

МЕТОД КОГГЕРА И Ю: ОТКРЫТАЯ КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ И РЕШЕНИЕ ЗАДАЧ РАНЖИРОВАНИЯ

Пархоменко В. А.¹, старший преподаватель, ✉ vladimir.parkhomenko@spbstu.ru,
orcid.org/0000-0001-7757-377X

Лазаренко А. А.¹, студент, lazarenko2.aa@edu.spbstu.ru, orcid.org/0009-0005-1029-8095

Щукин А. В.¹, канд. техн. наук, доцент, alexander.schukin@spbstu.ru,
orcid.org/0000-0002-9534-824X

¹ Санкт-Петербургский политехнический университет Петра Великого,
ул. Политехническая, д. 29, 195251, Санкт-Петербург, Россия

Аннотация

Рассмотрен метод Коггера и Ю для вычисления весов и ранжирования альтернатив. На основе этого метода парных сравнений разработана система поддержки принятия решений с открытым исходным кодом. Представлена архитектура разработанного приложения, проведены модульные тесты, включая примеры из публикаций и специально сгенерированные случаи. Пример В. В. Подиновского, иллюстрирующий некорректность метода анализа иерархий, показал, с нашей точки зрения, приемлемый результат методом Коггера и Ю. Эксперимент подтвердился и на новой оригинальной задаче, показывая близость результатов метода Коггера и Ю и теории важности критериев. Решена задача ранжирования алгоритмов генерации уровней для компьютерных игр в двухмерном пространстве.

Ключевые слова: многокритериальные задачи принятия решений, парные сравнения, метод Коггера и Ю, теория важности критериев, DASS, СВИРЬ.

Цитирование: Пархоменко В. А., Лазаренко А. А., Щукин А. В. Метод Коггера и Ю: открытая компьютерная реализация и решение задач ранжирования // Компьютерные инструменты в образовании. 2024. № 4. С. 24–43. doi:10.32603/2071-2340-2024-4-24-43

Благодарности: Авторы статьи благодарны коллективу разработчиков СППР СВИРЬ под руководством профессора С. В. Микони за предоставление возможности решения задач в системе, а также коллективу разработчиков СППР DASS под руководством профессора В. В. Подиновского за размещение инсталлятора системы в свободном доступе.

1. ВВЕДЕНИЕ

В реальных задачах ранжирования альтернатив трудно обойтись непосредственным сущностным сравнением или сравнением по одному критерию, и тогда прибегают к решению многокритериальной задачи. В работе рассмотрены методы индивидуального принятия решений. Оригинальные постановки по методам, базирующимся на парных

сравнениях, на основе log-Чебышевского метода аппроксимации, модификаций методов парных сравнений приведены в [1] и [2], соответственно. Обзоры по индивидуальным методам принятия решений с подробным анализом их работы можно найти, например, в [2–5].

В работе рассмотрен метод Коггера и Ю [6], являющийся одной из модификаций метода аналитической иерархии Саати [7]. Оба метода направлены на ранжирование альтернатив на основе матриц парных сравнений. Если реализаций метода аналитической иерархии довольно много, например, бесплатная, многофункциональная СППР Superdecisions [8], СППР с открытым исходным кодом [9], то открытых реализаций метода Коггера и Ю найти не удалось.

Целью работы является изучение метода Коггера и Ю для нахождения приоритетов (весов) альтернатив, их последующего ранжирования, а также создание системы поддержки ранжирования альтернатив на базе этого метода.

Разработано приложение с открытым исходным кодом [10]. Эксперименты показали, что на примере, обосновывающем некорректность метода аналитической иерархии [11], получен более корректный результат, близкий к результату теории важности критериев.

Далее в разделе 2 изложена общая постановка задачи ранжирования объектов по нескольким признакам. В разделе 3 подробно описывается метод Коггера и Ю. В разделе 4 представлено проектирование системы ранжирования на основе метода Коггера и Ю. В разделе 5 представлены шаги алгоритма с псевдокодом и иллюстративными примерами расчета. В разделе 6 описываются классы системы ранжирования на основе метода Коггера и Ю. В разделе 7 описывается модульное тестирование системы. В разделе 8 описывается ручное тестирование системы, включая действия по установке приложения.

В разделе 9 сравниваются решения метода Коггера и Ю с решениями метода анализа иерархий, метода взвешенных геометрических средних, а также log-Чебышевского метода аппроксимации [12]. В разделе 10 представлено сравнение решений метода Коггера и Ю с методом анализа иерархий и теорией важности критериев [4, 11]. В разделе 11 представлено решение задачи ранжирования алгоритмов генерации уровней для компьютерных игр в двухмерном пространстве методом Коггера и Ю, рассчитанным в разработанной системе, методом аналитической иерархии, рассчитанным в СППР СВИРЬ [13–15], а также методом теории важности критериев, рассчитанным в СППР DASS [16–18].

2. ПРОБЛЕМА РАНЖИРОВАНИЯ ОБЪЕКТОВ ПО ВАЖНОСТИ

Пусть дано конечное множество объектов (признаков или альтернатив) $G = \{g_1, g_2, \dots, g_n\}$, и необходимо построить вектор $w = (w_1, w_2, \dots, w_n)$ с неотрицательными компонентами w_i такими, что $\sum_{i=1}^n w_i = 1$.

Числа w_i можно интерпретировать как весовые коэффициенты, определяющие важность объектов g_i . Основным объектом исследования в рассматриваемом методе является матрица парных сравнений. Элементы данной матрицы интерпретируются как значения коэффициента превосходства одного объекта над другим. Если получается число больше единицы, то первый объект превосходит второй объект.

Предполагается, что лицо, принимающее решения (далее — ЛПР), имеет возможность отвечать на вопросы типа «Во сколько раз объект g_i превосходит объект g_j по важности?».

Коэффициенты МПС могут выбираться из различных шкал, например, метод Саати использует следующую бальную шкалу: $\left\{ \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 4, 5, 6, 7, 8, 9 \right\}$.

Для упрощения работы пользователя можно интерпретировать значения шкалы с помощью целых чисел или вербальных оценок.

3. ОПИСАНИЕ МЕТОДА КОГГЕРА И Ю

В отличие от метода Саати [7], в методе Коггера и Ю [6] используется треугольная матрица парных сравнений (отсутствует сторона с подкреплением ответов лица, принимающего решение, то есть значения ниже главной диагонали). Сначала находится диагональная делящая матрица:

$$D_{n \times n} = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n-1 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

В последствии данная матрица обращается и в результате получается матрица $D_{n \times n}^{-1}$, в которой ненулевые i -ые элементы образуются делением единицы на значение из ячейки исходной матрицы $D_{n \times n}$ с тем же самым i -ым индексом. В методе Коггера и Ю могут использоваться матрицы парных сравнений кратности предпочтения, которая имеет следующий вид:

$$S_{3 \times 3} = \begin{pmatrix} 1 & s_{12} & s_{13} \\ 0 & 1 & s_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

В пособии [5] и работе [19] предлагают опрашивать ЛПР $n-1$ раз по шкале кратности предпочтений (таблица 1).

Таблица 1. Шкалы предпочтений с различными степенями [1]

p/i	Абсолютное превосходство	Очень сильное превосходство	Сильное превосходство	Слабое превосходство	Равнозначность	Слабая уступка	Сильная уступка	Очень сильная уступка	Абсолютная уступка
1,5	5,0625	3,375	2,25	1,5	1	0,(6)	0,(4)	0,(296)	0,(1975)
2	16	8	4	2	1	0,5	0,25	0,125	0,0625
3	81	27	9	3	1	0,(3)	0,(1)	0,0(370)	0,(0..9)

Это необходимо для того, чтобы избежать противоречий в показаниях ЛПР. Остальные показатели рассчитываются с помощью сверхтранзитивности:

$$s_{ik} = s_{ij} \cdot s_{jk}. \tag{1}$$

Пусть s_{12} , s_{23} в формуле 1 — это ответы ЛПР, тогда s_{13} выводится автоматически по замыканию. Оценки всегда располагаются выше главной диагонали. Для избежания ошибок во время работы алгоритма матрица парных сравнений заполняется снизу вверх. Следующим шагом идет подсчет суммы каждой строки матрицы $D^{-1}S$ для создания вектора сумм:

$$a = \begin{bmatrix} \sum_{i=1}^n s_{1i} \\ \sum_{i=2}^n s_{2i} \\ \dots \\ \sum_{i=j}^n s_{ji} \\ 1 \end{bmatrix}.$$

В результате с помощью двух матриц и одного вектора можно найти вес каждого объекта:

$$D^{-1}Sa = w. \quad (2)$$

Вектор весов w имеет такую же размерность, как и вектор сумм a . Данный вектор все еще не готов к использованию, так как его необходимо нормализовать, полагая $\sum_{i=1}^n w_i = 1$. Для достижения такого результата можно рассмотреть вспомогательные формулы (3) и (4), где $|w|$ во второй формуле — мощность множества (размер вектора), а значение «2» означает, что для нормализации необходимо минимум 2 числа (при меньшем количестве эта операция не имеет смысла).

$$\omega_i = \frac{w_i}{\sum_{j=1}^n w_j} \quad (3)$$

$$\omega_i = 1 - \frac{w_i}{\sum_{j=1}^n w_j} - \frac{|w| - 2}{|w|} \quad (4)$$

Формулы (3) и (4) используются, когда необходимо нормализовать значения, подлежащие максимизации (чем больше, тем лучше) или минимизации (чем меньше, тем лучше), соответственно.

Когда веса признаков w и веса альтернатив ω_i по каждому признаку i рассчитаны, можно оценить альтернативы с помощью

$$evaluation = \omega w, \quad (5)$$

где ω — матрица весов альтернатив, составленная из последовательных ω_i .

Далее спроектируем систему, позволяющую ранжировать альтернативы на основе метода Коггера и Ю.

4. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ ПОДДЕРЖКИ РАНЖИРОВАНИЯ АЛЬТЕРНАТИВ НА ОСНОВЕ МЕТОДА КОГГЕРА И Ю

На рис. 1 представлена общая схема работы программы в нотации Archimate [20].

При запуске программы пользователь попадает в основной цикл (Main loop), в котором он может выбирать действия с помощью ввода с клавиатуры.

В настройках (Settings) пользователь может выбрать отображение дополнительной информации (Additional info) для просмотра МПС, способ оценивания (для сравнения реализован метод простого выбора из пособия [5] для сравнения с методом Коггера и Ю), основание предпочтения (Preference base, таблица 1).

Диалоговая система задает вопросы ЛПП и по его ответам оценивает приоритет альтернатив.

Для реализации указанного в подразделе функционала выбран язык общего назначения Java. Систему решено реализовать в виде консольного приложения. Алгоритмическое проектирование метода Коггера и Ю представлено в следующем разделе.

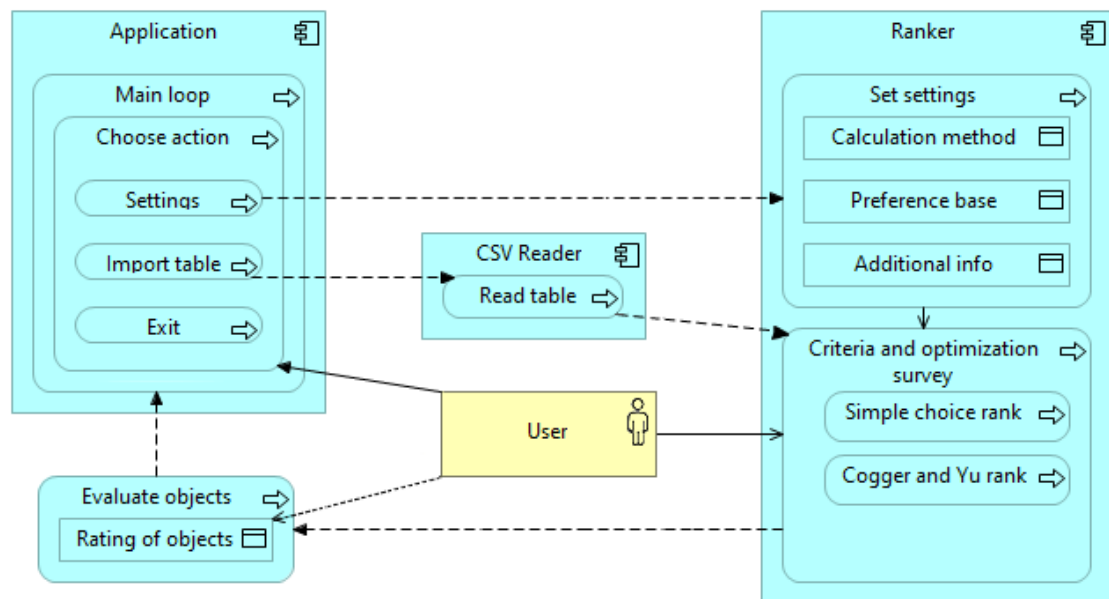


Рис. 1. Схема работы программы с реализацией метода Коггера и Ю

5. МЕТОД КОГГЕРА И Ю НА АЛГОРИТМИЧЕСКОМ УРОВНЕ

После того, как пользователь импортировал таблицу csv или txt формата, ему задают вопросы о значимости признаков по следующему алгоритму:

Алгоритм формирования матрицы значимости признаков

Вход: массив признаков таблицы M .

Выход: матрица парных сравнений S .

```

1. цикл с  $i \leftarrow 0$  до  $i \leq |M|$ , выполнять
2.     цикл с  $j \leftarrow i$  до  $j \geq 0$ , выполнять
3.         если  $j == i$  тогда
4.              $s_{ji} \leftarrow 1$ ;
5.         иначе
6.             если  $j == (i - 1)$  тогда
7.                  $buff \leftarrow$  ввод целого числа;
8.                  $s_{ji} \leftarrow p^{-buff+5}$ ;
9.             иначе
10.                 $s_{ji} \leftarrow s_{j,i-1} \cdot s_{j+1,i}$ ;
11.             $j \leftarrow j - 1$ ;
12.         $i \leftarrow i + 1$ ;
    
```

С помощью данного фрагмента кода в матрице задаются значения признаков: на диагонали 1, по соседству выше введенные пользователем значения, причем p^{-x+5} необходимо для приведения ответа к шкале предпочтений, остальные значения вычисляются транзитивно.

Рассмотрим пример: пользователь ввел для таблицы с четырьмя признаками ответы {3, 5, 7}, основание шкалы превосходства $p = 2$, тогда матрица парных сравнений строится следующим образом:

$$\begin{pmatrix} 1 & - & - & - \\ 0 & 1 & - & - \\ 0 & 0 & 1 & - \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2^{-3+5} & - & - \\ 0 & 1 & 2^{-5+5} & - \\ 0 & 0 & 1 & 2^{-7+5} \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 4 & 4 & 1 \\ 0 & 1 & 1 & 0,25 \\ 0 & 0 & 1 & 0,25 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Далее происходит умножение делящей матрицы D^{-1} на матрицу парных сравнений S . Это можно реализовать с помощью деления в следующем алгоритме:

Алгоритм произведения делящей матрицы на матрицу парных сравнений

Вход: матрица парных сравнений S .

Выход: матрица парных сравнений $D^{-1}S$.

```

1. цикл с  $i \leftarrow 0$  до  $i < S.rows$ , выполнять
2.   цикл с  $j \leftarrow i$  до  $j < S.cols$ , выполнять
3.      $s_{ij} \leftarrow \frac{s_{ij}}{S.rows-1}$ ;
4.      $j \leftarrow j + 1$ ;
5.    $i \leftarrow i + 1$ ;

```

В контексте рассматриваемого примера происходит деление каждой строки матрицы:

$$\left(\begin{array}{cccc|c} 1 & 4 & 4 & 1 & 4 \\ 0 & 1 & 1 & 0,25 & 3 \\ 0 & 0 & 1 & 0,25 & 2 \\ 0 & 0 & 0 & 1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{cccc} 0,25 & 1 & 1 & 0,25 \\ 0 & 0,33 & 0,33 & 0,08 \\ 0 & 0 & 0,5 & 0,13 \\ 0 & 0 & 0 & 1 \end{array} \right).$$

После этого рассчитывается сумма каждой строки и превращается в вектор следующим образом:

Алгоритм вычисления вектора сумм a

Вход: матрица парных сравнений $D^{-1}S$.

Выход: вектор сумм a .

```

1. цикл с  $i \leftarrow 0$  до  $i < DS.rows$ , выполнять
2.   цикл с  $j \leftarrow i$  до  $j < DS.cols$ , выполнять
3.      $a_i \leftarrow a_i + s_{ij}$ ;
4.      $j \leftarrow j + 1$ ;
5.    $i \leftarrow i + 1$ ;

```

Для рассматриваемого случая расчет вектора сумм представлен в преобразовании:

$$\left(\begin{array}{cccc} 0,25 & 1 & 1 & 0,25 \\ 0 & 0,33 & 0,33 & 0,08 \\ 0 & 0 & 0,5 & 0,13 \\ 0 & 0 & 0 & 1 \end{array} \right) \rightarrow \left(\begin{array}{c} 2,5 \\ 0,74 \\ 0,63 \\ 1 \end{array} \right).$$

После этого матрица $D^{-1}S$ умножается на вектор a , после чего получаются весовые коэффициенты w , в контексте примера следующим образом:

$$\left(\begin{array}{cccc} 0,25 & 1 & 1 & 0,25 \\ 0 & 0,33 & 0,33 & 0,08 \\ 0 & 0 & 0,5 & 0,13 \\ 0 & 0 & 0 & 1 \end{array} \right) \times \left(\begin{array}{c} 2,5 \\ 0,74 \\ 0,63 \\ 1 \end{array} \right) = \left(\begin{array}{c} 2,245 \\ 0,532 \\ 0,445 \\ 1 \end{array} \right).$$

После этого веса проходят нормализацию. Процесс нормализации представлен следующим образом:

Алгоритм нормализации вектора весов

Вход: вектор весов w .
Выход: нормализованный вектор весов ω .

1. **цикл с $i \leftarrow 0$ до $i < |w|$, выполнять**
2. $sum \leftarrow sum + w_i;$
3. $i \leftarrow i + 1;$
4. **цикл с $i \leftarrow 0$ до $i < |w|$, выполнять**
5. $\omega_i \leftarrow sum + w_i;$
6. $i \leftarrow i + 1;$

В рассматриваемом примере, получение нормализованного вектора весов признаков представлено преобразованием

$$\begin{pmatrix} 2,245 \\ 0,532 \\ 0,445 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0,53 \\ 0,13 \\ 0,1 \\ 0,24 \end{pmatrix}.$$

Аналогичным образом рассчитываются веса альтернатив по каждому признаку. Однако стоит учитывать, что при импорте таблиц значения признаков могут быть заданы не по транзитивной шкале. В таком случае свертранзитивная связь будет выполняться только для каждых трех элементов. Тогда можно использовать алгоритм, отраженный далее:

Алгоритм создания МПС при отсутствии транзитивности

Вход: массив значений признака m_k .
Выход: матрица парных сравнений S_k по признаку m_k .

1. **цикл с $i \leftarrow 0$ до $i < |m_k|$, выполнять**
2. **цикл с $j \leftarrow i$ до $j < |m_k|$, выполнять**
3. $s_{ij} \leftarrow \frac{m_{ik}}{m_{jk}};$
4. $j \leftarrow j + 1;$
5. $i \leftarrow i + 1;$

Пример. Пусть признак b имеет значения $\{65, 95, 105, 170\}$. Видно, что между соседними величинами разное относительное расстояние. В таком случае создается МПС, представленная результатом преобразования (6).

$$\begin{pmatrix} 1 & \frac{65}{95} & \frac{65}{105} & \frac{65}{170} \\ 0 & 1 & \frac{95}{105} & \frac{95}{170} \\ 0 & 0 & 1 & \frac{105}{170} \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0,68 & 0,62 & 0,38 \\ 0 & 1 & 0,9 & 0,56 \\ 0 & 0 & 1 & 0,62 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Если же попробовать посчитать МПС по первому алгоритму, то в транзитивности между элементами 65 и 170 возникнет ошибка, приведем расчет на примере (7).

$$\begin{pmatrix} 1 & \frac{65}{95} & - & - \\ 0 & 1 & \frac{95}{105} & - \\ 0 & 0 & 1 & \frac{105}{170} \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0,68 & 0,62 & 0,35 \\ 0 & 1 & 0,9 & 0,56 \\ 0 & 0 & 1 & 0,62 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Между (7) и (6) видно несоответствие между числами 0,38 и 0,35.

В силу этих причин авторы статьи рекомендуют импортировать таблицы после предварительной проверки сверхтранзитивности, вводить данные в систему вручную (с учетом верификации противоречий системой) или комбинировать оба алгоритма, например, чтобы исследовать противоречия в рассуждениях. В следующем разделе приведено проектирование классов приложения.

6. ПРОЕКТИРОВАНИЕ КЛАССОВ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Программа состоит из трех основных классов: Application, csvReader и ranker.

Класс Application запускает основной цикл и предоставляет возможность ввода для выбора действия. Пользователь может выбрать импорт таблицы, настройки и выход.

Настройки позволяют выбрать основание предпочтения (таблица 1), отображение дополнительной информации (МПС), и способ расчета оценки.

При выборе импорта таблицы пользователь вводит путь к файлу, после чего CSVReader читает файл и возвращает двумерный массив класса Object. Данный массив разделяется на массив признаков и двумерный массив альтернатив со значениями по каждому признаку. Эти данные передаются в класс Ranker.

Ranker проводит опрос пользователя по важности признаков и их направлении оптимизации.

Предполагается работа с тремя векторами: head, optimization, attributeAnswers, и одна матрица: data, которые переходят в методы SimpleRankObjects или CoggerYuRankObjects. Выбор метода зависит от выбранной настройки (по умолчанию задан CoggerYuRankObjects).

Открытая реализация программы приведена по ссылке [10]. Далее перейдем к краткому описанию функционального тестирования программы.

7. МОДУЛЬНОЕ ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для того чтобы проверить корректность работы приложения, были сформированы тестовые сценарии. Для реализации тестовых сценариев выбраны модульные тесты.

Проверяются следующие модули системы:

- произведение делящей матрицы на матрицу парных сравнений;
- нахождение вектора сумм;
- нахождение весов объектов (признаков или альтернатив);
- нормализация весов.

В тестах проверяется сходство полученных значений с ожидаемыми в окрестности $\epsilon = 0,001$ (погрешность расчета). В таблице 2 представлены входные данные и ожидаемые значения.

Для тестирования используется фреймворк JUnit. Он позволяет разработчикам писать и запускать тесты для проверки правильности работы отдельных модулей (классов, методов) в их коде. JUnit предоставляет аннотации, такие как @Test, @Before, @After и другие, которые упрощают процесс написания тестов и организацию тестового кода.

Разработанные модульные тесты показывают, что результат расчета системы совпадает с ожидаемым выходом, следовательно, есть доля уверенности, что расчеты в приложении производятся верно. Однако перед коммерческим использованием авторы рекомендуют дополнительно использовать другие техники тестирования, такие как фаззинг,

Таблица 2. Описание контрольных примеров

Тестируемый метод	Входные данные	Ожидаемый результат
Произведение $D^{-1}S$	$\begin{pmatrix} 1 & 4 & 3 & 7 \\ 0 & 1 & \frac{1}{3} & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix} [12]$	$\begin{pmatrix} \frac{1}{4} & 1 & \frac{3}{4} & \frac{7}{4} \\ 0 & \frac{1}{3} & \frac{1}{9} & 1 \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$
Расчет вектора сумм	$\begin{pmatrix} \frac{1}{4} & 1 & \frac{3}{4} & \frac{7}{4} \\ 0 & \frac{1}{3} & \frac{1}{9} & 1 \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \frac{15}{4} \\ \frac{13}{9} \\ 3 \\ 1 \end{pmatrix}$
Расчет весов	$\begin{pmatrix} \frac{1}{4} & 1 & \frac{3}{4} & \frac{7}{4} \\ 0 & \frac{1}{3} & \frac{1}{9} & 1 \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} \frac{15}{4} \\ \frac{13}{9} \\ 3 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 6,3819 \\ 1,8148 \\ 4 \\ 1 \end{pmatrix}$
Расчет ω	$\begin{pmatrix} 6,3819 \\ 1,8148 \\ 4 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0,4835 \\ 0,1375 \\ 0,3031 \\ 0,0757 \end{pmatrix}$

промышленные статические анализаторы и др. Тесты помещены в том же открытом репозитории, что и реализованное приложение [10].

Далее перейдем к описанию ручного тестирования программы.

8. РУЧНОЕ ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для начала работы с приложением необходимо скачать [10] и разархивировать приложение. Самым простым способом является использование среды разработки IntelliJ IDEA. Проект запускается из класса MCD.Main.java.

Альтернативным способом сборки является установка Maven в проект. В таком случае необходимо будет вручную составить pom.xml. То же самое можно сделать с помощью Gradle. Для этого необходимо удалить CoggerYu.iml, после чего инициализировать build.gradle.

Для работы необходимо импортировать таблицу вида (таблица 3).

Таблица 3. Вид импортируемых данных

Признак 1	Признак 2	...	Признак N
Значение ₁₁	Значение ₁₂	...	Значение _{1N}
...	...	...	...
Значение _{M1}	Значение _{M2}	...	Значение _{MN}

Важно, чтобы в названии признаков не было пробелов, также важно, чтобы все значения признаков были численными.

Для общения с диалоговой системой пользователь вводит числа, почти перед каждым вопросом пользователю показывается меню, чтобы можно было понять, какой список действий доступен.

Для начала вычислений пользователю необходимо импортировать файл расширения csv или txt, в которых присутствует заглавная строка, в качестве разделителей используется «;», а сами значения являются числовыми.

После импорта таблицы пользователь отвечает на вопросы о том, какие признаки для него наиболее важны и какие наименее важны. Также у пользователя спрашивают направление оптимизации для каждого признака. После происходит расчет всех значений, и пользователь получает оценку для каждой альтернативы. На рисунке 2 представлен пример работы с системой.

```

Введите путь к таблице CSV или TXT
C:/Test/Test.txt
Необходимо ответить на несколько вопросов, в качестве ответов принимаются
числа от 1 до 9, где |1 - абсолютно важнее |2 - очень сильно важнее
|3 - сильно важнее |4 - слабо важнее |5 - равноважен |6 - слабо уступает
|7 - сильно уступает |8 - очень сильно уступает |9 - абсолютно уступает|

Для вас цена важнее, чем количество_ядер? Насколько?
6
Для вас количество_ядер важнее, чем тактовая_частота? Насколько?
3
Для вас тактовая_частота важнее, чем TDP? Насколько?
2
Для вас предпочтительнее увеличение цена (0) или уменьшение (1)?
1
Для вас предпочтительнее увеличение количество_ядер (0) или уменьшение (1)?
0
Для вас предпочтительнее увеличение тактовая_частота (0) или уменьшение (1)?
0
Для вас предпочтительнее увеличение TDP (0) или уменьшение (1)?
1
g1 score: 0,211205
g2 score: 0,185232
g3 score: 0,201383
g4 score: 0,192981
g5 score: 0,209198
Введите номер команды:
1 - Выход
2 - Настройки
3 - Ввод таблицы

```

Рис. 2. Пример работы с диалоговой системой

Далее перейдем к решению прикладных задач методом Коггера и Ю и сравнению результатов с другими методами парных сравнений.

9. СРАВНЕНИЕ РЕЗУЛЬТАТОВ МЕТОДА КОГГЕРА И Ю С РЕШЕНИЕМ МАИ, СРЕДНЕГО ГЕОМЕТРИЧЕСКОГО И МИНМАКС log-ЧЕБЫШЕВСКОГО МЕТОДА АППРОКСИМАЦИИ

Рассмотрим пример из статьи [12]. В ней исследуется задача выбора руководителя проекта в области программирования. Кандидаты попарно оцениваются по $K = 4$ признакам: 1) персональные качества, 2) академические достижения, 3) опыт работы в команде, 4) уровень программирования. Матрица парных сравнений для метода Коггера и Ю имеет вид:

$$C = \begin{pmatrix} 1 & 4 & 3 & 7 \\ 0 & 1 & \frac{1}{3} & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$N = 4$, то есть рассмотрены кандидаты A_1, A_2, A_3, A_4 , результаты оценки которых по каждому признаку отражают следующие матрицы парных сравнений:

$$A_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 2 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 1 & 3 & 2 & 3 \\ 0 & 1 & 2 & 4 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Необходимо рассчитать веса признаков w матрицы C , а также веса альтернатив по признакам ω_i по матрицам A_1, A_2, A_3, A_4 . Строится делящая матрица D^{-1} :

$$D^{-1} = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Она будет идентичной для всех МПС в данной задаче, так как все они имеют одинаковый размер. После для каждого объекта (признаков или альтернатив по признаку) происходит умножение делящей матрицы D^{-1} на МПС объекта:

$$D^{-1}C = \begin{pmatrix} \frac{1}{4} & 1 & \frac{3}{4} & \frac{7}{4} \\ 0 & \frac{1}{3} & \frac{1}{9} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D^{-1}A_1 = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ 0 & \frac{1}{3} & 1 & \frac{2}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{6} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D^{-1}A_2 = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$D^{-1}A_3 = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{3}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D^{-1}A_4 = \begin{pmatrix} \frac{1}{4} & \frac{3}{4} & \frac{1}{2} & \frac{3}{4} \\ 0 & \frac{1}{3} & \frac{2}{3} & \frac{4}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Далее для каждой матрицы создается вектор a , элементами которого являются суммы каждой строки полученного произведения:

$$a_C = \begin{pmatrix} \frac{15}{4} \\ \frac{13}{9} \\ 3 \\ 1 \end{pmatrix}, \quad a_{A_1} = \begin{pmatrix} \frac{10}{4} \\ 2 \\ \frac{4}{6} \\ 1 \end{pmatrix}, \quad a_{A_2} = \begin{pmatrix} \frac{10}{4} \\ 2 \\ \frac{3}{2} \\ 1 \end{pmatrix}, \quad a_{A_3} = \begin{pmatrix} \frac{11}{4} \\ 2 \\ \frac{3}{2} \\ 1 \end{pmatrix}, \quad a_{A_4} = \begin{pmatrix} \frac{11}{4} \\ \frac{7}{3} \\ 1 \\ 1 \end{pmatrix}.$$

Затем необходимо умножить произведения $D^{-1}M$ на вектор a для получения весов:

$$D^{-1}Ca_C = w = \begin{pmatrix} 6,3819 \\ 1,8148 \\ 4 \\ 1 \end{pmatrix}, \quad D^{-1}A_1a_{A_1} = \omega_1 = \begin{pmatrix} 3,125 \\ 2 \\ 0,5 \\ 1 \end{pmatrix}, \quad D^{-1}A_2a_{A_2} = \omega_2 = \begin{pmatrix} 3,75 \\ 2,666 \\ 1,75 \\ 1 \end{pmatrix},$$

$$D^{-1}A_3a_{A_3} = \omega_3 = \begin{pmatrix} 3,125 \\ 2 \\ 0,5 \\ 1 \end{pmatrix}, \quad D^{-1}A_4a_{A_4} = \omega_4 = \begin{pmatrix} 3,75 \\ 2,666 \\ 1,75 \\ 1 \end{pmatrix}.$$

После этого все веса нормализуются по формуле (3) для выполнения условия $\sum_{i=1}^n w_i = 1$. Далее рассчитываются оценки по формуле (5):

$$\omega w = \begin{pmatrix} 0,4247 \\ 0,3072 \\ 0,1348 \\ 0,1330 \end{pmatrix}.$$

В статье [5] провели расчет несколькими способами.

С помощью метода Саати получаются оценки $(\omega w)^T = (0,4535; 0,2767; 0,1304; 0,1394)^T$.

С помощью метода взвешенных геометрических средних получаются оценки $(\omega w)^T = (2,1037; 1,3139; 0,5761; 0,6280)^T$.

Метод log-Чебышевской аппроксимации [5,24-26] дал результат $\min = (1; 0,4262; 0,1991; 0,3568)^T$ и $\max = (1; 0,8371; 0,2336; 0,4185)^T$.

Видно, что оценки приоритета 3 и 4 альтернатив по методу Коггера и Ю почти одинаковые и можно говорить об эквивалентности данных альтернатив по предпочтению, так как различие приоритетов альтернатив измеряется одной тысячной. Схожий результат получается МАИ, однако их ранги меняются местами. В двух оставшихся методах ранжирование альтернатив аналогично МАИ, однако различимость по значениям приоритетов больше. Это может быть полезным, если каждому объекту нужно сопоставить отдельный ранг.

10. СРАВНЕНИЕ РЕЗУЛЬТАТОВ МЕТОДА КОГГЕРА И Ю С РЕШЕНИЕМ МАИ И ТЕОРИИ ВАЖНОСТИ КРИТЕРИЕВ

Рассмотрим статью [17], где доказана некорректность метода аналитической иерархии Саати. Берется задача с двумя равноважными признаками, для них строится матрица парных сравнений:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Матрицы парных сравнений, где на диагонали и выше нее единицы, а ниже нули, в методе Коггера и Ю при любых размерах МПС дадут такой вектор весов, в котором все элементы равны. Для данной МПС $w = \begin{pmatrix} 0,5 \\ 0,5 \end{pmatrix}$.

Далее задаются две параметризованные матрицы парных сравнений альтернатив по каждому признаку:

$$F_1 = \begin{pmatrix} 1 & ab & a & 1 \\ 0 & 1 & \frac{1}{b} & \frac{1}{ab} \\ 0 & 0 & 1 & \frac{1}{a} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 1 & \frac{1}{a} & 1 & b \\ 0 & 1 & a & ab \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

При параметрах $a = 3$, $b = 3$ получаются следующие веса:

$$\omega_1 = \begin{pmatrix} 0,6045 \\ 0,0606 \\ 0,1116 \\ 0,2232 \end{pmatrix}, \quad \omega_2 = \begin{pmatrix} 0,1635 \\ 0,5420 \\ 0,2102 \\ 0,0841 \end{pmatrix},$$

после чего рассчитаем методом Коггера и Ю итоговые оценки (таблица 4).

Таблица 4. Итоговая оценка при различных параметрах a и b по методу Коггера и Ю

a	3	3	4
b	3	2	2
x_1	0,3840	0,3564	0,3768
x_2	0,3013	0,3051	0,3329
x_3	0,1609	0,1602	0,1327
x_4	0,1536	0,1781	0,1574

Основная претензия к методу Саати [7] в том, при всех параметрах из таблицы 4 получается ранжировка $x_2 > x_1 > x_4 > x_3$, хотя при расчете по теории важности критериев альтернатива x_1 должна быть лучше, чем x_2 (рис. 3).

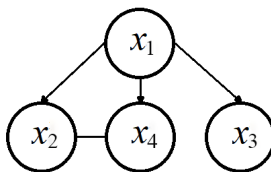


Рис. 3. Частичная упорядоченность вариантов по двум равноважным признакам [17]

Исходя из таблицы 4 методом Коггера и Ю получена ранжировка $x_1 > x_2 > x_3 > x_4$. Варианты x_3 и x_4 сменяют друг друга в полученной ранжировке в зависимости от параметров, но если считать объекты эквивалентными с погрешностью $\varepsilon = 0,02$, то данная ранжировка более соответствует результатам, полученным с помощью теории важности критериев [17], хотя x_2 и x_4 здесь по-прежнему не равноважны, а x_3 и x_4 являются сравнимыми и равноважными.

Далее перейдем к описанию задачи принятия решений, составленной авторами.

11. РЕШЕНИЕ ЗАДАЧИ РАНЖИРОВАНИЯ АЛГОРИТМОВ ГЕНЕРАЦИИ УРОВНЕЙ ДЛЯ КОМПЬЮТЕРНЫХ ИГР В ДВУХМЕРНОМ ПРОСТРАНСТВЕ

Решение задачи ранжирования алгоритмов генерации уровней для компьютерных игр проведено с целью выбора алгоритмов-кандидатов для дальнейших улучшений данных алгоритмов.

Составим таблицу 5 с оценкой альтернатив по критериям, а также укажем в ней направление оптимизации. Признаки: 1) повторяемость (количество паттернов в уровне), 2) управляемость (количество параметров, которые могут управлять алгоритмом), 3) удобство использования (как много усилий должен приложить пользователь, чтобы получить желаемую карту), порядковая шкала, 4) сложность реализации (количество

используемых библиотек, концептуальная сложность алгоритмов), порядковая шкала. Значения альтернатив представлены в таблице 5, где g_1 — IORand [21], g_2 — GB WFC [22], g_3 — PBDG [23], g_4 — WASPAS [24], g_5 — PCGQD [25], g_6 — RBCAA [26], g_7 — PSOCG [27]. Стрелки $\uparrow$ и $\downarrow$ означают, соответственно, максимизируется или минимизируется признак. В зависимости от этого будет производиться нормализация в соответствие с формулами (3) или (4).

Таблица 5. Оценка алгоритмов процедурной генерации игровых уровней

N	Повторяемость $\downarrow$	Управляемость $\uparrow$	Удобство использования $\uparrow$	Сложность реализации $\downarrow$
g_1	35	4	3	3
g_2	20	4	3	2
g_3	72	3	1	1
g_4	24	3	2	1
g_5	33	2	2	2
g_6	40	1	3	1
g_7	55	1	2	3

Важность признаков задается экспертом: (10, 8, 6, 5). Все матрицы парных сравнений строятся по алгоритму создания МПС при отсутствии транзитивности:

$$S = \begin{pmatrix} 1 & \frac{10}{8} & \frac{10}{6} & \frac{1}{5} \\ 0 & 1 & \frac{8}{6} & \frac{1}{5} \\ 0 & 0 & 1 & \frac{6}{5} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad S_1 = \begin{pmatrix} 1 & \frac{35}{20} & \frac{35}{72} & \frac{35}{24} & \frac{35}{33} & \frac{35}{40} & \frac{35}{55} \\ 0 & 1 & \frac{20}{72} & \frac{20}{24} & \frac{20}{33} & \frac{20}{40} & \frac{20}{55} \\ 0 & 0 & 1 & 3 & \frac{72}{33} & \frac{72}{40} & \frac{72}{55} \\ 0 & 0 & 0 & 1 & \frac{33}{40} & \frac{40}{55} & \frac{33}{55} \\ 0 & 0 & 0 & 0 & 1 & \frac{40}{33} & \frac{40}{55} \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{55}{40} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$S_2 = \begin{pmatrix} 1 & 1 & \frac{4}{3} & \frac{4}{3} & 2 & 4 & 4 \\ 0 & 1 & \frac{3}{4} & \frac{3}{4} & 2 & 4 & 4 \\ 0 & 0 & 1 & 1 & \frac{3}{2} & 3 & 3 \\ 0 & 0 & 0 & 1 & \frac{3}{2} & 3 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad S_3 = \begin{pmatrix} 1 & 1 & 3 & \frac{3}{2} & \frac{3}{2} & 1 & \frac{3}{2} \\ 0 & 1 & 3 & \frac{3}{2} & \frac{3}{2} & 1 & \frac{3}{2} \\ 0 & 0 & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 1 & \frac{2}{3} & 1 \\ 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad S_4 = \begin{pmatrix} 1 & \frac{3}{2} & 3 & 3 & \frac{3}{2} & 3 & 1 \\ 0 & 1 & 2 & 2 & 1 & 2 & \frac{2}{3} \\ 0 & 0 & 1 & 1 & \frac{1}{2} & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 1 & \frac{1}{2} & 1 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 2 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Для упрощения расчетов можно не создавать делящую матрицу D^{-1} , а поделить строки матриц согласно алгоритму произведения делящей матрицы на матрицу парных сравнений. Из произведения делящей матрицы на МПС получаются векторы сумм:

$$a = \begin{pmatrix} 1,4791 \\ 1,3111 \\ 1,1 \\ 1 \end{pmatrix}, \quad a_1 = \begin{pmatrix} 1,038 \\ 0,5968 \\ 1,8581 \\ 0,6909 \\ 0,8083 \\ 0,8636 \\ 1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 2,0952 \\ 2,2777 \\ 1,9 \\ 2,125 \\ 1,6666 \\ 1 \\ 1 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1,5 \\ 1,5833 \\ 0,5666 \\ 0,9166 \\ 0,8888 \\ 1,25 \\ 1 \end{pmatrix}, \quad a_4 = \begin{pmatrix} 2 \\ 1,4444 \\ 0,7666 \\ 0,7083 \\ 1,2222 \\ 0,6666 \\ 1 \end{pmatrix}.$$

С помощью данных векторов вычисляются соответственные веса, которые проходят нормализацию (формула (2), затем (3) или (4)):

$$w = \begin{pmatrix} 0,3250 \\ 0,2729 \\ 0,2150 \\ 0,1870 \end{pmatrix}, \quad \omega_1 = \begin{pmatrix} 0,1409 \\ 0,2052 \\ 0,0078 \\ 0,1950 \\ 0,1709 \\ 0,1565 \\ 0,1233 \end{pmatrix}, \quad \omega_2 = \begin{pmatrix} 0,1977 \\ 0,2077 \\ 0,1645 \\ 0,1744 \\ 0,1240 \\ 0,0656 \\ 0,0656 \end{pmatrix}, \quad \omega_3 = \begin{pmatrix} 0,1927 \\ 0,1919 \\ 0,0628 \\ 0,1198 \\ 0,1195 \\ 0,1811 \\ 0,1317 \end{pmatrix}, \quad \omega_4 = \begin{pmatrix} 0,0100 \\ 0,1032 \\ 0,1970 \\ 0,2023 \\ 0,1313 \\ 0,2138 \\ 0,1420 \end{pmatrix}.$$

После этого происходит расчет итоговой оценки по формуле (5): $\omega w = (0,1431; 0,1840; 0,0978; 0,1746; 0,1397; 0,1477; 0,1129)^T$.

Получена ранжировка $g_2 > g_4 > g_6 > g_1 > g_5 > g_7 > g_3$. Данный пример показывает, как с помощью одной таблицы с вариантами и оценки признаков получить оценку альтернатив. Также данный пример учитывает разное направление оптимизации признаков.

Для сравнения эту же задачу решим в системе поддержки принятия решений СВІРЬ [13–15] с помощью метода анализа иерархий. Получен вектор приоритетов объектов $(0,15; 0,17; 0,13; 0,16; 0,14; 0,14; 0,11)^T$, в результате чего получена ранжировка $g_2 > g_4 > g_1 > g_5 > g_6 > g_3 > g_7$. Видно, что по сравнению с методом Коггера и Ю часть рейтинга, начиная с третьего места сильно меняется.

Теперь можно воспользоваться системой поддержки принятия решений DASS [11, 16, 18], которая использует методы из теории важности критериев. Для начала необходимо привести шкалы таблицы 5 к единой шкале, а также сделать их сонаправленными. Для признака «повторяемость» берется четыре интервала: $\{[100; 75], [75; 50], [50; 25], [25; 0]\}$, по которым признак загрубляется. В признаке «сложность разработки» числа 1 и 3 меняют друг друга местами (таблица 6).

Таблица 6. Оценка алгоритмов процедурной генерации игровых уровней в единой шкале

N	Повторяемость	Управляемость	Удобство использования	Сложность реализации
g_1	3	4	3	1
g_2	4	4	3	2
g_3	2	3	1	3
g_4	4	3	2	3
g_5	3	2	2	2
g_6	3	1	3	3
g_7	2	1	2	1

Произведем расчет методом Коггера и Ю, а также с помощью DASS и СВІРЬ по этим данным. В таблице 7 приведены значения оценок альтернатив методом Коггера и Ю, DASS и СВІРЬ.

Методом Коггера и Ю получена ранжировка $g_2 > g_4 > g_6 \equiv g_1 > g_5 > g_3 > g_7$. Прочерки во второй колонке означают, что альтернативы были отброшены по Парето. Оставшиеся g_2, g_4, g_6 являются недоминируемыми. Если уточнять оценку встроенным средством DASS (авторам статьи не удалось найти математическое описание «уточнения»), то получается ранжировка $g_2 > g_4 > g_6$, что коррелирует с результатом, полученным методом

Таблица 7. Значения приоритетов сущностей по методу Коггера и Ю, теории важности критериев, рассчитанные в СППР DASS [16] и по МАИ в СППР СВирЬ [13]

G	Расчеты по Коггеру и Ю	Расчеты по теории важности критериев с помощью DASS	Расчеты по методу анализа иерархий с помощью СВирЬ
g_1	0,15	—	0,1609
g_2	0,18	0,657	0,1888
g_3	0,12	—	0,1262
g_4	0,17	0,248	0,1720
g_5	0,13	—	0,1288
g_6	0,15	0,094	0,1379
g_7	0,10	—	0,0855

Коггера и Ю. Отметим, что расчет по методу Коггера и Ю в суженном пространстве по Порето может также привести к большей различимости альтернатив по приоритетам.

Видно, что ранжировки методом Коггера и Ю по исходной таблице и таблице в единой шкале почти полностью совпадают, что может косвенно означать, что построенная единая порядковая шкала корректна. Методом анализа иерархий получена ранжировка $g_2 > g_4 > g_1 > g_6 > g_5 > g_3 > g_7$. Результат МАИ немного отклоняется от решений, полученных методом Коггера и Ю, меняется приоритет первой и шестой альтернатив.

Все три расчета подтверждают перспективность использования алгоритмов GB WFC [22] — g_2 и WASPAS [24] — g_4 для дальнейшей оптимизации в рамках генерации игровых уровней.

12. ЗАКЛЮЧЕНИЕ

В ходе исследования подробно проанализирован метод Коггера и Ю. Разработана соответствующая СППР с открытым исходным кодом [10], которая может быть использовано в рамках принятия решений в образовательных и иных целях.

Проведено пошаговое вычисление оценок альтернатив по статьям [12] и [17]. Выявлено, что ранжировка методом Коггера и Ю может отличаться от метода Саати, метода взвешенных геометрических средних и метода лог-Чебышевской аппроксимации [12]. Последние два метода показали более сильную различимость оценок приоритетов на рассмотренном примере, измеримую десятymi и сотыми долями целого числа вместо тысячных значений у методов Коггера и Ю и Саати. Последнее означает фактическую эквивалентность альтернатив по рангам.

Проведены расчёты рангов альтернатив на примере из статьи [17]. Выявлено, что метод Коггера и Ю показывает более согласованные результаты с теорией важности критериев, дополнительно сужая множество Парето, но без таких противоречий, которые указаны в [17] в рамках критики МАИ. На указанном примере это подтверждено с точностью до двух сотых.

В проведенном исследовании сформирована оригинальная задача ранжирования альтернатив — алгоритмов генерации уровней для компьютерных игр в двухмерном пространстве. Оно проведено с целью выбора алгоритмов-кандидатов для дальнейших улучшений данных алгоритмов. В результате решения задачи результаты рейтингов, полученных различными методами, показывают «уверенное» первое место у GB WFC [22], а второе — у WASPAS [24]. Данные алгоритмы выбраны для дальнейших улучшений.

В качестве дальнейших шагов авторы планируют проверить появившуюся в рамках данного исследования гипотезу: снижение противоречий в решении методом Коггера и Ю по сравнению с МАИ достигается, прежде всего, тем, что в методе расчета Коггера и Ю по построению снижается противоречивость в матрице парных сравнений.

Список литературы

1. Krivulin N. Rating alternatives from pairwise comparisons by solving tropical optimization problems // Proc. of 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, 2015. P. 162–167. doi:10.1109/FSKD.2015.7381933
2. Микони С. В. Теория принятия управленческих решений: учебное пособие. СПб.: Издательство «Лань», 2022. 384 с.
3. Петровский А. Б. Теория принятия решений: учебник для ВУЗов. М.: Издательский центр «Академия», 2009. 400 с.
4. Подиновский В. В. Многокритериальные задачи принятия решений: теория и методы анализа: учебник для вузов. М.: Издательство Юрайт, 2025. 486 с.
5. Черноруцкий И. Г. Методы принятия решения [Электронный ресурс]: учебное пособие; Санкт-Петербургский государственный политехнический университет [Электронный ресурс]. СПб., 2012. URL: <http://elibr.spbstu.ru/dl/2358.pdf> (дата обращения: 22.12.2024).
6. Cogger K. O., Yu P. L. Eigenweight vectors and least-distance approximation for revealed preference in pairwise weight ratios // Journal of optimization theory and applications. 1985. Vol. 46. P. 483–491.
7. Saaty T. L. A scaling method for priorities in hierarchical structures // Journal of mathematical psychology. 1977. Vol. 15, №. 3. P. 234–281.
8. СППР Superdecisions [Электронный ресурс] URL: <https://www.superdecisions.com/> (дата обращения: 22.12.2024).
9. Салимгареев И. А. АНР [Электронный ресурс]. URL: <https://github.com/DM-app-tools/ANP> (дата обращения: 22.12.2024).
10. Пархоменко В. А., Лазаренко А. А. Система поддержки ранжирования методом Коггера и Ю [Электронный ресурс]. URL: <https://github.com/DM-app-tools/CoggerYuDecisionHelper> (дата обращения: 22.12.2024).
11. Подиновский В. В., Подиновская О. В. О некорректности метода анализа иерархий // Проблемы управления. 2011. №. 1. С. 8–13.
12. Кривулин Н. К. и др. Решение многокритериальных задач оценки альтернатив на основе парных сравнений // Компьютерные инструменты в образовании. 2024. №. 2. С. 5–29. doi:10.32603/2071-2340-2024-2-5-29
13. СППР СВИРЬ [Электронный ресурс]. URL: <http://www.mcd-svir.ru/> (дата обращения: 22.12.2024).
14. Микони С. В. Система выбора и ранжирования «СВИРЬ» // Труды международного конгресса «Искусственный интеллект в XXI веке». Дивноморское 3–8.09.2001. М., 2001. Т. 1. С. 500–507.
15. Микони С. В. Бураков Д. П. Гарина М. И. Инструментальная система для выбора задач многокритериального выбора // Программные продукты и системы: научн. и научно-практич. изд. Тверь, 2009. № 4 (88). С. 6–9.
16. СППР DASS [Электронный ресурс]. URL: <http://mcdm.ru/soft/dass/> (дата обращения: 22.12.2024).
17. Подиновский В. В., Потапов М. А. Теоретические основы и системы поддержки принятия многокритериальных решений // Материалы XXXIV Международной конференции «Информационные технологии в науке, образовании, телекоммуникации и бизнесе» (20–30 мая 2007 г., Гурзуф, Украина). Приложение к журналу «Открытое образование». 2007. С. 87–89.
18. Подиновский В. В. Анализ задач многокритериального выбора методами теории важности критериев при помощи компьютерных систем поддержки принятия решений // Известия Российской академии наук. Теория и системы управления. 2008. № 2. С. 64–68.
19. Соколова О. А., Черноруцкий И. Г. Программная реализация диалоговой системы выбора вариантов на основе метода Коггера и Ю // Неделя науки СПбПУ. 2019. С. 64–67.
20. Инструмент для формализации Archimate [Электронный ресурс]. URL: <https://www.archimatetool.com/> (дата обращения: 22.12.2024).
21. Moreno-Armendáriz M. A. et al. IORand: A Procedural Videogame Level Generator Based on a Hybrid PCG Algorithm // Applied Sciences. 2022. Vol. 12, №. 8. P. 3792.

22. Kim H. et al. Graph based wave function collapse algorithm for procedural content generation in games // IEICE TRANSACTIONS on Information and Systems. 2020. Vol. 103, №. 8. P. 1901–1910.
23. Bagus Harisa A., Tai W. K. Pacing-based Procedural Dungeon Level Generation: Alternating Level Creation to Meet Designer's Expectations // International Journal of Computing and Digital Systems. 2022. Vol. 12, №. 1. P. 401–416.
24. Petrovas A., Bausys R. Procedural video game scene generation by genetic and neutrosophic WASPAS algorithms // Applied Sciences. 2022. Vol. 12, №. 2. P. 772.
25. Gravina D. et al. Procedural content generation through quality diversity // 2019 IEEE Conference on Games (CoG). IEEE, 2019. P. 1–8.
26. Gellel A., Sweetser P. A hybrid approach to procedural generation of roguelike video game levels // Proc. of the 15th Int. Conf. on the Foundations of Digital Games. 2020. P. 1–10.
27. De Pontes R. G., Gomes H. M., Seabra I. S. R. Particle swarm optimization for procedural content generation in an endless platform game // Entertainment Computing. 2022. Vol. 43. P. 100496.

Поступила в редакцию 08.12.2024, окончательный вариант — 22.12.2024.

Пархоменко Владимир Андреевич, старший преподаватель Высшей школы программной инженерии Института компьютерных технологий и кибербезопасности, СПбГПУ, Санкт-Петербург, ✉ vladimir.parkhomenko@spbstu.ru

Лазаренко Артем Андреевич, студент магистратуры Высшей школы программной инженерии Института компьютерных технологий и кибербезопасности, СПбГПУ, Санкт-Петербург, lazarenko2.aa@edu.spbstu.ru

Щукин Александр Валентинович, канд. техн. наук, доцент Высшей школы программной инженерии Института компьютерных технологий и кибербезопасности, СПбГПУ, Санкт-Петербург, alexander.schukin@spbstu.ru

Computer tools in education, 2024

№ 4: 24–43

<http://cte.eltech.ru>

[doi:10.32603/2071-2340-2024-4-24-43](https://doi.org/10.32603/2071-2340-2024-4-24-43)

Cogger and Yu Method: an Open Computer Implementation and Solution of Ranking Problems

Parkhomenko V. A.¹, Senior Lecturer, ✉ vladimir.parkhomenko@spbstu.ru,
orcid.org/0000-0001-7757-377X

Lazarenko A. A.¹, Magister, lazarenko2.aa@edu.spbstu.ru, orcid.org/0009-0005-1029-8095

Schukin A. V.¹, Cand. Sc., Associate Professor, alexander.schukin@spbstu.ru,
orcid.org/0000-0002-9534-824X

¹Peter the Great St. Petersburg Polytechnic University,
29 Politekhnikeskaya str., 195251, Saint Petersburg, Russia

Abstract

Cogger and Yu method for evaluation weights and ranking alternatives is considered. Based on this pairwise comparison method, an open source decision support system is developed. The architecture of the developed application is presented, unit tests are

conducted, including examples from publications and specially generated test cases. An example by V.V. Podinovsky, illustrating the incorrectness of hierarchy analysis method, showed, from our point of view, an acceptable result by Cogger and Yu method. The experiment is confirmed on a new original problem, showing the closeness of the results of Cogger and Yu method and criteria importance theory. The problem of ranking level generation algorithms for computer games in two-dimensional space is solved.

Keywords: multicriteria decision-making problems, paired comparisons, Cogger and Yu method, criteria importance theory, DASS, SVIR.

Citation: V. A. Parkhomenko, A. A. Lazarenko, and A. V. Schukin, "Cogger and Yu Method: an Open Computer Implementation and Solution of Ranking Problems," *Computer tools in education*, no. 4, pp. 24–43, 2024 (in Russian); doi:10.32603/2071-2340-2024-4-24-43

Acknowledgements: The paper authors are grateful to the DSS SVIR team of developers under the leadership of Professor S.V. Mikoni for providing the opportunity to solve problems in the system, as well as to the DSS DASS team of developers of under the leadership of Professor V.V. Podinovsky for making the system installer freely available.

References

1. N. Krivulin, "Rating alternatives from pairwise comparisons by solving tropical optimization problems," in *Proc. of 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015, pp. 162–167; doi:10.1109/FSKD.2015.7381933
2. S. V. Mikoni, "Theory of Management Decision-Making: a textbook," St. Petersburg, Russia: Lan Publishing House, 2022 (in Russian).
3. A. B. Petrovsky, *Theory of Decision-Making: a textbook for universities*, Moscow: Academy Publishing Center, 2009 (in Russian).
4. V. V. Podinovsky, *Multicriteria Decision-Making Problems: Theory and Methods of Analysis: a textbook for universities*, Moscow: Yurait Publishing House, 2025 (in Russian).
5. I. G. Chernorutsky, "Methods of Decision-Making," in *elib.spbstu.ru*, Saint Petersburg State Polytechnical University. St. Petersburg, 2012 (in Russian). [Online]. Available: <http://elib.spbstu.ru/dl/2358.pdf>.
6. K. O. Cogger and P. L. Yu, "Eigenweight vectors and least-distance approximation for revealed preference in pairwise weight ratios," *Journal of optimization theory and applications*, vol. 46, pp. 483–491, 1985.
7. T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of mathematical psychology*, vol. 15, no. 3, pp. 234–281, 1977.
8. Creative Decisions found., "Superdecisions DSS," in *superdecisions.com*, 2024. [Online]. Available: <https://www.superdecisions.com/>
9. I. A. Salimgareev, "AHP," in *github.com*, 2022. [Online]. Available: <https://github.com/DM-app-tools/AHP>
10. V. A. Parkhomenko and A. A. Lazarenko, "Ranking support system using the Cogger and Yu method," in *github.com*, 2024. [Online]. Available: <https://github.com/DM-app-tools/CoggerYuDecisionHelper>
11. V. V. Podinovsky and O. V. Podinovskaya, "On the incorrectness of the method hierarchy process analysis," *Problems of Control Sciences*, no. 1, pp. 8–13, 2011 (in Russian).
12. N. K. Krivulin et al., "Solution of multicriteria problems of alternatives evaluation based on paired comparisons," *Computer tools in education*, no. 2, pp. 5–29, 2024 (in Russian); doi:10.32603/2071-2340-2024-2-5-29
13. D. P. Burakov and S. V. Mikoni, *SVIR' DSS*, in *mcd-svir.ru*, 2009. [Online] (in Russian). Available: <http://www.mcd-svir.ru/>
14. S. V. Mikoni, "SVIR' selection and ranking system," in *Proc. of the international congress "Artificial Intelligence in the 21st century"*. Divnomorskoye 3-8.09.2001, vol. 1, pp. 500–507, 2001 (in Russian).
15. S. V. Mikoni, D. P. Burakov, and M. I. Garina, "Instrumental system for selecting multi-criteria choice problems," *Software products and systems: scientific and scientific-practical*, no. 4 (88), pp. 6–9, 2009 (in Russian).
16. "The decision support system DASS," in *mcodm.ru*, 2019. [Online]. Available: <http://mcodm.ru/soft/dass/>
17. V. V. Podinovsky and M. A. Potapov, "Theoretical foundations and systems for supporting multicriteria decisions," in *Proc. of the XXXIV International Conference "Information Technologies in Science, Education, Telecommunications and Business" (May 20-30, 2007, Gurzuf, Ukraine), Supplement to the journal "Open Education"*, pp. 87–89, 2007 (in Russian).
18. V. V. Podinovsky, "Analysis of multicriteria choice problems by methods of the theory of criteria importance using computer systems for supporting decision making," *Bulletin of the Russian Academy of Sciences. Theory and Control Systems*, no. 2, pp. 64–68, 2008 (in Russian).

19. O. A. Sokolova and I. G. Chernorutsky, "Software implementation of a dialog system for selecting options based on the Cogger and Yu method," *SPbPU Science Week*, pp. 64–67, 2019 (in Russian).
20. P. Beauvoir and J.-B. Sarrodie, "Archimate formalization tool," in *archimatetool.com*, 2024. [Online]. Available: <https://www.archimatetool.com/>
21. M. A. Moreno-Armendáriz et al., "IORand: A Procedural Videogame Level Generator Based on a Hybrid PCG Algorithm," *Applied Sciences*, vol. 12, no. 8, pp. 3792, 2022.
22. H. Kim et al., "Graph based wave function collapse algorithm for procedural content generation in games," *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 8, pp. 1901–1910, 2020.
23. A. Bagus Harisa and W. K. Tai, "Pacing-based Procedural Dungeon Level Generation: Alternating Level Creation to Meet Designer's Expectations," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 401–416, 2022.
24. A. Petrovas and R. Bausys, "Procedural video game scene generation by genetic and neutrosophic WASPAS algorithms," *Applied Sciences*, vol. 12, no. 2, p. 772, 2022.
25. Gravina D. et al., "Procedural content generation through quality diversity," *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019.
26. A. Gellel and P. Sweetser, "A hybrid approach to procedural generation of roguelike video game levels," in *Proc. of the 15th Int. Conf. on the Foundations of Digital Games*, pp. 1–10, 2020.
27. R. G. De Pontes, H. M. Gomes, and I. S. R. Seabra, "Particle swarm optimization for procedural content generation in an endless platform game," *Entertainment Computing*, vol. 43, pp. 100496, 2022.

Received 08-12-2024, the final version — 22-12-2024.

Vladimir Parkhomenko, Senior Lecturer of Higher School of Software Engineering, Institute of Computer Science and Cybersecurity, SPbPU, Saint Petersburg,
✉ vladimir.parkhomenko@spbstu.ru

Artem Lazarenko, Magister Student of Higher School of Software Engineering, Institute of Computer Science and Cybersecurity, SPbPU, Saint Petersburg, lazarenko2.aa@edu.spbstu.ru

Alexander Schukin, Candidate of Sciences (Tech.), associate professor of Higher School of Software Engineering, Institute of Computer Science and Cybersecurity, SPbPU, Saint Petersburg,
alexander.schukin@spbstu.ru