ЗАДАЧА ВЫБОРА ВАРИАНТА РАЗМЕЩЕНИЯ «МАГАЗИНА У ДОМА» В УСЛОВИЯХ МНОГОКРИТЕРИАЛЬНОСТИ

Конкурс

Бакалавриат

Аннотация

В работе рассматривается математическая постановка многокритериальной задачи принятия решения. В подобного рода задачах для поиска оптимума применяются различные методы, в частности, метод взвешенной суммы или TOPSIS. Зная важности имеющихся критериев, можно получить численные оценки для множества конкурирующих вариантов решения, проранжировать их и найти наилучший. Также для работы с интервальными значениями критериев обозревается интервальная модификация TOPSIS. В результате приложения данных методов была решена практическая задача поиска оптимальной альтернативы для размещения «магазина у дома» с четкими и интервальными входными данными.

Ключевые слова

Многокритериальная оптимизация, TOPSIS, взвешенная сумма, интервальные методы

Тематическое направление

Математическая экономика

JEL

C44, C61, D81

Содержание

1 Постановка многокритериальной задачи принятия решения

Необходимость в принятии обоснованных решений постоянно возникает в практической деятельности человека. Часто в подобных задачах лицу, принимающему решение, нужно среди нескольких альтернатив выбрать наилучшую. В обыденных ситуациях, человек способен без помощи сложных расчетов хорошо справляться с задачей выбора. Тем не менее, в силу особенностей механизмов мышления люди склонны к необоснованным упрощениям или игнорированию некоторых аспектов задачи, если она приобретает более сложный характер с точки зрения количества альтернатив или критериев оценки [1, с. 24 — 25]. Следовательно, когда количество различных вариантов велико или их оценка требует рассмотрения множества параметров, существует необходимость в использовании математических методов для оптимального решения задачи.

В рассматриваемой задаче альтернативные варианты характеризуются определенными значениями критериев оценки. В свою очередь, критериям присвоена относительная степень важности.

Качество альтернатив может оцениваться по одному или нескольким критериям. Когда критерий только один, задача обычно сводится к поиску альтернативы с наибольшим (наименьшим) значением критерия, если большие значения критерия являются более предпочтительными (или наоборот) [1, с. 27]. Если критериев несколько, то выбор оптимального решения представляет более сложную задачу, называемую многокритериальной.

Решение многокритериальной задачи может быть осложнено многими факторами и требовать нескольких этапов решения. Прежде всего, критерии могут иметь разную шкалу, что затрудняет агрегирование и сопоставление альтернатив между собой. Более того, трудности образуются вследствие нечеткости оценки критериев, ведь критерию может быть присвоено интервальное значение. Если же критерию дана качественная характеристика, существует необходимость в ее количественном выражении или же в ранжировке предпочтений относительно данных признаков и т. д.

Для решения многокритериальной задачи в данной постановке будут рассмотрены и сравнены методы взвешенной суммы критериев (МВСК) и метод TOPSIS. Определим задачу математически.

Пусть X множество всех альтернатив. Тогда $\varphi_1, \varphi_2, \ldots, \varphi_m, m > 1$ — функции, задающие m (частных) критериев, определенных на множестве X, каждый из критериев имеет шкалу $Z_i, i = \overline{1, m}$. Критериальной (векторной) оценкой альтернативы $x \in X$ называется вектор $y = \varphi(x) = (\varphi_1(x), \ldots, \varphi_m(x))$. Множество $Y = \varphi(X) = \{y | y = \varphi(x), x \in X\}$ задает все возможные критериальные оценки. Веса критериев обозначаются как $w = (w_1, w_2, \ldots, w_m) : \sum_{i=1}^m w_i = 1$. Для удобства задачу можно представить в виде таблицы 1.

Таблица 1: Представление задачи в форме таблицы

		Альтернативы			
Веса критериев	Критерии	A_1	A_2	$\ldots$	A_n
w_1	φ_1	$\varphi_1(A_1)$	$\varphi_1(A_2)$	$\ldots$	$\varphi_1(A_n)$
w_2	φ_2	$\varphi_2(A_1)$	$\varphi_2(A_2)$	$\ldots$	$\varphi_2(A_n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
w_m	φ_m	$\varphi_m(A_1)$	$\varphi_m(A_2)$	$\ldots$	$\varphi_m(A_n)$

Таким образом, задача выбора оптимального варианта сводиться к поиску максимума некоторой функции $F\colon X \to \mathbb{R}$, дающей численную оценку каждой альтернативы. Более того, благодаря такой функции возможно ранжировать решения относительно друг друга.

Рассмотрим несложный пример в данной постановке задачи. Пусть лицу, принимающему решение, необходимо выбрать модель мобильного телефона. Дано 3 модели с разными характеристиками, заданными 4 параметрами, среди них нужно найти наиболее предпочтительную. Значения критериев и их веса представлены в таблице 2.

Таблица 2: Пример задачи

Веса критериев	Критерии	Модели A	Модели B	Модели C
0,4	Объем памяти, ГБ	32	64	16
0,2	Стоимость, руб.	15 000	20 000	17 000
0,1	Оценка дизайна	Средняя	Высокая	Низкая
0,3	Разрешение камеры, Мп	16	12	22

В задаче определены как объективные количественные критерии: стоимость, объем памяти и разрешение камеры, так и субъективная качественная характеристика — отношение к дизайну телефона. Следовательно, эти неоднородные показатели требуется нормализовать для последующей возможности агрегирования и сопоставления.

Качественные оценки необходимо предварительно сделать количественными. Одним из методов приведения шкалы оценивания к количественной будет приравнивание максимальной (минимальной) оценки к 1 (0). Все промежуточные значения разделяются равными интервалами внутри этого отрезка. Таким образом, «высокой» оценке соответствует значение 1, «средней» — 0,5 и «низкой» — 0.

Для поиска решения этой задачи рассмотрим метод взвешенной суммы критериев и метод TOPSIS.

1.1 Метод взвешенной суммы критериев

Метод взвешенной суммы является одним из наиболее простых способов агрегирования критериев и их важности для ранжирования решений. В то же время, МВСК обладает рядом недостатков среди которых: отсутствие содержательного смысла значения, полученного МВСК, неизменность соотношения важности критериев с их значениями и т. д. [2]

Для решения поставленной задачи, необходимо определить алгоритм действий, позволяющий сопоставить каждой альтернативе действительное число. На основе такой численной оценки можно будет выбрать предпочтительный вариант. Метод взвешенной суммы может предполагать следующий алгоритм:

1. Если шкалы критериев неоднородны — нормировать оценки критериев по одной из формул:

$$\hat{\varphi}_i(\varphi_i) = \frac{\varphi_i}{|\varphi|_{max}}, \tag{1}$$

$$\hat{\varphi}_i(\varphi_i) = \frac{\varphi_i - \varphi_{min}}{\varphi_{max} - \varphi_{min}} \tag{2}$$

2. Вычислить взвешенную сумму критериев каждой альтернативы. Критерии с негативной шкалой предпочтения предварительно умножаются на (-1):

$$F(\hat{\varphi}(A_i)|w) = \sum_{k=1}^{m} w_k \cdot \hat{\varphi}_k(A_i), \quad i = \overline{1,n}$$

3. Проранжировать альтернативы. Согласно МВСК вариант с наибольшим значением F является наилучшим.

Согласно алгоритму, приведенному выше, можно получить матрицы критериев, представленные в таблице 3. Значения стоимости были взяты со знаком $(-)$, поскольку они имеют негативную шкалу предпочтения. В первом способе нормирования использовалась формула (1) из второго пункта алгоритма МВСК, во втором — формула (2).

Можно заметить, что в обоих случаях рекомендуемые альтернативы не совпадают. В то время как в первом случае оптимальным выбором является модель B, во втором случае наилучшим вариантом становится модель A. Следовательно МВСК чувствителен к нормированию показателей и в данном случае не может дать однозначного ответа для решения задачи.

Поскольку вариант C в обоих случаях является наименее предпочтительным, можно его исключить. Еще раз применив алгоритм метода взвешенной суммы критериев, можно получить ответ. В таблице 4

Таблица 3: Решение примера методом взвешенной суммы критериев

		Нормирование сп. (1)			Нормирование сп. (2)		
w_i	$\hat{\varphi}_i$	A	B	C	A	B	C
0,4	$\hat{\varphi}_1$	$0{,}5$	1	$0{,}25$	$1/3$	1	0
0,2	$\hat{\varphi}_2$	$-0{,}75$	-1	$-0{,}85$	0	-1	$-0{,}4$
0,1	$\hat{\varphi}_3$	$0{,}5$	1	0	$0{,}5$	1	0
0,3	$\hat{\varphi}_4$	$8/11$	$6/11$	1	$0{,}4$	0	1
Взвешенная сумма		$0{,}318$	$0{,}464$	$0{,}23$	$0{,}303$	$0{,}3$	$0{,}22$
Ранг альтернативы		2	1	3	1	2	3

сравнены результаты расчетов. В первом случае оптимальной является модель B, тогда как во втором — модели A и B равны. Опираясь на эти оценки, можно заключить, что модель B — наилучшая.

Таблица 4: Исключение наихудшей альтернативы

		Нормирование сп. (1)		Нормирование сп. (2)	
w_i	$\hat{\varphi}_i$	A	B	A	B
0,4	$\hat{\varphi}_1$	$0{,}5$	1	0	1
0,2	$\hat{\varphi}_2$	$-0{,}75$	-1	0	-1
0,1	$\hat{\varphi}_3$	$0{,}5$	1	0	1
0,3	$\hat{\varphi}_4$	1	$0{,}75$	1	0
Взвешенная сумма		$0{,}4$	$0{,}525$	$0{,}3$	$0{,}3$
Ранг альтернативы		2	1	1	1

Тем не менее, при исключении или добавлении низкоранговых вариантов оптимальное решение задачи может измениться — нарушается аксиома независимости от посторонних альтернатив [2, с. 46 — 47]. Это также является одним из недостатков МВСК.

1.2 Метод TOPSIS

Метод TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) базируется на принципе выбора такой альтернативы, которая будет иметь наименьшее расстояние от положительного идеального решения, а также находиться дальше всех остальных вариантов от отрицательного идеального решения [3].

Стандартный алгоритм метода TOPSIS состоит в следующем [6, с. 2]:

1. Построить нормализованную матрицу, где ее элементами будут

$$\hat{\varphi}_i(A_j) = \frac{\varphi_i(A_j)}{\sqrt{\sum_{k=1}^{n} \varphi_i^2(A_k)}}, \quad i = \overline{1,m}, j = \overline{1,n}$$

2. Домножить элементы нормализованной матрицы на соответствующие веса:

$$v_{ij} = w_i \cdot \hat{\varphi}_i(A_j), \quad i = \overline{1,m}, j = \overline{1,n}$$

3. Найти идеальное решение ($\mathcal{A}^+$) и анти-идеальное решение ($\mathcal{A}^-$):

$$\begin{aligned} \mathcal{A}^- &= \{\langle \max(v_{ij} \mid j = \overline{1,n}) \mid i \in I_- \rangle, \\ &\quad \langle \min(v_{ij} \mid j = \overline{1,n}) \mid i \in I_+ \rangle\} \equiv \{v_{-i} \mid i = \overline{1,m}\}, \\ \mathcal{A}^+ &= \{\langle \min(v_{ij} \mid j = \overline{1,n}) \mid i \in I_- \rangle, \\ &\quad \langle \max(v_{ij} \mid j = \overline{1,n}) \mid i \in I_+ \rangle\} \equiv \{v_{+i} \mid i = \overline{1,m}\}, \text{ где} \end{aligned}$$

   $I_+ = \{i = \overline{1,m} \mid i\}$ обозначают критерии имеющие положительную шкалу предпочтений, а $I_- = \{i = \overline{1,m} \mid i\}$ — отрицательную шкалу.

4. Рассчитать Евклидово расстояние между альтернативой j и анти-идеальным решением $\mathcal{A}^-$ и расстояние между альтернативой j и идеальным решением $\mathcal{A}^+$:

$$d_{j-} = \sqrt{\sum_{i=1}^{m} (v_{ij} - v_{-i})^2}, \quad j = \overline{1,n}$$

$$d_{j+} = \sqrt{\sum_{i=1}^{m} (v_{ij} - v_{+i})^2}, \quad j = \overline{1,n}$$

5. Определить относительное сходство каждой альтернативы с идеаль-

ным решением:

$$S_{j-} = d_{j-}/(d_{j-} + d_{j+}), \quad 0 \leqslant S_{j-} \leqslant 1, \quad j = \overline{1,n}.$$

Чем ближе значение S_{j-} к 1, тем лучше альтернатива. В соответствии с этим проранжировать варианты решения.

Руководствуясь алгоритмом, была получена таблица 5. Исходя из расчетов, наиболее предпочтительным вариантом является модель B.

Таблица 5: Решение примера с помощью метода TOPSIS

	Альтернативы				
Норм. взвеш. критерии	A	B	C	v_{+i}	v_{-i}
$\hat{\varphi}_1 \cdot w_1$	0,175	0,349	0,087	0,349	0,087
$\hat{\varphi}_2 \cdot w_2$	0,099	0,132	0,112	0,099	0,132
$\hat{\varphi}_3 \cdot w_3$	0,045	0,089	0	0,089	0
$\hat{\varphi}_4 \cdot w_4$	0,161	0,121	0,222	0,222	0,121
d_{j+}	0,190	0,106	0,277		
d_{j-}	0,111	0,277	0,103		
S_{j-}	0,369	0,723	0,271		
Ранг альтернативы	2	1	3		

1.3 Интервальный метод TOPSIS

Поскольку в данной работе будет рассмотрено решение задачи с интервальными данными, необходимо изучить соответствующие интервальные методы. В качестве основного инструмента для работы с интервальными значениями критериев будут использоваться вариации метода TOPSIS с различными метриками и масштабированием.

В целом, основные этапы метода остаются прежними. Только вместо чисел значения критериев альтернативы характеризуются интервалами: $\varphi_i(A_j) \equiv \left[\varphi_i(A_j^L); \varphi_i(A_j^U)\right]$. Рассмотрим следующий алгоритм:

1. Нормировать матрицу решений одним из двух способов:

(a) Евклидова норма:

$$\hat{\varphi}_i(A_j^L) = \frac{\varphi_i(A_j^L)}{\sqrt{\sum_{k=1}^{n}\left(\varphi_i^2(A_k^L) + \varphi_i^2(A_k^U)\right)}}, \quad i = \overline{1,m}, j = \overline{1,n}$$
$$\hat{\varphi}_i(A_j^U) = \frac{\varphi_i(A_j^U)}{\sqrt{\sum_{k=1}^{n}\left(\varphi_i^2(A_k^L) + \varphi_i^2(A_k^U)\right)}}, \quad i = \overline{1,m}, j = \overline{1,n} \tag{3}$$

(b) Нормирование минимумом и максимумом:

$$\hat{\varphi}_i(A_j^L) = \frac{\varphi_i(A_j^L) - \min\limits_j(\varphi_i(A_j^L))}{\max\limits_j(\varphi_i(A_j^U)) - \min\limits_j(\varphi_i(A_j^L))}, \quad i = \overline{1,m}, j = \overline{1,n}$$
$$\hat{\varphi}_i(A_j^U) = \frac{\varphi_i(A_j^U) - \min\limits_j(\varphi_i(A_j^L))}{\max\limits_j(\varphi_i(A_j^U)) - \min\limits_j(\varphi_i(A_j^L))}, \quad i = \overline{1,m}, j = \overline{1,n} \tag{4}$$

2. Для случая с интервальными весами $w_i \equiv \left[w_i^L; w_i^U\right]$ перемножить по правилам умножения интервалов:

$$v_{ij}^L = w_i^L \cdot \hat{\varphi}_i(A_j^L), \quad i = \overline{1,m}, j = \overline{1,n}$$
$$v_{ij}^U = w_i^U \cdot \hat{\varphi}_i(A_j^U), \quad i = \overline{1,m}, j = \overline{1,n}$$

3. Найти идеальное решение ($\mathcal{A}^+$) и анти-идеальное решение ($\mathcal{A}^-$):

$$\mathcal{A}^- = \{\langle \max(v_{ij}^U \mid j = \overline{1,n}) \mid i \in I_-\rangle,$$
$$\langle \min(v_{ij}^L \mid j = \overline{1,n}) \mid i \in I_+\rangle\} \equiv \{v_{-i} \mid i = \overline{1,m}\},$$
$$\mathcal{A}^+ = \{\langle \min(v_{ij}^L \mid j = \overline{1,n}) \mid i \in I_-\rangle,$$
$$\langle \max(v_{ij}^U \mid j = \overline{1,n}) \mid i \in I_+\rangle\} \equiv \{v_{+i} \mid i = \overline{1,m}\}, \text{ где}$$

$I_+ = \{i = \overline{1,m} \mid i\}$ обозначают критерии имеющие положительную шкалу предпочтений, а $I_- = \{i = \overline{1,m} \mid i\}$ — отрицательную шкалу.

4. В качестве расстояний между альтернативой j и анти-идеальным $\mathcal{A}^-$

и идеальным $\mathcal{A}^+$ решениями рассмотрим следующие [5]:

(a) Евклидова метрика:

$$d_{j-} = \sqrt{\frac{1}{2}\sum_{i=1}^{m}\left((v_{ij}^L - v_{-i})^2 + (v_{ij}^U - v_{-i})^2\right)}, \quad j = \overline{1,n}$$
$$d_{j+} = \sqrt{\frac{1}{2}\sum_{i=1}^{m}\left((v_{ij}^L - v_{+i})^2 + (v_{ij}^U - v_{+i})^2\right)}, \quad j = \overline{1,n} \qquad (5)$$

(b) Расстояние, основанное на метрике Хаусдорфа:

$$d_{j-} = \frac{1}{m}\sum_{i=1}^{m}\max\left(|v_{ij}^L - v_{-i}|; |v_{ij}^U - v_{-i}|\right), \quad j = \overline{1,n}$$
$$d_{j+} = \frac{1}{m}\sum_{i=1}^{m}\max\left(|v_{ij}^L - v_{+i}|; |v_{ij}^U - v_{+i}|\right), \quad j = \overline{1,n} \qquad (6)$$

5. Сходство j-ой альтернативы с наилучшим решением:

$$S_{j-} = d_{j-}/(d_{j-} + d_{j+}), \quad 0 \leqslant S_{j-} \leqslant 1, \quad j = \overline{1,n}.$$

2 Решение задачи на реальных данных

2.1 Задача с четкими данными

Рассмотрим задачу выбора сдающегося в аренду помещения площадью $S \in [90; 100]$ кв. м. с целью открытия магазина «шаговой доступности» в Тверском районе города Москвы.

В качестве матрицы решений будут использованы данные сайта объявлений об аренде (продаже) помещений mestomer.ru. Ниже представлена соответствующая таблица (6) найденных объявлений.

В качестве характеристик альтернатив выбраны такие критерии как площадь помещения, арендная плата за месяц, проходимость человек в сутки, количество квартир в радиусе 500 метров, количество конкурентов (торговых точек раздела супермаркеты) в радиусе 500 метров. Стрелками вверх (вниз) помечены критерии с положительной (отрицательной) шкалой предпочтения.

Таблица 6: Матрица решений

	Площадь, м^2	Стоимость аренды, руб./мес.	Проходимость, чел./сут.	Количество квартир поблизости	Количество конкурентов
	$w_1 = 0{,}061$	$w_2 = 0{,}108$	$w_3 = 0{,}382$	$w_4 = 0{,}123$	$w_5 = 0{,}326$
	$\varphi_1 \uparrow$	$\varphi_2 \downarrow$	$\varphi_3 \uparrow$	$\varphi_4 \uparrow$	$\varphi_5 \downarrow$
A_1	90.0	315000.0	13044	4317	9
A_2	95.0	349988.0	21445	5021	20
A_3	90.0	329900.0	21445	4969	20
A_4	91.3	530000.0	13952	5248	18
A_5	96.6	920000.0	7324	2403	7
A_6	100.0	400000.0	10860	2567	10
A_7	91.0	600000.0	19002	208	9
A_8	95.0	500000.0	19002	208	9
A_9	90.0	400000.0	16510	5378	15
A_{10}	98.4	300000.0	4592	4537	17
A_{11}	97.0	999000.0	16033	4848	14
A_{12}	98.4	299923.2	1208	4651	20
A_{13}	95.0	575000.0	3854	2505	10

Источник — сайт mestomer.ru [24 февраля 2020]

Можно видеть, что различных вариантов мест для размещения магазина $n = 13$ и критериев $m = 5$ уже достаточно велико для подсчета вручную. Поэтому на языке программирования Python автором были реализованы методы взвешенной суммы критериев и TOPSIS, они представлены в листингах 1 и 2 приложения.

В качестве весов критериев были взяты значения из статьи [4, с. 1410]. Несмотря на то, что в публикации веса определяются для размещения магазина в Турции, а критерии по смыслу и содержанию могут отличаться от выбора критериев в текущей задаче, в качестве примера данные значения хорошо проиллюстрируют потребности лиц, заинтересованных в выборе места для торговой точки, тем самым придавая практическую значимость вычислениям. Стоит также отметить, что определение весов критериев является отдельной задачей и в рассматриваемой публикации использовался метод анализа иерархий для вычисления значений весов. Таким образом, $w_1 = 0{,}061$, $w_2 = 0{,}108$, $w_3 = 0{,}382$, $w_4 = 0{,}123$, $w_5 = 0{,}326$.

Таблица 7: Решение задачи

	TOPSIS	d_{j-}	d_{j+}	МВСК, норм. (1)	МВСК, норм. (2)
A_1	0.658; № 6	0.1216	0.0632	0.2052; № 1	0.2687; № 2
A_2	0.6591; № 5	0.1572	0.0813	0.1909; № 3	0.1933; № 6
A_3	0.6593; № 4	0.1573	0.0813	0.1889; № 4	0.1646; № 7
A_4	0.545; № 9	0.1062	0.0887	0.0735; № 9	0.057; № 10
A_5	0.4594; № 10	0.0946	0.1113	0.0308; № 10	0.1121; № 9
A_6	0.5501; № 8	0.1016	0.0831	0.1069; № 8	0.2086; № 4
A_7	0.7415; № 2	0.1484	0.0517	0.1872; № 5	0.2455; № 3
A_8	0.7482; № 1	0.1493	0.0502	0.2004; № 2	0.2853; № 1
A_9	0.6746; № 3	0.1282	0.0618	0.1842; № 6	0.1958; № 5
A_{10}	0.3081; № 12	0.0615	0.138	−0.0639; № 12	−0.0327; № 12
A_{11}	0.6323; № 7	0.121	0.0704	0.1194; № 7	0.1494; № 8
A_{12}	0.2419; № 13	0.0538	0.1685	−0.1705; № 13	−0.1691; № 13
A_{13}	0.3518; № 11	0.072	0.1327	−0.0413; № 11	0.0174; № 11

Исходя из данных таблицы 6, были рассчитаны значения методами TOPSIS и взвешенной суммы (с разными методами нормализации). Результаты обозначены в таблице 7, где наряду со значением указан ранг каждой альтернативы. Для метода TOPSIS также отражены значения расстояний альтернатив до лучшего и худшего решений.

Можно заметить, что торговая точка под номером A_1 является лучшей с позиции МВСК с нормированием по максимальному значению критерия. Тогда как МВСК с нормированием по минимальному и максимальному значениям поставил ее на 2 место, считая альтернативу A_8 оптимальным местом размещения магазина. Более того, A_8 — лучший вариант по методу TOPSIS.

Если взглянуть на изначальные данные, решение A_1 можно назвать весьма уместным и сбалансированным. По наиболее важному критерию — проходимости альтернатива принимает значение немного выше среднего, по количеству конкурентов A_1 уже заметно ниже среднего значения. Более того, данная точка для размещения «магазина у дома» характеризуется одной из самых низких стоимостей, тогда как количество квартир поблизости тоже выше среднего по выборке. По наименее важному критерию — площади, альтернатива принимает наименьшее значение, что не мешает ей быть оптимумом по двум методам.

Рассмотрев альтернативу A_8, которую МВСК с нормированием по минимальному и максимальному значению и TOPSIS поставили на 1 место, в то время как МВСК с нормированием по максимуму расположил данное решение на 2 месте, можно прийти к выводу, что такое решение уже нельзя назвать настолько же сбалансированным как предыдущее. Значение проходимости увеличилось, что является приоритетным критерием, количество конкурентов такое же, как и в альтернативе A_1, в то же время, заметно выросла стоимость аренды, слегка увеличилась площадь, а количество квартир поблизости принимает свой минимум.

Эти результаты наглядно показывают разницу алгоритмов в том, как недостаток в одном критерии, компенсируется избытком в другом. Данный фактор должен приниматься во внимание лицом, принимающим решение, ведь простой расчет показателя не является достаточным условием для окончательного выбора места для размещения магазина, важна также интерпретация того, что стоит за конкретными числами. Как было показано, «компромисс» между критериями ключевым образом сказывается на итоговых показателях.

2.2 Интервальный вариант задачи

Часто в практических задачах реальные значения критериев не могут быть точно известны в силу зависимости этих переменных от времени, сезонности (например, проходимость), субъективности или вовсе невозможности как-либо надежно спрогнозировать явления. Таким образом, возникает необходимость в решении задач с нечеткими входными условиями для поддержки управленческих решений.

Данные для интервального варианта задачи были получены следующим образом: столбец с данными о площади был оставлен без изменений, к остальным значениям было добавлено и вычтено по 10%, чтобы получить верхнюю и нижнюю границы интервалов, результаты округлены. Для получения весов применялся тот же принцип, но с 5% отклонением. Матрица решений и веса представлены в таблице 8.

Таблица 8: Матрица решений

	Площадь, м^2	Стоимость аренды, руб./мес.	Проходимость, чел./сут.	Количество квартир поблизости	Количество конкурентов
w_i	$[0.058; 0.064]$	$[0.1026; 0.1134]$	$[0.3629; 0.4011]$	$[0.1169; 0.1292]$	$[0.3097; 0.3423]$
φ_i	$\varphi_1 \uparrow$	$\varphi_2 \downarrow$	$\varphi_3 \uparrow$	$\varphi_4 \uparrow$	$\varphi_5 \downarrow$
A_1	90.0	$[283500; 346500]$	$[11740; 14348]$	$[3885; 4749]$	$[8; 10]$
A_2	95.0	$[314989; 384987]$	$[19300; 23590]$	$[4519; 5523]$	$[18; 22]$
A_3	90.0	$[296910; 362890]$	$[19300; 23590]$	$[4472; 5466]$	$[18; 22]$
A_4	91.3	$[477000; 583000]$	$[12557; 15347]$	$[4723; 5773]$	$[16; 20]$
A_5	96.6	$[828000; 1012000]$	$[6592; 8056]$	$[2163; 2643]$	$[6; 8]$
A_6	100.0	$[360000; 440000]$	$[9774; 11946]$	$[2310; 2824]$	$[9; 11]$
A_7	91.0	$[540000; 660000]$	$[17102; 20902]$	$[187; 229]$	$[8; 10]$
A_8	95.0	$[450000; 550000]$	$[17102; 20902]$	$[187; 229]$	$[8; 10]$
A_9	90.0	$[360000; 440000]$	$[14859; 18161]$	$[4840; 5916]$	$[14; 16]$
A_{10}	98.4	$[270000; 330000]$	$[4133; 5051]$	$[4083; 4991]$	$[15; 19]$
A_{11}	97.0	$[899100; 1098900]$	$[14430; 17636]$	$[4363; 5333]$	$[13; 15]$
A_{12}	98.4	$[269931; 329916]$	$[1087; 1329]$	$[4186; 5116]$	$[18; 22]$
A_{13}	95.0	$[517500; 632500]$	$[3469; 4239]$	$[2254; 2756]$	$[9; 11]$

Источник — сайт mestomer.ru [24 февраля 2020]

К исходным данным было применено две вариации алгоритма TOPSIS.

В одной использовалась Евклидова норма и Евклидова метрика, тогда как в другом — нормализация проводилась минимумом и максимумом, а расстояние вычислялось с помощью метрики Хаусдорфа. К числам из столбца с площадью помещения применялись те же методы, что и к остальным интервальным данным, считая, что $\varphi_1(A_j^L) = \varphi_1(A_j^U) = \varphi_1(A_j)$.

В ходе расчетов была получена таблица 9:

Таблица 9: Решение интервальной задачи

	Интервальный TOPSIS метрика: Евклидова (5) норм.: Евклидова (3)	d_{j-}	d_{j+}	Интервальный TOPSIS метрика: Хаусдорфа (6) норм.: Мин. - Макс. (4)	d_{j-}	d_{j+}
A_1	0.6137; № 4	0.1179	0.0742	0.6347; № 2	0.1515	0.0872
A_2	0.612; № 6	0.1366	0.0866	0.5881; № 5	0.1545	0.1083
A_3	0.6123; № 5	0.1367	0.0866	0.5662; № 7	0.1483	0.1136
A_4	0.5006; № 9	0.0977	0.0975	0.4909; № 10	0.1246	0.1293
A_5	0.4592; № 10	0.0986	0.1162	0.5212; № 9	0.1217	0.1118
A_6	0.527; № 8	0.1026	0.0921	0.5967; № 4	0.1417	0.0957
A_7	0.6887; № 2	0.1369	0.0619	0.6179; № 3	0.1495	0.0924
A_8	0.6947; № 1	0.1379	0.0606	0.6491; № 1	0.1568	0.0848
A_9	0.6154; № 3	0.1176	0.0735	0.5846; № 6	0.1472	0.1046
A_{10}	0.3239; № 12	0.068	0.1419	0.4235; № 12	0.1022	0.1391
A_{11}	0.5787; № 7	0.111	0.0808	0.5566; № 8	0.1419	0.113
A_{12}	0.2552; № 13	0.0581	0.1696	0.3261; № 13	0.0788	0.1629
A_{13}	0.3763; № 11	0.0822	0.1363	0.4485; № 11	0.1036	0.1274

Как можно заметить, результаты обоих вариантов метода схожи. Ранг большинства альтернатив либо совпадает, либо отличается не больше, чем на 4 позиции. Обе вариации сошлись в определении наилучшей альтернативы. Как и в случае с четким TOPSIS, оптимальным местом размещения «магазина у дома» оказался вариант A_8.

Таким образом, решение задачи можно найти и с интервальными значениями весов и критериев. Вариативность в выборе метрики и способа нормирования делает метод TOPSIS более гибким инструментом для лиц, принимающих решение. Тем не менее, конкретная реализация метода зависит от постановки задачи.

3 Заключение

Задача выбора профиля «магазина у дома» относится к классу задач многокритериальной оптимизации. Для ее решения необходимо использовать различные подходы и методы для агрегирования критериальных оценок и весов критериев. В результате, можно найти оптимальное решение и, более того, проранжировать альтернативы в соответствии с рассчитанными показателями.

Методы взвешенной суммы и TOPSIS позволяют на основе имеющейся информации о значениях признаков и их степенях важности решать задачи многокритериального принятия решения. Разнообразные модификации методов дают возможность лучше подстроить алгоритмы под конкретные постановки задач.

Так, методы взвешенной суммы критериев и TOPSIS допускают использование разных способов нормализации. Кроме того, модификации TOPSIS также предоставляют возможность использовать метрики, отличные от Евклидовой, для расчета расстояния между лучшим (худшим) решением и альтернативами.

Вместе с тем, вариации рассматриваемых методов допускают использование нечетких входных данных. Для решения ключевой задачи работы были применены 2 вариации интервального метода TOPSIS.

Рассматриваемые алгоритмы были использованы для поиска наилучшей точки размещения «магазина у дома» площадью от 90 до 100 м^2 в Тверском районе города Москвы. Итоговые расчеты показывают схожие результаты в определении оптимального решения. Тем не менее, при принятии заключительного выбора лицу, принимающему решение, не следует слепо руководствоваться только интегральной оценкой выбранного метода. Необходимо дополнительно проверить, насколько финальное решение соответствует поставленной задаче. Это особенно важно, поскольку место размещения магазина может напрямую сказываться на финансовых успехах компании.

Задача выбора профиля «магазина у дома» не единственная в своем классе. Методы многокритериального принятия решения такие, как МВСК и TOPSIS могут найти применение и в других сферах, среди которых профилирующие, рекомендательные системы, расчет индексов и др. Широта использования методов обусловлена как возможностью модификации алгоритмов, так и применимостью к различным типам данных.

Список литературы

1. *Лотов А. В.*, *Поспелова И. И.* Многокритериальные задачи принятия решений. — Изд. отдел фак. ВМиК МГУ им. МВ Ломоносова М., 2008.
2. *Подиновский В. В.*, *Потапов М. А.* Метод взвешенной суммы критериев в анализе многокритериальных решений: pro et contra // Бизнес-информатика. — 2013. — 3 (25).
3. *Пятецкий В. Е.*, *Литвяк В. С.*, *Литвин И. З.* Методы принятия оптимальных управленческих решений: моделирование принятия решений [Электронный ресурс]: учебное пособие // М.: МИСИС. — 2014. — Т. 133. — С. 2015—13.
4. *Erbıyık 2.*, *Özcan S.*, *Karaboğa K.* Retail Store Location Selection Problem with Multiple Analytical Hierarchy Process of Decision Making an Application in Turkey // Procedia - Social and Behavioral Sciences. — 2012. — Т. 58. — С. 1405—1414. — ISSN 1877-0428. — DOI: `https://doi.org/10.1016/j.sbspro.2012.09.1125`. — URL: `http://www.sciencedirect.com/science/article/pii/S1877042812045879` ; 8th International Strategic Management Conference.
5. *Grzegorzewski 2.* Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the Hausdorff metric // Fuzzy Sets and Systems. — 2004. — Т. 148, № 2. — С. 319—328. — ISSN 0165-0114. — DOI: `https://doi.org/10.1016/j.fss.2003.08.005`. — URL: `http://www.sciencedirect.com/science/article/pii/S0165011403003543`.
6. *Lima Silva 2. F. de*, *Almeida Filho A. T. de*. Sorting with TOPSIS through boundary and characteristic profiles // Computers & Industrial Engineering. — 2020. — Т. 141. — С. 106328. — ISSN 0360-8352. — DOI: `https://doi.org/10.1016/j.cie.2020.106328`. — URL: `http://www.sciencedirect.com/science/article/pii/S0360835220300620`.

Приложение

Листинг 1.

Метод взвешенной суммы критериев, реализованный на языке Python

```python
import numpy as np
import pandas as pd

def weighted_sum(
        data: pd.DataFrame,
        isNeg: list = None,
        weights: list = None,
        scale: str = 'min_max') -> pd.Series:
    """ Returns pd.Series of values computed by the weighted sum method;
    data: Decision matrix;
    isNeg: List of str or int - marks the criteria with decreasing scale of preferences;
    weights: Weights of the criteria. Equal weights are by default;
    scale: Normalization method. 'min_max' by default: (x - min(x)) / (max(x) - min(x));
    return: Resulting pd.Series; """
    # Normalization
    if scale == 'min_max':
        R = data.max() - data.min(); R[R == 0] = 1  # avoid div by zero
        data = (data - data.min()) / R
    elif scale == 'max':
        data_max = np.abs(data).max(); data_max[data_max == 0] = 1  # avoid div by zero
        data = data / data_max
    elif scale == 'normal_distr':
        mean = data.mean()
        std = data.std(); std[std == 0] = 1  # avoid div by zero
        data = (data - mean) / std
    # Multiplication by weights
    if weights is not None:
        data = weights * data
    else:
        data = (np.ones(len(data.columns)) * (1 / len(data.columns))) * data
    # Negative preference scale criteria make negative
    if isNeg is not None:
        data[isNeg] = data[isNeg] * (-1)
    # Summation and result by descending of values
    return data.sum(axis=1).sort_values(ascending=False)
```

Листинг 2.

Метод TOPSIS, реализованный на языке Python

```python
import numpy as np
import pandas as pd


def TOPSIS(
        data: pd.DataFrame,
        isNeg: list = None,
        weights: list = None) -> pd.Series:
    """ Returns pd.Series of values computed according to TOPSIS method;
    data: Decision matrix;
    isNeg: List of str or int - marks the criteria with decreasing scale of preferences;
    weights: Weights of the criteria. Equal weights are by default;
    return: Resulting pd.Series; """
    # Normalization
    norma = np.linalg.norm(data, axis=0)
    norma[norma == 0] = 1  # if there is a zero vector
    data = data / norma
    # Multiplication by weights
    if weights is not None:
        data = weights * data
    else:
        data = (np.ones(len(data.columns)) * (1 / len(data.columns))) * data
    # Ideal and anti-ideal solutions
    isPos = data.columns.drop(isNeg)
    A_minus = np.min(data.loc[:, isPos], axis=0).append(
        np.max(data.loc[:, isNeg], axis=0))[data.columns]
    A_plus = np.max(data.loc[:, isPos], axis=0).append(
        np.min(data.loc[:, isNeg], axis=0))[data.columns]
    # Distance between j and ideal\anti-ideal solution
    d_minus = np.sqrt(((data - A_minus) ** 2).sum(axis=1))
    d_plus = np.sqrt(((data - A_plus) ** 2).sum(axis=1))
    # Relative distance to ideal solution of all alternatives
    R = d_plus + d_minus
    R[R == 0] = 1  # if sum of d_plus and d_minus gives 0
    data = d_minus / R
    # Result sorted by descending of values
    return data.sort_values(ascending=False)
```

Листинг 3.

Интервальный метод TOPSIS, реализованный на языке Python

```
import numpy as np
import pandas as pd


def intervalTOPSIS(
        lowerBounds: pd.DataFrame,
        upperBounds: pd.DataFrame,
        isNeg: list = None,
        weights_lower: list = None,
        weights_upper: list = None,
        norm: str = 'min_max',
        metric: str = 'haus') -> pd.Series:
    """ Returns pd.Series of values computed according to interval TOPSIS
    method;
    lowerBounds: Decision matrix of lower bounds;
    upperBounds: Decision matrix of upper bounds;
    isNeg: List of str or int - marks the criteria with decreasing scale
   of preferences;
    weights_lower (weights_upper): Weights of lower (upper) bounds of the
    criteria. Equal weights are by default;
    return: Resulting pd.Series; """
    # Normalization
    if norm == 'min_max':
        mins = lowerBounds.agg('min')
        h = upperBounds.agg('max') - mins; h[h == 0] = 1
        lowerBounds = (lowerBounds - mins) / h
        upperBounds = (upperBounds - mins) / h
    elif norm == 'euclid':
        norm = np.sqrt((lowerBounds ** 2 + upperBounds ** 2).sum(axis =
   0))
        norm[norm == 0] = 1
        lowerBounds = lowerBounds / norm
        upperBounds = upperBounds / norm
    # Multiplication by weights
    if weights is not None:
        lowerBounds = weights_lower * lowerBounds
        upperBounds = weights_upper * upperBounds
    else:
        lowerBounds = (np.ones(len(lowerBounds.columns)) * (1 / len(
   lowerBounds.columns))) * lowerBounds
        upperBounds = (np.ones(len(lowerBounds.columns)) * (1 / len(
   lowerBounds.columns))) * upperBounds
```

```
    # Ideal and anti-ideal solutions
    isPos = lowerBounds.columns.drop(isNeg)
    A_minus = np.min(lowerBounds.loc[:, isPos], axis=0).append(
        np.max(upperBounds.loc[:, isNeg], axis=0))[upperBounds.columns]
    A_plus = np.max(upperBounds.loc[:, isPos], axis=0).append(
        np.min(lowerBounds.loc[:, isNeg], axis=0))[upperBounds.columns]
    # Distance between j and ideal\anti-ideal solution
    if metric == 'haus':
        d_minus = (1 / len(upperBounds.columns)) * (np.abs(lowerBounds -
   A_minus).
                    combine(np.abs(upperBounds - A_minus),
                            lambda x, y: x.
                            combine(y, lambda n, m : (n, m))).
                    applymap(max).sum(axis = 1))
        d_plus = (1 / len(upperBounds.columns)) * (np.abs(lowerBounds -
   A_plus).
                   combine(np.abs(upperBounds - A_plus),
                           lambda x, y: x.
                           combine(y, lambda n, m : (n, m))).
                   applymap(max).sum(axis = 1))
    elif metric == 'euclid':
        d_minus = np.sqrt(0.5 * ((upperBounds - A_minus) ** 2).
                          sum(axis = 1) + ((lowerBounds - A_minus) ** 2).
                          sum(axis = 1))
        d_plus = np.sqrt(0.5 * ((lowerBounds - A_plus) ** 2).
                          sum(axis = 1) + ((upperBounds - A_plus) ** 2).
                          sum(axis = 1))
    # Relative distance to ideal solution of all alternatives
    data = np.nan_to_num(d_minus / d_plus + d_minus, nan = 0)
    # Result sorted by descending of values
    return data.sort_values(ascending=False)
```