



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**"МИРЭА - Российский технологический университет"**  
**РТУ МИРЭА**

---

**Институт Информационных Технологий**  
**Кафедра Вычислительной Техники**

**ПРАКТИЧЕСКАЯ РАБОТА №1**

**по дисциплине**  
**«Теория принятия решений»**  
**Метод Парето**

Студент группы: ИКБО-04-22

Кликушин В.И.

(Ф. И.О. студента)

Преподаватель

Железняк Л.М.

(Ф.И.О. преподавателя)

Москва 2024

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 МЕТОД ПАРЕТО .....	4
1.1 Выбор Парето-оптимального множества .....	4
1.2 Указание верхних/нижних границ критериев. ....	7
1.3 Субоптимизация .....	8
1.4 Лексикографическая оптимизация .....	8
1.5 Результаты работы программы .....	9
ЗАКЛЮЧЕНИЕ .....	11
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ .....	12
ПРИЛОЖЕНИЯ .....	13

# ВВЕДЕНИЕ

Цель работы: познакомиться с методом Парето и применить его для нахождения оптимальной альтернативы в заданной предметной области.

Предметная область: выбор оптимального высшего учебного заведения.

Метод Парето применяется в задачах многокритериальной оптимизации, т.е. если альтернативы нужно сравнивать по двум и более критериям. Например, к таким задачам относится задача выбора оптимальной квартиры, где критериями выступают цена, удалённость от метро, площадь и т.д.

Суть метода заключается в простом сравнении альтернатив друг с другом по критериям.

Для сужения получаемого множества оптимальных альтернатив существуют метод указания верхних/нижних границ; субоптимизация и лексикографическая оптимизация.)

# 1 МЕТОД ПАРЕТО

Суть метода Парето заключается в прямом сравнении альтернатив между собой. Сравнение производится по критериям, причём по каким-то критериям требуется максимизация, а по каким-то – минимизация. Если одна альтернатива лучше другой по всем критериям, то она называется доминирующей, а другая альтернатива называется доминируемой. Если есть критерии, по которым одна альтернатива лучше другой, и есть критерии, по которой она хуже, то эти альтернативы являются несравнимыми.

В Парето-оптимальное множество входят только те альтернативы, которые не хуже всех других альтернатив, т.е. которые не являются доминируемыми по сравнению с любой другой альтернативой.

## 1.1 Выбор Парето-оптимального множества

Задача: выбрать оптимальное высшее учебное заведение для поступления на направление «программная инженерия». Рассмотренные критерии выбора:

- Проходной балл на бюджет, взятый за 2023 год
- Количество бюджетных мест, взятое за 2023 год
- Минимальная стоимость обучения за год на 2023 год
- Размер государственной академической стипендии для студентов, сдавших сессию на «хорошо» и «отлично» на 2023 год
- Национальный рейтинг университета на основе информации с сайта [academia.interfax.ru](http://academia.interfax.ru)
- Расстояние до общежития (километры)

В таблице 1.1.1 приведены десять альтернатив и шесть критериев для выбора оптимального технического высшего учебного заведения.

Таблица 1.1.1 – Альтернативы и критерии

№	Варианты решений	Критерии					
		Проходной балл (+)	Количество бюджетных мест (-)	Стоимость обучения (руб.) (+)	Размер стипендии (руб.) (+)	Рейтинг университета (баллы) (+)	Расстояние до общежития (км) (-)
1	РТУ МИРЭА	276	285	320000	1800	843	13,7
2	МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4,2
3	ВШЭ	295	135	700000	1994	878	7,9
4	МАИ	247	25	277770	1765	761	12,4
5	ИТМО	304	45	349000	2252	838	3,4
6	СПбГУ	276	35	335300	1843	915	7,3
7	МТУСИ	263	60	290000	1799	705	8,4
8	СГУ им. Чернышевского	192	30	42960	1454	847	12,3
9	НИЯУ МИФИ	300	37	287000	1875	975	15
10	МФТИ	290	30	432000	1900	965	4,8

Знаком (-) указывается отрицательное стремление критерия (чем меньше, тем лучше), а знаком (+) – положительное (чем больше, тем лучше). В таблице 1.1.2 приведен результат попарного сравнения альтернатив.

Таблица 1.1.2 – Сравнения альтернатив

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	x	x	x	x	x	x	x	x	x	x
A2	A2	x	x	x	x	x	x	x	x	x
A3	A3	н	x	x	x	x	x	x	x	x
A4	н	н	н	x	x	x	x	x	x	x
A5	н	н	н	н	x	x	x	x	x	x
A6	A6	н	н	н	н	x	x	x	x	x
A7	н	н	н	н	A5	A6	x	x	x	x
A8	н	н	н	н	н	н	н	x	x	x
A9	н	н	н	н	н	н	н	н	x	x
A10	A10	н	н	н	н	A10	A10	A10	н	x

Парето-оптимальное множество определено альтернативами {2, 3, 4, 5, 9, 10} (МГТУ имени Н.Э. Баумана, ВШЭ, МАИ, ИТМО, НИЯУ МИФИ, МФТИ).

Парето-оптимальное множество представлено в таблице 1.1.3.

Таблица 1.1.3 – Парето-оптимальное множество

№	Варианты решений	Критерии					
		Проходной балл (+)	Количество бюджетных мест (-)	Стоимость обучения (руб.) (+)	Размер стипендии (руб.) (+)	Рейтинг университета (баллы) (+)	Расстояние до общежития (км) (-)
2	МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4,2
3	ВШЭ	295	135	700000	1994	878	7,9
4	МАИ	247	25	277770	1765	761	12,4
5	ИТМО	304	45	349000	2252	838	3,4
9	НИЯУ МИФИ	300	37	287000	1875	975	15

Продолжение таблицы 1.1.3

10	МФТИ	290	30	432000	1900	965	4,8
----	------	-----	----	--------	------	-----	-----

Очевидно, что выделение множества Парето часто не является удовлетворительным решением. Это связано с тем, что при достаточно большом исходном множестве вариантов множество Парето оказывается недопустимо большим для того, чтобы ЛПР было бы в состоянии осуществить выбор самостоятельно. Таким образом, выделение множества Парето можно рассматривать лишь как предварительный этап оптимизации, и налицо проблема дальнейшего сокращения этого множества.

## 1.2 Указание верхних/нижних границ критериев.

Установим верхнюю и нижнюю границы.

Нижняя граница: проходной балл должен быть не ниже 270 и рейтинг университета должен быть не ниже 840 баллов.

Верхняя граница: расстояние до общежития должно быть менее 14 км

В таблице 1.2.1 приведено множество альтернатив после отсечения по верхним и нижним границам:

Таблица 1.2.1 – Парето-оптимальное множество

№	Варианты решений	Критерии					
		Проходной балл (+)	Количество бюджетных мест (-)	Стоимость обучения (руб.) (+)	Размер стипендии (руб.) (+)	Рейтинг университета (баллы) (+)	Расстояние до общежития (км) (-)
1	РТУ МИРЭА	276	285	320000	1800	843	13,7
2	МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4,2
3	ВШЭ	295	135	700000	1994	878	7,9
6	СПбГУ	276	35	335300	1843	915	7,3
10	МФТИ	290	30	432000	1900	965	4,8

Заметим, что альтернативы А2, А3, А6, А10 доминируют над А1 в то время, как А10 доминирует над А6. Следовательно, Парето-оптимальное множество

состоит из альтернатив {2, 3, 10}. Единственную альтернативу выбрать не удалось в силу не слишком «жестких границ» для критериев.

### 1.3 Субоптимизация

Главный критерий: рейтинг университета.

Нижняя граница: проходной балл должен быть не ниже 290 и рейтинг университета должен быть не ниже 840 баллов.

Верхняя граница: расстояние до общежития должно быть менее 14 км

Отбросим варианты, которые не удовлетворяют данным ограничениям и составим таблицу 1.3.1 результата субоптимизации.

Таблица 1.3.1 – Результат субоптимизации

№	Варианты решений	Критерии					
		Проходной балл (+)	Количество бюджетных мест (-)	Стоимость обучения (руб.) (+)	Размер стипендии (руб.) (+)	Рейтинг университета (баллы) (+)	Расстояние до общежития (км) (-)
2	МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4,2
3	ВШЭ	295	135	700000	1994	878	7,9
10	МФТИ	290	30	432000	1900	965	4,8

Учитывая главный критерий, оптимальным решением является МФТИ, т.к. у этого университета наибольший рейтинг. Заметим, что окончательное решение имеет субъективный характер.

### 1.4 Лексикографическая оптимизация

Лексикографическая оптимизация основана на упорядочении критериев по их относительной важности. На первом шаге отбирают исходы, которые имеют максимальную оценку по важнейшему критерию. Если такой исход



единственный, то его и считают оптимальным. Если же таких исходов несколько, то среди них отбирают те, которые имеют максимальную оценку по следующему за важнейшим критерию.

Установим следующий приоритет критериев: рейтинг университета, проходной балл, стоимость обучения, количество бюджетных мест, расстояние до общежития, размер стипендии.

При сравнении альтернатив лишь по первому критерию остается всего одно решение, которое является наилучшим. Вынесем оптимальное решение в таблицу 1.4.1.

Таблица 1.4.1 – Лексикографическая оптимизация

№	Варианты решений	Критерии					
		Проходной балл (+)	Количество бюджетных мест (-)	Стоимость обучения (руб.) (+)	Размер стипендии (руб.) (+)	Рейтинг университета (баллы) (+)	Расстояние до общежития (км) (-)
9	НИЯУ МИФИ	300	37	287000	1875	975	15

Лексикографическая оптимизация позволяет выделить единственное наилучшее решение.

## 1.5 Результаты работы программы

Оптимальное множество Парето:							
Альтернатива	Проходной балл (+)	Кол-во бюджетных мест (-)	Стоимость обучения (+)	Размер стипендии (+)	Рейтинг университета (+)	Расстояние до общежития (-)	
0   МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4.2	
1   ВШЭ	295	135	700000	1994	878	7.9	
2   МАИ	247	25	277770	1765	761	12.4	
3   ИТМО	304	45	349000	2252	838	3.4	
4   НИЯУ МИФИ	300	37	287000	1875	975	15	
5   МФТИ	290	30	432000	1900	965	4.8	

Рисунок 1.5.1 – Оптимальное множество Парето

Установка верхних и нижних границ:							
Альтернатива	Проходной балл (+)	Кол-во бюджетных мест (-)	Стоимость обучения (+)	Размер стипендии (+)	Рейтинг университета (+)	Расстояние до общежития (-)	
0   МГТУ имени Н.Э. Баумана	291	84	363040	1854	964	4.2	
1   ВШЭ	295	135	700000	1994	878	7.9	
2   МФТИ	290	30	432000	1900	965	4.8	

Рисунок 1.5.2 – Установка верхних и нижних границ

Субоптимизация:							
Альтернатива	Проходной балл (+)	Кол-во бюджетных мест (-)	Стоимость обучения (+)	Размер стипендии (+)	Рейтинг университета (+)	Расстояние до общежития (-)	
0   МФТИ	290	30	432000	1900	965	4.8	

Рисунок 1.5.3 – Субоптимизация

Лексикографическая оптимизация:							
Альтернатива	Проходной балл (+)	Кол-во бюджетных мест (-)	Стоимость обучения (+)	Размер стипендии (+)	Рейтинг университета (+)	Расстояние до общежития (-)	
0   НИЯУ МИФИ	300	37	287000	1875	975	15	

Рисунок 1.5.4 – Лексикографическая оптимизация

```

print("Исходная таблица с альтернативами и критериями:".center(201))
print_table(data)

print("Оптимальное-множество Парето:".center(201))
print_table(create_Pareto_set(data))

print("Установка верхних и нижних границ:".center(201))
branches = [{"Проходной балл (+)": 270}, {"Рейтинг университета (+)": 840},
            {"Расстояние до общежития (-)": 14}]
print_table(branches_and_boundaries(data, branches))

print("Субоптимизация:".center(201))
branches = [{"Проходной балл (+)": 290}, {"Расстояние до общежития (-)": 14}]
main_criteria = "Рейтинг университета (+)"
print_table(suboptimization(data, branches, main_criteria))

print("Лексикографическая оптимизация:".center(201))
priority = ("Рейтинг университета (+)", "Проходной балл (+)", "Стоимость обучения (+)",
            "Кол-во бюджетных мест (-)", "Расстояние до общежития (-)",
            "Размер стипендии (+)")
print_table(lexical_optimization(data, priority))

```

**Рисунок 1.5.5 – Границы для оптимизации и приоритеты критериев**

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной практической работы мной был изучен метод Парето, применён на практике для определения оптимального решения в задаче нахождения оптимального высшего учебного заведения, а также сделана программная реализация. Это было проделано также и для методов сужения.

Основным плюсом метода Парето является простота реализации, однако в результате может получиться несколько оптимальных альтернатив, из-за чего ЛПР придётся самостоятельно выбирать одно из них. Методы сужения помогают решить эту проблему, однако выбор характера сужения носит субъективный характер.

## СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

## **ПРИЛОЖЕНИЯ**

Приложение А – Код реализации метода Парето на языке Python.

## Приложение А

### Код реализации метода Парето на языке Python.

#### *Листинг А.1. Реализация Парето.*

```
import csv
import pandas as pd

def print_table(data):
    '''Функция для вывода таблицы'''
    print(pd.DataFrame(data).to_markdown())

def compare_alternatives(a, b):
    '''Функция для сравнения альтернатив по отношению Парето-доминирования'''
    counter = 0
    for key in a:
        if '+' in key:
            counter += (float(a[key]) > float(b[key]))
        elif '-' in key:
            counter += (float(a[key]) < float(b[key]))
    return 1 if counter == len(a) - 1 else -1 if counter == 0 else 0

def create_Pareto_set(data):
    '''Функция для создания оптимального множества Парето по входящему множеству альтернатив'''
    losers, winners = [], []
    for i in range(len(data)):
        for j in range(i+1, len(data)):
            n = compare_alternatives(data[i], data[j])
            if n == 1:
                losers.append(data[j])
            elif n == -1:
                losers.append(data[i])
    for i in range(len(data)):
        if data[i] not in losers:
            winners.append(data[i])
    return winners

def branches_and_boundaries(data, branches):
    '''Метод указания верхних и нижних границ критериев'''
    winners = []
    for i in data:
        flag = False
        for j in branches:
            key, value = list(j.items())[0]
            if key.count('-'):
                if float(i[key]) > value:
                    flag = True
            else:
                if float(i[key]) < value:
                    flag = True
        if not flag:
            winners.append(i)
    return create_Pareto_set(winners)

def suboptimization(data, branches, main_criteria):
    '''Метод субоптимизации'''
```

*Продолжение Листинга А.1.*

```
data = branches_and_boundaries(data, branches)
maxi = max(data, key=lambda i: i[main_criteria])
return list(filter(lambda x: x[main_criteria] == maxi[main_criteria], data))

def lexical_optimization(data, priority):
    '''Лексикографический метод'''
    return [max(data, key = lambda item: tuple(item[key] for key in priority))]

with open('TPR_PRACT1_LIST.csv', encoding='utf-8') as file:
    data = [d for d in csv.DictReader(file)]

    print("Исходная таблица с альтернативами и критериями:".center(201))
    print_table(data)

    print("Оптимальное-множество Парето:".center(201))
    print_table(create_Pareto_set(data))

    print("Установка верхних и нижних границ:".center(201))
    branches = [{"Проходной балл (+)": 270}, {"Рейтинг университета (+)": 840},
                 {"Расстояние до общежития (-)": 14}]
    print_table(branches_and_boundaries(data, branches))

    print("Субоптимизация:".center(201))
    branches = [{"Проходной балл (+)": 290}, {"Расстояние до общежития (-)":
14}]
    main_criteria = "Рейтинг университета (+)"
    print_table(suboptimization(data, branches, main_criteria))

    print("Лексикографическая оптимизация:".center(201))
    priority = ("Рейтинг университета (+)", "Проходной балл (+)", "Стоимость
обучения (+)",
               "Кол-во бюджетных мест (-)", "Расстояние до общежития (-)",
               "Размер стипендии (+)")
    print_table(lexical_optimization(data, priority))
```