

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"МИРЭА - Российский технологический университет"РТУ МИРЭА |

**Институт** Информационных Технологий

**Кафедра** Вычислительной Техники

**ПРАКТИЧЕСКАЯ РАБОТА № 2**

**Метод Электра II**

**по дисциплине**

**«Теория принятия решений»**

Студент группы: ИКБО-05-19 Выонг Чыонг Шон *(Фамилия студента)*

Руководитель работы Железняк Л.М.\_

*(Фамилия преподавателя)*

|  |  |  |
| --- | --- | --- |
|  |  |  |

Москва 2021

**Метод Электра II**

Задача данной лабораторной работы: выбрать компанию для работы с использованием метода ЭЛЕКТРА.

Проанализировав информацию были выделены варианты решений (альтернативы) и их оценки, и сведены в табл. 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варианты решений | Критерии | | | | | |
| Зарплата  (тыс.руб/мес.)  (+) | Величина компания  (+) | Рабочее время (часов)  (-) | Квалификация персонала  (+) | Отпуск (дней)  (+) | Удаленность от дома (км)  (-) |
| Компания A | 63 | 4 | 6 | 5 | 20 | 30 |
| Компания B | 35 | 4 | 6 | 2 | 30 | 10 |
| Компания C | 49 | 5 | 8 | 4 | 36 | 20 |
| Компания D | 56 | 5 | 10 | 3 | 40 | 25 |
| Компания E | 28 | 2 | 5 | 3 | 60 | 12 |
| Компания F | 63 | 4 | 8 | 4 | 35 | 30 |
| Компания G | 42 | 3 | 7 | 2 | 24 | 5 |
| Компания H | 45.5 | 5 | 8 | 3 | 35 | 20 |
| Вес | 5 | 5 | 3 | 4 | 2 | 3 |
| Стремление | max | max | min | max | max | min |

Таблица 1: Таблица оценок проектов по критериям

Рассмотрим альтернативы A и B (i = A, j = B):

* PAB = 5 + 0 + 0 + 4 + 0 + 0 = 9
* NAB = 0 + 0 + 0 + 0 + 2 + 3 = 5
* DAB = PAB / NAB = 9/5 = 1.8 > 1 – принимаем
* PBA = 0 + 0 + 0 + 0 + 2 + 3 = 5
* NBA = 5 + 0 + 0 + 4 + 0 + 0 = 9
* DBA = PBA / NBA = 5/9 = 0.55 < 1 – отбрасываем

Рассмотрим альтернативы A и C (i = A, j = C):

* PAC = 5 + 0 + 3 + 4 + 0 + 0 = 12
* NAC = 0 + 5 + 0 + 0 + 2 + 3 = 10
* DAC = PAC / NAC = 12/10 = 1.2 > 1 – принимаем
* PCA = 0 + 5 + 0 + 0 + 2 + 3 = 10
* NCA = 5 + 0 + 3 + 4 + 0 + 0 = 12
* DCA = PCA / NCA = 10/12 = 0.83 < 1 – отбрасываем

Рассмотрим альтернативы A и D (i = A, j = D):

* PAD = 5 + 0 + 3 + 4 + 0 + 0 = 12
* NAD = 0 + 5 + 0 + 0 + 2 + 3 = 10
* DAD = PAD / NAD = 12/10 = 1.2 > 1 – принимаем
* PDA = 0 + 5 + 0 + 0 + 2 + 3 = 10
* NDA = 5 + 0 + 3 + 4 + 0 + 0 = 12
* DDA = PDA / NDA = 10/12 = 0.83 < 1 – отбрасываем

Рассмотрим альтернативы A и E (i = A, j = E):

* PAE = 5 + 5 + 0 + 4 + 0 + 0 = 14
* NAE = 0 + 0 + 3 + 0 + 2 + 3 = 8
* DAE = PAE / NAE = 14/8 = 1.75 > 1 – принимаем
* PEA = 0 + 0 + 3 + 0 + 2 + 3 = 8
* NEA = 5 + 5 + 0 + 4 + 0 + 0 = 14
* DEA = PEA / NEA = 8/14 = 0.57 < 1 – отбрасываем

Рассмотрим альтернативы A и F (i = A, j = F):

* PAF = 0 + 0 + 3 + 4 + 0 + 0 = 7
* NAF = 0 + 0 + 0 + 0 + 2 + 0 = 2
* DAF = PAF / NAF = 7/2 = 3.5 > 1 – принимаем
* PFA = 0 + 0 + 0 + 0 + 2 + 0 = 2
* NFA = 0 + 0 + 3 + 4 + 0 + 0 = 7
* DFA = PFA / NFA = 2/7 = 0.28 < 1 – отбрасываем

Рассмотрим альтернативы A и G (i = A, j = G):

* PAG = 5 + 5 + 3 + 4 + 0 + 0 = 17
* NAG = 0 + 0 + 0 + 0 + 2 + 3 = 5
* DAG = PAG / NAG = 17/5 = 3.4 > 1 – принимаем
* PGA = 0 + 0 + 0 + 0 + 2 + 3 = 5
* NGA = 5 + 5 + 3 + 4 + 0 + 0 = 17
* DGA = PGA / NGA = 5/17 = 0.29 < 1 – отбрасываем

Рассмотрим альтернативы A и H (i = A, j = H):

* PAH = 5 + 0 + 3 + 4 + 0 + 0 = 12
* NAH = 0 + 5 + 0 + 0 + 2 + 3 = 10
* DAH = PAH / NAH = 12/10 = 1.2 > 1 – принимаем
* PHA = 0 + 5 + 0 + 0 + 2 + 3 = 10
* NHA = 5 + 0 + 3 + 4 + 0 + 0 = 12
* DHA = PHA / NHA = 10/12 = 0.83 < 1 – отбрасываем

Рассмотрим альтернативы B и C (i = B, j = C):

* PBC = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NBC = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DBC = PBC / NBC = 6/16 = 0.375 < 1 – отбрасываем
* PCB = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NCB = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DCB = PCB / NCB = 16/6 = 2.67 > 1 – принимаем

Рассмотрим альтернативы B и D (i = B, j = D):

* PBD = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NBD = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DBD = PBD / NBD = 6/16 = 0.375 < 1 – отбрасываем
* PDB = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NDB = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DDB = PDB / NDB = 16/6 = 2.67 > 1 – принимаем

Рассмотрим альтернативы B и E (i = B, j = E):

* PBE = 5 + 5 + 0 + 0 + 0 + 3 = 13
* NBE = 0 + 0 + 3 + 4 + 2 + 0 = 9
* DBE = PBE / NBE = 13/9 = 1.44 > 1 – принимаем
* PEB = 0 + 0 + 3 + 4 + 2 + 0 = 9
* NEB = 5 + 5 + 0 + 0 + 0 + 3 = 13
* DEB = PDB / NDB = 9/13 = 0.69 < 1 – отбрасываем

Рассмотрим альтернативы B и F (i = B, j = F):

* PBF = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NBF = 5 + 0 + 0 + 4 + 2 + 0 = 11
* DBF = PBF / NBF = 6/11 = 0.54 < 1 – отбрасываем
* PFB = 5 + 0 + 0 + 4 + 2 + 0 = 11
* NFB = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DFB = PFB / NFB = 11/6 = 1.83 > 1 – принимаем

Рассмотрим альтернативы B и G (i = B, j = G):

* PBG = 0 + 5 + 3 + 0 + 2 + 0 = 10
* NBG = 5 + 0 + 0 + 0 + 0 + 3 = 8
* DBG = PBG / NBG = 10/8 = 1.25 > 1 – принимаем
* PGB = 5 + 0 + 0 + 0 + 0 + 3 = 8
* NGB = 0 + 5 + 3 + 0 + 2 + 0 = 10
* DGB = PGB / NGB = 8/10 = 0.8 < 1 – отбрасываем

Рассмотрим альтернативы B и H (i = B, j = H):

* PBH = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NBH = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DBH = PBH / NBH = 6/16 = 0.375 < 1 – отбрасываем
* PHB = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NHB = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DHB = PHB / NHB = 16/6 = 2.67 > 1 – принимаем

Рассмотрим альтернативы C и D (i = C, j = D):

* PCD = 0 + 0 + 3 + 4 + 0 + 3 = 10
* NCD = 5 + 0 + 0 + 0 + 2 + 0 = 7
* DCD = PCD / NCD = 10/7 = 1.42 > 1 – принимаем
* PDC = 5 + 0 + 0 + 0 + 2 + 0 = 7
* NDC = 0 + 0 + 3 + 4 + 0 + 3 = 10
* DDC = PDC / NDC = 7/10 = 0.7 < 1 – отбрасываем

Рассмотрим альтернативы C и E (i = C, j = E):

* PCE = 5 + 5 + 0 + 4 + 0 + 0 = 14
* NCE = 0 + 0 + 3 + 0 + 2 + 3 = 8
* DCE = PCE / NCE = 14/8 = 1.75 > 1 – принимаем
* PEC = 0 + 0 + 3 + 0 + 2 + 3 = 8
* NEC = 5 + 5 + 0 + 4 + 0 + 0 = 14
* DEC = PEC / NEC = 8/14 = 0.57 < 1 – отбрасываем

Рассмотрим альтернативы C и F (i = C, j = F):

* PCF = 0 + 5 + 0 + 0 + 2 + 3 = 10
* NCF = 5 + 0 + 0 + 0 + 0 + 0 = 5
* DCF = PCF / NCF = 10/5 = 2 > 1 – принимаем
* PFC = 5 + 0 + 0 + 0 + 0 + 0 = 5
* NFC = 0 + 5 + 0 + 0 + 2 + 3 = 10
* DFC = PFC / NFC = 5/10 = 0.5 < 1 – отбрасываем

Рассмотрим альтернативы C и G (i = C, j = G):

* PCG = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NCG = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DCG = PCG / NCG = 16/6 = 2.66 > 1 – принимаем
* PGC = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NGC = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DGC = PGC / NGC = 6/16 = 0.373 < 1 – отбрасываем

Рассмотрим альтернативы C и H (i = C, j = H):

* PCH = 5 + 0 + 0 + 4 + 2 + 0 = 11
* NCH = 0 + 0 + 0 + 0 + 0 + 0 = 0
* DCH = PCH / NCH = 11/0 – делить нельзя – отбрасываем
* PHC = 0 + 0 + 0 + 0 + 0 + 0 = 0
* NHC = 5 + 0 + 0 + 4 + 2 + 0 = 11
* DHC = PHC / NHC = 0/11 = ∞ – принимаем

Рассмотрим альтернативы D и E (i = D, j = E):

* PDE = 5 + 5 + 0 + 0 + 0 + 0 = 10
* NDE = 0 + 0 + 3 + 0 + 2 + 3 = 8
* DDE = PDE / NDE = 10/8 = 1.25 > 1 – принимаем
* PED = 0 + 0 + 3 + 0 + 2 + 3 = 8
* NED = 5 + 5 + 0 + 0 + 0 + 0 = 10
* DED = PED / NED = 8/10 = 0.8 < 1 – отбрасываем

Рассмотрим альтернативы D и F (i = D, j = F):

* PDF = 0 + 5 + 0 + 0 + 2 + 3 = 10
* NDF = 5 + 0 + 3 + 4 + 0 + 0 = 12
* DDF = PDF / NDF = 10/12 = 0.83 < 1 – отбрасываем
* PFD = 5 + 0 + 3 + 4 + 0 + 0 = 12
* NFD = 0 + 5 + 0 + 0 + 2 + 3 = 10
* DFD = PFD / NFD = 12/10 = 1.2 > 1 – принимаем

Рассмотрим альтернативы D и G (i = D, j = G):

* PDG = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NDG = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DDG = PDG / NDG = 16/6 = 2.66 > 1 – принимаем
* PGD = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NGD = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DGD = PGD / NGD = 6/16 = 0.375 < 1 – отбрасываем

Рассмотрим альтернативы D и H (i = D, j = H):

* PDH = 5 + 0 + 0 + 0 + 2 + 0 = 7
* NDH = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DDH = PDH / NDH = 7/6 = 1.16 > 1 – принимаем
* PHD = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NHD = 5 + 0 + 0 + 0 + 2 + 0 = 7
* DHD = PHD / NHD = 6/7 = 0.85 < 1 – отбрасываем

Рассмотрим альтернативы E и F (i = E, j = F):

* PEF = 0 + 0 + 3 + 0 + 2 + 3 = 8
* NEF = 5 + 5 + 0 + 4 + 0 + 0 = 14
* DEF = PEF / NEF = 8/14 = 0.57 < 1 – отбрасываем
* PFE = 5 + 5 + 0 + 4 + 0 + 0 = 14
* NFE = 0 + 0 + 3 + 0 + 2 + 3 = 8
* DFE = PFE / NFE = 14/8 = 1.73 > 1 – принимаем

Рассмотрим альтернативы E и G (i = E, j = G):

* PEG = 0 + 0 + 3 + 4 + 2 + 0 = 9
* NEG = 5 + 5 + 0 + 0 + 0 + 3 = 13
* DEG = PEG / NEG = 9/13 = 0.69 < 1 – отбрасываем
* PGE = 5 + 5 + 0 + 0 + 0 + 3 = 13
* NGE = 0 + 0 + 3 + 4 + 2 + 0 = 9
* DGE = PGE / NGE = 13/9 = 1.44 > 1 – принимаем

Рассмотрим альтернативы E и H (i = E, j =H):

* PEH = 0 + 0 + 3 + 0 + 2 + 3 = 8
* NEH = 5 + 5 + 0 + 0 + 0 + 0 = 10
* DEH = PEH / NEH = 8/10 = 0.8 < 1 – отбрасываем
* PHE = 5 + 5 + 0 + 0 + 0 + 0 = 10
* NHE = 0 + 0 + 3 + 0 + 2 + 3 = 8
* DHE = PHE / NHE = 10/8 = 1.25 > 1 – принимаем

Рассмотрим альтернативы F и G (i = F, j =G):

* PFG = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NFG = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DFG = PFG / NFG = 16/6 = 2.66 > 1 – принимаем
* PGF = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NGF = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DGF = PGF / NGF = 6/16 = 0.375 < 1 – отбрасываем

Рассмотрим альтернативы F и H (i = F, j =H):

* PFH = 5 + 0 + 0 + 4 + 0 + 0 = 9
* NFH = 0 + 5 + 0 + 0 + 0 + 3 = 8
* DFH = PFH / NFH = 9/8 = 1.125 > 1 – принимаем
* PHF = 0 + 5 + 0 + 0 + 0 + 3 = 8
* NHF = 5 + 0 + 0 + 4 + 0 + 0 = 9
* DHF = PHF / NHF = 8/9 = 0.88 < 1 – отбрасываем

Рассмотрим альтернативы G и H (i = G, j =H):

* PGH = 0 + 0 + 3 + 0 + 0 + 3 = 6
* NGH = 5 + 5 + 0 + 4 + 2 + 0 = 16
* DGH = PGH / NGH = 6/16 = 0.375 < 1 – отбрасываем
* PHG = 5 + 5 + 0 + 4 + 2 + 0 = 16
* NHG = 0 + 0 + 3 + 0 + 0 + 3 = 6
* DHG = PHG / NHG = 16/6 = 2.66 > 1 – принимаем

Составляем матрицу, внося вычисленные (и принятые) значения D. Матрица имеет смысл предпочтений проектов между собой. Для нашего

примера матрица выглядит следующим образом (см. табл. 2).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H |
| A | X | 1.8 | 1.2 | 1.2 | 1.75 | 3.5 | 3.4 | 1.2 |
| B | – | X | – | – | 1.44 | – | 1.25 | – |
| C | – | 2.67 | X | 1.42 | 1.75 | 2 | 2.66 | – |
| D | – | 2.67 | – | X | 1.25 | – | 2.66 | 1.16 |
| E | – | – | – | – | X | – | – | – |
| F | – | 1.83 | – | 1.2 | 1.73 | X | 2.66 | 1.125 |
| G | – | – | – | – | 1.44 | – | X | – |
| H | – | 2.67 | ∞ | – | 1.25 | – | 2.66 | X |

Таблица 2: Полная матрица предпочтений проектов, составленная методом Электра

По матрице строится граф предпочтений (рис. 1).

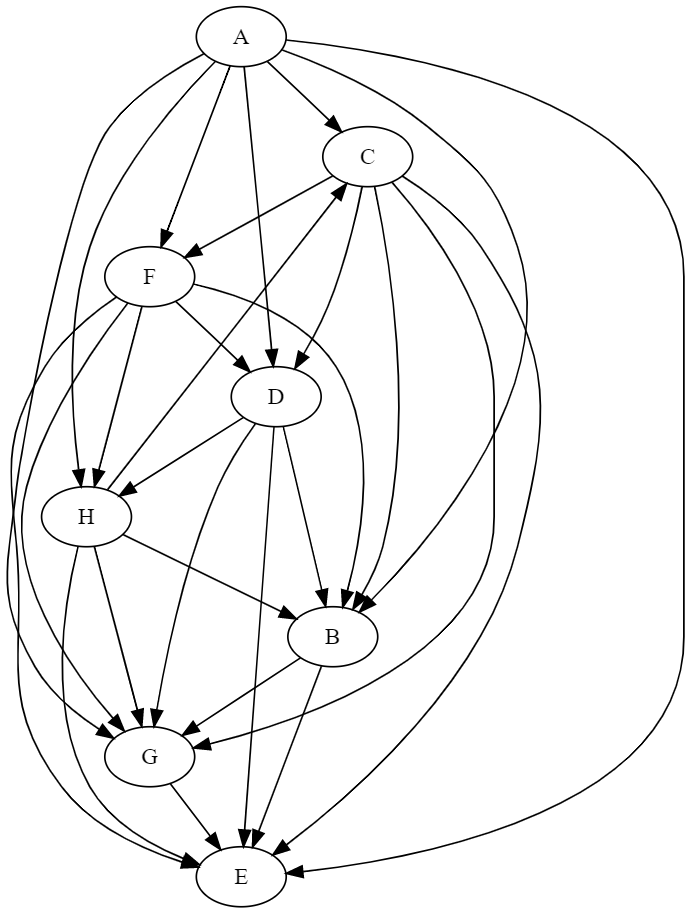


Рис 1: Вид графа предпочтений.

Между проектами C, F, H образуется петля, трудно определить какой проект лучше. Очевидно, что решение не получено, так как в графе присутствуют петли.

Назначим порог отбора предпочтений C = 1.16 (это соответствует тому, что мы попробуем учесть только более сильные связи в графе, не отвлекаясь на малозначимые расхождения в проектах). Таким образом, матрица разрежается. В ней остаются только самые сильные связи (см. табл. 4)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A(0) | B(1) | C(2) | D(3) | E(4) | F(5) | G(6) | H(7) |
| A | X | 1.8 | 1.2 | 1.2 | 1.75 | 3.5 | 3.4 | 1.2 |
| B | – | X | – | – | 1.44 | – | 1.25 | – |
| C | – | 2.67 | X | 1.42 | 1.75 | 2 | 2.66 | – |
| D | – | 2.67 | – | X | 1.25 | – | 2.66 | 1.16 |
| E | – | – | – | – | X | – | – | – |
| F | – | 1.83 | – | 1.2 | 1.73 | X | 2.66 | – |
| G | – | – | – | – | 1.44 | – | X | – |
| H | – | 2.67 | ∞ | – | 1.25 | – | 2.66 | X |

Таблица 3: Матрица предпочтений проектов, при пороге С=1.16

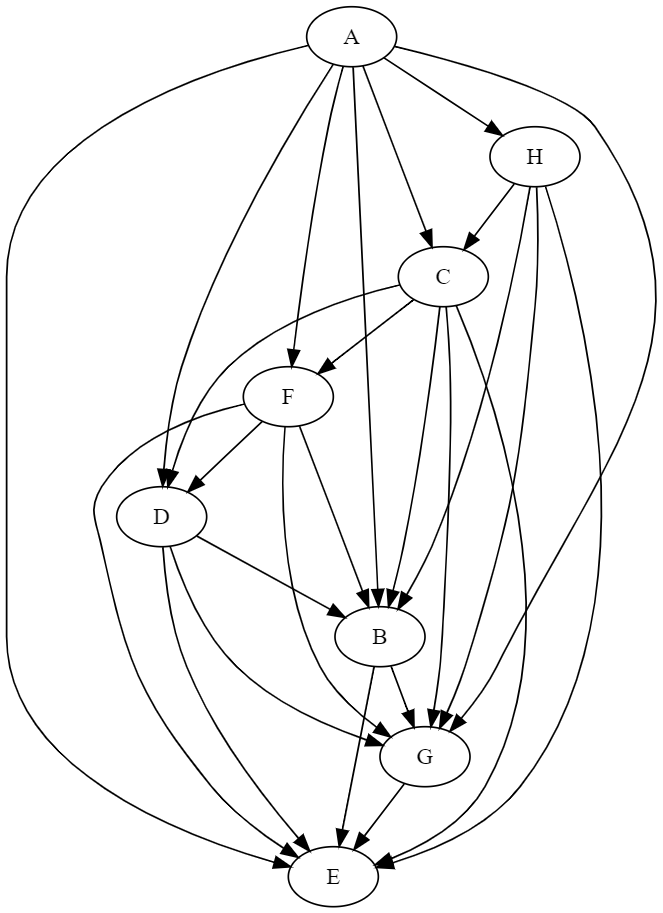


Рис. 2: Вид графа предпочтений для случая порога принятия решений

C = 1.16

Петель в графе нет, при этом граф остался целостным

**Вывод:**

Решение говорит нам о том, что лучший проект — A. На втором месте — проект H, третье место — проект C, четвертое место — проект F, пятое место — проект D, на шестом месте — проект B, на седьмом месте — проект G, и на последнем месте — проект E.

|  |
| --- |
| from collections import defaultdict  import queue  alternatives = []  weight\_crit = []  # Formated output alternative  def print\_alternative(alternatives):      row\_format = "{:^15} |" + "{:^18} | " \* (len(alternatives[0][1]) + len(alternatives[0][2]))      for alternative in alternatives:          al = [alternative[i][j] for i in range(len(alternative)) for j in range(len(alternative[i]))]          print(row\_format.format(\*al))  def input\_data():      fa = open("data/input.txt", "r")      for line in fa:          alter = []          alternative = line.strip("\n").split("|")          for criterion in alternative:              alter.append(criterion.split(";"))          alternatives.append(alter)      fw = open("data/weight.txt", "r")      for line in fw:          w = line.strip("\n").split("|")          for criterion in w:              weight\_crit.append(criterion.split(";"))  def cal\_D(alternatives):      c\_alternatives = alternatives.copy()      tag = c\_alternatives.pop(0)      D = [[0 for x in range(len(c\_alternatives))] for y in range(len(c\_alternatives))]        for i in range(0, len(c\_alternatives)):          for j in range(0, len(c\_alternatives)):              P = 0              N = 0              if i == j:                  continue              for pos\_criterion in range(len(c\_alternatives[i][1])):                  if float(c\_alternatives[i][1][pos\_criterion]) > float(c\_alternatives[j][1][pos\_criterion]):                      P += float(weight\_crit[0][pos\_criterion])                  elif float(c\_alternatives[i][1][pos\_criterion]) < float(c\_alternatives[j][1][pos\_criterion]):                      N += float(weight\_crit[0][pos\_criterion])              for neg\_criterion in range(len(c\_alternatives[i][2])):                  if float(c\_alternatives[i][2][neg\_criterion]) < float(c\_alternatives[j][2][neg\_criterion]):                      P += float(weight\_crit[1][neg\_criterion])                  elif float(c\_alternatives[i][2][neg\_criterion]) > float(c\_alternatives[j][2][neg\_criterion]):                      N += float(weight\_crit[1][neg\_criterion])              if N == 0:                  continue              elif P == 0:                  D[i][j] = float("inf")                if P/N < 1:                  continue              else:                  D[i][j] = P/N      return D  def get\_graph(mD, C):      G = defaultdict(list)      V = len(mD)      for i in range(0, len(mD)):          for j in range(0, len(mD[0])):              if mD[i][j] > C:                  G[i].append(j)      return G  def cyclic(g):      path = set()      visited = set()      def visit(vertex):          if vertex in visited:              return False          visited.add(vertex)          path.add(vertex)          for neighbour in g.get(vertex, ()):              if neighbour in path or visit(neighbour):                  return True          path.remove(vertex)          return False      return any(visit(v) for v in g)  def remove\_circle(mD):      lst = []      for i in range(len(D)):          for j in range(len(D[0])):              if D[i][j] > 0:                  lst.append(D[i][j])      lst = sorted(lst)      for i in lst:          graph = get\_graph(D, i)          if cyclic(graph) == True:              continue          else:              return graph  def find\_root(graph, V):      indegree = [0] \* V      for i in graph:          for j in graph[i]:              indegree[j] += 1        for i in range(V):          if indegree[i] == 0:              return i    def level\_graph(G, V, x):      level = [None] \* V      que = queue.Queue()      que.put(x)      level[x] = 0      while(not que.empty()):          x = que.get()          for i in range(len(G[x])):              b = G[x][i]              que.put(b)              level[b] = level[x] + 1      ans = []      for i in range(V):          l = []          l.append(alternatives[i+1][0])          l.append(level[i])          ans.append(l)      ans.sort(key = lambda x : x[1])      return ans  input\_data()  V = len(alternatives) - 1  D = cal\_D(alternatives)  g = remove\_circle(D)  ans = level\_graph(g, V, find\_root(g, V))  print("Alternatives: ")  print\_alternative(alternatives)  print("\n")  print("Final result:")  for index, alternative in enumerate(ans):      print(str(index + 1) + " - " + alternative[0][0]) |

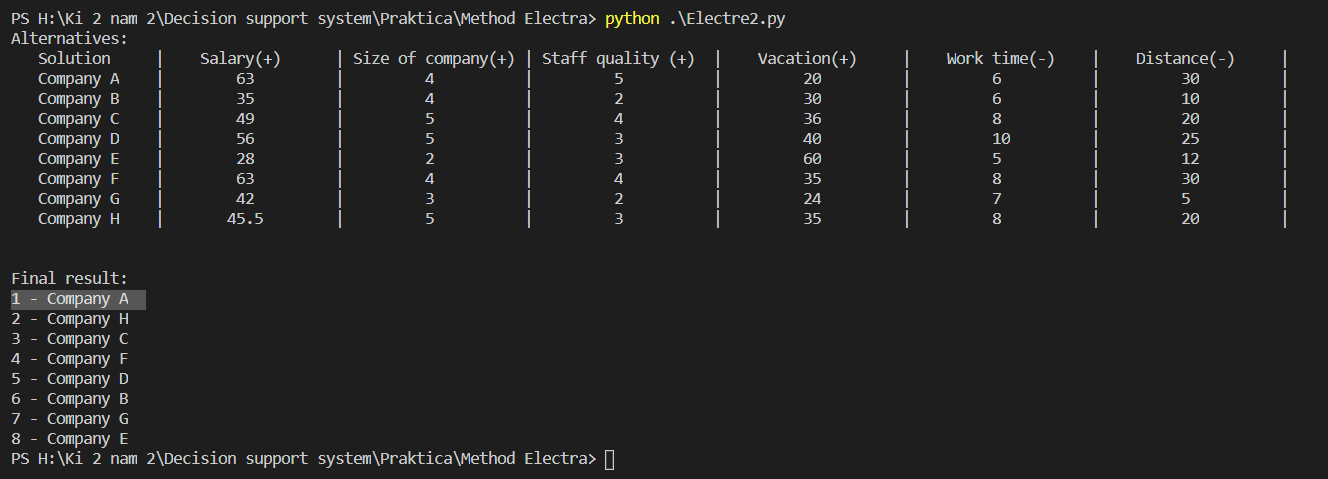


Рис 3: Результат 1

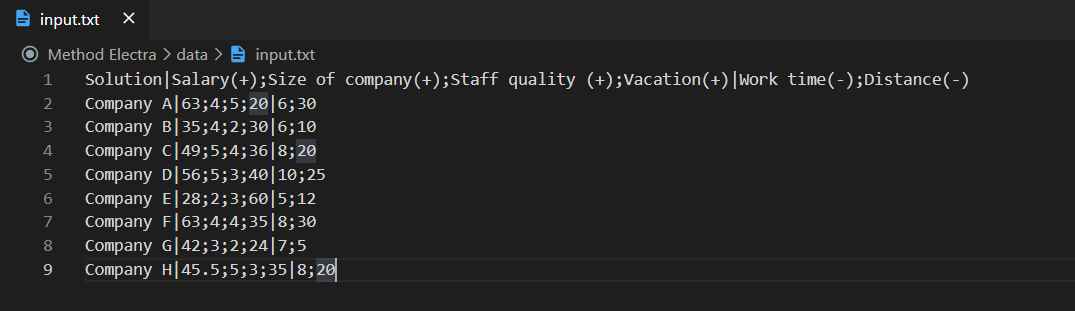


Рис 4: Файл критериев1

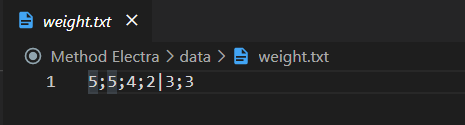


Рис 5: Файл вес (перед вертикального знака - вес положительного критерия, после вертикального знака - вес отрицательного критерия.)

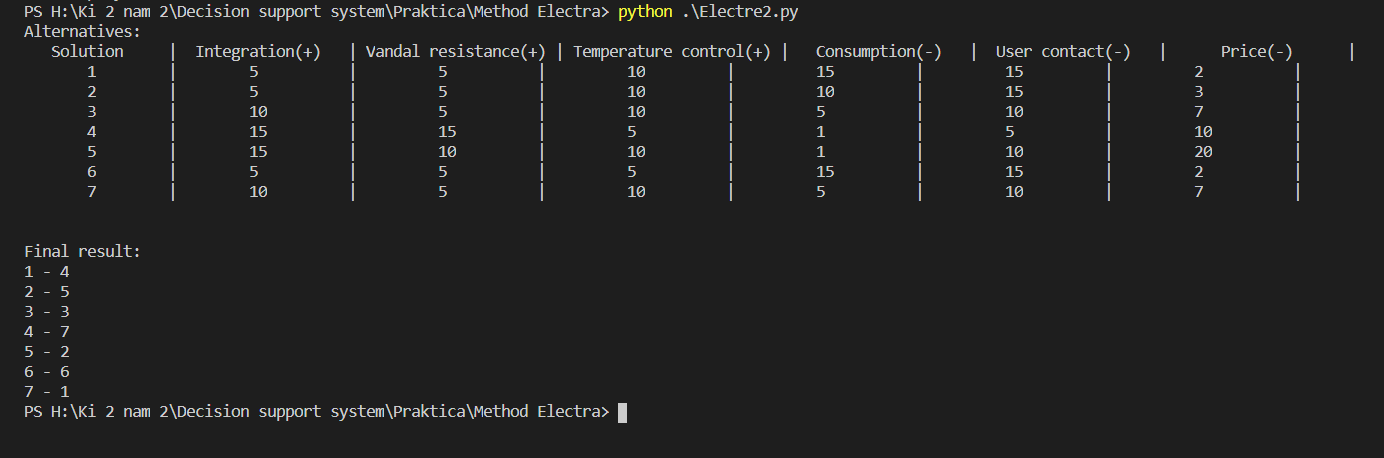


Рис 6: Результат 2

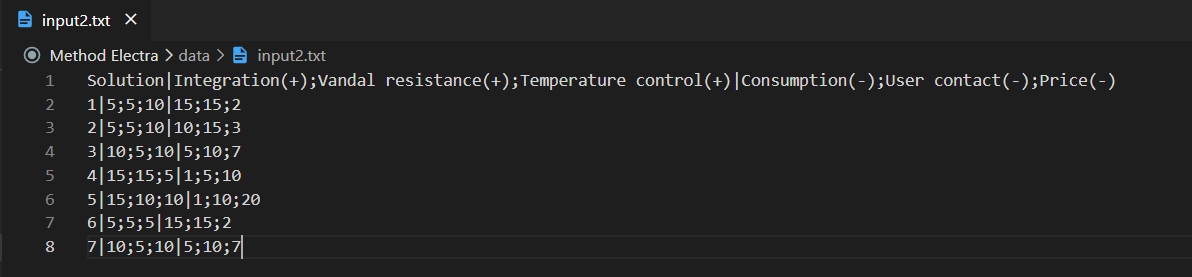


Рис 7: Файл критериев2 (В файле PDF с инструкциями)

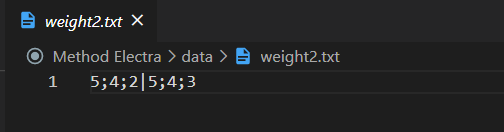


Рис 9: Файл вес