



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

Арефьев А. М.
(Ф.И.О. студента)

Руководитель

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.	4
1.3 Субоптимизация	4
1.4 Лексикографическая оптимизация	4
1.5 Результаты работы программы.....	4
ЗАКЛЮЧЕНИЕ	6
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	7
ПРИЛОЖЕНИЯ.....	8

ВВЕДЕНИЕ

Целью практической работы является познакомиться с методом многокритериальной оптимизации Парето, а также рассмотреть методы сужения оптимального множества альтернатив, а именно метод верхних/нижних границ, метод субоптимизации и метод лексикографической оптимизации.

В данной работе осуществляется применение метода Парето для определения оптимального выбора автомобиля, учитывая пять разнообразных критериев. Рассматриваются различные методы и подходы. Вручную и при помощи кода.

Метод Парето находит широкое применение в множестве задач, направленных на оптимизацию процессов и принятия решений. Этот метод помогает идентифицировать наиболее значимые факторы в наборе данных, сосредоточив внимание на ключевых проблемах или возможностях для улучшения.

1 МЕТОД ПАРЕТО

Алгоритм метода Парето:

1. Подготавливаются данные и классифицируются данные по категориям.
2. Выбираются оптимальные категории.
3. Применяются необходимые методы.
4. Визуализируется результат.
5. По визуализируемому результату применяется оптимальное решение

1.1 Выбор Парето-оптимального множества

Задача заключается в выборе оптимального автомобиля по построенным мною критериям. Было выбрано 10 альтернатив – различных марок авто по 5 критериев и составлена таблица 1.

Таблица 1. Альтернативы и критерии

№	Варианты	Критерии				
	Название	Цена, тыс. р -	Расход топлива, л/100км -	Надежность (из 10) +	Комфорт (из 10) +	Мощность двигателя, л.с. +
1	Skoda Kodiaq	2800	6.0	10	10	250
2	Vw tiguan	3000	6.1	10	9	240
3	Hyundai creta	1 600	6.9	6	3	140
4	Tank 300	2 000	8.5	5	5	220
5	Kia carnival	7 000	9.0	7	9	200
6	Toyota rav4	3 800	7.0	9	6	170
7	Kia seltos	1 700	6.5	5	4	160
8	Bmw x3	3100	6.2	9	10	230
9	Omoda c5	3 200	7.1	5	6	165
10	mitsubishi outlander	2900	5.9	8	8	220

Таблица 2. Сравнение альтернатив

	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x	x	x	x	x
2	A1	x	x	x	x	x	x	x	x	x
3	н	н	x	x	x	x	x	x	x	x
4	н	н	н	x	x	x	x	x	x	x
5	A1	A2	н	н	x	x	x	x	x	x
6	A1	A2	н	н	н	x	x	x	x	x
7	н	н	н	н	н	н	x	x	x	x
8	A1	н	н	н	A8	A8	н	x	x	x
9	A1	A2	н	н	н	н	н	A8	x	x
10	н	н	н	н	н	н	н	н	A10	x

Таблица 3. Парето-оптимальное множество

№	Варианты	Критерии				
	Название	Цена, тыс. р -	Расход топлива, л/100км -	Надежность (из 10) +	Комфорт (из 10) +	Мощность двигателя, л.с. +
1	Skoda Kodiaq	2800	6.0	10	10	250
2	Vw tiguan	3000	6.1	10	9	240
8	Bmw x3	3100	6.2	9	10	230
10	mitsubishi outlander	2900	5.9	8	8	220

Очевидно, что выделение множества Парето не является удовлетворительным решением.

1.2 Указание верхних/нижних границ критериев.

Установим границы. Пусть цена будет меньше или равна 3000, а

надежность больше или равна 8. Отообразим в таблицу 4.

Таблица 4. Результат верхних и нижних границ критериев

№	Варианты	Критерии				
	Название	Цена, тыс. р -	Расход топлива, л/100км -	Надежность (из 10) +	Комфорт (из 10) +	Мощность двигателя, л.с. +
1	Skoda Kodiaq	2800	6.0	10	10	250
2	Vw tiguan	3000	6.1	10	9	240
10	mitsubishi outlander	2900	5.9	8	8	220

1.3 Субоптимизация

Выберем главный критерий – цена. Установим границы надежность больше или равна 8 и мощность двигателя больше или равна 230. Отообразим в таблицу 5.

Таблица 5. Результат субоптимизации

№	Варианты	Критерии				
	Название	Цена, тыс. р -	Расход топлива, л/100км -	Надежность (из 10) +	Комфорт (из 10) +	Мощность двигателя, л.с. +
1	Skoda Kodiaq	2800	6.0	10	10	250

1.4 Лексикографическая оптимизация

Упорядочим критерии по их относительной важности:

1. Надежность
2. Комфорт
3. Мощность двигателя
4. Расход топлива

5. Цена

Отообразим результат в таблице 6.

Таблица 6. Результат лексикографической оптимизации

№	Варианты	Критерии				
	Название	Цена, тыс. р -	Расход топлива, л/100км -	Надежность (из 10) +	Комфорт (из 10) +	Мощность двигателя, л.с. +
1	Skoda Kodiaq	2800	6.0	10	10	250

1.5 Результаты работы программы

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	None	None	None	None	None	None	None	None	None	None
A2	A1	None	None	None	None	None	None	None	None	None
A3	н	н	None	None	None	None	None	None	None	None
A4	н	н	н	None	None	None	None	None	None	None
A5	A1	A2	н	н	None	None	None	None	None	None
A6	A1	A2	н	н	н	None	None	None	None	None
A7	н	н	н	н	н	н	None	None	None	None
A8	A1	н	н	н	A8	A8	н	None	None	None
A9	A1	A2	н	н	н	н	н	A8	None	None
A10	н	н	н	н	н	н	н	н	A10	None

Рисунок 1.5.1. Сравнение альтернатив

	Название	Цена	Расход топлива	Надежность	Комфорт	\
0	Skoda Kodiaq	2800	6.0	10	10	
1	Vw tiguan	3000	6.1	10	9	
9	mitsubishi outlander	2900	5.9	8	8	
Мощность двигателя						
0		250				
1		240				
9		220				

Рисунок 1.5.2. Результат верхних и нижних границ критериев

	Название	Цена	Расход топлива	Надежность	Комфорт	\
0	Skoda Kodiaq	2800	6.0	10	10	
1	Vw tiguan	3000	6.1	10	9	
9	mitsubishi outlander	2900	5.9	8	8	
Мощность двигателя						
0	250					
1	240					
9	220					

Рисунок 1.5.2. Результат верхних и нижних границ критериев

	Название	Цена	Расход топлива	Надежность	Комфорт	Мощность двигателя	
0	Skoda Kodiaq	2800	6.0	10	10	250	

Рисунок 1.5.3. Результат субоптимизации

Executed at 2024.02.24 18:12:48 in 44ms							
	Название	Цена	Расход топлива	Надежность	Комфорт	Мощность двигателя	
0	Skoda Kodiaq	2800	6.0	10	10	250	

Рисунок 1.5.4. Результат лексикографической оптимизацииЧ

ЗАКЛЮЧЕНИЕ

Мною была проведена работа по поиску оптимального автомобиля по моим критериям. Работа выполнялась при помощи метода Парето.

Могу выделить плюсы:

1. Простота
2. Универсальность
3. Эффективность

И минусы:

1. Ограниченность
2. Переоценка
3. Риск упущения

В заключение, могу сказать, что метод Парето является мощным инструментом для анализа и принятия решений, но, как и любой другой метод, имеет свои ограничения и должен использоваться с учетом специфики ситуации и в комбинации с другими инструментами анализа.

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 в блокноте Jupyter.

Приложение А

Код реализации метода Парето на языке Python.

Листинг А.1. . Реализация Парето.

```
import pandas as pd
import numpy as np

data = {
    'Название': ['Skoda Kodiaq', 'Vw tigan', 'Hyundai creta', 'Tank 300', 'Kia carnival',
                'Toyota rav4', 'Kia seltos', 'Bmw x3', 'Omoda c5', 'mitsubishi outlander'],
    'Цена': [2800, 3000, 1600, 2000, 7000, 3800, 1700, 3100, 3200, 2900],
    'Расход топлива': [6.0, 6.1, 6.9, 8.5, 9.0, 7.0, 6.5, 6.2, 7.1, 5.9],
    'Надежность': [10, 10, 6, 5, 7, 9, 5, 9, 5, 8],
    'Комфорт': [10, 9, 3, 5, 9, 6, 4, 10, 6, 8],
    'Мощность двигателя': [250, 240, 140, 220, 200, 170, 160, 230, 165, 220]
}

df = pd.DataFrame(data)
df
-----

def alternatives(alt1, alt2, directions):
    for i in range(len(directions)):
        if (directions[i] and alt1.iloc[i] < alt2.iloc[i]) or (not directions[i] and alt1.iloc[i] > alt2.iloc[i]):
            return False
    return True

def generate_comparison_df(altered_df, criteria_directions):
    comparison_matrix = np.full((10, 10), None)

    for i in range(10):
        for j in range(i):
            if alternatives(altered_df.iloc[i, 1:], altered_df.iloc[j, 1:], criteria_directions):
                comparison_matrix[i, j] = 'A' + str(i+1)
            elif alternatives(altered_df.iloc[j, 1:], altered_df.iloc[i, 1:], criteria_directions):
                comparison_matrix[i, j] = 'A' + str(j+1)
            else:
                comparison_matrix[i, j] = 'н'
    comparison_df = pd.DataFrame(comparison_matrix, columns=['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8',
                                                            'A9', 'A10'])
    comparison_df.index = ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10']

    return comparison_df
comparison_df = generate_comparison_df(df, criteria_directions)
print(comparison_df)

-----

print(df[(df['Цена'] <= 3000) & (df['Надежность'] >= 8)])

-----
```

```
df = df[(df['Надежность'] >=8) & (df['Мощность двигателя'] >= 230)]  
df.sort_values(['Цена'])  
print(df.head(1))
```

```
-----  
  
df = df.sort_values(  
    ['Надежность', 'Комфорт', 'Мощность двигателя', 'Расход топлива', 'Цена'],  
    ascending=[False, False, False, True, True]  
)  
print(df.head(1))
```