Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №4
«Методы и процедуры принятия решений при многих критериях»
Вариант № 3

Выполнил
студент группы 050502:
Крачковский А.В.

Проверил:
Селезнев А.И.

Минск 2023

1. Исходные данные для выполнения

Предлагаются шесть вариантов площадки для строительства нового предприятия. Характеристики площадок следующие.

Площадка	Пл1	Пл2	Пл3	Пл4	Пл5	Пл6
Уровень развития дорожной сети	средняя	плохая	развитая	развитая (немного лучше, чем для Пл3)	средняя	плохая
Энергоснабжение	хорошее	хорошее	плохое	среднее	очень хорошее	среднее
Затраты на подготовку к строительству, млн ден.ед.	3,5	2,5	3	3,5	3	2,0

Важность критериев оценивается двумя экспертами.

По мнению первого эксперта, наиболее важный критерий - затраты на подготовку к строительству, менее важны (и одинаково важны между собой) уровень развития дорожной сети и энергоснабжение.

По мнению второго эксперта, наиболее важный критерий - уровень развития дорожной сети, немного менее важный - затраты на подготовку к строительству, еще немного менее важный - энергоснабжение.

2. Выбор множества Парето

Выбор множества Парето-оптимальных решений (множества Парето) представляет собой отбор перспективных альтернатив, из которых затем отбирается одна (лучшая) альтернатива.

Выберем множества Парето:

Для удобства обозначим что А1 = „Уровень развития дорожной сети“, А2 = „Энергоснабжение“, а А3 = „Затраты на подготовку к строительству, млн ден. ед.“. При знаке „>“ будем подразумевать „лучше“, при „=“ равны и при „<“ хуже.

2.1 Сравним Пл1 и Пл2:

А1 Пл1 > Пл2;

А2 Пл1 = Пл2;

А3 Пл1 < Пл2

Таким образом, ни одну из альтернатив исключить нельзя, так как по некоторым критериям лучше одна, а по другим – другая.

2.2 Сравним Пл1 и Пл3.

А1 Пл1 < Пл3;

А2 Пл1 > Пл3;

А3 Пл1 < Пл3;

Ни одна из альтернатив не исключается.

2.3 Сравним Пл1 и Пл4.

А1 Пл1 < Пл4;

А2 Пл1 > Пл4;

А3 Пл1 = Пл4;

Ни одна из альтернатив не исключается.

2.4 Сравним Пл1 и Пл5.

А1 Пл1 = Пл5;

А2 Пл1 < Пл5;

А3 Пл1 < Пл5;

Здесь мы можем заметить что Пл1 ничем не превосходит Пл5, напротив, является более худшей альтернативой, из чего следует что мы можем исключить Пл1 из дальнейшего сравнения.

2.5 Остальные

Аналогично сравниваются остальные альтернативы. Но дальнейшие сравнения не приведут к исключению какой-либо альтернативы.

Таким образом, во множество Парето войдут альтернативы Пл2, Пл3, Пл4, Пл5 и Пл6. Именно из них будет затем выбираться лучшая альтернатива.

3. Второй способ анализа альтернатив

3.1 Метод предпочтений

Метод основан на ранжировании альтернатив, выполняемом группой экспертов. Каждый из экспертов (независимо от других) выполняет ранжирование альтернатив, т.е. указывает, какая из альтернатив, по его мнению, является лучшей, какая - следующей за ней, и т.д.

1 Каждому эксперту предлагается выполнить ранжирование альтернатив по предпочтению. В данном примере каждый эксперт присваивает номер 1 фактору, который (по его мнению) оказывает наибольшее влияние на рост производительности труда; 2 - следующему по важности фактору, и т.д. Оценки, указанные экспертами, сводятся в таблицу (матрицу) размером MxN, где M - количество экспертов, N- количество альтернатив (в данном примере - количество факторов роста производительности труда). Обозначим эти оценки как X_{ij}, i=1,...,M, j=1,...,N.

Ранжирование альтернатив по предпочтению представлено в таблице 3.1.1.

Таблица 3.1.1 — Матрица экспертных оценок для метода предпочтений

	А1	А2	А3
Э1	2	2	1
Э2	1	3	2

2 Затем производится преобразование матрицы оценок по формуле:

$B_{ij} = N - X_{ij}$, i=1,...,M, j=1,...,N.

Это означает, что каждая экспертная оценка вычитается из количества альтернатив.

Для данного примера получена матрица, приведенная в таблице 3.1.2.

Таблица 3.1.2 — Преобразованная матрица экспертных оценок для метода предпочтений

	А1	А2	А3
Э1	1	1	2
Э2	2	0	1

После выполнения вышеперечисленных шагов, можно загрузить данные в написанную программу (код программы можно увидеть в **Приложении А**, выполнение программы в **Приложении Б**, так же при работе программы создается конфигурационный файл, содержимое которого можно увидеть в **Приложении В**)

В данном примере $V_1 = 0.4286$; $V_2 = 0.1429$; $V_3 = 0.4286$

Чем больше вес, тем более предпочтительной является альтернатива (по мнению экспертов).

Таким образом, самыми предпочтительными альтернативами являются уровень развития дорожной сети и затраты на подготовку к строительству; следующая по важности альтернатива – энергоснабжение.

3.2 Модифицированный алгоритм Кемени-Снелла

Рассматриваемый алгоритм предназначен для ранжирования альтернатив с учетом их оценок по нескольким критериям.

Основное преимущество алгоритма – возможность анализа и выбора альтернатив, оцениваемых по критериям различных видов: числовым, качественным, “да-нет” и т.д. Алгоритм также позволяет учитывать суждения ЛПР о важности критериев.

Алгоритм основан на ранжировании и попарном сравнении альтернатив по каждому критерию.

Составим таблицу после выбора множества Парето (см. таблицу 3.2.1)

Таблица 3.2.1 — Множество Парето

Площадка	**Пл2**	**Пл3**	**Пл4**	**Пл5**	**Пл6**
Уровень развития дорожной сети	плохая	развитая	развитая (немного лучше, чем для Пл3)	средняя	плохая
Энергоснабжение	хорошее	плохое	среднее	очень хорошее	среднее
Затраты на подготовку к строительству, млн ден.ед.	2,5	3	3,5	3	2,0

Выбор альтернативы на основе модифицированного алгоритма Кемени–Снелла реализуется в следующем порядке.

1 С помощью одного из методов экспертных оценок находятся веса критериев, представляющие собой числовые оценки их важности. В данном примере использовался метод приоритетов (см. подраздел 3.1)

2 Выполняется ранжирование альтернатив по каждому из критериев. При этом лучшая альтернатива по данному критерию получает оценку (ранг) 1, следующая за ней – оценку 2, и т.д. Если альтернативы по данному критерию одинаковы, то они получают *одинаковые* оценки. Результаты ранжирования сводятся в матрицу. Для данной задачи матрица ранжирований приведена в таблице 3.2.2.

Таблица 3.2.2 — Матрица ранжирований

	Пл2	Пл3	Пл4	Пл5	Пл6
К1	4	2	1	3	4
К2	2	4	3	1	3
К3	2	3	4	3	1

3 На основе ранжирования альтернатив по каждому из критериев составляется матрица парных сравнений. Всего составляется M таких матриц, где M - количество критериев. Матрицы заполняются по правилам, приведенным в таблице 3.2.3.

Таблица 3.2.3 — Правила заполнения матриц парных сравнений в модифицированном алгоритме Кемени-Снелла

R^i_{jk}	Значение
1	По i-му критерию j-я альтернатива лучше k-й
-1	По i-му критерию j-я альтернатива хуже k-й
0	По i-му критерию j-я и k-я альтернативы одинаковы

Здесь i - номер матрицы (номер критерия).

Для рассматриваемой задачи матрицы парных сравнений по критериям К1-К3 приведены в таблицах 3.2.4 – 3.2.6.

Таблица 3.2.4 — Парные сравнения по критерию К1

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	-1	-1	-1	0
Пл3	1	-	-1	1	1
Пл4	1	1	-	1	1
Пл5	1	-1	-1	-	1
Пл6	0	-1	-1	-1	-

Таблица 3.2.5 — Парные сравнения по критерию К2

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	1	1	-1	1
Пл3	-1	-	-1	-1	-1
Пл4	-1	1	-	-1	0
Пл5	1	1	1	-	1
Пл6	-1	1	0	-1	-

Таблица 3.2.6 — Парные сравнения по критерию К3

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	1	1	1	-1
Пл3	-1	-	1	0	-1
Пл4	-1	-1	-	-1	-1
Пл5	-1	0	1	-	-1
Пл6	1	1	1	1	-

4 Составляется матрица потерь. Размерность матрицы - *N*x*N*, где *N* - количество альтернатив. Элементы матрицы потерь рассчитываются по следующей формуле:

$$R_{jk}=\sum_{i=1}^{M} V_i \cdot |R_{jk}^i - 1|, \qquad j=1,...,N,\ k=1,...,N.$$

Матрица потерь для рассматриваемой задачи приведена в таблице 3.2.7.

Таблица 3.2.7 — Матрица потерь

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	0.8571	0.8571	1.1429	1.2857
Пл3	1.1429	-	1.1429	0.7143	1.1429
Пл4	1.1429	0.8571	-	1.1429	1
Пл5	0.8571	1.2857	0.8571	-	0.8571
Пл6	0.7143	0.8571	1	1.1429	-

Приведем примеры расчета некоторых элементов матрицы потерь.

$$R_{23}=V_1 \cdot \left|R_{23}^1-1\right|+V_2 \cdot \left|R_{23}^2-1\right|+V_3 \cdot \left|R_{23}^3-1\right|=0.429 \cdot |-1-1|+0.143 \cdot |1-1|+0.429 \cdot |1-1|=0.858$$

$$R_{36}=V_1 \cdot \left|R_{36}^1-1\right|+V_2 \cdot \left|R_{36}^2-1\right|+V_3 \cdot \left|R_{36}^3-1\right|=0.429 \cdot |1-1|+0.143 \cdot |-1-1|+0.429 \cdot |-1-1|=1.144$$

Смысл элементов матрицы потерь следующий: чем больше элемент R_{jk}, тем больше отставание *j*-й альтернативы от *k*-й (тем хуже *j*-я альтернатива по сравнению с *k*-й).

5 Выполняется предварительное ранжирование альтернатив. Для этого находятся суммы строк матрицы потерь. Смысл этих сумм следующий: сумма j-й строки представляет собой оценку *отставания* j-й альтернативы от *всех остальных* альтернатив.

Альтернатива, которой соответствует *минимальная* сумма, предварительно считается *лучшей*. Строка и столбец этой альтернативы исключаются из матрицы потерь.

Суммирование строк матрицы потерь и исключение альтернатив выполняются до тех пор, пока не будет исключена вся матрица. Чем раньше исключена альтернатива, тем она лучше.

Выполним предварительное ранжирование для рассматриваемой задачи. Суммы строк матрицы потерь:

$P_2 = 4.1429$ $P_3 = 4.1429$ $P_4 = 4.142$ $P_5 = 3.8571$ $P_6 = 3.7143$

Предварительно лучшей считается альтернатива Пл6. Она исключается из матрицы потерь. Сокращенная матрица потерь приведена в таблице 3.2.8.

Таблица 3.2.8 — Первая сокращенная матрица потерь

	Пл2	Пл3	Пл4	Пл5
Пл2	—	0.8571	0.8571	1.1429
Пл3	1.1429	—	1.1429	0.715
Пл4	1.1429	0.8571	—	1.1429
Пл5	0.8571	1.287	0.8571	—

Суммы строк этой матрицы: $P_2 = 2.8571$; $P_3 = 3$; $P_4 = 3.1429$; $P_5 = 3$. Исключается альтернатива Пл2. Вторая сокращенная матрица потерь приведена в таблице 3.2.9.

Таблица 3.2.9 — Вторая сокращенная матрица потерь

	Пл3	Пл4	Пл5
Пл3	—	1.144	0.715
Пл4	0.8571	—	1.144
Пл5	1.287	0.8571	—

Суммы строк этой матрицы: $P_3 = 1.8571$; $P_4 = 2$; $P_5 = 2.1429$. Исключается альтернатива Пл3. Третья сокращенная матрица потерь приведена в таблице 3.2.10.

Таблица 3.2.10 — Третья сокращенная матрица потерь

	Пл4	Пл5
Пл4	—	1.144
Пл5	0.8571	—

Суммы строк этой матрицы: P_4 = 1.1429; P_5 = 0.8571. Лучшая альтернатива (из двух оставшихся) – Пл5.

Предварительное ранжирование альтернатив: Пл6, Пл2, Пл3, Пл5, Пл4.

6 Выполняется окончательное ранжирование альтернатив. Для этого альтернативы сравниваются попарно, начиная с конца предварительного ранжирования. Если сравниваются j-я и k-я альтернативы (при этом j-я альтернатива в предварительном ранжировании находится выше k-й) и выполняется условие $R_{jk} \leq R_{kj}$ (где R_{jk} и R_{kj} - элементы матрицы потерь), то альтернативы остаются в ранжировании на прежних местах (j-я альтернатива лучше k-й). Если $R_{jk} > R_{kj}$, то альтернативы меняются местами (j-я альтернатива хуже k-й).

Посмотри как происходило ранжирование в программе MatLab.

```
R43 is smaller than R34, so we do not swap them
R24 is smaller than R42, so we do not swap them
R12 is smaller than R21, so we do not swap them
R51 is smaller than R15, so we do not swap them
```

Поскольку программа не знает об исчезновении Пл1 с помощью множества паррето, все индексы уменьшились на один, для нашего случая их нужно увеличить на один.

Таким образом, окончательное ранжирование альтернатив следующее: Пл6, Пл2, Пл3, Пл5, Пл4. Лучший вариант строительства нового предприятия - площадка, обозначенная как Пл6.

3.3 Метод ЭЛЕКТРА

Метод предназначен для решения задач, в которых из имеющегося множества альтернатив требуется выбрать заданное количество лучших альтернатив с учетом их оценок по нескольким критериям, а также важности этих критериев.

Принцип работы метода следующий. Для каждой пары альтернатив (A_j и A_k) выдвигается предположение (гипотеза) о том, что альтернатива A_j лучше, чем A_k. Затем для каждой пары альтернатив находятся два индекса: индекс согласия (величина, подтверждающая предположение о превосходстве A_j над A_k) и индекс несогласия (величина, опровергающая это предположение). На основе анализа этих индексов выбирается одна или несколько лучших альтернатив ("ядро" альтернатив).

Исходные данные берутся из таблицы 3.2.1.

С помощью одного из методов экспертных оценок находятся веса критериев, представляющие собой числовые оценки их важности. В данном примере использовался метод предпочтений (см. подраздел 3.1)

Выбор лучших альтернатив по методу ЭЛЕКТРА реализуется в следующем порядке.

1 Оценки альтернатив приводятся к безразмерному виду. Эта операция может выполняться разными способами, например, так же, как в методике экспресс-анализа альтернатив.

Оценки по качественным критериям выражаются по пятибалльной шкале (“отлично”, “хорошо”, “удовлетворительно”, “плохо”, “очень плохо”), а затем выполняется переход к числовым оценкам с использованием **шкалы Харрингтона**. При этом оценке "отлично" соответствуют числовые оценки от 0,8 до 1; "хорошо" - от 0,63 до 0,8; "удовлетворительно" - от 0,37 до 0,63; "плохо" - от 0,2 до 0,37; "очень плохо" - от 0 до 0,2. Числовая оценка выставляется человеком: экспертом или лицом, принимающим решения (ЛПР). Например, если по некоторому критерию две альтернативы имеют оценку “хорошо”, но одна из них очень хорошая, а другая - немного хуже, то первой из альтернатив (лучшей) можно назначить оценку 0,8, а второй, например — 0,7;

Для оценок, имеющих вид "да-нет" (т.е. выражающих наличие или отсутствие некоторого показателя), обычно используются следующие числовые оценки: "да" - 0,67, "нет" - 0,33 (здесь предполагается, что оценка “да” более желательна, чем ”нет”).

Исходя из информации выше, составим таблицу безразмерных оценок. Так же для удобства переименуем критерии на К1, К2, К3.

Таблица 3.3.2 — Безразмерные оценки альтернатив

	Пл2	Пл3	Пл4	Пл5	Пл6
К1	0.36	0.8	0.9	0.62	0.36
К2	0.7	0.36	0.2	0.79	0.62
К3	0.7	0.62	0.36	0.62	0.9

2 Определяются индексы согласия C_{jk}, j=1,...,N, k=1,...,N (где N - количество альтернатив). Индекс согласия отражает степень согласия с предположением о том, что j-я альтернатива лучше k-й. В рассматриваемой реализации метода ЭЛЕКТРА индексы согласия находятся по формуле

$$C_{jk} = \sum_{i \in K^{+}} V_i, \qquad j=1,\ldots,N,\ k=1,\ldots,N,$$

где V_i - веса критериев;

K^{+} - подмножество критериев, по которым j-я альтернатива не хуже k-й.

Таким образом, индекс согласия C_{jk} находится как сумма весов критериев, по которым j-я альтернатива *не хуже* k-й. Чем больше индекс согласия, тем более выражено превосходство j-й альтернативы над k-й.

Индексы согласия для данной задачи приведены в таблице 3.3.3.

Таблица 3.3.3 — Матрица индексов согласия

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	0.4286	0.4286	0.5714	0.4286
Пл3	0.5714	-	0.428	0.1429	0.5714
Пл4	0.5714	0.5714	-	0.5714	0.5714
Пл5	0.4286	0.4286	0.4286	-	0.4286
Пл6	0.1429	0.4286	0.4286	0.5714	-

3 Определяются индексы несогласия D_{jk}, $j=1,...,N$, $k=1,...,N$. Индекс несогласия отражает степень несогласия с предположением о том, что j-я альтернатива лучше k-й. Индексы D_{jk} находятся по формуле:

$$D_{jk} = \max_{i \in K^{-}} \left(P_{ik} - P_{ij} \right), \qquad j=1,...,N,\ k=1,...,N,$$

где P_{ik}, P_{ij} - безразмерные оценки альтернатив (для данного примера они приведены в таблице 3.3.2);

K^{-} - подмножество критериев, по которым j-я альтернатива не превосходит k-ю.

Таким образом, индекс несогласия D_{jk} находится как максимальная из разностей оценок по критериям, по которым j-я альтернатива *не лучше* k-й. Чем больше индекс несогласия, тем менее выражено превосходство j-й альтернативы над k-й.

Индексы несогласия для данной задачи приведены в таблице 3.3.4.

Таблица 3.3.4 — Матрица индексов несогласия

	Пл2	Пл3	Пл4	Пл5	Пл6
Пл2	-	0.4400	0.5400	0.3500	0.2000

Пл3	0.4200	-	0.1000	0.4300	0.5400
Пл4	0.8400	0.4200	-	0.8500	0.9600
Пл5	0.0800	0.1800	0.2800	-	0.2800
Пл6	0.0800	0.4400	0.5400	0.4300	-

4 Для каждой альтернативы находится предельное значение индекса согласия:

$$C_j = \min_k C_{jk}, \qquad j=1,...,N.$$

Для рассматриваемого примера C_2 = 0.4286; C_3 = 0.1429; C_4 = 0.5714; C_5=0.4286; C_6=0.1429.

5 Для каждой альтернативы находится предельное значение индекса несогласия:

$$D_j = \max_k D_{jk}, \qquad j=1,...,N.$$

Таким образом, предельное значение индекса несогласия для j-й альтернативы находится как *максимальный* элемент j-й строки матрицы индексов несогласия. Эта величина отражает степень несогласия с предположением о превосходстве j-й альтернативы над другими альтернативами.

Для рассматриваемого примера D_2 = 0.54; D_3 = 0.54; D_4= 0.96; D_5.= 0.28 D_6= *0.54*.

6 Выделяются лучшие альтернативы (“ядро” альтернатив), удовлетворяющие условиям:

$C_j > C^*$,

$D_j < D^*$,

где C^*, D^* - пороговые значения индексов согласия и несогласия. Эти величины назначаются в зависимости от того, какое количество альтернатив требуется выбрать. Обычно сначала принимаются пороговые значения C^* = 0.5, D^* = 0.5; затем они изменяются в соответствии с количеством отбираемых альтернатив. Выбираются альтернативы, удовлетворяющие *обоим* условиям. В нашем случае, если выставить такие значения, то мы не сможем найти наилучшую альтернативу, поэтому уменьшим на 0.1 значение C.

Назначим пороговые значения C^* = 0.4, D^* = 0.5. Программа выдает результат 4, что в нашем случаем будет обозначать Пл5.

Приложение А

creatingFile.m

```
function creatingFile()
file = fopen('condition.txt', 'w');
fprintf(file, '# This is the file for your condition to run lab 4\n# All line starting with # will be ignored\n');
fprintf(file, '# This program solve ONLY second way of calculating\n');
fprintf(file, '# NOTE: the set of parreto must be already applied to your task\n');
fprintf(file, '# There will be used preference method\n#\n');
fprintf(file, '# Please write number of experts\n');
fprintf(file, '2\n');
fprintf(file, '# Please write number of alternatives\n');
fprintf(file, '3\n');
fprintf(file, '# Please write preferences of two experts\n');
fprintf(file, '2 2 1\n1 3 2\n');
fprintf(file, '# End of preferences method\n#\n');

fprintf(file, '# Modified Kemeny-Snell algorithm\n');
fprintf(file, '# Please write number of criteria\n');
fprintf(file, '5\n');
fprintf(file, '# Write alternatives with ranking according to each of the criteria\n');
fprintf(file, '4 2 1 3 4\n');
fprintf(file, '2 4 3 1 3\n');
fprintf(file, '2 3 4 3 1\n');
fprintf(file, '# End of modified Kemeny-Snell algorithm\n#\n');

fprintf(file, '# ELECTRA method\n# Please, write dimensionless estimates\n');
fprintf(file, '0.36 0.8 0.9 0.62 0.36\n');
fprintf(file, '0.7 0.36 0.2 0.79 0.62\n');
fprintf(file, '0.7 0.62 0.36 0.62 0.9\n');
fprintf(file, "# Enter C* value\n");
fprintf(file, '0.5\n');
fprintf(file, '# Enter D* value\n');
fprintf(file, '0.5\n');
fprintf(file, '# End of ELECTRA method\n# End of file');
end
```

electraMethod.m

```
function result = electraMethod(numericRatings, nAlternatives, nCriteria, alternativeWeights, c, d)
agreementMatrix = zeros(nCriteria, nCriteria);
% calculating agreement matrix
for i = 1:nCriteria
```

```
for j = 1:nCriteria

if i == j
continue;
end

for k = 1:nAlternatives

if numericRatings(k, i) > numericRatings(k, j)
agreementMatrix(i, j) = agreementMatrix(i, j) +
alternativeWeights(k);
end

end

end

end

disp('Agreement Matrix');
disp(agreementMatrix');

% calculating disagreement matrix
disagreementMatrix = zeros(nCriteria, nCriteria);

for i = 1:nCriteria

for j = 1:nCriteria

if i == j
continue;
end

for k = 1:nAlternatives

if numericRatings(k, j) < numericRatings(k, i)
disagreementMatrix(i, j) = disagreementMatrix(i, j) +
numericRatings(k, i) - numericRatings(k, j);
end

end

end

end

disp('Disagreement Matrix');
disp(disagreementMatrix');

% finding limit value of the agreement index
disp("Limit value of the agreement index");
```

```
agreementIndex = zeros(1, nCriteria);

for i = 1:nCriteria

for j = 1:nCriteria

if i == j
continue;
elseif agreementIndex(1, j) == 0
agreementIndex(1, j) = agreementMatrix(i, j);
elseif agreementIndex(1, j) > agreementMatrix(i, j)
agreementIndex(1, j) = agreementMatrix(i, j);
end

end

end

disp(agreementIndex);

% finding limit value of the disagreement index
disp("Limit value of the disagreement index");
disagreementIndex = zeros(1, nCriteria);

for i = 1:nCriteria

for j = 1:nCriteria

if i == j
continue;
elseif disagreementIndex(1, j) == 0
disagreementIndex(1, j) = disagreementMatrix(i, j);
elseif disagreementIndex(1, j) < disagreementMatrix(i, j)
disagreementIndex(1, j) = disagreementMatrix(i, j);
end

end

end

disp(disagreementIndex);

% calculating result
result = 0;
for i = 1:nCriteria
if agreementIndex(i) > c && disagreementIndex(i) < d
disp(agreementIndex(i) + " > " + c + " && " +
disagreementIndex(i) + " < " + d);
result = i;
break;
end
```

```
end
```

modifiedKemmenySnellMethod.m

```
function result = modifiedKemmenySnellMethod(rankingMatrix, 
nAlternatives, nCriteria, alternativeWeights)
% modifiedKemmenySnellMethod - modified Kemmeny-Snell method
matrcies = zeros(nCriteria, nCriteria, nAlternatives);

% Creating matrix of pairwise comparisons
for curAlternative = 1:nAlternatives

for curCriteria = 1:nCriteria

for i = 1:nCriteria
if curCriteria == i
matrcies(curCriteria, i, curAlternative) = 9;
elseif rankingMatrix(curAlternative, curCriteria) > 
rankingMatrix(curAlternative, i)
matrcies(curCriteria, i, curAlternative) = 1;
elseif rankingMatrix(curAlternative, curCriteria) < 
rankingMatrix(curAlternative, i)
matrcies(curCriteria, i, curAlternative) = -1;
end

end

end

end

% displaying created matrix
disp("Matrix of pairwise comparisons: ")
disp("Note: 9 means that that comprasion was skipped")

for i = 1:nAlternatives
disp(matrcies(:, :, i)');
end

% creating matrix of lost
lostMatrix = zeros(nCriteria, nCriteria);

for i = 1:nCriteria

for j = 1:nCriteria

if i == j
lostMatrix(i, j) = 0;
else
```

```
for k = 1:nAlternatives
lostMatrix(i, j) = lostMatrix(i, j) + (alternativeWeights(k) *
abs(matrcies(i, j, k) - 1));
end

end

end

end

disp("Matrix of lost: ")
disp("Note: 0 means that that comprasion was skipped")
disp(lostMatrix');

% Calculating preliminary ranking of alternatives
disp("Calculating preliminary ranking of alternatives: ");
result = zeros(1, nCriteria);
for prep = 1:nCriteria-1
preliminary_ranking_of_alternatives = zeros(1, nCriteria);

for i = 1:nCriteria

for j = 1:nCriteria
preliminary_ranking_of_alternatives(i) =
preliminary_ranking_of_alternatives(i) + lostMatrix(j, i);
end

end

% removing 0 (since its removed lines)
without_zeros_preliminary_ranking_of_alternatives =
preliminary_ranking_of_alternatives(preliminary_ranking_of_alter
natives ~= 0);
disp(without_zeros_preliminary_ranking_of_alternatives);

without_zeros_preliminary_ranking_of_alternatives =
min(without_zeros_preliminary_ranking_of_alternatives);
disp("The best is: " +
without_zeros_preliminary_ranking_of_alternatives);
idx = find(preliminary_ranking_of_alternatives ==
without_zeros_preliminary_ranking_of_alternatives);
lostMatrix(:, idx(1)) = 0; % setting all elements in column to 0
lostMatrix(idx(1), :) = 0; % setting all elements in row to 0
result(prep) = idx(1);
if(prep == nCriteria-1)
disp("The last is: " +
max(preliminary_ranking_of_alternatives));
result(prep+1) = find(preliminary_ranking_of_alternatives ==
max(preliminary_ranking_of_alternatives));
end
```

```
end

disp("Preliminary ranking of alternatives: ");
disp("Note:remember, indexes might change since program does not
know about criteria that was deleted by parreto");
disp(result);

% calculating final ranking of alternatives
disp("Calculating finel result: ");
for i = nCriteria-1:-1:1
if lostMatrix(result(i), result(i+1)) > lostMatrix(result(i+1),
result(i))
disp("R"+result(i)+result(i+1)+" is bigger than
R"+result(i+1)+result(i)+", so we swap them");
temp = result(i);
result(i) = result(i+1);
result(i+1) = temp;
else
disp("R"+result(i)+result(i+1)+" is smaller than
R"+result(i+1)+result(i)+", so we do not swap them");
end
end
disp("Final result: ")
disp(result);
end
```

preferencesMethod.m

```
function alternativeWeights = preferencesMethod(nExperts,
nAlternatives, methodPreference)
% First, convert the methodPreference
for i = 1:nExperts

for j = 1:nAlternatives
methodPreference(i, j) = nAlternatives - methodPreference(i, j);
end

end

% Second, get the sum of converted methodPreference
sumConvertedOpinions = zeros(nAlternatives);

for j = 1:nAlternatives

for i = 1:nExperts
sumConvertedOpinions(j) = sumConvertedOpinions(j) +
methodPreference(i, j);
end

end
```

```
% Third, getting the sum of all
sumAll = 0;
for i = 1:nAlternatives
sumAll = sumAll + sumConvertedOpinions(i);
end

% Fourth, getting the alternative weights
alternativeWeights = zeros(1, nAlternatives);

for j = 1:nAlternatives
alternativeWeights(j) = sumConvertedOpinions(j) / sumAll;
end

end
```

skipComments.m

```
function skipComments(fd)

while true
data = fread(fd, 1, 'char');

if data == '#'
fgetl(fd);
else
break;
end

end

fseek(fd, -1, 'cof');
end
```

main.m

```
fprintf('Welcome to the program for calculating lab4!\n');

filename = 'condition.txt';

if exist(filename, 'file') ~= 2
disp('File with your data was not found, now we will create
it');
disp('Please, fill up the file');
creatingFile();
return;
end

disp('File with your data was found, now we will read it');
fd = fopen(filename, 'r');
```

```
if fd == -1
disp('Error while opening file');
return;
end

% Read data from file
skipComments(fd);
nExperts = fscanf(fd, '%d', 1);
fread(fd, 1, 'char'); % skip newline
skipComments(fd);

nAlternatives = fscanf(fd, '%d', 1);
fread(fd, 1, 'char'); % skip newline
skipComments(fd);

% Reading array of experts
methodPreference = zeros(nExperts, nAlternatives);

for i = 1:nExperts

for j = 1:nAlternatives
methodPreference(i, j) = fscanf(fd, '%d', 1);
fread(fd, 1, 'char');
end

end

% Displaying readded info
disp("Number of experts: " + nExperts);
disp("Number of alternatives: " + nAlternatives);
disp("Method of preference:")
disp(methodPreference);

disp('Weights of alternatives:');
alternativeWeights = preferencesMethod(nExperts, nAlternatives,
methodPreference);
disp(alternativeWeights)

% reading number of criteria for kememeny-snell mythod
skipComments(fd);
nCriteria = fscanf(fd, '%d', 1);
fread(fd, 1, 'char'); % skip newline
skipComments(fd);

% reading array of criteria
rankingMatrix = zeros(nAlternatives, nCriteria);

for i = 1:nAlternatives

for j = 1:nCriteria
```

```
rankingMatrix(i, j) = fscanf(fd, '%d', 1);
fread(fd, 1, 'char');
end

end

disp('Ranking matrix:');
disp(rankingMatrix);

modifiedKemmenySnellMethod(rankingMatrix, nAlternatives,
nCriteria, alternativeWeights);

% reading numeric ratings for electra method
skipComments(fd);
numericRatings = zeros(nAlternatives, nCriteria);

for i = 1:nAlternatives

for j = 1:nCriteria
numericRatings(i, j) = fscanf(fd, '%f', 1);
fread(fd, 1, 'char');
end

end

skipComments(fd);
c = fscanf(fd, '%f', 1);
fread(fd, 1, 'char');
skipComments(fd);
d = fscanf(fd, '%f', 1);
fread(fd, 1, 'char');

disp('Numeric ratings:');
disp(numericRatings);

disp("C* = " + c);
disp("D* = " + d);

electraResult = electraMethod(numericRatings, nAlternatives,
nCriteria, alternativeWeights, c, d);
disp('Electra result:');
disp(electraResult);
disp("Ended Calculations");
```

Приложение Б

```
# This is the file for your condition to run lab 4
# All line starting with # will be ignored
# This program solve ONLY second way of calculating
# NOTE: the set of parreto must be already applied to your task
# There will be used preference method
#
# Please write number of experts
2
# Please write number of alternatives
3
# Please write preferences of two experts
2 2 1
1 3 2
# End of preferences method
#
# Modified Kemeny-Snell algorithm
# Please write number of criteria
5
# Write alternatives with ranking according to each of the criteria
4 2 1 3 4
2 4 3 1 3
2 3 4 3 1
# End of modified Kemeny-Snell algorithm
#
# ELECTRA method
# Please, write dimensionless estimates
0.36 0.8 0.9 0.62 0.36
0.7 0.36 0.2 0.79 0.62
0.7 0.62 0.36 0.62 0.9
# Enter C* value
0.4
# Enter D* value
0.5
# End of ELECTRA method
# End of file
```

Приложение В

```
>> run('./main.m')
Welcome to the program for calculating lab4!
File with your data was found, now we will read it
Number of experts: 2
Number of alternatives: 3
Method of preference:
     2     2     1
     1     3     2

Weights of alternatives:
    0.4286    0.1429    0.4286

Ranking matrix:
     4     2     1     3     4
     2     4     3     1     3
     2     3     4     3     1

Matrix of pairwise comparisons:
Note: 9 means that that comprasion was skipped
     9    -1    -1    -1     0
     1     9    -1     1     1
     1     1     9     1     1
     1    -1    -1     9     1
     0    -1    -1    -1     9

     9     1     1    -1     1
    -1     9    -1    -1    -1
    -1     1     9    -1     0
     1     1     1     9     1
    -1     1     0    -1     9

     9     1     1     1    -1
    -1     9     1     0    -1
    -1    -1     9    -1    -1
    -1     0     1     9    -1
     1     1     1     1     9

Matrix of lost:
Note: 0 means that that comprasion was skipped
         0    0.8571    0.8571    1.1429    1.2857
    1.1429         0    1.1429    0.7143    1.1429
    1.1429    0.8571         0    1.1429    1.0000
    0.8571    1.2857    0.8571         0    0.8571
    0.7143    0.8571    1.0000    1.1429         0

Calculating preliminary ranking of alternatives:
    4.1429    4.1429    4.1429    3.8571    3.7143

The best is: 3.7143
    2.8571    3.0000    3.1429    3.0000

The best is: 2.8571
```

```
    1.8571    2.0000    2.1429

The best is: 1.8571
    1.1429    0.8571

The best is: 0.85714
The last is: 1.1429
Preliminary ranking of alternatives:
Note:remember, indexes might change since program does not
know about criteria that was deleted by parreto
     5     1     2     4     3

Calculating finel result:
R43 is smaller than R34, so we do not swap them
R24 is smaller than R42, so we do not swap them
R12 is smaller than R21, so we do not swap them
R51 is smaller than R15, so we do not swap them
Final result:
     5     1     2     4     3

Numeric ratings:
    0.3600    0.8000    0.9000    0.6200    0.3600
    0.7000    0.3600    0.2000    0.7900    0.6200
    0.7000    0.6200    0.3600    0.6200    0.9000

C* = 0.4
D* = 0.5
Agreement Matrix
         0    0.4286    0.4286    0.5714    0.4286
    0.5714         0    0.4286    0.1429    0.5714
    0.5714    0.5714         0    0.5714    0.5714
    0.4286    0.4286    0.4286         0    0.4286
    0.1429    0.4286    0.4286    0.5714         0

Disagreement Matrix
         0    0.4400    0.5400    0.3500    0.2000
    0.4200         0    0.1000    0.4300    0.5400
    0.8400    0.4200         0    0.8500    0.9600
    0.0800    0.1800    0.2800         0    0.2800
    0.0800    0.4400    0.5400    0.4300         0

Limit value of the agreement index
    0.4286    0.1429    0.5714    0.4286    0.1429

Limit value of the disagreement index
    0.5400    0.5400    0.9600    0.2800    0.5400

0.42857 > 0.4 && 0.28 < 0.5
Index is 4
```