоят в минимизации нелинейных функционалов, и могут иметь ограничения в виде линейных векторных неравенств [30]. Решение многих таких задач опирается на экстремальное свойство спектрального радиуса матрицы и связано с его вычислением [31, 52, 54, 59]. Это свойство состоит в том, что спектральный радиус матрицы определяет минимум функции, которая задается этой матрицей с использованием оператора мультипликативно сопряженного транспонирования следующим образом.

Пусть спектральный радиус матрицы $\mathbf{A} \in \mathbb{X}^{n \times n}$ равен λ . Рассмотрим задачу

$$\min \mathbf{x}^- \mathbf{A} \mathbf{x}, \quad (1.6)$$

где минимум берется на множестве всех регулярных векторов $\mathbf{x} \in \mathbb{X}^n$. Такая задача имеет приложения, например, в сетевом планировании [40, 59], оптимальном размещении объектов [40, 54, 81], принятии решений [55, 82] и в других областях.

Полное решение этой задачи приводится в [31, 52, 59] в следующем виде.

Лемма 2. Пусть \mathbf{A} – матрица со спектральным радиусом $\lambda > 0$. Тогда минимум в задаче (1.6) равен λ , а все регулярные решения задачи имеют вид

$$\mathbf{x} = (\lambda^{-1} \mathbf{A})^* \mathbf{u}, \quad \mathbf{u} \in \mathbb{X}^n.$$

Известны решения для вариантов задачи (1.6) с целевой функцией более общего вида. Пусть, например, в дополнение к матрице $\mathbf{A} \in \mathbb{X}^{n \times n}$ заданы векторы $\mathbf{p}, \mathbf{q} \in \mathbb{X}^n$ и скаляр $r \in \mathbb{X}$. Требуется найти все регулярные векторы $\mathbf{x} \in \mathbb{X}^n$, обеспечивающие

$$\min \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} \oplus r. \quad (1.7)$$

Справедливо следующее утверждение, которое было получено в [59].

Теорема 2. Пусть \mathbf{A} – матрица со спектральным радиусом $\lambda > 0$, а \mathbf{q} – регулярный вектор. Тогда минимум в задаче (1.7) равен

$$\mu = \lambda \oplus \bigoplus_{m=1}^n (\mathbf{q}^- \mathbf{A}^{m-1} \mathbf{p})^{1/(m+1)} \oplus r, \quad (1.8)$$

а все регулярные решения задачи имеют вид

$$\mathbf{x} = (\mu^{-1} \mathbf{A})^* \mathbf{u}, \quad \mu^{-1} \mathbf{p} \leq \mathbf{u} \leq \mu (\mathbf{q}^- (\mu^{-1} \mathbf{A})^*)^-. \quad (1.9)$$

Другой вариант расширения задачи на экстремальное свойство спектрального радиуса – добавление ограничений на множество допустимых значений.

Пусть заданы матрицы $\mathbf{A}, \mathbf{B} \in \mathbb{X}^{n \times n}$ и вектор $\mathbf{p} \in \mathbb{X}^n$. Необходимо определить множество всех регулярных векторов $\mathbf{x} \in \mathbb{X}^n$, на которых достигается минимум в задаче

$$\begin{aligned} \min \quad & \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p}, \\ & \mathbf{B} \mathbf{x} \leq \mathbf{x}. \end{aligned} \quad (1.10)$$

Теорема 3. Пусть \mathbf{A} – матрица со спектральным радиусом $\lambda > 0$, а \mathbf{B} – матрица, для которой $\text{Tr}(\mathbf{B}) \leq \mathbb{1}$. Тогда минимум в задаче (1.10) равен

$$\theta = \lambda \oplus \bigoplus_{k=1}^{n-1} \bigoplus_{1 \leq i_1 + \dots + i_k \leq n-k} \text{tr}^{1/k}(\mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k}),$$

а все регулярные решения имеют вид

$$\mathbf{x} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u}, \quad \mathbf{u} \geq \theta^{-1} \mathbf{p}.$$

Ниже будет предложено решение задачи, которая является дальнейшим обобщением задачи (1.10) с целевой функцией, которая определена также, как в задаче (1.7).

1.4 Задача оптимизации с ограничениями

В этом разделе изучается новая задача оптимизации с нелинейной целевой функцией и ограничениями в форме линейного неравенства [63, 65, 67]. Приме-

няется подход, развитый в [31, 52, 53, 59], при котором вводится дополнительная переменная, описывающая минимальное значение целевой функции. Затем задача сводится к решению неравенства, в котором эта переменная выступает в роли параметра. Необходимые и достаточные условия существования решений неравенства используются для вычисления параметра, а общее решение неравенства берется в качестве решения исходной задачи оптимизации.

Пусть заданы матрицы $\mathbf{A}, \mathbf{B} \in \mathbb{X}^{n \times n}$, векторы $\mathbf{p}, \mathbf{q} \in \mathbb{X}^n$ и скаляр $r \in \mathbb{X}$. Требуется найти все регулярные векторы $\mathbf{x} \in \mathbb{X}^n$, которые решают задачу

$$\begin{aligned} \min \quad & \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} \oplus r, \\ & \mathbf{B} \mathbf{x} \leq \mathbf{x}. \end{aligned} \quad (1.11)$$

Теорема 4. Пусть \mathbf{A} – матрица со спектральным радиусом $\lambda > 0$, а \mathbf{B} – матрица, для которой $\text{Tr}(\mathbf{B}) \leq 1$. Для любого натурального m и $k = 1, \dots, m$ введем обозначения

$$\mathbf{S}_{0m} = \bigoplus_{i=0}^m \mathbf{B}^i, \quad \mathbf{S}_{km} = \bigoplus_{0 \leq i_0 + \dots + i_k \leq m-k} \mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k}.$$

Тогда минимум в задаче (1.11) равен

$$\theta = r \oplus \bigoplus_{k=1}^m \text{tr}^{1/k}(\mathbf{S}_{kn}) \oplus \bigoplus_{k=0}^{m-1} (\mathbf{q}^- \mathbf{S}_{k, m-1} \mathbf{p})^{1/(k+2)}, \quad (1.12)$$

а все регулярные решения имеют вид

$$\mathbf{x} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u}, \quad \theta^{-1} \mathbf{p} \leq \mathbf{u} \leq \theta (\mathbf{q}^- (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^-. \quad (1.13)$$

Доказательство. Сначала заметим (по лемме 2), что

$$\mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} \oplus r \geq \mathbf{x}^- \mathbf{A} \mathbf{x} \geq \lambda > 0,$$

откуда следует, что целевая функция в (1.11) ограничена снизу. Обозначим минимум целевой функции на множестве всех регулярных векторов \mathbf{x} через θ .

Тогда все регулярные решения задачи (1.11) получаются из системы

$$\begin{aligned} \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} \oplus r &= \theta, \\ \mathbf{B} \mathbf{x} &\leq \mathbf{x}. \end{aligned}$$

Так как θ по предположения – минимум целевой функции, то можно заметить равенство на неравенство

$$\begin{aligned} \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} \oplus r &\leq \theta, \\ \mathbf{B} \mathbf{x} &\leq \mathbf{x}. \end{aligned} \tag{1.14}$$

Первое неравенство в (1.14) равносильно системе неравенств

$$\begin{aligned} \mathbf{x}^- \mathbf{A} \mathbf{x} &\leq \theta, \\ \mathbf{x}^- \mathbf{p} &\leq \theta, \\ \mathbf{q}^- \mathbf{x} &\leq \theta, \\ r &\leq \theta. \end{aligned}$$

Перемножив соответствующие части второго и третьего неравенств, получим соотношение

$$\mathbf{q}^- \mathbf{p} \leq \mathbf{q}^- \mathbf{x} \mathbf{x}^- \mathbf{p} \leq \theta^2.$$

Следовательно, $\theta \geq (\mathbf{q}^- \mathbf{p})^{1/2}$. Учитывая четвертое неравенство и то, что

$$\theta \geq \mathbf{x}^- \mathbf{A} \mathbf{x} \geq \lambda,$$

находим нижнюю границу для θ в форме

$$\theta \geq \lambda \oplus (\mathbf{q}^- \mathbf{p})^{1/2} \oplus r.$$

Применив лемму 1 к первым трем неравенствам рассматриваемой системы, а затем умножая первые два из полученных неравенств на θ^{-1} , получаем

$$\begin{aligned} \theta^{-1} \mathbf{A} \mathbf{x} &\leq \mathbf{x}, \\ \theta^{-1} \mathbf{p} &\leq \mathbf{x}, \\ \mathbf{x} &\leq \theta \mathbf{q}. \end{aligned}$$

Наконец, объединяя эти неравенства с неравенством $\mathbf{B}\mathbf{x} \leq \mathbf{x}$, запишем систему (1.14) в виде двойного неравенства

$$(\theta^{-1}\mathbf{A} \oplus \mathbf{B})\mathbf{x} \oplus \theta^{-1}\mathbf{p} \leq \mathbf{x} \leq \theta\mathbf{q}. \quad (1.15)$$

По теореме 1 существование регулярных решений для левой части неравенства (1.15) равносильно выполнению условия

$$\mathrm{Tr}(\theta^{-1}\mathbf{A} \oplus \mathbf{B}) \leq \mathbb{1}.$$

Рассмотрим выражение

$$\mathrm{Tr}(\theta^{-1}\mathbf{A} \oplus \mathbf{B}) = \bigoplus_{k=1}^n \mathrm{tr}(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^k = \mathrm{tr} \left(\bigoplus_{k=1}^n (\theta^{-1}\mathbf{A} \oplus \mathbf{B})^k \right).$$

Сначала, применяя тождество (1.1) при $m = n$, запишем

$$\bigoplus_{k=1}^n (\theta^{-1}\mathbf{A} \oplus \mathbf{B})^k = \bigoplus_{k=1}^n \bigoplus_{0 \leq i_0 + \dots + i_k \leq n-k} \theta^{-k} (\mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k}) \oplus \bigoplus_{k=1}^n \mathbf{B}^k.$$

Теперь с учетом обозначения \mathbf{S}_{kn} получим

$$\mathrm{Tr}(\theta^{-1}\mathbf{A} \oplus \mathbf{B}) = \bigoplus_{k=1}^n \theta^{-k} \mathrm{tr} \mathbf{S}_{kn} \oplus \mathrm{Tr}(\mathbf{B}).$$

Заметим, что неравенство $\mathrm{Tr}(\mathbf{B}) \leq \mathbb{1}$ выполнено по условиям теоремы, поэтому требуется обеспечить выполнение неравенства

$$\bigoplus_{k=1}^n \theta^{-k} \mathrm{tr} \mathbf{S}_{kn} \leq \mathbb{1}.$$

Последнее неравенство эквивалентно системе неравенств

$$\theta^{-k} \mathrm{tr} \mathbf{S}_{kn} \leq \mathbb{1}, \quad k = 1, \dots, n.$$

Решение неравенств системы приводит к системе

$$\mathrm{tr}^{1/k}(\mathbf{S}_{kn}) \leq \theta, \quad k = 1, \dots, n,$$

которая, в свою очередь, равносильна одному неравенству

$$\theta \geq \bigoplus_{k=1}^n \mathrm{tr}^{1/k}(\mathbf{S}_{kn}).$$

Нетрудно заметить, что $\mathbf{S}_{kn} \geq \mathbf{A}^k$ для всех $k = 1, \dots, n$, откуда следует

$$\bigoplus_{k=1}^n \mathrm{tr}^{1/k}(\mathbf{S}_{kn}) \geq \bigoplus_{k=1}^n \mathrm{tr}^{1/k}(\mathbf{A}^k) = \lambda.$$

Тогда можно уточнить установленную ранее нижнюю границу для θ так

$$\theta \geq r \oplus (\mathbf{q}^- \mathbf{p})^{1/2} \oplus \bigoplus_{k=1}^n \mathrm{tr}^{1/k}(\mathbf{S}_{kn}).$$

Теперь необходимо получить решение неравенства (1.15). Применяя теорему 1, находим решение левой части неравенства в виде

$$\mathbf{x} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u},$$

где \mathbf{u} – любой регулярный вектор такой, что $\mathbf{u} \geq \theta^{-1} \mathbf{p}$.

С учетом полученного решения правое неравенство в (1.15) принимает форму

$$(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u} \leq \theta \mathbf{q}.$$

По лемме 1 решение этого неравенства записывается в виде

$$\mathbf{u} \leq \theta(\mathbf{q}^-(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^-.$$

Объединив оба неравенства для \mathbf{u} , имеем

$$\theta^{-1} \mathbf{p} \leq \mathbf{u} \leq \theta(\mathbf{q}^-(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^-.$$

Выясним, для каких значений θ множество регулярных решений полученного неравенства не пусто. Необходимо решить относительно θ неравенство

$$\theta^{-1}\mathbf{p} \leq \theta(\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*)^-. \quad (1.16)$$

Умножая неравенство (1.16) слева на

$$\theta^{-1}\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*,$$

приходим к неравенству

$$\theta^{-2}\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*\mathbf{p} \leq \mathbb{1}. \quad (1.17)$$

Теперь покажем, что неравенство (1.16) в свою очередь тоже является следствием (1.17). Для этого умножим неравенство (1.17) слева на

$$\theta(\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*)^-,$$

а затем применим неравенство (1.3). В результате получим

$$\theta^{-1}\mathbf{p} \leq \theta^{-1}(\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*)^-\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*\mathbf{p} \leq \theta(\mathbf{q}^-(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*)^-,$$

откуда следует, что оба неравенства эквивалентны.

Рассмотрим неравенство (1.17). Учитывая, что $\text{Tr}(\theta^{-1}\mathbf{A} \oplus \mathbf{B}) \leq \mathbb{1}$, можно записать

$$(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^* = \bigoplus_{m=0}^{n-1} (\theta^{-1}\mathbf{A} \oplus \mathbf{B})^m = \mathbf{I} \oplus \bigoplus_{m=1}^{n-1} (\theta^{-1}\mathbf{A} \oplus \mathbf{B})^m.$$

Также, как и в первой части доказательства, применим тождество (1.1) при $m = n - 1$. Используя обозначение $\mathbf{S}_{k,n-1}$, запишем

$$(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^* = \mathbf{I} \oplus \bigoplus_{k=1}^{n-1} \theta^{-k} \mathbf{S}_{k,n-1} \oplus \bigoplus_{k=1}^{n-1} \mathbf{B}^k.$$

Учитывая условия теоремы $\text{Tr}(\mathbf{B}) \leq \mathbb{1}$, имеем

$$\mathbf{I} \oplus \bigoplus_{k=1}^{n-1} \mathbf{B}^k = \bigoplus_{k=0}^{n-1} \mathbf{B}^k = \mathbf{S}_{0,n-1} = \mathbf{B}^*,$$

откуда следует, что

$$(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* = \bigoplus_{k=0}^{n-1} \theta^{-k} \mathbf{S}_{k,n-1}.$$

Подставляя полученное выражение в неравенство (1.17), приходим к неравенству

$$\bigoplus_{k=0}^{n-1} \theta^{-k-2} \mathbf{q}^- \mathbf{S}_{k,n-1} \mathbf{p} \leq \mathbb{1}.$$

Решая это неравенство относительно θ тем же путем, что и выше, получим

$$\theta \geq \bigoplus_{k=0}^{n-1} (\mathbf{q}^- \mathbf{S}_{k,n-1} \mathbf{p})^{1/(k+2)}.$$

Заметим, что при $k = 0$ правая часть неравенства равна

$$(\mathbf{q}^- \mathbf{B}^* \mathbf{p})^{1/2} \geq (\mathbf{q}^- \mathbf{p})^{1/2}.$$

Объединив все нижние границы, установленные для θ , можем записать неравенство

$$\theta \geq r \oplus \bigoplus_{k=1}^n \text{tr}^{1/k}(\mathbf{S}_{kn}) \oplus \bigoplus_{k=0}^{n-1} (\mathbf{q}^- \mathbf{S}_{k,n-1} \mathbf{p})^{1/(k+2)}.$$

Чтобы получить минимум целевой функции, заменим в этом соотношении знак неравенства на знак равенства. Осталось записать общее решение в форме

$$\mathbf{x} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u}, \quad \theta^{-1} \mathbf{p} \leq \mathbf{u} \leq \theta (\mathbf{q}^- (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^-,$$

и тем самым завершить доказательство теоремы. \square

Таким образом, было получено все множество решений задачи (1.11), что позволяет проводить фильтрацию найденных решений по различным критериям, а также дает возможность добавлять дополнительные ограничения или использовать задачу (1.11) в композиции с другими задачами. Заметим, что

в терминах полуполя $\mathbb{R}_{\max,+}$ рассматриваемая задача может быть записана, как задача линейного программирования. Однако решение задачи методами линейного программирования (например, симплекс-методом) приводит к итеративной процедуре нахождения решения, а не к прямым расчетным формулам для непосредственного нахождения решения. Кроме того, явные формулы позволяют точно определить количество операций, необходимых для получения результата, в то время как итеративные подходы допускают только оценку вычислительной сложности.

1.5 Оценка вычислительной сложности

В этом разделе производится оценка вычислительной сложности решения задачи (1.11), а также предлагается метод, позволяющий существенно уменьшить количество необходимых арифметических операций.

Основная сложность при решении задачи (1.11) состоит в необходимости вычисления слагаемых \mathbf{S}_{km} по формулам

$$\mathbf{S}_{0m} = \bigoplus_{i=0}^m \mathbf{B}^i, \quad \mathbf{S}_{km} = \bigoplus_{0 \leq i_0 + \dots + i_k \leq m-k} \mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k},$$

для всех $k = 1, \dots, m$. При этом общее количество слагаемых, входящих в указанные суммы составляет величину порядка 2^n .

Таким образом, при прямом подсчете θ вычислительная сложность оказывается экспоненциальной. Далее представлена схема вычислений, которая позволяет существенно снизить количество операций по сравнению с прямым подходом.

Обозначим через \mathbf{M}_{ki} сумму всевозможных произведений, состоящих из k матриц \mathbf{A} и $i - k$ матриц \mathbf{B} . При этом $\mathbf{M}_{0i} = \mathbf{B}^i$, $\mathbf{M}_{ii} = \mathbf{A}^i$, $\mathbf{M}_{00} = \mathbf{I}$. С учетом этих обозначений можно выразить суммы \mathbf{S}_{km} в виде

$$\mathbf{S}_{0m} = \bigoplus_{i=0}^m \mathbf{M}_{0i}, \quad \mathbf{S}_{km} = \bigoplus_{i=k}^m \mathbf{M}_{ki}, \quad k = 1, \dots, m. \quad (1.18)$$

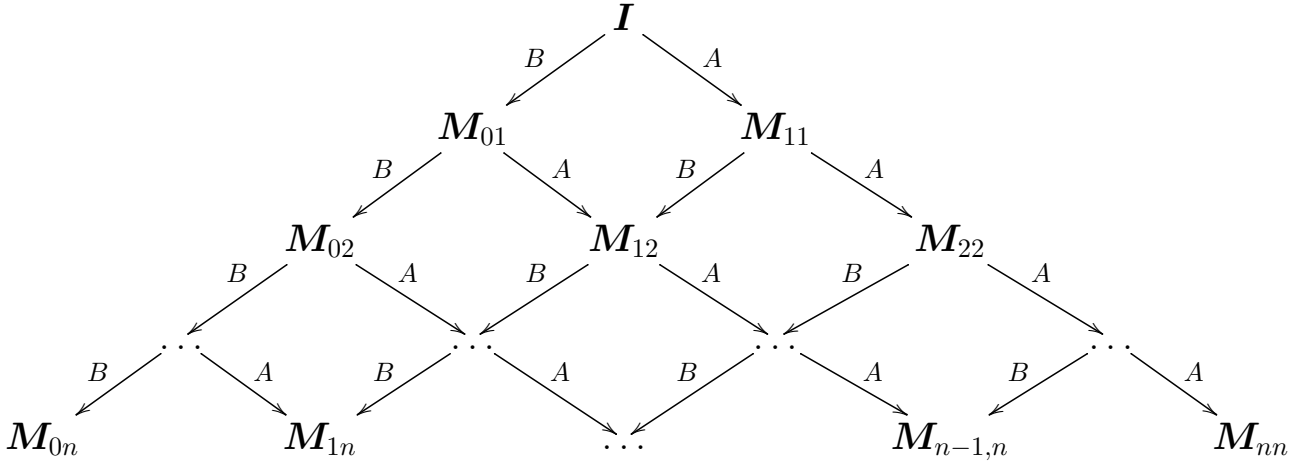


Рисунок 1.1: Схема подсчета слагаемых M_{km} .

Заметим, что матрицу M_{ki} (при $1 \leq k \leq i-1$), можно получить с помощью следующей рекуррентной формулы:

$$M_{ki} = BM_{k,i-1} \oplus AM_{k-1,i-1} = M_{k,i-1}B \oplus M_{k-1,i-1}A.$$

На основе этой формулы предлагается метод для подсчета слагаемых: поочередно вычисляются слагаемые M_{ki} , после чего по формулам (1.18) находятся все необходимые суммы S_{km} , и, соответственно, минимум θ целевой функции. Наглядная иллюстрация подобной схемы изображена на рис. 1.1.

Общее количество матриц M_{ki} , необходимых для вычисления θ , составляет величину $1 + \dots + (n+1) = (n+1)(n+2)/2$. Сложность нахождения слагаемых M_{ki} не превосходит $\mathcal{O}(n^2)$ операций с матрицами, а итоговая сложность нахождения всех регулярных решений задачи (1.11) оказывается полиномиальной. При использовании прямого метода перемножения матриц (сложность не более $\mathcal{O}(n^3)$), общая вычислительная сложность решения задачи не превосходит $\mathcal{O}(n^5)$.

Отсюда следует, что полученное решение не только является полным, но к тому же может быть представлено в простой аналитической форме в матрично-векторном виде. Это дает возможность проводить дальнейшее аналитическое исследование задачи, а также обеспечивает естественность программной реализации и распараллеливания вычислений.

1.6 Численные примеры

Чтобы проиллюстрировать полученные результаты, приведем примеры их использования в терминах полуполя $\mathbb{R}_{\max,+}$. Сначала рассмотрим задачу без ограничений, решение которой затем распространим на случай задачи с ограничениями.

1.6.1 Задача без ограничений

Пусть в задаче (1.11) отсутствуют ограничения и тогда она принимает форму (1.7). Предположим, что $n = 2$, а матрица \mathbf{A} , векторы \mathbf{p} и \mathbf{q}^- , а также скаляр r заданы в виде

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}, \quad \mathbf{q}^- = \begin{pmatrix} 1 & -1 \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad r = 2.$$

Для решения задачи применим теорему 2. Сначала по формуле (1.8) найдем минимум целевой функции

$$\mu = \lambda \oplus (\mathbf{q}^- \mathbf{p})^{1/2} \oplus (\mathbf{q}^- \mathbf{A} \mathbf{p})^{1/3} \oplus r.$$

Для этого вычислим

$$\mathbf{A}^2 = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 7 & 8 \end{pmatrix}, \quad \lambda = 4,$$

$$\mathbf{q}^- \mathbf{p} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2, \quad (\mathbf{q}^- \mathbf{p})^{1/2} = 1,$$

$$\mathbf{q}^- \mathbf{A} \mathbf{p} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4, \quad (\mathbf{q}^- \mathbf{A} \mathbf{p})^{1/3} = 4/3.$$

Отсюда получаем $\mu = 4$. Также подсчитаем

$$\mu^{-1} \mathbf{A} = \begin{pmatrix} -3 & -4 \\ -1 & 0 \end{pmatrix}, \quad (\mu^{-1} \mathbf{A})^* = \begin{pmatrix} 0 & -4 \\ -1 & 0 \end{pmatrix},$$

$$\mu^{-1}\mathbf{p} = \begin{pmatrix} -3 \\ -3 \end{pmatrix}, \quad \mu(\mathbf{q}^-(\mu^{-1}\mathbf{A})^*)^- = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$

Применяя формулу (1.9), находим решение в виде

$$\mathbf{x} = \begin{pmatrix} 0 & -4 \\ -1 & 0 \end{pmatrix} \mathbf{u}, \quad \begin{pmatrix} -3 \\ -3 \end{pmatrix} \leq \mathbf{u} \leq \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$

Переходя к обычной записи, положив

$$\mathbf{x} = (x_1, x_2)^T, \quad \mathbf{u} = (u_1, u_2)^T,$$

получаем следующие равенства:

$$\begin{aligned} x_1 &= \max(u_1, u_2 - 4), \\ x_2 &= \max(u_1 - 1, u_2), \end{aligned}$$

где компоненты вектора \mathbf{u} ограничены неравенствами

$$\begin{aligned} -3 &\leq u_1 \leq 3, \\ -3 &\leq u_2 \leq 5. \end{aligned}$$

Графическая иллюстрация решения в декартовой системе координат дана на рис. 1.2. Множество решений образует многоугольную область со штрихованными границами, которая получена пересечением полосы между двумя сплошными прямыми линиями, проведенными под углом 45° к координатным осям и прямоугольника, границы которого изображены пунктиром.

1.6.2 Задачи с ограничениями

Найдем решение задачи с ограничениями в виде (1.11). В качестве примера рассмотрим предыдущую задачу, в которой добавлено ограничение $\mathbf{B}\mathbf{x} \leq \mathbf{x}$ с матрицей ограничений

$$\mathbf{B} = \begin{pmatrix} 0 & -1 \\ -2 & 0 \end{pmatrix}.$$

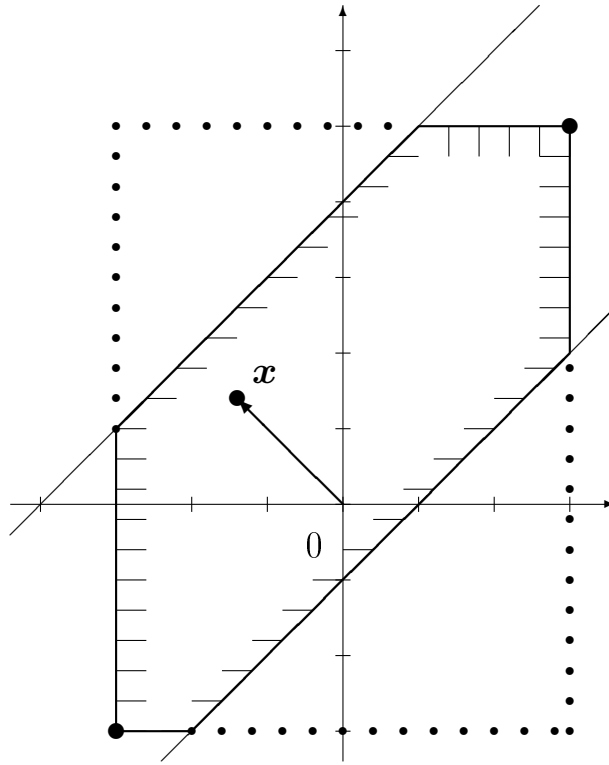


Рисунок 1.2: Пример множества решений задачи без ограничений.

Чтобы применить теорему 4, сначала вычислим

$$\mathbf{B}^2 = \mathbf{B} = \mathbf{B}^*, \quad \text{Tr}(\mathbf{B}) = 0 = \mathbb{1},$$

условие $\text{Tr}(\mathbf{B}) \leq \mathbb{1}$ выполнено.

Найдем матрицы

$$\mathbf{A}\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ -2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} = \mathbf{A},$$

$$\mathbf{B}\mathbf{A} = \begin{pmatrix} 0 & -1 \\ -2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}.$$

Применение формулы (1.12) для вычисления минимума целевой функции дает выражение

$$\theta = r \oplus \text{tr}(\mathbf{S}_{12}) \oplus \text{tr}^{1/2}(\mathbf{S}_{22}) \oplus (\mathbf{q}^- \mathbf{S}_{01} \mathbf{p})^{1/2} \oplus (\mathbf{q}^- \mathbf{S}_{11} \mathbf{p})^{1/3},$$

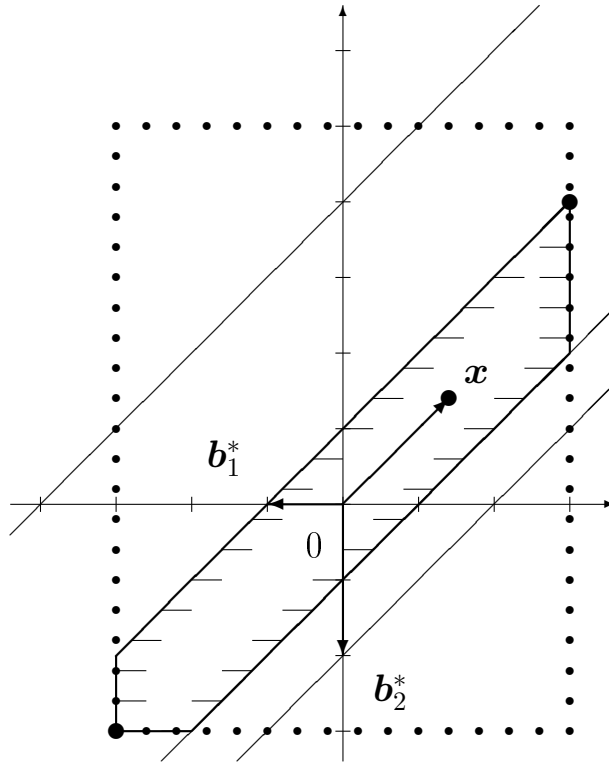


Рисунок 1.3: Пример множества решений задачи с ограничениями (множества пересекаются).

для которого находим

$$\mathbf{S}_{01} = \mathbf{I} \oplus \mathbf{B}, \quad \mathbf{S}_{11} = \mathbf{A},$$

$$\mathbf{S}_{12} = \mathbf{A} \oplus \mathbf{BA} \oplus \mathbf{AB} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}, \quad \mathbf{S}_{22} = \mathbf{A}^2 = \begin{pmatrix} 3 & 4 \\ 7 & 8 \end{pmatrix}.$$

Затем подсчитаем значения

$$\mathbf{q}^- \mathbf{S}_{01} \mathbf{p} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ -2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2, \quad (\mathbf{q}^- \mathbf{S}_{01} \mathbf{p})^{1/2} = 1,$$

$$\mathbf{q}^- \mathbf{S}_{11} \mathbf{p} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4, \quad (\mathbf{q}^- \mathbf{S}_{11} \mathbf{p})^{1/3} = 4/3,$$

$$\text{tr } \mathbf{S}_{12} = 4, \quad \text{tr}^{1/2}(\mathbf{S}_{22}) = 8/2 = 4.$$

После подстановки получаем, что минимум $\theta = 4$ не изменился. Осталось вычислить

$$\theta^{-1} \mathbf{A} \oplus \mathbf{B} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*,$$

$$\theta (\mathbf{q}^- (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^- = 4 \left(\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \right)^- = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

Применяя формулу (1.13), находим решение в виде

$$\mathbf{x} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \mathbf{u}, \quad \begin{pmatrix} -3 \\ -3 \end{pmatrix} \leq \mathbf{u} \leq \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

В терминах обычных операций решение принимает вид

$$x_1 = \max(u_1, u_2 - 1),$$

$$x_2 = \max(u_1 - 1, u_2),$$

при условии, что

$$-3 \leq u_1 \leq 3,$$

$$-3 \leq u_2 \leq 4.$$

Для рассматриваемого примера множество допустимых значений изображено на рис. 1.3 в форме полосы между прямыми, проведенными через концы векторов \mathbf{b}_1^* и \mathbf{b}_2^* (столбцов матрицы \mathbf{B}^*). В этом случае множество решений задачи без ограничений пересекается с допустимым множеством, а минимумы обеих задач равны.

Пусть в рассматриваемой задаче матрица \mathbf{B} задана следующим образом:

$$\mathbf{B} = \begin{pmatrix} 0 & -5 \\ 5 & -4 \end{pmatrix}.$$

Чтобы воспользоваться формулами из теоремы 4, вычислим

$$\mathbf{B}^2 = \begin{pmatrix} 0 & -5 \\ 5 & 0 \end{pmatrix}, \quad \text{Tr}(\mathbf{B}) = 0 = \mathbf{1},$$

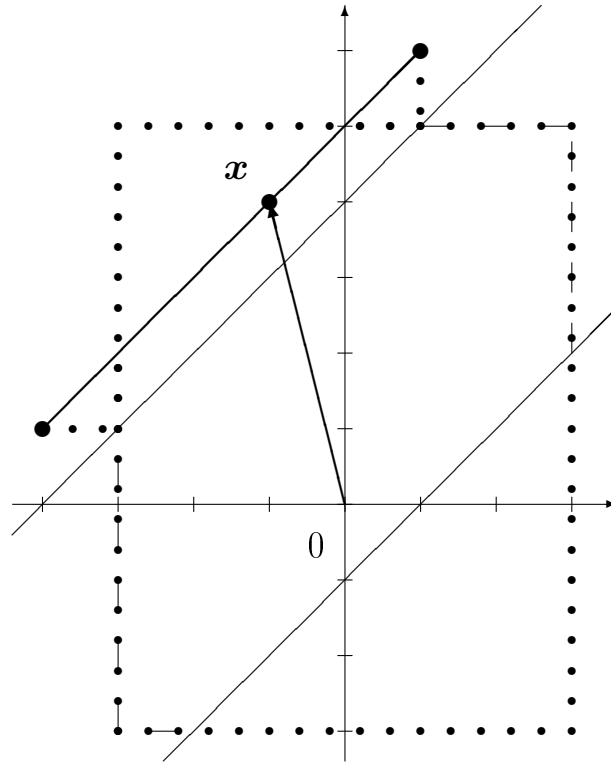


Рисунок 1.4: Пример множества решений задачи с ограничениями (множества не пересекаются).

условие $\text{Tr}(\mathbf{B}) \leq 1$ выполняется.

$$\mathbf{AB} = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & -5 \\ 5 & -4 \end{pmatrix} = \begin{pmatrix} 5 & -4 \\ 9 & 0 \end{pmatrix},$$

$$\mathbf{BA} = \begin{pmatrix} 0 & -5 \\ 5 & -4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 6 & 5 \end{pmatrix}.$$

Снова применим формулу (1.12) для нахождения минимума θ . Заметим, что в слагаемых \mathbf{S}_{11} , \mathbf{S}_{22} , а значит и в $(\mathbf{q}^- \mathbf{S}_{11} \mathbf{p})^{1/3}$ матрица \mathbf{B} не присутствует, поэтому воспользуемся их значениями из предыдущего примера. Подсчитаем недостающие слагаемые:

$$\mathbf{S}_{01} = \mathbf{I} \oplus \mathbf{B} = \mathbf{B}^*, \quad \mathbf{q}^- \mathbf{S}_{01} \mathbf{p} = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & -5 \\ 5 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 5,$$

$$(\mathbf{q}^- \mathbf{S}_{01} \mathbf{p})^{1/2} = 5/2, \quad \mathbf{S}_{12} = \begin{pmatrix} 5 & 0 \\ 9 & 5 \end{pmatrix}, \quad \text{tr } \mathbf{S}_{12} = 5.$$

Отсюда получаем $\theta = 5$, из чего следует, что минимум увеличился. Осталось вычислить

$$\theta^{-1} \mathbf{A} \oplus \mathbf{B} = \begin{pmatrix} -4 & -5 \\ -2 & -1 \end{pmatrix} \oplus \begin{pmatrix} 0 & -5 \\ 5 & -4 \end{pmatrix} = \begin{pmatrix} 0 & -5 \\ 5 & -1 \end{pmatrix},$$

$$(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* = \begin{pmatrix} 0 & -5 \\ 5 & 0 \end{pmatrix},$$

после чего находим нижнюю границу вектора \mathbf{u} как

$$\theta^{-1} \mathbf{p} = \begin{pmatrix} -4 \\ -4 \end{pmatrix}.$$

Верхняя граница вычисляется следующим образом:

$$\theta(\mathbf{q}^-(\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^- = 5 \left(\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & -5 \\ 5 & 0 \end{pmatrix} \right)^- = \begin{pmatrix} 1 \\ 6 \end{pmatrix}.$$

Решение находится по формуле (1.13) в виде

$$\mathbf{x} = \begin{pmatrix} 0 & -5 \\ 5 & 0 \end{pmatrix} \mathbf{u}, \quad \begin{pmatrix} -4 \\ -4 \end{pmatrix} \leq \mathbf{u} \leq \begin{pmatrix} 1 \\ 6 \end{pmatrix}.$$

С использованием обычных операций полученное решение записывается в форме

$$x_1 = \max(u_1, u_2 - 5),$$

$$x_2 = \max(u_1 + 5, u_2),$$

где компоненты вектора \mathbf{u} ограничены неравенствами

$$-4 \leq u_1 \leq 1,$$

$$-4 \leq u_2 \leq 6.$$

Множество решений задачи изображено на рис. 1.4 в виде отрезка прямой, проведенной под углом 45° к координатным осям. Этот пример иллюстрирует случай, когда допустимая область решений не пересекается с множеством решений задачи без ограничений, вследствие чего минимум в задаче возрастает.

Все вычисления в примерах настоящей главы были выполнены с использованием программ, реализованных на языке \mathcal{R} , исходный код которых приводится в приложении [A](#).

Глава 2

Практическое применение методов тропической оптимизации для планирование работ по дезактивации

Развитие современного общества трудно представить без новых технологий, которые позволяют преобразовывать окружающий мир, дают возможность использовать недоступные ранее территории и мощные источники энергии. Однако вместе с новыми возможностями приходит и риск возникновения чрезвычайных ситуаций антропогенной природы, а также связанных с ними проблем. Например, к серьезным проблемам [83–86] может привести авария на радиационно опасном объекте с радиоактивным загрязнением местности.

Разработка плана работ по ликвидации подобной аварии представляет собой задачу сетевого планирования. Для решения таких задач применяются детерминированные методы, такие как метод критического пути (СРМ) [87, 88] и диаграммы Ганта [89, 90], либо вероятностные, например, метод оценки и пересмотра планов (PERT) [91] и метод графической оценки и анализа (GERT) [92]. Задачи сетевого планирования можно сформулировать в виде задач оптимизации. Для их решения используются различные методы линейного и нелинейного программирования [93–97]. При этом зачастую подобное решение является алгоритмическим и не предоставляет полного решения задачи.

Одним из способов решения некоторых задач планирования производства и сетевого планирования является применение методов тропической математики [6, 40, 50, 98]. Идемпотентная (тропическая) математика представляет собой

область математики, которая занимается изучением полуколец и полуполей с идемпотентным сложением [7, 22, 46]. По сравнению с методами математического программирования, которые зачастую предоставляют лишь алгоритмическое решение, для некоторых классов задач сетевого планирования применение методов тропической оптимизации дает возможность получить в явном виде полное решение.

В настоящей главе рассматривается решение задачи планирования операции по ликвидации последствий радиационной аварии с помощью методов идемпотентной алгебры. Это решение получено путем применения теоремы, приведенной в предыдущей главе, которая была доказана в работе [63].

Глава устроена следующим образом: сначала формулируется задача ликвидатора, которая заключается в составлении плана работ по ликвидации последствий аварии. Затем строится математическая модель, которая может быть использована для решения задачи различными способами, например, методами линейного программирования. После этого задача формулируется в терминах идемпотентной математики, что позволяет применить для ее решения методы тропической оптимизации и получить решение в явном виде. Для иллюстрации приводится наглядный численный пример.

2.1 Задача ликвидатора

Рассмотрим следующую чрезвычайную ситуацию: предположим, что на некоторой территории произошла авария с радиоактивным загрязнением местности. Необходимо провести обследование местности, выработать план первоочередных работ и осуществить его.

Для этого в район заражения будут отправлены три группы: исследователи, проектировщики/руководители и исполнители работ. Перед началом работы каждой группы осуществляется ее доставка к месту аварии, а после завершения работы – ее эвакуация с места аварии. В силу некоторых обстоятельств (таких как погодные условия, наличие подходящего транспорта, опасность переоблучения населения и пр.) имеются ограничения на наиболее позднюю дату высадки, а также на наиболее раннюю возможную дату эвакуации групп после завершения задания (для разных групп ограничения могут отличаться). При

этом высадка каждой группы может осуществляться в любой момент до установленной конечной (наиболее поздней возможной) даты высадки, а эвакуация – в любой момент после установленной начальной (наиболее ранней возможной) даты эвакуации.

Для каждой группы известна продолжительность работ, которые она должна выполнить, а также зависимости между сроками выполнения этих работ и сроками выполнения других работ (для начала проектирования необходимо первичное исследование; для начала работ требуется проект и уточненные данные исследований; кроме того, у исследователей и проектировщиков есть дополнительные задачи).

Изменение мощности дозы во времени не учитывается так как время работы групп в зоне значительно меньше периода полураспада, чтобы можно было говорить о каком-либо снижении мощности дозы во времени. В общем случае максимальное время пребывания групп в зоне радиоактивного загрязнения определяется нормативными документами [99, 100].

Необходимо запланировать операцию таким образом, чтобы минимизировать максимальное время пребывания (промежуток между высадкой и эвакуацией) всех групп в опасной зоне для минимизации полученной дозы радиации с учетом всех ограничений.

Построение математической модели

Представим задачу планирования работ по ликвидации чрезвычайной ситуации в виде некоторой задачи планирования взаимосвязанных операций с учетом необходимости переброски исполнителей до места выполнения операций и их эвакуации обратно. Для этого применим математическую модель, которая, по-видимому, впервые была использована в работе [59].

Пусть имеется проект, состоящий из n операций. Для каждой операции с номером $i = 1, \dots, n$ обозначим время начала через x_i , а время окончания через y_i . Каждая операция завершается сразу, как только заданные условия для ее выполнения оказываются выполненными.

Ограничения типа «старт-финиш» устанавливаются для любых двух операций минимальный допустимый интервал времени между началом одной операции и завершением другой. Пусть a_{ij} — минимально возможная задержка между

началом операции $j = 1, \dots, n$ и окончанием операции i . В случае, если для какого-то j задержка не указана, положим $a_{ij} = -\infty$. Для любой операции i время окончания работы y_i не может быть меньше, чем $x_j + a_{ij}$ для всех $j = 1, \dots, n$. Таким образом, время окончания каждой из работ вычисляется как

$$y_i = \max(a_{i1} + x_1, \dots, a_{in} + x_n), \quad i = 1, \dots, n. \quad (2.1)$$

Ограничения «старт-старт» определяют минимальный интервал между началом выполнения двух операций. Обозначим через b_{ij} наименьший допустимый интервал времени между началом операции $j = 1, \dots, n$ и началом операции i . Тогда эта зависимость обуславливает неравенства

$$x_i \geq \max(b_{i1} + x_1, \dots, b_{in} + x_n), \quad i = 1, \dots, n. \quad (2.2)$$

Введем обозначения l_i и p_i для конечной даты высадки и начальной даты эвакуации i -й группы. Обозначим через s_i время (дату) переброски, а через t_i – эвакуации соответствующей группы. Их можно представить в следующей форме:

$$\begin{aligned} s_i &= \min(x_i, l_i) = -\max(-x_i, -l_i), \\ t_i &= \max(y_i, p_i), \quad i = 1, \dots, n. \end{aligned} \quad (2.3)$$

Тогда максимальное время пребывания в опасной зоне вычисляется как

$$\max(t_1 - s_1, \dots, t_n - s_n). \quad (2.4)$$

Запишем задачу планирования в виде задачи нахождения времени начала x_i и времени завершения y_i операций для каждой из групп, которые минимизируют максимальное время нахождения групп в районе заражения:

$$\begin{aligned} \min \quad & \max_{1 \leq i \leq n} (t_i - s_i), \\ & s_i = -\max(-x_i, -l_i), \\ & t_i = \max \left(\max_{1 \leq j \leq n} (a_{ij} + x_j), p_i \right), \\ & x_i \geq \max_{1 \leq j \leq n} (b_{ij} + x_j), \quad i = 1, \dots, n. \end{aligned} \quad (2.5)$$

Эту задачу можно решать методами линейного программирования, при этом решение будет алгоритмическим, и не гарантируется получение полного решения или решения в явном виде.

Далее будет представлено полное решение задачи ликвидатора при помощи методов тропической математики.

2.2 Решение задачи ликвидатора

Применим результат теоремы 4 для решения задачи ликвидатора, приведенной к форме (2.5).

Так как в формулировке задачи присутствуют только операции взятия максимума, сложения и вычисления противоположного (обратного по сложению), то задачу можно представить в терминах полуполя $\mathbb{R}_{\max,+}$. Для этого сначала введем следующие обозначения для матриц:

$$\mathbf{A} = (a_{ij}), \quad \mathbf{B} = (b_{ij}),$$

и для векторов

$$\mathbf{p} = (p_i), \quad \mathbf{l} = (l_i), \quad \mathbf{x} = (x_i), \quad \mathbf{s} = (s_i), \quad \mathbf{t} = (t_i).$$

При этом матрицы ограничений вида «старт-финиш» и «старт-старт» обозначаются \mathbf{A} и \mathbf{B} соответственно, \mathbf{x} – искомый вектор сроков начала работ, а векторы \mathbf{p} и \mathbf{l} определяют конечные даты высадки и начальные даты эвакуации для групп. Пользуясь матричной алгеброй над $\mathbb{R}_{\max,+}$, запишем по формулам (2.3) и (2.1) скорректированное время начала и окончания работ:

$$\begin{aligned} \mathbf{s} &= (\mathbf{x}^- \oplus \mathbf{l}^-)^-, \\ \mathbf{t} &= \mathbf{A}\mathbf{x} \oplus \mathbf{p}. \end{aligned}$$

Тогда по формуле (2.4) целевая функция приобретает форму

$$\mathbf{s}^- \mathbf{t} = (\mathbf{x}^- \oplus \mathbf{l}^-)(\mathbf{A}\mathbf{x} \oplus \mathbf{p}) = \mathbf{x}^- \mathbf{A}\mathbf{x} \oplus \mathbf{l}^- \mathbf{A}\mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{l}^- \mathbf{p}. \quad (2.6)$$

Если перевести ограничения «старт-старт» в формуле (2.2) на язык идемпотентной математики, то задача планирования сроков выполнения проекта принимает форму задачи тропической оптимизации:

$$\begin{aligned} \min \quad & \mathbf{x}^- \mathbf{A} \mathbf{x} \oplus \mathbf{l}^- \mathbf{A} \mathbf{x} \oplus \mathbf{x}^- \mathbf{p} \oplus \mathbf{l}^- \mathbf{p}, \\ & \mathbf{B} \mathbf{x} \leq \mathbf{x}. \end{aligned} \quad (2.7)$$

Нетрудно видеть из постановки задачи, что векторы \mathbf{l} , \mathbf{p} и \mathbf{x} являются конечными, что обеспечивает их регулярность в тропическом смысле. Заменив $\mathbf{l}^- \mathbf{A}$ на \mathbf{q}^- , $\mathbf{l}^- \mathbf{p}$ на r и, применив теорему 4, получаем решение задачи над полуполем $\mathbb{R}_{\max,+}$ в следующем виде.

Лемма 3. Пусть \mathbf{A} – матрица, $\lambda > 0$ – спектральный радиус \mathbf{A} , а \mathbf{B} – матрица такая, что $\text{Tr}(\mathbf{B}) \leq 0$. Для любого натурального m и $k = 1, \dots, m$ введем обозначения

$$\mathbf{S}_{0m} = \bigoplus_{i=0}^m \mathbf{B}^i, \quad \mathbf{S}_{km} = \bigoplus_{0 \leq i_0 + \dots + i_k \leq m-k} \mathbf{B}^{i_0} \mathbf{A} \mathbf{B}^{i_1} \dots \mathbf{A} \mathbf{B}^{i_k}.$$

Тогда минимум в задаче (2.6) равен

$$\theta = \mathbf{l}^- \mathbf{p} \oplus \bigoplus_{k=1}^n \text{tr}^{1/k}(\mathbf{S}_{kn}) \oplus \bigoplus_{k=0}^{n-1} (\mathbf{l}^- \mathbf{A} \mathbf{S}_{k,n-1} \mathbf{p})^{1/(k+2)},$$

а все решения имеют вид

$$\mathbf{x} = (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^* \mathbf{u}, \quad \theta^{-1} \mathbf{p} \leq \mathbf{u} \leq \theta (\mathbf{l}^- \mathbf{A} (\theta^{-1} \mathbf{A} \oplus \mathbf{B})^*)^-.$$

2.3 Численный пример

Применим следствие 3 для решения задачи ликвидатора.

Пусть заданы следующие матрицы ограничений:

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix},$$

где \mathbf{A} — матрица ограничений вида «старт-финиш», \mathbf{B} — «старт-старт». Помимо этого заданы конечные даты высадки групп — вектор \mathbf{l} и начальные даты эвакуации — вектор \mathbf{p} :

$$\mathbf{l} = \begin{pmatrix} 2 & 2 & 2 \end{pmatrix}^T, \quad \mathbf{p} = \begin{pmatrix} 7 & 5 & 10 \end{pmatrix}^T.$$

При этом столбцы (строки) идут в следующем порядке: исследователи, проектировщики и исполнители работ.

Для того, чтобы применить следствие 2.6, необходимо проверить условия существования регулярных решений. Вычислим матрицы

$$\mathbf{B}^2 = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix},$$

$$\mathbf{B}^3 = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \mathbf{B}^2.$$

После чего получаем, что

$$\text{Tr}(\mathbf{B}) = \text{tr } \mathbf{B} \oplus \text{tr } \mathbf{B}^2 \oplus \text{tr } \mathbf{B}^3 = 0 = \mathbb{1},$$

условие выполнено. Заметим, что $\mathbf{B}^* = \mathbf{B}^2$.

Также подсчитаем

$$\mathbf{A}^2 = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} 20 & 13 & 5 \\ 0 & 14 & 12 \\ 0 & 0 & 20 \end{pmatrix},$$

$$\mathbf{A}^3 = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} \begin{pmatrix} 20 & 13 & 5 \\ 0 & 14 & 12 \\ 0 & 0 & 20 \end{pmatrix} = \begin{pmatrix} 30 & 23 & 15 \\ 0 & 21 & 22 \\ 0 & 0 & 30 \end{pmatrix}, \quad \lambda = 10.$$

По формуле (2.6) находим минимум целевой функции

$$\theta = \mathbf{l}^- \mathbf{p} \oplus \text{tr}(\mathbf{S}_{13}) \oplus \text{tr}^{1/2}(\mathbf{S}_{23}) \oplus \text{tr}^{1/3}(\mathbf{S}_{33}) \oplus \\ \oplus (\mathbf{l}^- \mathbf{A} \mathbf{S}_{02} \mathbf{p})^{1/2} \oplus (\mathbf{l}^- \mathbf{A} \mathbf{S}_{12} \mathbf{p})^{1/3} \oplus (\mathbf{l}^- \mathbf{A} \mathbf{S}_{22} \mathbf{p})^{1/4}.$$

Вычислим матрицы \mathbf{M}_{km} , чтобы найти слагаемые \mathbf{S}_{km} по формулам (1.18). При этом значения матриц $\mathbf{M}_{01} = \mathbf{B}$, $\mathbf{M}_{02} = \mathbf{B}^2$, $\mathbf{M}_{03} = \mathbf{B}^3$, $\mathbf{M}_{11} = \mathbf{A}$, $\mathbf{M}_{22} = \mathbf{A}^2$ и $\mathbf{M}_{33} = \mathbf{A}^3$ были найдены ранее. Получим матрицы

$$\mathbf{M}_{12} = \mathbf{A} \mathbf{M}_{01} \oplus \mathbf{B} \mathbf{M}_{11}, \quad \mathbf{M}_{13} = \mathbf{A} \mathbf{M}_{02} \oplus \mathbf{B} \mathbf{M}_{12}, \quad \mathbf{M}_{23} = \mathbf{A} \mathbf{M}_{12} \oplus \mathbf{B} \mathbf{M}_{22}.$$

Вычислим матрицы, используемые в \mathbf{M}_{12} :

$$\mathbf{A} \mathbf{M}_{01} = \mathbf{A} \mathbf{B} = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 10 & 7 & 4 \\ 17 & 12 & 10 \end{pmatrix},$$

$$\mathbf{B} \mathbf{M}_{11} = \mathbf{B} \mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 10 & 10 \end{pmatrix},$$

откуда выводим \mathbf{M}_{12} равным

$$\mathbf{M}_{12} = \begin{pmatrix} 10 & 3 & 0 \\ 10 & 7 & 4 \\ 17 & 12 & 10 \end{pmatrix} \oplus \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 10 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix}.$$

После этого вычислим матрицы, необходимые для \mathbf{M}_{13} :

$$\mathbf{A} \mathbf{M}_{02} = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 11 & 7 & 4 \\ 17 & 12 & 10 \end{pmatrix},$$

$$\mathbf{B} \mathbf{M}_{12} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 14 & 9 & 7 \\ 17 & 12 & 10 \end{pmatrix},$$

$$\mathbf{M}_{13} = \begin{pmatrix} 10 & 3 & 0 \\ 11 & 7 & 4 \\ 17 & 12 & 10 \end{pmatrix} \oplus \begin{pmatrix} 10 & 3 & 0 \\ 14 & 9 & 7 \\ 17 & 12 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 14 & 9 & 7 \\ 17 & 12 & 10 \end{pmatrix}$$

Осталось найти матрицы для \mathbf{M}_{23} :

$$\mathbf{A}\mathbf{M}_{12} = \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix} = \begin{pmatrix} 20 & 13 & 10 \\ 20 & 14 & 14 \\ 27 & 22 & 20 \end{pmatrix},$$

$$\mathbf{B}\mathbf{M}_{22} = \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 20 & 13 & 5 \\ 0 & 14 & 12 \\ 0 & 0 & 20 \end{pmatrix} = \begin{pmatrix} 20 & 13 & 5 \\ 23 & 16 & 17 \\ 27 & 20 & 20 \end{pmatrix},$$

$$\mathbf{M}_{23} = \begin{pmatrix} 20 & 13 & 10 \\ 20 & 14 & 14 \\ 27 & 22 & 20 \end{pmatrix} \oplus \begin{pmatrix} 20 & 13 & 5 \\ 23 & 16 & 17 \\ 27 & 20 & 20 \end{pmatrix} = \begin{pmatrix} 20 & 13 & 10 \\ 23 & 16 & 17 \\ 27 & 22 & 20 \end{pmatrix}.$$

Для каждого слагаемого \mathbf{S}_{km} , необходимого для вычисления θ , рассмотрим его составные части. Некоторые из них уже известны:

$$\mathbf{S}_{02} = \mathbf{I} \oplus \mathbf{B} \oplus \mathbf{B}^2 = \mathbf{B}^*, \quad \mathbf{S}_{22} = \mathbf{M}_{22} = \mathbf{A}^2, \quad \mathbf{S}_{33} = \mathbf{M}_{33} = \mathbf{A}^3.$$

Получаем $\text{tr}^{1/3}(\mathbf{S}_{33}) = 30/3 = 10$.

Найдем оставшиеся слагаемые:

$$\mathbf{S}_{12} = \mathbf{M}_{12} \oplus \mathbf{M}_{11} = \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix} \oplus \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix},$$

$$\mathbf{S}_{23} = \mathbf{M}_{23} \oplus \mathbf{M}_{22} = \begin{pmatrix} 20 & 13 & 10 \\ 23 & 16 & 17 \\ 27 & 22 & 20 \end{pmatrix} \oplus \begin{pmatrix} 20 & 13 & 5 \\ 0 & 14 & 12 \\ 0 & 0 & 20 \end{pmatrix} = \begin{pmatrix} 20 & 13 & 10 \\ 23 & 16 & 17 \\ 27 & 22 & 20 \end{pmatrix},$$

$$\text{tr}(\mathbf{S}_{23}) = 20, \quad \text{tr}^{1/2}(\mathbf{S}_{23}) = 10.$$

$$\begin{aligned} \mathbf{S}_{13} = \mathbf{M}_{13} \oplus \mathbf{M}_{12} \oplus \mathbf{M}_{11} &= \begin{pmatrix} 10 & 3 & 0 \\ 14 & 9 & 7 \\ 17 & 12 & 10 \end{pmatrix} \oplus \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix} \oplus \\ &\oplus \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 3 & 0 \\ 14 & 9 & 4 \\ 17 & 12 & 10 \end{pmatrix}, \quad \text{tr}(\mathbf{S}_{13}) = 10. \end{aligned}$$

Найдем вектор

$$\mathbf{l}^- \mathbf{A} = \begin{pmatrix} -2 & -2 & -2 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix},$$

а затем подсчитаем

$$\mathbf{l}^- \mathbf{A} \mathbf{S}_{02} \mathbf{p} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 7 \\ 5 \\ 10 \end{pmatrix} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 7 \\ 11 \\ 14 \end{pmatrix} = 22,$$

$$\mathbf{l}^- \mathbf{A} \mathbf{S}_{12} \mathbf{p} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 10 & 3 & 0 \\ 13 & 7 & 7 \\ 17 & 12 & 10 \end{pmatrix} \begin{pmatrix} 7 \\ 5 \\ 10 \end{pmatrix} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 17 \\ 20 \\ 24 \end{pmatrix} = 32,$$

$$\mathbf{l}^- \mathbf{A} \mathbf{S}_{12} \mathbf{p} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 20 & 13 & 5 \\ 0 & 14 & 12 \\ 0 & 0 & 20 \end{pmatrix} \begin{pmatrix} 7 \\ 5 \\ 10 \end{pmatrix} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 27 \\ 22 \\ 30 \end{pmatrix} = 34,$$

$$(\mathbf{l}^- \mathbf{A} \mathbf{S}_{02} \mathbf{p})^{1/2} = 11, \quad (\mathbf{l}^- \mathbf{A} \mathbf{S}_{12} \mathbf{p})^{1/3} = 32/3, \quad (\mathbf{l}^- \mathbf{A} \mathbf{S}_{22} \mathbf{p})^{1/4} = 19/2.$$

Вычислив значение

$$l^{-}\mathbf{p} = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 7 \\ 5 \\ 10 \end{pmatrix} = 8,$$

получаем минимум целевой функции

$$\theta = \max(8, 10, 10, 10, 11, 32/3, 19/2) = 11.$$

Найдем матрицы, необходимые для записи полного решения:

$$\theta^{-1}\mathbf{A} = -11 \begin{pmatrix} 10 & 3 & 0 \\ 0 & 7 & 2 \\ 0 & 0 & 10 \end{pmatrix} = \begin{pmatrix} -1 & -8 & 0 \\ 0 & -4 & -9 \\ 0 & 0 & -1 \end{pmatrix},$$

$$\theta^{-1}\mathbf{A} \oplus \mathbf{B} = \begin{pmatrix} -1 & -8 & 0 \\ 0 & -4 & -9 \\ 0 & 0 & -1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -8 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix},$$

$$(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^2 = \begin{pmatrix} 0 & -8 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 & -8 & 0 \\ 3 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -8 & -11 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix}.$$

Заметим, что

$$(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^2 = (\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*.$$

Вычислим вектор

$$l^{-}\mathbf{A}(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^* = \begin{pmatrix} 8 & 5 & 8 \end{pmatrix} \begin{pmatrix} 0 & -8 & -11 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 15 & 10 & 8 \end{pmatrix}.$$

Левую и правую границы для вектора \mathbf{u} находим по формуле (2.6):

$$\theta^{-1}\mathbf{p} = \begin{pmatrix} -4 \\ -6 \\ -1 \end{pmatrix}, \quad \theta(\mathbf{l}^{-}\mathbf{A}(\theta^{-1}\mathbf{A} \oplus \mathbf{B})^*)^{-} = 11 \begin{pmatrix} -15 \\ -10 \\ -8 \end{pmatrix} = \begin{pmatrix} -4 \\ 1 \\ 3 \end{pmatrix}.$$

Получаем, что множество всех регулярных решений задачи (2.7) состоит из векторов вида

$$\mathbf{x} = \begin{pmatrix} 0 & -8 & -11 \\ 4 & 0 & -3 \\ 7 & 2 & 0 \end{pmatrix} \mathbf{u}, \quad \begin{pmatrix} -4 \\ -6 \\ -1 \end{pmatrix} \leq \mathbf{u} \leq \begin{pmatrix} -4 \\ 1 \\ 3 \end{pmatrix}.$$

В терминах обычных операций решение принимает вид

$$\begin{aligned} x_1 &= \max(u_1, u_2 - 8, u_3 - 11), \\ x_2 &= \max(u_1 + 4, u_2, u_3 - 3), \\ x_3 &= \max(u_1 + 7, u_2 + 2, u_3), \end{aligned}$$

при условии, что

$$\begin{aligned} u_1 &= -4, \\ -6 &\leq u_2 \leq 1, \\ -1 &\leq u_3 \leq 3. \end{aligned}$$

Так как $u_1 = -4$, с учетом максимальных значений u_2 и u_3 , имеем $x_1 = \max(u_1) = -4$, $x_2 = \max(0, u_2)$. Кроме того, так как $u_1 + 7 = 3$, а значения $u_2 + 2$ и u_3 в свою очередь также не превосходят 3, то $x_3 = 3$. Общее решение получаем в виде

$$\begin{aligned} x_1 &= -4, \\ 0 &\leq x_2 \leq 1, \\ x_3 &= 3. \end{aligned}$$

Все вычисления в примерах настоящей главы были выполнены с использованием программ, реализованных на языке R, исходный код которых приводится в приложении А.

Глава 3

Задача псевдочебышевской аппроксимации в тропическом векторном пространстве

Имеется целый ряд практических задач (см., например, [8, 44, 60–62]), которые сводятся к наилучшему приближенному решению в смысле метрики Чебышева векторного уравнения $\mathbf{Ax} = \mathbf{p}$, где \mathbf{A} и \mathbf{p} обозначают заданные матрицу и вектор, \mathbf{x} – неизвестный вектор, а произведение матрицы на вектор понимается в смысле тропической математики над полуполем $\mathbb{R}_{\max,+}$.

Рассмотрим, например, задачу оптимального планирования времени начала работ некоторого проекта с целью обеспечения заданных директивных сроков завершения работ. В этой задаче берется идемпотентное полуполе с операцией \max в роли сложения и арифметического сложения в роли умножения, координаты вектора \mathbf{x} имеют смысл времени начала выполнения для каждой работы, которое требуется определить, вектор \mathbf{p} задает директивные сроки завершения работ, матрица \mathbf{A} – ограничения снизу на величину интервалов времени между началом и завершением работ.

Проблема чебышевской аппроксимации сводится к задаче тропической математики по нахождению векторов \mathbf{x} , на которых достигается минимум в задаче

$$\min (\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{p}^- \mathbf{Ax}.$$

В дальнейшем при рассмотрении этой задачи не будем ограничиваться только полуполем $\mathbb{R}_{\max,+}$, для которого расстояние между векторами в точности

соответствует метрике Чебышева. Поэтому далее будем называть подобную задачу задачей псевдочebyшевской аппроксимации.

Для решения задачи в статьях [101, 102] предлагается подход, при котором вводится дополнительный параметр для обозначения минимума целевой функции, а затем задача сводится к решению параметризованных неравенств. С помощью такого подхода были получены решения в компактной векторной форме для рассматриваемой задачи, а также некоторых ее вариантов, включая задачи с ограничениями.

Цель настоящей главы состоит в исследовании и решении задачи тропической оптимизации с целевой функцией более общего вида. На основе применения методов решения с использованием разреженных матриц, разработанного в [103], находится полное решение задачи и его представление в компактной векторной форме [64, 66, 69].

Глава построена следующим образом. Сначала находится минимальное значение целевой функции задачи, предлагается описание множества решений в форме системы неравенств и приводится одно из решений. Далее с помощью разрежения матрицы задачи находится расширенное множество решений, а затем полное решение в виде некоторого семейства подмножеств. Предлагаются процедуры, позволяющие сократить число подмножеств, которые необходимо исследовать при построении полного решения. Показано, как полное решение задачи может быть записано в параметрическом виде в компактной векторной форме. Для иллюстрации полученных результатов приводится пример численного решения задачи на множестве трехмерных векторов.

3.1 Задачи тропической оптимизации

Задачи тропической оптимизации обычно состоят в минимизации или максимизации некоторой целевой функции, заданной на векторах над идемпотентным полуполем. Такие задачи возникают, например, при исследовании уравнения $\mathbf{Ax} = \mathbf{p}$, для которого требуется найти точное или приближенное решение [40, 44, 60–62].

Сначала предположим, что заданы матрица $\mathbf{A} \in \mathbb{X}^{n \times n}$ и вектор $\mathbf{p} \in \mathbb{X}^n$. Пусть требуется найти регулярные векторы $\mathbf{x} \in \mathbb{X}^n$, которые решают задачу

$$\min (\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{p}^- \mathbf{Ax} \quad (3.1)$$

Решение этой задачи обеспечивает наилучшее приближенное решение уравнения $\mathbf{Ax} = \mathbf{p}$ в смысле псевдочebyшевской метрики. Исследование задачи было проведено, например, в работе [102], где приводится частичное решение в следующем виде.

Лемма 4. *Для любых регулярных матрицы \mathbf{A} и вектора \mathbf{p} минимум в задаче (3.1) равен*

$$\Delta = ((\mathbf{A}(\mathbf{p}^- \mathbf{A})^-)^- \mathbf{p})^{1/2}$$

и достигается при

$$\mathbf{x} = \Delta(\mathbf{p}^- \mathbf{A})^-.$$

Обобщением задачи (3.1) с дополнительным вектором $\mathbf{q} \in \mathbb{X}^n$ является задача нахождения регулярных векторов \mathbf{x} , которые обеспечивают

$$\min (\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{Ax}. \quad (3.2)$$

Заметим, что при $\mathbf{A} = \mathbf{I}$ получается задача минимизации функции

$$\mathbf{x}^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x},$$

полное решение которой было получено в [101, 102].

В настоящей работе задача (3.2) рассматривается в следующей форме. Пусть заданы матрица $\mathbf{A} \in \mathbb{X}^{n \times n}$ и векторы $\mathbf{p}, \mathbf{q} \in \mathbb{X}^n$. Требуется найти все регулярные векторы $\mathbf{x} \in \mathbb{X}^n$, на которых достигается

$$\min (\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x}. \quad (3.3)$$

В работах [101, 102] было получено частичное решение этой задачи.

Ниже для решения задачи (3.3) сначала также, как в работах [101, 102], будет определен минимум целевой функции и получено одно из решений. Затем

для полного решения задачи строится система неравенств, которая определяет множество всех решений. На основе использования разреженных матриц, находится более широкое множество решений, а затем полное решение задачи в форме семейства подмножеств решений.

3.2 Предварительный анализ задачи

Цель этого раздела состоит в том, чтобы найти минимум целевой функции, охарактеризовать множество решений и описать некоторые свойства этого множества. Для этого будем использовать подход [59, 63, 101, 102], при котором вводится параметр для обозначения минимума целевой функции, а затем задача оптимизации сводится к решению параметризованного неравенства. Докажем следующее утверждение.

Лемма 5. Пусть \mathbf{A} – регулярная по строкам матрица, а \mathbf{p} и \mathbf{q} – регулярные векторы. Тогда минимум в задаче (3.3) равен

$$\Delta = ((\mathbf{A}\mathbf{q})^- \mathbf{p})^{1/2}, \quad (3.4)$$

а все регулярные решения \mathbf{x} определяются системой неравенств

$$\begin{aligned} \mathbf{A}\mathbf{x} &\geq \Delta^{-1}\mathbf{p}, \\ \mathbf{x} &\leq \Delta\mathbf{q}. \end{aligned} \quad (3.5)$$

В частности, минимум достигается при $\mathbf{x} = \Delta\mathbf{q}$.

Доказательство. Обозначим минимум целевой функции через θ . Рассмотрим равенство

$$\theta = (\mathbf{A}\mathbf{x})^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x}.$$

Заметим, что для всех регулярных векторов \mathbf{x} справедливо $\theta > 0$. Так как θ – минимум, то можно заменить равенство на неравенство

$$\theta \geq (\mathbf{A}\mathbf{x})^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x},$$

которое эквивалентно двум неравенствам

$$\begin{aligned}\theta &\geq (\mathbf{Ax})^- \mathbf{p}, \\ \theta &\geq \mathbf{q}^- \mathbf{x}.\end{aligned}$$

Решая с помощью леммы 1.4 первое неравенство относительно \mathbf{p} , а второе относительно \mathbf{x} , приходим к системе из двух неравенств

$$\begin{aligned}\mathbf{p} &\leq \theta \mathbf{Ax}, \\ \mathbf{x} &\leq \theta \mathbf{q}.\end{aligned}$$

После подстановки второго неравенства системы в первое, получаем

$$\mathbf{p} \leq \theta^2 \mathbf{Aq},$$

а затем

$$\theta \geq ((\mathbf{Aq})^- \mathbf{p})^{1/2} = \Delta,$$

и таким образом находим нижнюю границу для θ . Проверим, что эта граница достигается при $\mathbf{x} = \Delta \mathbf{q}$. Действительно, после подстановки в целевую функцию имеем

$$(\mathbf{Ax})^- \mathbf{p} \oplus \mathbf{q}^- \mathbf{x} = \Delta^{-1} (\mathbf{Aq})^- \mathbf{p} \oplus \Delta \mathbf{q}^- \mathbf{q} = \Delta.$$

Следовательно, нижняя граница Δ для θ оказывается точной и поэтому определяет минимум целевой функции.

Заменив в системе θ на Δ , а затем, умножив обе части первого неравенства на Δ^{-1} , получаем, что все регулярные решения задачи (3.3) должны удовлетворять системе (3.5). Учитывая эквивалентность преобразований, верно и обратное утверждение: любой регулярный вектор \mathbf{x} , удовлетворяющий системе (3.5), является решением задачи (3.3). \square

Следствие 1. *Множество решений задачи (3.3) вместе с любыми решениями содержит их всевозможные выпуклые линейные комбинации.*

Доказательство. Рассмотрим случай выпуклой линейной комбинации двух решений. Пусть \mathbf{x}_1 и \mathbf{x}_2 – решения задачи (3.3), а u_1 и u_2 – такие числа, что $u_1 \oplus u_2 = \mathbf{1}$.

Так как векторы \mathbf{x}_1 и \mathbf{x}_2 являются решениями задачи (3.3), то они определяются системой (3.5). Тогда их выпуклая комбинация удовлетворяет соотношениям

$$\begin{aligned} \mathbf{A}(u_1\mathbf{x}_1 \oplus u_2\mathbf{x}_2) &\geq u_1\Delta^{-1}\mathbf{p} \oplus u_2\Delta^{-1}\mathbf{p} = (u_1 \oplus u_2)\Delta^{-1}\mathbf{p} = \Delta^{-1}\mathbf{p}, \\ u_1\mathbf{x}_1 \oplus u_2\mathbf{x}_2 &\leq u_1\Delta\mathbf{q} \oplus u_2\Delta\mathbf{q} = (u_1 \oplus u_2)\Delta\mathbf{q} = \Delta\mathbf{q}. \end{aligned}$$

Следовательно, вектор $u_1\mathbf{x}_1 \oplus u_2\mathbf{x}_2$ также является решением задачи (3.3). Полученный результат легко обобщается на случай с произвольным количеством решений. \square

3.3 Разрежение матрицы задачи

Для расширения множества решений задачи (3.3) применим метод с использованием разрежения матриц, предложенный в [103]. Сначала преобразуем матрицу задачи $\mathbf{A} = (a_{ij})$ в разреженную матрицу $\widehat{\mathbf{A}} = (\widehat{a}_{ij})$, приравнивая к нулю все элементы, которые строго меньше порогового значения по следующему правилу:

$$\widehat{a}_{ij} = \begin{cases} a_{ij}, & \text{если } a_{ij} \geq \Delta^{-2}p_iq_j^{-1}; \\ 0, & \text{в противном случае.} \end{cases}$$

Далее матрицу $\widehat{\mathbf{A}}$, полученную при помощи такого преобразования, будем называть разреженной матрицей задачи. Свойства этой матрицы отражает следующий результат.

Лемма 6. Замена матрицы \mathbf{A} на $\widehat{\mathbf{A}}$ не меняет множество решений задачи (3.3).

Доказательство. Сначала проверим, что переход к разреженной матрице сохраняет минимальное значение Δ целевой функции, полученное в лемме 5. Для упрощения выкладок рассмотрим величину Δ^2 . Определим индексы k и s следующим образом:

$$k = \arg \max_{1 \leq i \leq n} (a_{i1}q_1 \oplus \dots \oplus a_{in}q_n)^{-1}p_i, \quad s = \arg \max_{1 \leq j \leq n} a_{kj}q_j,$$

и запишем выражение для Δ^2 в виде

$$\begin{aligned}\Delta^2 &= (\mathbf{A}\mathbf{q})^{-}\mathbf{p} = \bigoplus_{i=1}^n (a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n)^{-1}p_i = \\ &= (a_{k1}q_1 \oplus \cdots \oplus a_{kn}q_n)^{-1}p_k = (a_{ks}q_s)^{-1}p_k.\end{aligned}$$

Регулярность по строкам матрицы \mathbf{A} и регулярность вектора \mathbf{q} гарантируют, что

$$a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n > \mathbb{0}, \quad i = 1, \dots, n.$$

Кроме того, так как вектор \mathbf{p} – регулярный, то $\Delta > \mathbb{0}$, а значит и $a_{ks} > \mathbb{0}$.

Рассмотрим произвольную строку i матрицы \mathbf{A} . Из формулы для Δ^2 видно, что

$$\Delta^2 \geq (a_{i1}q_i \oplus \cdots \oplus a_{in}q_n)^{-1}p_i.$$

Это эквивалентно неравенству

$$a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n \geq \Delta^{-2}p_i,$$

которое верно только при условии, что выполняется неравенство

$$a_{ij}q_j \geq \Delta^{-2}p_i$$

для некоторого индекса j . Отсюда заключаем, что в каждой строке i матрицы \mathbf{A} существует по меньшей мере один элемент a_{ij} , который удовлетворяет условию

$$a_{ij} \geq \Delta^{-2}p_iq_j^{-1}. \quad (3.6)$$

Рассмотрим строку k матрицы \mathbf{A} , чтобы убедиться в справедливости неравенства

$$a_{kj} \leq \Delta^{-2}p_kq_j^{-1}$$

для всех j в этой строке. Если $a_{kj} = \mathbb{0}$, то это неравенство, очевидно, выполняется. При условии $a_{kj} > \mathbb{0}$ имеем

$$(a_{kj}q_j)^{-1}p_k \geq (a_{k1}q_1 \oplus \cdots \oplus a_{kn}q_n)^{-1}p_k = \Delta^2,$$

из чего следует, что неравенство также выполняется. Так как

$$\Delta^2 = (a_{ks}q_s)^{-1}p_k,$$

то в строке k есть элементы, которые превращают неравенство (3.6) в равенство, но нет ни одного, для которого (3.6) становится строгим неравенством.

Предположим теперь, что неравенство (3.6) не выполняется для некоторых i и j . Учитывая, что $p_i > \mathbb{0}$, можно записать

$$a_{ij} < \Delta^{-2}p_iq_j^{-1} \leq (a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n)q_j^{-1},$$

откуда получаем неравенство

$$a_{ij}q_j < a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n.$$

В этом случае уменьшение $a_{ij}q_j$ путем понижения значения a_{ij} вплоть до $\mathbb{0}$, не влияет на величину $a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n$ и, следовательно, на значение

$$\Delta^2 \geq (a_{i1}q_1 \oplus \cdots \oplus a_{in}q_n)^{-1}p_i.$$

Проверим, что все элементы a_{ij} , которые не удовлетворяют неравенству (3.6), можно обнулить, оставляя без изменений множество регулярных решений задачи (3.3).

Рассмотрим систему (3.5). Записав первое неравенство системы в скалярной форме, получим, что для каждого i выполняется неравенство

$$a_{i1}x_1 \oplus \cdots \oplus a_{in}x_n \geq \Delta^{-1}p_i.$$

Учитывая регулярность векторов \mathbf{x} и \mathbf{q} , умножая обе части второго неравенства системы на регулярный вектор \mathbf{q}^- слева, можно привести это неравенство к эквивалентному неравенству

$$\mathbf{q}^- \mathbf{x} \leq \Delta.$$

В скалярной форме получаем, что

$$q_1^{-1}x_1 \oplus \cdots \oplus q_n^{-1}x_n \leq \Delta.$$

Предположим, что в матрице \mathbf{A} имеется элемент, скажем a_{ij} , удовлетворяющий условию

$$a_{ij} < \Delta^{-2}p_iq_j^{-1}$$

и тем самым нарушающий неравенство (3.6). Тогда будем иметь

$$\begin{aligned} a_{ij}x_j &< \Delta^{-2}p_iq_j^{-1}x_j \leq \Delta^{-2}(q_1^{-1}x_1 \oplus \cdots \oplus q_n^{-1}x_n)p_i \leq \\ &\leq \Delta^{-1}p_i \leq a_{i1}x_1 \oplus \cdots \oplus a_{in}x_n. \end{aligned}$$

В этом случае слагаемое $a_{ij}x_j$ не вносит никакого вклада в значение суммы $a_{i1}x_1 \oplus \cdots \oplus a_{in}x_n$. Следовательно, можно положить $a_{ij} = \mathbb{0}$, не изменяя множество решений уравнения.

Осталось понять, что обнуление элементов a_{ij} , которые не удовлетворяют неравенству (3.6), равносильно замене матрицы \mathbf{A} на $\widehat{\mathbf{A}}$. \square

Заметим, что ненулевые элементы матрицы $\widehat{\mathbf{A}}$ отвечают условию

$$\widehat{a}_{ij} \geq \Delta^{-2}p_iq_j^{-1}.$$

Тогда для элементов матрицы $\widehat{\mathbf{A}}^- = (\widehat{a}_{ij}^-)$ справедливо соотношение

$$\widehat{a}_{ij}^- \leq \Delta^2q_ip_j^{-1}.$$

В матричной форме имеем неравенство

$$\widehat{\mathbf{A}}^- \leq \Delta^2\mathbf{q}\mathbf{p}^-,$$

которое будет использовано ниже.

3.4 Полное решение задачи

В этом разделе будет представлено полное решение задачи (3.3) в виде семейства решений, которое задается множеством матриц, полученных из разреженной матрицы задачи путем дальнейшего обнуления ее элементов.

Начнем с того, что расширим решение $\mathbf{x} = \Delta\mathbf{q}$, полученное в лемме 5, до некоторого множества с помощью следующего утверждения.

Лемма 7. Пусть \mathbf{A} – регулярная по строкам разреженная матрица задачи (3.3), где \mathbf{p} и \mathbf{q} – регулярные векторы, и $\Delta = ((\mathbf{A}\mathbf{q})^- \mathbf{p})^{1/2}$. Тогда минимум в задаче (3.3) равен Δ и достигается на любом векторе \mathbf{x} , который удовлетворяет условию

$$\Delta^{-1}\mathbf{A}^-\mathbf{p} \leq \mathbf{x} \leq \Delta\mathbf{q}. \quad (3.7)$$

Доказательство. Заметим, что определяемое неравенством (3.7) множество не пусто. Действительно, из того, что матрица \mathbf{A} является разреженной, следует неравенство

$$\mathbf{A}^- \leq \Delta^2\mathbf{q}\mathbf{p}^-,$$

после умножения которого на $\Delta^{-1}\mathbf{p}$ справа получаем

$$\Delta^{-1}\mathbf{A}^-\mathbf{p} \leq \Delta\mathbf{q}.$$

Покажем, что все векторы \mathbf{x} , которые удовлетворяют двойному неравенству (3.7), являются решениями системы (3.5). В силу того, что правое неравенство в (3.7) совпадает со вторым в системе (3.5), достаточно проверить выполнение неравенства

$$\mathbf{A}\mathbf{x} \geq \Delta^{-1}\mathbf{p}.$$

Так как матрица \mathbf{A} – регулярна по строкам, то $\mathbf{I} \leq \mathbf{A}\mathbf{A}^-$. Умножив левое неравенство из двойного неравенства (3.7) на \mathbf{A} слева, получаем, что

$$\Delta^{-1}\mathbf{p} \leq \Delta^{-1}\mathbf{A}\mathbf{A}^-\mathbf{p} \leq \mathbf{A}\mathbf{x},$$

откуда следует требуемое неравенство системы (3.5). □

Теперь сформулируем лемму, которая обеспечивает полное решение задачи (3.3).

Лемма 8. Пусть \mathbf{A} – регулярная по строкам разреженная матрица задачи (3.3), где \mathbf{p} и \mathbf{q} – регулярные векторы, и $\Delta = ((\mathbf{A}\mathbf{q})^- \mathbf{p})^{1/2}$. Обозначим через \mathcal{A} множество матриц, полученных из \mathbf{A} путем сохранения по одному ненулевому элементу в каждой строке и обнулением остальных.

Тогда минимум в задаче (3.3) равен Δ , а все регулярные решения \mathbf{x} образуют семейство решений, каждое из которых определяется условием

$$\Delta^{-1} \mathbf{A}_1^- \mathbf{p} \leq \mathbf{x} \leq \Delta \mathbf{q}, \quad \mathbf{A}_1 \in \mathcal{A}. \quad (3.8)$$

Доказательство. Как было показано выше, все решения задачи (3.3) задаются системой (3.5). Следовательно, для доказательства леммы требуется проверить, что каждому решению системы (3.5) соответствует некоторая матрица $\mathbf{A}_1 \in \mathcal{A}$ и наоборот.

Пусть вектор \mathbf{x} является решением системы (3.5). Рассмотрим векторное неравенство

$$\mathbf{A}\mathbf{x} \geq \Delta^{-1} \mathbf{p}$$

и исследуем каждое из соответствующих скалярных неравенств, чтобы определить максимальное слагаемое в левой части. Несложно видеть, что найдется матрица $\mathbf{A}_1 \in \mathcal{A}$ с ненулевыми элементами в каждой строке, которые соответствуют этим максимальным слагаемым.

Теперь предположим, что \mathbf{x} – решение неравенства (3.8) для некоторой матрицы \mathbf{A}_1 . Так как матрица \mathbf{A}_1 регулярна по строкам и $\mathbf{A}_1 \leq \mathbf{A}$, то

$$\mathbf{I} \leq \mathbf{A}_1 \mathbf{A}_1^- \leq \mathbf{A} \mathbf{A}_1^-.$$

Умножая левую часть неравенства (3.8) на \mathbf{A} слева, получаем

$$\mathbf{A}\mathbf{x} \geq \Delta^{-1} \mathbf{A} \mathbf{A}_1^- \mathbf{p} \geq \Delta^{-1} \mathbf{p}.$$

Из этого следует, что \mathbf{x} является решением системы (3.5), а значит и задачи (3.3). □

Отметим, что у различных множеств решений из семейства, описанного в лемме 8, могут быть пересекающиеся подмножества.

Для получения решения в параметрическом виде докажем следующую теорему.

Теорема 5. Пусть \mathbf{A} – регулярная по строкам разреженная матрица задачи (3.3), где \mathbf{p} и \mathbf{q} – регулярные векторы, и $\Delta = ((\mathbf{A}\mathbf{q})^- \mathbf{p})^{1/2}$. Обозначим через \mathcal{A} множество матриц, полученных из \mathbf{A} путем сохранения по одному ненулевому элементу в каждой строке и обнулением остальных, а через \mathbf{B} – матрицу, столбцами которой являются векторы $\mathbf{b}_1 = \Delta^{-1} \mathbf{A}_1^- \mathbf{p}$ для всех матриц $\mathbf{A}_1 \in \mathcal{A}$.

Тогда минимум в задаче (3.3) равен Δ , а все регулярные решения \mathbf{x} имеют вид

$$\mathbf{x} = \mathbf{B}\mathbf{u} \oplus \mathbf{v}, \quad \mathbf{v} \leq \Delta\mathbf{q}, \quad \mathbf{1}^T \mathbf{u} = \mathbb{1}. \quad (3.9)$$

Доказательство. Пусть множество \mathcal{A} содержит k матриц. Для каждого $i = 1, \dots, k$ возьмем матрицу $\mathbf{A}_i \in \mathcal{A}$ и определим левую часть $\mathbf{b}_i = \Delta^{-1} \mathbf{A}_i^- \mathbf{p}$ неравенства (3.8). Учитывая, что правая часть неравенства всегда равна $\Delta\mathbf{q}$, семейство решений описывается следующим набором двойных неравенств:

$$\mathbf{b}_1 \leq \mathbf{x}_1 \leq \Delta\mathbf{q}, \dots, \mathbf{b}_k \leq \mathbf{x}_k \leq \Delta\mathbf{q}.$$

Рассмотрим произвольные векторы $\mathbf{x}_1, \dots, \mathbf{x}_k$, для которых соответствующие двойные неравенства выполняются. По следствию 1 множество решений задачи (3.3) замкнуто относительно операции взятия выпуклой линейной комбинации. Следовательно, если $u_1 \oplus \dots \oplus u_k = \mathbb{1}$, то вектор $u_1 \mathbf{x}_1 \oplus \dots \oplus u_k \mathbf{x}_k$ также является решением.

Покажем, что общее решение задачи можно записать в виде

$$u_1 \mathbf{b}_1 \oplus \dots \oplus u_k \mathbf{b}_k \leq \mathbf{x} \leq \Delta\mathbf{q}, \quad u_1 \oplus \dots \oplus u_k = \mathbb{1}. \quad (3.10)$$

Если вектор \mathbf{x} – решение задачи (3.3), то он принадлежит семейству решений, определяемому нижней границей $\mathbf{b}_i = \Delta^{-1} \mathbf{A}_i^- \mathbf{p}$, которая задается некоторой матрицей $\mathbf{A}_i \in \mathcal{A}$. Чтобы представить вектор \mathbf{x} в форме (3.10), положим коэффициент u_i равным $\mathbb{1}$, а все остальные коэффициенты обнулим.

Пусть вектор \mathbf{x} удовлетворяет системе (3.10). Тогда существует такой индекс i , для которого выполняются соотношения $u_i = \mathbb{1}$ и

$$\Delta^{-1} \mathbf{A}_i^- \mathbf{p} = u_i \mathbf{b}_i \leq \mathbf{x} \leq \Delta \mathbf{q},$$

откуда следует, что \mathbf{x} принадлежит семейству решений, образованному матрицей \mathbf{A}_i .

Построим матрицу $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_k)$, используя векторы \mathbf{b}_i в качестве столбцов, и введем вектор $\mathbf{u} = (u_1, \dots, u_k)^T$. Теперь двойное неравенство и равенство системы (3.10) можно записать в векторной форме как

$$\mathbf{B}\mathbf{u} \leq \mathbf{x} \leq \Delta \mathbf{q}, \quad \mathbf{1}^T \mathbf{u} = \mathbb{1}.$$

С помощью вектора $\mathbf{v} = (v_1, \dots, v_n)^T$ запишем двойное неравенство в эквивалентной форме в виде системы

$$\begin{aligned} \mathbf{x} &= \mathbf{B}\mathbf{u} \oplus \mathbf{v}, \\ \mathbf{v} &\leq \Delta \mathbf{q}. \end{aligned}$$

Объединение этой системы с условием $\mathbf{1}^T \mathbf{u} = \mathbb{1}$ приводит к решению (3.9). □

В качестве следствия полученного результата представим решение задачи (3.1), которая является частным случаем задачи (3.3).

Следствие 2. Пусть \mathbf{A} – регулярная по строкам разреженная матрица задачи (3.1), где \mathbf{p} – регулярный вектор и $\Delta = ((\mathbf{A}(\mathbf{p}^- \mathbf{A})^-)^- \mathbf{p})^{1/2}$. Обозначим через \mathcal{A} множество матриц, полученных из \mathbf{A} путем сохранения по одному ненулевому элементу в каждой строке и обнулением остальных, а через \mathbf{B} – матрицу, столбцами которой являются векторы $\mathbf{b}_1 = \Delta^{-1} \mathbf{A}_1^- \mathbf{p}$ для всех матриц $\mathbf{A}_1 \in \mathcal{A}$.

Тогда минимум в задаче (3.3) равен Δ , а все регулярные решения \mathbf{x} имеют вид

$$\mathbf{x} = \mathbf{B}\mathbf{u} \oplus \mathbf{v}, \quad \mathbf{v} \leq \Delta(\mathbf{p}^- \mathbf{A})^-, \quad \mathbf{1}^T \mathbf{u} = \mathbb{1}.$$

Таким образом, для задачи псевдочебышевской аппроксимации было представлено множество всех решений, которые в дальнейшем могут быть отобраны с учетом дополнительных требований или ограничений.

3.5 Процедуры построения полного решения

Перебор всевозможных матриц $\mathbf{A}_1 \in \mathcal{A}$ для построения подмножеств семейства решений задачи в соответствии с результатом теоремы 5 может представлять определенные трудности. Кроме того, некоторые семейства решений могут содержаться в других семействах и поэтому при записи общего решения могут быть отброшены. Ниже описываются процедуры, позволяющие во многих случаях сократить число подмножеств, которые необходимо учесть при построении общего решения.

Рассмотрим следующую процедуру построения нижних границ в неравенстве (3.8) для семейства решений, которая заключается в последовательном выборе по одному ненулевому элементу в строках разреженной матрицы $\widehat{\mathbf{A}}$.

Предположим, что в матрице $\widehat{\mathbf{A}}$ зафиксированы элементы в некоторых строках, в результате чего получена матрица $\widetilde{\mathbf{A}} = (\widetilde{a}_{ij})$. Пусть выбран ненулевой элемент в одной из оставшихся строк, например элемент \widetilde{a}_{rj} в строке r и столбце j , значение которого фиксируется, а остальные элементы в этой строке замещаются нулями.

Всякий вектор $\mathbf{x} = (x_i)$, который является решением задачи (3.3), удовлетворяет системе (3.5), в частности, ее первому неравенству в форме

$$\widetilde{\mathbf{A}}\mathbf{x} \geq \Delta^{-1}\mathbf{p}.$$

Скалярное неравенство для строки r , где все элементы кроме \widetilde{a}_{rj} равны 0, записывается в виде

$$\widetilde{a}_{rj}x_j \geq \Delta^{-1}p_r,$$

что эквивалентно неравенству

$$x_j \geq \Delta^{-1}\widetilde{a}_{rj}^{-1}p_r.$$

В столбце j возьмем элемент \tilde{a}_{sj} , который расположен в одной из еще не рассмотренных строк s . При выполнении условия

$$\tilde{a}_{sj}p_s^{-1} \geq \tilde{a}_{rj}p_r^{-1},$$

которое эквивалентно условию

$$\tilde{a}_{sj} \geq \tilde{a}_{rj}p_r^{-1}p_s,$$

имеем следующую цепочку неравенств:

$$\tilde{a}_{sj}x_j \geq \tilde{a}_{rj}p_r^{-1}p_s\Delta^{-1}\tilde{a}_{rj}^{-1}p_r \geq \Delta^{-1}p_s.$$

Тогда неравенство

$$a_{s1}x_1 \oplus \cdots \oplus a_{sn}x_n \geq \Delta^{-1}p_s$$

в системе (3.5) выполняется вне зависимости от значений x_l для всех $l \neq j$. В этом случае дальнейшее исследование ненулевых элементов \tilde{a}_{sl} в строке s не может дать новых решений. Эти элементы можно заменить нулями, не нарушая неравенство, что уменьшает число альтернатив, которые требуется рассмотреть.

Заметим, что для проверки условия

$$\tilde{a}_{sj}p_s^{-1} \geq \tilde{a}_{rj}p_r^{-1}$$

удобно предварительно умножить каждую строку i матрицы $\hat{\mathbf{A}}$ на элемент p_i^{-1} , а затем исследовать элементы полученной матрицы, которую обозначим через $\hat{\mathbf{A}}' = (\hat{a}'_{ij})$.

Если в какой-то строке матрицы $\hat{\mathbf{A}}'$ преобладают элементы, являющиеся минимальными ненулевыми элементами в своих столбцах, то, вероятно, стоит начать перебор с подобной строки. Это позволит поочередно фиксировать такие элементы и не рассматривать строки, содержащие остальные ненулевые элементы соответствующих столбцов, а при отсутствии в таком столбце нулевых элементов сразу получать одну из границ.

Отсюда видно, что выбор строки в матрице может сыграть существенную роль для уменьшения количества рассматриваемых подмножеств семейства решений.

Некоторые нижние границы подмножеств, полученные с помощью процедуры построения полного семейства решений, могут быть несущественными в том смысле, что их удаление не повлияет на полученное множество решений. Нетрудно видеть, например, что если для каких-то двух границ \mathbf{b}_i и \mathbf{b}_j выполняется неравенство $\mathbf{b}_i \leq \mathbf{b}_j$, то граница \mathbf{b}_j может быть удалена из списка без потери решений.

Сформулируем критерий для отбрасывания несущественных границ.

Предложение 1. Пусть \mathbf{B} – матрица, столбцы которой определяют нижние границы для некоторого набора подмножеств семейства решений (3.8), а $\mathbf{b}_1 = \Delta^{-1}\mathbf{A}_1\mathbf{p}$, где $\mathbf{A}_1 \in \mathcal{A}$, – еще одна граница. Тогда граница \mathbf{b}_1 является несущественной, если выполняется неравенство

$$\mathbf{1}^T(\mathbf{b}_1^- \mathbf{B})^- \geq \mathbf{1}. \quad (3.11)$$

Доказательство. Подмножество с новой нижней границей \mathbf{b}_1 уже принадлежит семейству решений и может не учитываться, если найдется выпуклая комбинация нижних границ подмножеств семейства, которая не меньше, чем \mathbf{b}_1 . Учитывая, что имеющиеся нижние границы подмножеств образуют столбцы матрицы \mathbf{B} , это условие эквивалентно выполнению неравенства $\mathbf{B}\mathbf{u} \leq \mathbf{b}_1$ для некоторого вектора \mathbf{u} такого, что $\mathbf{1}^T\mathbf{u} = \mathbf{1}$.

Решение неравенства относительно вектора \mathbf{u} с помощью леммы 1.4 дает

$$\mathbf{u} \leq (\mathbf{b}_1^- \mathbf{B})^-.$$

Для выполнения условия $\mathbf{1}^T\mathbf{u} = \mathbf{1}$ необходимо и достаточно, чтобы хотя бы одна координата вектора $(\mathbf{b}_1^- \mathbf{B})^-$ была не меньше $\mathbf{1}$. Это условие записывается в виде неравенства

$$\mathbf{1}^T(\mathbf{b}_1^- \mathbf{B})^- \geq \mathbf{1},$$

которое является условием отбрасывания границы \mathbf{b}_1 . □

3.6 Алгоритм нахождения множества нижних границ

В разделе будет предложен алгоритм нахождения всех нижних границ для задачи псевдочebyшевской аппроксимации, необходимых для получения полного множества решений задачи (3.3), в котором используются процедуры, предложенные в разделе 3.5.

Пусть имеется разреженная матрица задачи $\mathbf{A} = (\mathbf{a}_i)$, где $\mathbf{a}_i = (a_{ij})$ – строки матрицы \mathbf{A} , а также регулярные векторы $\mathbf{p} = (p_i)$ и $\mathbf{q} = (q_i)$. По формуле (3.4) находим $\Delta = ((\mathbf{A}\mathbf{q})^{-1}\mathbf{p})^{1/2}$, после чего вычисляем матрицу $\widehat{\mathbf{A}}' = (\widehat{\mathbf{a}}_i')$, умножая каждую строку i матрицы \mathbf{A} на элемент p_i^{-1} : $\widehat{\mathbf{a}}_i' = p_i^{-1}\mathbf{a}_i$.

Множество всех решений задачи получаем с помощью следующей процедуры, передав ей в качестве аргументов матрицу \mathbf{A} и векторы \mathbf{p} , \mathbf{q} .

Algorithm 1 Алгоритм нахождения множества нижних границ.

```

1: function NEXTMATR( $\mathbf{A}$ ,  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $prev = \emptyset$ )
2:    $ll \leftarrow \emptyset$  ▷  $ll$  – список найденных границ, изначально пуст
3:    $i \leftarrow \text{BEST}(\widehat{\mathbf{A}}', prev)$  ▷ индекс наилучшей строки из оставшихся
4:   for  $j \leftarrow 1$  to  $n$  where  $a_{ij} \neq 0$  do
5:      $toFix \leftarrow \{i\}$ 
6:     ▷ находим элементы  $j$ -го столбца, которые можно зафиксировать
7:     for  $k \leftarrow 1$  to  $n$  where  $a_{kj} \neq 0$  and  $k \notin prev$  do
8:       if  $\widehat{a}'_{kj} \geq \widehat{a}'_{ij}$  then
9:          $toFix \leftarrow toFix \uplus \{k\}$  ▷ Добавляем их в список  $toFix$ 
10:      end if
11:    end for
12:     $\widetilde{\mathbf{A}} \leftarrow \text{FIX}(\mathbf{A}, j, toFix)$  ▷ фиксируем их, обнуляя другие в их строках
13:     $prev_{new} \leftarrow prev \uplus toFix$ 
14:    if SIZEOF( $prev_{new}$ ) ==  $n$  then ▷ если зафиксированы все строки
15:       $\mathbf{b}_1 \leftarrow \Delta^{-1}\widetilde{\mathbf{A}}^{-1}\mathbf{p}$  ▷ то получаем новую нижнюю границу
16:       $ll \leftarrow ll \uplus \mathbf{b}_1$  ▷ и добавляем ее в список
17:    else ▷ рекурсивно вызываем процедуру для оставшихся строк
18:       $ll \leftarrow ll \uplus \text{NEXTMATR}(\widetilde{\mathbf{A}}, \mathbf{p}, \mathbf{q}, prev_{new})$ 
19:    end if
20:  end for
21:   $ll \leftarrow \text{FILTERPOINTS}(ll)$  ▷ отбрасываем несущественные границы
22:  return  $ll$ 
23: end function

```

Аргумент *prev* служебный и используется процедурой для передачи информации об обработанных строках.

Для выбора наилучшей строки (процедура *Best* в строке 3 алгоритма 1) можно использовать различные подходы. Рассмотрим следующий подход: для каждого ненулевого элемента матрицы $\hat{\mathbf{A}}'$ определяем количество элементов столбца, которые не меньше этого элемента (и которые, соответственно, будут тоже зафиксированы при выборе этого элемента), вычисляя тем самым его «вес». Приняв «вес» нулевых элементов за $n/2$, находим общий «вес» каждой строки как сумму «весов» ее элементов. При этом больший «вес» строки соответствует большему приоритету для выбора.

Точная оценка вычислительной сложности алгоритма 1 затруднена из-за рекурсивности процедуры, а также из-за сильных различий в зависимости от структуры разреженной матрицы задачи. Поэтому была проведена статистическая оценка количества рассматриваемых матриц $\mathbf{A}_1 \in \mathcal{A}$, для чего для каждой размерности (от 3 до 10) были сгенерировано по 100000 матриц случайных матриц со стандартным нормальным распределением элементов. Для каждой такой матрицы алгоритм находил полное решение, а число рассмотренных в процессе матриц $\mathbf{A}_1 \in \mathcal{A}$ фиксировалось. После этого было вычислено среднее, дисперсия и медиана полученного распределения числа рассмотренных матриц, результаты приведены в таблице 3.1.

В наихудшем случае, когда все строки разреженной матрицы равнозначны, количество рассматриваемых матриц может быть экспоненциальным по размерности задачи, но в реальных задачах, как видно из таблицы, подобные случаи обычно не встречаются.

3.7 Численный пример

Рассмотрим задачу (3.3), заданную в полуполе $\mathbb{R}_{\max,+}$ при $n = 3$. Предположим, что имеются матрица \mathbf{A} и векторы \mathbf{p} и \mathbf{q} следующего вида:

$$\mathbf{A} = \begin{pmatrix} 5 & 5 & 2 \\ -1 & -7 & -3 \\ -2 & 1 & -6 \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

| Размерность | Среднее | Дисперсия | Медиана |
|-------------|---------|-----------|---------|
| 3 | 1.2 | 0.4 | 1 |
| 4 | 1.6 | 0.9 | 1 |
| 5 | 2.3 | 1.8 | 2 |
| 6 | 3.7 | 3.5 | 2 |
| 7 | 6.2 | 7.4 | 4 |
| 8 | 10.9 | 15.8 | 6 |
| 9 | 20.0 | 35.1 | 9 |
| 10 | 37.4 | 73.2 | 14 |

Таблица 3.1: Число рассмотренных процедурой вариантов в зависимости от размерности задачи

Найдем все регулярные векторы \mathbf{x} , которые решают задачу. Сначала по формуле (3.4) определим минимум Δ , для чего последовательно вычислим

$$\mathbf{A}\mathbf{q} = \begin{pmatrix} 5 & 5 & 2 \\ -1 & -7 & -3 \\ -2 & 1 & -6 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \\ 3 \end{pmatrix},$$

$$\Delta^2 = \begin{pmatrix} -7 & 0 & -3 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix} = (\mathbf{A}\mathbf{q})^- \mathbf{p} = 4, \quad \Delta = 2.$$

Для нахождения разреженной матрицы задачи построим пороговую матрицу следующим образом:

$$\Delta^{-2} \mathbf{p}\mathbf{q}^- = -4 \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix} \begin{pmatrix} -1 & -2 & -3 \end{pmatrix} = \begin{pmatrix} -5 & -6 & -7 \\ -1 & -2 & -3 \\ -2 & -3 & -4 \end{pmatrix}.$$

Заменяя нулем $\mathbb{0} = -\infty$ те элементы матрицы \mathbf{A} , которые строго меньше элементов матрицы $\Delta^{-2} \mathbf{p}\mathbf{q}^-$, находим матрицу $\hat{\mathbf{A}}$, а умножая каждую строку матрицы $\hat{\mathbf{A}}$ на соответствующий элемент вектора \mathbf{p}^- , получаем матрицу $\hat{\mathbf{A}}'$. В

результате получим

$$\widehat{\mathbf{A}} = \begin{pmatrix} 5 & 5 & 2 \\ -1 & 0 & -3 \\ -2 & 1 & 0 \end{pmatrix}, \quad \widehat{\mathbf{A}}' = \begin{pmatrix} 5 & 5 & 2 \\ -5 & 0 & -7 \\ -5 & -2 & 0 \end{pmatrix}.$$

Применим процедуру построения полного решения. Заметим, что в матрице $\widehat{\mathbf{A}}'$ во второй и в третьей строках все ненулевые элементы являются наименьшими ненулевыми элементами в своих столбцах. В силу этого следует начать с выбора соответствующих элементов в матрице $\widehat{\mathbf{A}}$, что позволит сократить перебор разреженных матриц. Для определенности начнем со второй строки.

Сначала зафиксируем элемент \widehat{a}_{21} . Учитывая, что в матрице $\widehat{\mathbf{A}}'$ элемент \widehat{a}'_{21} не превосходит элементов \widehat{a}'_{11} и \widehat{a}'_{31} , то можно оставить без изменений соответствующие элементы \widehat{a}_{11} и \widehat{a}_{31} матрицы $\widehat{\mathbf{A}}$, а остальные элементы этой матрицы заменить нулями.

В результате получаем матрицу \mathbf{A}_1 :

$$\mathbf{A}_1 = \begin{pmatrix} 5 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & 0 & 0 \end{pmatrix},$$

после чего находим нижнюю границу

$$\mathbf{b}_1 = \Delta^{-1} \mathbf{A}_1^- \mathbf{p} = -2 \begin{pmatrix} -5 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}.$$

Матрица нижних границ \mathbf{B}_1 приобретает вид

$$\mathbf{B}_1 = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}.$$

Теперь зафиксируем элемент \widehat{a}_{23} во второй строке матрицы $\widehat{\mathbf{A}}$. В силу того, что для элементов матрицы $\widehat{\mathbf{A}}'$ выполняется условие $\widehat{a}'_{23} \leq \widehat{a}'_{13}$, в матрице $\widehat{\mathbf{A}}$

можно оставить элемент \widehat{a}_{13} , а оставшиеся элементы первой строки этой матрицы заменить нулями.

В оставшейся третьей строке следует по очереди фиксировать каждый ее ненулевой элемент. Зафиксировав элемент \widehat{a}_{31} получаем матрицу

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -3 \\ -2 & 0 & 0 \end{pmatrix},$$

и находим вторую нижнюю границу:

$$\mathbf{b}_2 = \Delta^{-1} \mathbf{A}_2^- \mathbf{p} = -2 \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 5 \end{pmatrix}$$

После фиксации элемента \widehat{a}_{32} получаем границу

$$\mathbf{A}_3 = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -3 \\ 0 & 1 & 0 \end{pmatrix},$$

и вычисляем последнюю нижнюю границу

$$\mathbf{b}_3 = \Delta^{-1} \mathbf{A}_3^- \mathbf{p} = -2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ -2 & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix}.$$

Выясним, являются ли полученные границы существенными. Проверка условия (3.11) приводит к следующим результатам:

$$\mathbf{1}^T (\mathbf{b}_2^- \mathbf{B}_1)^- = \begin{pmatrix} 0 \end{pmatrix} \left(\begin{pmatrix} -3 & 0 & -5 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \right)^- = 0 = \mathbf{1},$$

$$\mathbf{1}^T(\mathbf{b}_3^- \mathbf{B}_1)^- = \begin{pmatrix} 0 \end{pmatrix} \left(\begin{pmatrix} 0 & 0 & -5 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \right)^- = 0 < \mathbf{1}.$$

Для границы \mathbf{b}_2 условие оказывается выполненным, а потому ее можно не учитывать. Заметим, что непосредственная проверка выполнения неравенства $\mathbf{b}_1 \leq \mathbf{b}_2$ подтверждает этот результат. Однако для границы \mathbf{b}_3 условие не выполняется и ее следует добавить к имеющейся границе \mathbf{b}_1 .

В итоге получим матрицу границ

$$\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_3) = \begin{pmatrix} 3 & 0 \\ 0 & 0 \\ 0 & 5 \end{pmatrix}.$$

Общее решение (3.9) задачи (3.3) записывается в форме

$$\mathbf{x} = \begin{pmatrix} 3 & 0 \\ 0 & 0 \\ 0 & 5 \end{pmatrix} \mathbf{u} \oplus \mathbf{v}, \quad \mathbf{v} \leq \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}, \quad \mathbf{1}^T \mathbf{u} = \mathbf{1}.$$

Представление решения в виде (3.8) дает два подмножества решений. Первое в обычных терминах записывается в виде

$$\begin{aligned} x_1 &= 3, \\ x_2 &\leq 4, \\ x_3 &\leq 5, \end{aligned}$$

а второе представимо в форме

$$\begin{aligned} x_1 &\leq 3, \\ 0 &\leq x_2 \leq 4, \\ x_3 &= 5. \end{aligned}$$

Все вычисления в примерах настоящей главы были выполнены с использованием программ, реализованных на языке \mathcal{R} , исходный код которых приводится в приложении **Б**.

Глава 4

Метод построения множества всех решений тропического линейного векторного неравенства

Различные практические задачи (например, транспортная задача в сетевой постановке, минимаксные задачи теории игр, см. [7]) приводят к необходимости решать тропические уравнения и неравенства вида

$$\mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{Ax} \geq \mathbf{b}.$$

Среди них неравенство $\mathbf{Ax} \leq \mathbf{b}$ решается в явном виде, и решение по лемме 1 имеет вид $\mathbf{x} \leq (\mathbf{b}^- \mathbf{A})^-$. Уравнение $\mathbf{Ax} = \mathbf{b}$ также может быть решено, однако его решение не всегда единственно, при этом, в случае единственности, оно может быть получено в виде $\mathbf{x} = (\mathbf{b}^- \mathbf{A})^-$ [32], а в случае отсутствия единственности может быть построена процедура, перебирающая все решения.

Если говорить о неравенстве $\mathbf{Ax} \geq \mathbf{b}$, то, по-видимому, в литературе ему не уделяется достаточного внимания. Поэтому в настоящей главе было проведено его исследование, и, так как в явном виде множество всех решений получить не удалось ввиду сложности данного множества, то предложен метод, с помощью которого можно получить все решения [68].

4.1 Метод нахождения решений неравенства

Множество решений задачи $\mathbf{Ax} \geq \mathbf{b}$ представляет собой объединение некоторых простых подмножеств. Далее будет описана конфигурация этих подмножеств и предложен метод, для нахождения решений.

4.1.1 Предварительная подготовка

Рассмотрим произвольный ненулевой вектор $\mathbf{p} \in \mathbb{X}^n$. Будем называть тропическим выпуклым конусом (далее конусом) множество всех векторов $\mathbf{c} \in \mathbb{X}^n$, таких, что $\mathbf{c} \geq \mathbf{p}$, а «гранью» – множество граничных точек конуса, лежащих на одной гиперплоскости.

Конус представляет собой многомерную фигуру, которая характеризуется количеством ненулевых компонент вектора \mathbf{p} . Так, в случае, когда в векторе $\mathbf{p} = (p_i)$ присутствует только одна ненулевая координата, например, p_j , соответствующий ему конус будет представлять собой полупространство пространства \mathbb{X}^n , отделенное от оставшейся части пространства гиперплоскостью, для всех точек которой j -я компонента равна p_j . Эта гиперплоскость является единственной «гранью» рассматриваемого конуса.

Если в векторе \mathbf{p} присутствует k ненулевых компонент, то его конус представляет собой $1/2^k$ часть пространства \mathbb{X}^n с k «гранями», каждая из которых является $1/2^{k-1}$ частью соответствующей ей гиперплоскости.

Рассмотрим неравенство

$$\mathbf{Ax} \geq \mathbf{b}. \quad (4.1)$$

Множество всех решений неравенства (4.1) представляет собой сложную фигуру, которая образуется объединением многомерных конусов, поэтому в явном виде решение получить трудно. Вместо этого предлагается метод, с помощью которого можно получить все решения алгоритмически.

Для начала заметим, что можно считать вектор $\mathbf{b} = (b_i)$, состоящим полностью из ненулевых компонент, т.к. если $b_i = 0$ для какого-либо i , то для этой компоненты неравенство выполняется автоматически, вне зависимости от вектора \mathbf{x} , соответственно можно убрать из рассмотрения i -ю строку матрицы \mathbf{A} и i -ую компоненту вектора \mathbf{b} .

После того, как избавились от нулевых компонент в векторе \mathbf{b} , можно «отнормировать» по нему матрицу \mathbf{A} . Действительно, так как для выполнения неравенства с конкретным b_i :

$$a_{i1}x_1 \oplus a_{i2}x_2 \oplus \cdots \oplus a_{im}x_m \geq b_i$$

рассматривается только i -я строка матрицы \mathbf{A} (и в других неравенствах эта строка не используется), то можно умножить i -ю строку \mathbf{A} на b_i^{-1} , сведя неравенство к виду $\tilde{\mathbf{A}}\mathbf{x} \geq \mathbf{1}$, где $\tilde{\mathbf{A}}$ – получившаяся матрица. В дальнейшем будем считать, что неравенство сведено к данному виду.

4.1.2 Общая идея

Для начала рассмотрим случай, когда у матрицы $\mathbf{A} = (\mathbf{a}_j)$, где $\mathbf{a}_j = (a_{ij})$ – столбцы матрицы \mathbf{A} , отсутствуют нулевые компоненты. Если $a_{11}x_1 \geq \mathbf{1}$, что равносильно тому, что $x_1 \geq a_{11}^{-1}$, то неравенство будет выполнено для первой компоненты вектора \mathbf{b} ($= \mathbf{1}$) вне зависимости от остальных векторов \mathbf{a}_i и прочих компонент вектора \mathbf{x} . Изобразим координатные оси x_1, x_2, \dots, x_n одну под другой (каждую будем рассматривать отдельно от остальных). Отметим и подпишем на всех осях точки по следующему правилу: на i -ой оси отмечаем точки a_{ji}^{-1} , указывая для каждой такой точки индекс j . Если в какой-то точке оказывается несколько индексов, следует записать их всех.

Если выбрана какая-либо точка из отмеченных на какой-либо оси в качестве соответствующего этой оси x_i , то неравенство автоматически выполняется для b_j , соответствующих индексу (или индексам) выбранной точки, а также всех, что находятся слева от нее. Получается, для решения задачи необходимо и достаточно выбрать точки x_i таким образом, чтобы в множество индексов, находящихся слева от выбранных, попадали все индексы $1, \dots, n$ вектора \mathbf{b} . В частности, понятно, что при отсутствии в матрице \mathbf{A} нулевых компонентов, неравенство будет выполняться в случае, если будет взята любая из крайних правых отмеченных точек, при этом оставшиеся компоненты вектора \mathbf{x} можно брать произвольно.

Данная схема показывает и то, каким образом выглядит общее решение неравенства: это объединение всевозможных наборов из точек, индексы кото-

рых покрывают числа $1, \dots, n$, вместе со всеми точками, находящимися правее этих наборов (объединение n -мерных конусов). Примечательно, что нам не важно, что происходит между отмеченными точками: так как новые индексы при рассмотрении промежуточной точки не добавляются (фактически движемся по «грани» одного из конусов, при этом до пересечения с другим ничего нового не получаем). То есть из абстрактного неравенства можно перейти к рассмотрению дискретной модели (с nt узлами).

В случае, когда в матрице \mathbf{A} есть нулевые компоненты, все остается по-прежнему. Отличие только в том, что не для каждой оси самая правая точка будет давать решение неравенства при произвольных остальных (для этого необходимо и достаточно отсутствие нулевых компонент в соответствующем векторе \mathbf{a}_i), соответственно и число узлов в дискретной модели уменьшится на количество нулевых компонент в матрице \mathbf{A} . Иногда бывает удобно выписывать нулевые компоненты отдельно для каждой оси. Заметим, что с помощью алгоритма можно наглядно увидеть, что у исходного неравенства всегда существует решение, если каждый из индексов $1, \dots, n$ вектора \mathbf{b} встречается хотя бы один раз на осях (это равносильно отсутствию нулевых строк матрицы \mathbf{A}).

Метод можно оптимизировать, если есть какая-либо априорная информация о векторах или же присутствуют дополнительные ограничения.

Пример реализации схемы приведен в приложении В.

4.2 Применение в задачах оптимизации

Часто в задаче требуется найти не просто решение неравенства $\mathbf{Ax} \geq \mathbf{b}$, но и выбрать наиболее оптимальное по каким-нибудь параметрам. Одной из самых простых, но при этом достаточно важных, является задача линейной оптимизации.

4.2.1 Линейная задача оптимизации

Пусть заданы матрица $\mathbf{A} \in \mathbb{X}^{m \times n}$ и вектор $\mathbf{b} \in \mathbb{X}^m$. Необходимо найти все векторы $\mathbf{x} \in \mathbb{X}^n$, которые решают задачу

$$\begin{aligned} \min \quad & \mathbf{p}^T \mathbf{x}, \\ & \mathbf{A}\mathbf{x} \geq \mathbf{b}. \end{aligned}$$

Будем считать, что $\mathbf{b} = \mathbf{1}$ и $\mathbf{A} = \tilde{\mathbf{A}}$ (см. (4.1.1)). Так же можно считать, что $p_i \neq 0$ для любой компоненты вектора \mathbf{p} (в противном случае, если $p_i = 0$, то соответствующая компонента x_i не будет учитываться и можно взять ее максимальной, получив автоматическое выполнение неравенства для всех ненулевых индексов вектора \mathbf{a}_i , т.е. сможем рассмотреть неравенство с меньшим количеством компонент).

Для начала, отметим на осях точки по принципу, описанному в (4.1.2). Так как компонента p_i действует только на x_i , а та, в свою очередь, только на вектор \mathbf{a}_i , то можем считать p_i «весом» (или «стоимостью») i -ой координатной оси, и соответственно появляется возможность «склеить» все оси в одну результирующую, отмечая поочередно на ней все точки с осей, умноженные на соответствующие этим осям «веса». Так как информация о том, к какому вектору принадлежит конкретная точка не сохраняется, то помимо индекса нам придется указывать и номер вектора (оси), к которому она относится.

После того, как на ось нанесены все узлы, простейшее решение получается автоматически в первой точке, на левом луче от которой будут встречаться все индексы $1, \dots, n$ (назовем ее критической). При этом все множество решений будет состоять из всех совокупностей точек, соответствующих наборам векторов, лежащих левее критической точки, при которых все индексы $1, \dots, n$ все еще встречаются.

4.2.2 Варианты работы с нелинейными задачами

По аналогии с линейной задачей оптимизации решается и нелинейная, где нужно минимизировать величину

$$f_1(x_1) \oplus f_2(x_2) \oplus \dots \oplus f_n(x_n)$$

в случае, если все f_i – монотонно возрастающие функции. При нанесении точек на результирующую ось, для вычисления «веса» каждой точки необходимо брать значение соответствующей ей функции f_i , примененной к этой точке. При этом, если $f_i \equiv \text{const}$ на всем промежутке между двумя точками i -ой оси, то на результирующей оси эти точки «склеятся».

В случае произвольной функции помимо узловых точек a_{ij} придется также рассматривать (соответственно отмечать и пересчитывать) и точки локальных экстремумов функций. Так как это достаточно специфическая задача, то ее подробное рассмотрение не проводилось.

4.3 Пример использования метода

Приведем пример поиска граничных решений неравенства (4.1).

Рассмотрим следующее неравенство:

$$\begin{pmatrix} 7 & -6 & -4 & 6 \\ -7 & -5 & 8 & 1 \\ 0 & -8 & 3 & -9 \\ -3 & 4 & 5 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geq \begin{pmatrix} -7 \\ 4 \\ 1 \\ 3 \end{pmatrix}$$

Умножим каждую строку матрицы на соответствующий ей b_i , переходя к неравенству:

$$\begin{pmatrix} 14 & 1 & 3 & 13 \\ -11 & -9 & 4 & -3 \\ -1 & -9 & 2 & -10 \\ -6 & 1 & 2 & -5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Отметим «узлы» на координатных осях (Рис. 4.1). Получаем на первой оси точки $(-14;1)$, $(11;2)$, $(1;3)$, $(6;4)$, после упорядочивания получаем $(-14;1)$, $(1;3)$, $(6;4)$, $(11;2)$.

На второй координатной оси в узле -1 получаем выполнение неравенства для двух компонент: первой и четвертой, а в узле 9 для оставшихся: второй и третьей.

На третьей оси отмечаем три узловые точки: -3 для первой компоненты, -4 для второй, а также -2 для третьей и четвертой.

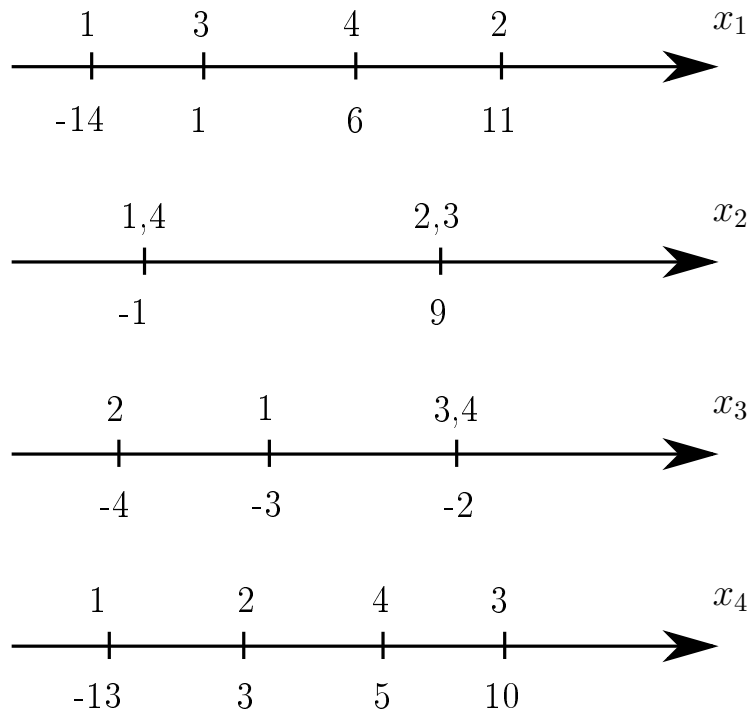


Рисунок 4.1: Узлы неравенства

На последней оси получаем 4 узла: -13 для первой компоненты, 3 для второй, 10 для третьей и 4 для четвертой.

Для всех осей соответственно после упорядочивания:

1. (-14;1), (1;3), (6;4), (11;2)
2. (-1;1,4), (9;2,3)
3. (-4;2), (-3;1), (-2;3,4)
4. (-13;1), (3;2), (4;4), (10;3)

Так как нулевых компонент ($-\infty$ для $\mathbb{R}_{\max,+}$) нет ни в одном из векторов, то автоматически получаются 4 «грани»: $(-\infty, -\infty, -\infty, 10)$, $(-\infty, -\infty, -2, -\infty)$, $(-\infty, 9, -\infty, -\infty)$ и $(11, -\infty, -\infty, -\infty)$ (под знаком « $-\infty$ » понимаем, что данную компоненту можно брать произвольно). Оставшиеся грани получаем перебором: $(1, -\infty, -\infty, 5)$, $(1, -1, -\infty, 3)$, $(1, -1, -4, -\infty)$, $(6, -\infty, -\infty, 3)$, $(6, -\infty, -4, -\infty)$. Перебор граней существенно сократило то, что выполнение

неравенства по третьей компоненте вектора \mathbf{b} на векторах $\mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ происходит только в крайних правых узлах (которые были автоматически включены в самом начале). Поэтому можно не рассматривать дополнительные наборы, в которых $x_1 < 1$ (при этом автоматически выполняется неравенство для 1 и 3 компонент, остается обеспечить выполнение для 2 и 4).

Других «граней» в неравенстве нет. Всякое решение получается прибавлением произвольного вектора к любому вектору из найденных наборов.

Заключение

Итоги выполненного исследования. Разработаны новые методы и алгоритмы численного решения многомерных минимаксных задач тропической оптимизации, а также программно-алгоритмическое обеспечение для их реализации в решении прикладных задач, возникающих при математическом моделировании естественнонаучных и научно-технических проблем.

Получены следующие результаты:

- Была полностью решена вычислительная задача с псевдоквадратичной целевой функцией и линейными ограничениями. В явном виде найдено ее решение, сформулированное в виде теоремы. Это решение представляет собой точный численный метод с конечным количеством шагов. Был проведен анализ вычислительной сложности разработанного метода и предложена схема вычислений, трудоемкость которой не превосходит $\mathcal{O}(n^5)$, где n означает размерность задачи.
- В работе была сформулирована и решена задача ликвидатора, которая заключается в составлении плана работ по ликвидации последствий радиационной аварии. Она была представлена в форме задачи сетевого планирования при наличии ограничений вида «старт-старт» и «старт-финиш» на порядок работ, а также дополнительных ограничений на сроки их выполнения. Далее задача сетевого планирования была переформулирована в терминах идемпотентной математики как вычислительная задача тропической оптимизации, после чего было получено решение задачи в явном виде в компактной векторной форме. Такое решение обеспечивает естественную алгоритмическую реализацию, а также позволяет модифицировать задачу, в том числе путем добавления дополнительных ограничений.

- Полученный результат использован для полного решения задачи ликвидатора. Стоит отметить, что, хотя в работе рассматривалась задача составления плана мероприятий по ликвидации чрезвычайной ситуации только с радиоактивным загрязнением территории, аналогичный подход может быть применен для планирования операции в случае химического и/или бактериологического (биологического) загрязнения.
- Была полностью решена расширенная задача псевдочебышевской аппроксимации в тропическом векторном пространстве. В отличие от известных ранее результатов, которые позволяют получить одно из решений задачи, в диссертационной работе получено множество всех решений задачи в виде семейства решений, которое задается множеством матриц, полученных из матрицы задачи путем оставления по одному элементу в каждой строке и обнулением остальных. Предложены процедуры для упрощения вычислений, а также разработан алгоритм, реализующий эти процедуры.
- Получены результаты исследования тропического векторного неравенства и предложен вычислительный метод, позволяющий найти все множество решений этого неравенства.

Рекомендации.

- Разработанные методы рекомендуются к применению в области создания алгоритмов численного решения задач тропической алгебры, связанных с задачами сетевого планирования, а также для решения прикладных задач, возникающих при математическом моделировании естественнонаучных и научно-технических проблем, таких, как, например, проблема ликвидации аварии антропогенной природы.
- На основании предложенных диссертантом методов и их программной реализации может быть разработано программное обеспечение для решения задач оптимизации. Помимо этого, представленные реализации могут использоваться для последующих исследований рассмотренных задач, в частности, с целью дальнейшего сокращения трудоемкости вычислений.

Перспективы дальнейшей разработки темы. Возможные направления исследовательской работы по теме диссертации:

- Добавление дополнительных ограничений к задаче с псевдоквадратичной целевой функцией и исследование возможностей ее применения для решения других вычислительных задач тропической оптимизации.
- Использование задачи с псевдоквадратичной целевой функцией совместно с расширенной задачей псевдочebyшевской аппроксимации, например, для нахождения численного решения в задаче оценки альтернатив на основе парных сравнений.
- Улучшение численного алгоритма в расширенной задаче псевдочebyшевской аппроксимации, необходимого для получения расчетной формулы, с целью дальнейшего сокращения вычислительной сложности.

Список литературы

1. *Golan, J. S.* Semirings and their applications. / J. S. Golan. — Dordrecht: Kluwer Academic Publishers, 1999. — 381 p.
2. *Głazek, K.* A guide to the literature on semirings and their applications in mathematics and information sciences. With complete bibliography. / K. Głazek. — Dordrecht: Kluwer Academic Publishers, 2002. — 392 p.
3. *Vandiver, H. S.* Note on a simple type of algebra in which the cancellation law of addition does not hold. / H. S. Vandiver // *Bulletin of the American Mathematical Society*. — 1934. — Vol. 40. — P. 914–920.
4. *Dedekind, R.* Über die Theorie der ganzen algebraischen Zahlen / R. Dedekind // *Vorlesungen über Zahlentheorie* / Ed. by L. Dirichlet. — Druck und Verlag Braunschweig, 1894. — P. 434–657.
5. *Kleene, S. C.* Representation of events in nerve nets and finite automata / S. C. Kleene // *Automata Studies*. — 1956. — P. 3–42.
6. *Cuninghame-Green, R. A.* Describing industrial processes with interference and approximating their steady-state behaviour / R. A. Cuninghame-Green // *Operations Research Quarterly*. — 1962. — Vol. 13, no. 1. — P. 95–100.
7. *Воробьев, Н. Н.* Экстремальная алгебра матриц / Н. Н. Воробьев // *Доклады АН СССР*. — 1963. — Т. 152, № 1. — С. 24–27.
8. *Воробьев, Н. Н.* Экстремальная алгебра положительных матриц / Н. Н. Воробьев // *Elektronische Informationsverarbeitung und Kybernetik*. — 1967. — Т. 3, № 1. — С. 39–71.

9. *Воробьев, Н. Н.* Экстремальная алгебра неотрицательных матриц / Н. Н. Воробьев // *Elektronische Informationsverarbeitung und Kybernetik*. — 1970. — Т. 4/5. — С. 302–312.
10. *Birkhoff, G.* Lattice theory. Rev. ed. / G. Birkhoff. — New York: American Mathematical Soc., 1948. — 285 p.
11. *Канторович, Л. В.* Функциональный анализ в полуупорядоченных пространствах / Л. В. Канторович, Б. З. Вулих, А. Г. Пинскер. — Гос. изд-во технико-теорет. лит-ры, 1950. — 550 с.
12. *Лунц, А. Г.* Алгебраические методы анализа и синтеза контактных схем / А. Г. Лунц // *Известия Российской академии наук. Серия математическая*. — 1952. — Т. 16. — С. 405–426.
13. *Поваров, Н. Г.* Матричные методы анализа релейно-контактных схем по условиям несрабатывания / Н. Г. Поваров // *Автоматика и телемеханика*. — 1954. — Т. 15. — С. 332–335.
14. *Shimbel, A.* Structure in communication nets / A. Shimbel // *Proceedings of the symposium on information networks / Polytechnic Institute of Brooklyn Nueva York*. — Vol. 4. — 1954.
15. *Bellman, R.* On a new functional transform in analysis: The maximum transform / R. Bellman, W. Karush // *Bulletin of the American Mathematical Society*. — 1961. — sep. — Vol. 67, no. 5. — P. 501–504.
16. *Канторович, Л. В.* О перемещении масс. / Л. В. Канторович // *Доклады Академии наук СССР*. — 1942. — Т. 37, № 7–8. — С. 227–229.
17. *Романовский И. В.* Несколько замечаний о функциональном преобозовании Беллмана-Каруша / Романовский И. В. // *Вестник Санкт-Петербургского университета. Серия 1. Математика, механика, астрономия*. — 1962. — Т. 17, № 18. — С. 148–150.
18. *Романовский И. В.* Асимптотическое поведение процессов динамического программирования с непрерывным множеством состояний. / Романовский

- И. В. // *Доклады Академии наук СССР*. — 1964. — Т. 159, № 6. — С. 1224–1227.
19. Корбут, А. А. Экстремальные пространства. / А. А. Корбут // *Доклады Академии наук СССР*. — 1965. — Т. 164, № 6. — С. 1229–1231.
20. Корбут, А. А. Экстремальные векторные пространства и их свойства / А. А. Корбут // *Elektronische Informationsverarbeitung und Kybernetik*. — 1972. — Т. 8/9. — С. 525–536.
21. Дудников, П. И. Эндоморфизмы полумодулей над полукольцами с идемпотентной операцией. / П. И. Дудников, С. Н. Самборский // *Известия Российской академии наук. Серия математическая*. — 1991. — Т. 55, № 1. — С. 93–109.
22. Маслов В. П. Идемпотентный анализ и его применение в оптимальном управлении / Маслов В. П., Колокольцов В. Н. — М.: Физматлит, 1994. — 144 с.
23. Maslov, V. P. On a new superposition principle for optimization problem / V. P. Maslov // *Séminaire Équations aux dérivées partielles (Polytechnique)*. — 1985. — P. 1–14.
24. Маслов, В. П. О новом принципе суперпозиции для задач оптимизации. / В. П. Маслов // *Успехи математических наук*. — 1987. — Т. 42, № 3(255). — С. 39–48.
25. Dobrokhotov, S. Yu. Quantization of the Bellman equation, exponential asymptotics and tunneling. / S. Yu. Dobrokhotov, V. N. Kolokoltsov, V. P. Maslov // *Idempotent analysis*. ed. by A.B. Sossinskij. — Providence, RI: American Mathematical Society, 1992. — P. 1–46.
26. Maslov, V. P. Stationary Hamilton-Jacobi and Bellman equations (existence and uniqueness of solutions). / V. P. Maslov, S. N. Samborskij // *Idempotent analysis*. ed. by A.B. Sossinskij. — Providence, RI: American Mathematical Society, 1992. — P. 119–133.

27. *Литвинов, Г. Л.* Идемпотентный функциональный анализ. Алгебраический подход / Г. Л. Литвинов, В. П. Маслов, Г. Б. Шпиз // *Математические заметки*. — 2001. — Т. 69, № 5. — С. 758–797.
28. *Litvinov, G. L.* The correspondence principle for idempotent calculus and some computer applications. / G. L. Litvinov, V. P. Maslov // *Idempotency. Based on a workshop, Bristol, UK, October 3–7, 1994*. — Cambridge: Cambridge University Press, 1998. — P. 420–443.
29. *Литвинов, Г. Л.* Деквантование Маслова, идемпотентная и тропическая математика: краткое введение / Г. Л. Литвинов // *Записки научных семинаров ПОМИ*. — 2005. — Т. 326. — С. 145–182.
30. *Krivulin, N.* Tropical optimization problems / N. Krivulin // *Advances in Economics and Optimization: Collected Scientific Studies Dedicated to the Memory of L. V. Kantorovich* / edited by L. A. Petrosyan, J. V. Romanovsky, D. W. K. Yeung. — New York: Nova Science Publishers, 2014. — Economic Issues, Problems and Perspectives. — P. 195–214.
31. *Krivulin, N.* A multidimensional tropical optimization problem with a nonlinear objective function and linear constraints / N. Krivulin // *Optimization*. — 2015. — Vol. 64, no. 5. — P. 1107–1129.
32. *Кривулин Н. К.* Методы идемпотентной алгебры в задачах моделирования и анализа сложных систем / Кривулин Н. К. — СПб.: Изд-во С.-Петербур. ун-та, 2009. — 256 с.
33. *Sergeev, S.* Minimal elements and cellular closures over the max-plus semiring. / S. Sergeev // *Idempotent and tropical mathematics and problems of mathematical physics* / Independent University of Moscow. — Vol. 2. — 2007. — P. 49–52.
34. *Gaubert, S.* Tropical linear-fractional programming and parametric mean payoff games / S. Gaubert, R. D. Katz, S. Sergeev // *Journal of Symbolic Computation*. — 2012. — Vol. 47, no. 12. — P. 1447–1478.

35. *Sergeev, S.* Max-algebraic attraction cones of nonnegative irreducible matrices. / S. Sergeev // *Linear Algebra and its Applications.* — 2011. — Vol. 435, no. 7. — P. 1736–1757.
36. *Mikhalkin, G.* Amoebas of algebraic varieties and tropical geometry. / G. Mikhalkin // *Different faces of geometry.* — New York, NY: Kluwer Academic/Plenum Publishers, 2004. — P. 257–300.
37. *Itenberg, I.* Tropical algebraic geometry. 2nd ed. / I. Itenberg, G. Mikhalkin, E. Shustin. — 2nd ed. edition. — Basel: Birkhäuser, 2009. — 104 p.
38. Brief introduction to tropical geometry. / E. Brugallé, I. Itenberg, G. Mikhalkin, K. Shaw // *Proceedings of the 21st Gökova geometry-topology conference, Gökova, Turkey, May 26-30, 2014.* — Somerville, MA: International Press; Gökova: Gökova Geometry-Topology Conferences (GGT), 2015. — P. 1–75.
39. *Казарян, М. Э.* Тропическая геометрия / М. Э. Казарян. — Московский центр непрерывного математического образования (МЦНМО), 2012. — 43 с.
40. *Cuninghame-Green, R. A.* Minimax algebra and applications / R. A. Cuninghame-Green // *Advances in Imaging and Electron Physics* / edited by P. W. Hawkes. — San Diego, CA: Academic Press, 1994. — Vol. 90 of *Advances in Imaging and Electron Physics.* — P. 1–121.
41. *Zimmermann, K.* Disjunctive optimization, max-separable problems and extremal algebras / K. Zimmermann // *Theoretical Computer Science.* — 2003. — Vol. 293, no. 1. — P. 45–54.
42. *Tharwat, A.* One class of separable optimization problems: solution method, application / A. Tharwat, K. Zimmermann // *Optimization.* — 2010. — Vol. 59, no. 5. — P. 619–625.
43. *Кривулин, Н. К.* Использование тропической оптимизации для решения минимаксных задач размещения с прямоугольной метрикой на прямой /

- Н. К. Кривулин, П. В. Плотников // *Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия.* — 2016. — Т. 3 (61). Вып. 4.
44. *Cunningham-Green, R. A.* Projections in minimax algebra / R. A. Cunningham-Green // *Mathematical Programming.* — 1976. — Vol. 10. — P. 111–123.
45. *Zimmermann, U.* Linear and combinatorial optimization in ordered algebraic structures / U. Zimmermann. — Amsterdam: Elsevier, 1981. — Vol. 10 of *Annals of Discrete Mathematics.* — 390 p.
46. *Butkovič, P.* Introduction to max-linear programming / P. Butkovič, A. Aminu // *IMA Journal of Management Mathematics.* — 2009. — Vol. 20, no. 3. — P. 233–249.
47. *Николаев, Д. А.* Моделирование и управление движением агента в неопределенной внешней среде методами идемпотентной алгебры / Д. А. Николаев // *Управление большими системами: сборник трудов.* — 2012. — № 40. — С. 311–318.
48. *Матвеевко, В. Д.* Оптимальные траектории схемы динамического программирования и экстремальные степени неотрицательных частиц / В. Д. Матвеевко // *Дискретная математика.* — 1990. — Т. 2, № 1. — С. 59–71.
49. *Кривулин, Н. К.* Решение задач математического программирования с использованием методов тропической оптимизации / Н. К. Кривулин, И. В. Романовский // *Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия.* — 2017. — Т. 4 (62). Вып. 3.
50. Synchronization and linearity: an algebra for discrete event systems / F. L. Baccelli, G. Cohen, G. J. Olsder, J.-P. Quadrat. Wiley Series in Probability and Statistics. — Chichester: Wiley, 1992. — 514 p.
51. *Grigoriev, D.* Tropical cryptography / D. Grigoriev, V. Shpilrain // *Communications in Algebra.* — 2014. — Vol. 42, no. 6. — P. 2624–2632.

52. *Krivulin, N.* A constrained tropical optimization problem: Complete solution and application example / N. Krivulin // Tropical and Idempotent Mathematics and Applications / edited by G. L. Litvinov, S. N. Sergeev. — Providence, RI: American Mathematical Society, 2014. — Vol. 616 of *Contemporary Mathematics*. — P. 163–177.
53. *Krivulin, N.* Complete solution of a constrained tropical optimization problem with application to location analysis / N. Krivulin // Relational and Algebraic Methods in Computer Science / edited by P. Höfner, P. Jipsen, W. Kahl, M. E. Müller. — Cham: Springer, 2014. — Vol. 8428 of *Lecture Notes in Computer Science*. — P. 362–378.
54. *Кривулин, Н. К.* Экстремальное свойство собственного значения неразложимых матриц в идемпотентной алгебре и решение задачи размещения Вебера—Ролса / Н. К. Кривулин // *Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия*. — 2011. — Т. 44, № 4. — С. 272–281.
55. *Gaubert, S.* Tropical linear-fractional programming and parametric mean payoff games / S. Gaubert, R. D. Katz, S. Sergeev // *Journal of Symbolic Computation*. — 2012. — Vol. 47, no. 12. — P. 1447 – 1478. — International Workshop on Invariant Generation.
56. *Butkovič, P.* Max-linear systems: theory and algorithms / P Butkovič. Springer Monographs in Mathematics. — London: Springer, 2010. — 272 p.
57. *Grigoriev, D.* On a tropical dual Nullstellensatz / D. Grigoriev // *Advances in Applied Mathematics*. — 2012. — Vol. 48, no. 2. — P. 457–464.
58. *Grigoriev, D.* Complexity of tropical Schur polynomials / D. Grigoriev, G. Koshevoy // *Journal of Symbolic Computation*. — 2016. — Vol. 74, no. 1. — P. 46–54.
59. *Krivulin, N.* Extremal properties of tropical eigenvalues and solutions to tropical optimization problems / N. Krivulin // *Linear Algebra and its Applications*. — 2015. — Vol. 468. — P. 211 – 232.

60. *Zimmermann, K.* Some optimization problems with extremal operations / K. Zimmermann // *Mathematical Programming at Oberwolfach II* / edited by B. Korte, K. Ritter. — Berlin, Heidelberg: Springer Berlin Heidelberg, 1984. — P. 237–251.
61. *Cechlárová, K.* Soluble approximation of linear systems in max-plus algebra / K. Cechlárová, R. A. Cuninghame-Green // *Kybernetika*. — 2003. — Vol. 39, no. 2. — P. 137–141.
62. *Butkovic, P.* On some properties of the image set of a max-linear mapping / P. Butkovic, K. P. Tam // *Tropical and Idempotent Mathematics*. — 2009. — Vol. 495. — P. 115–126.
63. *Кривулин, Н. К.* Решение задачи тропической оптимизации с линейными ограничениями / Н. К. Кривулин, **В. Н. Сорокин** // *Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия*. — 2015. — Т. 2 (60). Вып. 4. — С. 541–552.
64. *Кривулин, Н. К.* О решении одной многомерной задачи тропической оптимизации с использованием разрежения матриц / Н. К. Кривулин, **В. Н. Сорокин** // *Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия*. — 2018. — Т. 5 (63). Вып. 1. — С. 86–99.
65. *Krivulin, N. K.* Solution of a tropical optimization problem with linear constraints / N. K. Krivulin, **V. N. Sorokin** // *Vestnik St. Petersburg University: Mathematics*. — 2015. — Vol. 48, no. 4. — P. 224–232.
66. *Krivulin, N. K.* Solution of a multidimensional tropical optimization problem using matrix sparsification / N. K. Krivulin, **V. N. Sorokin** // *Vestnik St. Petersburg University: Mathematics*. — 2018. — Vol. 51, no. 1. — P. 66–76.
67. *Кривулин, Н. К.* Решение задач тропической оптимизации при наличии ограничений с приложением к управлению сроками проектов / Н. К. Кривулин, **В. Н. Сорокин** // *Модели и методы тропической математики в прикладных задачах экономики и управления. Сб. науч. статей. Вып. 2*

- / Под ред. Н. К. Кривулина. — Санкт–Петербург: Издательство «ВВМ», 2014. — С. 24–45.
68. **Сорокин, В. Н.** Метод построения всех решений линейного векторного неравенства в идемпотентной алгебре / **В. Н. Сорокин** // Модели и методы тропической математики в прикладных задачах экономики и управления. Сб. науч. статей. / Под ред. Н. К. Кривулина. — Санкт–Петербург: Издательство «ВВМ», 2013. — С. 108–120.
69. *Кривулин, Н. К.* Использование разрежения матриц для решения многомерной задачи тропической оптимизации / Н. К. Кривулин, **В. Н. Сорокин** // International Scientific Conference. Mathematical Modeling. Proceedings. Vol. 1. — Sofia: Scientific Technical Union of Mechanical Engineering “INDUSTRY-4.0”, 2017. — С. 36–39.
70. *Блюмин, С. Л.* Математические проблемы искусственного интеллекта: регулярность по Дж. фон Нейману в линейной и «линейной» алгебрах / С. Л. Блюмин // *Системы управления и информационные технологии.* — 2003. — № 1-2(12). — С. 90–94.
71. *Блюмин, С. Л.* Идемпотентный индексный экономический факторный анализ / С. Л. Блюмин // *Управление большими системами: сборник трудов.* — 2006. — № 14. — С. 34–39.
72. *Матвеевко, В. Д.* Оптимальные пути в ориентированных графах и собственные векторы в $\max\oplus$ системах / В. Д. Матвеевко // *Дискретная математика.* — 2009. — Т. 21, № 3. — С. 79–98.
73. *Николаев, Д. А.* Аналитическое описание дискретной динамики роботаманипулятора в неопределенной внешней среде методами идемпотентной математики / Д. А. Николаев // *Автоматика и телемеханика.* — 2012. — № 11. — С. 114–128.
74. *Gaubert, S.* Methods and applications of $(\max,+)$ linear algebra / S. Gaubert, M. Plus // *Annual Symposium on Theoretical Aspects of Computer Science. — Berlin.* — 1997. — P. 261–282.

75. *Guterman, A.* Tropical patterns of matrices and the Gondran–Minoux rank function / A. Guterman, Y. Shitov // *Linear Algebra and its Applications*. — 2012. — Vol. 437, no. 7. — P. 1793–1811.
76. *Guterman, A.* Rank functions of tropical matrices / A. Guterman, Y. Shitov // *Linear Algebra and its Applications*. — 2016. — Vol. 498. — P. 326–348.
77. *Shitov, Y.* Tropical matrices and group representations / Y. Shitov // *Journal of Algebra*. — 2012. — Vol. 370. — P. 1–4.
78. *Krivulin, N. K.* Solution of generalized linear vector equations in idempotent algebra / N. K. Krivulin // *Vestnik St. Petersburg University: Mathematics*. — 2006. — Vol. 39, no. 1. — P. 23–36.
79. *Krivulin, N.* A maximization problem in tropical mathematics: A complete solution and application examples / N. Krivulin // *Informatica*. — 2016. — Vol. 27, no. 3. — P. 587–606.
80. *Krivulin, N.* Direct solution to constrained tropical optimization problems with application to project scheduling / N. Krivulin // *Computational Management Science*. — 2017. — Vol. 14, no. 1. — P. 91–113.
81. *Кривулин, Н. К.* Об алгебраическом решении задачи Ролса о размещении на плоскости с прямоугольной метрикой / Н. К. Кривулин, П. В. Плотников // *Вестник Санкт-Петербургского университета. Серия 1. Математика. Механика. Астрономия*. — 2015. — Т. 2, № 2. — С. 194–202.
82. *Кривулин, Н. К.* Применение методов тропической оптимизации для оценки альтернатив на основе парных сравнений / Н. К. Кривулин, В. А. Агеев, И. В. Гладких // *Вестник Санкт-Петербургского университета. Серия 10. Прикладная математика. Информатика. Процессы управления*. — 2017. — № 1. — С. 27–41.
83. *Балонов, М. И.* Последствия Чернобыля: 20 лет спустя / М. И. Балонов // *Радиация и риск (Бюллетень Национального радиационно-эпидемиологического регистра)*. — 2006. — Т. 15, № 3-4. — С. 97–119.

84. Байда, С. Е. Мега-катастрофы как стратегическое и тактическое оружие войн нового поколения, возможность их прогнозирования и предупреждения / С. Е. Байда // *Технологии гражданской безопасности*. — 2010. — Т. 7, № 1-2. — С. 191–198.
85. Методологические подходы оценки влияния окружающей среды на состояние здоровья животных / И. М. Донник, И. А. Шкуратова, Н. А. Верещак и др. // *Аграрная наука Евро-северо-востока*. — 2006. — № 8. — С. 169–173.
86. Уроки Чернобыля и Фукусима: прогноз радиологических последствий / В. К. Иванов, В. В. Кащеев, С. Ю. Чекин и др. // *Радиация и риск (Бюллетень Национального радиационно-эпидемиологического регистра)*. — 2011. — Т. 20, № 3. — С. 6–15.
87. Nafkha, R. The critical path method in estimating project duration / R. Nafkha, A. Wiliński // *Information Systems in Management*. — 2016. — Vol. 5, no. 1. — P. 78–87.
88. Ökmen, Ö. A procedure for critical path method-based scheduling in linear construction projects / Ö Ökmen // *Journal of the South African Institution of Civil Engineering*. — 2013. — Vol. 55, no. 2. — P. 12–20.
89. Руководство к своду знаний по управлению проектами (Руководство РМВОК). — Project Management Institute, Inc, 2013. — 587 с. — Пятое издание.
90. Dynamic scheduling system of human-computer interaction for steelmaking-continuous casting process based on gantt chart and optimization model / Y. L. Zhou, J. C. Zhang, X. Wang, S. P. Yu // *Advanced Manufacturing and Information Engineering, Intelligent Instrumentation and Industry Development*. — Vol. 602 of *Applied Mechanics and Materials*. — Trans Tech Publications, 2014. — 10. — P. 727–730.
91. Agyei, W. Project planning and scheduling using PERT and CPM techniques with linear programming: case study / W. Agyei // *International Journal of Scientific & Technology Research*. — 2015. — Vol. 4, no. 8. — P. 222–227.

92. *Ramani, T.* Scheduling of industrialized construction project using graphical evaluation and review technique (GERT) / T. Ramani, R. Kannan // Second International Conference on Advances in Industrial Engineering Applications. — 2014. — P. 35–39.
93. *Романовский, И. В.* Алгоритмы решения экстремальных задач / И. В. Романовский. — Москва: Наука, 1977. — 352 с.
94. *Демьянов, В. Ф.* К теории нелинейных минимаксных задач / В. Ф. Демьянов, В. Н. Малоземов // *Успехи математических наук.* — 1971. — Т. 26, № 3 (159). — С. 53–104.
95. *Демьянов, В. Ф.* Введение в минимакс / В. Ф. Демьянов, В. Н. Малоземов. — Москва: Наука, 1972. — 368 с.
96. *Даугавет, В. А.* Квадратичная скорость сходимости одного метода линеаризации для решения дискретных минимаксных задач / В. А. Даугавет, В. Н. Малоземов // *Журнал вычислительной математики и математической физики.* — 1981. — Т. 21, № 4. — С. 835–843.
97. Моделирование систем и процессов: учебник. Серия 58. Бакалавр. Академический курс (1-е изд.) / В. Н. Волкова, В. Н. Козлов, А. Н. Фирсов и др. — Москва: Издательство Юрайт, 2015. — Т. 58 Бакалавр. Академический курс (1-е изд.). — 450 с.
98. *Маркова, Е. М.* Решение задачи сетевого планирования методами тропической алгебры / Е. М. Маркова, Д. А. Николаев // Сборник тезисов докладов научной конференции студентов и аспирантов Липецкого государственного технического университета. — 2016. — С. 384–387.
99. Нормы радиационной безопасности (НРБ-99/2009) / СанПин. — № 2.6.1.2523-09, Москва, 2009. Минздрав России.
100. Основные санитарные правила обеспечения радиационной безопасности (ОСПОРБ-99/2010) / СанПин. — № 2.6.1.2612-10, Москва, 2010. Минздрав России.

101. *Krivulin, N.* A new algebraic solution to multidimensional minimax location problems with Chebyshev distance / N. Krivulin // *WSEAS Transactions on Mathematics*. — 2012. — Vol. 11, no. 7. — P. 605–614.
102. *Krivulin, N.* Direct solutions to tropical optimization problems with nonlinear objective functions and boundary constraints / N. Krivulin, K. Zimmermann // *Mathematical Methods and Optimization Techniques in Engineering: Proc. 1st Intern. Conf. on Optimization Techniques in Engineering (OTENG '13)*, Antalya, Turkey. — WSEAS Press, 2013. — P. 86–91.
103. *Krivulin, N.* Algebraic solution of tropical optimization problems via matrix sparsification with application to scheduling / N. Krivulin // *Journal of Logical and Algebraic Methods in Programming*. — 2017. — Vol. 89. — P. 150–170.

Список рисунков

| | | |
|-----|--|----|
| 1.1 | Схема подсчета слагаемых M_{kt} | 31 |
| 1.2 | Пример множества решений задачи без ограничений. | 34 |
| 1.3 | Пример множества решений задачи с ограничениями (множества пересекаются). | 35 |
| 1.4 | Пример множества решений задачи с ограничениями (множества не пересекаются). | 37 |
| 4.1 | Узлы неравенства | 81 |

Приложение А

Программная реализация задачи с псевдоквадратичной целевой функцией

Для расчетов в главах 1 и 2 использовалось программное обеспечение, написанное на языке R , листинг которого приводится в настоящем приложении, а также размещен в репозитории по адресу <https://github.com/SovanSB/Idempotent/>. Эта программа позволяет проводить вычисления и решать рассмотренные выше задачи оптимизации в различных идемпотентных полуполях.

Структура программного обеспечения

В программе реализованы следующие функции:

- Функция мультипликативно сопряженного транспонирования — *conjInv*;
- Функция сложения матриц в идемпотентном полуполе — *parplus*;
- Функция перемножения матриц в идемпотентном полуполе — *multiply*;
- Оператор «звезда Клини» — *ast*;
- Функция вычисления следа матрицы в идемпотентном полуполе — *tr*;
- Функция вычисления тропического аналога определителя — *Tr*;
- Функция вычисления тропического спектрального радиуса матрицы — *spectr*;

- Функция, решающая задачу с псевдоквадратичной целевой функцией без ограничений (1.7), — *unconstr*;
- Функция вычисления необходимых компонент \mathbf{S}_{km} — *sCreate*;
- Функция, решающая задачу с псевдоквадратичной целевой функцией с ограничениями (1.11), — *constr*;
- Функции для работы в полуполе $\mathbb{R}_{\max,+}$;
- Функция нахождения минимума в расширенной задаче чебышевской аппроксимации (3.3), — *minimumFinder*;
- Функция разрежения матрицы в расширенной задаче чебышевской аппроксимации (3.3), — *sparsiify*;
- Функция получения модифицированной разреженной матрицы задачи, — *pmodify*;
- Функция для упорядочивания элементов в столбцах по возрастанию, — *ordering*;
- Функция поиска наилучшей строки из списка, — *maxRow*;
- Функция для отбрасывания избыточных границ, — *filterPoints*;
- Процедура поиска множества решений расширенной задачи чебышевской аппроксимации (3.3), — *nextmatr*.

Функция мультипликативно сопряженного транспонирования *conjInv* принимает на вход обязательный параметр — вектор, а также необязательные: функцию обращения числа *inv* и тропический нуль *zero*. Здесь и далее, если не указывать необязательные параметры, по умолчанию используется полуполе $\mathbb{R}_{\max,+}$.

Функция сложения матриц в идемпотентном полуполе *parplus* принимает на вход обязательный параметр — набор векторов/матриц, а также необязательные: функцию тропического сложения *plus* и параметр учета пропущенных значений *na.rm*. За основу для этой функции была взята встроенная функция

ptax. Используется в случае, если сложение матриц не задано явно другой функцией.

Функция перемножения матриц в идемпотентном полуполе *multiply* принимает на вход обязательные параметры — две матрицы подходящего размера, необязательные: функцию тропического сложения *plus* и функцию тропического умножения *mult*.

Оператор «звезда Клини» *ast* принимает на вход обязательный параметр — квадратную матрицу, необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, тропический нуль *zero*, тропическую единицу *identity*, функцию сложения матриц *pplus*.

Функция вычисления следа матрицы в идемпотентном полуполе *tr* принимает на вход обязательный параметр — квадратную матрицу, необязательный — функцию тропического сложения *plus*.

Функция вычисления тропического аналога определителя *Tr* принимает на вход обязательный параметр — квадратную матрицу, необязательные: функцию тропического сложения *plus* и функцию тропического умножения *mult*.

Функция вычисления спектрального радиуса матрицы в идемпотентном полуполе *spectr* принимает на вход обязательный параметр — квадратную матрицу, необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult* и функцию тропического возведения в степень *deg*.

Функция *unconstr*, решающая задачу без ограничений, принимает на вход обязательные параметры: матрицу \mathbf{A} , векторы \mathbf{p} , \mathbf{q} , скаляр r , необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, тропический нуль *zero*, тропическую единицу *identity*, функцию сложения матриц *pplus*, функцию тропического возведения в степень *deg*, функцию взятия обратного по умножению *inv*.

Функция *sCreate* для вычисления необходимых компонент \mathbf{S}_{km} принимает на вход обязательные параметры: матрицы \mathbf{A} и \mathbf{B} , необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, тропический нуль *zero*, тропическую единицу *identity*, функцию сложения матриц *pplus*. На выходе выдается список, содержащий *sArr* — массив матриц $\mathbf{S}_{k,n-1}$, $k = 0 \dots n - 1$, *snArr* — массив матриц \mathbf{S}_{kn} , $k = 0 \dots n$, а также исходные матрицы.

Функция *constr*, решающая задачу с ограничениями, принимает на вход обязательные параметры: матрицу \mathbf{A} , векторы \mathbf{p} , \mathbf{q} , скаляр r , матрицу \mathbf{B} , необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, тропический нуль *zero*, тропическую единицу *identity*, функцию сложения матриц *pplus*, функцию тропического возведения в степень *deg*, функцию взятия обратного по умножению *inv*.

Эти функции использовались для проведения расчетов в настоящей работе. Несмотря на то, что все примеры приводились в полуполе $\mathbb{R}_{\max,+}$, приведенные выше функции могут корректно работать и в иных полуполях при задании соответствующих этим полуполям необязательных параметров.

Листинг A.1: Исходные коды использованных функций

```

1 # Функция для мультипликативно сопряженного транспонирования матриц.
2 conjInv <- function(x, inv = maxplusinv, zero = -Inf) {
3   res <- matrix(zero, nrow(x), ncol(x))
4   res[x != zero] = inv(x[x != zero])
5   t(res)
6 }
7
8 # Функция перемножения матриц.
9 multiply <- function(A, B, plus = max, mult = add) {
10  if (ncol(A) != nrow(B))
11    stop("Incompatible matrices!")
12  rows <- nrow(A)
13  cols <- ncol(B)
14  res <- matrix(0., nrow = rows, ncol = cols)
15  for (i in 1:rows) {
16    for (j in 1:cols) {
17      res[i,j] <- plus(mult(A[i,], B[,j]))
18    }
19  }
20  res
21 }
22
23 # Функция сложения матриц. Используется, если сложение матриц не задано явно.
24 parplus <- function (... , plus = max, na.rm = FALSE)
25 {
26   elts <- list(...)
```

```

27  if (length(elts) == 0L)
28      stop("no arguments")
29  mmm <- elts[[1L]]
30  attr(mmm, "dim") <- NULL
31  has.na <- FALSE
32  for (each in elts[-1L]) {
33      attr(each, "dim") <- NULL
34      l1 <- length(each)
35      l2 <- length(mmm)
36      if (l2 < l1) {
37          if (l2 && l1%%l2)
38              warning("an argument will be fractionally recycled")
39          mmm <- rep(mmm, length.out = l1)
40      }
41      else if (l1 && l1 < l2) {
42          if (l2%%l1)
43              warning("an argument will be fractionally recycled")
44          each <- rep(each, length.out = l2)
45      }
46      nas <- cbind(is.na(mmm), is.na(each))
47      if (has.na || (has.na <- any(nas))) {
48          mmm[nas[, 1L]] <- each[nas[, 1L]]
49          each[nas[, 2L]] <- mmm[nas[, 2L]]
50      }
51      len <- length(mmm)
52      for (i in 1:len) {
53          mmm[i] <- plus(mmm[i], each[i])
54      }
55      if (has.na && !na.rm)
56          mmm[nas[, 1L] | nas[, 2L]] <- NA
57  }
58  mostattributes(mmm) <- attributes(elts[[1L]])
59  mmm
60 }
61
62 # Оператор Звезда<< Клини>>.
63 ast <- function(A, plus = max, mult = add, zero = -Inf,
64     identity = 0, pplus = NULL) {
65     d <- ncol(A)
66     if (d != nrow(A))

```



```

67     stop("Non-square matrix is given!")
68     id <- matrix(zero, d, d, byrow=TRUE)
69     if (d > 1)
70         diag(id) <- identity
71     else id <- identity
72     res <- id
73     temp <- A
74     if (is.null(pplus))
75         pplus <- function(...) parplus(..., plus=plus)
76     res <- pplus(res, temp)
77     if (d > 2) {
78         for (i in 1:(d-2)) {
79             temp <- multiply(temp, A, plus, mult)
80             res <- pplus(res, temp)
81         }
82     }
83     res
84 }
85
86 # Функция вычисления следа матрицы.
87 tr <- function(A, plus = max) {
88     d <- ncol(A)
89     if (d != nrow(A))
90         stop("Non-square matrix is given!")
91     if (d > 1)
92         temp <- plus(diag(A))
93     else
94         temp <- A[1,1]
95     temp
96 }
97
98 # Функция вычисления тропического аналога определителя.
99 Tr <- function(A, plus = max, mult = add) {
100     d <- ncol(A)
101     if (d != nrow(A))
102         warning("Non-square matrix is given!")
103     temp <- A
104     res <- tr(A, plus)
105     for (i in 2:d) {
106         temp <- multiply(temp, A, plus, mult)

```

```

107     res <- plus(res, tr(temp))
108   }
109   res
110 }
111
112 # Функция вычисления спектрального радиуса матрицы.
113 spectr <- function(A, plus = max, mult = add, deg = div) {
114   d <- ncol(A)
115   if (d != nrow(A))
116     warning("Non-square matrix is given!")
117   temp <- A
118   res <- tr(A)
119   for (i in 1:(d-1)) {
120     temp <- multiply(temp, A, plus, mult)
121     res <- plus(res, deg(tr(temp, plus), 1/(i+1)))
122   }
123   res
124 }
125
126 # Функция, решающая задачу без ограничений.
127 unconstr <- function(A, p, q, r, plus = max, mult = add,
128   zero = -Inf, identity = 0, pplus = NULL,
129   deg = div, inv = maxplusinv) {
130   lambda <- spectr(A, plus, mult, deg)
131   if (lambda == zero)
132     stop("Incorrect matrix: eigenvalue equals zero!")
133   myu <- plus(lambda, r)
134   qm <- conjInv(q, inv = inv, zero=zero)
135   d <- nrow(A)
136   temp <- matrix(zero, d, d, byrow=TRUE)
137   diag(temp) <- identity
138   for (i in 1:d) {
139     myu <- plus(myu, deg(multiply(multiply(qm, temp, plus, mult),
140       p, plus, mult), 1/(i + 1)))
141     temp <- multiply(temp, A, plus, mult)
142   }
143   myuminus <- inv(myu)
144   matr <- ast(mult(myuminus, A), plus, mult, zero, identity, pplus)
145   left <- mult(myuminus, p)
146   right <- mult(myu, conjInv(multiply(qm, matr, plus, mult),

```

```

147         inv = inv, zero = zero))
148     list(myu = myu, matr = matr, left = left, right = right, A = A,
149         p = p, q = q, r = r)
150 }
151
152 # Функция, вычисляющая S_{km}
153 sCreate <- function(A, B, plus = max, mult = add, zero = -Inf,
154     identity = 0, pplus = NULL) {
155     n <- nrow(A)
156     # Массивы, используемые для вычисления M_{km}. Переиспользуются.
157     mArr <- array(zero, c(n, n, n + 1))
158     mNew <- array(zero, c(n, n, n + 1))
159     # Массив, отвечающий за S_{k,n-1} после завершения работы функции.
160     # В процессе работы в нем хранятся S_{km} предыдущих слоев.
161     sArr <- array(zero, c(n, n, n + 1))
162     # Массив, отвечающий за S_{kn}.
163     snArr <- array(zero, c(n, n, n + 1))
164     # Необходимо добавить единичную матрицу M_{00}.
165     diag(sArr[, , 1]) <- identity
166     mArr[, , 1] <- B
167     mArr[, , 2] <- A
168     if (is.null(pplus))
169         pplus <- function(...) parplus(..., plus=plus)
170     for (i in 2:n) {
171         for (j in 1:(i + 1))
172             sArr[, , j] <- pplus(sArr[, , j], mNew[, , j])
173         # Вычисляем следующий слой M_{km}.
174         mNew[, , 1] <- multiply(B, mArr[, , 1], plus, mult)
175         mNew[, , i + 1] <- multiply(A, mArr[, , i], plus, mult)
176         for (j in 2:i)
177             mNew[, , j] <- pplus(multiply(A, mArr[, , j - 1], plus, mult),
178                 multiply(B, mArr[, , j]))
179         mArr <- mNew
180     }
181     snArr <- pplus(sArr, mNew)
182     list(sArr = sArr[, , 1:n], snArr = snArr, A = A, B = B)
183 }
184
185 # Функция, решающая задачу с ограничениями.
186 constr <- function(A, p, q, r, B, plus = max, mult = add,

```

```

187     zero = -Inf, identity = 0, pplus = pmax, deg = div,
188     inv = maxplusinv) {
189 lambda <- spectr(A, plus, mult, deg)
190 if (lambda == zero)
191     stop("Incorrect matrix: eigenvalue equals zero!")
192 # Проверяем, что  $\text{Tr}(B) \leq \mathbb{1}$ .
193 # Пользуемся тем, что  $x \leq y \Leftrightarrow x \oplus y = y$ .
194 trB <- Tr(B, plus, mult)
195 if ((trB != identity) && (plus(trB, identity) == trB))
196     stop("Incorrect matrix:  $\text{Tr}(B) > \mathbb{1}$ ")
197 n <- nrow(A)
198 temp <- sCreate(A, B, plus, mult, zero, identity, pplus)
199 myu <- r
200 for (i in 1:n) {
201     myu <- plus(myu, deg(tr(temp$sArr[,i+1], plus), 1/i),
202                 deg(multiply(multiply(
203                     conjInv(q, inv=inv, zero=zero),
204                     temp$sArr[,i], plus, mult),
205                     p, plus, mult), 1/(i + 1)))
206 }
207 myuminus <- inv(myu)
208 if (is.null(pplus))
209     pplus <- function(...) parplus(..., plus=plus)
210 matr <- ast(pplus(mult(myuminus, A), B), plus, mult, zero,
211             identity)
212 qm <- conjInv(q, inv = inv, zero = zero)
213 left <- mult(myuminus, p)
214 right <- mult(myu, conjInv(multiply(qm, matr, plus, mult),
215                             inv = inv, zero = zero))
216 xleft <- multiply(matr, left, plus, mult)
217 xright <- multiply(matr, right, plus, mult)
218 list(myu = myu, matr = matr, left = left, right = right, A = A,
219       p = p, q = q, r = r, B = B, xleft = xleft, xright = xright)
220 }
221
222 # Функция умножения чисел в полуполе  $R_{\{\max,+\}}$ .
223 add <- function(x, y) x + y
224
225 # Функция возведения в степень в полуполе  $R_{\{\max,+\}}$ .
226 div <- function(x, m) x * m

```

```
227  
228 # Функция взятия обратного в полуполе  $R_{\{\max, +\}}$ .  
229 maxplusinv <- function(x) -x
```

Приложение Б

Программная реализация расширенной задачи псевдочебышевской аппроксимации

Для расчетов в главе 3 использовались следующие процедуры, написанные на языке *R*, листинг которых приводится в настоящем приложении, а также размещен в репозитории по адресу <https://github.com/SovanSB/Idempotent/>. С их помощью можно проводить вычисления и решать рассмотренные выше задачи оптимизации в различных идемпотентных полуполях. Для их работы требуются основные функции работы с полуполями, исходные коды которых приведены в приложении А.

Структура программного обеспечения

В программе реализованы следующие функции:

- Функция нахождения минимума в расширенной задаче чебышевской аппроксимации (3.3), — *minimumFinder*;
- Функция разрежения матрицы в расширенной задаче чебышевской аппроксимации (3.3), — *sparsify*;
- Функция получения модифицированной разреженной матрицы задачи, — *pmodify*;

- Функция для упорядочивания элементов в столбцах по возрастанию, — *ordering*;
- Функция поиска наилучшей строки из списка, — *maxRow*;
- Функция для отбрасывания избыточных границ, — *filterPoints*;
- Процедура поиска множества решений расширенной задачи чебышевской аппроксимации (3.3), — *nextmatr*.

Функция *minimumFinder* для вычисления минимума целевой функции по лемме 5 в задаче (3.3) принимает на вход обязательные параметры: матрицу \mathbf{A} , векторы \mathbf{p} , \mathbf{q} , необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, функцию тропического возведения в степень *deg*, тропический нуль *zero*, тропическую единицу *identity*, функцию взятия обратного по умножению *inv*, функцию сложения матриц *pplus*.

Функция *sparsify* для нахождения разреженной матрицы задачи (3.3) принимает на вход обязательные параметры: матрицу \mathbf{A} , векторы \mathbf{p} , \mathbf{q} , необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, функцию тропического возведения в степень *deg*, тропический нуль *zero*, тропическую единицу *identity*, функцию взятия обратного по умножению *inv*, функцию сложения матриц *pplus*, функцию нахождения меньшего элемента *less*.

Функция *pmodify* для получения матрицы $\hat{\mathbf{A}}'$ для упрощения процедуры выбора следующей рассматриваемой строки принимает на вход обязательные параметры: матрицу \mathbf{A} , вектор \mathbf{p} , необязательные: функцию тропического умножения *mult*, функцию взятия обратного по умножению *inv*.

Функция *ordering* для создания порядковой матрицы для упрощения процедуры выбора следующей рассматриваемой строки принимает на вход обязательные параметры: матрицу \mathbf{A} , необязательные: функцию тропического сложения *plus*, тропический нуль *zero*. Большой номер в ячейке результирующей порядковой матрице отражает большее количество строк, которые будут зафиксированные при выборе соответствующего элемента разреженной матрицы.

Функция *maxRow* для создания порядковой матрицы для упрощения процедуры выбора следующей рассматриваемой строки принимает на вход обязательные параметры: матрицу *ordMatr*, и вектор строк для рассмотрения *numbers*.

Функция *filterPoints* для отбрасывания избыточных границ путём их попарного сравнения принимает на вход обязательный параметр: матрицу границ *cur*, необязательный: функцию сложения матриц *pplus*.

Рекурсивная функция *nextmatr* для поиска множества всех решений задачи (3.3) принимает на вход обязательные параметры: матрицу **A**, векторы **p**, **q**, служебные: *prev*, *selected*, *sorted* используются при рекурсивных вызовах функции, необязательные: функцию тропического сложения *plus*, функцию тропического умножения *mult*, функцию тропического возведения в степень *deg*, тропический нуль *zero*, тропическую единицу *identity*, функцию взятия обратного по умножению *inv*, функцию сложения матриц *pplus*, функцию нахождения меньшего элемента *less*, функция *ordering* преобразования матрицы **A** в желаемую порядковую, функция *maxRow* поиска наилучшей строки из списка, флаг *decreasing*, отражающий то, что строки меньшего порядка имеют больший приоритет. С помощью задания функций *ordering*, *maxRow* и флага *decreasing* можно встроить собственные дискриминационные функции для строк.

Эти функции использовались для проведения расчетов в главе 3. Несмотря на то, что все примеры рассматривались в полуполе $\mathbb{R}_{\max,+}$, приведенные выше функции могут корректно работать и в иных полуполях при задании соответствующих этим полу полям необязательных параметров.

Листинг Б.1: Исходные коды использованных функций

```

1
2 # Функция вычисления меньшего значения в полуполе  $\mathbb{R}_{\max,+}$ .
3 maxplusless <- function(x, y) x < y
4
5 # Функция нахождения минимума в задаче аппроксимации
6 minimumFinder <- function(A, p, q, plus=max, mult=add, deg=div,
7   zero=-Inf, identity=0, inv=maxplusinv, pplus=pmax) {
8   delta <- deg(multiply(conjInv(multiply(A, q, plus, mult),
9     inv, zero), p, plus, mult), 1/2)
10  delta[1,1]
11 }
```



```

12
13 # Функция разрежения матрицы в задаче аппроксимации
14 sparsify <- function(A, p, q, plus=max, mult=add, deg=div,
15   zero=-Inf, identity=0, inv=maxplusinv, pplus=pmax,
16   less=maxplusless) {
17   delta <- minimumFinder(A, p, q, plus, mult, deg, zero,
18     identity, inv, pplus)
19   n <- ncol(A)
20   As <- A
21   delta2 <- inv(mult(delta, delta))
22
23   for (i in 1:n) {
24     for (j in 1:n) {
25       if (less(A[i, j], mult(delta2, mult(p[i], inv(q[j]))))) {
26         # Обнуляем элементы ниже порогового значения
27         As[i, j] = zero
28       }
29     }
30   }
31   As
32 }
33
34 # Функция для получения матрицы  $\hat{A}'$ 
35 pmodify <- function(A, p, mult=add, inv=maxplusinv) {
36   n <- ncol(A)
37   k <- ncol(A)
38   mA <- matrix(NA, n, n)
39   for (i in 1:k) {
40     for (j in 1:n) {
41       mA[i, j] <- mult(A[i, j], inv(p[i, 1]))
42     }
43   }
44   mA
45 }
46
47 # Функция для создания порядковой матрицы
48 ordering <- function(A, plus=max, zero=-Inf) {
49   n <- ncol(A)
50   k <- nrow(A)
51   fixA <- A

```

```

52 ord <- matrix(rep(1:k, times=n), k, n)
53 tempord <- matrix(rep(1:k, times=n), k, n)
54 for (j in 1:n) {
55   for (i in 1:(k-1)) {
56     for (m in 1:(k-i)) {
57 # Упорядочиваем элементы столбца по возрастанию
58       if (plus(A[m,j], A[m+1,j]) == A[m,j]) {
59         tV <- A[m+1,j]
60         tN <- ord[m+1,j]
61         A[m+1,j] <- A[m,j]
62         ord[m+1,j] <- ord[m,j]
63         A[m,j] <- tV
64         ord[m,j] <- tN
65       }
66     }
67   }
68   tempord[,j] <- ord[,j]
69 }
70
71 res <- matrix(NA, k, n)
72 for (j in 1:n) {
73   res[tempord[,j],j] <- 1:k
74 # Назначаем одинаковые номера равным элементам
75   for (m in 1:(k-1)) {
76     if (plus(A[m,j], A[m+1,j]) == A[m,j]) {
77       res[ord[m,j],j] <- max(res[ord[m,j],j],
78                             res[ord[m+1,j],j])
79       res[ord[m+1,j],j] <- res[ord[m,j],j]
80     }
81   }
82   for (m in (k-1):1) {
83     if (plus(A[m,j], A[m+1,j]) == A[m,j]) {
84       res[ord[m,j],j] <- max(res[ord[m,j],j],
85                             res[ord[m+1,j],j])
86       res[ord[m+1,j],j] <- res[ord[m,j],j]
87     }
88   }
89 }
90 # Инвертируем порядок для удобства
91 res <- res - 1

```

```
92   res <- n-1 - res
93   res[fixA==zero] <- n
94   res
95 }
96
97 # Функция отбрасывания избыточных границ
98 filterPoints <- function(cur, pplus=pmax) {
99   if (is.null(dim(cur)) || is.na(dim(cur)[2])) {
100     if (is.null(cur)) warning("Ooops, cur is NULL")
101     dim(cur) <- c(length(cur), 1)
102   }
103   k <- ncol(cur)
104   toAdd <- matrix(TRUE, k, 1)
105   res <- NULL
106   if (k > 1) {
107     for (i in 1:(k-1)) {
108       if(toAdd[i,1]) {
109         for (j in (i+1):k) {
110           if(toAdd[j,]) {
111             temp <- pplus(cur[,i], cur[,j])
112             if (all(temp == cur[,i])) {
113               toAdd[i,1] <- FALSE
114               break
115             }
116             if (all(temp == cur[,j])) {
117               toAdd[j,1] <- FALSE
118             }
119           }
120         }
121       }
122     }
123     res <- cur[,toAdd[,1]]
124   }
125   else {
126     res <- cur[,1]
127   }
128   res
129 }
130
131 # Процедура выбора следующей строки порядковой матрицы
```

```

132 maxRow <- function(ordMatr, numbers) {
133   num <- length(numbers)
134   n <- ncol(ordMatr)
135   maxNum <- 1
136   maxValue <- -Inf
137   # Если осталась единственная строка, возвращаем номер
138   if (num < 2) {
139     maxNum <- numbers[[1]]
140   }
141   # Иначе считаем вес каждой из строк
142   else {
143     # Чем больше строк будет зафиксировано, тем лучше
144     for (i in 1:num) {
145       temp <- sum(ordMatr[numbers[[i]],], na.rm=T) -
146         n/2 * sum(ordMatr[numbers[[i]],]==n)
147       # Вес нулевого значения считаем n/2
148       if (temp > maxValue) {
149         maxValue <- temp
150         maxNum <- numbers[[i]]
151       }
152     }
153   }
154   # Возвращаем номер наилучшей строки
155   maxNum
156 }
157
158 # Процедура поиска множества решений задачи аппроксимации
159 # На первом шаге переменные prev, selected, sorted = NULL
160 nextmatr <- function(A, p, q, prev=NULL,
161   selected=NULL, sorted=NULL, plus=max,
162   mult=add, deg=div, zero=-Inf, identity=0,
163   inv=maxplusinv, pplus=pmax, less=maxplusless,
164   ordering=ordering, maxRow=maxRow, decreasingA=) {
165   n <- ncol(A)
166   quantity <- 0
167   # На первом шаге добавляем все строки для рассмотрения
168   fut <- 1:n
169   # Если шаг не первый, то prev -- зафиксированные строки
170   if (!is.null(prev)) {
171     # Удаляем их из будущего рассмотрения

```

```

172     fut <- fut[-prev]
173 }
174 else {
175 # Мы на первом шаге, получаем разреженную матрицу
176   A <- sparsify(A, p, q, plus, mult, deg, zero,
177               identity, inv, pplus, less)
178 # Для нулевых компонент p исследование не требуется
179   pnull <- which(p == zero)
180   if (length(pnull) > 0) {
181     prev <- c(prev, pnull)
182     for (i in pnull) {
183       selected <- c(selected,
184                     which(A[pnull,] != zero)[[1]])
185     }
186     fut <- fut[-pnull]
187   }
188 # Вычисляем порядковую матрицу
189   sorted <- ordering(pmodify(A, p, mult, inv),
190                     plus, zero)
191 }
192
193 ll <- NULL
194
195 # Если для рассмотрения осталось более одной строки
196 if (length(fut) > 1) {
197 # Выбираем из них наилучшую
198   best <- maxRow(sorted, fut)
199   if (best < 1) print("Error, best < 1")
200   if (best > n) print("Error, best > n")
201 # Исследуем все ненулевые значения в наилучшей строке
202   for (j in 1:n) {
203     if (A[best, j] != zero) {
204 # В зависимости от настроек decreasing фиксируем строки
205       if (decreasing) {
206         toFix <- which((sorted[fut, j] >= sorted[best, j])
207                       & (A[fut, j] != zero))
208       }
209       else {
210         toFix <- which((sorted[fut, j] <= sorted[best, j])
211                       & (A[fut, j] != zero))

```

```

212     }
213     toDel <- fut[toFix]
214     count <- length(toFix)
215 # Определяем оставшиеся для исследования строки
216     toStay <- fut[-toFix]
217 # Если такие строки есть, то рекурсивно исследуем их
218     if (length(toStay > 0)) {
219         res <- nextmatr(
220             A, p, q, c(prev,toDel),
221                 c(selected, rep(j, count)), sorted,
222                 plus, mult, deg, zero, identity, inv, pplus,
223                 less, ordering, maxRow, decreasing
224         )
225         ll <- cbind(ll, res[[1]])
226         quantity <- quantity + res[[2]]
227     }
228     else
229     {
230 # Если в текущей ветке не осталось неисследованных строк
231     toMult <- matrix(zero, n, n)
232     bindedList <- cbind(c(prev,toDel),
233                         c(selected, rep(j, count)))
234     toMult[bindedList] <- A[bindedList]
235     delta = minimumFinder(A, p, q, plus, mult, deg,
236                           zero, identity, inv, pplus)
237 # То формируем список найденных решений
238     ll <- cbind(ll, mult(inv(delta),
239                         multiply(conjInv(toMult, inv, zero),
240                                 p, plus, mult)))
241     quantity <- quantity + 1
242     }
243     }
244     }
245     }
246     else {
247 # Осталась последняя строка, исследуем ненулевые значения
248     num = sum(A[fut,] != zero)
249     ret <- matrix(zero, n, num)
250     counter = 1
251     tempA <- matrix(zero, n, n)

```

```
252     bindedList <- cbind(prev,selected)
253     tempA[bindedList] <- A[bindedList]
254     delta = minimumFinder(A, p, q, plus, mult, deg,
255         zero, identity, inv, pplus)
256 # Каждое из ненулевых значений формирует нижнюю границу
257     for (j in 1:n) {
258         if (A[fut,j] != zero) {
259             toMult <- tempA
260             toMult[fut, j] <- A[fut, j]
261             ret[,counter] <- mult(inv(delta), multiply(
262                 conjInv(toMult, inv, zero), p, plus, mult))
263             counter = counter + 1
264         }
265     }
266 # Формируем из них список, считаем общее число границ
267     ll <- cbind(ll, ret)
268     quantity <- num
269 }
270 # Отфильтровываем несущественные границы и возвращаем их
271     list(fp=filterPoints(ll), quant=quantity)
272 }
```

Приложение В

Схема поиска решений неравенства

В приложении с использованием псевдокода приведена схема поиска решений неравенства 4.1, рассмотренная в главе 4

Сначала нормируем матрицу \mathbf{A} как указано в (4.1.1). Создадим массив из m осей (списков узлов) и заполним его по принципу, описанному в (4.1.2) Введем следующие обозначения (переменные):

coordlist – список координат, для которых неравенство уже выполнено (изначально пустой)

nodelist – список уже добавленных узлов вместе с координатами, на которых стало выполняться неравенство после добавления данных узлов (изначально пустой)

cur – номер текущей оси (изначально равен единице)

nextnode – следующий узел на оси, NULL, если предыдущий был последним. Под $++nextnode$ будем понимать переход к следующему узлу. Под $nodelist := nodelist - nextnode$ - удаление из всех предыдущих узлов координат, которые встречаются в *nextnode*. Если при этом в каком-нибудь узле не останется координат: $hasNull(nodelist) == TRUE$, то есть добавление одного из предыдущих узлов стало бессмысленным, то возвращаемся на уровень выше. (изначально список пустой)

(все переменные используются локально на каждом уровне, т.е. для каждого уровня создается своя копия, которая впоследствии модифицируется)

$Addlist(coordlist, nodelist, nextnode)$ – процедура, которая пытается добавить следующий узел, действует по алгоритму: (Алгоритм 2)

Рекурсивно вызываем процедуру $Gfind(coordlist, nodelist, cur)$ (Алгоритм 3).

После завершения работы данной процедуры у нас оказывается набор «граней», то есть минимальный набор таких векторов \mathbf{v}_i , что неравенство $\mathbf{A}\mathbf{v} \geq \mathbf{b}$ выполнено для любого вектора \mathbf{v} , для которого $\mathbf{v} \geq \mathbf{v}_i$ при каком-то i . При этом, если вектор \mathbf{u} таков, что не существует индекса i , для которого справедливо неравенство $\mathbf{u} \geq \mathbf{v}_i$, то $\mathbf{A}\mathbf{u} \not\geq \mathbf{b}$.

Если требуется получить набор непересекающихся множеств, то его можно получить с помощью стандартной процедуры: сопоставим каждой «граней» полуоткрытый справа интервал (то есть пару векторов), который она задает. Произвольным образом перенумеруем все грани, получив упорядоченный набор множеств. После этого начинаем по порядку попарно сравнивать каждое множество с остальными. Если есть пересечение множества \mathbf{X} с каким-то множеством \mathbf{Y} , то заменяем множество с большим индексом (то есть \mathbf{Y}) на их разность $\mathbf{Y} \setminus \mathbf{X}$ (в теоретико-множественном смысле) и продолжаем сравнения дальше.

Algorithm 2 Процедура Addlist для добавления новых узлов.

```

1: function ADDLIST(out coordlist, out nodelist, out nextnode)
2:   if nextnode == NULL then                                ▷ Дошли до края координатной оси
3:     return FALSE
4:   end if
5:   nodelist ← nodelist - nextnode    ▷ Убираем повторяющиеся координаты
6:                                       ▷ из добавленных узлов, чтобы узлы отображали
7:                                       ▷ какие уникальные координаты они привносят
8:   if HASNULL(nodelist) == TRUE then ▷ Если в каком-то из предыдущих
9:                                       ▷ узлов не осталось уникальных координат
10:    return FALSE    ▷ то удалим этот узел: конфигурация не изменится
11:  end if
12:  if nextnode ∈ coordlist then
13:    return ADDLIST(coordlist, nodelist, ++nextnode)
14:                                       ▷ На данном шаге не удалось добавить ничего нового
15:                                       ▷ пытаемся добавить в следующем узле, проверяя
16:                                       ▷ не стало ли бессмысленным добавление какого-либо
17:                                       ▷ узла на прошлых уровнях
18:  else
19:    nodelist ← nodelist + nextnode    ▷ Добавляем новый узел
20:    coordlist ← coordlist + nextnode  ▷ и координаты в списки
21:    ++nextnode
22:    return TRUE
23:  end if
24: end function

```

Algorithm 3 Процедура Gfind: Рекурсивно ищет новые «границы», при нахождении распечатывает и продолжает работу.

```

1: function GFIND(coordlist, nodelist, cur)
2:   if cur < m then                                ▷ Если находимся не на последней оси
3:     GFIND(coordlist, nodelist, cur + 1)           ▷ то спускаемся на уровень ниже
4:     while ADDLIST(coordlist, nodelist, nextnode) == TRUE do
5:       if coordlist is full then                  ▷ Найдена новая грань
6:         PRINT(nodelist)                            ▷ Фиксируем ее и возвращаемся выше
7:         return
8:       else                                         ▷ Addlist изменила переменные, добавила узлы
9:         GFIND(coordlist, nodelist, cur+1)
10:      end if
11:     end while
12:   else                                             ▷ Это последняя ось, необходимо добрать недостающие индексы
13:     ▷ или вернуться на уровень выше, если это невозможно
14:     while !(coordlist is full) do
15:       if ADDLIST(coordlist, nodelist, nextnode) == TRUE then
16:         nextnode++
17:       else
18:         return
19:       end if
20:     end while
21:     PRINT(nodelist)                                 ▷ Выводим найденную грань
22:     return
23:   end if
24:   return                                           ▷ Ничего не нашли, возвращаемся выше
25: end function

```
