

## МНОГОКРИТЕРИАЛЬНЫЙ АНАЛИЗ НЕЧЕТКИХ ОБЪЕКТОВ С КЛАСТЕРИЗАЦИЕЙ ЭКСПЕРТНЫХ ОЦЕНОК <sup>1</sup>

В данной работе приводится способ многокритериального ранжирования альтернатив с использованием множества нечетких экспертных оценок, при этом предлагается метод кластеризации нечетких экспертных оценок с учетом весов критериев оценки.

The article represents a multicriterial alternatives' ranging method involving a set of the experts' fuzzy estimations. The experts' fuzzy estimations clustering method based upon the fuzzy criteria weights is also given in the article.

**Ключевые слова:** многокритериальное ранжирование, нечеткий анализ, мягкие вычисления, нечеткая кластеризация.

**Keywords:** multicriterial ranging, fuzzy analysis, soft computing, fuzzy clustering.

Рассмотрим следующую задачу, называемую задачей ранжирования. Пусть у нас имеется набор из  $K$  объектов (**альтернатив**), из которых надо выбрать один или несколько наиболее предпочтительных (или, что то же самое, упорядочить их по степени предпочтительности).

Более точно, результатом ранжирования должно стать разбиение объектов на классы  $C_1, C_2, \dots, C_k$  такие, что объекты из  $(i + 1)$ -ого класса предпочтительнее объектов из  $i$ -ого класса для любого индекса  $i$ , а объекты внутри каждого класса равноценны между собой.

Решение проходит в несколько этапов.

**Э т а п 1.** Информация об объектах, из которой надо извлечь данные о предпочтительности, поступает нам в следующей форме.

Вначале вводится набор **критериев**, по которым производится оценка объектов. Критерий представляет собой конечный набор из элементов, называемых **градациями**. Градации критерия – абстрактные объекты, они могут быть числами (если критерий количественный, например цена), устными описаниями признака („малый“, „средний“, „большой“) и т.д. – важно, чтобы они по возможности охватывали все возможные формы проявления признака, который оценивается критерием.

Каждой градации критерия ставится в соответствие вещественное число от 0 до 1, называемое **полезностью** (или степенью предпочтительности) этой градации.

Набор градаций вместе с их полезностями часто называют **шкалой критерия**.

Пусть число критериев равно  $M$ , число градаций  $i$ -ого критерия ( $i = 1, \dots, M$ ) равно  $\text{gradnumber}(i)$ , а полезность  $j$ -ой градации в  $i$ -ом критерии равна  $u_i^j$ .

---

<sup>1</sup>Работа осуществлена при поддержке РФФИ. Гранты №№:

1. 06-01-00576-а Поддержка принятия решений в слабоструктурированных предметных областях: анализ ситуаций и оценка альтернатив.
2. 07-01-13516-офи\_ц Гибридные интеллектуальные системы поддержки принятия решений и обработки информации.
3. 07-01-00782-а Гибридные интеллектуальные системы поддержки принятия решений.
4. Титова Н.В. благодарит также Фонд содействия отечественной науке.

*Замечание.* С этого момента без ограничения общности считаем, что объект у нас один, так как все шаги алгоритма ранжирования, кроме последнего, пятого, применяются к каждому из объектов поодиночке, а на пятом шаге уже ранжируются сами объекты. Тем самым в задаче станет на один индекс меньше, что упрощает запись.

**Э т а п 2.** Далее в задачу вводятся несколько **экспертов**. Пусть их число равно  $N$ . Цель эксперта – оценить объекты по критериям, то есть для каждого объекта указать, какая градация ему соответствует на каждой из шкал критериев. При самом удачном исходе, эксперт абсолютно уверен в оценке и указывает для объекта единственную градацию на каждой из шкал. Если же эксперт не уверен в оценке, то он дает **нечеткие оценки** по критериям. В общем случае, оценка имеет следующую форму: эксперт называет для каждой градации каждой шкалы число от 0 до 1, называемое **степенью уверенности** в данной градации (или степенью правдоподобия данной градации). В дальнейшем эксперты (или, что то же самое, наборы их оценок) будут обозначаться большими латинскими буквами  $A_1, A_2, \dots, A_N$ , а правдоподобие  $j$ -ой градации  $i$ -ого критерия для нашего объекта, названное экспертом  $A$ , обозначим  $p_{ij}^A$ .

Если все градации с ненулевыми степенями правдоподобия лежат внутри некоторого интервала или объединения интервалов, то оценка называется **интервальной**.

Если такая градация только одна (ее степень правдоподобия равна 1), то оценка называется **точечной**.

**Э т а п 3.** Таким образом, появляется некоторое упорядочивание объектов по предпочтительности, но только в пределах любого отдельно взятого критерия. Следующим шагом в теории принятия решений является построение т.н. **обобщенной шкалы**, на которой расположены все градации всех критериев, и каждой градации опять же соответствует новое значение полезности, задающее ее положение на обобщенной шкале. (Проще говоря, необходимо разбить набор из всех градаций всех критериев на классы предпочтительности  $C_1, C_2, \dots, C_p$ , то есть ранжировать все градации точно так же, как это в итоге предполагается сделать с набором объектов.)

Способов вычисления полезностей на обобщенной шкале на сегодняшний день существует несколько. Все они, очевидно, так или иначе требуют ввода дополнительной информации о критериях (так иначе будет неясно, как градации из разных критериев соотносятся между собой, чтобы вычислять глобальную полезность градации по ее локальным полезностям). В данной работе применяется т.н. способ с обобщенной шкалой.

В этом способе дополнительно требуется попарное сравнение каждого критерия с каждым. Ответ дается в форме указания одной из 9 возможных степеней превосходства – от 1 (нет превосходства) до 9 (абсолютное превосходство). Далее эти степени сравнения записываются в **матрицу попарных сравнений** размера  $M \times M$ , где на позиции  $(i, j)$  стоит степень превосходства  $i$  над  $j$ , если  $i$ -ый объект полезнее  $j$ -ого, или единица, деленная на эту степень, – в противном случае.

В [1] показано, что собственный вектор этой матрицы, соответствующий ее максимальному по модулю собственному числу (и затем нормализованный), имеет определенный физический смысл: на  $i$ -ой позиции у него стоит **вес**  $i$ -ого критерия, то есть число от 0 до 1, тем большее, чем большие степени превосходства над другими критериями проявлены данным критериям при сравнениях.

Далее локальная полезность каждой градации домножается на вес критерия, к которому она принадлежит, и полученное число берется в качестве глобальной полезности градации.

Э т а п 4. Далее необходимо агрегировать голоса экспертов по каждому отдельно взятому объекту, то есть из набора всех нечетких оценок, данных всеми экспертами по этому объекту, получить ровно одну нечеткую оценку того же вида, в известном смысле усредняющую мнение всех экспертов. При более детальном рассмотрении задачи становится ясно, что на самом деле логичнее не агрегировать все оценки в одну, а **кластеризовать** их – то есть разбить на группы сходных оценок, затем уже каждую группу (кластер) усреднить (при этом не произойдет особой потери информации, так как все оценки внутри кластера близки) и для каждого среднего значения кластера отдельно запустить шаг 5 решения. Тем самым программа, решающую задачу ранжирования, выдаст несколько ответов, из которых предлагается вручную выбрать нужный.

При усреднении же мнений на первом шаге (например, медианой Кемени), без кластеризации, велик шанс получить решение задачи, мало соответствующее истинному оптимальному результату. Особенно неточен результат, когда мнения экспертов разделяются на четкие, сильно различающиеся между собой группы.

Кластеризация мнений экспертов – отдельная содержательная задача. В математической постановке она выглядит так: мнение каждого эксперта (по всем критериям вместе) – это точка в многомерном пространстве, являющимся декартовым произведением  $N$  векторных пространств ( $N$  – число критериев), у каждого из которых столько же измерений, сколько градаций у соответствующего критерия. Действительно, на место каждой градации каждого критерия ставится вещественное число из  $[0, 1]$  (правдоподобие), а норма во всем пространстве вычисляется так:

(пусть степень уверенности эксперта  $X$  в том, что объекту соответствует  $j$ -ая градация  $i$ -ого критерия, равна  $p_{ij}^X$ , а вес  $i$ -ого критерия равен  $w_i$ )

$$d(A, B) = \sqrt{w_1 ((p_{11}^A - p_{11}^B)^2 + (p_{12}^A - p_{12}^B)^2 + \dots) + w_2 ((p_{21}^A - p_{21}^B)^2 + (p_{22}^A - p_{22}^B)^2 + \dots) + \dots}$$

Отсюда видно, что если предварительно домножить каждую координату точки на корень из соответствующего веса, то итоговое пространство превращается в евклидово.

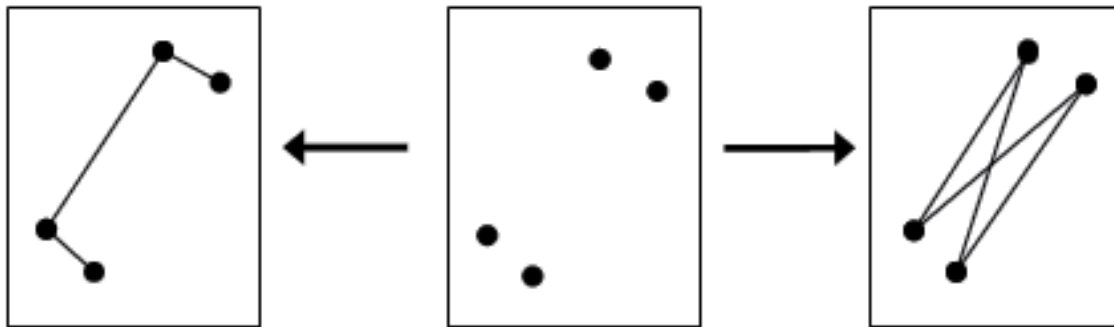
Данный способ удобен тем, что учитывает веса критериев и одновременно дает преимущества работы с евклидовой нормой – в частности, возможность использовать алгоритм QHull (см. ниже).

Итого, имеем классическую задачу кластеризации набора точек в евклидовом пространстве. Алгоритмов ее решения много, и одним из наиболее точных (и наиболее медленных, так как он принадлежит к т.н. иерархическим) является класс графовых алгоритмов. Они работают на следующем принципе: если взять некоторый связный граф, накрывающий наше множество точек, то при выкидывании из него ребер он будет распадаться на несвязные компоненты. Эти компоненты и берутся за искомые кластеры. Выкидывать же логично самые длинные ребра, так как самые короткие ребра должны соединять близкие точки в компонентах связности. (В качестве длины ребра берется евклидово расстояние между точками, им соединенными.)



Поэтому графовые алгоритмы строят некоторый покрывающий граф (или считают, что этот граф задан заранее), а затем выбрасывают из него ребра, оставляя самые короткие, пока не будет получено нужное число компонент связности или не выполнится какое-нибудь другое условие остановки алгоритма.

Так как разных графов, покрывающих данное множество точек, много, то встает вопрос, как выбрать из них тот, который минимален по суммарной длине ребер:



(Если взять граф, не минимальный по сумме длин ребер, то выкидывание из него длинных ребер даст неадекватную кластеризацию, как в случае на правой картинке.)

В классе графовых алгоритмов эту задачу решают те алгоритмы, которые для данного графа строят его **минимальное остовное дерево**, далее MST (minimal spanning tree) [4]. MST – это связный подграф данного графа, покрывающий все его вершины и минимальный по суммарной длине своих ребер. Этот класс методов имеет два преимущества:

- 1) Для дерева очень легко вычислить число компонент связности, получаемых при выбрасывании ребер: выбрасывание  $n$  ребер расщепляет дерево на  $n+1$  компоненту.
- 2) MST оптимально по сумме длин ребер, важность чего уже отмечалась выше.

Сложность задачи в том, что алгоритмы построения MST применяются к графу с известными длинами ребер, а не ко множеству точек. Если же попытаться создать этот покрывающий граф самым очевидным образом – нарисовать полный граф по точкам, то задача будет корректно решена, но это потребует  $N^2$  вычислений длины, не считая затрат на сам алгоритм.

Итак, необходимо придумать способ построения такого покрывающего графа, в котором было бы по возможности меньше ребер и сама длина их была бы как можно меньше. Опять же, самый естественный путь – это строить покрывающий граф, в котором ребрами соединяются только ближайшие точки, а дальние не соединяются. Такая задача решается с помощью **триангуляции Делоне** ([2]).

**Определение.**  $n$ -симплексом (в  $n$ -мерном евклидовом пространстве) называется выпуклая оболочка  $n+1$  точек, не лежащих в одной гиперплоскости.

Тем самым:

0-симплекс – это точка;

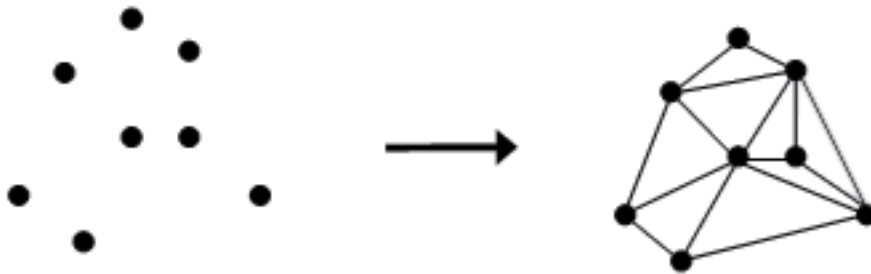
1-симплекс – это отрезок;

2-симплекс – это треугольник;

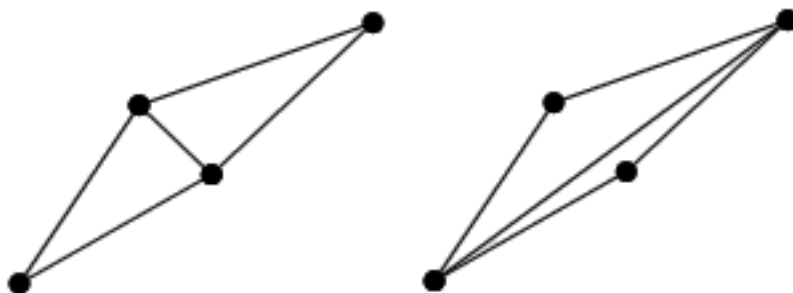
3-симплекс – это тетраэдр.

Выпуклая оболочка любых  $m$  из  $n$  точек сама является симплексом и называется  **$m$ -гранью симплекса**. 0-грани – это вершины, 1-грани – рёбра,  $(n-1)$ -грани называют просто гранями.

**Определение.** *Триангуляция* набора точек в  $n$ -мерном евклидовом пространстве – это разбиение их выпуклой оболочки на непересекающиеся  $n$ -симплексы.



**Определение.** *Триангуляция Делоне* – это такая триангуляция, что если около каждого ее симплекса описать гиперсферу, то ни одна из гиперсфер не будет содержать внутри себя других точек набора. (В плоском случае гиперсферы – это описанные около треугольников окружности.)



Слева – триангуляция Делоне, справа – триангуляция, ей не являющаяся.

Из определения понятно, что слишком далекие точки в триангуляции не соединяются ребрами (иначе бы описанная сфера была слишком большой и содержала в себе другие точки).

**Лемма.** (из [2]) *Триангуляция Делоне единственна, если в наборе точек нет ни одного поднабора из  $n+2$  точек, лежащих на одной гиперсфере (на плоскости это четверка точек, лежащих на одной окружности).*

**Лемма** (О локальной оптимальности триангуляции Делоне). *В триангуляции Делоне каждая точка гарантированно соединяется ребрами с двумя ближайшими к ней точками.*

**Лемма** (О вложении MST в триангуляцию). *Все ребра MST, покрывающего данное множество точек на плоскости, являются ребрами триангуляции Делоне этого множества точек.*

В программе, реализующей этот алгоритм, триангуляция Делоне строится с помощью программы Qhull ([www.qhull.org](http://www.qhull.org)) [3] – продукта с открытым кодом, реализующем быстрые методы построения выпуклой оболочки, триангуляции Делоне и диаграммы Вороного. Он применяется, в частности, в таких пакетах, как Matlab и Mathematica. Триангуляция в нем строится за  $O(N \ln N)$  шагов.

Далее для построения MST используется **алгоритм Краскала** построения MST (см. [4]). После построения встает вопрос, сколько ребер выбрасывать.

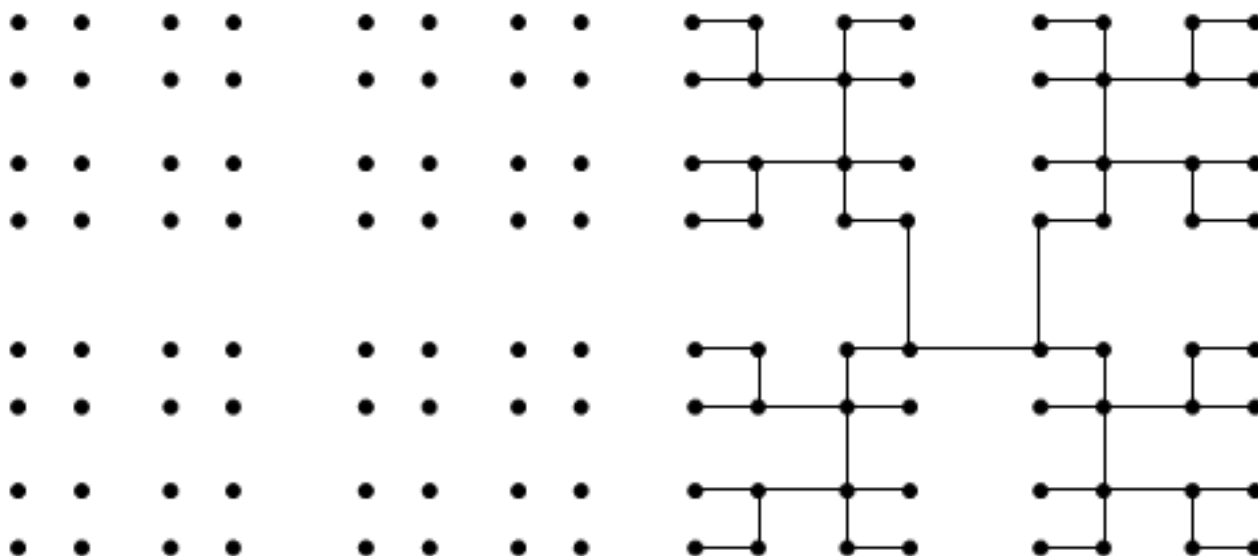
В работе для этого предлагается следующая стратегия: вначале все ребра MST сортируются по длине.

$$l_1, l_2, \dots, l_p \quad (l_1 \leq l_2 \leq l_3 \dots \leq l_p)$$

Затем по этому массиву длин вычисляется массив из относительных скачков длин:

$$\frac{l_2 - l_1}{l_1}, \frac{l_3 - l_2}{l_2} \dots \frac{l_p - l_{p-1}}{l_{p-1}}$$

Затем выбрасываются все ребра, которые длиннее того ребра, на котором произошел наибольший скачок (вместе с ним же). Если же есть несколько ребер с одинаковыми скачками, как, например, на рисунке:



... то выбрасывается минимально возможное число ребер, чтобы минимизировать число кластеров для большего удобства дальнейшей работы. К примеру, на данном рисунке из возможных вариантов разбиения на 4, 16 или 64 кластера в алгоритме выбирается первый вариант (4 кластера).

Прежде чем переходить к следующему шагу алгоритма, необходимо разрешить еще одну проблему. Расстояние между голосами экспертов  $X$  и  $Y$  вычисляется как расстояние между точками евклидова пространства размерности  $\text{gradnumber}(1) + \dots + \text{gradnumber}(M)$ . Иными словами, у этого пространства **столько же измерений, сколько в сумме существует градаций у всех критериев**. С учетом того, что для построения триангуляции Делоне в  $n$ -мерном пространстве нужен набор из не менее чем  $n + 2$  точек<sup>2</sup>, а всех градаций с ненулевыми степенями уверенности  $p$ , как правило, гораздо больше голосов экспертов, то перед нами встает вопрос, как избежать этой ситуации (когда практически в любой задаче голосов оказывается слишком мало для триангуляции).

*Способ 1:* Линейная оболочка  $M$  векторов в  $N$ -мерном линейном векторном пространстве (считаем  $M \ll N$ ) представляет собой не более чем  $M$ -мерную гиперплоскость, значит можно уменьшить размерность пространства поиска до  $M$  и добавить при необходимости один или два „фиктивных“ голоса (не совпадающих ни с одним из предыдущих) – тогда выполнится соотношение  $M \geq N + 2$ . Заметим, что на асимптотику числа необходимых вычислений добавление к  $M$  константы не повлияет.

*Способ 2:* Вводится метод перехода от исходного многомерного пространства голосов экспертов в пространство „нечетких точек“. Именно, будем считать каждый голос эксперта набором точек в пространстве  $\mathbb{R}^M$  ( $M$  – число критериев). Это пространство является декартовым произведением всех шкал градаций друг на друга (считаем градации вещественными числами; если же у нас шкала с дискретными градациями неколичественной природы, то заменяем каждую градацию шкалы на ее порядковый номер и погружаем шкалу в вещественную ось).  $k$ -ая координата этого пространства соответствует шкале  $k$ -ого критерия. Каждой точке из голоса эксперта приписана ее мера достоверности  $p$ , равная произведению достоверностей ее координат.

С введением этого пространства  $\mathbb{R}^M$  мы получаем не что иное, как **задачу нечеткой кластеризации**.  $\mathbb{R}^M$  не только обладает малой размерностью, но и дает возможность предварительно уменьшить число кластеризуемых точек без потери информации. Это обеспечивается следующими леммами.

**Лемма (О блоках).** *Для любого голоса эксперта точки, которым присвоена ненулевая достоверность  $p$ , образуют объединение  $M$ -мерных дискретных параллелепипедов с гранями, параллельными координатным плоскостям.*

**Лемма (О близости).** *В MST точки внутри каждого блока будут лежать в одной и той же компоненте связности.*

**Лемма (О центре тяжести).** *Если за центр кластера точек с радиус-векторами  $r_1, \dots, r_n$  и мерами достоверности  $p_1, \dots, p_n$  брать точку  $\frac{p_1 r_1 + \dots + p_n r_n}{p_1 + \dots + p_n}$  с достоверностью  $p_1 + \dots + p_n$ , то результат вычисления центра кластера не зависит от порядка слагаемых и группировки слагаемых в промежуточные кластеры внутри основного. (то есть можно вначале вычислять центры промежуточных подкластеров, выбираемых произвольным образом, и считать их затем за новые точки.)*

После перечисления лемм ясно, что если вначале заменить все блоки на их центры тяжести, а затем для этих центров начать решать задачу триангуляции, то результат не будет отличаться от результата решения задачи, если бы мы в задаче триангуляции стали обрабатывать все точки сразу, без предварительного уменьшения их числа.

После разбиения на кластеры и замены каждого кластера на его центр можно переходить к шагу 5.

<sup>2</sup>В противном случае приходится строить MST из полного графа по точкам, то есть остановиться на сложности  $O(n^2)$  построения исходного графа для поиска MST вместо  $O(n \cdot \ln n)$ .

Э т а п 5. Вообще говоря, обобщенная шкала уже может дать нам некоторое ранжирование объектов, так как глобальную функцию полезности объекта можно определить уже на этом шаге (например, просуммировав полезности его градаций – так называемая аддитивная функция полезности), а если есть функция полезности – значит, есть и ранжирование. Кроме того, ранжирование уже может быть у нас в распоряжении и без построения обобщенной шкалы (например, при использовании метода средней полезности или медианы Кемени).

Проблема в том, что нечеткое отношение, задаваемое этим ранжированием, может не является транзитивным и рефлексивным, что с практической точки зрения обозначает логическое противоречие в ранжировании. Более точно, верны следующие 2 леммы [5]. Пусть  $A = (a_{ij})$  – матрица нечеткого отношения, тогда:

**Лемма.** *Нечеткое отношение транзитивно тогда и только тогда, когда не существует таких элементов  $i, j, k$ , что  $j$  предпочтительнее  $i$ ,  $k$  предпочтительнее  $j$ ,  $i$  предпочтительнее  $k$  на дереве отношения.*

**Лемма.** *Транзитивное нечеткое отношение рефлексивно тогда и только тогда, когда не существует таких элементов  $i, j, k$ , что  $k$  одинаково эквивалентен объектам  $i$  и  $j$ , а объект  $i$  предпочтительнее объекта  $j$  на дереве отношения.*

Как следствие, необходимо перейти к аппроксимации данного (вообще говоря, нереклексивного и нетранзитивного) отношения ближайшим к нему в некоторой метрике рефлексивным транзитивным отношением (**нечеткой квазисерией**). В качестве метрики, то есть меры расстояния между двумя матрицами отношений, традиционно берется **чебышевская метрика**:

$$\mu(A, B) = \max_{i,j} |a_{ij} - b_{ij}|.$$

Итак, стоит следующая задача оптимизации. Пусть  $A = (a_{ij})$  – приближаемая матрица,  $A^*$  – ее транзитивное замыкание,  $T = (t_{ij})$  – произвольная аппроксимирующая матрица,  $\hat{T} = (\hat{t}_{ij})$  – оптимальная аппроксимирующая матрица:

$$\begin{aligned} \mu &= \mu(A, B) \rightarrow \min; \\ a_{ij} - \mu &\leq t_{ij} \leq a_{ij} + \mu; \\ t_{ij} &\geq 1, \quad t_{ii} = 1, \quad t_{ij} + t_{ji} = 1. \end{aligned}$$

Используемый метод аппроксимации описан в [5, 11]. Применив его, мы получаем на выходе искомую квазисерию, что завершает решение задачи ранжирования.



## Список литературы

- [1] *Т.Саати* Принятие решений – Метод анализа иерархий. – М.: Радио и связь, 1993.
- [2] *Ф.Препарата, М.Шеймос* Вычислительная геометрия. – М.: Мир, 1989.
- [3] *С.В.Barber, D.P. Dobkin, H. T. Huhdanpaa* The Quickhull Algorithm for Convex Hulls. ACM, Transactions on Mathematical Software (TOMS). – New York, 1996.
- [4] *А.Ахо, Д.Хопкрофт, Д.Ульман* Структуры данных и алгоритмы. – Пер. с англ.: М., Издательский дом "Вильямс", 2003.
- [5] *Макеев С.П., Шахнов И.Ф.* Упорядочение объектов в иерархических системах. Известия АН СССР. Технич. кибернет. № 3. – М., 1991.
- [6] *Бурков В.Н., Новиков Д.А.* Как управлять организациями. – М.: Синтез, 2003.
- [7] *Kosko B.* Fuzzy thinking. – New York: Hyperion, 1993.
- [8] *Сылов В.Б.* Принятие стратегических решений в нечеткой обстановке. – М.: ИНПРО-РЕС, 1995.
- [9] *Фестингер Л.* Теория когнитивного диссонанса. – СПб.: Ювента, 1999.
- [10] *Кулмич А.А.* Методология когнитивного моделирования сложных плохо определенных ситуаций. Труды второй международной конференции по проблемам управления. – М.: ИПУ РАН, 2003.
- [11] *Шахнов И.Ф.* Некоторые вопросы предпланового анализа крупномасштабных проектов. Исследование операций (модели, системы, решения). – М.: ВЦ РАН, 2002.