УДК 519.7, 519.8 DOI: 10.17587/it.27.235-241

С. И. Колесникова, д-р техн. наук, проф., skolesnikova@yandex.ru,
С. А. Караванова, магистрант, sofy.karav@gmail.com,
Санкт-Петербургский государственный университет аэрокосмического приборостроения

Оптимизация алгоритма многокритериального выбора с динамически пополняемым большим набором альтернатив*

Корректное ранжирование динамически пополняемого большого набора альтернатив в многокритериальных задачах выбора основано на алгоритме корректного учета предпочтений на множестве пар относительных весов сравниваемых между собой координат собственных векторов матриц, применяемых в классическом алгоритме анализа иерархий. В таком исполнении алгоритм обеспечивает сохранение ранее достигнутых предпочтений при добавлении новых альтернатив и, тем самым, дает возможность оптимизации при обработке больших объемов динамически изменяемых данных, что существенно расширяет применимость популярного алгоритма.

Ключевые слова: *большие данные, корректный алгоритм ранжирования альтернатив, оптимизация алгоритма ранжирования большого набора альтернатив, показатели эффективности алгоритма попарных сравнений, алгоритмическая сложность*

Введение

Хорошо известна проблема, характерная для методов многокритериального выбора, использующих попарные сравнения альтернатив по каждому из критериев, а затем аддитивную скаляризацию полученных локальных приоритетов и весовых коэффициентов критериев [1—6]: при добавлении новой альтернативы ранее достигнутые предпочтения могут измениться [2, 4, 5]. Указанная нежелательная особенность классического метода анализа иерархий (Analytic Hierarchy Process (AHP)) ограничивает его применение для динамически пополняемых наборов данных большого объема (рекрутинг, оценивание проектов, ...).

Пример 1. Действительно, рассмотрев сначала две альтернативы A_1, A_2 по двум критериям C_1, C_2 с весовыми коэффициентами $w(C_1) = 0{,}4$; $w(C_2) = 0{,}6$ с матрицами парных сравнений (МПС) вида $\mathbf{A}_1 = \begin{bmatrix} 1 & 1/3 \\ 3 & 1 \end{bmatrix}$; $\mathbf{A}_2 = \begin{bmatrix} 1 & 4 \\ 1/4 & 1 \end{bmatrix}$, получим итоговый вектор предпочтений, из которого следует $w_{n=2}(A_1) = 0{,}58 > w_{n=2}(A_2) = 0{,}42$.

Увеличив множество альтернатив до трех и оставляя оценки первых двух альтернатив без изменения, получим, согласно классическому AHP, МПС вида

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 1/3 & 3 \\ 3 & 1 & 3 \\ 1/3 & 1/3 & 1 \end{bmatrix}; \ \mathbf{B}_2 = \begin{bmatrix} 1 & 4 & 1/3 \\ 1/4 & 1 & 1/8 \\ 3 & 8 & 1 \end{bmatrix},$$

приводящие к изменившимся предпочтениям относительно исходных двух альтернатив $w_{n=3}(A_1) = 0{,}266 < w_{n=3}(A_2) = 0{,}278, w_{n=3}(A_3) = 0{,}456$. В работе [5] предложен и далее неоднократно применен в задачах тестового распознавания и распознавания состояний динамических систем (например, [6]) алгоритм, ранее названный модификацией [6] классического метода анализа иерархий AHP [1]. Более точным будет утверждение, что ниже обсуждаемый алгоритм есть синтез классического AHP и корректного (безошибочного на выборке альтернатив) алгоритма парных соотношений координат собственных векторов как характеристик МПС.

В данном тексте на этот алгоритм будем ссылаться как на AHP-C (Correct algorithm for ranking alternatives by comparing the coordinates of eigenvectors of matrices of pairwise comparisons).

Для лучшего понимания идеологии алгоритма AHP-C приведем краткое его изложение [5, 6] и пример применения, поскольку заложенная в нем идея оценивания альтернатив и вычисления весовых коэффициентов в каждой их паре привела к возможности его оптимизации для больших наборов исходных данных, суть которой показана ниже.

Алгоритм AHP-C

Шаг 1. Для заданных наборов критериев $C = \{C_1, C_2, \ldots, C_m\}$, альтернатив $A = \{a_1, a_2, \ldots, a_n\}$ сначала конструируются МПС (как и в классическом AHP алгоритме); итогом первого

*Исследование выполнено при частичной поддержке РФФИ, проект № 20-08-00747.

шага является набор нормализованных оценок собственных векторов для каждой МПС:

$$\mathbf{W}_1 = \begin{Bmatrix} w_{1C_1} & \dots & w_{nC_1} \\ \dots & \dots & \dots \\ w_{1C_m} & \dots & w_{nC_m} \end{Bmatrix}, \quad (1)$$

где w_{iC_j} — оценка i-й альтернативы по C_j-му критерию, $i = \overline{1,n}$; $j = \overline{1,m}$.

Шаг 2. Формируется набор из матриц $\mathbf{W}_{2j}, j = \overline{1,m}$, элементами которых являются двухкомпонентные векторы:

$$\mathbf{W}_{2j} = \begin{Bmatrix} (w_{1C_j}, w_{1C_j}) & (w_{1C_j}, w_{2C_j})\dots & (w_{1C_j}, w_{nC_j}) \\ \dots & \dots & \dots \\ (w_{nC_j}, w_{1C_j}) & (w_{nC_j}, w_{2C_j})\dots & (w_{nC_j}, w_{nC_j}) \end{Bmatrix}, \quad (2)$$

$$j = \overline{1,m}.$$

Шаг 3. Нормализуются векторы $(\tilde{w}_{iC_j}, \tilde{w}_{kC_j})$, $i,k = \overline{1,n}$; $j = \overline{1,m}$, матриц $\mathbf{W}_{2j}, j = \overline{1,m}$, по формулам

$$\tilde{w}_{iC_j} = \frac{w_{iC_j}}{w_{iC_j} + w_{kC_j}}, \; \tilde{w}_{kC_j} = \frac{w_{kC_j}}{w_{iC_j} + w_{kC_j}}, \quad (3)$$

$$i,k = \overline{1,n}; \; j = \overline{1,m}.$$

Получаем совокупность матриц $\mathbf{W}_{3j}, j = \overline{1,m}$, элементами которых являются двухкомпонентные нормализованные векторы:

$$\mathbf{W}_{3j} = \begin{Bmatrix} (0,5;0,5) & (\tilde{w}_{1C_j}, \tilde{w}_{2C_j})\dots & (\tilde{w}_{1C_j}, \tilde{w}_{nC_j}) \\ \dots & \dots & \dots \\ (\tilde{w}_{nC_j}, \tilde{w}_{1C_j}) & (\tilde{w}_{nC_j}, \tilde{w}_{2C_j})\dots & (0,5;0,5) \end{Bmatrix}, \quad (4)$$

$$j = \overline{1,m}.$$

Шаг 4. Формируется итоговая матрица, элементы которой суть векторы из взвешенных по критериям C_j компонент матриц $\mathbf{W}_{3j}$:

$$\mathbf{W}_4 = \begin{Bmatrix} (0,5;0,5) & \left(\sum_{j=1}^m C_j \tilde{w}_{1C_j}, \sum_{j=1}^m C_j \tilde{w}_{2C_j}\right)\dots & \left(\sum_{j=1}^m C_j \tilde{w}_{1C_j}, \sum_{j=1}^m C_j \tilde{w}_{nC_j}\right) \\ \dots & \dots & \dots \\ \left(\sum_{j=1}^m C_j \tilde{w}_{nC_j}, \sum_{j=1}^m C_j \tilde{w}_{1C_j}\right) & \left(\sum_{j=1}^m C_j \tilde{w}_{nC_j}, \sum_{j=1}^m C_j \tilde{w}_{2C_j}\right)\dots & (0,5;0,5) \end{Bmatrix}, \quad (5)$$

элементы которой затем нормализуются по формулам, аналогичным (3). Нормализованные векторы полученной матрицы $\mathbf{U}$ обозначим как $(u_i^{(i,k)}, u_k^{(i,k)}), i,k = \overline{1,n}$. Здесь громоздкость обозначений вызвана необходимостью указать, что операция аддитивной скаляризации по критериям проводится для каждой пары $(i,k), i,k = \overline{1,n}$, отдельно:

$$u_i^{(i,k)} = \frac{\sum_{j=1}^m C_j \tilde{w}_{iC_j}}{\sum_{j=1}^m C_j \tilde{w}_{iC_j} + \sum_{j=1}^m C_j \tilde{w}_{kC_j}},$$

$$u_k^{(i,k)} = \frac{\sum_{j=1}^m C_j \tilde{w}_{kC_j}}{\sum_{j=1}^m C_j \tilde{w}_{iC_j} + \sum_{j=1}^m C_j \tilde{w}_{kC_j}}, \quad i,k = \overline{1,n}. \quad (6)$$

Шаг 5. Формируется итоговый вектор весовых коэффициентов по формулам (суммы по первым компонентам каждой строки матрицы $\mathbf{W}_5 = \|(u_i^{(i,k)}, u_k^{(i,k)})\|, i,k = \overline{1,n}$)

$$\omega_{a_1} = \sum_{k=1}^n u_1^{(1,k)}, \omega_{a_2} = \sum_{k=1}^n u_2^{(2,k)}, \dots, \omega_{a_n} = \sum_{k=1}^n u_n^{(n,k)}, \quad (7)$$

который при необходимости дальнейшего анализа полученного ранжирования нормализуется: $\mathbf{W}^* = (\omega_{a_1}^*, \omega_{a_2}^*, \dots, \omega_{a_n}^*)$.

Изложенный подход позволил применять идеологию AHP-C для динамически пополняемых наборов альтернатив, на заботясь о проверке соблюдения ранее достигнутых предпочтений.

Пример 2. Приведем теперь кратко шаги AHP-C в условиях выше рассмотренного примера 1.

Согласно AHP-C (шаг 1) исходными данными для работы алгоритма являются нормализованные собственные векторы матриц $\mathbf{B}_1$, $\mathbf{B}_2$, здесь равные

$$\mathbf{W}_{11}: \; 0,281 \quad 0,584 \quad 0,135;$$
$$\mathbf{W}_{12}: \; 0,256 \quad 0,073 \quad 0,671.$$

Здесь и ниже первый индекс в $\mathbf{W}_{ij}$ указывает на номер шага алгоритма, второй — на номер критерия.

Далее формируются матрицы сравнений координат полученных собственных векторов, затем все полученные двумерные векторы нормализуются (шаги 2, 3).

Для полученного выше собственного вектора $\mathbf{W}_{11} = (0,281;\ 0,584;\ 0,135)$ исходная $\mathbf{W}_{21}$ и нормализованная $\mathbf{W}_{31}$ матрицы попарного сравнения его координат имеют вид, соответственно:

$$\mathbf{W}_{21} = \begin{pmatrix} (1;1) & (0,281;0,584) & (0,281;0,135) \\ (0,584;0,281) & (1;1) & (0,584;0,135) \\ (0,135;0,281) & (0,135;0,584) & (1;1) \end{pmatrix};$$

$$\mathbf{W}_{31} = \begin{pmatrix} (0,5;0,5) & (\boxed{0,325};0,675) & (\boxed{0,675};0,325) \\ (0,675;0,325) & (0,5;0,5) & (0,812;0,188) \\ (0,325;0,675) & (0,188;0,812) & (0,5;0,5) \end{pmatrix}.$$

Аналогичные матрицы получаем для собственного вектора $\mathbf{W}_{12} = (0,256;\ 0,073;\ 0,671)$:

$$\mathbf{W}_{22} = \begin{pmatrix} (1;1) & (0,256;0,073) & (0,256;0,671) \\ (0,073;0,256) & (1;1) & (0,073;0,671) \\ (0,671;0,256) & (0,671;0,073) & (1;1) \end{pmatrix};$$

$$\mathbf{W}_{32} = \begin{pmatrix} (0,5;0,5) & (\boxed{0,777};0,223) & (\boxed{0,276};0,724) \\ (0,223;0,777) & (0,5;0,5) & (0,098;0,902) \\ (0,724;0,276) & (0,902;0,098) & (0,5;0,5) \end{pmatrix}.$$

Замечание 1. В каждой паре фиксированная альтернатива имеет свой коэффициент значимости, определяющий вклад в результат сравнения именно в данной паре. В рамки обведены значения, участвующие в формуле их скаляризации как линейной свертки по весовым коэффициентам критериев в каждой паре нормализованных векторов матриц $\mathbf{W}_{31}$, $\mathbf{W}_{32}$.

На шаге 4 осуществляется операция аддитивной свертки по критериям и альтернативам в *каждой паре* (главную диагональ по понятной причине не меняем):

$$\mathbf{W}_4 = \begin{Bmatrix} (0,5;0,5) & w_{4(1,2)} & w_{4(1,3)} \\ w_{4(2,1)} & (0,5;0,5) & w_{4(2,3)} \\ w_{4(3,1)} & w_{4(3,2)} & (0,5;0,5) \end{Bmatrix},$$

где

$$w_{4(1,2)} = \left(\sum_{j=1}^{m} C_j \tilde{w}_{1C_j}, \sum_{j=1}^{m} C_j \tilde{w}_{2C_j}\right) =$$
$$= \left(\boxed{W_{C_1} 0,325 + W_{C_2} 0,777}; W_{C_1} 0,675 + W_{C_2} 0,223\right);$$

$$w_{4(1,3)} = \left(\sum_{j=1}^{m} C_j \tilde{w}_{1C_j}, \sum_{j=1}^{m} C_j \tilde{w}_{3C_j}\right) =$$
$$= \left(\boxed{W_{C_1} 0,675 + W_{C_2} 0,276}; W_{C_1} 0,325 + W_{C_2} 0,724\right);$$

$$w_{4(2,3)} = \left(\sum_{j=1}^{m} C_j \tilde{w}_{2C_j}, \sum_{j=1}^{m} C_j \tilde{w}_{3C_j}\right) =$$
$$= \left(W_{C_1} 0,812 + W_{C_2} 0,098; W_{C_1} 0,188 + W_{C_2} 0,902\right).$$

Положив $w(C_1) = 0,4$; $w(C_2) = 0,6$ (см. пример 1), получаем матрицу $\mathbf{W}_4$ в явном виде и ее нормализованный аналог $\mathbf{U}$, соответственно:

$$\mathbf{U} = \begin{pmatrix} (0,5;0,5) & (0,596;0,404) & (0,436;0,564) \\ (0,404;0,596) & (0,5;0,5) & (0,384;0,616) \\ (0,564;0,436) & (0,616;0,384) & (0,5;0,5) \end{pmatrix}.$$

Согласно формуле (7) (шаг 5) получаем итоговый вектор весовых коэффициентов альтернатив:

$$\omega_{A_1} = \sum_{k=1}^{3} u_1^{(1,k)} = 0,5 + 0,596 + 0,436 = 1,532;$$

$$\omega_{A_2} = \sum_{k=1}^{3} u_2^{(2,k)} = 0,404 + 0,5 + 0,384 = 1,288;$$

$$\omega_{A_3} = \sum_{k=1}^{3} u_3^{(3,k)} = 0,564 + 0,616 + 0,5 = 1,680;$$

затем его нормализуем: $\mathbf{W}^* = (0,340;0,286;0,373)$.

Сравнение с классическим алгоритмом AHP $\mathbf{W}^{AHP} = (0,266;0,278;0,456)$, применение которого нарушило ранее достигнутое предпочтение $(A_1 \succ A_2)$, говорит в пользу алгоритма AHP-C.

Мотивация и постановка задачи оптимизации алгоритма AHP-C

Даны множество альтернатив $A = \{A_1, \ldots, A_n\}$ (n велико, $n \sim 1000$ и более) и множество критериев $C = \{C_1, \ldots, C_m\}$. Требуется получить ранжированный список альтернатив A^*, свободный от недостатка изменений ранее достигнутых предпочтений при добавлении новых альтернатив, при этом желательно, чтобы алгоритмическая сложность искомого алгоритма ранжирования была соизмерима с логарифмической.

Теоретически алгоритм AHP-C применим и для больших данных, однако его алгоритмическая сложность и требуемый достаточно большой объем оперативной памяти представляют собой определенное препятствие для практического применения, поскольку дополнительно к большому объему альтернатив (ранжирование клиентов, проектов, принятия решений в рекрутинге и т. д.) имеется необходимость получения достаточного объема сравнительной экспертной информации.

Пример 3. Задача выбора из 5000 альтернатив и двух критериев для числа с двойной точностью, размер которого 8 байт, характеризуется показателями (реализация выполнена на языке Java):

- $5000 \cdot 5000 \cdot 2 \cdot 8$ = 381,47 Мбайт — для МПС алгоритма AHP;
- $5000 \cdot 5000 \cdot 2 \cdot 2 \cdot 8$ = 762,94 Мбайт — для матриц относительных локальных приоритетов, требуемых для алгоритма AHP-C;

- $5000 \cdot 5000 \cdot 2 \cdot 8 = 381{,}47$ Мбайт — для обобщенной матрицы **U**, компонентами которой являются локальные весовые коэффициенты альтернатив относительно всей совокупности критериев.

Замечание 2. Полученные данные могут несколько отличаться в зависимости от выбранного типа данных для хранения вычислений и реализации. В приведенном выше примере 3 для хранения пар значений в массиве использовались ссылочные объекты. Размер объекта в этой конкретной среде выполнения рассчитывается как сумма размера ссылки и размера полей объекта. В нашем случае это 8 байт на хранение ссылки и 16 байт на хранение двух значений типа double. Если округлять размер данных до размера, кратному машинному слову [7, 8], тогда итоговый объем оперативной памяти вырастет не менее, чем в полтора раза.

Если данные МПС будут сразу удаляться (что для сред выполнения с автоматическим удалением неиспользуемых объектов имеет место), все равно потребуется около 3 Гб только на хранение промежуточного результата (здесь не принимается в расчет объем памяти для хранения информации об исходных данных).

Традиционно задача получения ранжированного списка в многокритериальной задаче решалась путем применения какого-либо алгоритма вычисления весовых коэффициентов альтернатив и последующей сортировкой, согласованной с их значениями. Далее будет показано, как изменение этой последовательности действий позволит снять ограничение на мощность множества альтернатив и ускорить вычисление.

Решение задачи

Для решения проблемы корректного применения AHP-C на большом наборе альтернатив предлагается декомпозиция исходной задачи выбора на всем множестве альтернатив на последовательность задач выбора на паре альтернатив, выполнение каждой из которых интерпретируется как функция сравнения двух элементов согласно обычному алгоритму сортировки числовой последовательности (например, вставками [9]).

Любой алгоритм сортировки базируется на сравнении элементов по какому-либо признаку(-ам). В связке с алгоритмом AHP-C сортировка будет происходить как последовательность следующих действий:

— получить два элемента из набора A согласно алгоритму сортировки;

— рассчитать для этой пары веса путем AHP-C;

—если $a_i > a_j$, выполнить перестановку, иначе перейти к следующему шагу.

Далее представлена реализация с использованием библиотечного метода sort, куда мы передаем функцию сравнения с AHP-C

```
alternatives.sort((o1, o2) -> {// o1, o2 - текущие элементы сравнения в сортировке
List<Alternative> list = new ArrayList<>();
list.add(o1);// добавляем в список
list.add(o2);
calculateWeight(list, criteriaList); // вычисляем веса
return Double.compare(list.get(1).getWeight(), list.get(0).getWeight()); //return a1>a0
});
```

Очевидно, что такой подход предъявляет свои требования к используемым методам решения многокритериальных задач. В первую очередь это корректность получаемых результатов вычислений исходного алгоритма. Под корректностью здесь понимается отсутствие нарушения бинарных отношений транзитивности $a > b$ и $b > c \rightarrow a > c$, выражающих предпочтения альтернатив в отсортированной части ранжируемого набора.

На рис. 1 это явление проиллюстрировано на примере сортировки вставками: элемент $a[6]$ встал на неверную позицию, так как при контрольной проверке расчета весов в группе из первых трех элементов выяснилось, что он меньше 15.

Указанное условие означает, что при внесении во множество альтернатив нового элемента не должны нарушаться отношения доминирования между уже находящимися во множестве элементами.

Таким образом, рассматриваемый алгоритм оптимизации (назовем его AHP-CS — от Sorting an array of data) использует последовательные сравнения пар для расчета весовых коэффициентов всего множества альтернатив, что стало возможным только благодаря корректной модификации алгоритма AHP-C, которому отведена роль части сортирующей функции в новом алгоритме AHP+CS.

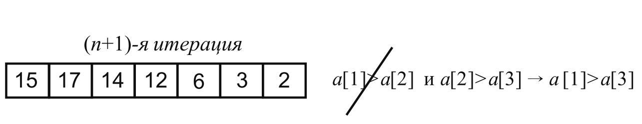

Рис. 1. Иллюстрация нарушения порядка следования

Численное моделирование алгоритмов AHP-C и AHP-CS. Анализ результатов

Алгоритмическая сложность алгоритма получения решения теперь зависит исключительно от числа сравнений в выбранном способе сортировки. На графике (рис. 2) приведены результаты тестирования времени выполнения AHP-C и AHP-CS на одинаковых наборах альтернатив. На рис. 3 также приведен график зависимости времени выполнения для AHP-CS на больших наборах альтернатив.

Требования к единовременно используемой оперативной памяти снижены. Теперь не нужно хранить результаты вычислений за исключением итоговых весов. Остальные объекты могут удаляться сборщиком мусора. К примеру, тест на 500 000 альтернатив и 2 критерия показал, что используется в тот момент только 140 Мб (рис. 4).

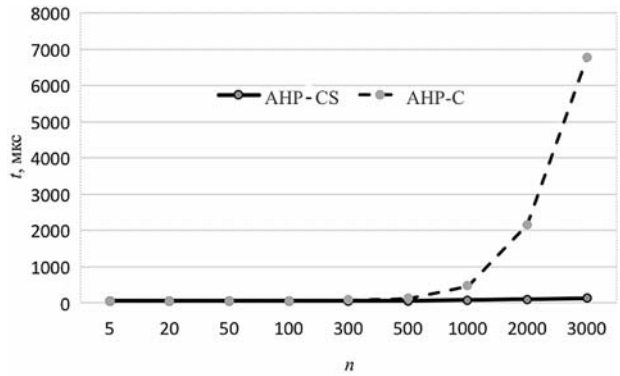

Рис. 2. График сравнения времени *t* выполнения для AHP-C и AHP-CS в зависимости от числа *n* альтернатив

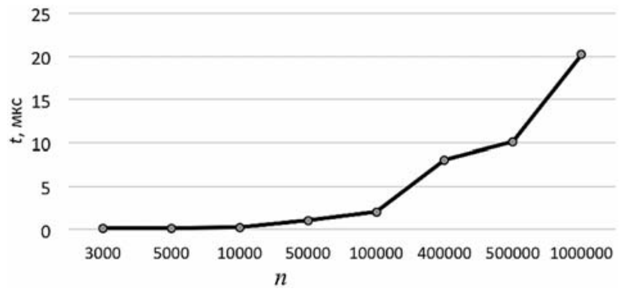

Рис. 3. График зависимости времени *t* выполнения от числа *n* альтернатив для AHP-CS

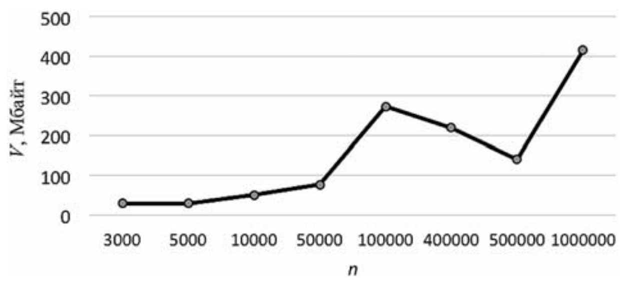

Рис. 4. График потребления объема *V* оперативной памяти в зависимости от числа *n* альтернатив для AHP-CS

Заключение

Основным результатом данной работы является распространение одного из самых популярных (по числу публикаций) методов решения многокритериальных задач выбора на наборы альтернатив и критериев большого объема в целях повышения скорости расчетов и снижения требований к вычислительным ресурсам. Поставленная задача оптимизации модифицированного алгоритма AHP-C выполнена на основе последовательного просмотра пар и их упорядочивания на принципах сортировки числового массива.

Исследуются алгоритмические сложности и затраты оперативной памяти существующих алгоритмов. Поясняются требования к алгоритмам с точки зрения корректности их работы на динамических множествах альтернатив.

Приводится иллюстративный пример и результаты тестирования предложенного алгоритма.

Важно заметить, что предложенная в статье модификация классического метода AHP стала возможной только при наличии алгоритма AHP-C, заведомо свободного от некорректного выбора в динамически изменяющихся наборах.

Согласно автору быстрой сортировки, "преждевременная оптимизация — корень всех бед" [10], поэтому осторожность в осуществлении любой оптимизации заключается в создании первоначально корректно работающего алгоритма (здесь — AHP-C).

Список литературы

1. **Саати Т. Л.** Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1989. 311 с.
2. **Эрроу К. Дж.** Коллективный выбор и индивидуальные ценности. М.: ГУ ВШЭ, 2004. 204 с.
3. **Подиновский В. В., Ногин В. Д.** Парето-оптимальные решения многокритериальных задач. М.: Наука, 2007.
4. **Ногин В. Д.** Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев // Журнал вычислительной математики и математической физики. 2004. Т. 44, № 7. С. 1259—1268.
5. **Самохвалов Ю. Я.** Групповой учет относительного превосходства альтернатив в задачах принятия решений // Кибернетика и системный анализ. 2003. № 6. С. 141—145.
6. **Колесникова С. И.** Особенности применения линейной свертки критериев в методе парных сравнений // Информационные технологии. 2011. № 1. С. 24—29.
7. **Oracle** documentation: Официальный сайт. URL: https://docs.oracle.com/en (дата обращения: 08.09.2020).
8. **Лафоре Р.** Структуры данных и алгоритмы Java. СПБ.: Питер, 2018. 704 с.
9. **Кнут Д. Э.** Искусство программирования. Т. 3. Сортировка и поиск / Пер. под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). М.: Вильямс, 2007. Т. 3. 832 с.
10. **Hoare C.** Quicksort // The Computer Journal. 1962. Vol. 5, N. 1. P. 10—16. DOI:10.1093/comjnl/5.1.10.

ПРИЛОЖЕНИЕ

Задача и ее решение на основе AHP-C и AHP-CS. Выполнить отбор наиболее перспективного кандидата на должность секретаря-референта по следующим критериям (табл. П1):

- уровень владения иностранным языком (ИЯ) — C_1;
- опыт работы — C_2;
- уровень владения офисными инструментами (ПК) — C_3.

Таблица П1

Исходные данные

Номер альтернативы	Уровень владения ИЯ, в баллах	Опыт работы, в годах	Уровень владения ПК, в баллах
1	7	9	5
2	8	10	8
3	9	5	2
4	1	5	9
5	2	3	5
6	8	8	4
7	3	7	4
8	7	10	3

Пусть $c_1 = 0{,}2$, $c_2 = c_3 = 0{,}4$, где $c_i = W(C_i)$ — весовой коэффициент i-го критерия, $i = 1,2,3$.

Этап 1. Реализация первых шагов классического AHP — составление матриц парных сравнений и получение соответствующих весовых оценок альтернатив по каждому критерию (расчеты сделаны с точностью до 10^{-3}).

МПС 1. Уровень владения ИЯ.

1,000 0,875 0,778 7,000 3,500 0,875 2,333 1,000
1,143 1,000 0,889 8,000 4,000 1,000 2,667 1,143
1,286 1,125 1,000 9,000 4,500 1,125 3,000 1,286
0,143 0,125 0,111 1,000 0,500 0,125 0,333 0,143
0,286 0,250 0,222 2,000 1,000 0,250 0,667 0,286
1,143 1,000 0,889 8,000 4,000 1,000 2,667 1,143
0,429 0,375 0,333 3,000 1,500 0,375 1,000 0,429
1,000 0,875 0,778 7,000 3,500 0,875 2,333 1,000

ВКА-1

$$W_{C_1} = (w_{1C_1}, \ldots, w_{8C_1}) =$$
$$= (0{,}156;\ 0{,}178;\ 0{,}200;\ 0{,}022; 0{,}044;\ 0{,}178;\ 0{,}067;\ 0{,}156).$$

МПС 2. Опыт работы в годах.

1,000 0,900 1,800 1,800 3,000 1,125 1,286 0,900
1,111 1,000 2,000 2,000 3,333 1,250 1,429 1,000
0,556 0,500 1,000 1,000 1,667 0,625 0,714 0,500
0,556 0,500 1,000 1,000 1,667 0,625 0,714 0,500
0,333 0,300 0,600 0,600 1,000 0,375 0,429 0,300
0,889 0,800 1,600 1,600 2,667 1,000 1,143 0,800
0,778 0,700 1,400 1,400 2,333 0,875 1,000 0,700
1,111 1,000 2,000 2,000 3,333 1,250 1,429 1,000

ВКА-2

$$W_{C_2} = (w_{1C_2}, \ldots, w_{8C_2}) =$$
$$= (0{,}158;\ 0{,}175; 0{,}088;\ 0{,}088; 0{,}053;\ 0{,}140;\ 0{,}123;\ 0{,}175).$$

МПС 3. Уровень владения ПК

1,000 0,625 2,500 0,556 1,000 1,250 1,250 1,667
1,600 1,000 4,000 0,889 1,600 2,000 2,000 2,667
0,400 0,250 1,000 0,222 0,400 0,500 0,500 0,667
1,800 1,125 4,500 1,000 1,800 2,250 2,250 3,000
1,000 0,625 2,500 0,556 1,000 1,250 1,250 1,667
0,800 0,500 2,000 0,444 0,800 1,000 1,000 1,333
0,800 0,500 2,000 0,444 0,800 1,000 1,000 1,333
0,600 0,375 1,500 0,333 0,600 0,750 0,750 1,000

ВКА-3

$$W_{C_3} = (w_{1C_3}, \ldots, w_{8C_3}) =$$
$$= (0{,}125;\ 0{,}200;\ 0{,}050;\ 0{,}225;\ 0{,}125;\ 0{,}100;\ 0{,}100;\ 0{,}075).$$

Этап 2. Реализация корректной модификации AHP-C на основе операции попарного сравнения элементов каждого вектора ВКА-i, $i = 1, 2, 3$, с *последующей нормализацией в каждой паре.*

Таблица П2

Ранжированный список альтернатив по трем алгоритмам

Ранг	AHP-C	AHP	AHP-CS
1	2	2	2
2	1	1	1
3	6	6	6
4	8	8	8
5	7	4	7
6	4	7	4
7	3	3	3
8	5	5	5

Относительные весовые коэффициенты (ОВК) на основе ВКА-1:

(1,000;1,000) (0,158;0,175) (0,158;0,088) (0,158;0,088) (0,158;0,053) (0,158;0,140) (0,158;0,123) (0,158;0,175)
(0,175;0,158) (1,000;1,000) (0,175;0,088) (0,175;0,088) (0,175;0,053) (0,175;0,140) (0,175;0,123) (0,175;0,175)
(0,088;0,158) (0,088;0,175) (1,000;1,000) (0,088;0,088) (0,088;0,053) (0,088;0,140) (0,088;0,123) (0,088;0,175)
(0,088;0,158) (0,088;0,175) (0,088;0,088) (1,000;1,000) (0,088;0,053) (0,088;0,140) (0,088;0,123) (0,088;0,175)
(0,053;0,158) (0,053;0,175) (0,053;0,088) (0,053;0,088) (1,000;1,000) (0,053;0,140) (0,053;0,123) (0,053;0,175)
(0,140;0,158) (0,140;0,175) (0,140;0,088) (0,140;0,088) (0,140;0,053) (1,000;1,000) (0,140;0,123) (0,140;0,175)
(0,123;0,158) (0,123;0,175) (0,123;0,088) (0,123;0,088) (0,123;0,053) (0,123;0,140) (1,000;1,000) (0,123;0,175)
(0,175;0,158) (0,175;0,175) (0,175;0,088) (0,175;0,088) (0,175;0,053) (0,175;0,140) (0,175;0,123) (1,000;1,000)

Нормализованные ОВК-1 на основе ВКА-1:

(0,500;0,500) (0,474;0,526) (0,643;0,357) (0,643;0,357) (0,750;0,250) (0,529;0,471) (0,563;0,438) (0,474;0,526)
(0,526;0,474) (0,500;0,500) (0,667;0,333) (0,667;0,333) (0,769;0,231) (0,556;0,444) (0,588;0,412) (0,500;0,500)
(0,357;0,643) (0,333;0,667) (0,500;0,500) (0,500;0,500) (0,625;0,375) (0,385;0,615) (0,417;0,583) (0,333;0,667)
(0,357;0,643) (0,333;0,667) (0,500;0,500) (0,500;0,500) (0,625;0,375) (0,385;0,615) (0,417;0,583) (0,333;0,667)
(0,250;0,750) (0,231;0,769) (0,375;0,625) (0,375;0,625) (0,500;0,500) (0,273;0,727) (0,300;0,700) (0,231;0,769)
(0,471;0,529) (0,444;0,556) (0,615;0,385) (0,615;0,385) (0,727;0,273) (0,500;0,500) (0,533;0,467) (0,444;0,556)
(0,438;0,563) (0,412;0,588) (0,583;0,417) (0,583;0,417) (0,700;0,300) (0,467;0,533) (0,500;0,500) (0,412;0,588)
(0,526;0,474) (0,500;0,500) (0,667;0,333) (0,667;0,333) (0,769;0,231) (0,556;0,444) (0,588;0,412) (0,500;0,500)

Относительные весовые коэффициенты на основе ВКА-2:

(1,000;1,000) (0,125;0,200) (0,125;0,050) (0,125;0,225) (0,125;0,125) (0,125;0,100) (0,125;0,100) (0,125;0,075)
(0,200;0,125) (1,000;1,000) (0,200;0,050) (0,200;0,225) (0,200;0,125) (0,200;0,100) (0,200;0,100) (0,200;0,075)
(0,050;0,125) (0,050;0,200) (1,000;1,000) (0,050;0,225) (0,050;0,125) (0,050;0,100) (0,050;0,100) (0,050;0,075)
(0,225;0,125) (0,225;0,200) (0,225;0,050) (1,000;1,000) (0,225;0,125) (0,225;0,100) (0,225;0,100) (0,225;0,075)
(0,125;0,125) (0,125;0,200) (0,125;0,050) (0,125;0,225) (1,000;1,000) (0,125;0,100) (0,125;0,100) (0,125;0,075)
(0,100;0,125) (0,100;0,200) (0,100;0,050) (0,100;0,225) (0,100;0,125) (1,000;1,000) (0,100;0,100) (0,100;0,075)
(0,100;0,125) (0,100;0,200) (0,100;0,050) (0,100;0,225) (0,100;0,125) (0,100;0,100) (1,000;1,000) (0,100;0,075)
(0,075;0,125) (0,075;0,200) (0,075;0,050) (0,075;0,225) (0,075;0,125) (0,075;0,100) (0,075;0,100) (1,000;1,000)

Нормализованные ОВК-2 на основе ВКА-2:

(0,500;0,500) (0,385;0,615) (0,714;0,286) (0,357;0,643) (0,500;0,500) (0,556;0,444) (0,556;0,444) (0,625;0,375)
(0,615;0,385) (0,500;0,500) (0,800;0,200) (0,471;0,529) (0,615;0,385) (0,667;0,333) (0,667;0,333) (0,727;0,273)
(0,286;0,714) (0,200;0,800) (0,500;0,500) (0,182;0,818) (0,286;0,714) (0,333;0,667) (0,333;0,667) (0,400;0,600)
(0,643;0,357) (0,529;0,471) (0,818;0,182) (0,500;0,500) (0,643;0,357) (0,692;0,308) (0,692;0,308) (0,750;0,250)
(0,500;0,500) (0,385;0,615) (0,714;0,286) (0,357;0,643) (0,500;0,500) (0,556;0,444) (0,556;0,444) (0,625;0,375)
(0,444;0,556) (0,333;0,667) (0,667;0,333) (0,308;0,692) (0,444;0,556) (0,500;0,500) (0,500;0,500) (0,571;0,429)
(0,444;0,556) (0,333;0,667) (0,667;0,333) (0,308;0,692) (0,444;0,556) (0,500;0,500) (0,500;0,500) (0,571;0,429)
(0,375;0,625) (0,273;0,727) (0,600;0,400) (0,250;0,750) (0,375;0,625) (0,429;0,571) (0,429;0,571) (0,500;0,500)

Относительные весовые коэффициенты на основе ВКА-3:

(1,000;1,000) (0,156;0,178) (0,156;0,200) (0,156;0,022) (0,156;0,044) (0,156;0,178) (0,156;0,067) (0,156;0,156)
(0,178;0,156) (1,000;1,000) (0,178;0,200) (0,178;0,022) (0,178;0,044) (0,178;0,178) (0,178;0,067) (0,178;0,156)
(0,200;0,156) (0,200;0,178) (1,000;1,000) (0,200;0,022) (0,200;0,044) (0,200;0,178) (0,200;0,067) (0,200;0,156)
(0,022;0,156) (0,022;0,178) (0,022;0,200) (1,000;1,000) (0,022;0,044) (0,022;0,178) (0,022;0,067) (0,022;0,156)
(0,044;0,156) (0,044;0,178) (0,044;0,200) (0,044;0,022) (1,000;1,000) (0,044;0,178) (0,044;0,067) (0,044;0,156)
(0,178;0,156) (0,178;0,178) (0,178;0,200) (0,178;0,022) (0,178;0,044) (1,000;1,000) (0,178;0,067) (0,178;0,156)
(0,067;0,156) (0,067;0,178) (0,067;0,200) (0,067;0,022) (0,067;0,044) (0,067;0,178) (1,000;1,000) (0,067;0,156)
(0,156;0,156) (0,156;0,178) (0,156;0,200) (0,156;0,022) (0,156;0,044) (0,156;0,178) (0,156;0,067) (1,000;1,000)

Нормализованные ОВК-3 на основе ВКА-3:

(0,500;0,500) (0,467;0,533) (0,438;0,563) (0,875;0,125) (0,778;0,222) (0,467;0,533) (0,700;0,300) (0,500;0,500)
(0,533;0,467) (0,500;0,500) (0,471;0,529) (0,889;0,111) (0,800;0,200) (0,500;0,500) (0,727;0,273) (0,533;0,467)
(0,563;0,438) (0,529;0,471) (0,500;0,500) (0,900;0,100) (0,818;0,182) (0,529;0,471) (0,750;0,250) (0,563;0,438)
(0,125;0,875) (0,111;0,889) (0,100;0,900) (0,500;0,500) (0,333;0,667) (0,111;0,889) (0,250;0,750) (0,125;0,875)
(0,222;0,778) (0,200;0,800) (0,182;0,818) (0,667;0,333) (0,500;0,500) (0,200;0,800) (0,400;0,600) (0,222;0,778)
(0,533;0,467) (0,500;0,500) (0,471;0,529) (0,889;0,111) (0,800;0,200) (0,500;0,500) (0,727;0,273) (0,533;0,467)
(0,300;0,700) (0,273;0,727) (0,250;0,750) (0,750;0,250) (0,600;0,400) (0,273;0,727) (0,500;0,500) (0,300;0,700)
(0,500;0,500) (0,467;0,533) (0,438;0,563) (0,875;0,125) (0,778;0,222) (0,467;0,533) (0,700;0,300) (0,500;0,500)

Этап 3. Итоговая реализация корректной модификации AHP-CS. Получение линейной свертки:

(0,500;0,500) (0,437;0,563) (0,630;0,370) (0,575;0,425) (0,656;0,344) (0,527;0,473) (0,587;0,413) (0,539;0,461)
(0,563;0,437) (0,500;0,500) (0,681;0,319) (0,633;0,367) (0,714;0,286) (0,589;0,411) (0,647;0,353) (0,598;0,402)
(0,370;0,630) (0,319;0,681) (0,500;0,500) (0,453;0,547) (0,528;0,472) (0,393;0,607) (0,450;0,550) (0,406;0,594)
(0,425;0,575) (0,367;0,633) (0,547;0,453) (0,500;0,500) (0,574;0,426) (0,453;0,547) (0,494;0,506) (0,458;0,542)
(0,344;0,656) (0,286;0,714) (0,472;0,528) (0,426;0,574) (0,500;0,500) (0,371;0,629) (0,422;0,578) (0,387;0,613)
(0,473;0,527) (0,411;0,589) (0,607;0,393) (0,547;0,453) (0,629;0,371) (0,500;0,500) (0,559;0,441) (0,513;0,487)
(0,413;0,587) (0,353;0,647) (0,550;0,450) (0,506;0,494) (0,578;0,422) (0,441;0,559) (0,500;0,500) (0,453;0,547)
(0,461;0,539) (0,402;0,598) (0,594;0,406) (0,542;0,458) (0,613;0,387) (0,487;0,513) (0,547;0,453) (0,500;0,500)

S. I. Kolesnikova, D. Sc., skolesnikova@yandex.ru, **S. A. Karavanova,** Master's Student,
Saint-Petersburg State University of Aerospace Instrumentation, St. Petersburg, 190000, Russian Federation

Optimization of Multi-Criterial Selection Algorithm with a Dynamically Filled Large Set of Alternatives

We consider the problem of the correct ranking of a dynamically replenished large set of alternatives in multicriteria choice problems that use in the solution the previously obtained modified method for analyzing hierarchies, based on the operation of additive convolution of local priorities not on the obtained set of characteristics of paired comparison matrices (as in the classical method), but on a set of pairs the relative weights of the coordinates of the eigenvectors being compared with each other for each criterion and the subsequent operation of additive convolution according to the criteria and alternatives in each pair. In this version, the algorithm ensures that previously achieved preferences are preserved when adding new alternatives and, thereby, makes it possible to optimize when processing large volumes of dynamically changing data, which significantly expands the applicability of the popular algorithm.

Keywords: *big data, correct algorithm for ranking alternatives, optimization of the algorithm for ranking a large set of alternatives, performance indicators of the algorithm for pairwise comparisons; algorithmic complexity*

Acknowledgements: The reported study was funded by RFBR according to the research project № 20-08-00747

DOI: 10.17587/it.27.235-241

References

1. **Saaty L. T.** Decision making. The analytic hierarchy process, Moskow, Radio i svyaz, 1989, 311 p. (in Russian).
2. **Arrow K. J.** Collective choice and individual values, Moskow, Nacionalnyj issledovatelski universitet Vysshej shkoly ekonomiki, 2004, 204 p. (in Russian).
3. **Podinovskii V. V., Noghin V. D.** Pareto-optimal Decisions in Multicriteria Optimization Problems, Moscow, Nauka, 2007, 256 p. (in Russian).
4. **Noghin V. D.** Simplified version of the hierarchy analysis method based on nonlinear convolution of criteria, *Zhurnal Vychislitelnoj Matematiki i Matematicheskoj Fiziki*, 2004, vol. 44, no 7, pp. 1259—1268 (in Russian).
5. **Samohvalov U. Ya.** Group accounting for the relative superiority of alternatives in decision-making problems, *Kibernetika i Sistemny Analiz*, 2003, no. 6, pp. 141—145 (in Russian).
6. **Kolesnikova S. I.** Features of the application of linear convolution of criteria in the method of paired comparisons, *Informacionnye Tekhnologii*, 2003, no. 1, pp. 24—29 (in Russian).
7. Oracle documentation: Official site. URL: https://docs.oracle.com/en (date accessed: 09/08/2020).
8. **Laphore R.** Java data structures and algorithms, St. Petersburg, Piter, 2018, 704 p. (in Russian).
9. **Knut D. E.** The art of programming, trans. ed. V. T. Tertyshny (ch. 5) and I. V. Krasikov (ch. 6), Moskow, Williams, 2007, vol. 3, 832 p. (in Russian).
10. **Hoare C.** Quicksort, *The Computer Journal*, 1962, vol. 5, no. 1, pp. 10—16.