УДК 004.832.2 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait.2021.2/3510
Поступила в редакцию 28.05.2021
Подписана в печать 19.07.2021

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОСТРОЕНИЯ АГРЕГИРОВАННЫХ РЕЙТИНГОВ НА ОСНОВЕ МЕТОДА ПОРОГОВОГО АГРЕГИРОВАНИЯ

© 2021 С. В. Разумников✉

Юргинский технологический институт (филиал)
Национального исследовательского Томского политехнического университета (ЮТИ ТПУ)
ул. Ленинградская, 26, 652055 Юрга, Российская Федерация

Аннотация. Проблема выбора наилучшей альтернативы является одной из важных направлений в теории принятия решений. Как правило, при выборе альтернативы оцениваются по множеству критериев, и формируется проранжированный вектор, по которому уже определяется лучшая. В случае важного выбора необходимо учитывать некомпенсаторный характер значений критериев. Для избегания проблемы с компенсацией предлагается формировать агрегированный рейтинг с применением правила порогового агрегирования. В статье представлена модель некомпенсаторного агрегирования для построения рейтингов, в основе которой лежит правило порогового агрегирования, применяемого в задачах многокритериального оценивания. Использование данного метода не позволит компенсировать низкие оценки экспертов другими более высокими оценками по другим критериям. Также представлена схема этапов оценки по данной модели и алгоритм для разработки программного обеспечения «Формирование агрегированного рейтинга». В алгоритме описаны используемые функции расчетов и приведен листинг кода. Представленная модель была запрограммирована на языке C # в среде *Visual Studio 2019* в виде Windows приложения. Данная программа позволяет вести удобный и быстрый расчет индекса предпочтений и строить проранжированный агрегированный рейтинг, в том числе в виде графика, а также сохранять информацию в базу данных. Приведен пример расчета в программе для построения рейтинга поставщиков облачных услуг. Программа универсальна и может быть использована для построения обобщенного рейтинга оцениваемых альтернатив в любой области, а также для принятия решений при выборе лучшей альтернативы.

Ключевые слова: пороговое агрегирование, модель, алгоритм, схема, программа, листинг, рейтинг, критерии, альтернативы, градации.

ВВЕДЕНИЕ

При принятии решений о выборе какой-либо альтернативы необходимо узнать, какой из рассматриваемых вариантов будет лучше [1–3]. Здесь важно установить критерии, по которым будет производиться оценка, и определиться со шкалой [4, 5]. Таким образом, встает задача многокритериального принятия решений, в который необходимо применить системный подход и учесть все нюансы. В итоге мы должны получить лучший вариант решения поставленной задачи, который будет агрегирован по всем оцениваемым параметрам [6–8].

При выборе также следует уделить внимание некомпенсаторному характеру значений критериев. Это необходимо для того, чтобы сильные стороны по одному критерию не перекрывали слабые стороны по другому критерию, так как низкое значения одного параметра может сильно повлиять на результат работы. Учесть эти нюансы позволяет метод

✉ Разумников Сергей Викторович
e-mail: demolove7@inbox.ru

порогового агрегирования, используя который, можно построить рейтинг, который будет отражать сравнительную значимость различных альтернатив для предприятия. Использование данного метода не позволит компенсировать низкие оценки экспертов другими более высокими оценками по другим критериям [9]. Благодаря этому свойству уменьшится возможность принудительного улучшения альтернативы своих значений в обобщенном интегральном показателе [10].

ЛПР часто не осознает, насколько важно составить список альтернатив. В итоге может быть выбрана из рассматриваемых альтернатив не самая лучшая. В таком случае качество выбора будет ограничено качеством оцениваемых альтернатив. Более полный список имеющихся альтернатив окажет большую помощь в принятии решений. Составление списка — является неотъемлемой частью процесса принятия решений. Когда имеется неполный список альтернатив или он не продуман, то принять решение невозможно. Однако если альтернативы перечислены четко, то мы уже имеем конкретную поставленную задачу выбора одной из перечисленных альтернатив [11]. Для получения такого списка формируется рейтинг альтернатив. После выполненного анализа и оценки, можно составить проранжированный рейтинг, по которому уже будет четко видно лучшие варианты для принятия решений [12].

1. ПОСТАНОВКА ЗАДАЧИ

В статье представлена математическая модель для построения рейтинга и, соответственно, выбора лучшей альтернативы на основе метода порогового агрегирования.

Использование такой модели очень трудоемко. Если необходимо построить рейтинг большого количества альтернатив, оценивать их по многим критериям, при этом использовать градационные ранжировки больше четырех, то такое вычисление может занять целый день. Разработка программного обеспечения, в котором будет запрограммирована модель, и будет иметься возможность сохранять необходимые данные, в разы сократит время работы с моделью, избежав при этом изнуряющих процедур расчета. Тем более, при ручном выполнении имеет место быть человеческий фактор, можно упустить данные или неправильно произвести вычисление.

Для программирования предложенной модели был выбран язык C#, который является объектно-ориентированным, простым, и в то же время очень мощным языком программирования [13]. C# имеет возможность создать программное приложение в Windows Forms (в виде Windows приложения) с удобным и понятным интерфейсом для пользователя.

Целью данной работы является разработка алгоритма и программного обеспечения на основе математической модели некомпенсаторного порогового агрегирования, позволяющей строить рейтинг оцениваемых альтернатив.

2. МАТЕРИАЛЫ И МЕТОДЫ

2.1. Модель порогового агрегирования

Рассмотрим математическую модель для построения агрегированного рейтинга [14]. Пусть k — количество альтернатив, которые необходимо оценить по n критериям, m — число градаций. В качестве альтернатив могут выступать различные субъекты или объекты, например сотрудники, поставщики или проекты.

Оцениваться альтернативы могут по выбранной градационной шкале, которая может быть от двух и выше. Часто применяется пяти градационная шкала, так как это привычная для эксперта шкала оценивания (например, в школах и университетах). Также удобна 10-ти балльная шкала для разнесения по градациям. Более 10 использовать не уместно, хотя и возможно. Будем придерживаться десяти градационной шкалы для создания программного обеспечения при построении агрегированных рейтингов.

Для каждой оцениваемой альтернативы x из k сформируем вектор $(x_1,...x_n)$, где x_i — это ранг альтернативы по заданному критерию n, т. е. $x_i \in \{1,...,m\}$, $i = 1,..,m$.

С использованием градационной шкалы эксперт выставляет оценки, далее множество k ранжируется, т. е. для альтернативы находим порядковый номер в рейтинге.

Для каждого множества $x_j \in k$ записываем $x = (x_1, ..., x_n)$. Таким образом, множество k представится в виде $(x_1, ..., x_n)$.

Для формирования рейтинга применим метод порогового агрегирования [15] и рассчитаем индексы предпочтения для каждой альтернативы.

Правило порогового агрегирования заключается в том, что мы находим количество каждой оценки, сначала единиц, потом двоек, то есть оценок, которые на единицу выше и т. д. до последней градации [10].

Согласно правилу порогового агрегирования, индекс предпочтения альтернативы будет равен сумме количеств сочетаний из a по b (1) [14–18]:

$$F(x) = \sum_{j=1}^{m} C_{a(j)}^{b(j)}, \quad (1)$$

где возможно появление следующих сочетаний, которые доопределены: $C_{-1}^{0} = 1$, и $C_n^{n+1} = 0$.

a и b будут зависеть от j и определяться как (2), (3):

$$a(j) = n - V_j(x) + m - j - 1; \quad (2)$$

$$b(j) = m - j; \quad (3)$$

$V_j(x)$ — это количество рангов j в векторе x, т. е. $V_j(x) = \{1 \le i \le n : x_i = j\}$. $0 \le V_j(x) \le n$ для всех $j \in \{1, ...m\}$, и $x \in k$, и $V_1(x) + ... + V_n(x) = n$ для всех $x \in k$.

$V_j(x)$ будет рассчитываться как сумма (4):

$$V_j(x) = \sum_{q=1}^{j} \eta(q) \text{ и } \sum_{j=1}^{m} V_j(x) = n. \quad (4)$$

$\eta(q)$ — это количество параметров, по которым оцениваемая альтернатива имеет значение q (q: градация от 1 до m).

После нахождения количества a и b найдем C, используя формулу из комбинаторики, а именно формулу сочетания (5):

$$C_a^b = \frac{a!}{b!(a-b)!}. \quad (5)$$

Функция предпочтения F (для $m \ge 4$) в явном виде представлена в работах [14] и [15]. Если подставим (2) и (3) в (1), то получим следующую комбинаторную формулу (6):

$$F(x) = \sum_{j=1}^{m} C_{n-V_j(x)+m-j-1}^{m-j} = \sum_{j=1}^{m} C_{n-(\eta(1)+\eta(2)+...+\eta(j))+m-j-1}^{m-j}. \quad (6)$$

Нормированный индекс равен (7):

$$I_{Threshold} = \frac{F}{F_{\max}}, \quad (7)$$

где $F_{\max}$ — это максимальное рассчитанное значение для индекса предпочтения. Диапазон нормированного индекса находится — от 0 до 1. Чем выше значение индекса, тем, соответственно, выше значение показателя.

На рис. 1 представлена схема (алгоритм) этапов оценки по модели некомпенсаторного агрегирования

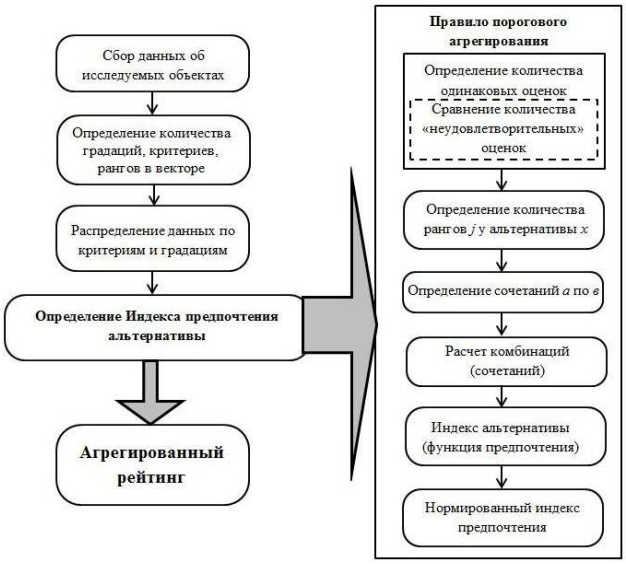

Рис. 1. Схема этапов оценки по модели некомпенсаторного агрегирования
[Fig. 1. Scheme of the assessment stages using the non-compensatory aggregation model]

Для получения агрегированного рейтинга оцениваемых объектов необходимо выполнение этапов, представленных в данной схеме.

2.2. Алгоритм программы «Формирование агрегированного рейтинга»

На рис. 2 представлен алгоритм разработки программного обеспечения для формирования агрегированных рейтингов.

В программе предусмотрено пять выводов результатов (промежуточных и итоговый ин-

декс предпочтения). Вначале необходим ввод трех параметров, и далее заполнить данные по критериям и альтернативам. Поскольку расчет предусматривает работу с большим количеством данных, то удобно представлять их в виде матрицы (таблицы). Поэтому в алгоритме применяются циклы для перебора элементов в каждой строке, т. е. используются двумерные массивы.

3. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

3.1. Разработка программного обеспечения на языке C

Реализуем наш алгоритм в Visual Studio 2019 на языке C #.

Программа будет позволять хранить данные в базе и сохранять их в виде файла *txt*.

Программа реализует следующие функции:

1. Учет данных об альтернативах, критериях и градациях для оценивания.

2. Расчет параметров и количества сочетаний.

3. Расчет индекса предпочтения для построения агрегированного рейтинга.

4. Формирование графика рейтинга.

Согласно построенному алгоритму в программе предусмотрено 5 выводов данных: количество одинаковых оценок, сумма V_j, значения векторов a и b, количества сочетаний, индексы предпочтений (на основе которого и будет строиться рейтинг).

При запуске программы можно внести название проекта и ФИО сотрудника, выполняющего рейтингование. Для начала работы необходимо ввести количество критериев, альтернатив и градаций. Именно на основе этих введенных параметров будут формироваться таблицы для ввода данных, и производиться расчеты согласно некомпенсаторной модели.

Программа разделена на 6 вкладок. После внесения параметров на первой вкладке «*Ввод данных*» вводятся значения критериев для каждой альтернативы. Сами альтернативы можно будет выгрузить из базы данных или из специально подготовленного файла. Отбор критериев можно также осуществить, нажав на кнопку «Критерии». Заполнение базы данных по альтернативам и критериям может быть уникальным для каждого предприятия, исходя из его специфики.

Программа разработана максимально универсально для любой области, где можно, так или иначе, составлять рейтинги.

На 2-ой вкладке «*Формирование векторов*» (рис. 3) пользователю предлагается выбрать: «разнести по градациям» или «оставить без изменений». Если на первой вкладке вносились реальные значения критериев, то необходимо будет выбрать «разнести по градациям», сформируется чистая таблица, в которую нужно будет разнести оценки. Если первоначально сразу вносились экспертные оценки (например, в случае оценки качественных показателей), то необходимо выбрать «оставить без изменений», и введенные данные на 1-й вкладке автоматически скопируются во 2-ю вкладку. Либо, если сразу планируется проставлять экспертные оценки, то можно заполнять 2-ю вкладку, миновав 1-ю.

После ввода параметров и значений критериев для альтернатив согласно правилу порогового агрегирования и алгоритма идет поиск количества одинаковых оценок. Для этого в алгоритме реализована функция *findSameElements*, в которой используется временной словарь *Dictionary*. Благодаря этому словарю идет заполнение строчек альтернатив количеством одинаковых оценок, т. е. если например, у альтернативы две оценки со значением 3, то в третьей ячейке искомой матрицы сформируется значение 2.

Листинг для функции «findSameElements», реализующей расчет количества оценок:

```
private void findSameElements(int rows, int
columns, int[,] mass)
{
 // сначала подготовили грид
 makeHeaders(dataGridView5, rows, 10, «Век-
тор»);
 // временный Словарь для поиска повторений
Dictionary<int,    int>    dict    =    new
Dictionary<int, int>();
  for (int i = 0; i < rows; i++)
 {
  dict.Clear();
//заполняем начальными значениями
  for (int j = 1; j < 11; j++)
   {
    dict[j] = 0;
```

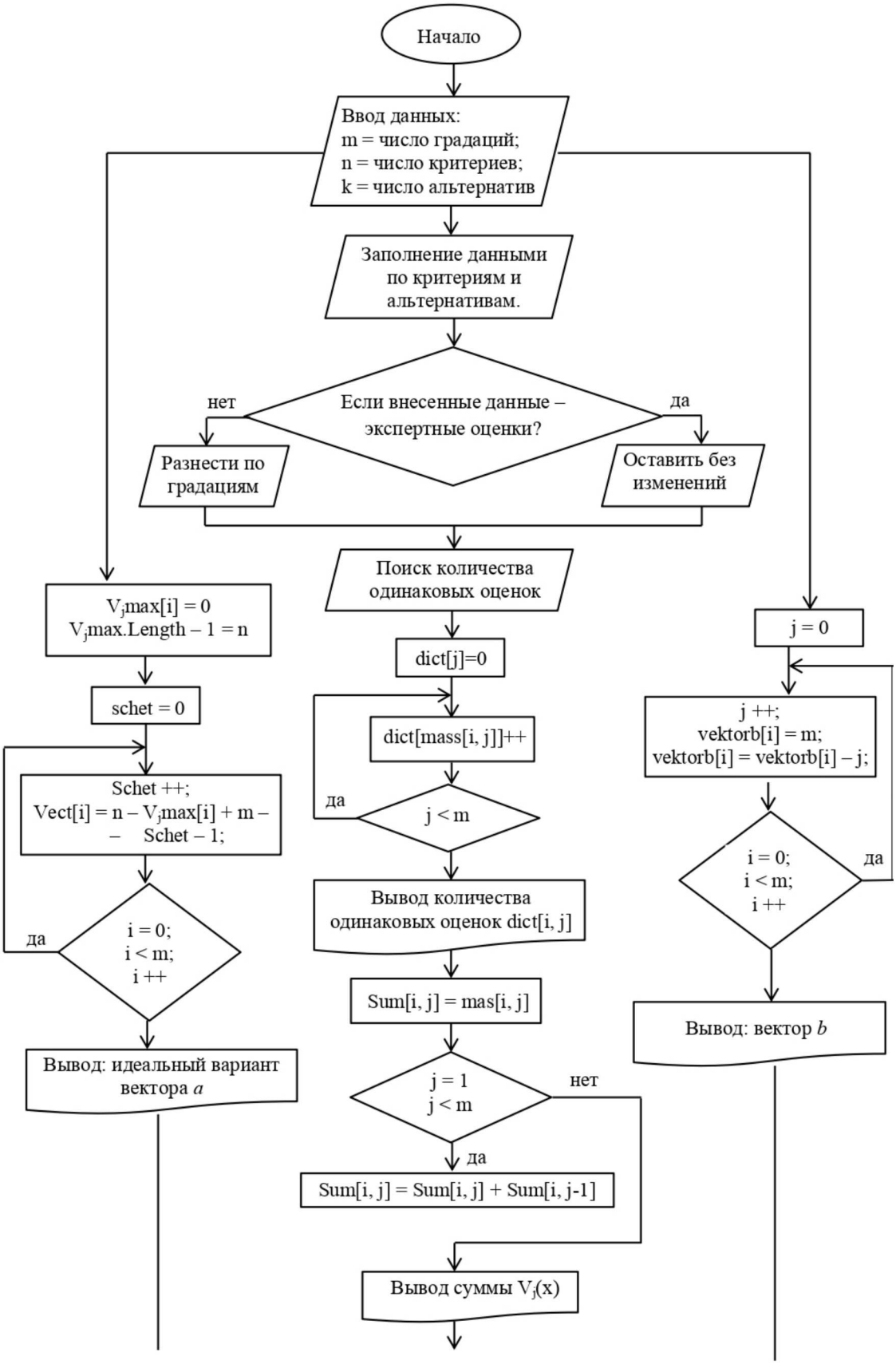

Рис. 2. Алгоритм программы «Формирование агрегированного рейтинга»
[Fig. 2. Algorithm of the program «Formation of the aggregated rating»]

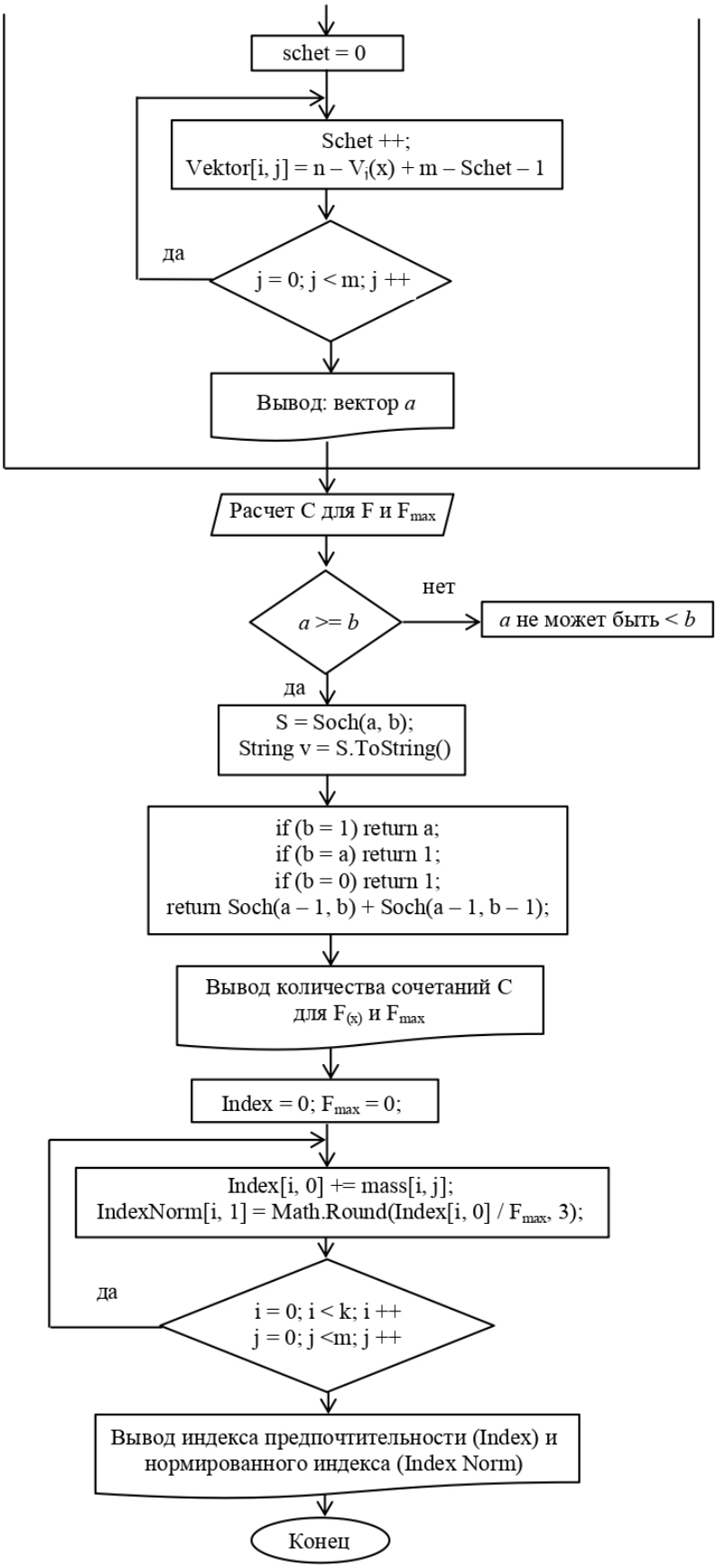

Рис. 2. Продолжение алгоритма программы «Формирование агрегированного рейтинга»
[Fig. 2. Continuation of the algorithm of the program «Formation of the aggregated rating»]

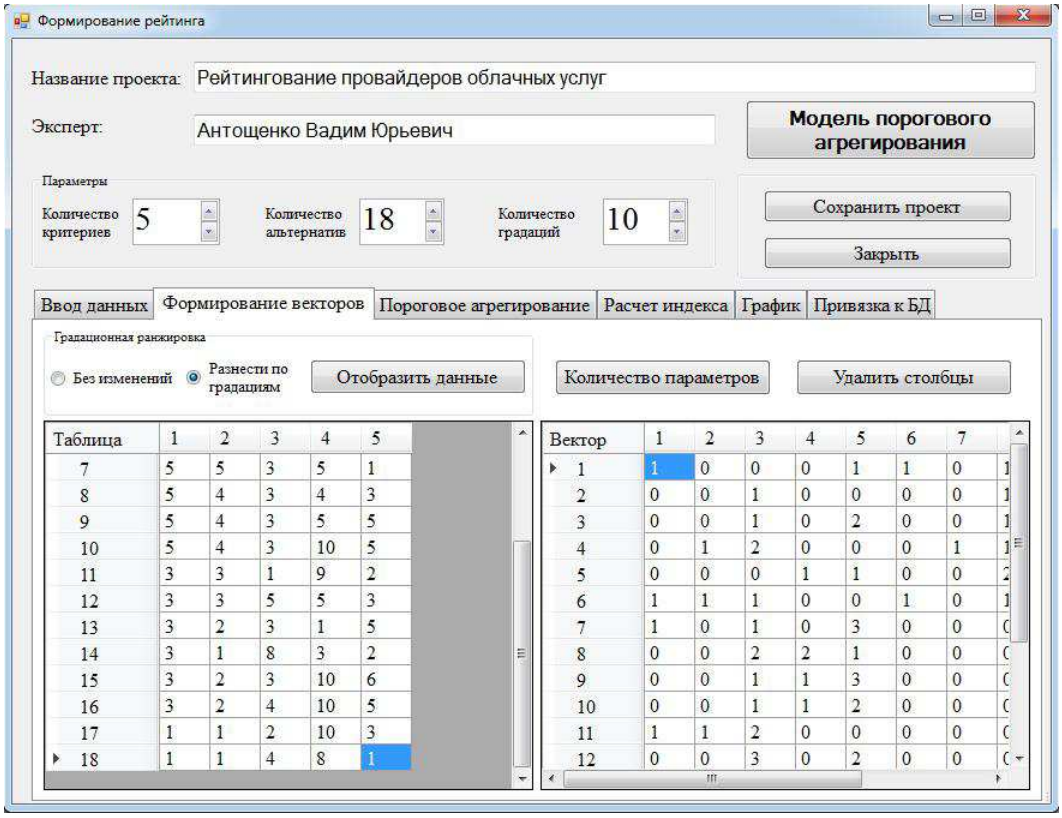

Рис. 3. Диалоговое окно программы «Формирование агрегированных рейтингов», вкладка «Формирование векторов»

[Fig. 3. Dialog box of the program «Formation of aggregated ratings», tab «Formation of vectors»]

```
   }
//делаем поиск
  for (int j = 0; j < columns; j++)
   {
    dict[mass[i, j]]++;
   }
// Словарь готов - нужно его вывести в текущую строчку грида
  for (int j = 0; j < 10; j++)
   {
dataGridView5.Rows[i].Cells[j].Value = Convert.ToString(dict[j + 1]);
   }
 }
}
```

Функция *findSameElements* для вывода количества одинаковых оценок запрограммирована для кнопки «Количество параметров».

Далее по алгоритму идет вывод суммы V_j, а также значений векторов a и b. Это реализовано на 3-й вкладке «*Пороговое агрегирование*», где производятся промежуточные расчеты по правилу (рис. 4). В дальнейшем ввода никакого не потребуется, только последовательное нажатие кнопок, которые отвечают за промежуточный вывод результатов.

За расчет суммы V_j отвечает функция *SummaElements*, по которой будет производиться расчет по формуле (4). Листинг для функции «SummaElements»:

```
private void SummaElements(int a, int c, int[,] mass)
{
 int[,] summV = new int[a, c];
 for (int i = 0; i < a; i++)
  {
   for (int j = 0; j < c; j++)
    {
     summV[i, j] = mass[i, j];
dataGridView7.Rows[i].Cells[j].Value = summV[i, j];
    }
  }
 for (int i = 0; i < a; i++)
  {
   for (int j = 1; j < c; j++)
    {
     summV[i, j]= summV[i, j]+summV[i,j-1];
dataGridView7.Rows[i].Cells[j].Value = summV[i, j];
    }
  }
}
```

Программа для построения агрегированных рейтингов

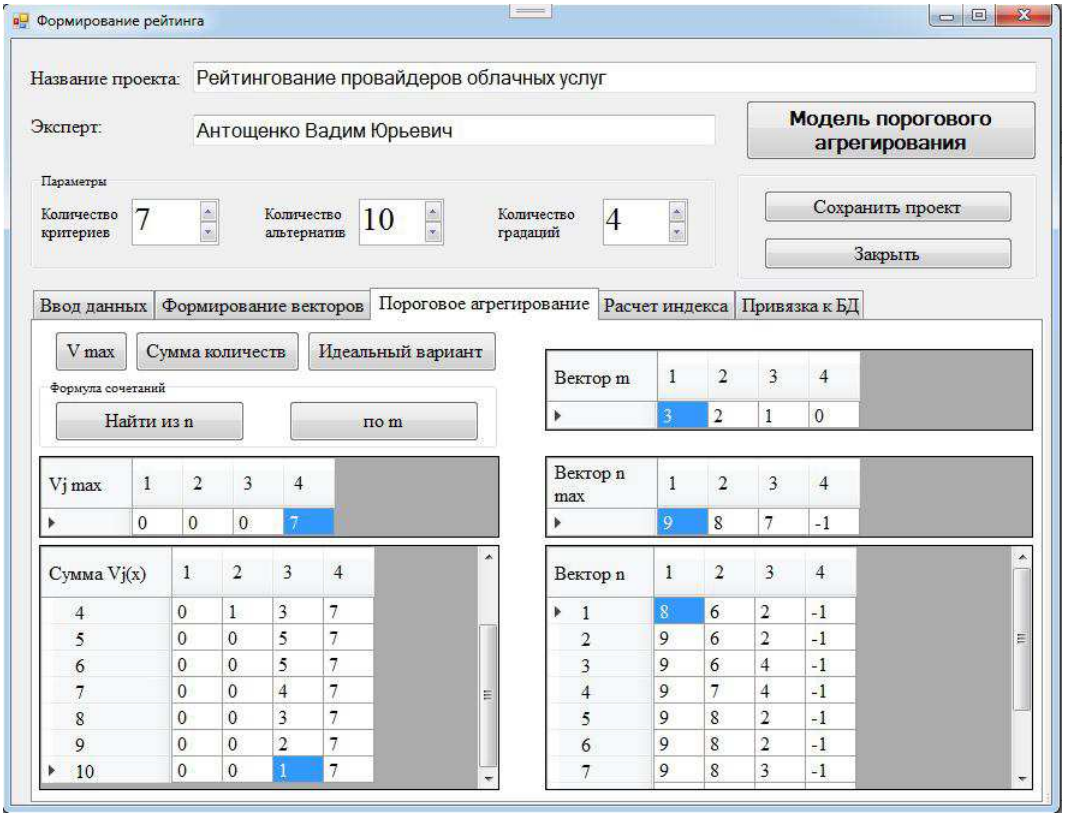

Рис. 4. Диалоговое окно программы «Формирование агрегированных рейтингов», вкладка «Пороговое агрегирование»

[Fig. 4. Dialog box of the program «Formation of aggregated ratings», tab «Threshold aggregation»]

За расчет значений вектора a отвечает функция *findVektorN*, по корой будет производиться расчет по формуле (2). Листинг для функции «findVektorN»:

```
private void findVektorN(int a, int c, int[,] mass)
{
 int[,] vekt = new int[a, c];
  int b = Convert.ToInt32(numericUpDown1.Value); //количество столбцов (критериев)
 int Kriteriy = b; // количество строк (альтернатив)
 int Rang = c; // количество градаций
 int Schet; // счетчик

 for (int i=0; i<a; i++)
  {
   Schet=0;
   for (int j=0; j<c; j++)
    {
     Schet++;
     vekt[i, j] = Kriteriy  - mass[i, j] + Rang - Schet - 1;
    }
  }
   for (int i = 0; i < a; i++)
    {
     for (int j = 0; j < c; j++)
      {
   dataGridView6.Rows[i].Cells[j].Value = vekt[i, j];
      }
    }
}
```

За расчет значений вектора b отвечает функция *findVektorM*, по корой будет производиться расчет по формуле (3).

Четвертый вывод — это количество сочетаний, который представлен на вкладке «Расчет индекса» (рис. 5). Здесь производится расчет сочетаний a по b, включая для F_{max}, а также расчет самих индексов предпочтения вместе с нормированными значениями. Перед расчетом индекса необходимо нажать на кнопку «Применить условие», чтобы сработало ограничение $C_{-1}^{0} = 1$. При нажатии на кнопку «Fmax» будет выведено его значение, которое будет участвовать при расчете нормированного индекса. По кнопке «Сформировать рейтинг» произойдет ранжирование оцениваемых альтернатив по индексу от меньшего к большему. По кнопке «Формула сочетания»

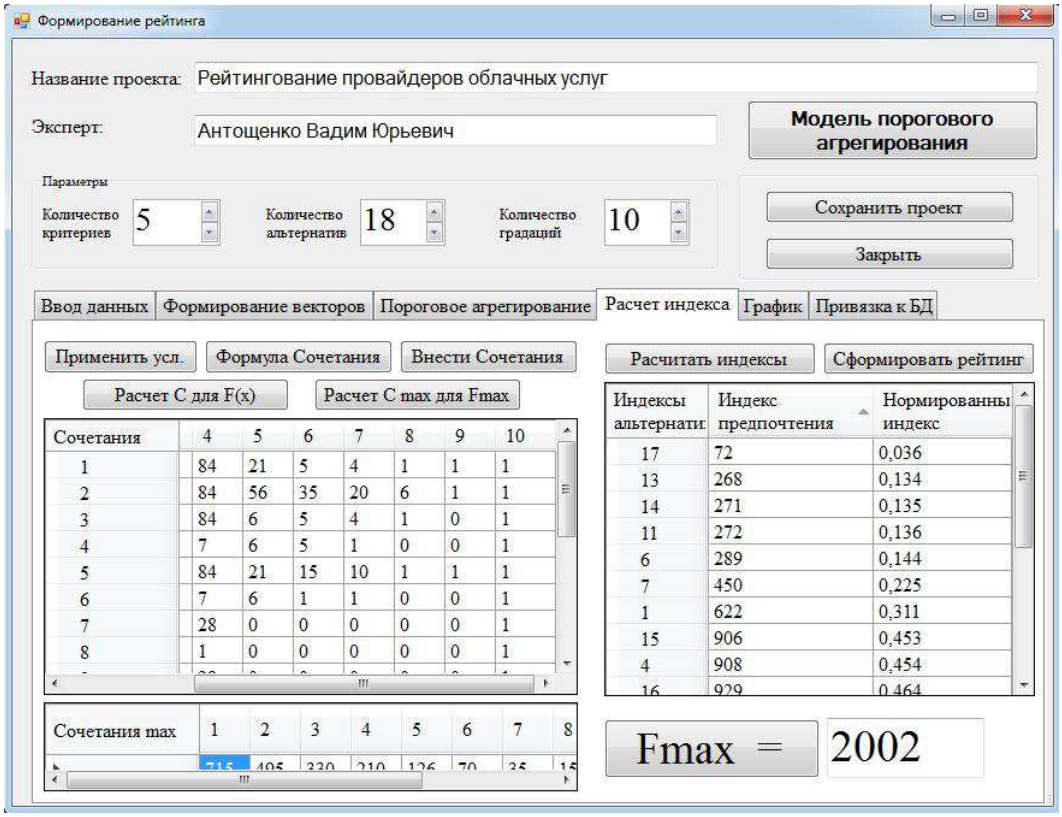

Рис. 5. Диалоговое окно программы «Формирование агрегированных рейтингов», вкладка «Расчет индекса»

[Fig. 5. Dialog box of the program «Formation of aggregated ratings», tab «Index calculation»]

вызовется форма, в которой можно забить любые значения *a* и *b* для проверки или просто расчета значения количества сочетаний.

Расчет количества сочетаний реализуют функции *IndexNorm* и *SochN*. Листинг для этих функций:

```
private void IndexNorm(int a, int c, int[]
massk, int[,] mass)
{
 for (int i = 0; i < a; i++)
 {
  for (int j = 0; j < c; j++)
  {
   int k = massk[j];
   int n = mass[i, j];
   if (n >= k)
   {
    int b = SochN(n, k);
    string v = b.ToString();
      dataGridView10.Rows[i].Cells[j].Value
= v;
   }
    else
   {
    MessageBox.Show(«N не может быть мень-
ше,чем K»);
   }
  }
 }
}
public static int SochN(int n, int k)
{
if (k == 1) return n;
if (k == n) return 1;
if (k == 0) return 1;
return SochN(n - 1, k) + SochN(n - 1, k -
1);
}
```

При программировании формулы сочетания (5) использовался метод рекурсии.

За вывод значений итогового индекса предпочтений по формуле (1) и нормированного индекса отвечает функция *Index*. Листинг для функции:

```
private void Index(int a, int c, int[,]
mass)
{
 int[,] koef = new int[a, c];
 int F = int.Parse(textBox3.Text);
 for (int i = 0; i < a; i++)
 {
  koef[i,0] = 0;
 }
 for (int i = 0; i < a; i++) //Выводим век-
```

```
тор (первый столбец)
 {
  for (int j = 0; j < c; j++)
 {
 koef[i, 0] += mass[i, j];
   dataGridView9.Rows[i].Cells[0].Value   =
koef[i, 0];
 }
     dataGridView9.Rows[i].Cells[1].Value=
Math.Round(Convert.ToDouble(koef[i,  0]  /
Convert.ToDouble(F)),3);
 }
}
```

Здесь нормированный индекс округлен до тысячных.

На 5-й вкладке «График» можно сформировать график рейтинга для всех оцениваемых альтернатив. По оси x отражены альтернативы, по оси y — значение индекса предпочтения.

На 6-й вкладке «Привязка к БД» имеется возможность сохранить полученные результаты в базу данных, выгрузив в файл, и сформировать другого вида график.

4. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Предложенная программа для формирования агрегированных рейтингов позволяет вести удобный и быстрый расчет индекса предпочтений и строить проранжированный агрегированный рейтинг, в том числе в виде графика, а также сохранять информацию в базу данных. Промежуточные расчеты по модели (согласно схеме этапов оценки), которые выводятся в соответствующих вкладках программы, позволяют пользователю их увидеть и проверить при необходимости.

После реализации программы на языке C# было проведено тестирование на реальных примерах [19–21], а именно построение рейтингов для:

– провайдеров облачных услуг;

– популярных SaaS-сервисов на основе организации каналов поддержки клиентов;

– крупнейших поставщиков IaaS.

Первоначально все расчеты были выполнены вручную с использованием онлайн калькулятора по нахождению количества сочетаний. Минус такого калькулятора в том, что он не учитывает ограничения: $C_{-1}^{0}=1$, и $C_{n}^{n+1}=0$. Рассчитывает только C при $a>b$. Так и должно быть, только при выполнении расчетов по модели могут возникнуть и такие ограничения, которые хоть и несущественные, но влияют на итоговый результат. В программе «Формирование агрегированных рейтингов» учтен этот момент. При появлении ситуации, что $C_{n}^{n+1}=0$, программа покажет уведомление, что b не может быть больше a, но при этом не исказит верный ответ, и в соответствующую ячейку подставится пустое множество. В случае ситуации $C_{-1}^{0}=1$, в соответствующую ячейку автоматически подставится 1. А данное определение образуется в конце каждого вектора при вычислении количеств сочетаний.

После выполнения расчетов в программе все выведенные результаты сошлись с расчетами, выполненными вручную, что подтверждает правильность реализации алгоритма и программного кода.

Пользователю достаточно внести начальные параметры, а затем данные альтернатив по критериям. Дальнейших внесений не потребуется. Программа полностью выполнит все расчеты и выведет промежуточные и итоговые результаты в соответствующие поля. Пользователю только важно выполнить последовательное нажатие кнопок. При возникновении вопросов или для пояснений алгоритма действий в программе предусмотрена кнопка «Модель порогового агрегирования», где поэтапно описаны шаги, которые необходимо выполнить для реализации расчетов по модели. К этим шагам прописаны формулы, поясняющий алгоритм модели. В любом случае работа в программе проста и интуитивно понятна.

Расчеты, которые необходимо было выполнять вручную в течение дня, программа безошибочно выполнит за секунды после внесения необходимых данных. Применение информационных технологий совместно с запрограммированным математическим обеспечением существенно облегчает работу эксперта и ЛПР в принятии обоснованного решения.

Не смотря на наличие онлайн калькуляторов, которые позволяют рассчитать количе-

ство сочетаний, новизна данной программы в том, что она учитывает ограничения, которые онлайн калькулятор не воспринимает, и самое главное, что она обеспечивает все этапы реализации предложенной модели на основе правила порогового агрегирования. Программу можно установить на компьютер как Windows приложение, а также привязать и добавить как модуль в систему поддержки принятия решений (СППР), написанных на C#, 1С: Предприятие или других платформах/языках программирования.

Использование данной программы позволит формировать агрегированные рейтинги в любых областях, где, так или иначе, оцениваются альтернативы для определения лучшей из них. Таким образом, данная программа универсальна. Она зарегистрирована и на нее получено Свидетельство о государственной регистрации программы для ЭВМ № 2021619283 от 8 июня 2021 г [22] «Построение обобщенных рейтингов методом порогового агрегирования».

ЗАКЛЮЧЕНИЕ

В статье представлена модель некомпенсаторного агрегирования для построения рейтингов, в основе которой лежит правило порогового агрегирования, применяемого в задачах многокритериального оценивания. Была разработана схема этапов оценки по данной модели и алгоритм для разработки программного обеспечения «Формирование агрегированного рейтинга». Данная модель была запрограммирована на языке C# в среде *Visual Studio 2019* в виде Windows приложения. Программа протестирована на примере реальных данных при построении списка лучших альтернатив. Приведен пример расчета и построения рейтинга для поставщиков облачных услуг.

БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке Стипендии Президента РФ, № СП-3174.2019.5.

КОНФЛИКТ ИНТЕРЕСОВ

Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Zuheros, C.* Sentiment Analysis based Multi-Person Multi-criteria Decision Making methodology using natural language processing and deep learning for smarter decision aid. Case study of restaurant choice using TripAdvisor reviews / C. Zuheros, E. Martínez-Cámara, E. Herrera-Viedma, F. Herrera // Information Fusion. 2021. – 68. – P. 22–36. DOI: 10.1016/j.inffus.2020.10.019.

2. *Разумников, С. В.* Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM // Доклады томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61. DOI: 10.21293/1818-0442-2020-23-1-53-61.

3. Multi-criteria optimization and decision-making in radiotherapy / S. Breedveld, D. Craft, R. Haveren, B. Heijmen // European Journal of Operational Research. – 2019. – Vol. 277, No. 1. – P. 1–19. DOI: 10.1016/j.ejor.2018.08.019.

4. Measurement of chip morphology and multi criteria optimization of turning parameters for machining of AISI 4340 steel using Y-ZTA cutting insert / B. K. Singh, H. Roy, B. Mondal, S. S. Roy, N. Mandal // Measurement: Journal of the International Measurement Confederation. – 2019. – Vol. 142. – P. 181–194. DOI: 10.1016/j.measurement.2019.04.064.

5. *Meyen, S.* Group decisions based on confidence weighted majority voting / S. Meyen, D. M. B. Sigg, U. Luxburg, V. H. Franz // Cognitive Research: Principles and Implications. 2021. – 6(1), 18. DOI: 10.1186/s41235-021-00279-0.

6. *Heidary Dahooie J.* A novel dynamic credit risk evaluation method using data envelopment analysis with common weights and combination of multi-attribute decision-making methods / J. Heidary Dahooie, S. H. Razavi Hajiagha, S. Faraz-

mehr, E. K. Zavadskas, J. Antucheviciene // Computers and Operations Research. 2021. – 129, 105223. DOI: 10.1016/j.cor.2021.105223.

7. Sensitivity analysis and multi-criteria optimization of SMA cable restrainers for longitudinal seismic protection of isolated simply supported highway bridges / J. Q. Wang, S. Li, F. Hedayati Dezfuli, M. S. Alam // Engineering Structures. – 2019. – Vol. 189. – P. 509–522. DOI: 10.1016/j.engstruct.2019.03.091.

8. *Андрейчиков, А. В.* Системный анализ стратегических решений в инноватике. Математические, эвристические и интеллектуальные методы системного анализа и синтеза инноваций / А. В. Андрейчиков, О. Н. Андрейчикова. – М. : Книжный дом «ЛИБРИКОМ», 2013. – 304 с.

9. *Разумников, С. В.* Некомпенсаторное агрегирование и рейтингование провайдеров облачных услуг // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2018. – Т. 21, № 4. – С. 63–69. DOI: 10.21293/1818-0442-2018-21-4-63-69.

10. *Алескеров, Ф. Т.* Бинарные отношения, графы и коллективные решения / Ф. Т. Алескеров, Э. Л. Хабина, Д. А. Шварц. – 2-е изд., перераб. и доп. – М. : ФИЗМАТЛИТ. – 2012. – 344 с.

11. *Калягин, В. А.* Аксиоматическая модель некомпенсаторного агрегирования: Препринт WP7/2009/01 / В. А. Калягин, В. В. Чистяков. – М. : Изд. дом ГУ ВШЭ, –2009. – 76 с.

12. *Maroukhine, O. V.* Expert support system for making decision by the results of computer-based testing within the ends of teaching quality evaluation / O.V. Maroukhine, O. G. Berestneva // Proceedings – KORUS 2003: 7th Korea-Russia International Symposium on Science and Technology. – 2003. – Vol. 2. – P. 416–419.

13. *Oikonomou, E. K.* Sustainable coastal zone management of Strymonikos Gulf: Implementation of the analytic hierarchy process through an application designed using the programming language C# (sharp) / E. K. Oikonomou, E. Yiannakopoulou // World Review of Science, Technology and Sustainable Development. 2021. – 17(1). – P. 54–80. DOI: 10.1504/WRSTSD.2021.114021.

14. Оценка вклада научных работников методом порогового агрегирования / Ф. Т. Алескеров, Е. С. Катаева, В. В. Писляков, А. И. Якуба // Управление большими системами. Специальный выпуск 44: «Наукометрия и экспертиза в управлении наукой». – 2013 г. – С. 172–189.

15. *Алескеров, Ф. Т.* Пороговое агрегирование трехградационных ранжировок / Ф. Т. Алескеров, Д. А. Юзбашев, В. И. Якуба // Автоматика и телемеханика. – 2007. – Выпуск 1. – С. 147–152.

16. *Aleskerov, F.* A threshold aggregation of three-graded rankings / F. Aleskerov, V. Yakuba, D. Yuzbashev. // Math. Social Sci. 53 –2007. – P. 106–110.

17. *Aleskerov, F.* The threshold aggregation / F. Aleskerov, V. V. Chistyakov, V. A. Kalyagin // Econ. lett. 107. – No 2. – 2010. – P. 161–162.

18. *Aleskerov, F.* Social threshold aggregations / F. Aleskerov, V. Chistyakov, V. Kalyagin // Social Choice and Welfare. – 2010. – Vol. 35, № 4. – P. 627–646.

19. *Razumnikov, S. V.* Models of evaluating efficiency and risks on integration of cloud-base IT-services of the machine-building enterprise: a system approach / S. V. Razumnikov, A. K. Kurmanbay // IOP Conference Series: Materials Science and Engineering. – 2016. – Vol. 124, No. 1. – P. 1–5. DOI: 10.1088/1757-899X/124/1/012089.

20. *Razumnikov, S. V.* Integrated model to assess cloud deployment effectiveness when developing an IT-strategy / S. V. Razumnikov, D. Prankevich // IOP Conference Series: Materials Science and Engineering. – 2016. – Vol. 127: Urgent Problems of Modern Mechanical Engineering. DOI: 10.1088/1757-899X/127/1/012018.

21. *Sultan, N.* Knowledge management in the age of cloud computing and Web 2.0: Experiencing the power of disruptive innovations // International journal of information management. – 2013. – Vol. 33, No. 1. – P. 160–165. DOI: 10.1016/j.ijinfomgt.2012.08.006.

22. Свидетельство о государственной регистрации программы для ЭВМ «Построение обобщенных рейтингов методом порогового агрегирования» / Разумников С.В.; заявитель и правообладатель ФГАОУ ВО Национальный исследовательский Томский политехнический университет – № 2021619283; заявл. 02.06.21; опубл. 08.06.21.

Разумников Сергей Викторович — канд. техн. наук, доцент Юргинского технологического института (филиала) Национального исследовательского Томского политехнического университета (ЮТИ ТПУ).
E-mail: demolove7@inbox.ru
ORCID iD: https://orcid.org/0000-0002-1417-498X

DOI: https://doi.org/10.17308/sait.2021.2/3510 ISSN 1995-5499
Received 28.05.2021
Accepted 19.07.2021

DEVELOPMENT OF SOFTWARE FOR BUILDING AGGREGATE RATINGS BASED ON THE THRESHOLD AGGREGATION METHOD

© 2021 S. V. Razumnikov✉

Yurga Technological Institute (branch) of the National Research Tomsk Polytechnic University
26, Leningradskaya Street, 652055 Yurga, Russian Federation

Annotation. The selection of the best alternative is an important problem in the theory of decision making. As a rule, when choosing the alternatives, they are evaluated according to various criteria, and a rank vector is formed to determine the best one. If the situation is serious, it is necessary to take into account the non-compensatory nature of the values of the criteria. To avoid the compensation problem, we suggest forming an aggregate rating using the threshold aggregation rule. The article presents a non-compensatory aggregation model that can be used for ranking. The model is based on the rule of threshold aggregation used in multi-criteria assessment problems. This method does not allow compensating for low scores by higher scores for other criteria. The article also presents a diagram of the assessment stages for this model and an algorithm for the development of software "Building an aggregate rating". The algorithm describes the used calculation functions and shows the code listing. The presented model was programmed using C# in the Visual Studio 2019 environment as a Windows application. This program is convenient for the quick calculation of the preference index and building an aggregate rating, which can be presented as a graph. It can also store information in a database. In our study, we used the program to build a rating of cloud service providers. The programme is universal and can be used to build a generalized rating of evaluated alternatives in any area, as well as to make decisions when selecting the best alternative.
Keywords: threshold aggregation, model, algorithm, scheme, program, listing, rating, criteria, alternatives, gradations.

CONFLICT OF INTEREST

The author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Zuheros C., Martínez-Cámara E., Herrera-Viedma E., Herrera F.* (2021) Sentiment Analysis based Multi-Person Multi-criteria Decision Making methodology using natural language processing and deep learning for smarter decision aid. Case study of restaurant choice using TripAdvisor reviews / C. Zuheros // Information Fusion. 68. P. 22–36. DOI: 10.1016/j.inffus.2020.10.019.

2. *Razumnikov S. V.* (2020) Planning the development of a cloud strategy based on the application of multicriteria optimization and the STEM method // Reports of the Tomsk State University of Control Systems and Radioelectronics.

✉ Razumnikov Sergei V.
e-mail: demolove7@inbox.ru

T. 23, No. 1. P. 53–61. DOI: 10.21293/1818-0442-2020-23-1-53-61.

3. *Breedveld S., Craft D., Haveren R., Heijmen B.* (2019) Multi-criteria optimization and decision-making in radiotherapy // European Journal of Operational Research. Vol. 277, No. 1. P. 1–19. DOI: 10.1016/j.ejor.2018.08.019.

4. *Singh B. K., Roy H., Mondal B., Roy S. S., Mandal N.* (2019) Measurement of chip morphology and multi criteria optimization of turning parameters for machining of AISI 4340 steel using Y-ZTA cutting insert / // Measurement: Journal of the International Measurement Confederation. Vol. 142. P. 181–194. DOI: 10.1016/j.measurement.2019.04.064.

5. *Meyen S., Sigg D. M. B., Luxburg U., Franz V. H.* (2021) Group decisions based on confidence weighted majority voting // Cognitive Research: Principles and Implications. 6(1). 18. DOI: 10.1186/s41235-021-00279-0.

6. *Heidary Dahooie J., Razavi Hajiagha S. H., Farazmehr S., Zavadskas E. K., Antucheviciene J.* (2021) A novel dynamic credit risk evaluation method using data envelopment analysis with common weights and combination of multi-attribute decision-making methods // Computers and Operations Research. 129. 105223. DOI: 10.1016/j.cor.2021.105223.

7. *Wang J. Q., Li S., Hedayati Dezfuli F., Alam M. S.* (2019) Sensitivity analysis and multi-criteria optimization of SMA cable restrainers for longitudinal seismic protection of isolated simply supported highway bridges // Engineering Structures. Vol. 189. P. 509–522. DOI: 10.1016/j.engstruct.2019.03.091.

8. *Andreychikov A. V., Andreychikova O. N.* (2013) System analysis of strategic decisions in innovation. Mathematical, heuristic and intellectual methods of system analysis and synthesis of innovations. Moscow : Book house "LIBRIKOM". 304 p.

9. *Razumnikov S. V.* (2018) Non-compensatory aggregation and rating of cloud service providers // Reports of the Tomsk State University of Control Systems and Radioelectronics. T. 21, No. 4. P. 63–69.DOI: 10.21293/1818-0442-2018-21-4-63-69.

10. Aleskerov F. T., Khabina E. L., Schwartz D. A. (2012) Binary relations, graphs and collective decisions. 2nd ed., Rev. and add. Moscow : FIZMATLIT. 344 p.

11. *Kalyagin V. A., Chistyakov V. V.* (2009) Axiomatic model of non-compensatory aggregation: Preprint WP7 / 2009/01. Moscow : Publishing house. House of the State University Higher School of Economics. 76 p.

12. *Maroukhine O. V., Berestneva O. G.* (2003) Expert support system for making decision by the results of computer-based testing within the ends of teaching quality evaluation // Proceedings – KORUS 2003: 7th Korea-Russia International Symposium on Science and Technology. Vol. 2. P. 416–419.

13. *Oikonomou E. K., Yiannakopoulou E.* (2021) Sustainable coastal zone management of Strymonikos Gulf: Implementation of the analytic hierarchy process through an application designed using the programming language C# (sharp) // World Review of Science, Technology and Sustainable Development. 17(1). P. 54–80. DOI: 10.1504/WRSTSD.2021.114021.

14. *Aleskerov F. T., Kataeva E. S., Pislyakov V. V., Yakuba A. I.* (2013) Assessment of the contribution of researchers by the threshold aggregation method // Management of large systems. Special issue 44: "Scientometrics and Expertise in Science Management". P. 172–189.

15. *Aleskerov F. T., Yuzbashev D. A., Yakuba V. I.* (2007) Threshold aggregation of three-gradation rankings // Automation and telemechanics. Issue 1. P. 147–152.

16. *Aleskerov F., Yakuba V., Yuzbashev D.* (2007) A threshold aggregation of three-graded rankings // Math. Social Sci. 53. P. 106–110.

17. *Aleskerov F., Chistyakov V. V., Kalyagin V. A.* (2010) The threshold aggregation // Econ. lett. 107. No 2. P. 161–162.

18. *Aleskerov F., Chistyakov V., Kalyagin V.* (2010) Social threshold aggregations / F. Aleskerov, // Social Choice and Welfare. Vol. 35, No 4. P. 627–646.

19. *Razumnikov S. V., Kurmanbay A. K.* (2016) Models of evaluating efficiency and risks on integration of cloud-base IT-services of the machine-building enterprise: a system approach / S.V. Razumnikov, // IOP Conference Series: Materials Science and Engineering. Vol. 124, No. 1. P. 1–5. DOI: 10.1088/1757-899X/124/1/012089.

20. *Razumnikov S. V., Prankevich D.* (2016) Integrated model to assess cloud deployment effectiveness when developing an IT-strategy // IOP Conference Series: Materials Science and Engineering. Vol. 127: Urgent Problems of Modern Mechanical Engineering. DOI: 10.1088/1757-899X/127/1/012018.

21. *Sultan N.* (2013) Knowledge management in the age of cloud computing and Web 2.0: Experiencing the power of disruptive innovations // International journal of information management. Vol. 33, No. 1. P. 160–165. DOI: 10.1016/j.ijinfomgt.2012.08.006.

22. *Razumnikov S. V.* (2021) Certificate of state registration of the computer program "Building generalized ratings by the threshold aggregation method"; applicant and copyright holder National Research Tomsk Polytechnic University. No. 2021619283; app. 06/02/21; publ. 08.06.21.

Razumnikov Sergei V. — PhD in Technical Sciences, Associate Professor, Yurga Institute of Technology, TPU Affilate.
E-mail: demolove7@inbox.ru
ORCID iD: https://orcid.org/0000-0002-1417-498X