УДК 519.161

ПРИМЕНЕНИЕ ПОРЯДКОВОЙ НОРМАЛИЗАЦИИ И СКРЕМБЛИРОВАНИЯ КРИТЕРИЕВ ДЛЯ РЕШЕНИЯ МНОГОКРИТЕРИАЛЬНЫХ ЗАДАЧ

О.А. Подоляка, доцент, к.т.н., ХНАДУ,
А.Н. Подоляка, ст. преподаватель, НАУ «ХАИ»

Аннотация. *Представлен новый метод решения класса многокритериальных задач, сводимых к поиску наибольшего звездного покрытия, в основу которого положены процедуры порядковой нормализации и скремблирования (перемешивания) критериев.*

Ключевые слова: *многокритериальная задача, нормализация критериев, наибольшее звездное покрытие, задача о назначениях, паросочетания.*

ЗАСТОСУВАННЯ ПОРЯДКОВОЇ НОРМАЛІЗАЦІЇ ТА СКРЕМБЛЮВАННЯ КРИТЕРІЇВ ДЛЯ РОЗВ’ЯЗАННЯ БАГАТОКРИТЕРІАЛЬНИХ ЗАДАЧ

О.О. Подоляка, доцент, к.т.н., ХНАДУ,
О.М. Подоляка, ст. викладач, НАУ «ХАІ»

Анотація. *Представлено новий метод розв’язання класу багатокритеріальних задач, які зводяться до пошуку найбільшого зоряного покриття, в основі якого покладено процедури порядкової нормалізації та скремблювання (перемішування) критеріїв.*

Ключові слова: *багатокритеріальна задача, нормалізація критеріїв, найбільше зоряне покриття, задача про призначення, паросполучення.*

USAGE OF ORDINAL NORMALIZATION AND CRITERIONS SCRAMBLING FOR SOLUTION OF MULTICRITERION PROBLEMS

O. Podolyaka, assistant professor, cand. eng. sc., KhNAHU,
A. Podolyaka, senior lecturer, NAU «KhAI»

Abstract. *A new method for solving a class of multicriteria problems, reducible to find greatest star coverage, which is based on an ordinal normalization procedure and scrambling (reshuffle) criteria.*

Key words: *multicriteria problems, criteria normalization, greatest star coverage, assignment problem, matching.*

Введение

Рассмотрим решение многокритериальной задачи на примере процедуры приобретения недвижимости. Например, при покупке квартиры Вы учитываете следующие критерии: бюджет, площадь, район, цена и т.д. Каждый критерий определяет соответствующее подмножество рыночных альтернатив и задача Вас, как покупателя найти разумный компромисс, т.е. реализовать процедуру пересечения ваших желаний и возможностей на заданном множестве альтернатив (квартир). К сожалению, на данный момент развития научной мысли, никто не придумал универсальный машинный алгоритм, реализующий этот компромисс. Большинство эффективных алгоритмов могут только предложить некоторые эффективные альтернативы на основе некоторой общей схемы рационального поведения. Следует отметить, что решение многих практических многокритериальных задач представляет огромную сложность, как с оптимизационной, так и вычислительной точек зрения, поскольку число альтернатив обычно экспоненциально. Поэтому, основной задачей многокритериальной оптимизации является поиск компактного множества эффективных альтернатив, которые человек может осмыслить и принять решение, опираясь на свой опыт и предпочтения.

Анализ проблем многокритериальной оптимизации

Перечислим основные проблемы многокритериальной оптимизации:
1. *Разнородность критериев.* В ходе решения задачи на компьютере приходится сравнивать: доллары, километры, килограммы, секунды, м^2 и т.д.
2. *Различие критериальных шкал.* Например, шкала да-нет; четырехбальная шкала; шкалы: цены, времени, температуры и т.п. В этом случае максимальные значения бедных оценочных шкал задавливают допустимые значения широких шкал.
3. *Оценка важности критериев.* Сводится к определению весовых коэффициентов критериев целевой функции [5,6,7]. После определения этих коэффициентов все критерии становятся равнозначными.
4. *Противоречивость критериев.* Например, при покупке квартиры, выбор престижного района влечет уменьшение площади и увеличение цены.
5. *Выбор принципа оптимальности.* С точки зрения классической оптимизации решения многокритериальных задач не являются оптимальными. Из-за противоречивости критериев оптимум одного из критериев влечет за собой ухудшение других критериев. Поэтому решения таких задач называют эффективными с точки зрения выбранного принципа оптимальности или выбранной схемы поиска компромисса [4].
6. *Принцип равномерности.* Качественные решения должны удовлетворять фундаментальному принципу равномерности, согласно, которому элементы решений должны быть максимально равнозначными (близкими по весу). Неравномерные решения недопустимы, поскольку влекут катастрофический вторичный ущерб (потеря репутации, клиентов, жизней и т.п.). Следует отметить, что равномерность отражает разумный баланс критериального выигрыша и проигрыша. Удовлетворение этого принципа уменьшает общий выигрыш, но повышает качество решений.

Проблемы методов решения многокритериальных задач

Следует отметить, что большинство методов решения многокритериальных задач состоит из двух этапов – нормализации, которую обычно считают тривиальной, и оптимизационной процедуры на основе выбранной схемы компромисса. Классическая нормализация состоит в приведении каждого из критериев задачи к безразмерному виду путем простого линейного преобразования - деления взвешенного значения на максимальный элемент критериальной последовательности. Предполагается, что нормированные значения всех критериев принадлежат одному типу, для которого допустимы соответствующие арифметические операторы и операции сравнения. Обычно нормализация выполняется, по следующей формуле

$$C_i = \frac{C_i^0 - C_i^{\min}}{C_i^{\max} - C_i^{\min}}$$

где: C_i - нормированный критерий;

$C_i^{\max}$ - максимальное значение критерия;

$C_i^{\min}$ - минимальное значение критерия;

C_i^0 - исходное измеренное значение критерия.

Но, такая нормализация создает массу серьезных проблем. Следует понимать, что деление на максимальное значение всех критериев в процедуре классической нормализации изменяет важность соответствующих критериев по отношению друг к другу, если их значения распределены неравномерно. Необходимо подчеркнуть, что равномерное распределение значений является исключительным. В реальном мире очень мало богатейших, красивейших и т.п., и простым смертным равняться с ними противоестественно. Задача определения корректного максимума для каждого критерия является сложной, т.к. он может быть статистическим выбросом, или находиться вне пределов критериальной последовательности. Так же отметим, что часто предпочтения людей вдоль критериальной шкалы распределяются нелинейно. Например, люди обычно равняются на себе подобных по образованию, культуре, доходу и т.п. и формируют соответствующие группы или кластеры. Поэтому оценки представителей других групп, в том числе и лучших, будут ниже. Как проводить нормализацию критериев в этом случае?

Подведем некоторые итоги. Поскольку, максимумы критериев различны и критерии обычно распределены не равномерно, классическая нормализация нелинейно искажает важность критериев вдоль критериальной шкалы. В этом случае ошибка некорректного определения весов критериев закладывается уже на начальном этапе решения задачи, т.е. на вход оптимизационной процедуры передаются искаженные данные. Устранить ошибку может нелинейная нормализация.

В работе [1] она представлена тремя процедурами порядковой нормализацией LOS, MOS и GOS. Они определяют: нижнее значение нормированного критерия LOS, полученное на основании отношения строгого предпочтения; верхнее значение GOS, полученное с учетом строго предпочтения и безразличия и MOS – их среднее. Эти процедуры могут применяться для нормализации критериев следующих многокритериальных задач:
1. задачи поиска наибольшего паросочетания двудольного графа (задача о назначениях);
2. задачи поиска *наибольшего звездного покрытия* (НЗП);
3. задачи поиска 2-фактора.

В работе [2] рассматриваются эти задачи в однокритериальной постановке.

Математическая модель задачи

Рассмотрим оптимизационную задачу поиска НЗП для графа $G_{n,n\cdot h}(V_1,V_2,E)$ [2,3] в матричной постановке (h - степень звезды). В ней отражена принадлежность элемента β_{ij} матрицы двудольного графа β некоторому наибольшему звездному покрытию $E_h^V \in E$. Это достигается путем умножения этого значения на соответствующее значение $x \in \{0,1\}$

Пусть: M - строки, N - столбцы, $h = N/M$ - степень звезды и, тогда постановка имеет вид.

$$\begin{cases} w\left(E_h^*\right) = \sum_i^M \sum_j^N \left(x_{ij}\beta_{ij}\right) \to \min \\ \sum_{i=1}^M x_{ij} = k \cdot h,\ j = \overline{1,N} \\ \sum_{j=1}^N x_{ij} = k, i = \overline{1,M} \\ x \in \{0,1\} \\ k = 1 \end{cases} \quad (1)$$

Выражение

$$w\left(E_h^*\right) = \min_{E_h^{'} \in E_h^{'all}} \left[w\left(E_h^V\right)\right] = \sum_i^M \sum_j^N \left(x_{ij}\beta_{ij}\right) \to \min$$

представляет функционал однокритериальной задачи.

При $k=1$ выражения $\sum_{i=1}^M x_{ij} = h$ и $\sum_{j=1}^N x_{ij} = 1$ показывают, что в каждой строке есть h элементов β_{ij}, а каждом столбце исходной матрицы находится один элемент решения β_{ij}, т.е. эти выражения определяют звезды $E_h^{'}$.

Следует отметить, что:
1. При $h=1$ (1) представляет собой постановку классической задачи о назначениях (ЗН). Различные варианты этой задачи и алгоритмы ее решения подробно рассмотрены в работе [8];
2. Задача поиска наибольшего паросочетания может быть представлена эквивалентной оптимизационной постановкой (1) с матрицей назначений элементами которой есть значения 0 или 1 (есть ребро, нет ребра);
3. Постановка задачи (1) при $k>1$ заслуживает отдельного рассмотрения. Отметим, только что эта задача может быть

сведена к последовательности решений k задач НЗП в матрицах размерности $MxN_j, N_j = N \cdot j, \quad j \in \overline{1,k}$;

4. При $h = k = 2$ получаем постановку задачи поиска 2-фактора, или поиска покрытия графа остовными циклами [8].

В работах [2,3] рассматриваются вопросы сведения вышеперечисленных оптимизационных задач к поиску наибольшего звездного покрытия и метод решения этих задач.

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

$$\beta_{ij} = \left\{ C_{ij}^k, k = \overline{1,L} \right\} \tag{2}$$

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

Многокритериальная задача поиска НЗП с множеством равнозначных критериев:

$$\begin{cases} \rho\left(E_{(1)h}^*\right) = \min\limits_{E_{(1)h}^V} \sum\limits_{(i,j)\in E_{(1)h}^V} C_{ij}^1 \\ \rho\left(E_{(2)h}^*\right) = \min\limits_{E_{(2)h}^V} \sum\limits_{(i,j)\in E_{(2)h}^V} C_{ij}^2 \\ \ldots \\ \rho\left(E_{(k)h}^*\right) = \min\limits_{E_{(k)h}^V} \sum\limits_{(i,j)\in E_{(k)h}^V} C_{ij}^k \end{cases} \tag{3}$$

где: $\min\limits_{E_{(k)h}^V} \sum\limits_{(i,j)\in E_{(k)h}^V} C_{ij}^k$ – вес НЗП $E_{(k)h}^V$ k-го критерия; C_{ij}^k – вес k-го критерия.

Обоснование предложенного метода

Ключевой идеей работы [1] было объединение этапов нормализации и оптимизации в виде одной полиномиальной $O(n^3 \log n)$ человеко-машинной процедуры (ЧМП), в которой *лицо принимающее решение* (ЛПР) определяло необходимую стратегию поиска эффективного решения по каждому из критериев задачи. Эта стратегия определялась соответствующей процедурой нормализации: LOS-пессимизм, GOS-оптимизм и средневзвешенной - MOS. На начальном этапе ЧМП задается средневзвешенная стратегия по всем критериям задачи. Затем выполнялся оптимизационная процедура и этап анализа полученных решений. В ходе, которого ЛПР определяло веса, каких критериев нужно поднять, а каких опустить. Для этого он выбирал соответствующее нижнее, среднее или верхнее нормированное значение. Этапы оптимизации и анализа повторяются, пока не будет найдено эффективное решение.

Отметим, что число возможных стратегий ЧМП равно 3^n, где n – число критериев задачи. Его можно считать постоянным, т.к. постоянно число критериев. Однако заметим, что наблюдается экспоненциальный рост числа стратегий при увеличении числа критериев. Например, для десятикритериальной задачи число стратегий составляет 59049. Конечно, количество эффективных стратегий во много раз меньше, но следует отметить, что анализ, даже оставшихся, представляет сложный и трудоемкий для ЛПР мыслительный процесс.

Поэтому основным достоинством нового метода есть – существенное облегчение работы ЛПР, которое достигается путем исключения его из процесса выбора стратегий, или выбора схемы компромисса. Данный метод сам находит компромисс при помощи процедур перестановки противоречивых критериев и выполнения над полученными последовательностями процедур порядковой нормализации. Затем выполняется сведение многокритериальной задачи к однокритериальной путем сложения весов критериев соответствующих элементов последовательностей. И далее выполняется оптимизационная процедура, которая предоставляет ЛПР решения для анализа. Если полученное решение не удовлетворяет пользователя системы он нажимает кнопку next, указанные процедуры повторяются и ему для анализа предоставляется новое решение.

Итак, в работе представлен новый метод решения многокритериальной задачи, в котором тесно связаны процедуры нормализации, скремблирования и оптимизации. Схема компромисса реализуется в методе через нормализацию. Процедура перестановки критериев и их *сложения* требует отдельного пояснения.

Идея порядковой нормализации состоит в замене упорядоченной последовательности весов критериев на упорядоченную последовательность порядковых номеров или индексов. Например, подобная ситуация наблюдается, когда подводятся результаты спортивных соревнований. Т.е. вычисляется соответствующая позиция спортсмена в общем рейтинге в зависимости от показанного результата (время, вес, длина, баллы и т.п.). Примером данной шкалы в макроэкономике являются оценки рейтинговых агентств Moody's и Fitch. В этом случае выполняется отображение размерных величин произвольной величины в соответствующее компактное множество безразмерных оценок или индексов. По своей сути это отображение является нелинейной нормализацией поскольку нелинейной последовательности результатов ставится в соответствие линейная последовательность порядковых номеров.

Теперь рассмотрим процесс нормализации, применительно к решаемым задачам. Заметим, что если некоторое подмножество элементов решения (ребер звездного покрытия) заменить на другое допустимое подмножество, то получится новое решение. Специфика, рассматриваемых перестановочных задач состоит в том, что можно делать замены по два (2 на 2), по три (3 на 3) и т.д. и не допускаются одиночные замены. Если замена увеличивает наш выигрыш или является эквивалентной она получает призовой бал. Следовательно, в процессе построения решений сравниваются веса соответствующих подмножеств, т.е. выполняется сравнение групп элементов. Так вот, в данной нормализации учитываются только сравнения пар элементов (2 на 2). Такое упрощение облегчает поиск компромисса многокритериальной задачи. Конечно, для повышения точности нормализации можно учитывать перестановки элементов более высоких порядков, например, замены по два и по три. Но, во-первых, алгоритм такой нормализации будет иметь нехорошую вычислительную сложность $O(n^5)$. И во-вторых, такая точность оказывает слабое влияние на поиск качественного решения, поскольку точность нормализации теряется на фоне влияния критериальных противоречий на качество решений. Рассмотрим математическую модель нормализации.

Алгоритмы порядковой сортировки

Пусть: C – исходный массив;
L,G,M – массивы упорядоченных индексов;
less_count(C,C[i]), greater_count(C,C[i]), count(C ,C[i]) – функции определения количества элементов массива C: меньших, больших и равных элементу C[i].
Замечания: Индексация в массивах начинается с нуля. Целевая функция минимизируется.
Вычисление массивов L, G, M при минимизации функционала.

Осторожный пессимизм (<)
Алгоритм less_order_sort (LOS):
L[i] = LOS(C,C[i]) = less_count(C,C[i]), (4)
Осторожный оптимизм (<=)
Алгоритм greater_ order _sort (GOS):
G[i] = GOS(C,C[i]) =
= N - 1 -greater_count(C,C[i]) =
= less_count(C,C[i]) + count(C ,C[i]) - 1, (5)
Средневзвешенная стратегия
Алгоритм mean_ order _sort (MOS):
M[i] = MOS(C,C[i]) =
=(LOS(C,C[i]) +GOS(C,C[i]))/2 (6)

i	0	1	2	3	4	5	6	7
C	-50	10	10	90	-20	10	90	-20
L	0	3	3	6	1	3	6	1
G	0	5	5	7	2	5	7	2
M	0	4	4	6,5	1,5	4	6,5	1,5

Рис.1. Пример порядковой сортировки массива C

Нормализация - вычисление порядкового значения критерия

$$L_{ij}^k = \sum_{r=0,i\neq r}^{N-1} LOS\left(C_{i-r}^k, \left(C_{ij}^k - C_{rj}^k \right) \right), \quad (7)$$

$$G_{ij}^k = \sum_{r=0,i\neq r}^{N-1} GOS\left(C_{i-r}^k, \left(C_{ij}^k - C_{rj}^k \right) \right), \quad (8)$$

$$M_{ij}^k = \left(L_{ij}^k + G_{ij}^k \right) / 2, \quad (9)$$

где: L^k и G^k - порядковые матрицы, в которых находятся интегральные нижние и верхние оценки k -го критерия C_{ij}^k ; L_{ij}^k и G_{ij}^k

элементы i-ой строки и j-го столбца порядковой матрицы; C_{i-r}^{k} - массив, вычисляемый как разность строк i и r исходной матрицы смежности двудольного графа C^{k} k-го критерия; M_{ij}^{k} - среднее значение. Чтобы работать с целыми числами элементы матрицы нужно умножить на два.
Следует сказать, что вычисление нормированных значений по формулам (7,8) имеет сложность $O\left(n^{4}\right)$. В работе [1] приведен алгоритм нормализации, основанный на быстрой сортировке массивов, сложность которого равна $O(n^{3}\log n)$.

Необходимо отметить, что огромным преимуществом порядковой нормализации является получение существенного разнообразия нормированных значений критерия для изначально бедных критериальных шкал. Это разнообразие объясняется учетом всевозможных парных сочетаний критериев. В этом смысле, данную нормализацию можно рассматривать, как процедуру поиска скрытых знаний в исходных данных задачи.

Перестановки и суммирование критериев

Пришло время обосновать перестановки нормированных значений и допустимость математических операций. Известно, что некачественные решения получаются поскольку целевые функции многокритериальной задачи нацелены исключительно на максимальный выигрыш. Поэтому, в некачественных решениях потери по отдельным критериям теряются на фоне суммарного выигрыша. Понятно, чтобы получить эффективные решения нужно, чтобы при нормализации максимумы не задавливали минимумы. Это является огромной проблемой для современной науки. В данной работе ее предлагается решить при помощи процедур перестановки конфликтных критериев и порядковой нормализации. Рассмотрим процедуру перестановки критериев элементов решения многокритериальной задачи.

Чтобы понять, как эта процедура работает, рассмотрим следующий пример. Судейский комитет должен определить рейтинг сборных различных стран путем суммирования рейтинга танцевальных пар. Допустим, украинский танцор имеет высокий рейтинг, а его партнерша – низкий. Поменяем местами рейтинги этой пары спортсменов и дадим ответы на два важных вопроса:
1. Как изменится рейтинг танцевальной пары и рейтинг сборной?
2. Как изменятся рейтинги во множестве мальчиков и во множестве девочек украинской сборной?

Ответ на первый вопрос очевиден – рейтинги пары и сборной останутся прежними, поскольку перестановка не изменяет множество оценок спортсменов. Значит арифметические операции, и операции сравнения танцевальных пар будут корректными поскольку они сохраняют упорядочение пар и сборных.
Следует отметить, что если танцевальную пару понимать, как некоторую сущность, которая описывается двумя критериями, то сложение этих критериев представляет процедуру сведения многокритериальной задачи к однокритериальной. Для того чтобы обосновать такое Наглое сведение рассмотрим второй вопрос.

Ответ на него позволяет сделать ряд важных выводов с точки зрения многокритериальной оптимизации. После замены рейтинги мальчиков изменятся, т.к. понижение рейтинга мальчика изменяет упорядочение танцоров. Т.е. для мужчин, у которых рейтинг до изменения был выше он не изменится, а танцоры с рейтингами ниже предыдущего рейтинга и выше рейтинга партнерши поднимутся на одну позицию. Для девочек, которые попадают в указанный диапазон, произойдет понижение рейтинга. Следовательно, парные сочетания спортсменов оказывают непосредственное влияние на их рейтинги. Поэтому такая нечестная перестановка оценок имеет смысл, поскольку позволяет, во-первых, выделить несбалансированную пару в сборной, и во-вторых, вынуждает оптимизационный алгоритм искать для нее более эффективную замену. В этом случае важно понимать, что процедуры перестановки и пересчета рейтингов реализуют поиск компромисса многокритериальной задачи т.е. позволяют получить более эффективные решения. В предложенном методе решения задачи перестановка рейтингов аналогична указанной, а пересчет рейтингов выполняет порядковая нормализация.

Особенности критериальных перестановок

В данных момент следует обратить ваше внимание на три важных момента, связанных с перестановкой элементов критериальных векторов.

Во-первых, для рассматриваемых многокритериальных задач допустима перестановка оценок только в пределах отдельного элемента решения (вектора критериев), в нашем случае - танцевальной пары. Значит, запрещена перестановка критериев различных элементов решений (т.е. не допускается обмен партнерами или партнершами).

Во-вторых, нужно определить элементы (вектора), для которых следует выполнить перестановку оценок. Здесь возможны два варианта:

1. перестановка оценок выполняется для всех элементов последовательности;
2. перестановка оценок выполняется для элементов, соответствующих некоторому эффективному решению многокритериальной задачи. Обычно, это решение находится путем решения соответствующей однокритериальной задачи.

И в-третьих, нужно понимать, что существует несколько способов перестановки оценок. Например, можно переставлять оценки равноудаленные от центра упорядоченного вектора. Или использовать более мягкую схему – менять оценки расположенные на половине длины вектора критериев. Для некоторых прикладных задач с большим числом критериев можно обосновать варианты перестановок для выделенных подгрупп элементов.

Процедуры скремблирования

Пусть $C_{ij} = \left\{ C_{ij}^k, k = \overline{1,K} \right\}$ - критериальный вектор. Упорядочим по весу критериальный вектор C_{ij}.

Процедуру скремблирования scramble_1 реализуется путем перестановки местами весов критериев равноудаленных от центра упорядоченного вектора C_{ij}, т.е. с индексами (i) и (K-i+1), (i<=K). А процедуру scramble_2 выполним посредством перестановки местами критериев с индексами (i) и (i+K/2), (i<=K/2).

Пример. Допустим, критериальный вектор состоит и четырех элементов $\left(9^1, 4^2, 1^3, 7^4\right)$. Для каждого критерия указан вес и в верхнем регистре номер. $\left(1^3, 4^2 \mid 7^4, 9^1\right)$ - упорядоченный вектор. Тогда результат scramble_1 - $\left(1^1, 7^2, 9^3, 4^4\right)$, а scramble_2 - $\left(4^1, 9^2, 7^3, 1^4\right)$.

Алгоритм решения многокритериальной задачи

Алгоритм решения задачи, образованный процедурами скремблирования, порядковой нормализации критериев и оптимизации называется Scramble_optimize. Давайте рассмотрим несколько важных вопросов касательно данного алгоритма.

1. Сколько раз следует выполнить алгоритм?
2. Является ли алгоритм скремблирования сходящимся?

Заметим, что оба этих вопроса связаны. И прежде, чем ответить на них следует сказать, что порядковая нормализация упрощает модель данных многокритериальной задачи. Это означает, что она искажает исходные данные по установленным правилам. В этом смысле, рейтинг команд футбольного турнира значительно беднее результатов всех его матчей. Нормализация и скремблирование определяют аналогичный рейтинг, который упрощает выбор лучших элементов. Однако, нужно помнить, что любой рейтинг – это упрощение, которое обедняет модель данных и у каждого упрощения есть предел. В нашем случае, его определяет пользователь системы в процессе анализа решений. Максимально рекомендованное число нажатий кнопки Scramble_optimizeравняется количеству критериев.

Начальная нормализация называется нормализацией нулевого порядка, ее данные являются входными для алгоритма Scramble_optimize. После каждого выполнения алгоритма порядок нормализации повышается на единицу.

Следует отметить, что сходимость алгоритма скремблирования представляет исключительно научный интерес, так как на практике все решает пользователь. А науку интересуют вопросы сходимости данных, накопления ошибок. Это сложные вопросы, надеюсь дать на них ответ в дальнейших исследованиях.

Общая схема алгоритма Scramble_optimize

1. Скремблирование всех критериев $M_{ij}^{k} = scramble\left(M_{ij}^{k}\right)$. Вычисление матриц M^{k}.
2. Порядковая нормализация всех критериев по выбранной ЛПР стратегии, вычисление матриц M^{k}, $M_{ij}^{k} = L_{ij}^{k} + G_{ij}^{k}$.
3. Получение однокритериальной задачи путем суммирования критериев. Вычисление матрицы M, $M_{ij} = \sum_{k=1}^{K} M_{ij}^{k}$.
4. Оптимизация - решение однокритериальной задачи, представленной матрицей M.

Пример решения многокритериальной задачи о назначениях

Прежде чем рассматривать практический пример следует отметить, что данная реализация алгоритма Scramble_optimize имеет следующие особенности:
1. Работа алгоритма будет рассмотрена на примере решения четырехкритериальной задачи о назначениях. Выбор ЗН объясняется тем что она широко известна [8] и не требует дополнительных поясней. Однако, нужно отметить, что процедуры скремблирования и нормализации допустимы для всех оптимизационных задач, сводимых к поиску звездного покрытия;
2. Будет рассмотрена сложная многокритериальная ЗН, у которой критериальные шкалы задачи будут иметь существенные различия;
3. Нормализация всех критериев осуществляется на основе средневзвешенной стратегии MOS;
4. Скремблирование критериев выполняется для всех векторов матрицы смежности двудольного графа;
5. Скремблирование критериев будет реализовано при помощи процедуры scramble_2.

Теперь рассмотрим практический пример.

Допустим, чтобы получить назначение в соответствующее силовое подразделение офицер должен получить максимальные оценки по следующим критериям:
1. iq - [80-140];
2. пол - [0,1];
3. тактика - [60-100];
4. огневая подготовка - [3,4,5].

Командование желает определить наиболее эффективное назначение для всех военнослужащих.

Данные задачи представлены матрицей, в которой строки – это офицеры, а столбцы – спецподразделения (назначения)

.	0	1	2	3	4	5	6
0	(85;1;65;3)	(97;1;93;3)	(87;0;86;3)	(136;0;82;5)	**(116;1;86;4)**	(100;0;91;4)	**(132;1;82;5)**
1	(95;0;80;4)	(133;1;78;5)	**(108;1;87;5)**	(123;1;76;4)	(84;0;86;3)	(116;1;77;4)	(84;0;61;3)
2	(104;1;93;4)	**(128;1;84;5)**	(122;1;72;3)	(110;0;74;5)	(98;0;69;4)	(118;0;63;3)	(118;1;98;4)
3	(125;1;66;3)	(129;0;80;5)	(135;0;70;3)	**(103;1;95;5)**	**(92;0;98;5)**	(88;0;87;3)	(97;1;68;3)
4	(110;1;71;3)	(121;0;89;5)	(139;0;76;3)	(103;0;65;4)	(89;0;100;3)	**(87;0;78;4)**	**(130;1;83;3)**
5	**(128;1;75;4)**	(110;0;76;4)	(103;0;80;4)	(85;1;93;4)	(113;1;78;5)	(130;1;70;3)	(133;0;93;3)
6	(122;0;66;4)	(106;1;84;4)	(101;0;89;5)	**(133;1;97;5)**	**(131;1;81;4)**	**(92;1;96;4)**	(115;0;96;3)

Рис. 2. Исходная матрица четырехкритериальной задачи о назначениях (элементы всех найденных решений выделены заливкой)

Шаг 1. Вычисляем порядковую матрицу четырехкритериальной задачи о назначениях, каждый элемент которой представлен MOS значением соответствующего критерия.

	0	1	2	3	4	5	6
0	(14;42;18;22)	(22;51;52;7)	(10;23;42;20)	(62;14;23;47)	(50;56;34;37)	(36;19;52;48)	**(58;47;31;71)**
1	(22;12;50;45)	(60;48;28;49)	**(33;58;60;64)**	(54;47;27;16)	(21;18;46;10)	(56;52;35;46)	(6;17;6;22)
2	(22;42;68;46)	**(41;50;44;46)**	(41;60;26;15)	(29;15;28;46)	(30;19;10;32)	(52;20;14;15)	(37;46;62;52)
3	(60;48;32;19)	(50;18;28;53)	(58;29;14;19)	(22;57;58;51)	**(28;23;64;63)**	(18;24;48;21)	(16;53;8;26)
4	(34;53;34;27)	(36;23;51;59)	(68;35;29;27)	(27;24;5;27)	(16;29;62;18)	**(16;31;30;60)**	(55;57;41;34)
5	**(56;44;41;52)**	(24;13;22;23)	(18;24;36;44)	(6;48;57;21)	(39;54;23;62)	(58;54;15;21)	(51;15;58;29)
6	(44;11;9;41)	(19;49;27;15)	(24;23;45;63)	**(52;47;54;44)**	(68;53;13;30)	(16;52;58;41)	(29;17;46;18)

Рис. 3. Порядковая матрица $M^4_{(1)}$

	0	1	2	3	4	5	6
0	96	132	95	146	177	155	**207**
1	129	185	**215**	144	95	189	51
2	178	**181**	142	118	91	101	197
3	159	149	120	188	**178**	111	103
4	148	169	159	83	125	**137**	187
5	**193**	82	122	132	178	148	153
6	105	110	155	**197**	164	167	110

Рис. 4. Матрица соответствующей однокритериальной задачи $M^1_{(1)}$

Элементы решений: [0,4], [0,6], [1,2], [2,1], [3,3], [3,4], [4,5], [4,6], [5,0], [6,3], [6,5]

Шаг 2. Выполняем скремблирование критериев и вычисляем порядковую матрицу.

	0	1	2	3	4	5	6
0	(22;18;39;16)	(56;22;12;54)	(21;42;15;10)	(20;42;62;16)	(34;36;48;61)	(54;49;34;16)	(30;70;60;49)
1	(50;44;22;16)	(49;60;53;26)	(60;64;58;34)	(26;15;54;45)	(18;20;18;46)	(44;56;44;34)	(18;6;22;6)
2	(52;68;40;20)	(52;42;39;46)	(19;26;61;41)	(53;30;25;27)	(17;34;24;10)	(14;13;54;24)	(58;52;34;33)
3	(32;12;60;50)	(15;50;60;26)	(16;14;28;58)	(58;51;22;62)	(64;64;17;28)	(24;48;19;18)	(61;16;25;10)
4	(53;34;23;35)	(54;52;13;34)	(28;27;43;68)	(18;28;23;6)	(27;16;20;60)	(49;16;68;28)	(44;36;56;49)
5	(41;59;56;50)	(18;22;9;24)	(33;44;17;23)	(42;7;15;58)	(62;24;58;38)	(16;19;58;58)	(12;52;35;58)
6	(8;40;44;4)	(58;20;16;26)	(46;64;24;24)	(54;53;52;44)	(28;14;54;68)	(42;16;35;58)	(16;46;26;28)

Рис. 5. Порядковая матрица $M^4_{(2)}$

	0	1	2	3	4	5	6
0	95	144	88	140	179	153	**209**
1	132	188	**216**	140	102	178	52
2	180	**179**	147	135	85	105	177
3	154	151	116	193	**173**	109	112
4	145	153	166	75	123	**161**	185
5	**206**	73	117	122	182	151	157
6	96	120	158	**203**	164	151	116

Рис. 6. Матрица соответствующей однокритериальной задачи $M^1_{(2)}$

Элементы решений: [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3]

Шаг 3. Скремблирование критериев, вычисление порядковой матрицы

	0	1	2	3	4	5	6
0	(30;30;13;25)	(13;48;62;22)	(12;12;26;31)	(40;13;17;64)	(35;44;54;42)	(22;19;59;56)	(60;58;37;64)
1	(30;14;38;48)	(58;34;41;57)	(52;66;40;58)	(54;52;26;18)	(26;12;47;18)	(34;40;44;57)	(4;15;9;16)
2	(32;34;64;53)	(40;46;58;44)	(60;48;22;18)	(30;18;38;53)	(42;12;10;22)	(30;42;8;8)	(42;58;44;32)
3	(58;58;10;28)	(41;6;32;70)	(55;34;14;16)	(26;60;56;52)	(17;22;67;58)	(16;20;30;52)	(16;58;10;26)
4	(32;46;40;22)	(18;24;46;60)	(63;44;26;28)	(40;9;2;22)	(46;22;30;28)	(25;64;30;49)	(60;52;46;34)
5	(56;52;45;64)	(17;14;11;20)	(12;13;46;44)	(8;43;49;18)	(36;66;20;62)	(54;57;19;28)	(48;32;50;24)
6	(31;7;10;38)	(23;28;54;21)	(22;31;65;50)	(42;50;56;52)	(64;58;32;16)	(20;61;43;21)	(31;26;42;14)

Рис. 7. Порядковая матрица $M^4_{(3)}$

	0	1	2	3	4	5	6
0	98	145	81	134	175	156	**219**
1	130	190	**216**	150	103	175	44
2	183	**188**	148	139	86	88	176
3	154	149	119	**194**	**164**	118	110
4	140	148	161	73	126	**168**	192
5	**217**	62	115	118	184	158	154
6	86	126	168	**200**	**170**	145	113

Рис. 8. Матрица соответствующей однокритериальной задачи $M^1_{(3)}$

Элементы решений: [0,6], [1,2], [2,1], [3,3], [3,4], [4,5], [5,0], [6,3], [6,4]

Шаг 4. Скремблирование критериев, вычисление порядковой матрицы

	0	1	2	3	4	5	6
0	(13;11;35;37)	(60;16;12;57)	(24;25;13;18)	(14;52;51;16)	(48;36;34;53)	(62;48;32;12)	(58;69;66;36)
1	(46;42;22;20)	(46;64;53;32)	(58;45;68;40)	(30;14;60;54)	(50;15;18;20)	(55;45;33;44)	(11;4;15;4)
2	(57;66;34;23)	(52;46;42;51)	(22;19;46;56)	(41;53;18;22)	(14;24;36;12)	(9;6;37;36)	(50;36;48;52)
3	(21;18;61;60)	(12;51;60;28)	(24;8;58;32)	(58;26;45;59)	(57;66;26;16)	(42;38;14;24)	(22;16;50;16)
4	(46;22;22;48)	(44;56;16;26)	(35;31;46;54)	(10;35;16;6)	(29;32;24;44)	(50;35;56;26)	(44;55;44;56)
5	(51;57;62;51)	(12;2;18;28)	(48;42;12;6)	(42;10;14;56)	(65;34;64;22)	(26;23;47;67)	(23;44;32;50)
6	(6;30;38;9)	(38;17;31;38)	(61;62;19;36)	(46;58;52;50)	(27;20;52;70)	(40;25;22;54)	(16;28;25;38)

Рис. 9. Порядковая матрица $M^4_{(4)}$

	0	1	2	3	4	5	6
0	96	145	80	133	171	154	229
1	130	195	**211**	158	103	177	34
2	180	**191**	143	134	86	88	186
3	160	151	122	188	**165**	118	104
4	138	142	166	67	129	**167**	199
5	**221**	60	108	122	185	163	149
6	83	124	178	**206**	169	141	107

Рис. 10. Матрица соответствующей однокритериальной задачи $M^1_{(4)}$

Элементы решений: [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3].

Полученные решения:
Шаг 1
A= { [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3] }
B= { [0,4], [1,2], [2,1], [3,3], [4,6], [5,0], [6,5] }
Шаг 2
A= { [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3] }
Шаг 3
A= { [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3] }
C= { [0,6], [1,2], [2,1], [3,3], [4,5], [5,0], [6,4] }
Шаг 4
A= { [0,6], [1,2], [2,1], [3,4], [4,5], [5,0], [6,3] }

Выводы

Нужно отметить, что на каждой из четырех итераций алгоритма было найдено решение **A**. Это значит, что изменение весов ребер графа процедурами скремблирования и нормализации с высокой точностью сохраняет упорядочение решений многокритериальной задачи.

Порядковую нормализацию можно рассматривать, как процедуру поиска скрытых знаний в исходных данных задачи. Обратите внимание на разнообразие оценок для второго критерия «пол» в порядковых матрицах $M^4_{(i)}$, вычисленных процедурой MOS.

В ходе алгоритма найдено множество состоящее из трех близких решений {A,B,C}. Значит, метод эффективно находит компактное множество альтернатив, которые ЛПР может легко анализировать.

Первые три пункта выводов определяют область применения метода Scamble_normalize. Его можно эффективно применять при решении наиболее сложных многокритериальных задач, характеризующихся большой размерностью и большим числом критериев с различными измерительными шкалами и диапазонами.

Метод, в основе которого лежат алгоритмы порядковой нормализации и скремблировании критериев использует новые идеи и подходы к решению многокритериальных задач. К его несомненным достоинствам можно отнести упрощение: процесса построения многокритериальных решений, процедур сравнения альтернатив, а также процедуры свертки функционалов многокритериальной задачи. Метод является строго полиномиальным, его вычислительная сложность равна $N^3 \log N$, она определяется сложностью порядковой нормализации.

Литература

1. Ніконов О.Я. Математичні методи розв'язання багатокритеріальної задачі про призначення / О.Я. Ніконов, О.О. Подоляка, О.М. Подоляка, О.В. Скакаліна // Вісник ХНАДУ. - 2011. - №55. - С. 23-33.
2. Подоляка А.Н. Эффективное решение задачи покрытия двудольного графа звездами и некоторых ее обобщений / А.Н. Подоляка, О.А. Подоляка, Е.В. Скакалина // Вісник Чернігівського державного технологічного ун-ту. - 2012. - №4(61). - С. 172 - 179.
3. Подоляка О.А. Поиск наибольшего покрытия двудольного графа звездами заданной степени / О.А. Подоляка А.Н. Подоляка // Автомобіль і електроніка. Сучасні технології. – 2015 . – Вип. 7. – С. 126 -132.
4. Штойер Р. Многокритериальная оптимизация: теория, вычисления, приложения / Ральф Штойер; [пер. с английского Е.М. Столяровой]. - М.: Радио и связь, 1992. – 504 с.
5. 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.
6. Подиновский В.В. Введение в теорию важности критериев в многокритериальных задачах принятия решений / В.В. Подиновский. – М.: Физматлит, 2007. – 64 с.
7. Подиновский В.В. Количественная важность критериев с дискретной шкалой первой порядковой метрики / В.В. Подиновский //Автоматика и телемеханика. – 2004. – №8. – С. 196–203.

8 Ловас Л. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии / Л. Ловас, М. Пламмер. - М.: Мир, 1998. – 653 с.