

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** |

**Институт** Информационных Технологий

**Кафедра** Вычислительной Техники

**ПРАКТИЧЕСКАЯ РАБОТА №1**

**по дисциплине**

**«Теория принятия решений»**

**Метод Парето**

Студент группы: ИКБО-41-23 Трофимов А.А. *(Ф. И.О. студента)*

Преподаватель Железняк Л.М.

*(Ф.И.О. преподавателя)*

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

Москва 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc130762925)

[1 МЕТОД ПАРЕТО 4](#_Toc130762926)

[1.1 Выбор Парето-оптимального множества 4](#_Toc130762927)

[1.2 Указание верхних/нижних границ критериев. 4](#_Toc130762928)

[1.3 Субоптимизация 4](#_Toc130762929)

[1.4 Лексикографическая оптимизация 4](#_Toc130762930)

[1.5 Результаты работы программы 4](#_Toc130762931)

[ЗАКЛЮЧЕНИЕ 6](#_Toc130762932)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 7](#_Toc130762933)

[ПРИЛОЖЕНИЯ 8](#_Toc130762934)

ВВЕДЕНИЕ

Метод Парето в теории принятия решений помогает выбрать оптимальное решение, когда нужно учесть несколько факторов. В основном, он заключается в том, чтобы найти такие варианты решений, которые не уступают по важным критериям и не ухудшают ситуацию в других аспектах. Например, если нужно выбрать проект для инвестиций, важно учитывать несколько факторов: стоимость, риск, сроки и возможную прибыль. Метод Парето помогает найти такие проекты, которые будут наилучшими по этим критериям, не будучи хуже по каким-то другим. То есть решение, при котором нельзя улучшить один из факторов, не ухудшая другие. Этот метод полезен, когда нужно сбалансировать несколько интересов, например, в бизнесе или при принятии решений в команде. Он позволяет находить решения, которые максимально удовлетворяют все стороны и минимизируют негативные последствия для любой из них. В общем, метод Парето помогает принимать более сбалансированные и рациональные решения, когда нужно учитывать множество переменных и ограничений.

1. МЕТОД ПАРЕТО

Метод Парето представляет собой сравнение критериев различных альтернатив. При сравнении берутся все критерии и сравниваются образом доминации.

Если положительный критерий у альтернативы А больше, чем у альтернативы Б, то альтернатива А считается доминирующей, в противном случае доминированной.

Если отрицательный критерий у альтернативы А больше, чем у альтернативы Б, то альтернатива Б считается доминирующей, иначе считается доминированной.

Если не все критерии у альтернативы А считаются доминирующими, то такая альтернатива считается несравнимой с альтернативой Б.

## **Выбор Парето-оптимального множества**

Приведём пример выбора молока в магазине с использованием Парето оптимального множества решений. (Таблица 1)

*Таблица 1 – Пример таблицы*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Альтернативы | Цена (руб)(-) | Вес (г)(+) | Калорийность(-) | Срок хранения(+) |
| 1.Простоквашино | 99 | 930 | 53 | 16 |
| 2.Домик в деревне | 89 | 930 | 53 | 15 |
| 3.Молочный знак | 71 | 900 | 53 | 14 |
| 4.ЭкоНива | 114 | 1000 | 60 | 365 |
| 5.Просто | 85 | 970 | 53 | 180 |
| 6.Сарафаново | 129 | 970 | 46 | 180 |
| 7.Зелёная линия | 79 | 900 | 40 | 10 |
| 8.Искренне ваш | 85 | 930 | 53 | 16 |
| 9.Экомилк | 89 | 900 | 70 | 21 |

Сравним попарно все альтернативы и сведём их в таблицу 2

*Таблица 2 – Сравнение*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | x | x | x | x | x | x | x | x | x |
| 2 | н | x | x | x | x | x | x | x | x |
| 3 | н | н | x | x | x | x | x | x | x |
| 4 | н | н | н | x | x | x | x | x | x |
| 5 | A5 | A5 | н | н | x | x | x | x | x |
| 6 | н | н | н | н | н | x | x | x | x |
| 7 | н | н | н | н | н | н | x | x | x |
| 8 | А8 | A8 | н | н | н | н | н | x | x |
| 9 | н | н | н | н | н | н | н | н | x |

# Указание верхних/нижних границ критериев.

Установим для таблицы нижнюю границу: вес не менее 950 и срок хранения не менее 20

*Таблица 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Альтернативы | Цена (руб)(-) | Вес (г)(+) | Калорийность(-) | Срок хранения(+) |
| 4.ЭкоНива | 114 | 1000 | 60 | 365 |
| 5.Просто | 85 | 970 | 53 | 180 |
| 6.Сарафаново | 129 | 970 | 46 | 180 |

Из них оптимальными по Парето является 5.

**1.3 Субоптимизация**

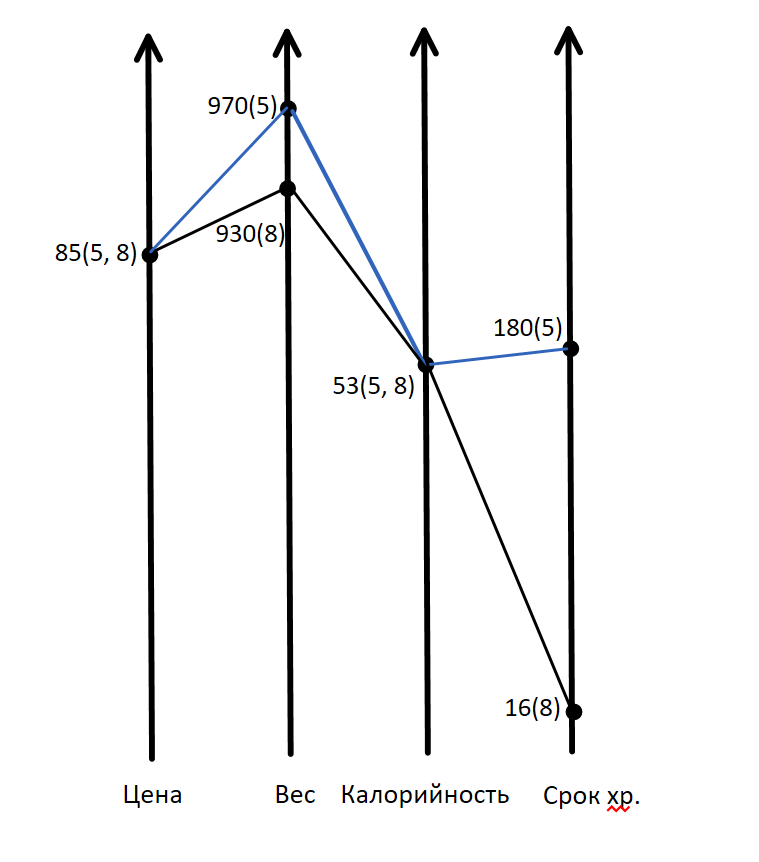
Пусть в качестве главного критерия выступает критерий цена   
Вес не менее 930, каллорийность не менее 53, срок хранения не менее 20

*Таблица 4*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Альтернативы | Цена (руб)(-) | Вес (г)(+) | Калорийность(-) | Срок хранения(+) |
| 4.ЭкоНива | 114 | 1000 | 60 | 365 |
| 5.Просто | 85 | 970 | 53 | 180 |

Из таблицы 4 видно что остались варианты 4 и 5 из них вариант 5 имеет меньшую цену

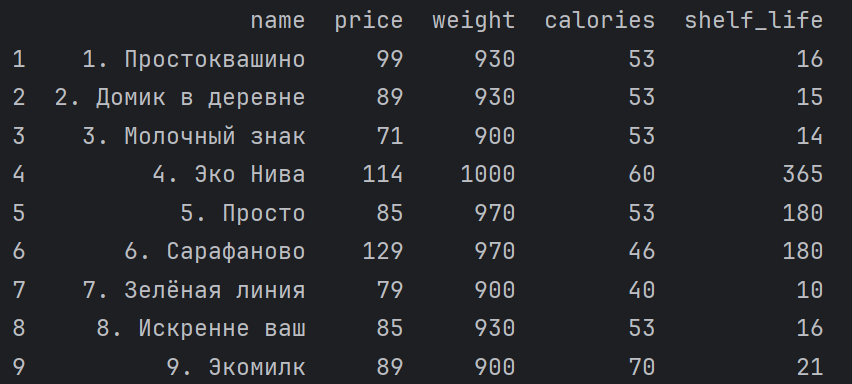
* 1. **Лексикографическая оптимизация**



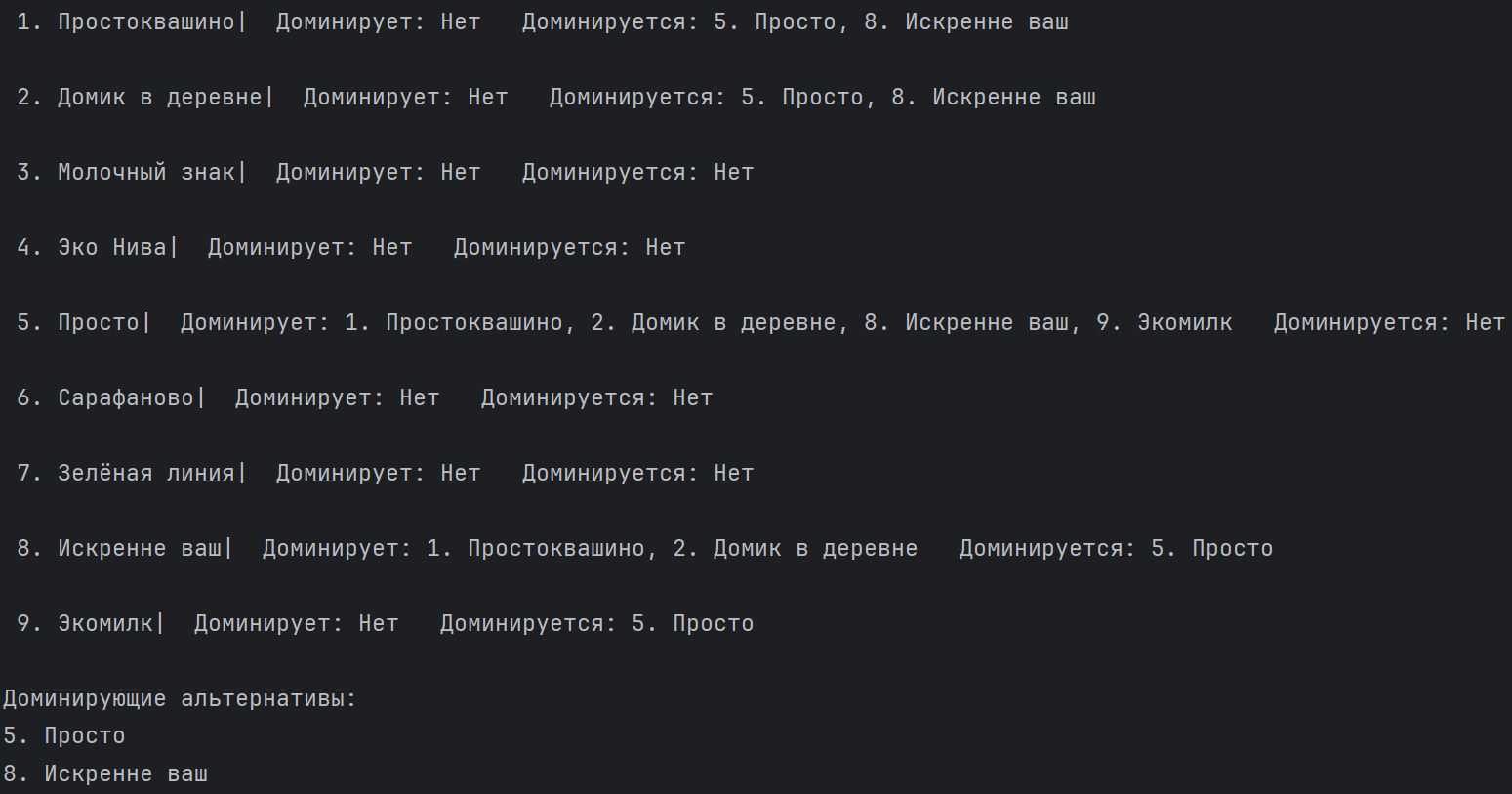
*Рисунок 1 - Лексикографическая оптимизация*

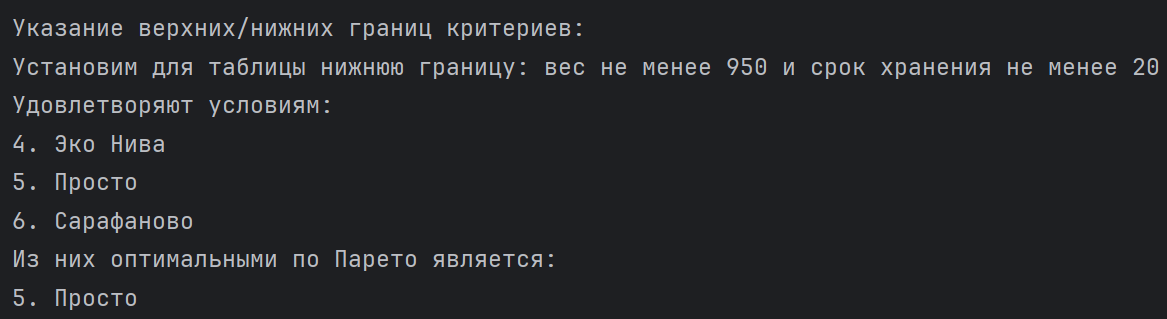
Исходя из графика лучший вариант 5

**1.5 Результаты работы программы**

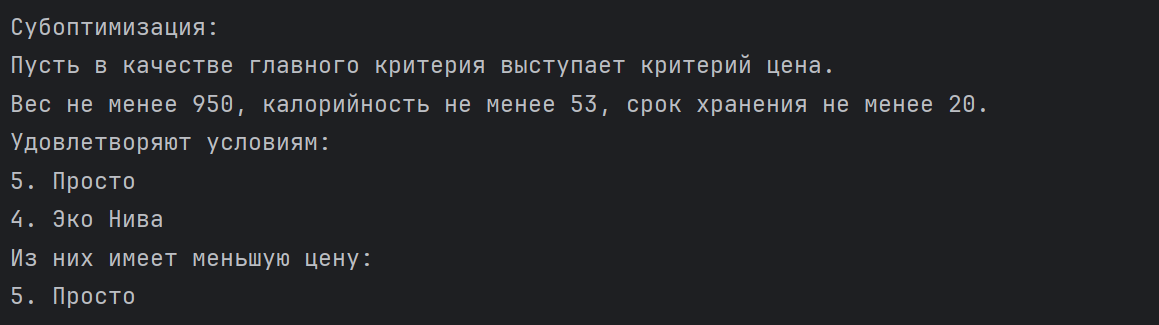
****

**Рисунок 2.0 –** Таблица альтернатив

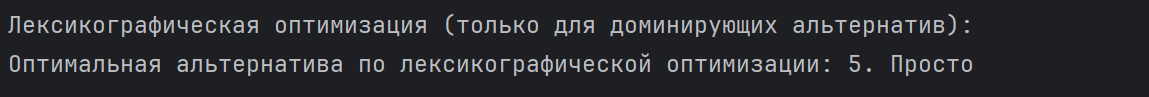
**Рисунок 2.1 –** Выбор Парето-оптимального множества



**Рисунок 2.2 –** Указание верхних/нижних границ критериев



**Рисунок 2.3 -** Субоптимизация



**Рисунок 2.4 -** Лексикографическая оптимизация

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной работы рассмотрен метод Парето для многокритериального выбора оптимального варианта. В качестве примера использовалась задача выбора молока в магазине, для чего были выделены альтернативы, проанализированы их критерии.

Метод Парето является эффективным инструментом для предварительного отбора решений в многокритериальных задачах, но для окончательного выбора оптимального варианта зачастую необходимо применение дополнительных методов сужения множества решений, учитывающих предпочтения лица, принимающего решение (ЛПР).

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

**ПРИЛОЖЕНИЯ**

Приложение А – Код реализации метода Парето на языке Python.

**Приложение А**

Код реализации метода Парето на языке Python.

Листинг А.1. . Реализация Парето.

|  |
| --- |
| import pandas as pd  alternatives = [  {'name': '1. Простоквашино', 'price': 99, 'weight': 930, 'calories': 53, 'shelf\_life': 16},  {'name': '2. Домик в деревне', 'price': 89, 'weight': 930, 'calories': 53, 'shelf\_life': 15},  {'name': '3. Молочный знак', 'price': 71, 'weight': 900, 'calories': 53, 'shelf\_life': 14},  {'name': '4. Эко Нива', 'price': 114, 'weight': 1000, 'calories': 60, 'shelf\_life': 365},  {'name': '5. Просто', 'price': 85, 'weight': 970, 'calories': 53, 'shelf\_life': 180},  {'name': '6. Сарафаново', 'price': 129, 'weight': 970, 'calories': 46, 'shelf\_life': 180},  {'name': '7. Зелёная линия', 'price': 79, 'weight': 900, 'calories': 40, 'shelf\_life': 10},  {'name': '8. Искренне ваш', 'price': 85, 'weight': 930, 'calories': 53, 'shelf\_life': 16},  {'name': '9. Экомилк', 'price': 89, 'weight': 900, 'calories': 70, 'shelf\_life': 21},  ]  df = pd.DataFrame(alternatives)  df.index = df.index + 1  print(df, '\n')  def dominates(a, b):  # Проверяем, что 'a' не хуже 'b' по всем критериям  better\_in\_any = False  if a['price'] > b['price']: # цена: меньше лучше  return False  if a['price'] < b['price']:  better\_in\_any = True  if a['weight'] < b['weight']: # вес: больше лучше  return False  if a['weight'] > b['weight']:  better\_in\_any = True  if a['calories'] > b['calories']: # калории: меньше лучше  return False  if a['calories'] < b['calories']:  better\_in\_any = True  if a['shelf\_life'] < b['shelf\_life']: # срок: больше лучше  return False  if a['shelf\_life'] > b['shelf\_life']:  better\_in\_any = True  return better\_in\_any  def compare\_alternatives(alternatives):  relations = {}  for i, a in enumerate(alternatives):  relations[a['name']] = {'dominates': [], 'dominated\_by': [], 'incomparable': []}  for j, b in enumerate(alternatives):  if i == j:  continue  if dominates(a, b):  relations[a['name']]['dominates'].append(b['name'])  elif dominates(b, a):  relations[a['name']]['dominated\_by'].append(b['name'])  else:  relations[a['name']]['incomparable'].append(b['name'])  return relations  def Lower\_boundaries(a):  if a['weight'] < 950:  return False  elif a['shelf\_life'] < 20:  return False  else:  return True  def ParetoLowerBoundaries():  global Lower\_bound  Lower\_bound = []  for i in alternatives:  if Lower\_boundaries(i):  Lower\_bound.append(i['name'])  def dada(a, b):  global dada\_list  dada\_list = []  for i in a:  if i in b:  dada\_list.append(i)  def Sub\_optimization(a):  if a['weight'] < 950:  return False  elif a['calories'] < 53:  return False  elif a['shelf\_life'] < 20:  return False  else:  return True  def ParetoSubOptimization():  global Sub\_opt  Sub\_opt = []  for i in alternatives:  if Sub\_optimization(i):  Sub\_opt.append([i['price'], i['name']])  def lexicographic\_optimization(alternatives, criteria\_order):  remaining = alternatives  for criterion, reverse in criteria\_order:  # Сортируем по текущему критерию  remaining.sort(key=lambda x: x[criterion], reverse=reverse)  # Выбираем лучшие по текущему критерию  best\_value = remaining[0][criterion]  remaining = [alt for alt in remaining if alt[criterion] == best\_value]  return remaining[0] if remaining else None  def main():  # Сравнение альтернатив  relations = compare\_alternatives(alternatives)  for alt, rel in relations.items():  print(f" {alt}|", f" Доминирует: {', '.join(rel['dominates']) if rel['dominates'] else 'Нет'}",  f" Доминируется: {', '.join(rel['dominated\_by']) if rel['dominated\_by'] else 'Нет'}")  print()  # Доминирующие альтернативы  dominant\_alternatives = [alt for alt, rel in relations.items() if rel['dominates']]  print("Доминирующие альтернативы:")  for i in sorted(dominant\_alternatives):  print(i)  print()  # Указание верхних/нижних границ критериев  print('Указание верхних/нижних границ критериев:')  print('Установим для таблицы нижнюю границу: вес не менее 950 и срок хранения не менее 20')  print('Удовлетворяют условиям:')  ParetoLowerBoundaries()  for i in sorted(Lower\_bound):  print(i)  print('Из них оптимальными по Парето является:')  dada(Lower\_bound, dominant\_alternatives)  for i in sorted(dada\_list):  print(i)  # Субоптимизация  print('\nСубоптимизация:')  print('Пусть в качестве главного критерия выступает критерий цена.')  print('Вес не менее 950, калорийность не менее 53, срок хранения не менее 20.')  print('Удовлетворяют условиям:')  ParetoSubOptimization()  for i in sorted(Sub\_opt):  print(i[1])  print('Из них имеет меньшую цену:')  print(min(Sub\_opt)[1])  # Лексикографическая оптимизация (только для доминирующих альтернатив)  print('\nЛексикографическая оптимизация (только для доминирующих альтернатив):')  dominant\_alt\_objects = [alt for alt in alternatives if alt['name'] in dominant\_alternatives]  criteria\_order = [('price', False), ('weight', True), ('calories', False), ('shelf\_life', True)]  optimal = lexicographic\_optimization(dominant\_alt\_objects, criteria\_order)  print(f"Оптимальная альтернатива по лексикографической оптимизации: {optimal['name']}")  if \_\_name\_\_ == '\_\_main\_\_':  main() |