

Московский физико-технический институт
(государственный университет)

Факультет радиотехники и кибернетики
Кафедра предсказательного моделирования и оптимизации

Новый метод многокритериальной оптимизации, основанный на локальной геометрии множества Парето

Выпускная квалификационная
работа на степень магистра
студента 711гр.
Пересторонина Н.О.

Научный руководитель
к.ф.-м.н. Губарев Ф.В.

г. Москва,
2013г.

Аннотация

В инженерных приложениях часто встречаются случаи, когда систему необходимо оптимизировать по нескольким критериям одновременно. Ответ в таких задачах не единственный и представляет собой множество Парето-оптимальных решений. Зная его локальную геометрию и несколько оптимальных точек, можно эффективно находить решение задачи. В работе описан метод решения гладких нелинейных задач многокритериальной оптимизации с ограничениями, позволяющий находить как одну Парето-оптимальную точку, так и восстанавливать весь Парето-фронт. Проведенное сравнение метода с альтернативными способами решения многокритериальных задач, такими как генетические алгоритмы и скаляризация, показывает высокую эффективность предложенного алгоритма.

Оглавление

1	Задача оптимизации	1
1.1	Постановка задачи	1
1.2	Парето-оптимальность	3
1.3	Формулировки многокритериальной задачи	4
2	Способы решения задач многокритериальной оптимизации	6
2.1	Сведение задачи к однокритериальной	6
2.2	Эволюционные алгоритмы	8
2.3	Цель работы	9
3	Поиск Парето-оптимального решения	10
3.1	Теория	10
3.1.1	Однокритериальная задача	10
3.1.2	Многокритериальная задача с ограничениями	11
3.1.3	Многокритериальная задача без ограничений	12
3.2	Метод наискорейшего спуска	13
3.3	Метод Ньютона	15
3.4	Выбор длины шага	16
3.5	Практические аспекты	17
4	Нахождение Парето-фронта	19
4.1	Локальная геометрия множества Парето	19
4.2	Рассеивание	21
4.3	Описание алгоритма	23
5	Практические результаты	25
5.1	Методика сравнения	25
5.2	Задачи	27
5.2.1	ZDT1	27
5.2.2	ZDT2	28
5.2.3	TP7	29
6	Заключение	31

Список иллюстраций

1.1	Локальный и глобальный минимумы	2
1.2	Парето-множество и Парето-фронт	3
1.3	Формулировки многокритериальной задачи	4
a	Локальная	4
b	Глобальная	4
2.1	Методы скаляризации	7
a	Метод взвешенных сумм	7
b	Метод ε -ограничений	7
3.1	Возможные направления спуска	12
a	Далеко от решения	12
b	Близко к решению	12
3.2	Градиенты в Парето-оптимальной точке	13
3.3	Правило Армихо	17
4.1	Шаг вдоль множества Парето	22
a	Пространство параметров	22
b	Пространство целевых функций	22
4.2	Нахождение Парето-фронта	23
5.1	Метрика Q	26
5.2	Парето-фронты для задачи ZDT1 с $n = 2$	28
5.3	Парето-фронты для задачи ZDT2 с $n = 2$	29
5.4	Решение задачи TP7 алгоритмом GTOpt	30
5.5	Решение задачи TP7 алгоритмом NSGA2	30

Глава 1

Задача оптимизации

1.1 Постановка задачи

Оптимизационные задачи встречаются повсеместно. Экономисты минимизируют издержки и максимизируют прибыль. Производители оптимизируют рабочие процессы для того, чтобы добиться максимального выпуска продукции при минимальном браке. Инженеры стремятся снизить аэродинамическое сопротивление и вес самолетов, соблюдая при этом высокие требования к прочности и надежности. Наконец, многие физические явления описываются в виде экстремальных принципов. Луч света движется по пути, минимизирующему время движения.

Для формализации конкретной задачи и записи ее на математическом языке нужно, в первую очередь, определить *целевые функции*. Целевая функция — это количественная мера производительности системы, та величина, которую мы хотим минимизировать или максимизировать: время, прибыль, аэродинамическое сопротивление. Целевая функция зависит от *переменных* — неизвестных величин, варьируемых в процессе оптимизации так, чтобы получить оптимальное решение. Помимо этого, в задаче часто присутствуют *ограничения*: процент брака не должен превышать определенной величины, конструкция самолета должна обладать заданным запасом прочности.

Сформулируем теперь задачу оптимизации в общем виде.

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{f}(\mathbf{x}) \tag{1.1}$$

при ограничениях

$$\mathbf{g}(\mathbf{x}) \leq 0,$$

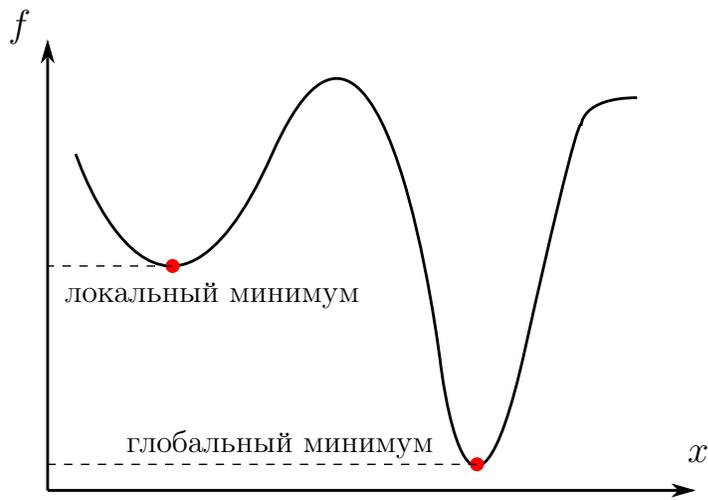


Рис. 1.1: Локальный и глобальный минимумы

где

$\mathbf{x} = (x_1, \dots, x_n)$ — вектор переменных (параметров),
 $\mathbf{f} = (f_1, \dots, f_k)$ — векторнозначная целевая функция,
 $\mathbf{g} = (g_1, \dots, g_m)$ — векторнозначная функция ограничений.

Точка \mathbf{x} , удовлетворяющая всем ограничениям, называется *допустимой*. Множество всех допустимых точек обозначим $\Omega = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \leq 0\}$.

Если \mathbf{f} состоит из одной компоненты (скалярная функция), оптимизация называется *однокритериальной*, иначе — *многокритериальной*.

Используя введенные выше обозначения, перепишем 1.1:

$$f(\mathbf{x}) \rightarrow \min, \mathbf{x} \in \Omega \text{ — однокритериальная оптимизация,} \quad (1.2)$$

$$(f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \rightarrow \min, \mathbf{x} \in \Omega \text{ — многокритериальная оптимизация.} \quad (1.3)$$

Рассмотрим решение однокритериальной задачи.

Определение 1. Точка \mathbf{x}^* называется *глобальным минимумом* функции f если $f(\mathbf{x}^*) < f(\mathbf{x})$ для $\forall \mathbf{x} \in \Omega$.

Определение 2. Точка \mathbf{x}^* называется *локальным минимумом* функции f если существует окрестность \mathcal{N} точки \mathbf{x}^* такая, что $f(\mathbf{x}^*) < f(\mathbf{x})$ для $\forall \mathbf{x} \in \mathcal{N}$.

Различие этих двух определений показано на Рис. 1.1.

Для многокритериального случая важными являются понятия Парето-доминирования и Парето-оптимальности. Рассмотрим их подробнее.

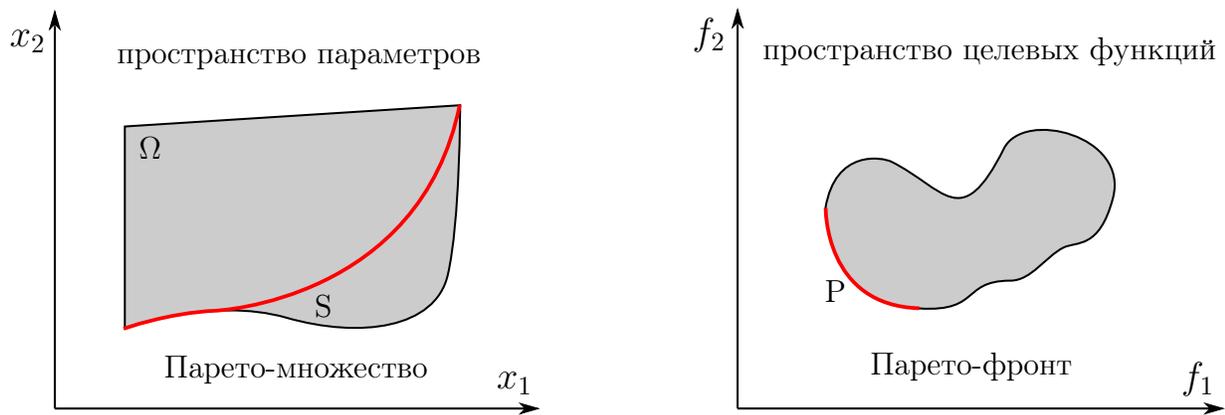


Рис. 1.2: Парето-множество и Парето-фронт

1.2 Парето-оптимальность

Основная идея Парето-оптимальности (названа в честь итальянского экономиста Вильфредо Парето (1848–1923), который использовал это понятие в своих работах по экономической эффективности): мы не можем улучшить наше решение по одному из показателей, не ухудшив при этом по другому.

Пусть $\mathbf{a} = (a_1, \dots, a_n)$, $\mathbf{b} = (b_1, \dots, b_n) \in \mathbb{R}^n$.

Определение 3. \mathbf{a} доминирует \mathbf{b} ($\mathbf{a} \succ \mathbf{b}$), если $a_i \leq b_i$ для всех i от 1 до n и $a_i < b_i$ по крайней мере для одного из $i \in \{1, \dots, n\}$. Иначе, считаем, что \mathbf{a} не доминирует \mathbf{b} ($\mathbf{a} \not\succeq \mathbf{b}$).

Если $(\mathbf{a} \not\succeq \mathbf{b})$ и $(\mathbf{b} \not\succeq \mathbf{a})$, то \mathbf{a} не сравнимо с \mathbf{b} ($\mathbf{a} \not\ll \mathbf{b}$).

В дальнейшем, если говорится об отношениях доминирования между двумя множествами векторов из пространства параметров, то под этим понимаются соотношения для их образов в пространстве целевых функций.

Решением задачи многокритериальной оптимизации является *Парето-множество* $S \subset \Omega$ всех *Парето-оптимальных* допустимых точек, т.е. таких, которые не доминируются никакими другими допустимыми точками. Парето-множеству S соответствует *Парето-фронт* $P = \mathbf{f}(S)$ (образ Парето-множества в пространстве целевых функций) (см. Рис. 1.2).

Таким образом, в отличие от однокритериальной задачи, ответом является не одна точка, а множество точек. В Парето-множестве точки не сравнимы между собой, т.е. все решения задачи равноценны.

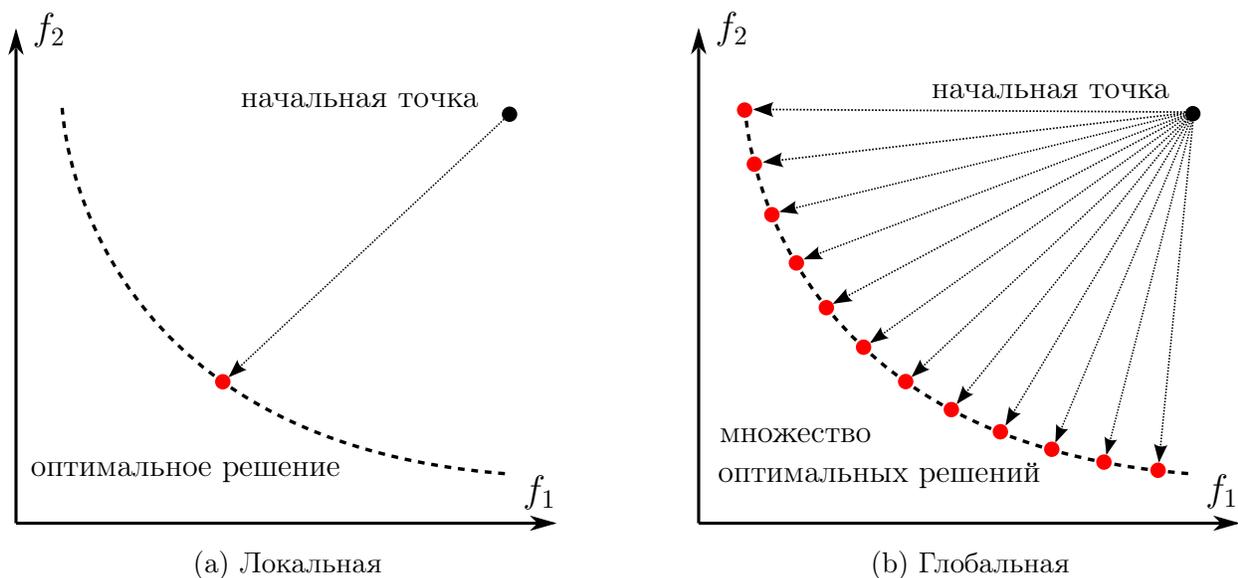


Рис. 1.3: Формулировки многокритериальной задачи

1.3 Формулировки многокритериальной задачи

Оптимизационные алгоритмы итеративны. Стартуя с некоторого начального приближения, алгоритм генерирует последовательность точек до тех пор, пока не будет выполнено условие останова. Таким образом, в процессе оптимизации целевые функции и ограничения вычисляются на конечном множестве точек. При этом, множество Парето-оптимальных решений в общем случае содержит бесконечное количество точек. Соответственно, на практике задача многокритериальной оптимизации переформулируется так, чтобы ее можно было решить с помощью итеративного алгоритма. Существуют по крайней мере три таких формулировки.

Локальная

Найти одно Парето-оптимальное решение (ближайшее к заданной начальной точке) (Рис. 1.3а).

Глобальная

Найти конечное множество Парето-оптимальных решений, достаточно хорошо описывающее (покрывающее) истинный Парето-фронт (Рис. 1.3б).

Интерактивная

Найти Парето-оптимальное решение, максимально удовлетворяющее предпочтениям *эксперта*.

Последняя формулировка предполагает участие в процессе решения эксперта — человека, принимающего решения и разбирающегося в предметной области. Задача алгоритма — предоставить эксперту вспомогательную информацию о текущем решении

(например, насколько увеличится одна из целевых функций, если попытаться уменьшить другую), и возможность выразить свои предпочтения относительно решения (например, тем или иным образом классифицировать целевые функции в данной точке). Предпочтения эксперта затем учитываются при нахождении следующего решения, информация о котором также предоставляется пользователю. Подробнее про интерактивные алгоритмы можно прочитать, например в [1].

В дальнейшем будут рассматриваться только первые две задачи (локальная и глобальная).

Глава 2

Способы решения задач многокритериальной оптимизации

В предыдущей главе была приведена строгая математическая формулировка многокритериальной задачи и ее решения. Перейдем теперь от теории в практику и обсудим способы решения таких задач.

2.1 Сведение задачи к однокритериальной

Одним из способов решения задач 1.3 является *скаляризация* — сведение задачи к однокритериальной 1.2. Однокритериальная оптимизация исследуется сравнительно давно, существуют эффективные алгоритмы общего назначения, решающие задачи нелинейной оптимизации с ограничениями, такие как *последовательное квадратичное программирование* (Sequential quadratic programming, SQP). Подробное описание теории и практики однокритериальной оптимизации, в том числе метода SQP, можно найти в книге [2].

Один из методов скаляризации — построение вспомогательной целевой функции $f : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$f(\mathbf{x}) = h(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \quad (2.1)$$

путем введения функционала $h : \mathbb{R}^k \rightarrow \mathbb{R}$, принимающего значения компонент векторной целевой функции, и сводящего, таким образом, задачу к однокритериальной. Функция h , как правило, зависит от некоторых параметров: варьируя их и решая каждый раз задачу 1.2, получаем различные точки Парето-фронта.

Популярным подходом является *метод взвешенной суммы*, в котором функционал

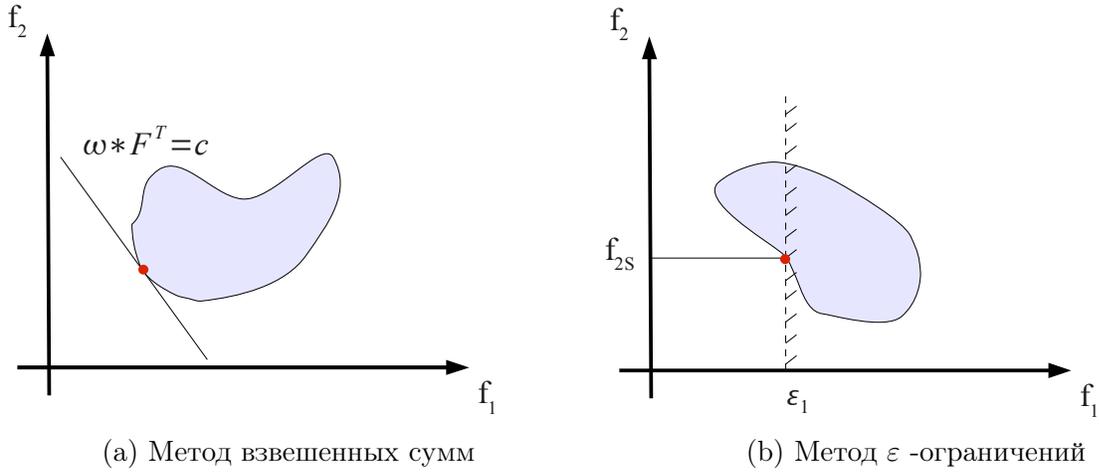


Рис. 2.1: Методы скаляризации

2.1 берется линейным:

$$f(\mathbf{x}) = \sum_{i=1}^k \omega_i \cdot f_i(\mathbf{x}), \quad (2.2)$$

$$\omega_i > 0, \quad i = 1, \dots, k,$$

$$\sum_{i=1}^k \omega_i = 1.$$

Геометрическая интерпретация в случае двух целевых функций представлена на Рис. 2.1a. Минимизация интерпретируется как поиск такого значения c , при котором линия $\omega \cdot \mathbf{f}^T = c$ касается границы множества допустимых значений целевых функций ($\mathbf{f}(\Omega)$). Этот метод имеет, однако, существенный недостаток, т.к. с помощью него не могут быть получены невыпуклые части Парето-фронта.

Этого недостатка лишена *функция скаляризации Чебышева*:

$$f(\mathbf{x}) = \max_i \omega_i |f_i(\mathbf{x}) - z_i|, \quad (2.3)$$

$$z_i < \min_{x \in \Omega} f_i(\mathbf{x}),$$

$$i = 1, \dots, k.$$

Однако $f(\mathbf{x})$ в данном случае является недифференцируемой, что затрудняет решение получившейся однокритериальной задачи с помощью градиентных методов.

Другим способом свести задачу 1.3 к 1.2 является *метод ε -ограничений*:

$$\begin{aligned} \min \quad & f_p(\mathbf{x}) \\ \text{при ограничениях} \quad & f_i(\mathbf{x}) \leq \varepsilon_i, \quad i = 1, \dots, k, \quad i \neq p \end{aligned} \quad (2.4)$$

Изменяя p и ε_i можно получать различные точки Парето-фронта. Иллюстрация к данному методу представлена на Рис. 2.1b.

Одним из недостатков рассмотренных методов является то, что для получения нескольких точек на Парето-фронте нужно несколько раз отдельно решить однокритериальную задачу, причем на каждой итерации этого решения вычисляются значения всех целевых функций. Значит, чтобы восстановить Парето-фронт с приемлемой точностью, потребуется большое количество итераций. Отдельной проблемой является подбор параметров $(\omega_i, \varepsilon_i)$, т.к. его трудно выполнить автоматически без каких-либо предположений относительно вида Парето-фронта.

Исчерпывающий обзор методов скаляризации дан в книге [3].

2.2 Эволюционные алгоритмы

Эволюционные алгоритмы, как следует из названия, основаны на идеях, изложенных Дарвиным в своем труде «Происхождение видов». Они используют понятия популяции, мутации, скрещивания, выживания сильнейшего. К эволюционным алгоритмам относятся *генетические алгоритмы*, часто использующиеся для решения оптимизационных задач.

Идея алгоритма состоит в том, что каждая точка пространства параметров является членом некоторой популяции P размера N . Начальная популяция выбирается случайно. Имеется функция приспособленности, которая показывает насколько данный член популяции хорошо адаптирован к «условиям среды». Из популяции P выбирается подмножество $P_{parents}$ наиболее приспособленных родителей. Из $P_{parents}$ затем, путем операций мутации и скрещивания получается новая популяция P . Процесс идет, пока не выполнится некоторый критерий остановки. В одномерном случае решением является «наиболее приспособленный» член последней популяции. В многомерном случае, как правило, заранее задается определенное количество популяций и их размер, а затем Парето-оптимальные решения выбираются из всех точек, вычисленных алгоритмом. Одним из самых популярных генетических алгоритмов, используемых для решения задач многокритериальной оптимизации, является NSGA2 [4].

Главным недостатком таких алгоритмов является то, что Парето-оптимальность полученных решений никак не гарантируется. К тому же, алгоритм не учитывает локаль-

ную информацию о функциях (например, производные), что приводит к большому количеству вычислений целевых функций «далеко» от Парето-фронта. Подробнее про эволюционные алгоритмы многокритериальной оптимизации можно прочитать, например, в [5].

2.3 Цель работы

Описанная здесь методика не использует веса для целевых функций, замену целевых функций ограничениями и другие способы скаляризации. Она основана на расширении алгоритмов одномерного поиска (метод наискорейшего спуска, метод Ньютона) на многокритериальный случай для нахождения одного оптимального решения и идеях метода продолжения [6] (решения системы нелинейных уравнений) для нахождения множества решений. В работе описаны теоретические основы алгоритма, а также некоторые практические аспекты его реализации.

Глава 3

Поиск Парето-оптимального решения

3.1 Теория

В дальнейшем будем предполагать все функции непрерывно дифференцируемыми.

3.1.1 Однокритериальная задача

Рассмотрим задачу однокритериальной оптимизации. Требуется найти минимум функции $f(\mathbf{x})$. Начиная с заданной точки \mathbf{x}_0 , будем генерировать последовательность точек $\mathbf{x}_{k+1}(\mathbf{x}_k)$ таких, что $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

Пусть $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{v}$, где \mathbf{v} — единичный вектор направления, а α — длина шага. Вычислим \mathbf{v} . Из теоремы Тейлора следует:

$$f(\mathbf{x}_k + \alpha \mathbf{v}) = f(\mathbf{x}_k) + \alpha \mathbf{v}^T \nabla f(\mathbf{x}_k) + o(|\mathbf{v}|). \quad (3.1)$$

То, насколько изменится f вдоль направления \mathbf{v} , зависит от произведения $\mathbf{v}^T \nabla f_k$. Получаем следующую задачу:

$$\min_p \mathbf{v}^T \nabla f_k, \quad \text{при условии } |\mathbf{v}| = 1. \quad (3.2)$$

Поскольку $\mathbf{v}^T \nabla f_k = |\mathbf{v}| |\nabla f_k| \cos \theta = |\nabla f_k| \cos \theta$, то минимум 3.2 достигается при $\theta = \pi$ и $\cos \theta = -1$. Таким образом, получаем:

$$\mathbf{v} = -\frac{\nabla f_k}{|\nabla f_k|} \quad (3.3)$$

— направление наискорейшего спуска.

Вообще, любое направление, отличающееся от $-\nabla f_k$ меньше, чем на $\pi/2$ является направлением спуска, т.е. приводит к уменьшению f при достаточно малых α . При

$\nabla f_k = 0$ такого направления не существует, \mathbf{x}_k является минимумом функции f .

Теорема 1. Если \mathbf{x}^* — локальный экстремум, то $\nabla f(\mathbf{x}^*) = 0$.

Это хорошо известное необходимое условие локального экстремума функции f . Теперь рассмотрим эти же вопросы для многокритериальных задач.

3.1.2 Многокритериальная задача с ограничениями

Определение 4. Ограничение $g_i(\mathbf{x})$ в точке \mathbf{x}^* называется активным, если $g_i(\mathbf{x}^*) = 0$. Множество всех активных ограничений $\{i \in 1, \dots, k \mid g_i(\mathbf{x}^*) = 0\}$ обозначим $\mathcal{A}(\mathbf{x}^*)$.

Сформулируем следующую теорему (необходимое условие Парето-оптимальности Каруша–Куна–Такера) [7]:

Теорема 2. Пусть выполнено условие: множество векторов $\{\nabla g_i(\mathbf{x}^*) \mid i \in \mathcal{A}(\mathbf{x}^*)\}$ линейно независимо. Если точка \mathbf{x}^* является (локально) Парето-оптимальной для задачи 1.1, то существуют вектора $\boldsymbol{\lambda} \in \mathbb{R}^k$ и $\boldsymbol{\mu} \in \mathbb{R}^m$ такие, что:

$$\begin{aligned} \sum_{i=1}^k \lambda_i \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^m \mu_j \nabla g_j(\mathbf{x}^*) &= \mathbf{0}, \\ \mu_j g_j(\mathbf{x}^*) &= 0, \quad j = 1, \dots, m, \\ \boldsymbol{\lambda} &\geq \mathbf{0}, \quad \boldsymbol{\lambda} \neq \mathbf{0}, \\ \boldsymbol{\mu} &\geq \mathbf{0}. \end{aligned} \tag{3.4}$$

В ряде случаев удобно использовать формулировку необходимого условия для задачи 1.3, которая зависит только от геометрии множества допустимых значений Ω , а не от конкретного ее алгебраического описания с помощью функции $\mathbf{g}(\mathbf{x})$. Для этого нужно ввести понятия касательного вектора и касательного конуса.

Определение 5. Вектор $\mathbf{v} \in \mathbb{R}^n$ называется касательным вектором к Ω в т. $\mathbf{x} \in \Omega$, если для всех векторных последовательностей $\{\mathbf{x}_i\} : \mathbf{x}_i \rightarrow \mathbf{x}, \mathbf{x}_i \in \Omega$ и для всех положительных скалярных последовательностей $t_i \rightarrow 0$ существует последовательность $\mathbf{v}_i \rightarrow \mathbf{v}$ такая, что $\mathbf{x}_i + t_i \mathbf{v}_i \in \Omega \forall i$.

Определение 6. Касательный конус $T_\Omega(\mathbf{x})$ — множество всех касательных векторов к Ω в т. \mathbf{x} .

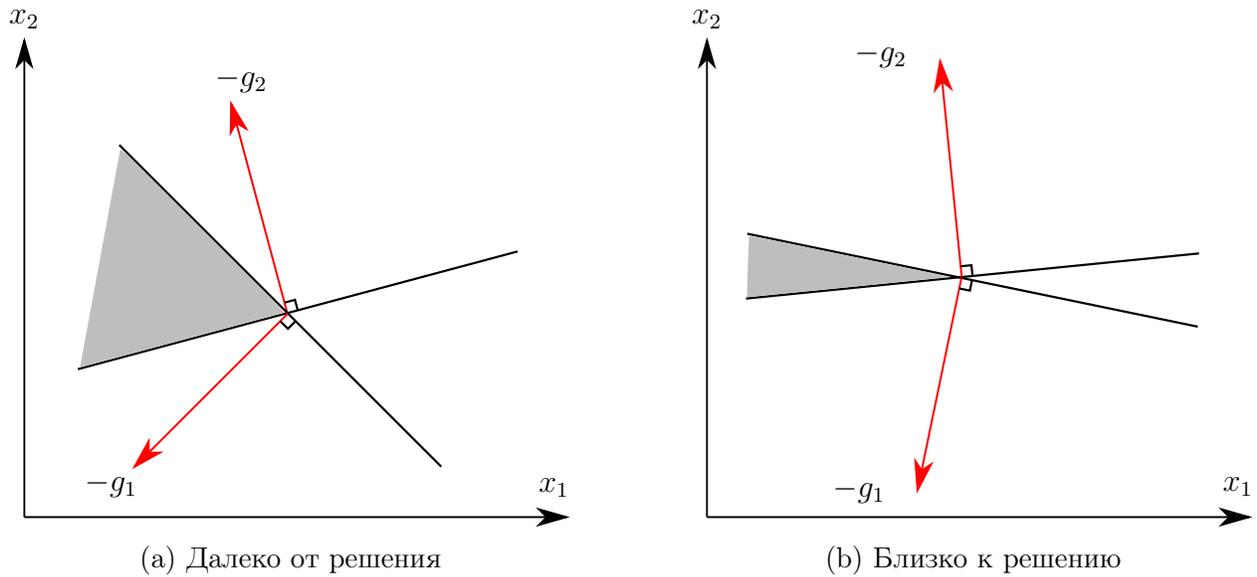


Рис. 3.1: Возможные направления спуска

Теорема 3. Если точка \mathbf{x}^* является (локально) Парето-оптимальной для задачи 1.3, то система:

$$\begin{aligned} (\nabla f_i(\mathbf{x}^*))^T \mathbf{v} < 0, \quad i = 1, \dots, k, \\ \mathbf{v} \in T_\Omega(\mathbf{x}) \end{aligned} \quad (3.5)$$

не имеет решений.

Определение 7. Точка, в которой выполнены необходимые условия (локальной) Парето-оптимальности (3.4 или 3.5) называется критической.

3.1.3 Многокритериальная задача без ограничений

Смысл приведенных выше теорем проще всего пояснить на задачах без ограничений.

При $m = 0$ теорема 2 утверждает, что в Парето-оптимальной точке нулевой вектор может быть записан в виде линейной комбинации градиентов компонент целевой функции с положительными коэффициентами.

Для задачи без ограничений теорема 3 переформулируется следующим образом.

$$Im(\nabla \mathbf{f}(\mathbf{x}^*)) \cap (-\mathbb{R}_{++})^k = \emptyset, \quad (3.6)$$

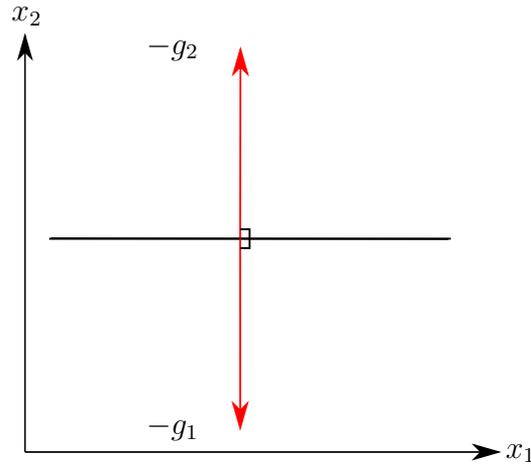


Рис. 3.2: Градиенты в Парето-оптимальной точке

где:

$Im(M)$ — образ линейного отображения M ,

$\nabla \mathbf{f}(\mathbf{x})$ — матрица Якоби функции $\mathbf{f}(\mathbf{x}) : (\nabla \mathbf{f}(\mathbf{x}))_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{x})$,

\mathbb{R}_{++} — множество положительных вещественных чисел.

Таким образом, если точка \mathbf{x}^* не является Парето-оптимальной, значит существует вектор $\mathbf{v} \in \mathbb{R}^n$ такой, что

$$\nabla \mathbf{f}(\mathbf{x}^*) \mathbf{v} \in (-\mathbb{R}_{++})^k \quad (3.7)$$

— *направление спуска*. Вдоль этого вектора функция $\mathbf{f}(\mathbf{x})$ будет уменьшаться по всем своим компонентам (при достаточно малом положительном шаге).

Проиллюстрируем приведенные выше формулировки для задачи с $n = k = 2$ и $m = 0$ (Рис. 3.1). Здесь g_1 и g_2 — градиенты функции, т.е. $\nabla \mathbf{f}(\mathbf{x}) = (g_1 \ g_2)^T$. Серым цветом обозначен конус возможных направлений спуска. Этот конус уменьшается при приближении к Парето-оптимальной точке. Наконец, в точке решения он «схлопывается», а градиенты функции становятся линейно зависимыми (Рис. 3.2).

3.2 Метод наискорейшего спуска

Возникает вопрос: какое из возможных направлений спуска выбрать? Очевидно, мы хотим максимально уменьшить целевую функцию по всем компонентам. Рассмотрим пока задачу без ограничений. Введем функцию $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ [8]:

$$\mathbf{h}_x(\mathbf{v}) := \max_i \{(\nabla \mathbf{f}(\mathbf{x}) \mathbf{v})_i \mid i = 1, \dots, k\}. \quad (3.8)$$

Рассмотрим следующую задачу:

$$\begin{aligned} \min \quad & \mathbf{h}_x(\mathbf{v}) \\ \text{при условии} \quad & |\mathbf{v}|_\infty \leq 1 \end{aligned} \tag{3.9}$$

Вместо L_∞ нормы можно использовать и другие, но ограничение в таком виде приводит к наиболее простой формулировке задачи. Переформулировав 3.9, получаем задачу линейной оптимизации, решив которую, можно найти оптимальное направление спуска:

$$\begin{aligned} \min \quad & t \\ \text{при ограничениях} \quad & (\nabla f_i(\mathbf{x}))^T \mathbf{v} \leq t, \quad i = 1, \dots, k, \\ & |\mathbf{v}|_\infty \leq 1 \end{aligned} \tag{3.10}$$

Можно доказать, что итеративный алгоритм $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{v}$, использующий данное направление спуска, при некоторых условиях, накладываемых на выбор длины шага α , сходится к критической точке [8].

При $k = 1$ (однокритериальная оптимизация) получаем $\mathbf{v} = -\nabla f_k / |\nabla f_k|$, т.е. данный алгоритм является расширением метода наискорейшего спуска на многокритериальный случай.

В однокритериальном случае алгоритм останавливается, если норма градиента достаточно мала. Но это условие остановки не подходит для многокритериальных задач, т.к. градиенты в них не обращаются в ноль в критической точке. Однако, в качестве критерия можно использовать величину t решения задачи 3.10, т.к. обращение ее в ноль означает, что алгоритм сошелся в критическую точку.

Теперь рассмотрим задачу с ограничениями. Воспользуемся следующей теоремой [2].

Теорема 4. *Рассмотрим множество*

$$C(\mathbf{x}^*) := \{\mathbf{v} \in \mathbb{R}^n \mid (\nabla g_i(\mathbf{x}^*))^T \mathbf{v} \leq 0, \quad i \in \mathcal{A}(\mathbf{x}^*)\}.$$

Если вектора $\{\nabla g_i(\mathbf{x}^) \mid i \in \mathcal{A}(\mathbf{x}^*)\}$ линейно независимы, то*

$$C(\mathbf{x}^*) = T_\Omega(\mathbf{x}^*).$$

Таким образом, из 3.5 получаем: для того, чтобы точка \mathbf{x}^* была Парето-оптималь-

ной, необходимо, чтобы следующая система не имела решений:

$$\begin{aligned} (\nabla f_i(\mathbf{x}^*))^T \mathbf{v} &< 0, & i = 1, \dots, k, \\ (\nabla g_j(\mathbf{x}^*))^T \mathbf{v} &\leq 0, & j \in \mathcal{A}(\mathbf{x}^*). \end{aligned} \quad (3.11)$$

Вместо задачи 3.10, соответственно, получаем [8]:

$$\begin{aligned} \min \quad & t \\ \text{при ограничениях} \quad & (\nabla f_i(\mathbf{x}))^T \mathbf{v} \leq t, & i = 1, \dots, k, \\ & (\nabla g_j(\mathbf{x}))^T \mathbf{v} \leq t, & j \text{ активных ограничений,} \\ & |\mathbf{v}|_\infty \leq 1. \end{aligned} \quad (3.12)$$

3.3 Метод Ньютона

Главный недостаток методов первого порядка (таких, как метод наискорейшего спуска) заключается в их линейной сходимости к оптимальной точке. Обратим также внимание на то, что решение задачи 3.12 не дает никакой оценки длины шага, т.к. $|\mathbf{v}|_\infty$ всегда равно 1. В этом разделе мы рассмотрим метод второго порядка (т.е. использующий вторые производные функций $\mathbf{f}(\mathbf{x})$ и $\mathbf{g}(\mathbf{x})$), сходящийся сверхлинейно или квадратично.

Начнем с однокритериального случая и теоремы Тейлора:

$$f(\mathbf{x}_k + \mathbf{v}) = f(\mathbf{x}_k) + \mathbf{v}^T \nabla f(\mathbf{x}_k) + \frac{1}{2} \mathbf{v}^T \nabla^2 f(\mathbf{x}_k) \mathbf{v} + o(|\mathbf{v}|^2). \quad (3.13)$$

Нужно найти такой \mathbf{v} , который минимизирует $m(\mathbf{v}) = \mathbf{v}^T \nabla f_k + \frac{1}{2} \mathbf{v}^T \nabla^2 f_k \mathbf{v}$. Если гессиан $\nabla^2 f_k$ положительно определен, то, приравнявая $m'(\mathbf{v})$ к нулю, получаем:

$$\mathbf{v} = -\nabla^2 f_k^{-1} \nabla f_k. \quad (3.14)$$

В многокритериальном случае для нахождения \mathbf{v} предлагается решать следующую задачу [9]:

$$\min_{\mathbf{v} \in \mathbb{R}^n} \max_{i=1, \dots, k} (\nabla f_i(\mathbf{x}))^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \nabla^2 f_i(\mathbf{x}) \mathbf{v}. \quad (3.15)$$

Если все гессианы положительно определены, то задача 3.15 всегда имеет единственное решение. При $k = 1$ решением является 3.14.

Таким образом, разность $\max_{i=1, \dots, k} f_i(\mathbf{x} + \mathbf{v}) - f_i(\mathbf{x})$ аппроксимируется максимумом локальных квадратичных моделей компонент целевой функции в т. \mathbf{x} . Задача 3.15 не

является гладкой, но она может быть переформулирована следующим образом:

$$\begin{aligned} \min \quad & t \\ \text{при условии} \quad & (\nabla f_i(\mathbf{x}))^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \nabla^2 f_i(\mathbf{x}) \mathbf{v} \leq t, \quad i = 1, \dots, k. \end{aligned} \quad (3.16)$$

В [9] доказывается, что алгоритм, использующий решение 3.16 в качестве направления спуска, при некоторых условиях сходится к локально Парето-оптимальной точке сверхлинейно, если $\mathbf{f}(\mathbf{x})$ дважды непрерывно дифференцируема и квадратично, если вторые производные $\mathbf{f}(\mathbf{x})$ непрерывны по Липшицу.

Для оптимизации с ограничениями получаем следующую задачу:

$$\begin{aligned} \min \quad & t \\ \text{при ограничениях} \quad & (\nabla f_i(\mathbf{x}))^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \nabla^2 f_i(\mathbf{x}) \mathbf{v} \leq t, \quad i = 1, \dots, k, \\ & (\nabla g_j(\mathbf{x}))^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \nabla^2 g_j(\mathbf{x}) \mathbf{v} \leq t, \quad j \text{ активных ограничений.} \end{aligned} \quad (3.17)$$

3.4 Выбор длины шага

Рассмотрим теперь вопрос о нахождении длины шага. Начнем с однокритериального случая.

Оптимальное значение длины шага α — это решение следующей одномерной задачи:

$$\min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{v}), \quad \alpha > 0. \quad (3.18)$$

Однако, точное решение задачи 3.18 требует большого количества вычислений целевой функции. На практике применяют другой подход: подбирают такие значения α , чтобы они удовлетворяли некоторым условиям. Вот пример таких условий (правила Вулфа [10]):

$$f(\mathbf{x}_k + \alpha \mathbf{v}) \leq f(\mathbf{x}_k) + c_1 \alpha \nabla f_k^T \mathbf{v}, \quad (3.19)$$

$$\nabla f(\mathbf{x}_k + \alpha \mathbf{v})^T \mathbf{v} \geq c_2 \nabla f_k^T \mathbf{v}, \quad (3.20)$$

где $0 < c_1 < c_2 < 1$.

Введем обозначение $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v})$. Первое условие (также известное как правило Армихо) говорит о том, что $\phi(\alpha)$ не должна превышать значения некоторой убывающей линейной функции, равной $f(\mathbf{x}_k)$ в нуле. Это условие удовлетворяется для всех достаточно малых α . Второе условие означает, что производная $\phi'(\alpha)$ по крайней мере в c_2 раз больше, чем $\phi'(0)$.

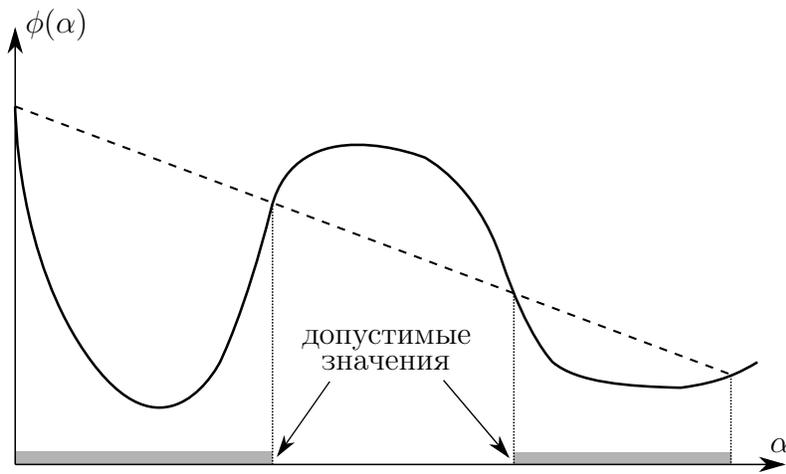


Рис. 3.3: Правило Армико

В многокритериальном случае условие 3.20 не работает, поскольку значения производных отдельных компонент ничего не говорят об оптимальности текущей точки. Правило Армико можно расширить на многокритериальный случай: неравенство 3.19 следует понимать покомпонентно. Процедура поиска α формулируется так: начиная с $\alpha = 1$, пока 3.19 не выполнится, полагаем $\alpha := \alpha/2$.

Теорема 5. Если $\nabla f(\mathbf{x})\mathbf{v} < 0$, то существует $\epsilon > 0$ (зависящий, вообще говоря, от \mathbf{x} , \mathbf{v} и c) такой, что

$$\mathbf{f}(\mathbf{x} + \alpha\mathbf{v}) < \mathbf{f}(\mathbf{x}) + c \cdot \alpha \nabla f(\mathbf{x})\mathbf{v}$$

для всех $\alpha \in (0, \epsilon)$.

Из этой теоремы следует конечность вышеприведенной процедуры. Сходимость многокритериальных метода наискорейшего спуска и метода Ньютона при использовании данного алгоритма для поиска шага доказывается в [8] и [9].

3.5 Практические аспекты

На практике якобиан и тем более гессиан задачи зачастую неизвестен. Для нахождения первых производных используются различные схемы численного дифференцирования, а для вторых производных — различные приближения гессиана.

Для того, чтобы задача 3.17 имела решение, достаточно положительной определенности хотя бы одного из гессианов. Для приближения гессиана активных ограничений используются формула SR-1 (симметричная ранга один), для аппроксимации гессиана целевых функций - формула BFGS (Бройдена–Флетчера–Гольдфарба–Шанно) [2]. Формула BFGS гарантирует положительную определенность гессиана, если первое приближение также было положительно определенным.

Находить ответ задачи 3.17 можно с помощью любого метода решения задач нелинейной оптимизации, например, SQP. Она может быть достаточно сложной, однако является «внутренней», т.е. при ее решении не производится вычислений целевой функции и ограничений. Предполагается, что в реальных задачах функции являются достаточно «тяжелыми», чтобы временем решения внутренних задач можно было пренебречь.

Глава 4

Нахождение Парето-фронта

В этой главе будет рассмотрен алгоритм, решающий глобальную формулировку задачи многокритериальной оптимизации (см. раздел 1.3). Основная идея, лежащая в основе алгоритма: максимально избежать вычислений целевых функций и ограничений вдали от Парето-фронта. Многие существующие алгоритмы (например, генетические), выполняют много вычислений далеко от оптимального множества, пытаясь «одновременно» двигать все множество точек в сторону Парето-фронта. При этом, во многих случаях Парето-фронт обладает определенными геометрическими свойствами, позволяющими найти его практически без вычислений функций за его пределами. Предлагаемый алгоритм находит несколько оптимальных точек, а затем выполняет некоторую процедуру распространения вдоль Парето-фронта.

Предполагается, что целевые функции и ограничения в задаче 1.1 гладкие, а Парето-фронт состоит из конечного множества отдельных компонент, каждая из которых описывается некоторой гладкой функцией.

4.1 Локальная геометрия множества Парето

Опишем локальную геометрию множества Парето для задачи без ограничений. Вернемся к задаче 3.10 поиска оптимального направления спуска. Условие $|\mathbf{v}|_\infty \leq 1$ — не единственный способ ограничить длину вектора \mathbf{v} (и не получить $-\infty$ в качестве решения). Также, это может быть сделано с помощью дополнительного слагаемого в целевой функции:

$$\begin{aligned} \min \quad & t + \frac{1}{2}|\mathbf{v}|_2^2 \\ \text{при ограничениях} \quad & (\nabla f_i(\mathbf{x}))^T \mathbf{v} \leq t, \quad i = 1, \dots, k. \end{aligned} \tag{4.1}$$

Таким образом, получаем выпуклую квадратичную задачу с линейными ограничениями. Ей можно сопоставить двойственную задачу, и ее решение будет совпадать с решением прямой задачи:

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2} |\nabla \mathbf{f}^T \boldsymbol{\lambda}|_2^2 \\ \text{при ограничениях} \quad & \sum_{i=1}^k \lambda_i = 1, \\ & \lambda_i \geq 0. \end{aligned} \tag{4.2}$$

Направление спуска является линейной комбинацией градиентов компонент целевых функций с коэффициентами из вектора оптимальных значений двойственных переменных:

$$\mathbf{v} = - \sum_{i=1}^k \lambda_i^* \nabla f_i = - \nabla \mathbf{f}^T \boldsymbol{\lambda}. \tag{4.3}$$

Сравнивая задачу 4.2 с условиями Каруша–Куна–Такера (теорема 2), сразу получаем, что $\mathbf{v} = 0$, если \mathbf{x} — критическая точка.

Рассмотрим гессиан задачи:

$$\frac{1}{2} |\nabla \mathbf{f}^T \boldsymbol{\lambda}|_2^2 = \frac{1}{2} \boldsymbol{\lambda}^T \nabla \mathbf{f} \nabla \mathbf{f}^T \boldsymbol{\lambda} = \frac{1}{2} \boldsymbol{\lambda}^T G \boldsymbol{\lambda},$$

где $G = \nabla \mathbf{f} \nabla \mathbf{f}^T$ — положительно полуопределенная симметричная матрица. Ее собственные вектора составляют ортонормированный базис. Т.к. $\nabla \mathbf{f}^T \boldsymbol{\lambda}^* = 0$, то $\boldsymbol{\lambda}^*$ является собственным вектором G с нулевым собственным числом.

Рассмотрим остальные собственные вектора и соответствующие им собственные числа: $\{\boldsymbol{\lambda}_e, \mu_e, G \boldsymbol{\lambda}_e = \mu_e \boldsymbol{\lambda}_e \mid e = 1, \dots, k-1\}$. Вектора

$$\mathbf{v}_e = \frac{1}{\sqrt{\mu_e}} \nabla f^T \boldsymbol{\lambda}_e \tag{4.4}$$

составляют ортонормированный базис в касательном Парето множеству пространстве:

$$\mathbf{v}_{e'}^T \mathbf{v}_{e''} = \frac{1}{\sqrt{\mu_{e'} \mu_{e''}}} \boldsymbol{\lambda}_{e'}^T \nabla \mathbf{f} \nabla \mathbf{f}^T \boldsymbol{\lambda}_{e''} = \frac{\mu_{e''}}{\sqrt{\mu_{e'} \mu_{e''}}} \boldsymbol{\lambda}_{e'}^T \boldsymbol{\lambda}_{e''} = \begin{cases} 1, & e' = e'' \\ 0, & e' \neq e'' \end{cases}.$$

В общем случае, точки, удовлетворяющие условиям ККТ, формируют некоторое многообразие размерности $k-1$ (строгое изложение этого утверждения, а также условий невырожденности множества Парето дано в статье [11]). В невырожденном случае, соответственно, $\mu_e > 0$ для всех $e = 1, \dots, k-1$.

Разберем случай задачи с ограничениями. Пусть $\mathcal{A}(\mathbf{x}^*)$ — множество активных ограничений и вектора $\{\nabla g_i(\mathbf{x}^*) \mid i \in \mathcal{A}(\mathbf{x}^*)\}$ линейно независимы. Ортогональное дополнение к линейной оболочке этих векторов является касательным пространством к поверхности активных ограничений. Пусть $\mathcal{P}_{\mathcal{A}}$ — ортогональный проектор на касательное пространство:

$$\begin{aligned}\mathcal{P}_{\mathcal{A}}^2 &= \mathcal{P}_{\mathcal{A}}, \\ \mathcal{P}_{\mathcal{A}} J^T &= 0.\end{aligned}$$

где J — матрица Якоби активных ограничений. Выразим $\mathcal{P}_{\mathcal{A}}$ через J :

$$\mathcal{P}_{\mathcal{A}} = I - J^T(JJ^T)^{-1}J. \quad (4.5)$$

Градиент функции f при ограничениях (приведенный градиент) выражается через $\mathcal{P}_{\mathcal{A}}$ с помощью формулы [12]:

$$\nabla f|_{\mathcal{A}} = \mathcal{P}_{\mathcal{A}} \nabla f.$$

Таким образом, для задачи с ограничениями вместо 4.4 получаем:

$$\mathbf{v}_e = \frac{1}{\sqrt{\mu_e}} \mathcal{P}_{\mathcal{A}} \nabla f^T \boldsymbol{\lambda}_e \quad (4.6)$$

— локальная геометрия множества Парето полностью определяется градиентами всех целевых функций и активных ограничений.

4.2 Рассеивание

Имея одну Парето оптимальную точку, можно выполнить некоторый малый шаг в пространстве параметров (перейти из точки \mathbf{x}^* в точку $\mathbf{x} = \mathbf{x}^* + \epsilon \mathbf{v}$) и затем воспользоваться методами, описанными в главе 3, для нахождения следующей оптимальной точки. Рассмотрим сначала вопрос выбора направления \mathbf{v} .

Касательное пространство является линейной аппроксимацией множества Парето в данной точке: при движении по касательной на (бесконечно малое) расстояние ϵ условия ККТ в точке \mathbf{x} будут выполняться с точностью $o(\epsilon)$. Эта ошибка, в свою очередь, влияет на количество итераций алгоритма, решающего локальную формулировку задачи, т.к. определяет «расстояние» от точки до Парето-множества. При этом, очевидно, если делать шаг перпендикулярно множеству Парето, то при последующем решении задачи в качестве ответа получится та же самая точка. Таким образом, эффективно делать шаг именно в касательном пространстве: алгоритм «уйдет» достаточно далеко

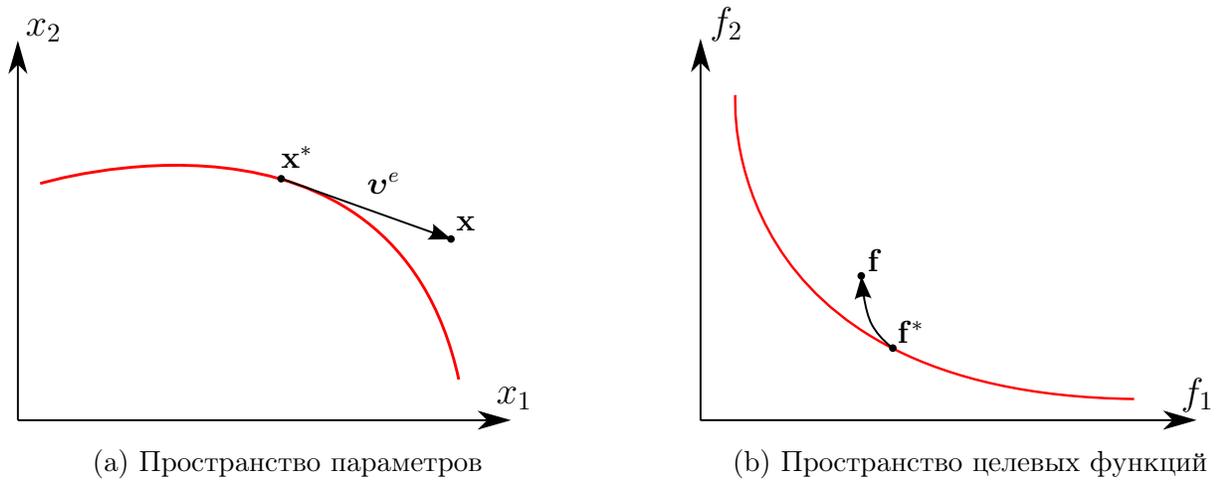


Рис. 4.1: Шаг вдоль множества Парето

от текущей точки, но не слишком далеко от множества оптимальных точек. Поэтому в качестве направлений для шага в пространстве параметров используются вектора ортонормированного базиса \mathbf{v}_e (4.4, 4.6). Такие шаги вдоль множества Парето в дальнейшем будем называть рассеиванием (см. Рис. 4.1).

Понятно, что на практике длина шага ϵ конечна. Она подбирается следующим образом: алгоритм имеет входной параметр *плотность фронта*, приблизительно соответствующий тому, сколько точек на Парето-фронте должно получиться в результате его выполнения. Этот параметр не задает точное количество Парето оптимальных точек, которые будут найдены алгоритмом, поскольку результат существенно зависит от геометрии фронта, которая заранее неизвестна. Зная также размер Парето-фронта вдоль каждой из осей в пространстве целевых функций, можно представить это пространство как объединение непересекающихся прямоугольных параллелепипедов одинакового размера. Цель алгоритма, таким образом, поместить по крайней мере одно оптимальное решение в каждый параллелепипед (конечно, какие-то из параллелепипедов могут вовсе не содержать оптимальных точек). Значит, подбирать длину шага ϵ нужно таким образом, чтобы точки (пространства целевых функций) $\mathbf{f}(\mathbf{x}^*)$ и $\mathbf{f}(\mathbf{x}^* + \epsilon \mathbf{v}_e)$ оказались в соседних параллелепипедах.

Как оценить размер Парето-фронта? Делается это с помощью нахождения *якорей* — минимумов отдельных компонент целевых функций безотносительно значения других компонент:

$$\mathbf{f}_a^i = \mathbf{f}(\mathbf{x}_a^i), \quad \mathbf{x}_a^i = \operatorname{argmin}_{\mathbf{x} \in \Omega} f_i(\mathbf{x}) \quad i = 1, \dots, k. \quad (4.7)$$

Решив соответствующие задачи, с помощью значений якорей можно найти две точки:

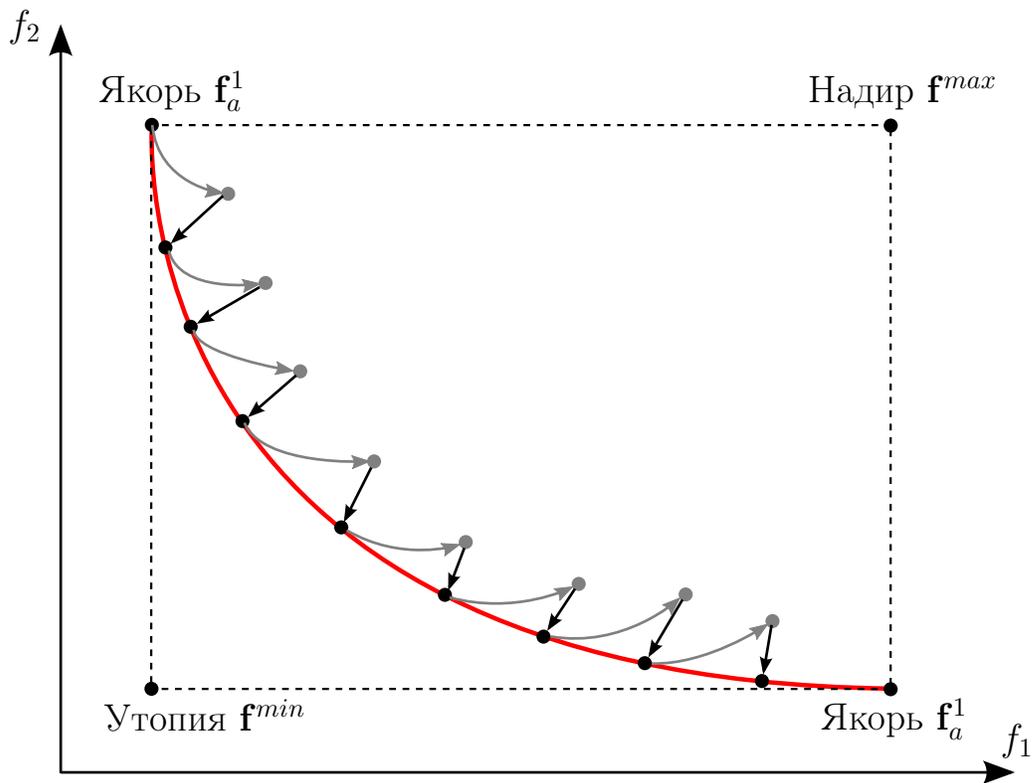


Рис. 4.2: Нахождение Парето-фронта

\mathbf{f}_{min} (утопия) и \mathbf{f}_{max} (надир):

$$\mathbf{f}^{min} = \{\min_i(f_a^i) \quad j = 1, \dots, k\}, \quad (4.8)$$

$$\mathbf{f}^{max} = \{\max_i(f_a^i) \quad j = 1, \dots, k\}. \quad (4.9)$$

Эти точки определяют границы всего Парето-фронта. Длины сторон прямоугольных параллелепипедов, на которые разбивается пространство целевых функций, находятся следующим образом:

$$bb_i = \frac{(\mathbf{f}_i^{max} - \mathbf{f}_i^{min})}{\text{плотность фронта}}.$$

4.3 Описание алгоритма

Алгоритм хранит два множества точек:

Неоптимальное множество N

Точки, полученные в результате рассеивания из оптимальных решений (в общем случае, не являются оптимальными по Парето).

Оптимальное множество P

Найденные оптимальные решения.

Шаги алгоритма:

1. Находим якоря $\{\mathbf{f}_a^i, i = 1, \dots, k\}$, добавляем их в оба множества.
2. Если $N \neq \emptyset$, то берем точку из N в качестве начальной для многокритериальных метода наискорейшего спуска или метода Ньютона (см. главу 3) и находим с помощью этих алгоритмов оптимальную точку \mathbf{x}^* , иначе **переходим к п. 6**,
3. Если \mathbf{x}^* оказывается в одном параллелепипеде с одной из точек из множества P , то **переходим к п. 2**.
4. Добавляем \mathbf{x}^* в P и находим базис касательного пространства (вектора \mathbf{v}_e) к Парето множеству в данной точке.
5. Выполняем рассеивание из \mathbf{x}^* .
 - (а) Берем касательное направление из множества $\{\pm \mathbf{v}_e, e = 1, \dots, k - 1\}$ и движемся вдоль него, пока не выйдем за пределы параллелепипеда с центром в текущей оптимальной точке (в пространстве целевых функций).
 - (б) Если получившаяся точка \mathbf{x}_{trial} находится в одном параллелепипеде с одной из точек из P или N , **переходим к п. (а)**.
 - (с) Добавляем \mathbf{x}_{trial} в N и **переходим к п. 2**.
6. Возвращаем множество P в качестве результата.

Работа данного алгоритма проиллюстрирована на Рис. 4.2.

Глава 5

Практические результаты

В этой главе будет проведено сравнение работы изложенного выше алгоритма с другими подходами к решению задач многокритериальной оптимизации (см. главу 2).

5.1 Методика сравнения

Всего в сравнении участвуют 3 алгоритма:

GTOpt — реализация описанного градиентного метода, основанного на локальной геометрии множества Парето,

NSGA2 — генетический алгоритм многокритериальной оптимизации [4],

AS — алгоритм адаптивной скаляризации.

Проблема сопоставления результатов работы двух алгоритмов в том, что Парето-фронты, вообще говоря, несравнимы между собой. Рассмотрим два множества точек (в пространстве целевых функций), недоминирующих друг друга, полученных в результате работы двух алгоритмов на одной задаче: P' и P'' . Часть точек из P' может доминировать какие-то точки из P'' и наоборот, так что нельзя однозначно сказать, какой из результатов лучше.

Для сравнения результатов введем метрику Q , основанную на расстоянии Хаусдорфа между множествами. В общем случае, если A и B являются двумя подмножествами некоторого метрического пространства с метрикой ρ , расстояние Хаусдорфа между A и B определяется по формуле

$$d(A, B) = \max(\sup_{x \in A} \rho(x, B), \sup_{y \in B} \rho(y, A)), \quad (5.1)$$

где $\rho(x, B) = \inf_{y \in B} \rho(x, y)$. Пусть ρ — евклидова метрика, A — множество, доминируемое истинным Парето-фронтом, B — множество, доминируемое приближительным

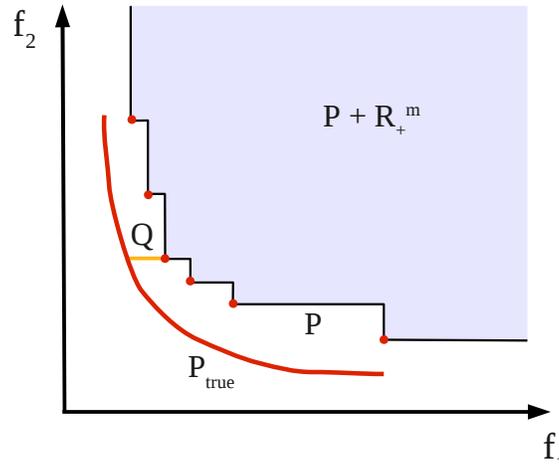


Рис. 5.1: Метрика Q

Парето-фронтом (результатом работы алгоритма). Определим [13]

$$Q(P) = d(P_{\text{true}} + \mathbb{R}_+^n, P + \mathbb{R}_+^n) = \sup_{\mathbf{p} \in P_{\text{true}}} \rho(\mathbf{p}, P + \mathbb{R}_+^n). \quad (5.2)$$

Чем приблизительный Парето-фронт «ближе» к истинному Парето-фронту, тем значение Q меньше (см. Рис. 5.1).

Лучший алгоритм должен получить хорошее приближение к истинному Парето-фронту за как можно меньшее количество итераций (т.е. сравнение оптимизаторов — само по себе многокритериальная задача). Корректным будет сравнение Q при фиксированном количестве итераций или, наоборот, сравнение количества итераций, необходимое для нахождения Парето-фронта с заданной точностью.

Поведение оптимизаторов регулируется опциями: плотностью фронта для алгоритма GTOpt, размером и количеством популяций для NSGA2 и количеством подзадач для адаптивной скаляризации. Все 4 опции являются дискретными величинами. Увеличение значения опции в каждом случае приводит к росту числа итераций алгоритма (что предполагает улучшение его результата).

5.2 Задачи

5.2.1 ZDT1

Формулировка [14]

$$\min_{0 < \mathbf{x} < 1} \begin{cases} x_1 \\ g(\mathbf{x})(1 - \sqrt{\frac{x_1}{g(\mathbf{x})}}) \end{cases}, \quad \text{где } g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

Парето-фронт

$$f_2 = 1 - \sqrt{f_1}$$

Результаты

Алгоритм	Количество итераций	Q
NSGA2	520	0.023
AS	645	0.024
GTOpt	154	0.023

Таблица 5.1: ZDT1, n = 2

Алгоритм	Количество итераций	Q
NSGA2	2368	0.022
AS	3765	0.018
GTOpt	488	0.023

Таблица 5.2: ZDT1, n = 10

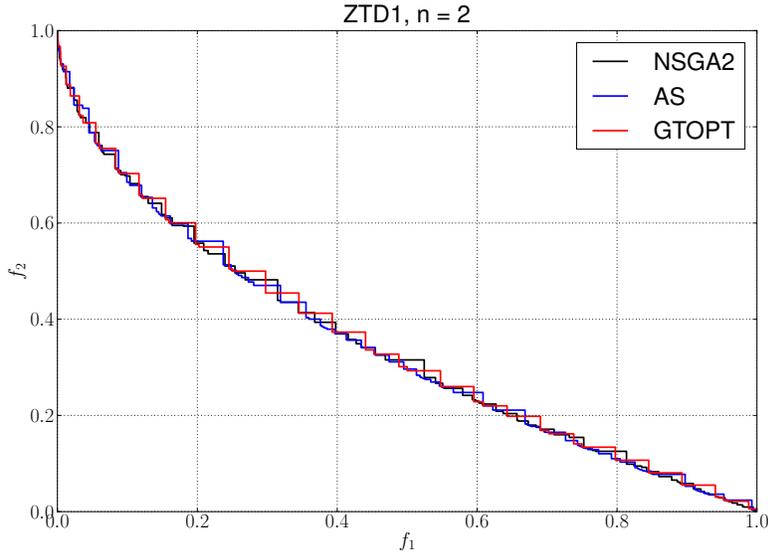


Рис. 5.2: Парето-фронты для задачи ZDT1 с $n = 2$

Здесь мы фиксировали Q и подбирали значения опций таким образом, чтобы получить Парето-фронт с заданным Q . Небольшие различия в метрике обусловлены дискретностью опций. Видно, что алгоритму GTOpt требуется в несколько раз меньше вычислений целевых функций для достижения заданной точности приближения истинного Парето-фронта. Причем, преимущество алгоритма увеличивается с ростом размерности пространства параметров.

5.2.2 ZDT2

Формулировка [14]

$$\min_{0 < \mathbf{x} < 1} \begin{cases} x_1 \\ g(\mathbf{x}) \left(1 - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})} \right)^2 \right) \end{cases}, \quad \text{где} \quad g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

Парето-фронт

$$f_2 = 1 - f_1^2$$

Результаты

Алгоритм	Количество итераций	Q
NSGA2	280	0.13
AS	270	0.01
GTOpt	267	0.001

Таблица 5.3: ZDT2, $n = 2$

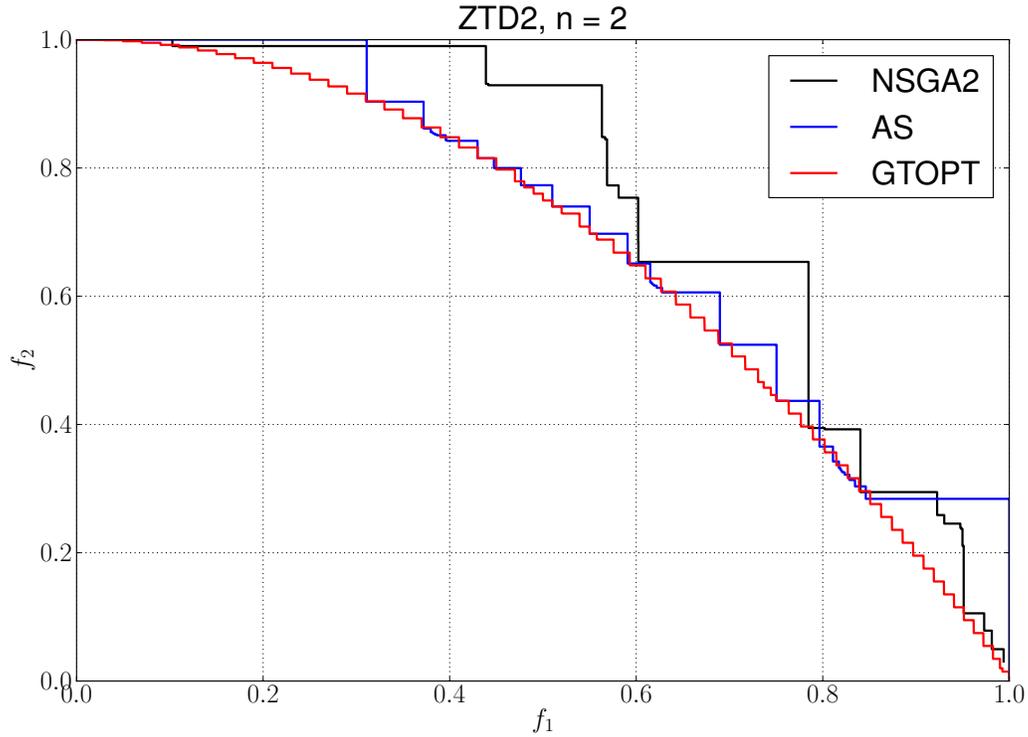


Рис. 5.3: Парето-фронты для задачи ZDT2 с $n = 2$

Особенность этой задачи заключается в невыпуклом Парето-фронте. Здесь сравнивались решения разными алгоритмами при фиксированном количестве итераций. Из значения Q и Рис. 5.3 видно, что результат GTOpt значительно лучше.

5.2.3 TP7

Формулировка

$$\min_{x_1, x_2} \begin{cases} x_1 \\ (x_2 + 1)/x_1 \end{cases} \quad \text{при ограничениях} \quad \begin{array}{ll} 0.1 < x_1 < 1 & x_2 + 9x_1 - 6 > 0 \\ 0 < x_2 < 0.5 & -x_2 + 9x_1 - 1 > 0 \end{array}$$

Результаты

В этой задаче Парето-фронт определяется не только целевыми функциями, но и ограничениями. Из Рис. 5.5 видно, что генетический алгоритм плохо восстанавливает ту часть Парето-фронта, которая определяется активными ограничениями, даже при большом количестве итераций (ср. с Рис. 5.4).

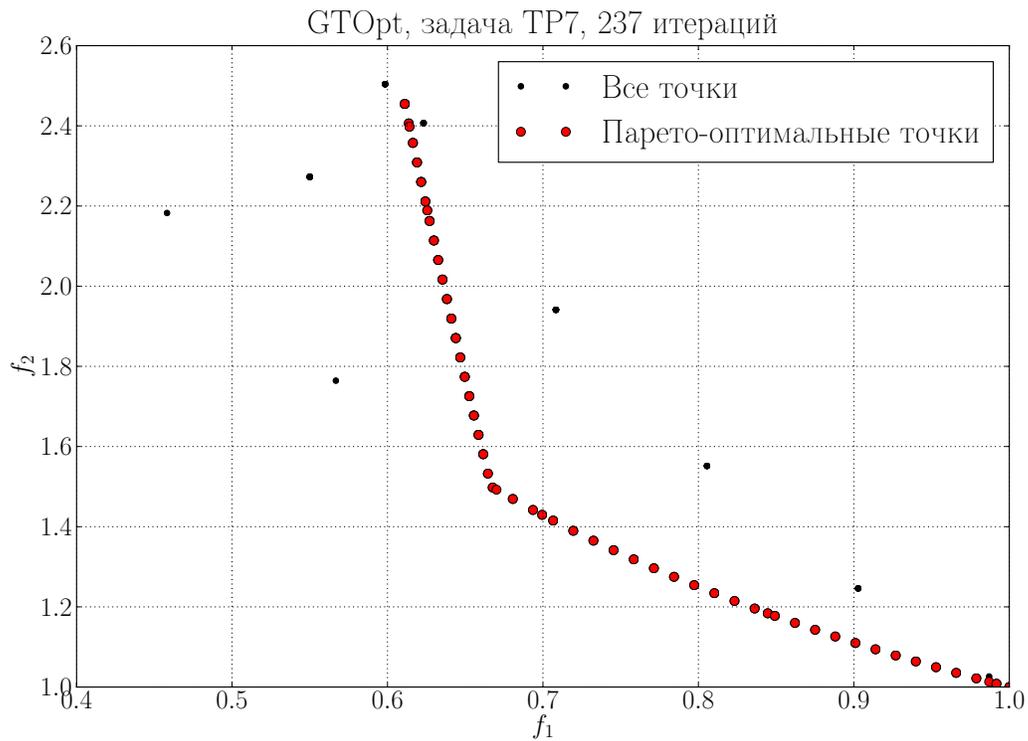


Рис. 5.4: Решение задачи TP7 алгоритмом GTOpt

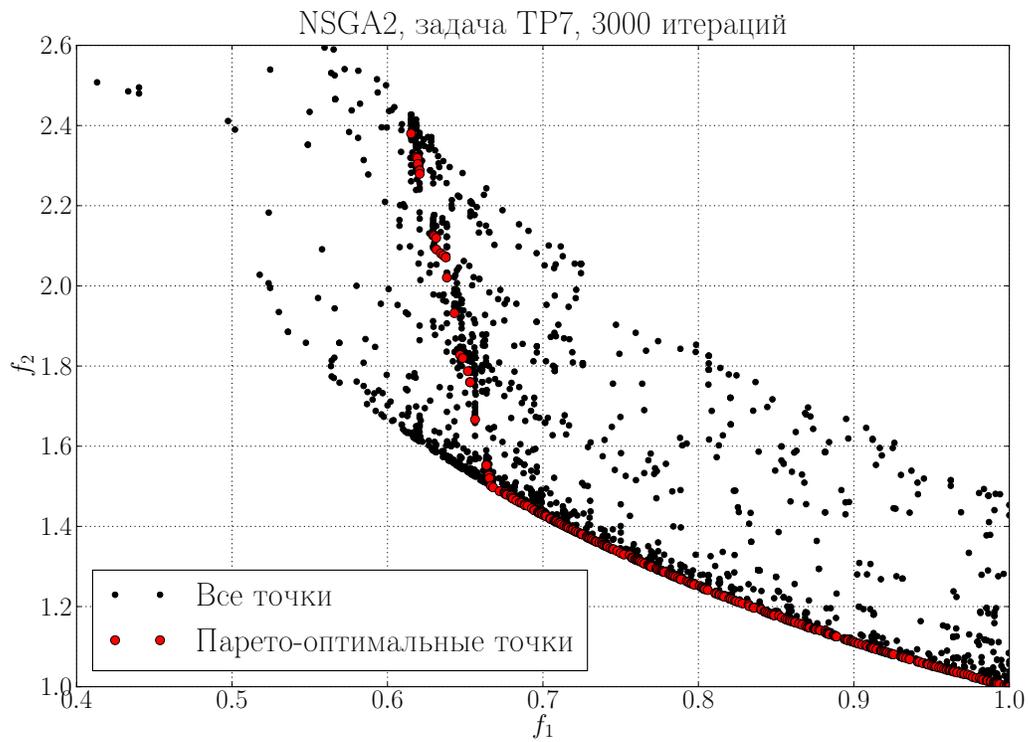


Рис. 5.5: Решение задачи TP7 алгоритмом NSGA2

Глава 6

Заключение

Описанный алгоритм успешно применяется на практике, в том числе для решения реальных инженерных задач. Область его применения может быть расширена с помощью различных методик. Например, специального вида численное дифференцирование с изменяющейся длиной шага позволяет оптимизировать функции с шумом. Используя алгоритмы глобального поиска для нахождения якорей, можно оптимизировать мультимодальные функции.

Одно из планируемых в дальнейшем улучшений алгоритма: усовершенствование методики выбора длины шага (см. раздел 3.4). На данный момент сформулированы правила Вулфа для многокритериального случая, и доказана сходимость метода наискорейшего спуска при их использовании для нахождения длины шага. Требуется проверка эффективности данного метода на практике.

Планируется также работа над обоснованием сходимости расширения ньютоновских и квазиньютоновских методов на многокритериальный случай в задаче с ограничениями (в этом плане интересна работа [15]).

Литература

- [1] Kaisa Miettinen, Francisco Ruiz, and Andrzej P Wierzbicki: *Introduction to multiobjective optimization: interactive approaches*. In *Multiobjective Optimization*, pages 27–57. Springer, 2008.
- [2] Jorge Nocedal and Stephen J Wright: *Numerical optimization*. Springer Science+ Business Media, 2006.
- [3] Gabriele Eichfelder: *Adaptive scalarization methods in multiobjective optimization*. Springer, 2008.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan: *A fast and elitist multiobjective genetic algorithm: Nsga-ii*. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [5] Kalyanmoy Deb: *Introduction to evolutionary multiobjective optimization*. In *Multiobjective Optimization*, pages 59–96. Springer, 2008.
- [6] Eugene L Allgower and Kurt Georg: *Numerical continuation methods*, volume 13. Springer-Verlag Berlin, 1990.
- [7] Kaisa Miettinen: *Nonlinear multiobjective optimization*, volume 12. Springer, 1999.
- [8] Jörg Fliege and Benar Fux Svaiter: *Steepest descent methods for multicriteria optimization*. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
- [9] Joerg Fliege, LM Grana Drummond, and Benar F Svaiter: *Newton’s method for multiobjective optimization*. *SIAM Journal on Optimization*, 20(2):602–626, 2009.
- [10] Philip Wolfe: *Convergence conditions for ascent methods*. *Siam Review*, 11(2):226–235, 1969.
- [11] Claus Hillermeier: *Generalized homotopy approach to multiobjective optimization*. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.

- [12] SV Utyuzhnikov, J Maginot, and MD Guenov: *Local approximation of pareto surface*. In *Proceedings of the World Congress on Engineering*, volume 2, pages 898–903. Citeseer, 2007.
- [13] Н.О. Пересторонин и Д.А. Яроцкий: *Тестирование алгоритмов многокритериальной оптимизации*. В *Труды 53-й научной конференции МФТИ*, страницы 122–123, Москва-Долгопрудный, 2010.
- [14] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele: *Comparison of multiobjective evolutionary algorithms: Empirical results*. *Evolutionary computation*, 8(2):173–195, 2000.
- [15] Joydeep Dutta: *Strong kkt, second order conditions and non-solid cones in vector optimization*. In *Recent Developments in Vector Optimization*, pages 127–167. Springer, 2012.