

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики Кафедра
математических методов исследования операций

Отчет

о научно-исследовательской работе

**«Разработка и программная реализация алгоритма сравнительного анализа
инвестиционных проектов на основе применения аппарата интервальных
нечетких чисел второго типа»**

Направление 02.04.02 – Фундаментальная информатика и информационные
технологии

Программа – Машинное обучение и интеллектуальные информационные
технологии

Курс 1 группа 13 2 курс 1 семестр

Обучающийся

Крутских Т.В.

Руководитель

д.т.н., проф. Каширина И. Л.

Содержание

Введение	3
1. Постановка задачи исследования.....	5
2. Алгоритмическая и программная реализация сравнительной оценки эффективности инвестиционных проектов	7
2.1. Алгоритм оценки эффективности проекта на основе интервальных нечетких чисел второго типа.....	7
2.3.1 Практическое применение реализация построенного алгоритма ...	22
Заключение.....	39
Список используемых источников	41

Введение

В современной экономике принято оценивать эффективность инвестиционных проектов перед их реализацией. Это позволяет инвесторам и руководителям принимать обоснованные решения о вложении капитала и выборе наиболее перспективных проектов. Однако оценка эффективности инвестиционных проектов является сложной задачей, так как в нее входят различные аспекты, включая финансовые показатели, риски, временные рамки и другие факторы.

Для более точной и надежной оценки эффективности инвестиционных проектов в последние годы активно применяются методы нечеткой логики. Нечеткая логика позволяет учитывать неопределенность и нечеткость входных данных и принимать решения на основе нечетких правил. Один из таких методов - использование интервальных нечетких чисел второго типа.

Интервальные нечеткие числа второго типа представляют собой комбинацию интервалов и функций принадлежности, которые описывают неопределенность и нечеткость параметров проекта. Они позволяют учитывать различные степени уверенности и возможности в рамках оценки эффективности проекта.

В данной работе будет рассмотрен алгоритм оценки эффективности инвестиционных проектов на основе интервальных нечетких чисел второго типа. Этот алгоритм позволяет учесть неопределенность и различные факторы, влияющие на эффективность проекта, и дает более полное представление о его потенциальной прибыльности и рисках.

Целью данной работы является разработка алгоритма оценки эффективности инвестиционных проектов с использованием интервальных нечетких чисел второго типа и его программная реализация. Это позволит инвесторам и руководителям более точно оценить потенциальные инвестиционные возможности и принять обоснованные решения о вложении капитала.

В дальнейшем разделе работы будет представлен сам алгоритм оценки

эффективности проекта на основе интервальных нечетких чисел второго типа, а также описание его программной реализации.

1. Постановка задачи исследования

Задача исследования состоит в разработке и программной реализации алгоритма для сравнительной оценки эффективности инвестиционных проектов с использованием интервальных нечетких чисел второго типа.

Алгоритм должен основываться на теории интервальных нечетких чисел второго типа, которая позволяет учитывать неопределенность и нечеткость при оценке эффективности инвестиционных проектов. Интервальные нечеткие числа второго типа представляют собой интервалы, где каждое значение в интервале имеет свою степень принадлежности.

Алгоритмическая и программная реализация должны включать следующие шаги:

- Сбор данных. Сбор и анализ информации о различных инвестиционных.
- Проектах, включая ожидаемую доходность, риски, стоимость и другие факторы.
- Моделирование интервальных нечетких чисел. Разработка математической модели для представления интервальных нечетких чисел второго типа. Это может включать определение функции принадлежности и операций над интервальными нечеткими числами.
- Оценка эффективности проектов. На основе собранных данных и модели интервальных нечетких чисел, разработка метода для оценки эффективности инвестиционных проектов. Этот метод должен учитывать неопределенность и нечеткость входных данных.
- Сравнительный анализ. Разработка алгоритма для сравнительного анализа инвестиционных проектов на основе оценки их эффективности. Этот алгоритм должен учитывать важность различных факторов и весовые коэффициенты, задаваемые пользователем.
- Реализация программного обеспечения. Разработка программного кода, основанного на разработанном алгоритме. Программное обеспечение

должно предоставлять удобный интерфейс для ввода данных о проектах, вычисления оценок и вывода результатов сравнительного анализа.

- Тестирование и валидация. Проведение тестирования программного обеспечения для проверки его корректности и эффективности. Валидация алгоритма может включать сравнение его результатов с известными моделями и методами оценки эффективности инвестиционных проектов.

- Документирование и анализ результатов. Документирование разработанного алгоритма и программного обеспечения, а также анализ полученных результатов сравнительного анализа инвестиционных проектов.

В итоге, разработанный алгоритм и программное обеспечение должны предоставлять возможность сравнивать эффективность различных инвестиционных проектов, учитывая неопределенность и нечеткость входных данных с использованием интервальных нечетких чисел второго типа.

2. Алгоритмическая и программная реализация сравнительной оценки эффективности инвестиционных проектов

2.1. Алгоритм оценки эффективности проекта на основе интервальных нечетких чисел второго типа

Чтобы уменьшить риск возможных потерь, для оценивания каких-либо экономических показателей привлекаются эксперты. Достаточно часто экспертные оценки предоставляются в виде нечетких чисел. Выбор именно нечетких множеств для определения показателей не является случайным, так как они хорошо зарекомендовали себя при обработке нечеткой информации и при адекватной формализации экспертных знаний.

Однако все эксперты имеют разные опыт и квалификацию, поэтому полученные от них оценки следует формализовать для использования в интеллектуальных системах анализа данных. За экспертной информацией должен осуществляться контроль качества в виде коэффициентов парной и общей согласованности.

Алгоритм поиска сильно отличающихся экспертных критериев:

1. Для каждого эксперта вычислить сумму показателей парного согласованности его критерия с другими критериями. Данные записываются в матрицу согласованности показателей.

2. Выбирается критерий, имеющий наименьшую сумму показателей согласованности, и исключается из дальнейшего рассмотрения.

3. Повторно определяется показатель общей согласованности. Если показатель удовлетворяет заданному порогу, то переход на шаг 4. Если же показатель общей согласованности все еще не является удовлетворительным, то переход на шаг 2.

4. Анализ завершается при показателе общей согласованности, удовлетворяющему пороговому значению.

В данном исследовании эксперты задают входные данные на основе интервальных значений при помощи функции принадлежности первого порядка.

Далее на основе построения обобщенного экспертного критерия строится функция принадлежности второго порядка.

Выведем алгоритм, позволяющий на основе интервальных нечетких чисел второго типа производить оценку эффективности инвестиционного проекта.

1. Вводится информация о количестве экспертов k , участвующих в оценке будущих доходах от проекта, длительность проекта T , ставка дисконтирования r , первоначальные взносы I (2.1.), выраженные интервальным нечетким числом:

$$I = (i_l^U, i_m^U, i_r^U), (i_l^L, i_m^L, i_r^L), \quad (2.1)$$

Где $(i_l^U, i_m^U, i_r^U), (i_l^L, i_m^L, i_r^L)$ – опорные значения интервального нечеткого множества;

2. Предположим, что информация, полученная от k экспертов, формализована согласно методу [6] и соответственно для каждого эксперта построено k треугольно-образных функций принадлежности:

$$\mu_i(x) = (a_l, a_m, a_r), i = \overline{1, k}, \quad (2.2)$$

Где, соответственно a_l, a_r – левый и правый параметр нечеткости, а a_m – центральное значение. Построенные нечеткие числа будем называть критериями экспертов оценивания.

3. Производится оценка согласованности экспертов на основе понятия обычного множества, ближайшего к нечеткому множеству.

В теории экспертного оценивания для определения согласованности двух и более экспертных критериев применяются различные показатели. Опираются эти показатели или на парные сравнения рангов, присвоенных одним и тем же объектам или на арифметические преобразования рангов. В данном исследовании, с целью анализа нечеткой экспертной информации предложено использовать понятие обычного множества, ближайшего к данному нечеткому множеству.

С целью анализа экспертной информации определим количественные показатели согласованности экспертных критериев на основе абстрактных понятий – значений функций принадлежности.

Так, в качестве показателя согласованности критериев i -го и j -го экспертов может быть рассмотрено отношение (2.3):

$$k_{ij} = \frac{a_{0i} \cap a_{0j}}{a_{0i} \cup a_{0j}}, i = \overline{1, k}, j = \overline{1, k} \quad (2.3)$$

где, a_{0i}, a_{0j} – обычные ближайшие множества к нечетким множествам.

Вычисляется показатель общей согласованности K . Показатель общей согласованности критериев экспертов можно рассчитать согласно формуле (2.4):

$$k = \frac{a_{01} \cap a_{02} \cap \dots \cap a_{0k}}{a_{01} \cup a_{02} \cup \dots \cup a_{0k}}, \quad (2.4)$$

где, $a_{0i}, i = \overline{1, k}$ – обычные ближайшие множества к нечетким множествам.

Если значение аддитивного показателя общей согласованности близко к нулю, то можно сделать вывод, например, о нечеткой постановке процедуры оценивания или о некомпетентности одного или нескольких экспертов.

Если аддитивный показатель общей согласованности больше 0.7, то переход на шаг 6, иначе переход на шаг 5;

4. В случае возникновения неудовлетворительного показателя общей согласованности появляется необходимость выявления тех экспертов, чье мнение существенно отличается от остальных. Их исключение происходит по алгоритму поиска сильно отличающихся экспертных критериев.

Переход на шаг 3;

5. Строится обобщенный экспертный критерий на основе интервальных нечетких чисел второго типа, который выражает доход от инвестиционного проекта на конкретный период реализации.

После анализа показателей парной и общей согласованности, на основе экспертных критериев $X_i, i = \overline{1, k}$ определяется обобщенный экспертный критерий X , для которого сформулировано условие Парето [4]. Для каждого экспертного критерия X_i , определен весовой коэффициент $\omega_i, i = \overline{1, k}$

Параметры функций принадлежности $\mu_i(x) = (a_l, a_m, a_r)$ обобщенного экспертного критерия представлены в условии: Неизвестные параметры находятся из системы нормальных уравнений (2.6) - (2.8):

$$\frac{\partial F}{\partial a_l} = 2[\sum_{i=1}^k \omega_i(a_l^i - a_l)] = 0 \quad (2.6)$$

$$\frac{\partial F}{\partial a_m} = 2[\sum_{i=1}^k \omega_i(a_m^i - a_m)] = 0 \quad (2.7)$$

$$\frac{\partial F}{\partial a_r} = 2[\sum_{i=1}^k \omega_i(a_r^i - a_r)] = 0 \quad (2.8)$$

Полученные решения представлены в (2.9) - (2.11):

$$a_l = \sum_{i=1}^k \omega_i a_l^i \quad (2.9)$$

$$a_m = \sum_{i=1}^k \omega_i a_m^i \quad (2.10)$$

$$a_r = \sum_{i=1}^k \omega_i a_r^i \quad (2.11)$$

Таким образом, обобщенный экспертный критерий предстает в виде (2.12):

$$F = [\sum_{i=1}^k \omega_i a_l^i, \sum_{i=1}^k \omega_i a_m^i, \sum_{i=1}^k \omega_i a_r^i] \quad (2.12)$$

Стоит отметить, что использование нечетких чисел первого типа усредняет мнения различных экспертов, что является скорее минусом, чем плюсом. Для учета разброса экспертных мнений и нечеткости самой степени уверенности экспертов в оценке того или иного показателя О. М. Полещук [5] предложила использование интервальных нечетких множества второго типа.

Для построения интервальных нечетких оценок предлагается использовать метод доверительных интервалов.

Рассматриваются параметры функций принадлежности k экспертных критериев

$$a_l^i, a_m^i, a_r^i, i = \overline{1, k}.$$

Для каждого из параметров имеется выборка, состоящая из k элементов (2.13) - (2.15):

$$a_l^1, a_l^2, \dots, a_l^k \quad (2.13)$$

$$a_m^1, a_m^2, \dots, a_m^k \quad (2.14)$$

$$a_r^1, a_r^2, \dots, a_r^k \quad (2.15)$$

Кроме этого, параметры функции принадлежности (2.2) определяются согласно (2.9) - (2.11).

На основе этих параметров вычисляются коэффициенты S_L и S_R как

(2.16) – (2.17):

$$S_L = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (a_l^i - a_l)^2} \quad (2.16)$$

$$S_R = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (a_r^i - a_r)^2} \quad (2.17)$$

Согласно [5] доверительные интервалы на основе распределения Стьюдента для параметров обобщенного экспертного критерия имеют вид (2.18) - (2.19):

$$\sum_{i=1}^k \omega_i a_l^i - \frac{S_L \Delta_{k-1, \alpha}}{\sqrt{k}} \leq \hat{a}_l \leq \sum_{i=1}^k \omega_i a_l^i + \frac{S_L \Delta_{k-1, \alpha}}{\sqrt{k}}, \quad (2.18)$$

$$\sum_{i=1}^k \omega_i a_r^i - \frac{S_R \Delta_{k-1, \alpha}}{\sqrt{k}} \leq \hat{a}_r \leq \sum_{i=1}^k \omega_i a_r^i + \frac{S_R \Delta_{k-1, \alpha}}{\sqrt{k}}, \quad (2.19)$$

Где $\Delta_{k-1, \alpha}$ находится из таблицы вероятностей $P(|t_{k-1}| > \Delta_{k-1, \alpha}) = \alpha$ распределения Стьюдента t_{k-1} .

Таким образом, обобщенный экспертный критерий предстает в виде интервального нечеткого множества второго типа, верхняя и нижняя функции принадлежности, которых, соответственно задаются параметрами (2.20) и (2.21):

$$\overline{f(x)} = \sum_{i=1}^k \omega_i a_l^i - \frac{S_L \Delta_{k-1, \alpha}}{\sqrt{k}}, \sum_{i=1}^k \omega_i a_m^i, \sum_{i=1}^k \omega_i a_r^i + \frac{S_R \Delta_{k-1, \alpha}}{\sqrt{k}}, \quad (2.20)$$

$$\underline{f(x)} = \sum_{i=1}^k \omega_i a_l^i + \frac{S_L \Delta_{k-1, \alpha}}{\sqrt{k}}, \sum_{i=1}^k \omega_i a_m^i, \sum_{i=1}^k \omega_i a_r^i - \frac{S_R \Delta_{k-1, \alpha}}{\sqrt{k}}, \quad (2.21)$$

Полученный обобщенный экспертный критерий, построенный на основе интервальных нечетких чисел второго типа, позволяет учесть разброс мнений экспертов, а также получить интервальную оценку степени уверенности группы экспертов в том или ином решении.

Обобщенный экспертный критерий позволяет выразить доходы от проекта на каждом промежутке реализации с целью последующего использования в анализе инвестиционного проекта.

6. Построенные спрогнозированные доходы для каждого промежутка реализации подставляются в формулу (2.3) для нахождения коэффициента

$$NPV = - \sum_{t=1}^k \frac{I_t}{(1+r)^t} + \sum_{t=k}^T \frac{C_t}{(1+r)^t}, \quad (2.22)$$

NPV:

Где I_t – величина взносов, C_t – доходы в период времени t , t – шаг расчета (год, квартал, месяц и т.д.),

7. Происходит переход к интервальным нечётким числам первого типа (2.23):

$$B = \theta A^L + (1 - \theta)A^U, \quad (2.23)$$

Где B – интервальное нечеткое число первого типа, A^U – верхняя мера принадлежности, A^L – нижняя мера принадлежности, θ – коэффициент риска

8. Далее необходимо выполнить нахождение вероятности доминирования одного проекта над другим. Этот процесс осуществляется с оценкой всех рассматриваемых проектов. Для этого нужно применить формулу (2.24)

$$L(A_\rho \geq A_\beta) = \max \left\{ \left[1 - \frac{\sum_{\xi=1}^3 \max(a_{\xi\beta} - a_{\xi\rho}, 0) + (a_{3\beta} - a_{1\rho})}{\sum_{\xi=1}^3 |a_{\xi\beta} - a_{\xi\rho}| + (a_{3\beta} - a_{1\beta})}, 1 \right], 0 \right\} \quad (2.24)$$

9. Завершить работу алгоритма и получить ранжированный список инвестиционных проектов по принципу от потенциально менее успешного – к потенциально более успешному.

Блок-схема данного алгоритма представлена на рисунке (2.1.).

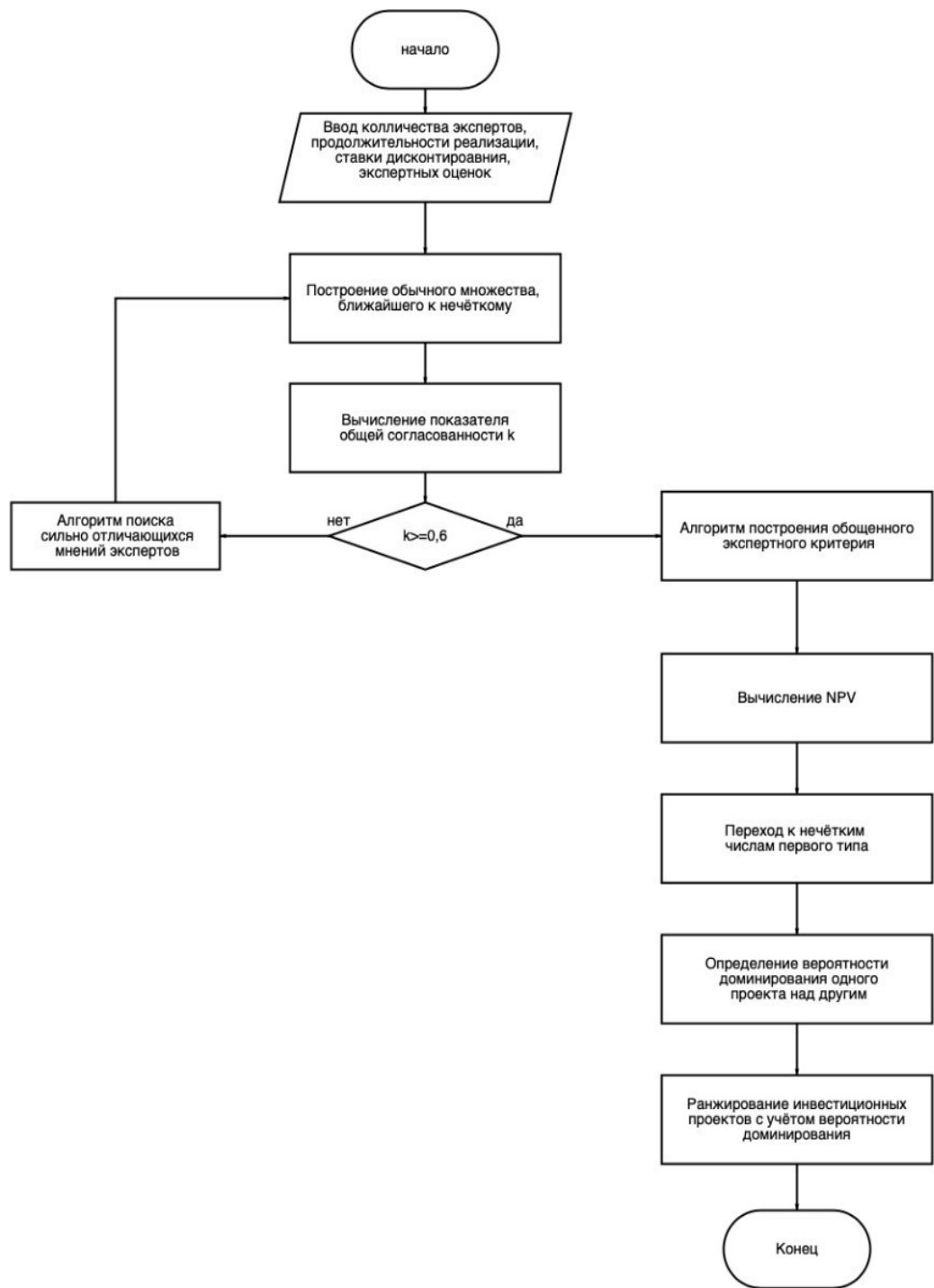


Рис. 2.1. Блок-схема алгоритма оценки эффективности инвестиционного проекта

2.3.1 Программная реализация построенного алгоритма

Построенный в предыдущем параграфе алгоритм реализован на языке программирования Python 3. Данный выбор не является случайным, так как в данном языке предоставлены проработанные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию. Стандартная библиотека включает большой набор полезных переносимых функций, начиная от функционала для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на Си или C++.

Так, в данном исследовании использовались следующие вспомогательные библиотеки:

- `numpy` – это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами;
- `skfuzzy` – этот пакет реализует множество полезных инструментов и функций для вычислений, связанных с нечеткой логикой;
- `matplotlib.pyplot` – библиотека используется для визуализации результатов. Matplotlib содержит разные интерфейсы (способов построения фигуры) и может взаимодействовать с большим количеством бекендов.

В следующем разделе будут подробно описаны составные части программы.

2.1.2. Описание методов программы

Для начала рассмотрим методы используемых библиотек:

1. numpy:

1.1. Метод *shape()* – это кортеж натуральных чисел, показывающий длину массива по каждой оси;

1.2. Функция *zeros(shape, dtype=float, order='C')* – возвращает массив указанной формы и типа, состоящий из нулей;

1.3. Функция *eye(N, M=None, k=0, dtype=<type 'float'>, order='C')* – возвращает двумерный массив, у которого элементы на главной диагонали равны единице, а остальные – нулями;

1.4. Функция *delete(a, obj, axis=None)* – удаляет указанные элементы на указанной оси;

1.5. Функция *array(object, dtype=None, *, copy=True, order='K', subok=False, ndmin=0, like=None)* – создает массив;

1.6. Функция *arange([start,]stop, [step,]dtype=None)* – возвращает одномерный массив с равномерно разнесенными значениями внутри заданного интервала;

2. Math:

2.1. Функция *sqrt(num)* – возвращает квадратный корень из числа *num*.

3. Skfuzzy:

3.1. Функция *trimf(x, abc)* – генерирует треугольные функции принадлежности.

4. matplotlib.pyplot:

4.1. Функция *subplots(nrows=1, ncols=1, sharex=False, sharey=False, squeeze=True, subplot_kw=None, gridspec_kw=None)* – создает фигуру и набор подзаголовков;

4.2. Функция *plot(*args, scalex=True, scaley=True, data=None, **kwargs)* – построение графика *y* и *x* в виде линий и/или маркеров;

4.3. Метод *set_title(label, fontdict=None, loc='center')*,

pad=None,
***kwargs*) – добавляет заголовок к области графика;

4.4. Метод *legend()* – размещает легенду на область видимости графика;

4.5. Метод *savefig(*args, **kwargs)* – сохраняет полученный график как изображение.

Ниже представлены методы, реализованные для алгоритма оценки эффективности инвестиционного проекта.

1. Метод *matrix_for_one_year(dataframe, year)* – возвращает матрицу для текущего года. Параметр *dataframe* состоит из критериев экспертов по всем годам, а параметр *year* означает текущий год, для которого необходимо проанализировать мнения экспертов;

2. Метод *simple_plenty(year_matrix)* – строит ближайшее четкое множество для нечеткого множества. Входной параметр *year_matrix* представляет собой матрицу с нечеткими критериями экспертов;

3. Метод *M_x(mat, experts)* – находит среднее значение по столбцам матрицы. Входной параметр *experts* задает количество экспертов, а *mat* – является матрицей, для которой нужно найти средние значения;

4. Метод *intersection(mat)* – находит пересечение строк передаваемой матрицы *mat* и возвращает длину этого пересечения;

5. Метод *union(mat)* – находит объединение строк передаваемой матрицы

mat и возвращает длину этого объединения;

6. Метод *consistency_matrix(mat, experts)* – строит матрицу согласованности для группы экспертов. На главной диагонали стоят единицы, а элемент k_{ij} определяется как показатель парной согласованности i и j экспертов. Входной параметр *mat* представляет собой обычное множество, ближайшее к нечеткому множеству, а *experts* задает количество экспертов;

7. Метод *sum_opinion(mat)* – подчитывает сумму в строке матрицы согласованности критериев *mat*;

8. Метод *consistency_check(s_m, m)* – определяет согласованность

критериев, при необходимости функция удаляет сильно отличающийся критерий из

рассмотрения. Входной параметр s_m задает обычное множество, ближайшее к нечеткому множеству, а m представляет матрицу критериев экспертов для определенного промежутка времени.

9. Метод $f()$ – позволяет построить обобщенный экспертный критерий с помощью доверительных интервалов на основе критерия Стьюдента. Возвращает верхнюю и нижнюю меру принадлежности нечеткого числа второго типа, а также коэффициент общей согласованности для конкретного промежутка реализации;

10. Метод $cf(mas)$ – находит сумму доходов по всему сроку реализации проекта. Входной параметр mas представляет собой массив;

11. Метод $fuzzy_npv()$ – рассчитывает интервальное нечеткое NPV;

12. Метод $check_project()$ – на основе среднего значения интервального нечеткого числа позволяет определить, принимать проект или нет.

2.1.3. Программный продукт

Критерий Стьюдента (также известный как t-критерий Стьюдента или просто t- тест) — это статистический метод, который используется для проверки гипотез о различии между средними значениями двух независимых выборок. Он был разработан Уильямом Стьюдентом (псевдоним Уильяма Северин С. Госсета) в начале 20-го века.

Основная идея критерия Стьюдента заключается в сравнении средних значений двух выборок с учетом их дисперсий и размеров выборок. Критерий позволяет определить, является ли различие между средними значениями статистически значимым или возникло случайно.

В общем случае критерий Стьюдента может быть применен в двух вариантах: для независимых выборок и для зависимых выборок (парный t-тест). Оба варианта основаны на распределении Стьюдента.

Для использования критерия Стьюдента важно выполнение некоторых предпосылок, таких как нормальность распределения данных и равенство дисперсий в сравниваемых группах (хотя в некоторых случаях существуют модификации критерия для работы с нарушениями этих предпосылок).

Применение критерия Стьюдента широко распространено в различных областях, включая науку, медицину, социальные науки, экономику и т.д. Он позволяет исследователям оценивать значимость различий между группами и делать выводы о влиянии рассматриваемого фактора на исследуемый показатель.

Однако стоит отметить, что критерий Стьюдента имеет свои ограничения и предположения, и его использование требует аккуратности. При работе с данным методом рекомендуется учитывать особенности конкретного исследования и проконсультироваться со специалистом в области статистики для правильного применения и интерпретации результатов.

Этот код представляет собой небольшую программу на языке Python с использованием библиотеки Tkinter для создания графического интерфейса пользователя (GUI) (рисунок 1).

Функция `win_k` создает окно с заголовком, который передается в качестве аргумента `title`. Заголовок окна устанавливается с помощью метода `title` объекта Tk, созданного из класса `tkinter.Tk` (рисунок 2).

```
In [11]: def APOS(om):
        mass_a=[]
        mass_b=[]
        for om in Om:
            mass_a.append(om[0])
            mass_b.append(om[1])
        np_mass_a=np.array(mass_a)
        np_mass_b=np.array(mass_b)
        APOS=round((np.min(np_mass_b)-np.max(np_mass_a))/(np.max(np_mass_b)-np.min(np_mass_a)),2)
        return APOS
```

Рис. 1. Функция возвращает аддитивный показатель общей согласованности

```

In [5]: def win_risk():
        window=tk.Tk()
        window.title('Выберете коэффициент риска')

        def Sub():
            window.destroy()

        risk = tk.DoubleVar()
        risk.set(0.5)
        risk0 = tk.Radiobutton(text="0 -Абсолютное неприятие риска",variable=risk, value=0)
        risk0_1 = tk.Radiobutton(text="0.1 -Очень сильное неприятие риска",variable=risk, value=0.1)
        risk0_2 = tk.Radiobutton(text="0.2 -Сильное неприятие риска",variable=risk, value=0.2)
        risk0_3 = tk.Radiobutton(text="0.3 -Среднее неприятие риска",variable=risk, value=0.3)
        risk0_4 = tk.Radiobutton(text="0.4 -Слабое неприятие риска",variable=risk, value=0.4)
        risk0_5 = tk.Radiobutton(text="0.5 -Нейтральное отношение к риску",variable=risk, value=0.5)
        risk0_6 = tk.Radiobutton(text="0.6 -Слабое предпочтение риска",variable=risk, value=0.6)
        risk0_7 = tk.Radiobutton(text="0.7 -Среднее предпочтение риска",variable=risk, value=0.7)
        risk0_8 = tk.Radiobutton(text="0.8 -Сильное предпочтение риска",variable=risk, value=0.8)
        risk0_9 = tk.Radiobutton(text="0.9 -Очень сильное предпочтение риска",variable=risk, value=0.9)
        risk1 = tk.Radiobutton(text="1 -Абсолютное предпочтение риска",variable=risk, value=1)
        button = tk.Button(text="Выберите",command=Sub)

        risk0.pack()
        risk0_1.pack()
        risk0_2.pack()
        risk0_3.pack()
        risk0_4.pack()
        risk0_5.pack()
        risk0_6.pack()
        risk0_7.pack()
        risk0_8.pack()
        risk0_9.pack()
        risk1.pack()
        button.pack()
        window.mainloop()
        return float(risk.get())

```

Рис. 2. Функция построения оконного интерфейса, ввод коэффициента риска

Внутри функции определена вложенная функция Sub, которая вызывается при нажатии на кнопку "Далее". Функция Sub вызывает метод destroy окна window, что приводит к его закрытию.

Затем создаются объекты StringVar (b) для хранения значения, введенного пользователем, и Label для отображения текста "Введите количество проектов". Объект Entry (entry) предоставляет поле ввода, в которое пользователь может ввести количество проектов. Значение по умолчанию для поля ввода устанавливается с помощью метода insert..

Далее создается объект Button (button) с надписью "Далее" и привязкой к функции Sub при нажатии на кнопку.

Объекты Label, Entry и Button упаковываются с помощью метода pack, чтобы они были отображены в окне в порядке их создания, как форма ввода количества проектов (рисунок 3,4).

```
In [4]: def win1(title):
        window=tk.Tk()
        window.title(str(title))

        def Sub():
            window.destroy()

        a1=tk.StringVar()
        label=tk.Label(text="Введите количество экспертов")
        entry = tk.Entry(textvariable=a1)
        entry.insert(tk.END,0)

        a2=tk.StringVar()
        label2=tk.Label(text="Введите срок реализации")
        entry2=tk.Entry(textvariable=a2)
        entry2.insert(tk.END,0)

        a3=tk.StringVar()
        label3=tk.Label(text="Введите ставку дисконтирования в %")
        entry3=tk.Entry(textvariable=a3)
        entry3.insert(tk.END,0)

        a4=tk.StringVar()
        label4=tk.Label(text="Введите первоначальные вложения")
        entry4=tk.Entry(textvariable=a4)
        entry4.insert(tk.END,0)

        button = tk.Button(window,text="Далее",command=Sub)

        label.pack()
        entry.pack()
        label2.pack()
        entry2.pack()
        label3.pack()
        entry3.pack()
        label4.pack()
        entry4.pack()
        button.pack()
```

Рис. 3. Функция построения оконного интерфейса, ввод первых параметров

```
In [1]: import numpy as np
        import math
        import matplotlib.pyplot as plt
        import tkinter as tk
```

Зададим t исходя из таблицы критериев Стюдента

```
In [2]: T=[4.3,3.18,2.77,2.57,2.44,2.36,2.3,2.26]
```

Форма ввода количества проектов

```
In [3]: def win_k(title):
        window=tk.Tk()
        window.title(str(title))

        def Sub():
            window.destroy()

        b=tk.StringVar()
        label=tk.Label(text="Введите количество проектов")
        entry = tk.Entry(textvariable=b)
        entry.insert(tk.END,0)

        button = tk.Button(window,text="Далее",command=Sub)

        label.pack()
        entry.pack()
        button.pack()

        window.mainloop()
        bb=int(b.get())
        return bb
```

Рис. 4. Форма ввода количества проектов

```

*****
*      Проект 1      *
*****
Год 1
Количество экспертов=4

[-225.0, -180.0, -150.0]
[-220.0, -200.0, -130.0]
[-230.0, -180.0, -160.0]
[-250.0, -165.0, -160.0]

```

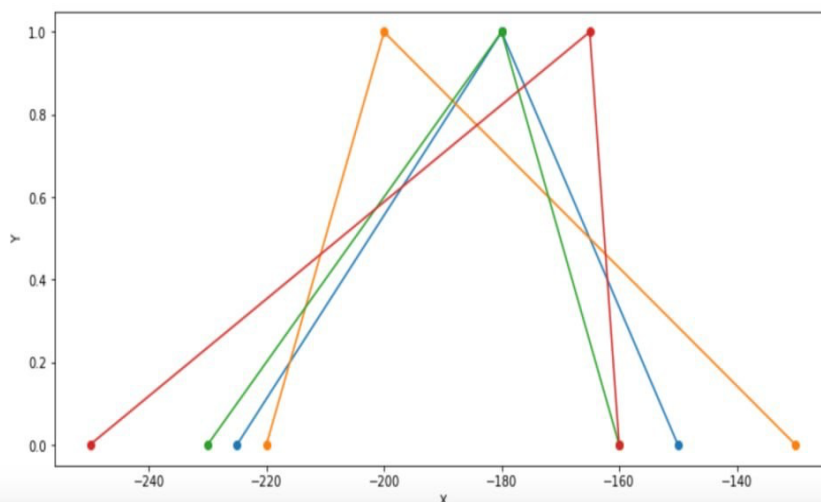


Рис. 5. Проект 1

В моей программе реализуется функция с именем NPV_1.

Функция NPV_1 принимает два аргумента: NPV и risk. Вероятно, NPV представляет собой двумерный массив или список (матрицу) значений, а risk - числовое значение, вероятностный коэффициент.

Внутри функции создается пустой список f с помощью `f = []`. Затем происходит добавление трех элементов в этот список с помощью метода `append`. В первом `append` происходит вычисление значения на основе формулы: $\text{risk} \cdot \text{NPV}[0][0] + (1 - \text{risk}) \cdot \text{NPV}[1][0]$. Здесь `NPV[0][0]` и `NPV[1][0]` представляют элементы матрицы NPV. Аналогично, во втором `append` добавляется значение `NPV[0][1]`, а в третьем `append` - значение $\text{risk} \cdot \text{NPV}[0][2] + (1 - \text{risk}) \cdot \text{NPV}[1][2]$.

Наконец, функция возвращает список f с помощью выражения `return f`.

Итак, эта функция, вероятно, выполняет какие-то вычисления на основе матрицы NPV и вероятностного коэффициента risk и возвращает результат в виде списка f.

На картинке график построения NPV первого типа для одного проекта за первый год с 4мя экспертами.

```
In [22]: def Cf_1(R,M_x):
Cf=[]
for mx in M_x:
Cf.append(mx/R)
return Cf
```

Рис. 6. Матрица Cf 1 типа

```
In [23]: def Cf_sum_1(Cf):
a=0
b=0
c=0
for cf in Cf:
a+=cf[0]
b+=cf[1]
c+=cf[2]
Cf_sum=[a,b,c]
return Cf_sum
```

Рис. 7. Матрица Cf_sum 1 типа

```
In [24]: def NPV_1(NPV,risk):
f=[]
f.append(risk*NPV[0][0]+(1-risk)*NPV[1][0])
f.append(NPV[0][1])
f.append(risk*NPV[0][2]+(1-risk)*NPV[1][2])
return f
```

Рис. 8. Нахождение NPV 1 типа

2.3.1 Практическое применение реализация построенного алгоритма

Рассмотрим практическое применение алгоритма, разработанного в параграфе 2.1, на примере трёх инвестиционных проектах, находящихся в равных условиях. То есть, срок реализации составляет 3 года, группа экспертов состоит из 4 лиц, ставка дисконтирования принимается равной 6%, первоначальный взнос равен 1456.

Экспертные критерии для первого, второго и третьего проектов представлены в таблицах 3.1., 3.2. и 3.3. соответственно.

Таблица. 3.1.

Экспертные критерии в виде нечетких чисел первого типа для первого проекта

k	1 год			2 год			3 год		
	C_l	C_m	C_r	C_l	C_m	C_r	C_l	C_m	C_r
1	-225	-180	-150	370	495	510	585	670	705
2	-220	-200	-130	390	498	500	590	670	700
3	-230	-180	-160	420	470	520	600	665	710
4	-250	-165	-160	400	485	505	600	650	710

Таблица. 3.2

k	1 год			2 год			3 год		
	c_l	c_m	c_r	c_l	c_m	c_r	c_l	c_m	c_r
1	-120	-100	-50	400	510	590	620	680	730
2	-115	-90	-60	410	515	585	610	685	735
3	-125	-110	-55	430	515	600	625	690	725
4	-120	-95	-50	425	520	590	620	685	720

Таблица.3.3.

Экспертные критерии в виде нечетких чисел первого типа для третьего проекта

k	1 год			2 год			3 год		
	C_l	C_m	C_r	C_l	C_m	C_r	C_l	C_m	C_r
1	-300	-250	-150	240	380	460	500	560	620
2	-290	-255	-155	230	395	455	510	565	625
3	-295	-260	-160	235	400	465	505	570	630
4	-300	-255	-165	245	380	466	515	555	620

В качестве примера рассмотрим 1 год. На рисунках 3.1., 3.2. и 3.3. представлена визуализация экспертных критериев для каждого проекта данного периода.

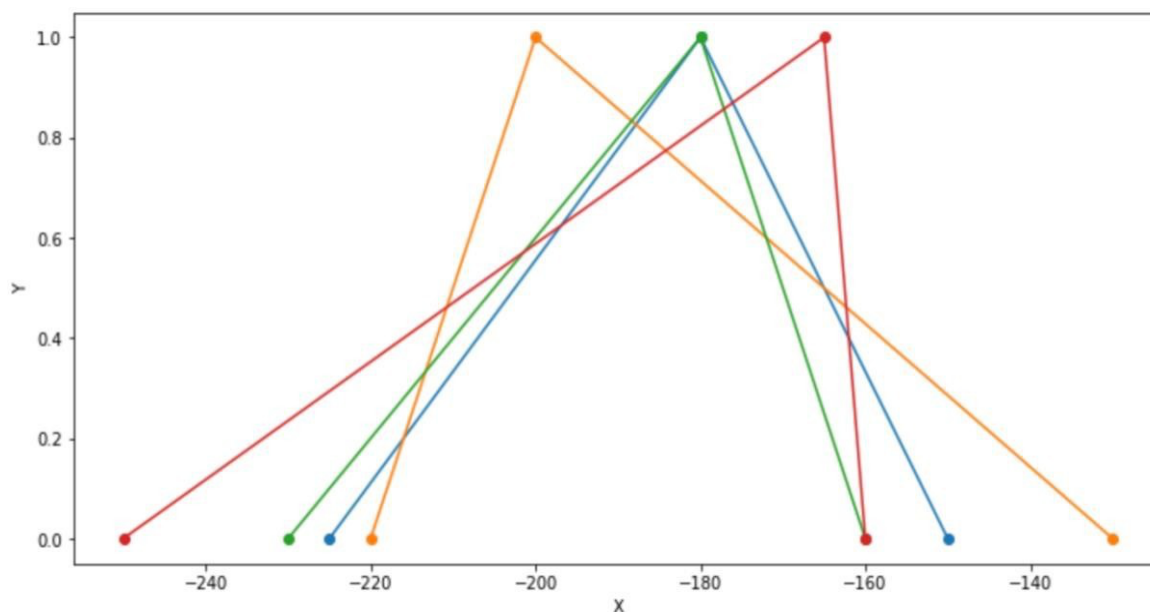


Рис. 3.1. График прогнозируемых доходов для первого проекта за 1 год

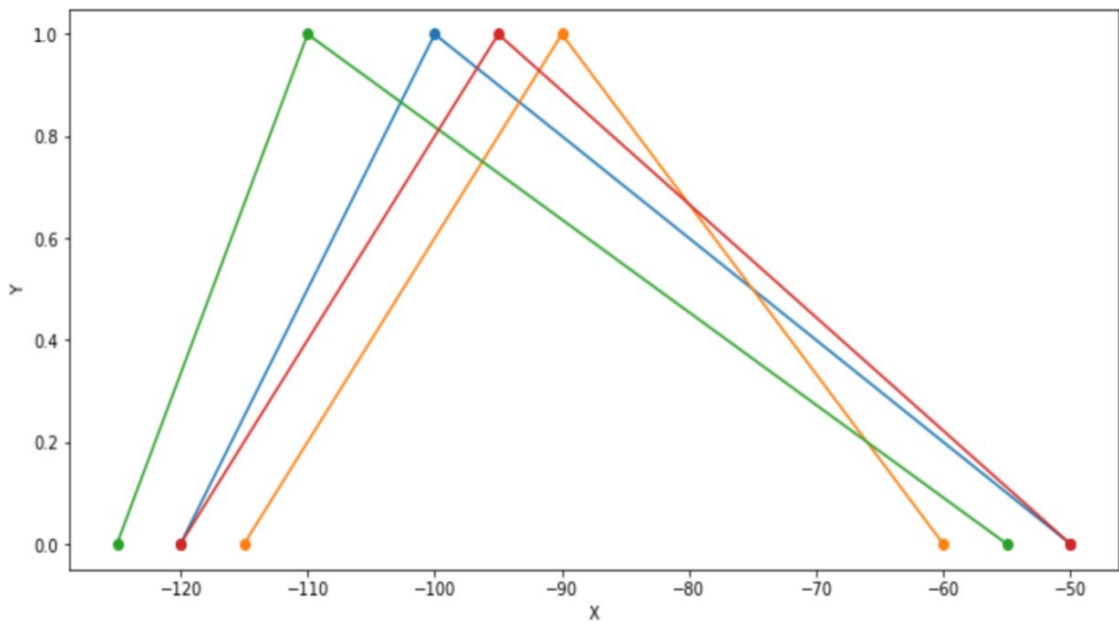


Рис. 3.2. График прогнозируемых доходов для второго проекта за 1 год

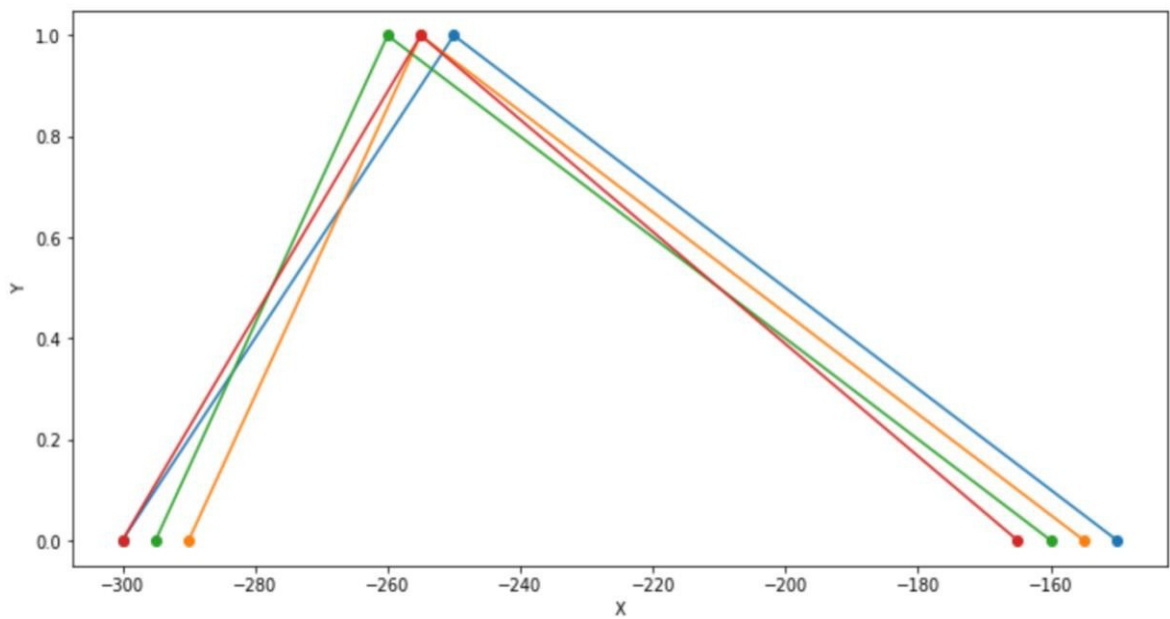


Рис. 3.3. График прогнозируемых доходов для третьего проекта за 1 год

По приведенным данным, используя написанную программу, производится расчёт матрицы согласованности экспертов. Результаты расчета представлены в таблицах 3.4., 3.5. и 3.6. соответственно для первого и второго проектов.

Таблица.3.4.

Матрица согласованности экспертов для первого проекта за 1 год

	<i>K1</i>	<i>K2</i>	<i>K3</i>	<i>K4</i>
<i>K1</i>	1.0	0.83	0.81	0.83
<i>K2</i>	0.83	1.0	0.78	0.89
<i>K3</i>	0.81	0.78	1.0	0.78
<i>K4</i>	0.83	0.89	0.78	1.0

Таблица.3.5.

Матрица согласованности экспертов для второго проекта за 1 год

	<i>K1</i>	<i>K2</i>	<i>K3</i>	<i>K4</i>
<i>K1</i>	1.0	0.79	0.65	0.87
<i>K2</i>	0.79	1.0	0.47	0.79
<i>K3</i>	0.65	0.47	1.0	0.56
<i>K4</i>	0.87	0.79	0.56	1.0

Таблица.3.6.

Матрица согласованности экспертов для третьего проекта за 1 год

	<i>K1</i>	<i>K2</i>	<i>K3</i>	<i>K4</i>
<i>K1</i>	1.0	0.9	0.84	0.84
<i>K2</i>	0.9	1.0	0.86	0.86
<i>K3</i>	0.84	0.86	1.0	1.0
<i>K4</i>	0.84	0.86	1.0	1.0

Далее рассчитывается аддитивный показатель общей согласованности экспертных критериев, который должен превышать порог, равный 0.6. Результаты для первого второго и третьего проектов за 1 год представлены в таблице 3.7.

*Результаты расчета показателя общей согласованности
экспертных критериев за 1 год*

	Аддитивный показатель общей согласованности за 1 год
Первый проект	0.68
Второй проект	0.44
Третий проект	0.81

Как видно из таблицы 3.7., для второго проекта показатель общей согласованности ниже заданного порога, поэтому для него производится перерасчет включаемых экспертных критериев по алгоритму поиска сильно отличающихся мнений рисунок 3.4.

Сумма столбцов матрицы парного сходства
[3.31 3.05 2.68 3.22]

Показатель согласованности $0.44 < 0.6$
Удаляем данные эксперта №3
Количество экспертов=3

[-120.0, -100.0, -50.0]
[-115.0, -90.0, -60.0]
[-120.0, -95.0, -50.0]

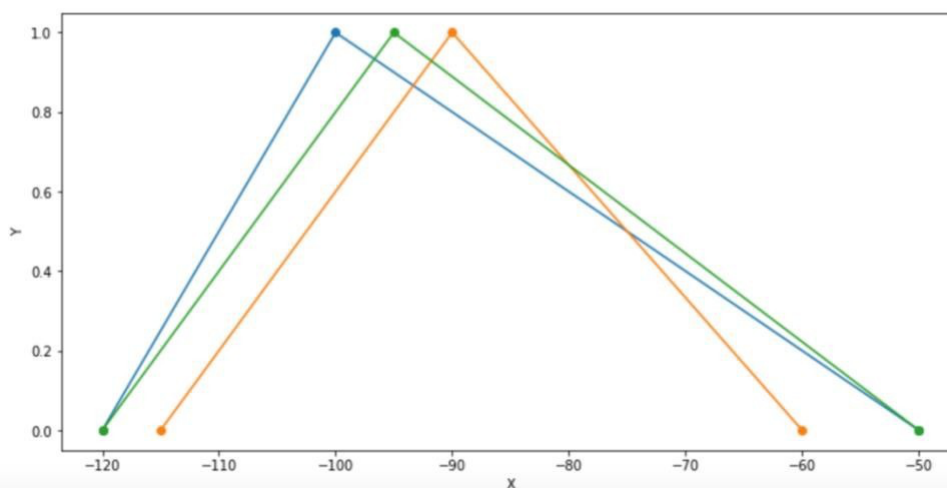


Рис. 3.4. Поиск сильно отличающегося мнения во втором проекте за 1 год

После повторной проверки адаптивного показателя согласованности у второго проекта за первый год, был получен результат, равный 0.73, что является больше порогового значения. Далее производится построение

обобщенного экспертного критерия с помощью доверительных интервалов распределения Стьюдента.

В таблицах 3.8., 3.9. и 3.10. представлены результаты построенного обобщенного экспертного критерия для первого и второго проектов соответственно.

Таблица.3.8.

Обобщенный экспертный критерий за 1 год для первого проекта

	C_l	C_m	C_r
Верхняя мера принадлежности	-252.158	-181.25	-127.514
Нижняя мера принадлежности	-210.341	-181.25	-172.485

Таблица.3.9.

Обобщенный экспертный критерий за 1 год для второго проекта

	C_l	C_m	C_r
Верхняя мера принадлежности	-123.400	-95.0	-43.1981
Нижняя мера принадлежности	-113.265	-95.0	-63.46

Таблица.3.10.

Обобщенный экспертный критерий за 1 год для третьего проекта

	C_l	C_m	C_r
Верхняя мера принадлежности	-303.861	-255.0	-147.236
Нижняя мера принадлежности	-288.638	-255.0	-167.763

На рисунках 3.5., 3.6. и 3.7. представлены графики полученных обобщенных экспертных критериев.

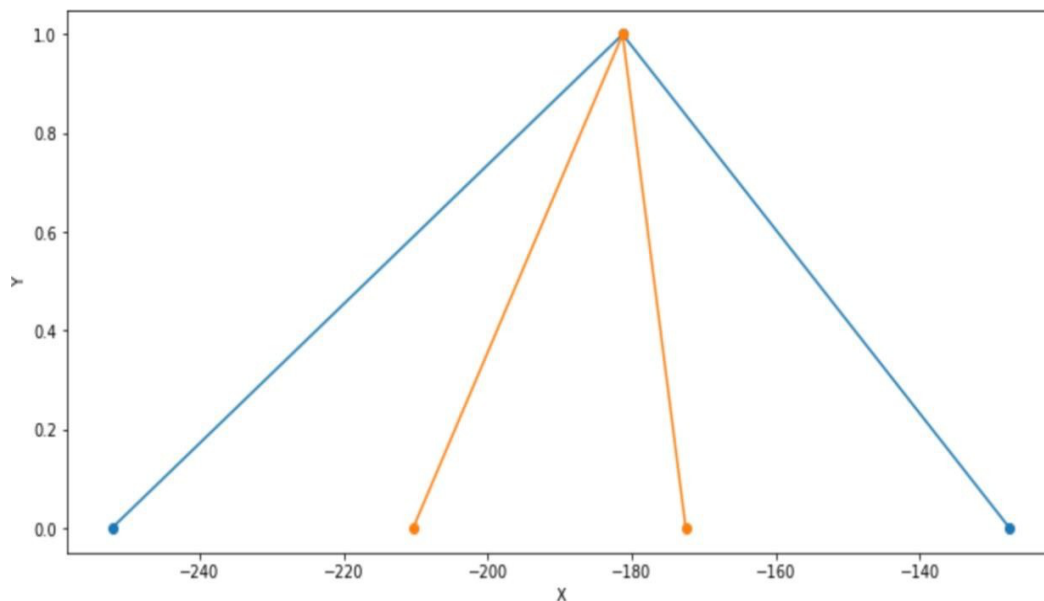


Рис.3.5. График обобщенного экспертного критерия для первого проекта

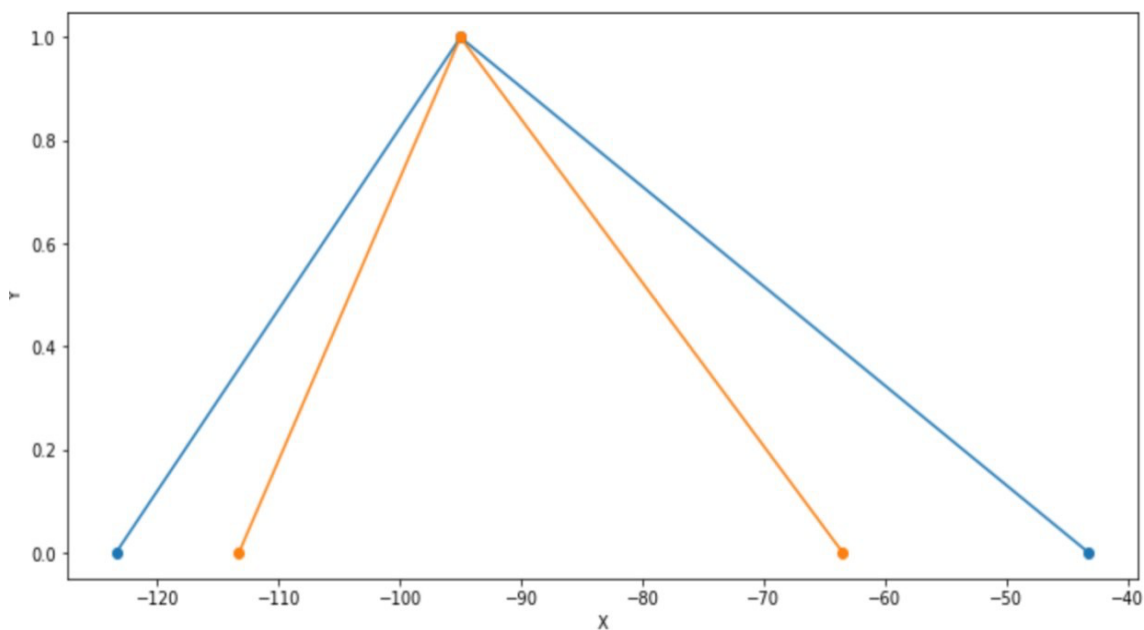


Рис.3.6. График обобщенного экспертного критерия для второго проекта

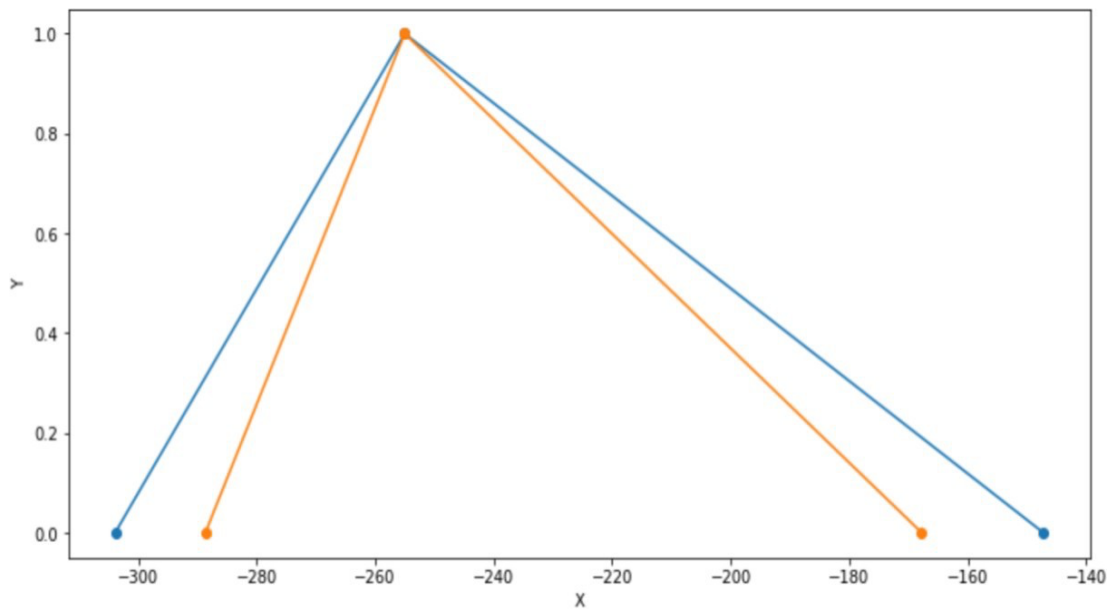


Рис.3.7. График обобщенного экспертного критерия для третьего проекта

После построения обобщенного экспертного критерия для каждого промежутка времени можно приступить к вычислению показателя NPV. Данный параметр вычисляется по формуле (2.3) построенного алгоритма в параграфе 2.1.

В таблицах 3.11., 3.12. и 3.13. представлен показатель NPV, выраженный интервальным нечетким числом второго порядка, для каждого проекта.

Таблица.3.11.

Показатель NPV для первого проекта

	npv_l	npv_m	npv_r
Верхняя мера принадлежности	-883.282	-636.265	-512.054
Нижняя мера принадлежности	-764.893	-636.265	-591.430

Таблица.3.12.

Показатель NPV для второго проекта

	npv_l	npv_m	npv_r
Верхняя мера принадлежности	-710.324	-512.135	-342.199
Нижняя мера принадлежности	-644.995	-512.135	-396.363

Таблица.3.13.

Показатель NPV для третьего проекта

	npv_l	npv_m	npv_r
Верхняя мера принадлежности	-1122.93	-878.29	-646.89
Нижняя мера принадлежности	-1073.06	-878.29	-693.38

На рисунках 3.7., 3.8. и 3.9 представлены графики построенных показателей NPV для первого, второго и третьего проектов соответственно.

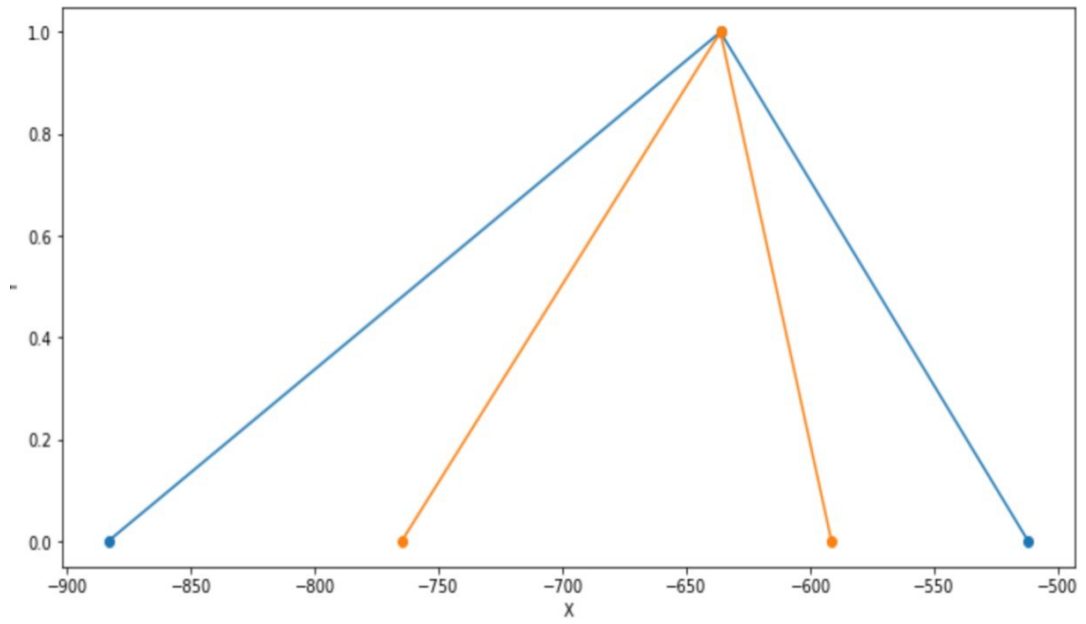


Рис.3.7. График интервального нечеткого NPV для первого проекта

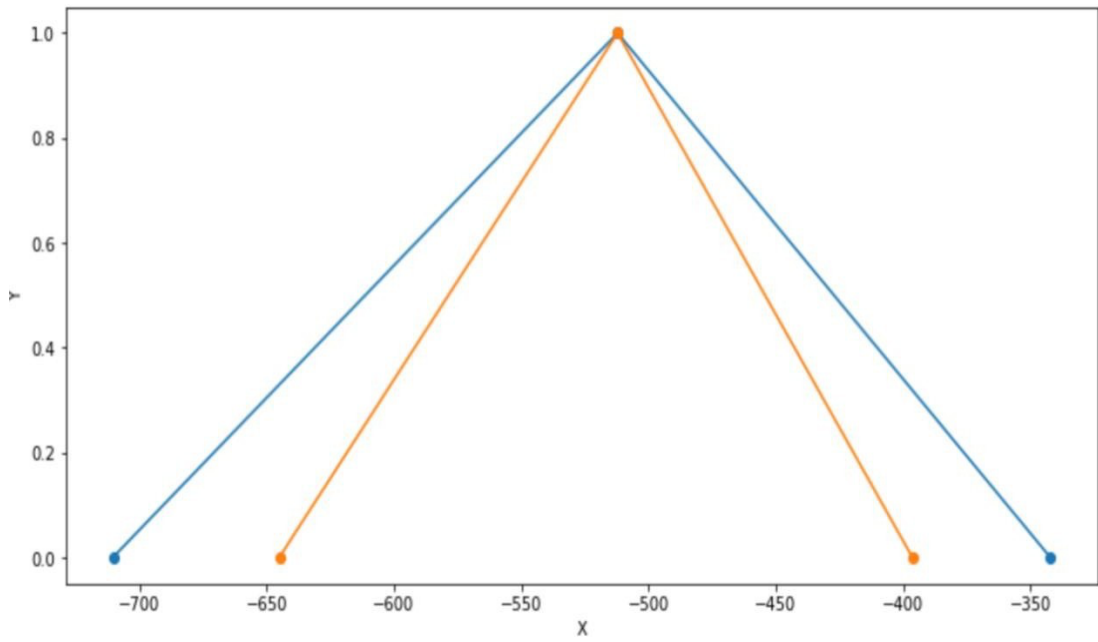


Рис.3.8. График интервального нечеткого NPV для второго проекта

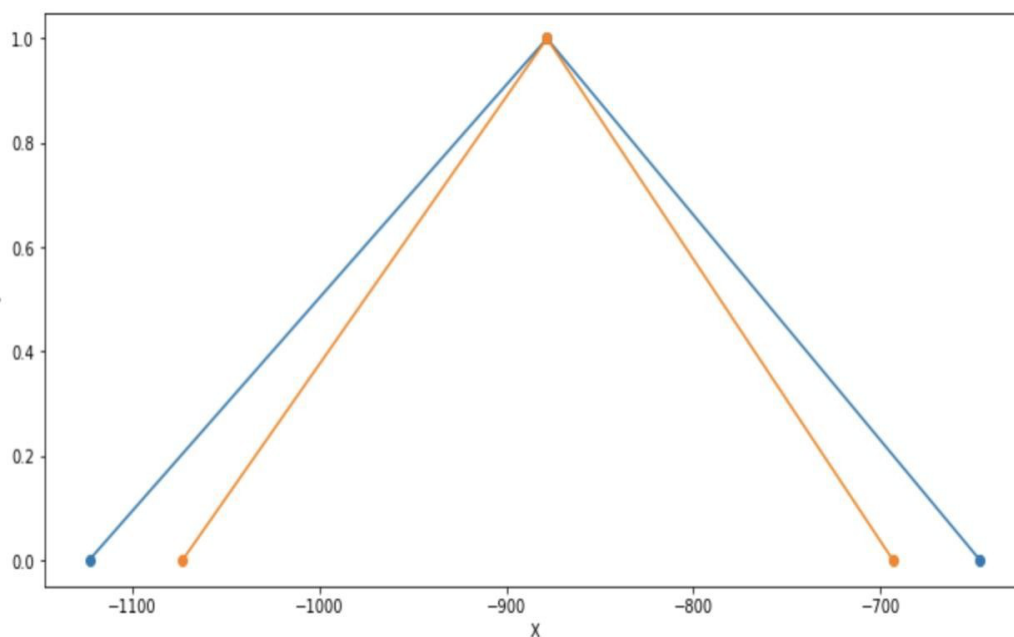


Рис.3.9. График интервального нечеткого NPV для третьего проекта

Для оценки полученных интервальных нечетких чисел предварительно необходимо перейти к нечетким числам первого порядка, с помощью формулы (2.23.). В таблице 11 представлены результаты средних значений NPV для первого и второго проектов. Графики обобщенного NPV первого типа для трех проектов представлены на рисунках 3.10., 3.11. и 3.12.

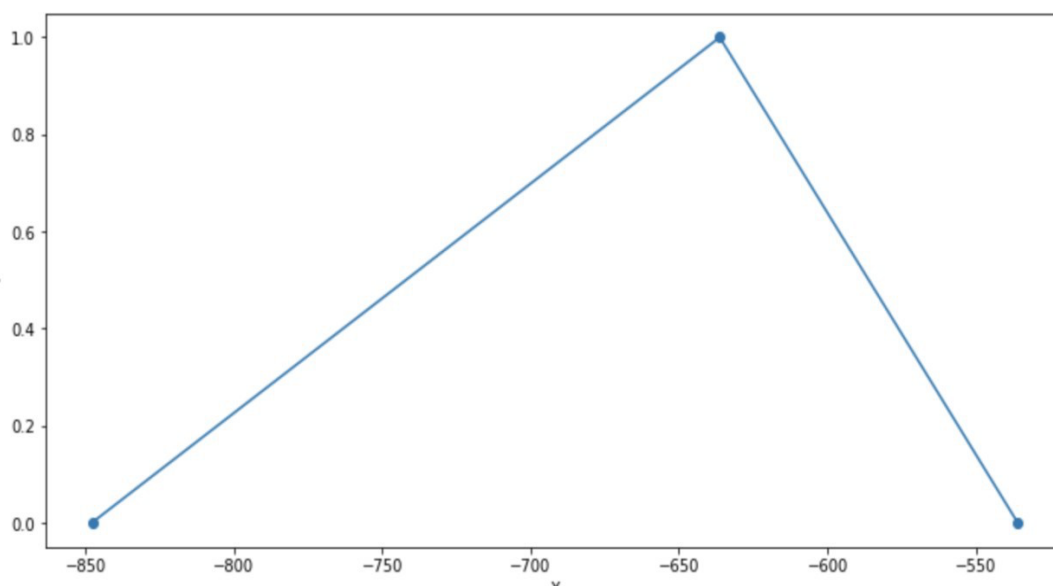


Рис.3.10. График интервального нечеткого NPV первого типа для первого проекта

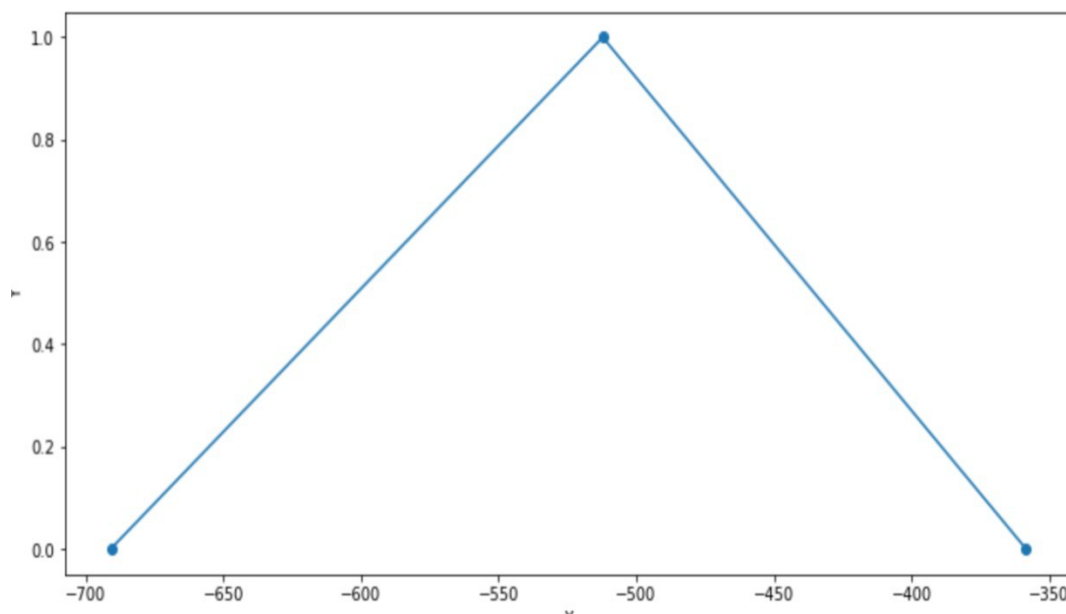


Рис.3.11. График интервального нечеткого NPV первого типа для второго проекта

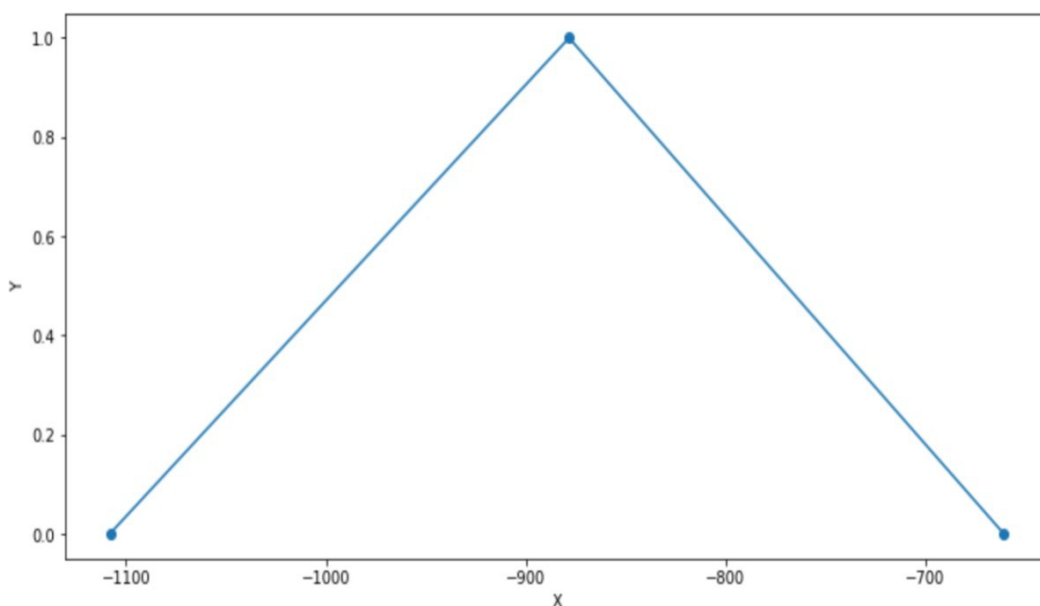


Рис.3.12. График интервального нечеткого NPV первого типа для третьего проекта

В таблицах 3.14., 3.15. и 3.16. представлены полученные опорные значения интервального нечеткого множества первого типа для каждого проекта.

Таблица.3.14.

Показатель NPV первого типа для первого проекта

	a_l^U	a_m^U	a_r^U
Опорные значения	-847.765	-636.256	-535.867

Таблица.3.15.

Показатель NPV первого типа для второго проекта

	a_l^U	a_m^U	a_r^U
Опорные значения	-690.725	-512.135	-358.448

Таблица.3.16.

Показатель NPV первого типа для третьего проекта

	a_l^U	a_m^U	a_r^U
Опорные значения	-1107.97	-878.29	-660,84

Далее, с помощью формулы (2.24), было выполнено нахождение вероятности доминирования одного проекта над другим, с последующим ранжированием по принципу от потенциально менее успешного – к потенциально более успешному. Результат работы на рисунке 3.13.

Ранжирование проектов по шансу на успех:
 Сравниваем проект №1 и проект №2
 Проект 1 успешнее, чем проект 2 на 14%

Сравниваем проект №1 и проект №3
 Проект 1 успешнее, чем проект 3 на 87%

Сравниваем проект №2 и проект №1
 Проект 2 успешнее, чем проект 1 на 86%

Сравниваем проект №2 и проект №3
 Проект 2 успешнее, чем проект 3 на 98%

Сравниваем проект №3 и проект №1
 Проект 3 успешнее, чем проект 1 на 13%

Сравниваем проект №3 и проект №2
 Проект 3 успешнее, чем проект 2 на 2%

Сортировка проектов по восходящей
 Ранг 1 Проект 3
 Ранг 2 Проект 1
 Ранг 3 Проект 2

Рис.3.13. Результат сравнительного анализа инвестиционных проектов

2.3.1 Анализ результатов

В предыдущем пункте были рассмотрены три инвестиционных проекта, для которых необходимо определить эффективность с помощью показателя NPV. В качестве входных данных использовались такие параметры как, длительность реализации, количество экспертов, участвующих в оценке доходов от проекта, ставка дисконтирования, а также первоначальные взносы.

На этапе согласованности критериев экспертов возникла ситуация, когда для повышения общей согласованности экспертов пришлось убрать из рассмотрения одного из экспертов. Такой был обнаружен во втором проекте в первый период. Таблица 3.2.1 отображает процесс исключения.

Таблица.3.17.

	<i>K1</i>	<i>K2</i>	<i>K3</i>	<i>K4</i>
<i>K1</i>	1	0.79	0.65	0.87
<i>K2</i>	0.79	1	0.47	0.79
<i>K3</i>	0.65	0.47	1	0.56

→

	<i>K1</i>	<i>K2</i>	<i>K4</i>
<i>K1</i>	1	0.79	0.87
<i>K2</i>	0.79	1	0.79
<i>K4</i>	0.87	0.79	1

Продолжение табл. 3.17.

	<i>K1</i>	<i>K2</i>	<i>K3</i>	<i>K4</i>
<i>K4</i>	0.87	0.79	0.56	1

После проведения процедуры исключения экспертных мнений, сильно отличающихся от большинства критериев, было достигнуто оптимальное значение общей согласованности между экспертами для каждого промежутка реализации. Таблица 3.18. отображает согласованность экспертных критериев для первого, второго и третьего проектов на всем сроке работы.

Таблица.3.18.

Показатель общей согласованности			
Промежуток	1 год	2 год	3 год
Первый проект	0.68	0.71	0.76
Второй проект	0.73	0.76	0.72
Третий проект	0.81	0.84	0.71

Как видно из таблицы, все показатели больше порогового значения (0.600), а значит, экспертные критерии согласованны, и их можно использовать для построения обобщенного экспертного критерия. Обобщенный экспертный критерий строится по алгоритму параграфа 2.1 для каждого года, а затем используется при расчете показателя NPV.

Для нахождения вероятности доминирования одного проекта над другим, необходимо выполнить переход к нечетким числам первого типа. Данной операции соответствует 9 шаг алгоритма из раздела 2.1. Полученный результат ранжирования для всех проектов представлен на рисунке 3.13, который дублирует рисунок 3.12 из предыдущего параграфа.

Ранжирование проектов по шансу на успех:
Сравниваем проект №1 и проект №2
Проект 1 успешнее, чем проект 2 на 14%

Сравниваем проект №1 и проект №3
Проект 1 успешнее, чем проект 3 на 87%

Сравниваем проект №2 и проект №1
Проект 2 успешнее, чем проект 1 на 86%

Сравниваем проект №2 и проект №3
Проект 2 успешнее, чем проект 3 на 98%

Сравниваем проект №3 и проект №1
Проект 3 успешнее, чем проект 1 на 13%

Сравниваем проект №3 и проект №2
Проект 3 успешнее, чем проект 2 на 2%

Сортировка проектов по восходящей
Ранг 1 Проект 3
Ранг 2 Проект 1
Ранг 3 Проект 2

Рис.3.13. Результат сравнительного анализа инвестиционных проектов

По рисунку 3.13. видно, что вероятнее всего второй проект окажется наиболее успешным среди всех представленных. Он имеет практически стопроцентный шанс на успех в сравнении с третьим проектом и, в целом, потенциально более успешный чем первый. Исходя из рисунка можно также сделать вывод о том, что первый проект, в свою очередь, довольно сильно доминирует над третьим проектом, что делает последний самым невыгодным к рассмотрению. Исходя из всего вышеперечисленного, можно утверждать, что выгоднее всего будет принять к рассмотрению второй проект с наибольшим шансом на успех в сравнении с остальными.

Заключение

В результате проведённого исследования было разработано алгоритмическое и программное обеспечение сравнительного анализа инвестиционных проектов, базирующиеся на аппарате интервальных нечетких чисел второго типа. Алгоритм позволяет обработать экспертные суждения о параметрах проекта, представленные в виде интервальных нечетких чисел первого типа, вычислить NPV проекта в виде нечеткого интервального числа второго типа и провести сравнительный анализ различных проектов по их NPV. Для сравнительного анализа используется подход вероятностного ранжирования интервальных нечетких чисел второго типа. Программная реализация представляет собой приложение, выполненное на языке Python 3 в среде разработки Anaconda Jupiter Notebook.

В ходе работы были выполнены следующие теоретические и практические задачи:

- Изучены принципы, модели и методы оценки эффективности оценки инвестиционных проектов;
- Проанализирована возможность использования аппарата нечетких интервальных чисел второго порядка для оценки эффективности инвестиционных проектов;
- Разработан алгоритм представления параметров проекта в виде интервальных нечетких чисел второго типа, базирующихся на специальной процедуре обработки экспертных оценок параметров проекта, представленных в виде обычных нечетких чисел;
- Построен алгоритм нахождения NPV проекта, базирующегося на применении арифметических операций над нечеткими интервальными числами второго типа;
- Разработан алгоритм вероятностного ранжирования альтернативных инвестиционных проектов;
- Разработано программное обеспечение по предложенным в работе алгоритмам;

➤ Проведены тестирования разработанного программного обеспечения.

Разобран непосредственно алгоритм оценки эффективности инвестиционного проекта на основе интервальных нечетких чисел второго типа, а также описаны составные части программы, реализованной на языке программирования Python 3.

Результатом программной реализации алгоритма является ранжированный список инвестиционных проектов, сформированных по принципу от потенциально более успешного к потенциально менее успешному.

Дальнейшее развитие разработанного алгоритма будет выражено в создании веб-приложения с удобным пользовательским интерфейсом. Это позволит пользователям легко взаимодействовать с функционалом программы. Также предусматривается разработка возможности загрузки экспертных оценок из файлов Excel, что ускорит и упростит процесс использования.

Список используемых источников

- 1) Chang P.-T. Ranking of Fuzzy Sets Based on the Concept of Existence / P.- T.Chang, E.S. Lee // Computers and mathematics with applications. – Elsevier, 1994. – Vol.27, pp. 1–21.
- 2) Ибрагимов В.А. Элементы нечёткой математики: [Электронный ресурс]
// URL: http://www.anl.az/el_ru/i/iv_enm.pdf
- 3) Алексеев А.В. Обработка нечёткой информации в системах принятия решений /А.В. Алексеев, А.Н. Борисов Г.В. Меркурьева и др. – М.: Радио и связь, 1989. – 304 с.
- 4) Rao P.P.B. Ranking generalized fuzzy numbers using area, mode, spreads and weight / P.P.B. Rao, N.R. Shankar // International Journal of Applied Science and Engineering. – 2012, № 10. – Vol. 1, pp. 41–57.
- 5) Abbasbandy S. Ranking fuzzy numbers using fuzzy maximizing-minimizing points / S. Abbasbandy, T. Allahviranloo, S. Salahshour // Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-1022) and LFA-2011.
– AtlantisPress, 2011. – pp.763–769.
- 6) Ahmadian A. A New Distance Measure for Trapezoidal Fuzzy Numbers / A.Ahmaidan, M.J. Ebadi, F. Bt. Ismail et al. // Mathematical Problems in Engineering, vol. 2013 – 2013. – <http://dx.doi.org/10.1155/2013/424186>
- 7) Леденева Т.М. Параметрический метод сравнения нечетких чисел / Т.М. Леденева, Д.А. Черменев, С.С. Жданова // Вестник Воронежского государственного технического университета. 2010, № 6. – Т. 6, С. 62–66.
- 8) Лебедев Г.Н. Методы решения задач управления предприятием в условиях расплывчатой неопределённости / Г.Н. Лебедев, М.Г. Матвеев, М.Е. Семёнов, О.И. Канищева // Вестник ВГУ. Сер. «Системный анализ и информационные технологии». 2012, № 1. – С.
Воронцов Я.А. Исследование свойств линейного отображения в задачах с нечёткими параметрами / Я.А. Воронцов, М.Г. Матвеев // Информатика:

проблемы, методология, технологии: материалы XIII Международной научно-методической конференции. – Воронеж, издательско-полиграфический центр Воронежского

государственного университета, 2013. – Т. 1, С. 298–304.

9) Воронцов Я.А. Влияние преобразования L на результаты арифметических операций с нечёткими LR -числами / Я.А. Воронцов, М.Г. Матвеев

// Сборник трудов международного научно-технического семинара «Современные технологии в задачах управления, автоматике и обработки информации». Алушта, 2013. – М.: Изд-во МГУПИ, 2013. – С. 11–12.

10) A. Hadi-vencheh and M. Allame, On the relation between a fuzzy number and its centroid, Computer and Mathematics with Applications 59 (2010) 3578-3582

11) M. Friedman, M. Ma and A. Kandel, Numerical solution of fuzzy differential and integral equations, Fuzzy Sets and System 106 (1999) 35-48