

УДК 519.161

МАТЕМАТИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

О.Я. Никонов, доцент, д.т.н., О.А. Подоляка, доцент, к.т.н., ХНАДУ, А.Н. Подоляка, ст. преподаватель, НАУ «ХАИ», Е.В. Скакалина, к.т.н., начальник отдела внедрения и эксплуатации информационных систем, НефтеГазоДобывающее Управление «Полтаванефтегаз»

***Аннотация.** Рассматривается метод решения многокритериальной задачи о назначениях, основанный на алгоритмах порядковой нормализации критериев, достоинством которого является существенное упрощение процедуры построения векторных решений и их сравнения, а также процесс свертки функционалов многокритериальных задач.*

***Ключевые слова:** многокритериальная задача о назначениях, векторная целевая функция, нормализация критериев.*

МАТЕМАТИЧНІ МЕТОДИ РОЗВ'ЯЗАННЯ БАГАТОКРИТЕРІЙНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

О.Я. Ніконов, доцент, д.т.н., О.О. Подоляка, доцент, к.т.н., ХНАДУ, О.М. Подоляка, ст. викладач, НАУ «ХАІ», О.В. Скакаліна, к.т.н., начальник відділу впровадження і експлуатації інформаційних систем, НафтоГазоВидобувне Управління «Полтаванафтогаз»

***Анотація.** Розглядається метод розв'язання багатокритерійної задачі про призначення, що базується на алгоритмах порядкової нормалізації критеріїв, достоїнством якого є істотне спрощення процедур побудови векторних розв'язків та їх порівняння, а також процес згортки функціоналів багатокритеріальних задач.*

***Ключові слова:** багатокритеріальна задача про призначення, векторна цільова функція, нормалізація критеріїв.*

MATHEMATICAL METHODS OF SOLVING MULTICRITERION ASSIGNMENT PROBLEM

O. Nikonov, Associate Professor, Doctor of Technical Science, O. Podolyaka, Associate Professor, Candidate of Technical Science, KhNAHU, A. Podolyaka, senior lecturer, NAU «KhAI», E. Skakalina, Candidate of Technical Science, Head of department of Implementation and Operation of Information Systems, Oil and Gas Production Department «Poltavaneftgaz»

***Abstract.** The method of solving multicriterion assignment problem based on the algorithms of ordinal normalization criteria, the main advantage of which is significant simplification of the vector solutions construction and their comparisons as well as the process of convolution of multicriterion problems functionals is considered.*

***Key words:** multicriterion assignment problem, vector objective function, criteria normalization.*

Введение

Результаты исследования задач планирования и управления на транспорте показывают, что в реальной постановке эти задачи являются многокритериальными, т.к. характеризуются множеством различных параметров или критериев. В связи с этим традиционные методы решения задач оказываются малоэффективными. Эти задачи, во-первых, требуют высокой квалификации персонала, который формирует критерии и данные, определяет цели и оценивает качество полученных решений, а во-вторых, они характеризуются большой вычислительной сложностью.

В качестве критериев оптимальности транспортных задач в первую очередь выступают количественные характеристики перевозочного процесса: стоимостные параметры доставки грузов, объемно-массовые характеристики груза, временные параметры, технические характеристики транспортных средств, характеристики персонала и т.п. Критерии задач имеют различные единицы измерения и шкалы, что существенно усложняет процедуры построения и сравнения многокритериальных решений.

Представление математических моделей транспортного процесса в виде многокритериальных постановок транспортных задач представляет естественный процесс формирования цивилизованного рынка транспортных услуг в Украине и естественное стремление транспортных предприятий удовлетворить интересы всех участников транспортного бизнеса.

Анализ публикаций

Существующие на сегодняшний день методы многокритериальной оптимизации условно можно разделить на две группы. Методы первой группы сводят многокритериальную задачу к однокритериальной путем свертывания векторного критерия в суперкритерий, который затем оптимизируется одним из методов однокритериальной оптимизации. Существуют различные виды свертки [1]. С помощью этих методов можно решать сложные задачи большой размерности, т.к. они характеризуются высокой вычислительной эффективностью. Однако в теории принятия решений принято считать, что решения, полученные этими методами, являются недостаточно эффективными.

Ко второй группе можно отнести остальные методы многокритериальной оптимизации [1, 2]:

- справедливого компромисса;
- приближения к идеальному решению;
- метод последовательных уступок;
- анализа иерархий;
- генетические алгоритмы;
- нейронные сети.

Данные методы позволяют получать качественные решения, но характеризуются большой сложностью и требуют высочайшей квалификации персонала. Это, в основном, не полиномиальные методы и человеко-машинные процедуры, оценить время выполнения которых не представляется возможным.

Для решения многокритериальной задачи о назначениях (МЗН) в работе применяется метод свертки функционала на основе выбора доминирующего критерия. Чтобы улучшить качество решений, предлагается использовать подготовительный этап, в основе которого лежат алгоритмы нормализации критериев задачи. Данный метод решения МЗН характеризуется высоким быстродействием, так как в нем используются эффективные полиномиальные алгоритмы нормализации и решения задачи о назначениях.

Следует отметить, что основной проблемой нормализации является слабая оценка важности критериев, с точки зрения возможности построения решений задачи. Наиболее остро данная проблема проявляется, когда критерии задачи имеют различные шкалы (т.е. множество значений одного критерия состоит из малого числа элементов, а множество значений другого – из большого числа элементов). Эти особенности практически не учитываются в известных алгоритмах нормализации [1–4] и поэтому кардинально усложняют схему поиска компромисса при решении многокритериальной задачи. Известные аддитивные или мультипликативные схемы нормализации и методы поиска компромисса не в состоянии адекватно решить эту проблему [3–6].

Для решения вышеперечисленных проблем предлагается определить следующие характеристики ценности критерия, которые должны учитываться при его нормализации.

1) Нормализованное значение должно отражать величину выигрыша (важности) при использовании этого элемента в решении.

2) Нормализация критерия должна отображать размерное значение веса критерия в безразмерное компактное счетное множество значений. Если необходимо, значения элементов этого множества можно отобразить в диапазон $[0,1]$.

3) Нормализация не должна менять порядок следования решений или, по крайней мере, изменять его незначительно.

Цель и постановка задачи

Данное исследование направлено на разработку математического метода решения многокритериальной задачи о назначениях (МЗН), в котором лицо, принимающее (ЛПР), будет играть свою роль только на этапе подготовки исходных данных и выбора приоритетных целей, а также на заключительном этапе выбора решения задачи из полученного множества альтернативных решений. Этап подготовки данных означает выбор ЛПР метода нормализации соответствующего критерия, который отражает подходящую, в данном случае, схему значимости критерия. Если решение задачи по каким-либо причинам не устраивает ЛПР, он может изменить значения соответствующих критериев на основе одной из трех схем нормализации, отражающих: пессимизм, оптимизм или средневзвешенную оценку и построить новое решение. Следует отметить, что разработанные алгоритмы нормализации преобразуют взвешенные значения критериев к единой шкале, что существенным образом упрощают процедуру свертки критериев в один суперкритерий.

Рассмотрим математические постановки задач.

Задача 1. Классическая задача о назначениях

Необходимо найти совершенное паросочетание (перестановку) $\pi^* \in P$ с минимальным весом для заданной матрицы β^{MM} , где M – порядок матрицы. Элемент матрицы β_{ij} представляет один критерий.

Функционал классической однокритериальной задачи о назначениях

$$\rho(\pi^*) = \min_{\pi} \sum_{i,j \in \pi} \beta_{ij}, \quad (1)$$

где $\rho(\pi) = \sum_{i,j \in \pi} \beta_{ij}$ – вес паросочетания π ;

β_{ij} – элемент матрицы, принадлежащий паросочетанию.

Задача 2. Многокритериальная задача о назначениях с множеством упорядоченных по важности критериев

$$\rho(\pi^*) = \min_{\pi} \sum_{i,j \in \pi} \beta_{ij}, \quad (2)$$

где $\rho(\pi) = \sum_{i,j \in \pi} \beta_{ij}$ – вес паросочетания π .

Элемент β_{ij} представляет собой вектор упорядоченных по важности критериев.

В многокритериальной постановке элемент матрицы β_{ij} представляет сложный тип данных, называемый вектором критериев.

$$\beta_{ij} = \{C_{ij}^k, k = \overline{1, L}\}, \quad (3)$$

где L – число критериев; C_{ij}^k – значение k -го критерия элемента решения β_{ij} .

Задача 3. Многокритериальная задача о назначениях с множеством равнозначных критериев

$$\begin{cases} \rho(\pi_1^*) = \min_{\pi_1} \sum_{i,j \in \pi_1} C_{ij}^1 \\ \rho(\pi_2^*) = \min_{\pi_2} \sum_{i,j \in \pi_2} C_{ij}^2 \\ \dots \\ \rho(\pi_k^*) = \min_{\pi_k} \sum_{i,j \in \pi_k} C_{ij}^k \end{cases} \quad (4)$$

где $\min_{\pi_k} \sum_{i,j \in \pi_k} C_{ij}^k$ – вес паросочетания π_k ;

C_{ij}^k – вес k -го критерия элемента матрицы β_{ij} , принадлежащего паросочетанию π_k .

Следует отметить, что C_{ij}^k – это размерные величины, которые могут быть различного рода, например: стоимость, температура, время, вес, расстояние и т.п. Они могут иметь различные единицы измерения, шкалы, эталоны и т.п. Нормализованное безраз-

мерное значение критерия обозначается малой буквой c_{ij}^k .

В отличие от однокритериальных задач, например, классической ЗН, решения многокритериальных задач противоречивы, поскольку невозможно с абсолютной уверенностью утверждать, что то или иное решение строго оптимально. Это объясняется тем, что в подавляющем большинстве случаев $\pi_1^* \cup \pi_2^* \cup \dots \cup \pi_k^* = \emptyset$. Поэтому в основе алгоритмов решения многокритериальных задач лежит некоторая схема поиска компромисса, при которой $\pi_1 \cup \pi_2 \cup \dots \cup \pi_k \neq \emptyset$.

Задача 4. Минимаксная задача о назначениях

Функционал задачи имеет вид

$$F_b = \min_{\pi \in P} \max_{i,j \in \pi} \beta_{ij}. \quad (5)$$

Он определяет ограничения на вес максимального элемента решения (его вес должен быть минимальным).

Задача имеет важное практическое приложение, например, в задачах оптимизации распределения задач в параллельных вычислительных системах (ПВС). Примером таких систем являются вычислительные кластеры. В таких системах основная задача разбивается на множество подзадач, решаемых на независимых параллельно работающих машинах, различающихся по своим техническим характеристикам. Общее время решения задачи в подобных системах определяется максимальным временем решения подзадачи. Функционал F_b минимизирует время решения самой длинной подзадачи, а значит, минимизирует время решения основной задачи.

Следует отметить, что задача оптимальной расстановки персонала на конвейере также описывается данной математической моделью. Пусть на некотором конвейере (поточной линии) имеется n рабочих мест. Руководитель должен расставить по своим местам n работников. Известна производительность труда C_{ij} i -го работника на j -м месте. Производительность конвейера определяется минимальной производительностью занятых на нём работников. Рабочее место, на котором реализуется минимальная производительность при заданной расстановке работников (другими словами, при заданном *назначе-*

нии), и есть узкое место этого назначения. Таким образом, необходимо найти такое распределение работников по рабочим местам, при котором скорость конвейера (т.е. минимальная производительность занятых работников) будет максимальной.

Для нас важно, что эта задача находит применение в алгоритмах решения многокритериальных задач, поэтому в данной работе она рассматривается как один из методов свертки функционала многокритериальной задачи.

Если $\beta_{ij} = \{c_{ij}^k, k = \overline{1, L}\}$ – вектор нормализованных критериев, то оператор $\max_{i,j \in \pi} \beta_{ij}$ функционала (5) определяет максимальный критерий c_{ij}^k (худший критерий) каждого элемента β_{ij} . Т.е. этот оператор реализует сведение задачи к однокритериальной путем выделения доминирующего критерия. Затем оператор $\min_{\pi \in P}$ решает классическую ЗН – реализует выбор лучших из худших на множестве перестановок P . Следует отметить, что минимаксная стратегия весьма пессимистична.

Функционал минимаксной МЗН имеет вид

$$F_b = \min_{\pi \in P} \max_{\substack{k=\overline{1,L} \\ i,j \in \pi}} c_{ij}^k. \quad (6)$$

Минимаксная стратегия используется для решения МЗН в данной работе. Следует отметить, что перед решением минимаксной задачи выполняется сложный подготовительный этап нормализации, вычислительная сложность которого составляет $N^3 \log N$. Она превосходит сложность эффективных алгоритмов решения классической ЗН, например, алгоритма Эдмондса-Карпа, имеющего сложность N^3 . Поэтому пессимизм данной стратегии построения решения здесь оправдан, так как алгоритмы нормализации, представленные в работе, точно определяют ценность (важность) элементов матрицы β_{ij} .

Метод решения многокритериальной задачи

На начальном этапе решения МЗН производится нормализация критериев. В ее основе лежит следующая вычислительная схема.

Оценка важности критерия классической задачи о назначениях

В начале определим схему вычисления выигрыша G_{ij} при использовании элемента матрицы β_{ij} классической однокритериальной задачи о назначениях (1).

Использование элемента β_{ij} в решении задачи приводит к исключению всех остальных элементов в строке i и столбце j . Выигрыш вычисляется как сумма разностей значений элемента β_{ij} и всех элементов столбца j . Чтобы определить выигрыш элементов строки i относительно всех строк матрицы, нужно вычислить разность этой строки и всех строк. Отметим, что вычисление выигрыша элементов одной строки не выполняется, т.к. разность $\beta_{ij} - \beta_{ij} = 0$. Т.е. при расчете выигрышей всех элементов строки i получается нулевая строка, в которой суммарный выигрыш равен нулю. Поэтому формула определения выигрыша имеет вид

$$G_{ij} = \sum_{r=0}^{N-1} (\beta_{rj} - \beta_{ij}) \quad (7)$$

$G_{ij} = \sum_{r=0}^{N-1} (\beta_{rj} - \beta_{ij}) = \sum_{r=0}^{N-1} \beta_{rj} - \sum_{r=0}^{N-1} \beta_{ij} = S_j - N\beta_{ij}$,
где $S_j = \sum_{r=0}^{N-1} \beta_{rj}$ – сумма элементов столбца матриц ЗН.

Функционал ЗН для матриц выигрышей G имеет вид

$$\begin{aligned} F_G &= \sum_{i,j \in \pi} G_{ij} \rightarrow \max; \\ F_G &= \sum_{i,j \in \pi} G_{ij} = \sum_{i,j \in \pi} (S_j - N\beta_{ij}) = \\ &= \sum_{j=0}^{N-1} S_j - N \sum_{i,j \in \pi} \beta_{ij} = S - N \sum_{i,j \in \pi} \beta_{ij}; \end{aligned} \quad (8)$$

где константа $S = \sum_{j=0}^{N-1} S_j$ – сумма элементов матрицы.

$$F_G = \sum_{i,j \in \pi} (-\beta_{ij}) \rightarrow \max$$

Тогда

$$F = \sum_{i,j \in \pi} \beta_{ij} \rightarrow \min \Leftrightarrow F_G = \sum_{i,j \in \pi} G_{ij} \rightarrow \max$$

$$F = \sum_{i,j \in \pi} (-G_{ij}) \rightarrow \min. \quad (9)$$

На основании свойств ЗН [1] матрицы $(-1)G$ и β эквивалентны, т.е. имеют одно и то же множество оптимальных решений.

Оценка важности критерия многокритериальной задачи о назначениях

В основе метода решения многокритериальной задачи о назначениях лежат алгоритмы нормализации критериев, которые удовлетворяют требованиям оценки важности критериев, приведенных выше. Важность критерия или соответствующего элемента матрицы назначений определяется величиной выигрыша, которая достигается при использовании этого критерия в решении задачи. Алгоритмы нормализации вычисляют значение этого выигрыша. Они используют аддитивную схему определения важности критерия.

Главным отличием алгоритма нормализации от известных схем является предварительный этап, на котором реализуется отображение критерия в безразмерное компактное счетное множество значений. Каждый элемент этого множества находится в диапазоне $0, (N-1)^2$, где N – порядок матрицы. Это отображение реализуют алгоритмы порядковой сортировки. Имеется три таких алгоритма: LOS (Less order sort), GOS (Greater order sort) и MOS (Mean order sort).

Алгоритмы порядковой сортировки

Пусть:

V – исходный массив;

I – массив упорядоченных индексов;

$\text{less_count}(V, V[i])$ – функция определения количества элементов массива V , меньших $V[i]$;

$\text{greater_count}(V, V[i])$ – функция определения количества элементов массива V , больших $V[i]$;

$\text{count}(V, V[i])$ – функция определения количества элементов массива V , равных $V[i]$.

Именование алгоритмов порядковой сортировки осуществляется на основании используемой в них функции определения количества элементов.

LESS_ORDER_SORT (LOS)

Массив I вычисляется по формуле

$$I[i] = \text{LOS}(V, V[i]) = \\ = \text{less_count}(V, V[i]), i \in \overline{0, N-1}$$

Пример. Порядковая сортировка LOS

№	0	1	2	3	4	5	6	7
V	5	7	7	8	6	7	8	6
I	0	3	3	6	1	3	6	1

GREATER_ORDER_SORT (GOS)

Массив I вычисляется по формуле

$$I[i] = \text{GOS}(V, V[i]) = \\ = N - 1 - \text{greater_count}(V, V[i]), i \in \overline{N-1, 0}$$

Пример. Порядковая сортировка GOS

10	0	1	2	3	4	5	6	7
V	5	7	7	8	6	7	8	6
I	0	5	5	7	2	5	7	2

MEAN_ORDER_SORT (MOS)

Индекс, вычисленный алгоритмом MOS, – это среднее значение индексов алгоритмов LOS и GOS.

$$I[i] = \text{MOS}(V, V[i]) = \\ = (\text{LOS}(V, V[i]) + \text{GOS}(V, V[i])) / 2$$

Свойства алгоритмов порядковой сортировки LOS и GOS

1. Индексы неповторяющихся элементов массива V , вычисленные алгоритмами LOS, GOS и MOS, будут одинаковыми. И наоборот, индексы повторяющихся элементов массива V , вычисленные алгоритмами LOS, GOS и MOS, будут различными.
2. Алгоритмы LOS и GOS симметричны относительно начала отсчета и методов определения количества элементов `less_count` и `greater_count`.
3. Массив V отсортирован алгоритмом GOS. Если элемент $V[i]$ встречается в массиве V один раз и его индекс $I[i]$, $i > 0$, то в массиве I всегда имеется элемент с индексом $(I[i] - 1)$.
4. Массив V отсортирован алгоритмом GOS. Если элемент $V[i]$ встречается в массиве V n раз ($1 < n < N$) и его индекс $I[i]$, то в массиве I всегда имеется элемент с индексом $(I[i] - n)$.

5. Индексы всех равных $V[i]$ элементов массива V , вычисленные алгоритмом LOS, всегда будут меньше индексов, вычисленных алгоритмом GOS.

6. Индекс $I[i]$, $i > 0$ элемента $V[i]$, вычисленный алгоритмом LOS, означает, что в массиве V есть ровно $I[i] + 1$ меньших $V[i]$ элементов и $N - I[i] - 1$ больших или равных $V[i]$ элементов.

7. Индекс $I[i]$ элемента $V[i]$, вычисленный алгоритмом GOS, означает, что в массиве V есть ровно $I[i] + 1$ меньших или равных $V[i]$ элементов и $N - I[i] - 1$ строго больших $V[i]$ элементов.

Основные зависимости

$$\begin{aligned} \text{LOS}(V, V[i]) + \text{GOS}(-V, -V[i]) &= N - 1 \\ \text{GOS}(V, V[i]) - \text{LOS}(V, V[i]) - \text{count}(V, V[i]) + 1 &= 0 \\ \text{LOS}(V, V[i]) &= \text{GOS}(V, V[i]) - \text{count}(V, V[i]) + 1 \\ \text{GOS}(-V, -V[i]) &= N - 2 + \text{count}(V, V[i]) - \text{GOS}(V, V[i]) \\ \text{MOS}(-V, -V[i]) &= N - 1 - \text{MOS}(V, V[i]) \\ \text{GOS}(V, V[i]) &= \text{MOS}(V, V[i]) - (\text{count}(V, V[i]) - 1) / 2 \\ \text{LOS}(V, V[i]) &= \text{MOS}(V, V[i]) + (\text{count}(V, V[i]) - 1) / 2 \end{aligned}$$

Алгоритм LOS

Следует отметить, что непосредственное вычисление массива I , по формуле

$$I[i] = \text{LOS}(V, V[i]) = \text{less_count}(V, V[i])$$

имеет квадратичную сложность. В работе представлен эффективный алгоритм расчета I , вычислительная сложность которого определяется сложностью алгоритма сортировки массива и равна соответственно $N \log N$. Рассмотрим этот алгоритм.

Каждый элемент массива V в алгоритме представлен классом `Line_pair`, атрибутами которого являются значение элемента `first` и индекс в исходном массиве `second`. Этот класс необходим, чтобы определить индексы элементов в исходном массиве после его сортировки. Под значением `first` в нашем случае понимается выигрыш при использовании соответствующего элемента в решении многокритериальной задачи.

Линейные пары (ЛП)

Пусть `Line_pair` – линейная пара, первый элемент которой `first` – это значение i -го элемента массива V , а второй – `second` – это его

индекс в массиве V ($V[i], i$). Линейные пары сравниваются по первому полю, т.е. по значению $V[i]$; Vlp – массив линейных пар.

```
class greater_Line_pair
{public:
    bool operator()(const Line_pair& l,
                    const Line_pair& r) const
    {return l.first > r.first;};};
```

Реализация алгоритма порядковой сортировки LOS на C++

```
index_sort(vector<Line_pair>& Vlp, vector<size_type>& I) //I - массив результат
//1.Сортировка массива Vlp
sort(Vlp.begin(), Vlp.end(), greater_Line_pair());
//2. Порядковая сортировка, расчет массива I
// число меньших Vlp[i]
size_type less_count;
// число равных Vlp[i]
size_type equal_count;
size_type i;
size_type N = Vlp.size();
//индекс максимального = 0
I[Vlp[0].second] = less_count;
for(i = 1; i < N; ++i)
    equal_count = 1;
// если текущая ЛП равна по значению
// предыдущей
while(Vlp[i-1].first == Vlp[i].first)
    I[Vlp[i].second] = less_count;
    ++equal_count;
    ++i;
// если ЛП не равны по значению
if(i < N)
    // определить индекс текущей ЛП
    less_count = less_count + equal_count;
    I[Vlp[i].second] = less_count;
```

Сложность представленного алгоритма определяется первым этапом «1. Сортировка массива V » и равна $N \log N$. Сложность второго этапа – линейная.

Алгоритм вычисления порядковой матрицы

Выше была рассмотрена схема вычисления выигрыша G_{ij} элемента решения β_{ij} , которая определялась по следующей формуле

$$G_{ij} = \sum_{r=0}^{N-1} (\beta_{rj} - \beta_{ij}).$$

В данной работе предлагается применять подобную схему расчета, но для определения

выигрыша использовать соответствующий алгоритм порядковой сортировки, например, LOS. В этом алгоритме минимальный элемент массива V имеет максимальный индекс, а значит, дает максимальный выигрыш.

В общем случае чем меньше элемент, тем больше его индекс или выигрыш. Поэтому разность $(\beta_{rj} - \beta_{ij})$ следует умножить на минус единицу, т.к. решается задача минимизации, и формула расчета выигрыша будет следующей

$$I_{ij} = \sum_{r=0}^{N-1} LOS([\beta_{rj} - \beta_{ij}], V^{ir}), \quad (10)$$

где I – *порядковая матрица*, в которой находятся аппроксимированные значения выигрышей элементов β_{ij} ; V^{ir} – массив, содержащий разности элементов строк i и r исходной матрицы β , т.е. k -й элемент массива рассчитывается по формуле

$$V_k^{ri} = \beta_{ik} - \beta_{rk}, \quad k = \overline{1, N}. \quad (11)$$

Строка i элемента β_{ij} называется *клиентской*, а вычитаемая строка r – *серверной строкой*.

Алгоритм $LOS([\beta_{rj} - \beta_{ij}], V^{ri})$ определяет индекс элемента $[\beta_{rj} - \beta_{ij}]$ или аппроксимированный выигрыш элемента β_{ij} для заданной пары строк i и r .

Свойства выигрыша алгоритма LOS

1. Порядок следования столбцов элементов $(\beta_{rj} - \beta_{ij})$, $j = \overline{1, N}$ после сортировки массива V^{ri} и порядковой сортировки $LOS([\beta_{rj} - \beta_{ij}], V^{ri})$, $j = \overline{1, N}$ будет одинаковым. Значит алгоритм LOS сохраняет порядок решений при попарных перестановках элементов решений ЗН.

2. При вычислении порядковой матрицы осуществляется преобразование выбранного критерия МЗН в конечное множество целых чисел (индексов), количество элементов которого находится в интервале $[0, (N-1)^2]$.

Данные алгоритма:

$s1$ и $s2$ – индексы некоторых строк матрицы;

V – массив, представляющий разность строк $s1$ и $s2$, который вычисляется функцией `construct_V(s1,s2)` по формуле (11);

I – массив упорядоченных индексов, который вычисляется соответствующим алгоритмом порядковой сортировки `order_sort(V)` на основании массива V ;

Im – матрица размера $M \times N$, содержащая значения выигрышей, рассчитанных алгоритмом порядковой сортировки для каждой пары строк исходной матрицы β . Эта матрица вычисляется по формуле (10) функцией `Sum_orderes_for_client_row`, на вход которой передаются I и $s1$.

Реализация алгоритма вычисления порядковой матрицы Im имеет вид

`Sum_orderes_for_client_row(Im)`

1. `for(s1=0; s1<N; ++s1)`
2. `for(s2=0; s2<N; ++s2)`
3. `V = construct_V(s1,s2)`
4. `I = order_sort(V)`
5. `Sum_orderes_for_client_row(Im,I,s1)`

Следует отметить, алгоритм можно сделать более эффективным, если воспользоваться основными свойствами алгоритмов порядковой сортировки [1].

Индексы выигрышей в матрице Im совпадают с индексами соответствующих элементов β_{ij} в матрице назначений.

Метод решения МЗН, представленной порядковой матрицей

Следует понимать, что представление элементов решений в виде выигрышей, вычисленных с помощью алгоритмов порядковой сортировки, предназначено только для упрощения построения решений в известных методах решения многокритериальных задач. Основным достоинством порядковой нормализации является точность определения ценности элемента при его использовании в решении. Таким образом, после вычисления порядковой матрицы может быть использован любой метод решения многокритериальной задачи.

Сформулируем аксиому, на основании которой производится свертка функционала задачи. Так как нормализованные элементы вектора критериев являются равноценными по условию задачи, то в данном случае справедлива следующая аксиома.

Аксиома упорядочения критериев

Упорядочение по весу нормализованных критериев всех векторов критериев задачи не изменяет ценность решений многокритериальной задачи.

На основании данной аксиомы, во-первых, просто реализовать выделение доминирующего критерия, а во-вторых, удовлетворить основополагающему принципу равномерности решений многокритериальных задач.

Алгоритм решения МЗН, представленной порядковой матрицей

1. *Нормализация.* Вычисление аппроксимирующей порядковой матрицы Im .
2. *Выделение доминирующего критерия.* Сортировка критериев Im на основе аксиомы упорядочения критериев и построение матрицы I_{sort} .
3. *Решение МЗН* для матрицы I_{sort} .

Пример решения МЗН

Решение трехкритериальной ЗН, имеющей множество равнозначных критериев.

Пусть в организации разрабатывается N проектов, которые характеризуются следующими параметрами.

1. Критерий 1 – *стоимость*;
2. Критерий 2 – *эффективность*;
3. Критерий 3 – *сроки реализации*.

Необходимо назначить исполнителей таким образом, чтобы обеспечить достаточное качество выполнения проектов и при этом интегральную эффективность выполнения всех проектов организации.

Критерий 1 определялся в диапазоне $[0,32)$ при помощи генератора случайных чисел.

Критерий 2 генерировался в диапазоне $[2,9)$ на основании критерия 1 с использованием коэффициента корреляции, равного 0,7.

Критерий 3 генерировался в диапазоне [0,5) на основании критерия 1 с использованием коэффициента корреляции, равного 0,5.

```
mcap.rand_int_criterion_matrix(0,0,33);
mcap.rand_correlation_int_criterion_matrix(1,2,
9,0,0,33,0.7);
mcap.rand_correlation_int_criterion_matrix(2,0,
5,1,2,9,0.5);
```

Следует отметить, корреляция критериев является необходимой, иначе многокритериальная матрица будет несбалансированной, а значит, не согласовываться с большинством практических многокритериальных задач.

Параметрами метода

rand_int_criterion_matrix являются: номер критерия и диапазон генерации случайных чисел.

Параметрами метода

rand_correlation_int_criterion_matrix являются: номер критерия, его диапазон, базовый критерий, на основании которого генерируется критерий и коэффициент корреляции.

MCAP (MultiCrierion Assignmnt Problem) – класс, представляющий многокритериальную ЗН, в котором реализованы: методы ввода и генерации исходных данных, методы решения задачи и представления результатов решения.

Этапы решения задачи

1) Объявление объекта, представляющего многокритериальную ЗН, ввод числа строк, столбцов матрицы назначений и числа критериев.

```
MCAP mcap(M,N,3);
```

2) Выбор метода нормализации (аппроксимации) каждого критерия задачи.

```
vector<int> order_sort(3);
order_sort[0] = MCAP::GOS;
order_sort[1] = MCAP::GOS;
order_sort[2] = MCAP::GOS;
```

3) Вычисление порядковой матрицы Im.

```
mcap.calculate_Im(order_sort);
```

4) Выделение доминирующего критерия. Сортировка критериев Im на основе аксиомы упорядочения критериев и построение матрицы Isort.

```
mcap.sort_criteria();
```

5) Решение минимаксной ЗН для матрицы Isort.

```
mcap.solve_by_main_criterion();
```

6) Вывод и сохранение результатов решения

```
mcap.cout_alg_results();
mcap.save_alg_results();
mcap.save_all_criterion_ini_matrix();
mcap.save_approximated_criterion_matrix();
```

Исходные данные МЗН

	0	1	2	3	4
0	(17;5;3)	(28;5;3)	(1;3;2)	(30;7;5)	(7;4;3)
1	(2;3;2)	(0;2;2)	(31;6;3)	(31;6;4)	(30;6;3)
2	(8;4;3)	(8;3;2)	(14;6;3)	(9;3;2)	(26;6;3)
3	(30;5;2)	(25;9;4)	(2;3;2)	(21;5;2)	(9;3;2)
4	(31;7;4)	(20;5;3)	(12;4;2)	(23;5;2)	(7;5;3)

Матрица выигрышей и порядковая матрица нулевой строки матрицы первого критерия, вычисленная алгоритмом порядковой сортировки.

Матрица выигрышей нулевой строки, вычисленная по формуле (7)	Порядковая матрица нулевой строки, вычисленная по формуле (10)
0 0 0 0 0	4 4 4 4 4
-15 -28 30 1 23	1 0 4 2 3
-9 -20 13 -21 19	2 1 3 0 4
13 -3 1 -9 2	4 1 2 0 3
14 -8 11 -7 0	4 0 3 1 2
Интегральный выигрыш нулевой строки (17,28,1,30,7) исходной матрицы	Аппроксимированный интегральный выигрыш нулевой строки исходной матрицы Im
3 -59 55 -36 44	15 6 16 7 16

Порядковая матрица выигрышей всех критериев

	0	1	2	3	4
0	(15;14;15)	(6;10;14)	(16;18;18)	(7;6;6)	(16;14;15)
1	(18;18;19)	(18;19;18)	(6;7;10)	(10;10;8)	(8;10;12)
2	(15;13;12)	(12;15;18)	(10;8;9)	(17;18;17)	(6;10;13)
3	(6;13;16)	(9;4;4)	(17;16;14)	(15;12;15)	(13;19;17)
4	(6;5;5)	(15;14;13)	(11;16;17)	(11;16;18)	(17;13;13)

Цветом выделены элементы оптимальной перестановки.

Решение задачи

Перестановка:

[0,2] [1,0] [2,3] [3,4] [4,4]

Значения элементов решения:

(1;3;2) (2;3;2) (9;3;2) (9;3;2) (20;5;3)

Длина решения: (41,23,11)

Выводы

Рассмотренный в работе метод решения МЗН, основанный на алгоритмах порядковой нормализации, расширяет спектр современных методов решения многокритериальных задач.

Основным достоинством алгоритмов нормализации является то, что взвешенные значения критериев приводятся к единой шкале. Это существенно упрощает процедуры построения векторных решений и их сравнения, а также процесс свертки функционалов многокритериальных задач. Алгоритмы нормализации обладают высоким быстродействием, так как имеют полиномиальную сложность $N^3 \log N$.

Предложенный метод решения МЗН позволяет решать задачи большой размерности с большим количеством критериев, поскольку исключает ЛПР из процесса построения решений задачи. ЛПР играет свою роль на этапе выбора одной из трех стратегий (алгоритмов) нормализации соответствующего критерия и на заключительном этапе анализа построенных алгоритмом решений.

Литература

1. Штойер Р. Многокритериальная оптимизация: теория, вычисления, приложения / Ральф Штойер; пер. с английского Е.М. Столяровой. – М.: Радио и связь, 1992. – 504 с.
2. Подиновский В.В. Парето-оптимальные решения многокритериальных задач / В.В. Подиновский, В.Д. Ногин. – М.: Наука, 1982. – 267 с.
3. Анохин А.М. Методы определения коэффициентов важности критериев / А.М. Анохин, В.А. Глотов, В.В. Павельев, А.М. Черкашин // Автоматика и телемеханика. – 1997. – №8. – С. 3–35.
4. Подиновский В.В. Введение в теорию важности критериев в многокритериальных задачах принятия решений / В.В. Подиновский. – М.: Физматлит, 2007. – 64 с.
5. Подиновский В.В. Количественная важность критериев с дискретной шкалой первой порядковой метрики / В.В. Подиновский // Автоматика и телемеханика. – 2004. – №8. – С. 196–203.
6. Салтыков С.А. Экспериментальное сопоставление методов взвешенной суммы, теории полезности и теории важности критериев для решения многокритериальных задач с балльными критериями / С.А. Салтыков // Управление большими системами. – М.: ИПУ РАН, 2010. – Вып. 29. – С. 16–41.

Рецензент О.П. Алексеев, профессор, д.т.н., ХНАДУ.

Статья поступила в редакцию 30 октября 2011 г.