Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Институт информатики, математики и робототехники

Кафедра вычислительной математики и кибернетики

Лабораторная работа № 5

по дисциплине «Теория принятия решений»

на тему: «Метод PROMETHEE»

Выполнили:

студенты группы ПРО-431 К.И. Арменшин

Д.В. Блинов

Т.А. Тюрганов

Проверил: И.А. Нагимова

Уфа – 2025

**Цель:**

Целью работы является освоение способа построения рейтинга объектов.

**Задание:**

1. Изучение многокритериального метода построения рейтингов PROMETHEE.
2. Реализация метода в виде программного продукта.

# Ход работы

**Постановка задачи**

***Дано***:

 – множество альтернатив,

 – множество критериев.

***Требуется***: упорядочить альтернативы по ценности.

**Описание примера задачи построения рейтинга для метода PROMETHEE**

Задача выбора фреймворка на языке программирования JavaScript для разработки приложений, работающих в браузере. Выбраны следующие 4 альтернативы:

Vue

React

Angular

Svelte

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Альтернативы | Фреймворк | Простота обучения и использования | Производительность | Гибкость и экосистема |
| a1 | Vue | 4 | 5 | 4 |
| a2 | React | 3 | 3 | 9 |
| a3 | Angular | 1 | 3 | 6 |
| a4 | Svelte | 4 | 6 | 3 |

1. Сравнение производилось по следующим критериям:
   1. Простота обучения и использования – 0,5. Насколько скоро будет происходить реакция на изменение состояния
   2. Насколько легко новичку изучить фреймворк и начать разрабатывать приложения.
   3. Качество документации, наличие сообщества и обучающих материалов.
2. Производительность – 0,3
   1. Скорость рендеринга, обновления DOM и работы приложения.
   2. Оптимизация для больших и сложных приложений.
3. Гибкость и экосистема – 0,2
   1. Наличие готовых решений, библиотек и инструментов.
   2. Возможность интеграции с другими технологиями и поддержка сторонних пакетов.

***Шаг 1. Парные сравнения альтернатив по каждому критерию.*** Для каждой пары альтернатив  рассчитывается разница по каждому критерию *fk*:



Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

***Шаг 2. Вычисление мер предпочтения по критериям****.*

Вычисление мер предпочтения, обозначающихся , осуществляется как отображение  в [0,1]:



где *Pk* – положительная неубывающая функция, принимающая 6 форм:



Обычная функция

Изображение выглядит как текст, линия, Шрифт, рукописный текст

Контент, сгенерированный ИИ, может содержать ошибки.

*U*-образная функция



*V*-образная функция

Изображение выглядит как текст, линия, Шрифт, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Уровневая функция



*V*-образная функция с порогами безразличия



Функция Гаусса

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

***Шаг 3. Вычисление индексов предпочтения для каждой альтернативы.*** Как только парные сравнения действий были сделаны для каждого критерия, можно агрегировать эти значения, чтобы получить глобальную степень предпочтения при помощи взвешенной суммы – индекса предпочтения:



где *wk* – относительная важность критерия ,.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

***Шаг 4. Вычисление положительных, отрицательных и чистых оценок***:

− положительный поток;

− отрицательный поток;

 − чистый поток (PROMETHEE-II).

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Результат ранжирования альтернатив**



**Руководство пользователя на примере задачи.**

1. Для построения таблицы в левой верхней части программы зададим количество альтернатив и критериев. В нашем случае 4 и 3 соответственно.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Поля ввода количества альтернатив и критериев

1. Ниже сформировалась таблица, которую заполняем данными из задачи.

Изображение выглядит как текст, снимок экрана, линия, число

Контент, сгенерированный ИИ, может содержать ошибки.

Заполненная таблица

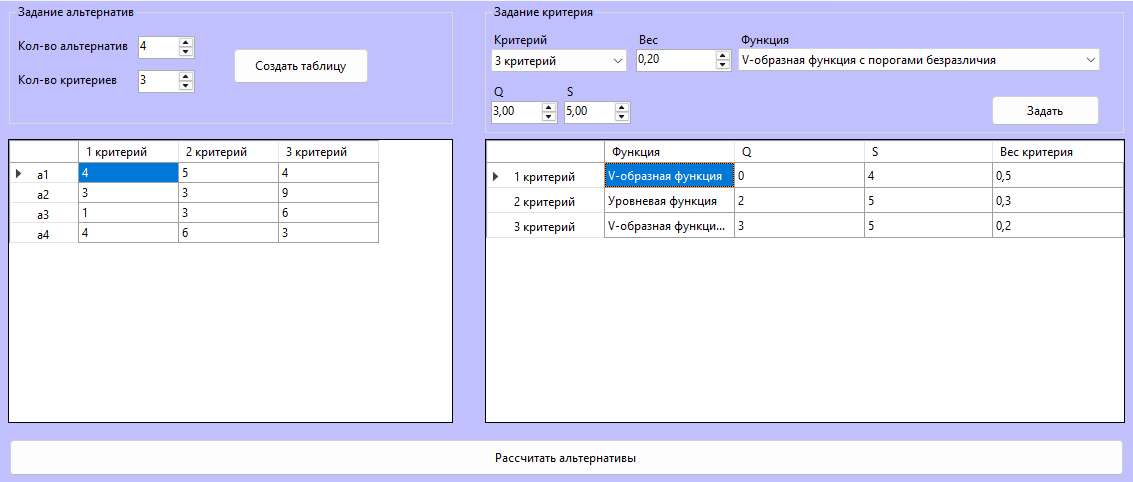
1. В правой части окна задаем меры предпочтения по каждому критерию. Сумма весов всех критериев должна быть строго равна 1. В зависимости от вида функции могут быть доступны для заполнения параметры Q и S.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Поля ввода меры предпочтения по 1-му критерию

В результате окно программы выглядит следующим образом:



Окно программы

1. Введённые значения можно сохранить в файл, используя кнопку «Сохраниь» и указав путь и название файла.



Кнопка «Сохраниь»

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Путь и название файла

1. Также ранее сохранённые значения можно загрузить из файла, используя кнопку «Открыть» и указав путь и название файла.



Кнопка «Открыть»



Путь и название файла

1. Далее нажимаем кнопку рассчитать. Откроется окно с результатами ранжирования с пошаговым отображением процесса моделирования.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Окно результатов ранжирования

В результате было получено ранжирование: Vue, React, Svelte, Angular.

**Вывод**

В ходе выполнения данной лабораторной работы был изучен метод многокритериального построения рейтингов PROMETHEE, а также реализован алгоритм и интерфейс приложения на высокоуровневом языке программирования C# (Windows Forms).