МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева»

На правах рукописи

Шкаберина Гузель Шарипжановна

МОДЕЛИ И АЛГОРИТМЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ПРОДУКЦИИ

05.13.01 – Системный анализ, управление и обработка информации

(космические и информационные технологии)

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Казаковцев Л.А.

Красноярск – 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования. Современные средства сбора данных позволяют аккумулировать большие объемы многомерной информации об объекте исследования. Эта информация становится ценным источником знаний об объекте при ее обработке соответствующими методами и алгоритмами интеллектуального анализа. Задачи автоматической классификации называют задачами автоматической группировки, задачами кластерного анализа (обучение без учителя). Методы автоматической группировки (АГ) являются частью машинного обучения, актуальность которого возрастает с каждым годом.

АГ предполагает разделение множества объектов на подмножества (группы) так, чтобы объекты из одного подмножества были более похожи друг на друга, чем на объекты из других подмножеств по какому-либо критерию. Методы АГ на сегодняшний день применяются во многих сферах деятельности, например, в биологии, биоинформатике, медицине, маркетинге, в производстве при проверке качества промышленных изделий и т.д. Существуют отрасли с повышенными требованиями к качеству продукции, где решение задач АГ требует получения максимально точного и стабильного результата. Под точностью подразумеваем снижение доли ошибок АГ, а под стабильностью – повторяемость результата при многократных запусках алгоритма.

Многие широко используемые модели АГ являются моделями теории размещения. Для непрерывных задач размещения на сегодняшний день разработаны алгоритмы лишь для наиболее распространенных мер расстояния (евклидово, манхэттенское). Однако, учет особенностей пространства признаков конкретной практической задачи при выборе меры расстояния может привести к повышению точности АГ объектов.

Поиск алгоритма АГ объектов, обладающего одновременно высокой точностью и стабильностью результата, и при этом высокой скоростью работы, является одной из проблем АГ объектов. В представленной работе рассматриваем задачу автоматической группировки, а также задачу классификации. Диссертационная работа посвящена исследованию и разработке новых

алгоритмов автоматической группировки объектов, которые позволяют повысить точность и стабильность результата решения практических задач.

Степень разработанности темы. Одной из наиболее известных моделей кластерного анализа является модель k-средних, которая была предложена Г. Штейнгаузом и в дальнейшем алгоритмически реализована С. Ллойдом. В работе Л. Кауфмана и П. Дж. Руссива представлены близкая модель k-медоид. Весомый вклад в задачи автоматической группировки и размещения объектов внесли Б. Дюран и П. Оделл. Ц. Дрезнером, Х. Хамахером, П. Хансеном, Ю. А. Кочетовым и Н. Младеновичем представлен метод поиска с чередующимися окрестностями для задач с большой размерностью данных.

О. Алпом, Э. Эркутом и Ц. Дрезнером, а в дальнейшем А.Н. Антамошкиным и Л.А. Казаковцевым предложены генетические и другие алгоритмы с жадной эвристической процедурой для задачи автоматической группировки на основе моделей теории размещения.

Задачи автоматической группировки на основе разделения смесей вероятностных распределений исследованы С. Ньюкомбом, К. Пирсоном, Б. Эвериттом, Д.М. Титтерингтоном, Дж. Маклаланом, В.Ю. Королевым, Дж. Гримом, О.К. Исаенко, В.Ю. Урбахом, А.П. Демстером, Н.М. Лейрда, Д.Б. Рубиным. Д.В. Сташков распространил подход Казаковцева-Антамошкина на такие задачи. В работах И.П. Рожнова и др. предложены алгоритмы поиска с чередованием жадных эвристических процедур для задач k-средних, k-медоид**.**

Улучшить результат АГ объектов с повышенными требованиями к точности и стабильности результата известными алгоритмами на текущий момент крайне трудно без значительного увеличения временных затрат. При решении практических задач АГ объектов, например, при решении задач выделения однородных партий промышленной продукции, вопросы вызывает адекватность моделей и, как следствие, точность АГ промышленной продукции. Таким образом, существует запрос на разработку новых моделей. Все же возможна также и разработка алгоритмов, обеспечивающих дальнейшее улучшение результата на основе выбранной модели, например, модели k-средних.

Объектом диссертационного исследования являются задачи автоматической группировки многомерных данных. **Предметом исследования** – модели и алгоритмы их решения.

Целью исследования является повышение точности и стабильности результата решения задач автоматической группировки объектов.

Поставленная цель достигается путем решения следующих **задач**:

1. сравнительный анализ известных алгоритмов решения задач автоматической группировки объектов с повышенными требованиями к точности разделения объектов на группы и стабильности результата при многократных запусках алгоритмов;

2. построение модели автоматической группировки объектов на основе модели k-средних с расстоянием Махаланобиса, а также с применением методов факторного анализа для предварительного снижения размерности исходных данных, которая позволяет снизить долю ошибок автоматической группировки промышленной продукции по сравнению с известными моделями;

3. разработка генетического алгоритма для задачи k-средних с перекрестной мутацией, позволяющего повысить точность и стабильность решения по достигаемому значению целевой функции за фиксированное время выполнения по сравнению с известными алгоритмами автоматической группировки объектов;

4. разработка алгоритма обучения двухслойной сигмоидальной искусственной нейронной сети с регуляризацией, повышающего точность решения задачи классификации промышленных изделий на однородные производственные партии при наличии малоинформативных признаков.

Научная новизна:

1. Предложена новая модель для решения задач автоматической группировки промышленной продукции на основе модели k-средних с расстоянием Махаланобиса с применением метода главных компонент. Применение новой модели позволяет повысить точность решения (индекс Рэнда) для задачи выделения однородных производственных партий изделий по данным тестовых испытаний.

2. Предложен новый алгоритм автоматической группировки объектов, основанный на оптимизационной модели k-средних с мерой расстояния Махаланобиса и средневзвешенной ковариационной матрицей, рассчитанной по обучающей выборке. Алгоритм позволяет снизить долю ошибок (повысить индекс Рэнда) при выявлении однородных производственных партий продукции по результатам тестовых испытаний.

3. Разработан новый генетический алгоритм для задачи k-средних с применением единой жадной агломеративной эвристической процедуры в качестве оператора скрещивания и оператора мутации. Применение данного алгоритма позволяет статистически значимо повысить точность результата (улучшить достигаемое значение целевой функции в рамках выбранной математической модели решения задачи автоматической группировки), а также его стабильность, за фиксированное время, по сравнению с известными алгоритмами автоматической группировки.

4. Разработан новый алгоритм обучения двухслойной сигмоидальной искусственной нейронной сети с регуляризацией, демонстрирующий более высокую точность классификации промышленной продукции по данным тестовых испытаний в сравнении с методами обучения таких нейронных сетей при известных методах регуляризации.

Теоретическая значимость работы. Предложенный комплекс алгоритмов и моделей дополняет список эффективных методов решения задач АГ объектов. Принцип использования единой процедуры в качестве оператора скрещивания и мутации создает основу для синтеза новых эффективных алгоритмов для более широкого круга NP-трудных задач.

Практическая значимость работы. Предложенные модели автоматической группировки могут применяться для задач разделения объектов как при производстве и тестировании продукции, так и в других отраслях, где требуется классификация изделий с особыми требованиями качества. Использование предложенных алгоритмов решения задач автоматической группировки и классификации в различных системах, предназначенных для решения таких задач

за фиксированное время, позволяет существенно повысить точность получаемых решений в рамках выбранной модели по получаемому значению целевой функции. Диссертационное исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации в рамках государственного задания № FEFE-2020-0013 «Развитие теории самоконфигурирующихся алгоритмов машинного обучения для моделирования и прогнозирования характеристик компонентов сложных систем».

Методология и методы исследования. Методологической базой являются работы по методам кластеризации и классификации. Использованы методы системного анализа, исследования операций, теории оптимизации, прикладной статистики, корреляционного анализа, факторного анализа, интеллектуального анализа данных.

Положения, выносимые на защиту:

1. Модель автоматической группировки промышленной продукции на основе модели k-средних с расстоянием Махаланобиса с применением метода главных компонент в комбинации с отбором информативных признаков позволяет повысить точность решения (индекс Рэнда) для задачи выделения однородных производственных партий изделий по данным тестовых испытаний.

2. Алгоритм автоматической группировки объектов, основанный на оптимизационной модели k-средних с мерой расстояния Махаланобиса и средневзвешенной ковариационной матрицей, рассчитанной по обучающей выборке и учитывающей обобщенные корреляционные зависимости между признаками объектов в однородных группах, позволяет повысить точность решения (по индексу Рэнда) при выявлении однородных производственных партий продукции по результатам тестовых испытаний.

3. Генетический алгоритм для задачи k-средних с применением единой жадной агломеративной эвристической процедуры в качестве оператора скрещивания и оператора мутации, за счет увеличения разнообразия в популяции, повышает точность результата (улучшет достигаемое значение целевой функции в рамках выбранной математической модели решения задачи автоматической

группировки), а также его стабильность, за фиксированное время, по сравнению с известными алгоритмами автоматической группировки.

4. Алгоритм обучения двухслойной сигмоидальной искусственной нейронной сети с регуляризацией позволяет достичь более высокой точности классификации промышленной продукции по данным тестовых испытаний по сравнению с методами обучения таких нейронных сетей с известными методами регуляризации.

Степень достоверности результатов. Достоверность результатов подтверждается корректным применением современных методов исследования, которые были применены в большом наборе экспериментов.

Апробация результатов. Основные положения и результаты докладывались на международных конференциях и семинарах: Mathematical Optimization Theory and Operations Research (MOTOR, 2019, г. Екатеринбург и 2020 г., Новосибирск), «Advanced Technologies in Material Science, Mechanical and Automation Engineering» (04-06 апреля 2019 г., Красноярск), 2019 International Conference on Information Technologies (InfoTech, 2019 и 2020 гг., Болгария), 2019 International Russian Automation Conference (RusAutoCon, 2019 г., Сочи), «Advanced Technologies in Aerospace, Mechanical and Automation Engineering» - MIST: Aerospace (2019, г. Красноярск), Workshop on science of DataScience (2019, Trieste, Italy), Математические модели принятия решений (2020, Новосибирск), Математическое моделирование и дискретная оптимизация (2020, Омск), Проблемы математического и численного моделирования (2020, Красноярск), Математическое моделирование (2020, г. Москва).

Публикации. Основные теоретические и практические результаты диссертации изложены в 16 публикациях, среди которых 4 работы в ведущих рецензируемых журналах, рекомендуемых действующим перечнем ВАК, 11 – в международных изданиях, индексируемых в системах цитирования Web of Science и Scopus. Имеется свидетельство о государственной регистрации программы для ЭВМ.

ГЛАВА 1. АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ОБЪЕКТОВ

Глава посвящена анализу текущего состояния проблем, связанных с автоматической группировкой объектов, и обзору методов их решения. Проанализированы и представлены существующие проблемы в области автоматической группировки объектов при обязательных требованиях к высокой точности и стабильности получаемого результата при многократных запусках алгоритма.

1.1 Постановка задачи автоматической классификации объектов

Согласно прогнозам, представленным международной исследовательской компанией IDC (International Data Corporation) в своем докладе «Data Age 2025», прогнозируемый рост объема данных на 2025 год составит $163*10^{21}$ байтов [1]. С каждым годом объем информации в мире увеличивается в геометрической прогрессии. Важно из огромного количества многомерной информации извлечь качественные данные, которые являются ценным источником знаний об объекте при ее обработке соответствующими методами и алгоритмами интеллектуального анализа. В зависимости от методов и способов сбора, хранения, обработки, передачи, анализа и оценки информации можем получить различные решения. На этапе анализа данных можно получить максимально полезную информацию об объекте. На данном этапе применяются следующие статистические методы анализа: корреляционный анализ, регрессионный анализ, канонический анализ, методы сравнения средних, частотный анализ, кросстабуляция, анализ соответствий, кластерный анализ, дискриминантный анализ, факторный анализ, деревья классификации, анализ главных компонент и классификация, многомерное шкалирование, моделирование структурными уравнениями, методы анализа выживаемости, временные ряды, нейронные сети, планирование экспериментов, карты контроля качества и другие [2].

Выделяют два класса задач обучения при анализе данных: обучение без учителя (задача кластеризации, объединение объектов в группы (кластеры), принадлежность которых к определенному кластеру заранее неизвестна) и обучение с учителем (задача классификации, в обучающей выборке объект заранее отнесен к определенному классу) [3].

В процессе группировки объектов некоторого множества в определенные группы (подмножества) учитываются общие признаки объекта и алгоритмы, с помощью которых происходило разделение.

Задачу автоматической классификации (она же – задача автоматической группировки, задача кластеризации, задача обучения без учителя, численная таксономия) можно описать следующим образом: учитывая меру подобия (различия), выраженную, например, в виде функции расстояний, разбить N объектов некоторого множества на k непересекающихся подмножеств так, чтобы объекты одного подмножества обладали схожими признаками друг с другом и не обладали ими с объектами из других подмножеств. Представленную задачу также называют задачей четкой кластеризации.

В свою очередь, в задачах нечеткой кластеризации для каждого объекта из некоторого множества вычисляется степень его принадлежности каждому k-му подмножеству.

Полученный результат кластеризации зависит от первоначально выбранных количества подмножеств, а также выбранной меры подобия (различия) [4].

Задачи автоматической классификации на сегодняшний день применяются во многих сферах деятельности, например, обработка информации, получаемой с метеорологических искусственных спутников Земли [5], определение безопасных моделей пассажирских самолетов [6], в задачах из области медицины и ветеринарии [7,8,9], сельского хозяйства [10,11], биологии [12,13], при анализе результатов социологических исследований [14,15], при проверке качества промышленных изделий при производстве [16,17,18,19,20].

В [21] предлагается следующая постановка задачи кластерного анализа: пусть имеется выборка объектов исследования $s=\{o^{(1)},...,o^{(N)}\}$. Требуется

сформировать $k \geq 2$ классов (групп объектов); число классов может быть как выбрано заранее, так и не задано (в последнем случае оптимальное количество кластеров должно быть определено автоматически). Каждый объект описывается с помощью набора переменных $X_1,...,X_M$. Набор $X=\{X_1 ,...,X_M\}$ может включать переменные разных типов. Пусть D_j обозначает множество значений переменной X_j. Обозначим через $x = x(o) = x_1(o),...,x_M(o)$ набор наблюдений переменных для объекта o, где $x_j(o)$ есть значение переменной X_j для данного объекта. Соответствующий выборке набор наблюдений переменных будем представлять в виде таблицы данных V с N строками и M столбцами: $V=\{x_j^{(i)}\}$, $i = 1,2,...,N$, $j=1,2,...,M$; при этом значение $x_j^{(i)}$, находящееся на пересечении i-й строки и j-го столбца соответствует наблюдению j-й переменной для i-го объекта: $x_j^{(i)} = X_j(o^{(i)})$.

Некоторое подмножество задач автоматической группировки можно рассматривать с точки зрения задачи размещения объектов. В задаче кластеризации на основе центроидов цель состоит в том, чтобы разделить N точек данных на классы, часто называемые цветами [22], так, чтобы точки одного цвета были близки друг к другу (или, что эквивалентно, так, чтобы точки разного цвета находились далеко друг от друга). Задачи размещения объектов формулируются с точки зрения задачи целочисленного линейного программирования следующим образом [23].

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} d_i y_{ij} + \sum_{j=1}^{n} f_j x_j,$$

где

$$\sum_{j=1}^{n} y_{ij} = 1,\ i = 1, \dots, M,$$

$$\sum_{i=1}^{m} d_i y_{ij} \leq u_j x_j,\ j = 1, \dots, N,$$

$$y_{ij} \geq 0, i = 1, \dots, m,\ j = 1, \dots, N,$$

$$x_i \in \{0,1\},\ i = 1, \dots, M.$$

В приведенном выше определении задачи размещения объектов n – количество объектов, u_j - емкость каждого объекта, m – количество заказчиков, d_i – спрос каждого заказчика, f_j – стоимость размещения, c_{ij} – цена перемещения одной единицы товара из объекта j заказчику i. Переменная x_j принимает значение 1, если j-й объект задействован, и 0, если нет, y_{ij} – доля удовлетворенного спроса.

Конкретный выбор целевой функции f_j приводит к различным вариантам задачи размещения и, следовательно, к различным вариантам задачи кластеризации на основе центроидов. Например, можно выбрать минимизацию суммы расстояний от каждой точки до каждого из центроидов (задача Вебера), или можно выбрать минимизацию максимума всех таких расстояний (задача 1 центра).

А. Вебером была сформулирована задача нахождения оптимального размещения промышленного предприятия. В задаче требуется найти точку на плоскости, которая минимизируют сумму цен перевозок из этой точки в N точек потребления, где для разных точек потребления назначается своя цена перевозки на единицу расстояния. Задача Вебера обобщает поиск геометрической медианы, для которой цены перевозок полагаются равными для всех точек потребления, и задачу нахождения точки Ферма, геометрической медианы трех точек. Строго определить задачу Ферма-Вебера можно следующим образом [24]. Дан набор из n точек в d-мерном пространстве $a^{(1)}, \ldots, a^{(n)} \in \mathbb{R}^d$, необходимо найти такую точку $x_* \in \mathbb{R}^d$, которая минимизирует сумму евклидовых расстояний до них:

$$x_* \in \arg\min_{x \in \mathbb{R}^d} f(x), \text{ где } f(x) \stackrel{\text{def}}{=} \sum_{i \in [n]} \left\| x - a^{(i)} \right\|_2$$

Однако, для задачи нахождения геометрической медианы в пространстве с Евклидовыми расстояниями не существует детерминированного решения, возможны лишь численные приближения к нему [25].

Одной из наиболее известных моделей автоматической группировки является модель *k*-средних [26-31], которая была предложена Г. Штейнгаузом [32].

$$\operatorname{argmin} F(X_1,\ldots,X_k) = \sum \min_{j \in \{1,k\}} \left\| X_j - A_i \right\|^2. \qquad (1.1)$$

Целью задачи *k*-средних является нахождение k точек (центров, центроидов) $X_1,\ldots X_k$ в d-мерном пространстве, таких, чтобы сумма квадратов расстояний от известных точек (векторов данных) $A_1,\ldots,A_N$ до ближайшей из искомых точек достигала минимума.

1.2 Известные методы решения задач автоматической классификации объектов

Оптимизационную модель (1.1) можно рассматривать как задачу размещения (требуется разместить оптимальным образом точки X_j). Теория размещения длительное время развивалась отдельно от кластерного анализа, решая при этом очень близкие или полностью идентичные задачи.

Э. Вайсфельд предложил итеративную процедуру для решения задачи Вебера, – одной из простейших задач размещения, основанную на итерационном взвешенном методе наименьших квадратов [33]. Этот алгоритм определяет набор весов, которые обратно пропорциональны расстояниям от текущей оценки до точек выборки, и создает новую оценку, которая является средневзвешенным значением выборки в соответствии с этими весами. Алгоритм Вайсфельда не всегда сходится и работает очень медленно. Многими исследователями разработаны модификации алгоритма Вайсфельда [34-39]. Г. Куном и Р. Куэном [40] предложен итеративный метод, обобщающий алгоритм Вайсфельда для невзвешенной задачи.

Методы локального поиска для решения задач размещения используются в [41-43]. Стандартный алгоритм локального спуска начинает работу с некоторого начального решения $S = \{X_1,\ldots,X_k\}$, выбранного случайно или с помощью какого-либо вспомогательного алгоритма. На каждом шаге локального спуска происходит переход от текущего решения к соседнему решению с меньшим значением целевой функции до тех пор, пока не будет достигнут локальный оптимум. Ключевой задачей является нахождение множества соседних решений $n(S)$. Элементы этого набора формируются путем применения определенной процедуры к решению S. На каждом шаге локального поиска функция соседства $n(S)$ задает набор возможных направлений поиска. Функции соседства могут быть самыми разнообразными, и отношение соседства не всегда симметрично. Алгоритмы локального поиска широко применяются для решения NP-трудных задач дискретной оптимизации. Однако, простой локальный спуск не позволяет

находить глобальный оптимум задачи. Свое дальнейшее развитие методы локального поиска получили в так называемых метаэвристиках, в частности, в алгоритме поиска с чередующимися окрестностями (variable neighborhoods search, VNS), предложенном Н. Младеновичем и П. Хансеном [44-47]. Её основная идея состоит в систематическом изменении функции окрестности и соответствующем изменении ландшафта в ходе локального поиска. Зависимость результата локального поиска от выбора окрестности уменьшается, если мы используем определенный набор окрестностей и чередуем их.

Метод чередующихся окрестностей может быть реализован одним из трех способов: детерминированным, вероятностным или смешанным, сочетающим в себе два предыдущих. Детерминированный локальный спуск с чередующимися окрестностями (variable neighborhood descent, VND) [48] предполагает фиксированный порядок смены окрестностей и поиск локального минимума относительно каждой из них. Вероятностный локальный спуск с чередующимися окрестностями (reduced VNS, RVNS) [46] получается из предыдущего метода при случайном выборе точек из окрестности $N_k(x)$. Этап поиска наилучшей точки в окрестности опускается. Алгоритм RVNS наиболее продуктивен при решении задач большой размерности, когда применение детерминированного варианта требует слишком много машинного времени для выполнения одной итерации. Однако, в случае задач большой размерности сложность выполнения одной итерации становится весьма большой, и требуются новые подходы для разработки эффективных методов локального поиска.

Основная схема локального поиска с чередующимися окрестностями (VNS) является комбинацией двух предыдущих вариантов (VND и RVNS) [46].

Алгоритм VNS (Variable Neighborhoods Search)

Шаг 1. Выбрать окрестности O_k, $k=1,..k_{max}$, и начальную точку x;

Шаг 2. Установить k←1;

ПОКА $k \leq k_{max}$

Шаг 3. Случайным образом выбрать точку $x' \in O_k(x)$;

Шаг 4. Применить локальный спуск с начальной точки x', не меняя координат, по которым x и x' совпадают. Полученный локальный оптимум обозначается x";

Шаг 5. ЕСЛИ $F(x'') < F(x)$

ТОГДА $x \leftarrow x''$, $k \leftarrow 1$,

ИНАЧЕ $k \leftarrow k + 1$.

КОНЕЦ ПОКА

Алгоритм VNS с ориентацией на исследование удаленных частей допустимой области (skewed VNS, SVNS) [49]. Эта модификация основной схемы особенно полезна в тех случаях, когда наилучшее решение уже найдено в некоторой достаточно большой окрестности текущего решения. В такой ситуации для успешного продолжения поиска необходимо найти новые перспективные части допустимой области. Обычно для этих целей используют простую процедуру старта с новой, например, случайной точки.

Алгоритмы VNS с декомпозицией на стадии локального поиска (Variable Neighbourhood Decomposition Search, VNDS) [50] подразумевают модификацию основной схемы алгоритма. Первый уровень оптимизации алгоритма связан с выбором окрестности, а второй — с локальным поиском относительно выбранной окрестности. Алгоритм VNDS можно рассматривать как один из способов встраивания в основную схему VNS классических приближенных методов, которые используются в комбинаторной оптимизации.

Параллельный VNS (parallel VNS, PVNS) [51] — еще один способ повышения эффективности основной схемы. Используют и гибридные конструкции VNS с другими метазвристиками: вероятностными жадными алгоритмами с адаптацией (Greedy Randomized Adaptive Search Procedure, GRASP) [52], поиском с запретами (Tabu Search) [53], генетическими алгоритмами и др.

Задача k-среднии была алгоритмически реализована С. Ллойдом [54]. Для вектора наблюдений X алгоритм k-средних предназначен для определения k центров и назначения точек данных (объектов) каждому центру для

формирования кластеров C_j, $j = 1..k$, при этом сводя к минимуму различие объектов внутри кластера. В работах Дж. Маккуина [55] базовый алгоритм k-средних, известный также как алгоритм Ллойда, состоит из итеративного повторения двух шагов:

Дано: k исходных кластерных центров (центроидов).

Шаг 1. Создание нового кластера C_j, назначением каждой точки данных ближайшему центру кластера (центроиду).

Шаг 2. Вычисление новых кластерных центров.

Повторение шагов 1 и 2, пока внутри каждого кластера изменения не прекратятся.

В алгоритме k-средних необходимо первоначально спрогнозировать количество групп (подмножеств). Кроме того, полученный результат зависит от начального выбора центров.

В работах М.Г. Садовского описан метод динамических ядер [56, 57], которых состоит из повторяющихся шагов. Метод является итерационной процедурой, каждая итерация состоит из двух шагов. Первоначально все объекты некоторого множества *F* разбиваются на *k* классов. Для каждого класса определяется ядро. На первом шаге для всех объектов и для каждого ядра вычисляется расстояние от объекта до каждого из ядер, таким образом происходит разбиение объектов множества *F* при фиксированных значениях ядер. На втором шаге принадлежность каждого объекта переопределяется: объект переносится в тот класс, чье ядро к объекту ближе всего. Шаги повторяются до тех пор, пока объекты не перестанут менять свою принадлежность к классу.

В работе Л. Кауфмана и П. Дж. Руссива представлена близкая к k-средних модель k-медоид (PAM, Partitioning Around Medoids) [58]. В отличие от алгоритма k-средних, в качестве центров выступают кластеризуемые объекты (медоиды), входящие в состав исследуемого множества. В Алгоритме PAM первоначально прогнозируется количество групп *k*, как и в алгоритме k-средних. Предварительно устанавливаем *k* объектов в качестве медоидов. На шаге 1 создаем новые кластеры, назначая каждую точку данных ближайшему медоиду. На шаге 2

формируем новые кластерные центры (медоиды), заменяя медоид на объект из множества, который улучшает (снижает) значение целевой функции. Повторение шагов 1 и 2 происходит, пока внутри каждого кластера изменения не прекратятся. Алгоритм более устойчив к выбросам и шумам, чем алгоритм k-средних, однако, неэффективен в применении к большим наборам данных из-за временной сложности. Кроме того, в алгоритме PAM вся матрица различий рассчитывается одновременно, что обеспечивает пространственную сложность алгоритма.

Алгоритм k-медиан [59, 60] является вариацией алгоритма k-средних, где для определения центроида кластера вместо среднего вычисляется медиана. Алгоритм k-медиан использует l_1-норму, в отличие от алгоритма k-средних, где используется квадрат l_2-нормы.

Л. Кауфманом и П. Дж. Руссивом предложен алгоритм CLARA (Clustering Large Applications, [61]), основанный на алгоритме PAM, для кластеризации объектов в больших базах данных с целью сокращения времени вычислений. Идея алгоритма заключается в следующем. Алгоритм PAM применяется к случайно выбранным объектам из множества для определения набора медоидов. Для выбора оптимального набора медоидов алгоритм запускается многократно. Результат работы алгоритма зависит от первоначального набора объектов. Это приводит к хорошим показателям качества кластеризации на тестовой выборке, но может оказаться ошибочным в применении ко всему набору данных. К недостаткам алгоритма можно отнести то, что его производительность быстро падает ниже приемлемого уровня с увеличением количества кластеров и размера выборки. Кроме того, он неустойчив к смещению выборки и начальному набору данных

Авторами работы [62] предлагается алгоритм Clustering Large Applications based upon Randomized Search, CLARANS, который направлен на использование рандомизированного поиска для облегчения кластеризации большого количества объектов.

Алгоритмы k-средних, k-медиан, k-медоид, PAM, CLARA, CLARANS относятся к классу алгоритмов кластеризации, основанных на методах разбиения (Partitioning clustering).

Нечеткие методы решения задач АГ (Fuzzy clustering) предложены Ш. Тамурой, С. Хигучи, Х. Танакой, К. Танакой, Дж. Ватадой, К. Асаи, Н.Ю. Музыченко, М.О. Ройбу. Алгоритм Fuzzy C-means (FCM) [63] позволяет разбить имеющееся множество элементов мощностью N на заданное число нечётких множеств *k*. Метод нечеткой кластеризации С-средних можно рассматривать как усовершенствованный метод k-средних, при котором для каждого элемента из рассматриваемого множества рассчитывается степень его принадлежности каждому из кластеров. Метод нечеткой кластеризации С-средних имеет ограниченное применение из-за существенного недостатка — невозможность корректного разбиения на кластеры, в случае, когда кластеры имеют различную дисперсию по различным осям элементов (например, кластер имеет форму эллипса).

Методы иерархической кластеризации (Hierarchical clustering) используют в своей основе идею последовательной иерархической декомпозиции множества объектов. К методам данной группы можно отнести алгоритм BIRCH (balanced iterative reducing and clustering using hierarchies) [64]. Алгоритм основан на следующих двух этапах. В ходе первого этапа формируется предварительный набор кластеров. На втором этапе к выявленным кластерам применяются другие алгоритмы кластеризации - пригодные для работы в оперативной памяти. В силу того, что метод является инкрементным и не требует наличия полного набора данных в один момент, он хорошо подходит для работы с большими массивами данных. Основным недостатком алгоритма является то, что он требует задания некоторых порогов плотности точек, а это не всегда приемлемо. Алгоритм работает только на числовых данных и чувствителен к форме и размерам кластеров (выделяет только кластеры выпуклой или сферической формы), а также, к порядку данных.

К классу алгоритмов, основанных на плотности (Density-based clustering) относятся, например, алгоритмы OPTICS и DBSCAN. DBSCAN (Density-based spatial clustering of applications with noise) [65] предложен М. Эстером, Г Кригелем, Ё. Сандером и С. Су и основан на плотности набора точек в некотором пространстве. Алгоритм группирует вместе точки, которые тесно расположены (точки со многими близкими соседями), помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко).

Алгоритм OPTICS (Ordering Points to Identify the Clustering Structure) [66], основан на алгоритме DBSCAN, однако, избегает проблемы обнаружения содержательных кластеров в данных, имеющих различные плотности. Точки данных упорядочиваются таким образом, что пространственно близкие точки становятся соседними. Кроме того, для каждой точки запоминается специальное расстояние, представляющее плотность, которую следует принять для кластера, чтобы точки принадлежали одному кластеру.

Задачи автоматической группировки на основе разделения смесей вероятностных распределений исследованы С. Ньюкомбом, К. Пирсоном, Б. Эвериттом, Д.М. Титтерингтоном, Дж. Маклаланом, В.Ю. Королевым, Дж. Гримом, О.К. Исаенко, В.Ю. Урбахом, А.П. Демстером, Н.М. Лейрда, Д.Б. Рубиным. Д.В. Сташков распространил подход Антамошкина – Казаковцева на такие задачи.

Характеристики объектов некоторого множества (смеси) являются случайной величиной, распределенной по некоторому известному закону распределения, например, гауссовскому распределению. Результатом решения разбиения смеси (исследуемого множества) является объединение объектов множества в группы (подмножества) таким образом, чтобы их характеристики соответствовали одному известному закону распределения.

Одним из самых популярных алгоритмов в этом классе является EM-алгоритм, используемый в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей, в случае,

когда модель зависит от некоторых скрытых переменных [67-71]. Каждая итерация алгоритма состоит из двух шагов. На Е-шаге (expectation) вычисляется ожидаемое значение функции правдоподобия, при этом скрытые переменные рассматриваются как наблюдаемые. На М-шаге (maximization) вычисляется оценка максимального правдоподобия, таким образом увеличивается ожидаемое правдоподобие, вычисляемое на Е-шаге. Затем это значение используется для Е-шага на следующей итерации. Алгоритм выполняется до сходимости [67]. Классический ЕМ-алгоритм относится к так называемым «жадным» алгоритмам, то есть он бросается на первый попавшийся локальный максимум, который может сильно отличаться от глобального.

Данный алгоритм неустойчив по начальным данным, так как он находит локальный экстремум, значение которого может оказаться гораздо ниже, чем глобальный максимум [69-71]. В зависимости от выбора начального приближения алгоритм может сходиться к разным точкам. Также может сильно варьироваться скорость сходимости. Вдобавок к этому, алгоритм не позволяет определять количество компонент смеси. Эта величина является структурным параметром алгоритма.

Для устранения недостатков были разработаны различные модификации алгоритма (медианные модификации, SEM-алгоритм, CEM-алгоритм, MCEM и SAEM-алгоритмы) [67, 69,71].

Дрезнер и другие авторы работы [72] предложили новые эвристические процедуры, включая генетический алгоритм (ГА), для решения задачи p-медианы на небольшом наборе данных. Другие авторы предлагают подходы, основанные на уменьшении количества данных [73]: упрощение задачи путем случайного (или детерминированного) выбора некоторой части исходного набора данных и использования этих результатов в качестве начального решения алгоритма k-средних на полном наборе данных [59, 74–76]. Такие подходы к агрегированию, а также уменьшение количества векторов данных [77] позволяют решать крупномасштабные задачи (до нескольких миллионов векторов данных) в разумные сроки. Однако такие подходы приводят к снижению точности.

Диссертационная работа направлена на получение наиболее точных решений, мы рассматриваем только методы, которые оценивают целевую функцию (1.1) напрямую, без использования методов агрегирования или аппроксимации.

Современная литература предлагает множество эвристических подходов [78,79] к установке начальных центроидов для алгоритма k-средних, которые в основном представляют собой различные эволюционные и случайные методы поиска. Алгоритмы локального поиска и рандомизированные алгоритмы на их основе представлены в большом количестве публикаций. Например, алгоритмы Variable Neighborhood Search (VNS) [46,80,81] или алгоритмы агломерации [82, 83] иногда показывают хорошие результаты. Процедуры инициализации для алгоритмов локального поиска также широко представлены, включая случайное заполнение и оценку распределения точек спроса [79]. Однако во многих случаях даже многократные запуски алгоритмов простого локального поиска из различных случайно сгенерированных решений не обеспечивают решения проблемы, близкой к глобальному оптимуму. Более сложные алгоритмы [21] позволяют получить значения целевой функции (1.1) во много раз лучше, чем методы локального поиска [81].

Популярной идеей является использование генетических алгоритмов и других эволюционных подходов для улучшения результатов локального поиска [84–87]. Такие алгоритмы объединяют локальные минимумы, полученные с помощью алгоритма k-средних. ГА оперируют определенным набором (популяцией) решений-кандидатов и включают в себя специальные генетические операторы (алгоритмы) инициализации отбора, скрещивания и мутации. Оператор мутации случайным образом изменяет полученные решения и обеспечивает некоторое разнообразие в популяции. Тем не менее, в таких алгоритмах с увеличением количества итераций популяция вырождается в определенный набор решений, близких друг к другу. Более крупные популяции, а также динамично растущие популяции улучшают эту ситуацию. Однако более простые алгоритмы, основанные на использовании тех же жадных агломерационных процедур [81,88], часто показывают лучшие результаты при том же времени вычислений.

В Главе 3 не учитываются важные вопросы в области кластерного анализа, такие как адекватность модели (1.1) и соответствие результатов алгоритма фактическому (реальному) разделению объектов, если таковое известно. Работа нацелена на точность и стабильность полученного значения целевой функции (1.1) в рамках модели k-средних. Когда цена ошибки высока [73], а также при сравнительной оценке точности алгоритма по сравнению с определенным стандартным решением (необязательно глобально оптимальным), необходимо получить результат, который было бы трудно улучшить другими известными методами без значительного увеличения времени расчета. Развитие массивных систем параллельной обработки, таких как графические процессоры (GPU), делает многократный запуск адаптированных версий алгоритмов локального поиска чрезвычайно дешевым. В этом случае крупномасштабные задачи (до нескольких миллионов векторов данных) могут быть решены с использованием самых современных алгоритмов, обеспечивающих высочайшую точность.

1.3 Эволюционные алгоритмы и жадные эвристические процедуры

В области кластерного анализа существует проблема поиска глобального экстремума у критерия качества группировки. Критерий качества является функцией, обладающей множеством локальных экстремумов [21]. Для получения решения необходимо решить сложную комбинаторную задачу поиска оптимального варианта группировки объектов. Возможно получить решение такой задачи полным перебором: число различных вариантов разбиения N объектов на k групп. Вычислительная сложность алгоритма полного перебора экспоненциально зависит от объема данных в задаче:

$$M(N,k) = \frac{\sum_{i=1}^{k}(-1)^i\binom{k}{i}(k-i)^N}{k!}.$$

Классические алгоритмы кластерного анализа используют ограничения, например, по числу и форме получаемых групп, выполняют направленный поиск в относительно небольшом подмножестве пространства возможных решений, при

этом не гарантируя нахождение глобального экстремума. Для поиска оптимального решения может применяться мультистарт данных методов с рандомизированными процедурами выбора начальных решений и/или рандомизированной процедурой локального поиска [89, 90] или более сложные подходы [21], многие из которых основаны на идеях, позаимствованных из живой природы. К таковым можно отнести генетические (эволюционные) алгоритмы [91], нейронные сети [92], методы имитации отжига [93] и т.д. В работе «Выбор эффективной системы зависимых признаков» [94] предложен метод случайного поиска с адаптацией, который является одним из первых в эволюционных алгоритмах. Эволюционные алгоритмы используют принципы и терминологию, связанную с природной эволюцией, например, популяция, хромосома, ген, аллель, процедура селекции, процедура рекомбинации (скрещивания, кроссинговера) и процедура мутации. Основные понятия приведены в Главе 3. Генетический алгоритм, предложенный в [95], имитирует адаптацию популяции решения к некому аналогу окружающей среды, причем алгоритм вводит специальную меру приспособленности (fitness function, иначе – функция полезности), которая, как правило, определяется на основе целевой функции задачи. По умолчанию мы в данном исследовании будем считать понятия «целевая функция» и «функция приспособленности» эквивалентными. Процесс работы алгоритма представляет собой последовательную смену популяций, состоящих из N_{POP} решений [96]. Отметим, что популяция может сменяться целиком или частично, размер популяции может оставаться неизменным либо меняться динамически по ходу работы алгоритма [97,98].

Размер популяции (N_{POP}) является важным параметром эволюционного алгоритма. Ограниченная по размеру популяция при отсутствии процедуры мутации (либо при незначительном влиянии мутации) приводит к тому, что популяция «вырождается», то есть алгоритм начинает генерировать одни и те же решения [99]. В [100] авторы указывают, что наилучшие результаты, как правило, достигаются при приблизительно одинаковом числе сменяющихся поколений решений и числе самих решений. При вычислительно сложной процедуре

скрещивания сложно предсказать время ее выполнения, и, следовательно, сложно предсказать число сменяющихся поколений за установленное время. В [96] исследована зависимость времени достижения локального оптимума от размера популяции и размера турнира (размер турнира s – число решений, среди которых выбираются лучшие в качестве родительских решений, порождающих решение следующего поколения). Если h – число неоптимальных решений в окрестности, L – минимальная вероятность достижения локального оптимума в очередном поколении, E – вероятность того, что хотя бы одно решение - потомок будет по значению целевой функции не хуже родительской, r – доля решений популяции, участвующих в турнирной селекции, то размер популяции и размер турнира определяются как

$$N_{POP} = 2\left\lceil \frac{1+\ln h}{L \cdot E\left(\frac{1}{e^{2r}}\right)} \right\rceil, s = \lceil r N_{POP} \rceil,$$

где $\lceil \cdot \rceil$ - округление вверх.

Для оценки размера популяции N_{POP} требуется решения сложных задач по определению значений *h, L, E. B* [96] оценки этих значений даны для простых процедур скрещивания. Если же в процедуру кроссинговера включен локальный поиск, то выбор любых двух родительских «особей» дает в результате локальный оптимум в качестве потомка, то есть для достижения локального оптимума достаточно N_{POP}=2, что ни в коей мере не отражает размер популяции, позволяющей получить решения, близкие к глобальному оптимуму. Поскольку оптимальный размер популяции, необходимый для эффективной работы эволюционного алгоритма, установлен лишь для генетических алгоритмов с простыми способами рекомбинации, в настоящей работе оценка оптимального размера популяции выполнялась исключительно экспериментальным путем.

Генетический алгоритм, как правило, рассматривают в качестве стратегии глобального поиска. Более того, для многих версий генетического алгоритма характерен следующий ключевой момент [101]: в то время, как классические методы локального поиска легко находят собственно локальные оптимумы задачи, генетический алгоритм часто предъявляет в качестве ответа глобальный

оптимум. Само понятие глобального оптимума задачи конструктивным не является [101]: даже получив его, трудно проверить и доказать его глобальную оптимальность.

Основная отличительная особенность жадных эвристических методов состоит в том, что они, являясь методами, последовательно улучшающими известный результат (методами локального поиска), в некоторой окрестности известного решения, выбирают в качестве следующего решения тот вариант, который дает наибольший прирост целевой функции (наибольшее уменьшение значения – в случае минимизации).

Эволюционные алгоритмы, в том числе генетические алгоритмы, представляют лишь общую схему организации поиска, реализация которой зависит от выбора процедур селекции, мутации и особенно процедуры скрещивания, которая изначально задумывалась как простая рандомизированная процедура [95], а затем была адаптирована под решение конкретных задач [102]. Практика решения многих NP-трудных задач показывает эффективность перехода от рандомизированных процедур рекомбинации к поиску наилучшего способа рекомбинации [103, 104]. В [103] рассматривается задача построения наилучшей хромосомы - потомка от двух известных хромосом - родителей. При этом предусмотрено кодирование хромосом виде последовательностей нулей и единиц, причем при совпадении значения цифры в родительских последовательностях она наследуется хромосоме - потомку, при несовпадении решается задача выбора такого значения несовпадающей цифры, при котором потомок имеет наилучшее значение целевой функции. Отметим, что наилучшее значение целевой функции у потомка в первом поколении не гарантирует наилучших значений в последующих поколениях – в данном случае алгоритм выбора наилучшего потомка в первом поколении тоже является в некотором смысле жадным. Но и такая ограниченная задача построения наилучшего потомка от двух известных родительских решений оказалась NP-трудной, например, применительно к p-медианной задаче (при этом для других NP-трудных задач оптимальный потомок может быть найден за полиномиальное время). Это позволяет сделать предположение о том, что

оптимальный выбор процедуры скрещивания может быть задачей не менее трудной, чем собственно нахождение оптимального решения задачи оптимизации. Различные методы локального поиска, в том числе некоторые жадные алгоритмы, для дискретных задач размещения (в том числе для задачи о p-медиане) исследованы в диссертации Ю.А. Кочетова [101]. Отмечается, что рандомизированный характер поиска позволяет сократить погрешность результата. Эффективность конкретных методов локального поиска и применяемой окрестности сильно зависит от задачи. Тем не менее, поиск в простых окрестностях, как правило, достаточно эффективен для большинства практических дискретных задач автоматической группировки.

Для задач автоматической группировки на основе моделей теории размещения А.Н. Антамошкиным и Л.А. Казаковцевым предложен метод жадных эвристик [105, 106]. В большенстве случаев алгоритмы этого метода рандомизированы, но при этом получаемые результаты стабильны. В методе жадных эвристик используются эволюционные алгоритмы как один из способов организации глобального поиска, в том числе подходы и Красноярской школы эволюционных алгоритмов [107,108]. Для задач автоматической группировки при разделении множества объектов на k групп метод жадных эвристик можно представить в виде трех вложенных циклов [69]:

ЦИКЛ 1. Цикл, осуществляющий стратегию глобального поиска, генерирующий промежуточные решения, представленные множествами, мощность которых выше значения *k*.

ЦИКЛ 2. Выполнение жадной эвристической процедуры, в ходе которой могут запускаться алгоритмы локального поиска, улучшающие промежуточные решения, мощность которых приведена к значению *k*.

ЦИКЛ 3. Цикл локального поиска, предусматривающий оценку последствий исключения элементов из промежуточного решения.

1.4 Меры сходства и критерии оценивания классификации объектов

В задачах автоматической группировки ключевым понятием является понятие метрики объектов. Под метрикой, как правило, понимают функцию или формулу, определяющую расстояние между любыми точками и классами в метрическом пространстве R^p [109, 110]. Метрическое пространство есть множество точек с функцией расстояния $d(x_i,y_i)$. Расстояние порядка p между двумя точками определяется по функции Минковского (l_p-норма) [111]:

$$d_p(x,y) = \left(\sum_{i=1}^{M} |x_i - y_i|^p\right)^{\frac{1}{p}},$$

где x, y – векторы значений параметров, M – размерность вектора.

Параметр p, определяется исследователем, с его помощью можно прогрессивно увеличить или уменьшить вес i-й переменной. Частные случаи функции Минковского зависят от значения параметра p.

При p=2 функция вычисляет Евклидово расстояние между двумя точками $x=(x_1,\ldots,x_M)^T$ и $y=(y_1,\ldots,y_M)^T$ [110,111] (l_2-норма).

$$d_2(x,y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2},$$

где M– размерность вектора.

Квадрат евклидова расстояния определяется:

$$d_E(x,y) = \sum_{i=1}^{M} (x_i - y_i)^2.$$

При $p=1$ функция вычисляет манхэттенское расстояние [111], также называемое расстоянием городских кварталов или прямоугольным (l_1-норма):

$$d_1(x,y) = \sum_{i=1}^{M} |x_i - y_i|,$$

где x, y – векторы значений параметров, M – размерность вектора.

При $p=\infty$ функция вычисляет расстояние Чебышева или расстояние шахматной доски, возвращая наибольшее значение разности между параметрами объекта по модулю:

$$d(x,y) = max|x_i - y_i|,$$

где *x*, *y* – векторы значений параметров, *M* – размерность вектора.

При *p=0* функция вычисляет Расстояние Хэмминга. В этом случае функция рассматривается как мера различия векторов, параметры которых задаются значениями 0 или 1 [112]:

$$d(x,y) = \sum_{i=1}^{M} |x_i - y_i|,$$

где *x*, *y* – бинарные вектора размерности *M*.

В федеральном стандарте 1037C (Телекоммуникации: глоссарий телекоммуникационных терминов) расстояние Хэмминга определяется как количество позиций цифр, в которых соответствующие цифры двух двоичных слов одинаковой длины различаются [113]. Данная мера возвращает число несовпадений значений соответствующих параметров.

Для частных случаев функции Минковского, зависящих от параметра *p*, справедливы (доказательны) следующие утверждения: при $p \geq 1$ и $p=\infty$ расстояние является метрикой. При $p<1$ расстояние не является метрикой.

Кроме случаев зависимости вариантов функции от параметра *p* существуют и другие способы вычисления расстояний, например, такие как:

Расстояние Махаланобиса [110, 112, 114]. Расстояние Махаланобиса можно определить как меру несходства (различия) между векторами из одного распределения вероятностей с матрицей ковариации *C*. Расстоянием Махаланобиса между двумя точками $x=(x_1,\ldots,x_M)^T$ и $y=(y_1,\ldots,y_M)^T$ называется функция вида

$$d_M(x,y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^T \mathrm{C}^{-1} (x_i - y_i)},$$

где C – матрица ковариации. M – размерность вектора. Если матрица ковариации является единичной, то расстояние Махалонобиса становится равным Евклидову расстоянию [115,116]. Если матрица является диагональной, то расстояние Махаланобиса называют нормализованным расстоянием Евклида. Матрица ковариации определяется как:

$$\mathrm{C} = cov(x, y) = \mu\big[\big(\mathrm{x} - \mu(x)\big)\big(y - \mu(y)\big)\big],$$

где μ – математическое ожидание.

Косинусное расстояние [116]. Меру сходства можно оценить через косинус между двумя векторами:

$$d(x, y) = 1 - arcco\,s\left(\frac{\sum_{i=1}^{M} x_i y_i}{\sqrt{\sum_{i=1}^{M} x_i^2}\sqrt{\sum_{i=1}^{M} y_{i=1}^2}}\right),$$

где x, y – векторы размерности M.

Если угол между векторами равен 0, то векторы полностью совпадают, косинус равен 1.

Наиболее популярным является евклидово расстояние или квадрат евклидова расстояния, второй по популярности является манхэттенское расстояние. В большенстве случаев в литературе представлены задачи именно с этими типами метрики Минковского.

Для проверки адекватности моделей и алгоритмов АГ объектов проанализированы различные критерии оценки качества кластеризации и классификации. В свою очередь критерии оценки подразделяются на внутренние, внешние. К внешним относятся критерии, которые при оценке качества учитывают предварительную информацию о группировке объектов. Такие метрики используют, как правило, для задач автоматической группировки. К внутренним критериям относятся критерии оценки, которые опираются на информацию, полученную только из множества данных.

В литературных источниках представлен широкий ряд критериев оценки качества кластеризации для случаев, когда отсутствует априорная информация о разделении множества на подмножества, например, такие как: индекс Калински –

Харабаш [117,118], индекс Данна [117,119], индекс Девиса – Болдуина [117,120], индексы действительности CS [121] и PS [117,122], на основе индексов Данна и Девиса – Болдуина, индекс Кржановски-Лая [123], индекс SD [117, 124], индекс S_Dwb [117,125], индекс RMSSTD (Root – Mean – square standard deviation) [117, 126], индекс оценки силуэта [61,127,128]. А также освещены критерии оценки для случаев, когда известна предварительная оценка разделения множества на подмножетсва: индекс Rand [129], индекс Adjusted Rand [130,131], Hubert Γ statistic (модифицированная и его нормализованная версия) [117, 132], оценка точности (Precision). Рассмотрим некоторые их них.

Модифицированная Γ-статистика Гиберта (The modified Hubert Γ statistic):

$$\Gamma = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i,j)Q(i,j),$$

$$M = N\frac{(N-1)}{2},$$

где N – количество элементов в кластеризуемом множестве X, P – матрица близости, каждый элемент которой определяется, как расстояние до всех остальных объектов множества X, Q – матрица размерности $N \times N$, в которой элемент q_{ij} соответствует расстоянию между центрами кластеров, в которых находятся элементы x_i и x_j соответственно. Для матриц P и Q используют одну и ту же меру расстояния. Чем выше значение критерия Γ, тем качество кластеризации выше.

Нормализованная Hubert Γ statistic вычисляется как:

$$\tilde{\Gamma} = \frac{\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(P(i,j)-\mu_P)(Q(i,j)-\mu_Q)}{\sigma_P \sigma_Q},$$

где μ_P, μ_Q – среднее значение P и Q, σ_P, σ_Q – среднеквадратичное отклонение P,Q. Значение критерия находится в диапазоне от -1 до +1. Чем ближе $\tilde{\Gamma}$ стремится к +1, тем выше качество кластеризации.

Индекс оценки силуэта (Silhouette index) определяется как:

$$SWC = \frac{1}{N}\sum_{j=1}^{N} S_{x_j},$$

$$S_{x_j} = \frac{b_{pj}-a_{pj}}{\max(a_{pj},b_{pj})},$$

$$b_{pj} = min_{q \neq p} d_{qj},$$

где d_{qj} – среднее расстояние от x_j до объектов из другого кластера c_q ($q \neq p$), a_{pj} – среднее расстояние от x_j до других объектов из того же кластера c_p, при условии , что элемент x_j принадлежит кластеру c_p. Лучшая группировка характеризуется максимальным индексом оценки *SWC*, это достигается когда расстояние внутри кластера a_{pj} мало, а расстояние между элементами соседних кластеров b_{pj} велико.

Индекс Rand (Index Rand, *IR*). Индекс Rand оценивает, насколько много из тех пар элементов, которые находились в одном классе, и тех пар элементов, которые находились в разных классах, сохранили это состояние после кластеризации алгоритмом:

$$IR = \frac{TP+FN}{TP+TN+FP+FN},$$

где

TP – элементы принадлежат одному кластеру и одному классу;

TN – элементы принадлежат одному кластеру, но разным классам;

FP – элементы принадлежат разным кластерам, но одному классу;

FN – элементы принадлежат разным кластерам и разным классам.

Значение *IR* показывает, насколько совпадают кластеры с заданными классами, 1 демонстрирует полное совпадение, а 0 – его отсутствие.

Индекс Adjusted Rand (скорректированный индекс Rand).

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}},$$

где n_{ij}, b_j, a_i – данные из таблицы сопряженности. Значения индекса ARI находятся в диапазоне от -1 до +1. Чем значение ARI ближе к 1, тем точнее качество кластеризации. Множество A, состоит из *N* объектов, каждый объект предварительно принадлежит одному из кластеров $X=\{X_1,X_2,\ldots,X_k\}$, в свою очередь после применения одного из алгоритмов кластеризации к множеству A, получаем новое разделение объектов по кластерам $Y=\{Y_1,Y_2,\ldots,Y_l\}$. Совпадения

или различия априорной и полученной информации отражаются в таблице сопряженности размерностью $k \times l$. Строки таблицы соответсвуют значениям из X, а столбцы – значениям из Y. На пересечении строки и столбца указывается количество объектов n_{ij}, входящих как и в X_i, так и в Y_j. Сумма элементов i-й строки соответсвует a_i, а сумма элементов j столбца соответствует b_j, при этом $\sum_{i=1}^{k} a_i = N$ и $\sum_{j=1}^{l} b_j = N$. Общий вид и основные элементы таблицы сопряженности приведены в таблице 1.1.

Таблица 1.1 – Таблица сопряженности

	Y_1	Y_2	…	Y_l	$\sum$
X_1	n_{11}	n_{12}	…	n_{1l}	a_1
X_2	n_{21}	n_{22}	…	n_{2l}	a_2
…	…	…	…	…	…
X_k	n_{k1}	n_{k2}	…	n_{kl}	a_k
$\sum$	b_1	b_2	…	b_l	N

Точность (Precision). Под точностью понимаем долю верно классифицируемых объектов ко всем объектам множества.

$$Precision = \frac{TP}{TP+FP},$$

где

TP - элементы принадлежат одному кластеру и одному классу;

FP - элементы принадлежат разным кластерам, но одному классу.

Площадь под ROC – кривой (AUC (area under the curve) ROC-кривой (receiver operating characteristic)) [133]. ROC- кривая (кривая ошибок) отражает зависимость между долей истинно положительных примеров (чувствительность) и долей ложных положительных примеров (специфичность). AUC является показателем, который дает количественную интерпретацию ROC – кривой. В работе [134] приводится шкала оценок значений показателя AUC. Чем выше значение показателя AUC, тем точнее качество классификации.

Выводы по Главе 1

Анализ литературы показал, что известные методы решения задач автоматической классификации можно отнести к одному из двух классов. Методы первого класса нацелены на быстрое решение задач большой размерности (до нескольких миллионов векторов данных). Такие методы позволяют решить задачу в разумные сроки, но при этом приводят к снижению точности результата. Методы второго класса, применимые для решения задачи автоматической группировки на небольшом наборе данных, дают более точный результат, но при этом требуется значительные вычислительные затраты.

Во многих случаях даже многократные запуски простых алгоритмов локального поиска из различных случайно сгенерированных решений не обеспечивают решение задачи, близкое к глобальному оптимуму. В зависисмости от начального приближения алгоритм может сходиться к разным точкам, то есть алгоритм дает нестабильный результат, а также варьируется скорость сходимости.

При этом, для непрерывных задач размещения, к которым можно отнести k-средних, на сегодняший день разработаны алгоритмы лишь для наиболее распространенных мер расстояния, таких как евклидово и манхэттенское расстояния. Улучшить результат автоматической классификации объектов с повышенными требованиями к точности и стабильности результата известными алгоритмами на текущий момент крайне трудно без значительного увеличения временных затрат. При решении практических задач автоматической классификации объектов, например, при решении задач выделения однородных партий промышленной продукции, вопросы вызывает адекватность моделей и, как следствие, точность автоматической группировки промышленной продукции. Тем не менее, возможна разработка алгоритмов, обеспечивающих дальнейшее улучшение результата на основе выбранной модели, например, модели k-средних, чему посвящены главы 2 и 3 диссертационной работы.

ГЛАВА 2 ОПТИМИЗАЦИОННАЯ МОДЕЛЬ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ, ОСНОВАННОЙ НА МОДЕЛИ K-СРЕДНИХ

Глава посвящена проблеме понижения размерности данных с применением методов факторного анализа, а также разработке оптимизационной модели автоматической группировки объектов, основанной на модели k-средних, позволяющей повысить качество автоматической группировки (по индексу Рэнда).

При решении задач автоматической группировки промышленной продукции [135-138] на однородные партии по данным тестовых испытаний приходится иметь дело с данными достаточно большой размерности – до нескольких тысяч признаков для каждого экземпляра продукции. При этом часть таких признаков являются малоинформативными, и их включение в модель автоматической группировки лишь ухудшает точность решения задачи. Между некоторыми из характеристик зачастую присутствуют явные корреляционные зависимости (рисунок 2.1).

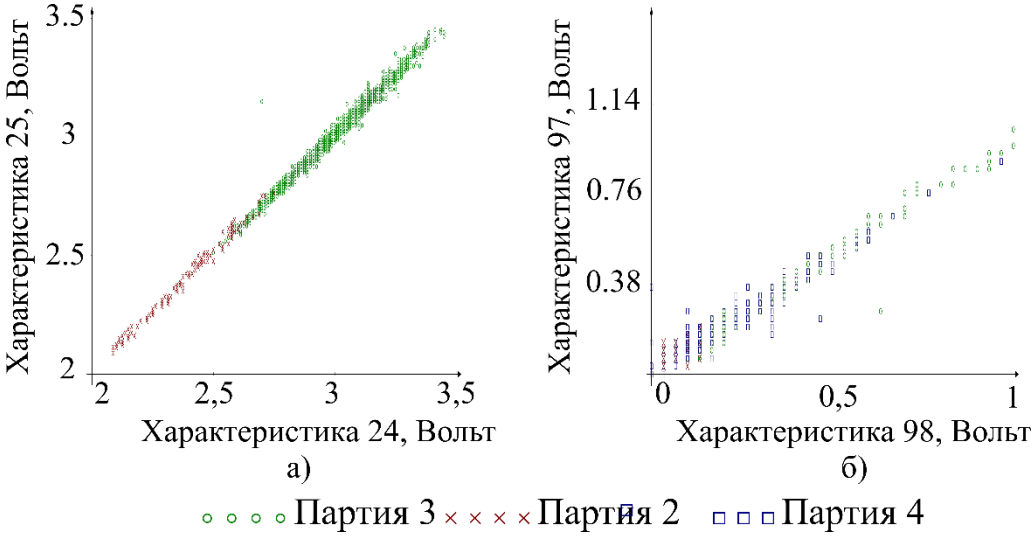


Рисунок 2.1 – Пример статистической зависимости между некоторыми характеристиками электрорадиоизделий [26]

Методы факторного анализа позволяют снизить размерность данных за счет использования этих зависимостей. Кроме того, такие зависимости могут быть учтены при решении задач автоматической группировки и иными способами.

Одним из таких способов является решение задачи автоматической группировки как задачи разделения смеси распределений [18,69,139,140]. При этом учет корреляционных зависимостей в таких моделях возможен лишь при наличии больших массивов данных, и решение задач требует многократно больших вычислительных затрат по сравнению, например, с задачей к-средних. При практическом решении задач приходится переходить к некоррелированным распределениям [18,69,141,142]. Альтернативным способом учета кореляционных зависимостей является переход в модели k-средних к особым мерам расстояния, в которых эти зависимости учтены. В данной главе рассмотрено как применение методов факторного анализа, так и переход в модели k-средних к расстоянию Махаланобиса, учитывающему зависимости между признаками группируемых объектов.)

2.1 Методы факторного анализа в задачах автоматической группировки

Изучая вопросы наследственности и антропометрии, Ф. Гальтон ввел понятия коэффициента корреляции, факторного анализа. Идея факторного анализа заключалась в следующем: если несколько измеряемых характеристик объекта имеют согласованную динамику, то существует фактор, который позволяет установить взаимосвязь между характеристиками, при этом сам фактор не имеет доступных прямых измерений [143]. К. Пирсон разработал более совершенные методы корреляционного и факторного анализа [144]. В своей статье «On Lines and Planes of Closest Fit Systems of Points in Space» К. Пирсон представил метод главных компонент [145], а Г. Хоттелинг впоследствии дал математическое обоснование методу. В свою очередь, Ч. Спирмен представил двухфакторную теорию [146]. Основная идея теории заключается в определении общего (генерального) g–фактора и дополнительного фактора. Генеральный g-фактор объяснял корреляцию между тестами, отвечающими за интеллектуальные возможности, дополнительный фактор выявлял характерные интеллектуальные возможности. Данная теория не оправдала себя на большой выборке тестов, а

также работает только с положительной корреляцией. Данные, которые не подходили под решение, исключались. К. Хользингер в своей бифакторной теории предложил определить групповые факторы, помимо генерального и дополнительного факторов [146]. В групповые факторы входили данные, которые могли быть исключены по теории Спирмена. Л. Тёрстоун опроверг теорию Спирмена о генеральном g-факторе. Он отказался от генерального фактора, что позволило выявлять сразу несколько групповых факторов. Это привело к многофакторному анализу [147]. Л. Тёрстоун применил матричную алгебру в факторном анализе, указав на то, что количество факторов соответствует рангу корреляционной матрицы [146]. Используя многофакторный анализ, Р. Кателл разработал 16-факторный личностный опросник [148]. Д. Гилфорд разработал кубическую модель интеллекта. Таким образом, их школы расширили область применения факторного анализа.

В развитие группового метода факторного анализа большой вклад внесли П. Хорст, Л. Гуттман, К. Хользингер и Л. Тёрстоун. Основная идея метода заключается в одновременном выделении факторов. Каждый фактор соответствует группе входных данных, которые наилучшим образом коррелируют между собой.

Предложенные подходы, в основе которых лежит математический подход, не могут определить точное количество факторов, необходимых для распределения их по переменным, оценить их нагрузки на переменные. Для решения данной проблемы предложен статистический подход. Д. Лоули предложил применить метод максимального правдоподобия для оценки нагрузки факторов, С. Рао на основе идей Д. Лоули представил метод канонического факторного анализа. Х. Кайзер и Дж. Каффри представили альфа-факторный анализ. Также большой вклад внес Р. Баргман. В своей работе он предложил критерий значимости простой структуры [146, 149], который позволяет определить необходимое количество факторов.

В настоящий момент факторный анализ имеет широкую область применения в здравоохранении, психологии, экономике, в сельском хозяйстве и т.д..

Например, в медицине факторный анализ используют для оценки спектральных показателей сердечного ритма пациента [150], в экономике при анализе эффективности использования трудовых ресурсов организации[151]. В сельском хозяйстве при оценке признаков урожайности озимой ржи [152]. В машиностроении для оптимизации размера заказываемых партий запасных частей станций техобслуживания [153]. В авиации для оценивания текущего уровня аварийности авиационных событий [154] и прогнозирования уровня безопасности полетов [155, 156]. В данных работах доказана практическая значимость факторного анализа.

Факторный анализ относится к методам многомерного статистического анализа информации. На практике используется четыре техники факторного анализа (R,Q,P,O – техники) [146]. В представленной работе используется R-техника, она позволяет определить связь между входными характеристиками объектов и полученными факторами.

Существует шесть типов факторного анализа [146]. Детерминированный факторный анализ изучает влияние факторов на результативный показатель. Данный показатель представляется произведением или суммой факторов, либо частными факторами. Связь между факторами и результативным показателем отслеживается четко и является определенной. Стохастический факторный анализ исследует влияние факторов на результативный показатель, связь между которыми носит вероятностный характер. Статический факторный анализ изучает влияние факторов на результативный показатель на конкретное событие или дату. В динамическом факторном анализе осуществляется оценка факторов, исследование причинно-следственных связей с результативным показателем в процессе изменения во времени. Ретроспективный факторный анализ изучает влияние факторов и их оценки на результативный показатель, изменяющийся в прошлом временном периоде. Перспективный факторный анализ рассматривает возможное влияние факторов на конечный результат в будущем.

Основная цель факторного анализа заключается в определении факторов, которые выявляют наличие взаимосвязи между входными характеристиками, то есть понижении размерности входных характеристик.

Задачей факторного анализа является нахождение структуры, которая бы достаточно точно отражала и воспроизводила реальные, существующие в природе зависимости [146]. Анализ основан на определении факторной модели [157]

$$X_i = \sum_{j=1}^{f} a_{ij} F_j + u_i, \quad (2.1)$$

где X_i – вектор значений i-ой характеристики ($i = 1..M$), F_j – первичные факторы ($j = 1..f$), a_{ij} – коэффициенты, называемые факторными нагрузками, u_i – характерные (специфические) факторы, описывающие ту часть характеристики, которая не входит ни в один фактор. При $f < M$ происходит понижение размерности исходной задачи.

В решении задач с помощью факторного анализа существует шесть ключевых проблем в определении простой структуры [146, 158].

Первая проблема связана с робастностью оценки исходных данных. В первоначальных исходных данных возможно присутствие грубых ошибок, которые могут повлиять на интерпретацию полученных результатов. Данная проблема решается методом робастного оценивания среднего значения и среднеквадратичного отклонения [158].

Исходные данные представляем матрицей размерности $N \times M$, где N – количество изучаемых объектов, M – количество характеристик объекта, элемент матрицы x_{ij} (i=1..N, j=1..M) – значение j-й характистики i-го объекта. Далее формируется матрица из нормированных значений исходной матрицы.

Существует несколько точек зрения о нормальном распределении характеристик. Х. Харман и А. Ренчер [146, 157, 159] в своих работах утверждают, что для построения линейной факторной модели необходимо, чтобы входные данные имели нормальное распределение и между переменными существовала линейная связь. Дж.-О. Ким опровергает данное утверждение [146, 160].

Нормирование данных является обязательным этапом в преобразовании исходных данных в задачах многомерного статистического анализа, особенно если данные различаются в единицах измерения. Существует различные способы нормирования данных [161], приведем примеры некоторых из них.

1. $a_{i,j} = \frac{a_{i,j} - \overline{a_j}}{\sigma_j}$, при $\sigma_j \neq 0$,

2. $a_{i,j} = \frac{a_{i,j}}{\overline{a_j}}$, при $\overline{a_j} \neq 0$,

3. $a_{i,j} = \frac{a_{i,j}}{a_j^э}$, при $a_j^э \neq 0$,

4. $a_{i,j} = \frac{a_{i,j}}{a_j^{max}}$, при $a_j^{max} \neq 0$,

5. $a_{i,j} = \frac{a_{i,j} - \overline{a_j}}{a_j^{max} - a_j^{min}}$, при $a_j^{max} - a_j^{min} \neq 0$,

где $a_{i,j}$ - значение характеристики до нормирования, $\overline{a_j}$ - среднее значение характеристики, σ_j – среднеквадратичное отклонение характеристики, a_j^{max} – максимальное значение характеристики, a_j^{min} – минимальное значение характеристики, $a_j^э$ – эталонное значение характеристики.

Представленные способы нормирования имеют существенный недостаток в задачах автоматической группировки изделий. Каждое изделие имеет большое количество входных характеристик. Характеристика, у которой разброс значений невелик, не должна оказывать влияние на группировку изделий. При реальной обработке данных это условие не выполняется. Авторами работы [16] предложен новый способ нормирования характеристик:

$$a_{i,j} = \frac{a_{i,j} - \overline{a_j}}{\delta_j^{max} - \delta_j^{min}}, \text{при } \delta_j^{max} - \delta_j^{min} \neq 0, \quad (2.2)$$

где $a_{i,j}$ - значение характеристики до нормирования, $\overline{a_j}$ - среднее значение характеристики, δ_j^{max},δ_j^{min} – нижняя и верхняя граница дрейфа характеристики соответственно. Под дрейфом понимается величина изменения характеристик изделий в ходе проведения дополнительного неразрушающего контроля, имитирующего жесткие условия эксплуатации. В [16] экспериментально показано, что этот способ нормирования дает разбиение по производственным партиям с гораздо меньшим количеством ошибок.

Вторая проблема связана с установлением оценок общности. Составляется матрица парных корреляций характеристик изучаемых объектов. Процесс

нахождения матрицы парных коэффициентов корреляции является первым этапом в факторном анализе. Если на главной диагонали матрицы стоят единицы, то процесс поиска факторов осуществляется с помощью компонентного анализа (метод главных компонент). Для факторного анализа используется корреляционная матрица, на главной диагонали которой используют оценки общностей (редуцированная корреляционная матрица).

На следующем шаге решаем саму задачу факторного анализа. Определяем матрицу факторных отображений *A*, ее элементы являются факторными нагрузками. Каждый фактор представляет собой столбец, а каждая переменная – строку. При геометрической интепретации факторами являются оси координат, положение векторов-переменных определяется значениями факторных нагрузок, являющихся координатами концов этих векторов.

Третья проблема состоит в определении факторов, это связано с тем, что существует большое количество матриц *A*, которые могут воспроизвести редуцированную корреляционную матрицу.

Отсюда формулируется четвертая проблема – вращение, которое позволяет определить конкретную матрицу *A*. Существует два основных метода вращения – ортогональное и косоугольное. Пятая проблемой заключается в оценке значений факторов. Шестая проблема связана с изучением динамики процессов.

Существует ряд методов и критериев для решения обозначенных проблем факторного анализа.

Критерии проверки значимости исходной корреляционной матрицы

Для оценки значимости исходной корреляционной матрицы используются критерии Бартлетта-Уилкса, Лоули и Кайзера.

Критерий сферичности Бартлетта-Уилкса [146] предлагается использовать, когда наблюдается слабая корреляционная зависимость между входными характеристиками. Данный критерий зависит от объема выборки и количества измеряемых характеристик.

$$\chi^2 = -\left[N - \frac{1}{6}(2M + 5)\right] ln|R|, \text{ с } \frac{M(M-1)}{2} \text{ степенями свободы,} \qquad (2.3)$$

где N- объем выборки, M - количество измеряемых характеристик, $|R|$ - определитель корреляционной матрицы.

Критерий основан на преобразовании определителя матрицы в статистику χ^2 [150]. Если корреляция между характеристиками слабая, то определитель корреляционной матрицы будет приближен к 1. Если наблюдается корреляция у двух и более характеристик, то определитель будет приближен к 0. С помощью данного критерия проверяется гипотеза о единичной матрице.

В свою очередь, Д. Лоули упростил критерий Бартлетта-Уилкса и предложил следующий подход

$$\chi^2 = N \sum_{i<j} r_{ij}^2 \;\; i,j = 1..M \tag{2.4}$$

где N – объем выборки, M – количество измеряемых характеристик.

Если полученный результат не удовлетворяет выдвинутой гипотезе о значимости корреляционной матрицы, то процедуру выделения факторов проводить бессмысленно.

Критерий адекватности выборки Кайзера (MSA) [146, 150, 159, 160] не зависит от объема выборки. Данный критерий сравнивает значения коэффициентов корреляции со значениями частных коэффициентов корреляции.

$$MSA = \frac{\sum_{i \neq j} r_{ij}^2}{\sum_{i \neq j} r_{ij}^2 + \sum_{i=j} q_{ij}^2}, \tag{2.5}$$

где $\sum_{i \neq j} r_{ij}^2$ – сумма квадратов элементов матрицы коэффициентов корреляции, за исключением элементов главной диагонали, $\sum_{i=j} q_{ij}^2$ – сумма квадратов соответствующих частных коэффициентов корреляции. Частный коэффициент оценивает тесноту связи между i и j характеристиками при фиксированном значении остальных характеристик. Если частный коэффициент корреляции превышает значение исходного коэффициента корреляции, то остальные характеристики оказывают влияние на связь между i-й и j-й характеристиками и ослабляют их зависимость. В обратной ситуации на данную связь не оказывают влияние остальные характеристики. Критерий MSA показывает целесообразность использования факторного анализа для корреляционной матрицы, высокое значение подтверждает эту гипотезу, небольшое значение опровергает.

Если значение критерия MSA>0,9, то говорят о безусловной адекватности; > 0,8 – высокая адекватность; > 0,7 – приемлемая адекватность; > 0,6 – удовлетворительная адекватность; > 0,5 – низкая адекватность; ≤ 0,5 – факторный анализ неприменим к выборке [150, 160].

С точки зрения К. Иберлы, проверка на значимость исходной корреляционной матрицы, используя представленные критерии, является не обязательной и играет второстепенную роль [146].

Критерии предварительной оценки числа выделяемых факторов

Предложенные критерии, позволяют предварительно оценить количество выделяемых факторов.

Критерий, предложенный Х. Кайзером и К. Дикманом (Кайзера) [146, 162], позволяет определить количество значимых факторов. Суть заключается в следующем: факторы, вклад которых в общую дисперсию меньше единицы, не участвуют в дальнейшем рассмотрении. Данный критерий основан на величине собственных значений факторов. При этом, чем больше входных данных, тем больше формируется факторов [146]. Данный критерий можно представить наглядно, что позволяет легко выделить факторы.

Критерий отсеивания, предложенный Р. Каттелом (scree-test) [146, 163], позволяет графически определить количество факторов, подлежащих выделению. Для этого на график наносятся собственные значения факторов в порядке убывания (график «каменистой осыпи»). Далее определяется точка, в которой убывание собственных значений слева направо максимально замедляется. Данная точка показывает количество значимых факторов.

Существует еще один способ выделения факторов. В рассмотрении оставляют только то количество факторов, которые описывают заданный процент общей дисперсии. Х. Харман [146,159] предложил исключать из рассмотрения компоненты, которые не оказывают существенный вклад в суммарную дисперсию.

Критерии проверки адекватности факторной модели

Д. Лоули [146] предложил тест, который позволяет оценить, насколько точно полученные факторы воспроизводят корреляционную матрицу. Для этого необходимо вычислить критерий значимости

$$\chi^2 = (N-1) ln \frac{|R^+|}{|R|} \quad \text{с } \frac{1}{2}((M-r)^2 - M - r) \text{ степенями свободы,} \tag{2.6}$$

где $|R^+|$ – определитель воспроизведённой матрицы корреляций с помощью определенной модели, $|R|$ – определитель исходной корреляционной матрицы, N-объем выборки, M – количество входных характеристик, r – количество выделенных факторов. Если полученное значение превышает табличное значение χ^2, то необходимо выделить большее количество факторов, а если не превышает, то r является искомым числом. Для использования данного теста необходимо учесть несколько фактов. Исследуемые характеристики должны иметь нормальное распределение, факторные нагрузки должны быть определены с помощью метода максимального правдоподобия, объем выборки должен быть достаточно большим.

М. Бартлеттом предложен усовершенствованный вариант (2.6) для средней выборки

$$\chi^2 = n' ln \frac{|R^+|}{|R|}, n' = N - \frac{1}{6}(2M+5) - \frac{2}{3}r. \tag{2.7}$$

С точки зрения К. Иберла, тест Лоули и критерий Бартлетта дают хорошую оценку для факторной модели при использовании метода максимального правдоподобия.

Д. Лоули и А. Максвелл представили свой упрощенный вариант определения критерия (2.7) [146, 164]

$$\chi^2 = n' \sum_{k<i} \frac{r^2_{\text{ост}_{ik}}}{u_i^2 * u_k^2}, n' = N - \frac{1}{6}(2M+5) - \frac{2}{3}r, \tag{2.8}$$

где $r^2_{\text{ост}_{ik}}$ – остаточные коэффициенты корреляции после выделения факторов, u_i^2, u_k^2 – значения характерностей переменных. Если полученное значение критерия не превышает табличное значение χ^2, то полученное количество факторов достаточно для воспроизведения корреляционной матрицы [146].

Критерий Хэмфри является простым способом оценки факторной модели. Для оценки необходимо знать N – объем выборки, найти максимальные значения двух факторных нагрузок *max1, max2*.

$$|r_{max1} \times r_{max2}| > \frac{2}{\sqrt{N}}. \quad (2.9)$$

Если условие выполняется, то простаяструктура факторной модели считается достигнутой. Понятие простой структуры было предложено Л.Л. Тэрстоуном и подразумевает переход от хаотического набора экспериментальных данных к стройной структуре, связывающие переменные с выделенными факторами [146].

Также при оценке достижения простой структуры пользуются тестом Баргмана [149]. При этом рассчитывают количество нулевых нагрузок для каждого *j*-го фактора:

$$\left|\frac{a_{ij}}{h_i}\right| < 0{,}1, \quad (2.10)$$

где a_{ij} – факторная нагрузка по i-й переменной, h_i – квадратный корень из общности (под общностью понимается дисперсия переменной, объясняемая общими факторами). Если количество нулевых нагрузок не ниже табличного значения, простая структура считается достигнутой.

Методы выделения факторов

На сегодняшний день существует два подхода в выделении факторов: математический и статистический. Математический подход точно не дает ответа, сколько необходимо факторов для определения взаимодействия переменных в выборке. К математическому подходу относятся метод главных факторов и центроидный метод. Факторы выделяют без учета ошибки выборки [146].

Метод главных компонент. Идея метода главных компонент (principal components) была предложена К. Пирсоном, Х. Хотеллинг представил его алгебраическое описание [146]. Суть данного метода заключается, в поиске последовательности ортогональных осей координат. Вдоль каждой оси в убывающем порядке определяется максимум полной и оставшихся дисперсий. Данный метод применим к разным исходным матрицам. Если в исходной матрице

на главной диагонали находятся единицы, то говорят об отсутствии характерных факторов, общности равны единице [146] и получают модель компонентного анализа. Если на главной диагонали матрицы находятся общности, то получают факторную модель.

Х. Харман в своей работе, представил модель компонентного анализа [150]

$$X_j = \sum_{i=1}^{M} a_{ij} F_j \, , j = 1..M \tag{2.11}$$

где M – количество рассматриваемых переменных, X_j – переменная, которая линейно зависит от M некоррелируемых между собой и упорядоченных по значению вклада в суммарную дисперсию компонент в порядке убывания F_j. Первая компонента оказывает наибольшее влияние на суммарную дисперсию наблюдаемых переменных, последняя – минимальное. Коэффициент a_{ij} – нагрузка, которая показывает степень влияния компоненты на рассматриваемую переменную, то есть отображает линейную корреляцию между компонентами и переменными. При этом главные компоненты ортогональны. Для определения значимости компоненты F_j вычисляют их собственные значения λ_j

$$\lambda_j{}^2 = \sum_{i=1}^{M} a_{ij}{}^2 \, , j = 1..M, \tag{2.12}$$

где M – количество переменных.

На практике, как правило, определяют несколько первых главных компонент, так как первые компоненты практически объясняют суммарную дисперсию. Критерии предварительной оценки числа выделяемых факторов описаны выше.

Нет единого мнения в вопросе, является ли метод главных компонент одним из этапов в расчетах внутри факторного анализа, либо факторный анализ является частным случаем метода главных компонент. Иберла считает, что «… Под анализом главных факторов подразумевается приложение метода главных компонент к редуцированной корреляционной матрице после оценки общностей» [146]. Метод главных компонент рассматривается как инструмент решения проблем факторов [146].

Метод главных компонент позволяет оценить вклады переменных в суммарную дисперсию, в свою очередь факторный анализ определяет факторы,

которые выявляют наличие взаимосвязи между входными характеристиками, таким образом, понижается размерность входных характеристик.

Метод главных факторов. В факторном анализе учитываются все исследуемые переменные, и для построения факторной модели они важны. Впервые метод возник в работах Спирмена [146].

Х. Харман [146, 157] также в своих исследованиях представил модель факторного анализа (2.1). Факторные нагрузки a_{ij} показывают линейную корреляцию между факторами и исследуемыми характеристиками. В отличие от модели компонентного анализа (2.11), в которой количество главных компонент равно количеству исследуемых переменных, в модели факторного анализа в фактор объединяются характеристики, которые имеют сильную корреляцию между собой, таким образом, понижая размерность входных данных.

Суммарная дисперсия в компонентном анализе описывается всеми входными характеристиками. В факторном анализе дисперсия исследуемой характеристики состоит из двух слагаемых: дисперсии, обусловленной общим фактором F_j и дисперсии, обусловленной характерным фактором u_i. При поиске факторной модели необходимо найти минимальное количество факторов, которые более полно объясняли бы взаимосвязи между наблюдаемыми переменными, минимизируя зависимость остальных переменных со специфическими факторами u_i. При этом модель должна содержательно интерпретировать полученные факторы [150].

Факторные нагрузки a_{ij} возведенные в квадрат представляют собой величину общности конкретной характеристики X_i и фактора F_j. Если факторы ортогональны, значение общности характеристики X_i по всем факторам определяется как:

$$h_i{}^2 = \sum_{j=1}^{r} a_{ij}{}^2 , i = 1..M, \qquad (2.13)$$

где r– количество общих факторов, M – количество переменных. При этом значение должно находиться в пределах от 0 до 1. Значение общности переменной X_i показывает, какая часть ее дисперсии объясняется факторами. Чем ближе значение стремится к единице, тем факторная модель точнее описывает

дисперсию исследуемой характеристики X_i, а также показывает высокую взаимосвязь характеристики X_i с другими характеристиками. Если значение стремится к нулю, то взаимосвязь между характеристиками низкая.

Для определения значимости фактора F_j необходимо вычислить его собственное значение и поделить на количество переменных *M*. Получаем долю суммарной дисперсии характеристик, которую объясняет каждый фактор [150]. Собственные значения вычисляются по формуле (2.12). Требование метода – каждый фактор должен учитывать максимум дисперсии.

Центроидный метод. Данный метод привлекателен из-за простоты расчетов. Идея метода заключается в следующем: исходные характеристики графически представляются точками на r-мерном пространстве (факторное отображение), где также указывается нулевая точка. Для определения однозначного решения, необходимо, чтобы первая координатная ось проходила через центр скопления точек, при этом остальные оси должны быть ортогональны ей [146]. Тогда координаты центра тяжести равны $(\frac{1}{M}\sum a_{i1}, 0,0, \dots, 0)$. Учитывая данное условие, нагрузка первого центроидного фактора вычисляется по формуле:

$$a_{i1} = t \cdot \sum_{i=1}^{M} r_{ki}, k = 1, .., M, \qquad (2.14)$$

$$t = \frac{1}{\sqrt{T}}, T = (\sum a_{i1})^2.$$

Затем определяется остаточная дисперсия

$$R_1 = R_h - R^+,$$

где R_h – редуцированная корреляционная матрица, R^+ - воспроизведенная корреляционная матрица.

Для вычисления следующих центроидных факторов используют эту же процедуру вычислений для остаточных корреляционных матриц R_i. Вычисления продолжаются до тех пор, пока остаточная матрица не примет вид нулевой матрицы. Для определения последующих центров тяжести, используют следующее правило. Если сумма проекций всех точек на ортогональные оси равна нулю и совпадает с началом координат после выделения *i*-центроидного фактора, необходимо сменить знаки нескольким переменным, чтобы точки находились по

одну сторону от оси *i*-го фактора. Таким образом формируется новый центр тяжести. Если сумма проекций всех точек равна нулю и не совпадает с началом координат, то процедуру вычисления останавливают.

К. Иберла подчеркнул, что смена знака является слабым местом в данном методе, необходим огромный опыт исследователя [146].

Метод главных осей. В методе главных осей используется идея метода главных компонент [164]. Но исходной матрицей выступает редуцированная корреляционная матрица. В первой итерации общности h_i^2 вычисляются по методы главных компонент. В следующих итерациях собственные значения λ_j^2 и факторные нагрузки a_{ij} вычисляются по значениям общностей, найденных на предыдущем шаге. Окончательное решение определяется либо заданным количеством итераций, либо достижением минимальной разницы между текущим значением общности и значением общности, найденным на предыдущем шаге.

Данный метод можно применить как к корреляционной матрице, так и к ковариационной матрице, что является основным достоинством данного метода.

Метод максимального правдоподобия. К. Иберла в своей работе приводит самое главное отличие метода максимального правдоподобия от остальных методов. Суть метода заключается в следующем: «… результирующие факторы инварианты относительно изменений масштаба переменных» [146]. Предполагается, что исследуемые характеристики должны иметь нормальное распределение, факторы должны быть ортогональны. В своих работах Д. Лоули, Х. Харман, и А. Максвелл [157, 164] приводят подробное описание метода. Цель метода состоит в нахождении общих факторных и характерных нагрузок, которые более точно опишут исходную корреляционную матрицу исследуемых характеристик (выборка из генеральной совокупности данных).

$$R = R_h + U^2, \qquad (2.15)$$

$$R_h = AA',$$

где R – матрица корреляций, на главной диагонали которой стоят единицы, R_h – матрица корреляций, на главной диагонали которой стоят общности (редуцированная корреляционная матрица), U – диагональная матрица, на

главной диагонали которой стоят нагрузки характерных факторов, A – редуцированная факторная матрица [146].

Используя функцию максимального правдоподобия, находят элементы (оценки) a_{ij} и u_i^2 соответствующих матриц A и U [164]:

$$A = \frac{-1}{2} n\, ln|R| - \frac{1}{2} n \sum_{i,j} a_{ij} r^{ij}, \quad (2.16)$$

где a_{ij} – элемент выборочной ковариационной матрицы, представляющий собой выборочные оценки дисперсий и ковариаций с n степенями свободы, r^{ij} – элемент матрицы R^{-1}. Данную функцию необходимо максимизировать по a_{ij} и u_i^2.

Д. Лоули и А. Максвелл в своей работе [164] отметили, что метод максимального правдоподобия приводит к единственным оценкам r_{ij}, но для оценок a_{ij} существует бесконечное количество решений. Поэтому вводится дополнительное условие [164]:

$$J = A'[U^2]^{-1}A,$$

которое позволяет определиться с единственным решением, то есть подбирается диагональная матрица A [146]:

$$A = J^{-1}A'[U^2]^{-1}(R - U^2). \quad (2.17)$$

Следующим шагом вычисляются элементы матрицы U^2

$$u_i^2 = 1 - h_i^2. \quad (2.18)$$

К. Иберла [146] предложил в качестве начальных условий использовать центроидные оценки для сокращения количества итераций в поиске более точного приближения к исходной матрице.

По мнению К. Иберлы метод главных факторов остается самым распространенным способом выделения факторов [146].

Методы вращения

Существует два подхода к проблеме вращения осей. Первый подход осуществляется графическим отображением осей, которые должны пройти через скопление точек. Второй подход связан с аналитическими методами. Недостатком в аналитических методах считается, что окончательный результат решения не до

конца согласуется с простой структурой. В графическом методе такой проблемы нет.

Вращение применяется для нахождения в пространстве факторов одной из возможных систем координат, упрощающих факторную структуру. Следствием этого является максимизация высоких корреляций и минимизация низких. Проблема вращения формулируется следующим образом [143]: необходимо найти матрицу преобразования *T* такую, что:

$$A^{\cdot} = AT,\ R_h = AA' = A^{\cdot}A^{\cdot\prime}, \tag{2.19}$$

где $A^{\cdot}$– факторная матрица в новой системе координат, A – ортогональная матрица исходная матрица фактурных нагрузок после процедуры выделения факторов, R_h–редуцированная корреляционная матрица.

Матрица преобразования *T* ортогональна, если выполняется условие:

$$T'T = I, \tag{2.20}$$

где T' – транспонированная матрица *T, I* – единичная матрица.

П. Хофстеттер придерживается поиска ортогонального вращения.

Над проблемой вращения работали П. Хорст, Л. Тэрстоун, Л. Тукер. Ими предложены критерии решения задач вращения. Самый удачный метод предложен Л. Тукером. Также были предложены методы ортогонального и косоугольного вращений. Вращение называется ортогональным, если в процессе вращения координатных осей прямой угол остается неизменным.

Ортогональный метод. Ортогональный метод вращения применяется, когда факторы ортогональны друг к другу. Но если данное условие не соблюдается, то полученное решение можно неверно интерпретировать.

Г. Фергюсоном предложен критерий вращения, который отличался от постулатов Тэрстоуна. Идея состоит в следующем, критерий экономного описания переменных принимает минимальное значение, когда большинство точек лежит рядом с осями [146].

Квартимакс. Существенным недостатком данного критерия является выделение генерального фактора. Данный критерий упрощает описание входной характеристики и не учитывает столбцы факторной матрицы [146]. То есть

происходит упрощение строк матрицы факторных нагрузок, происходит минимизация сложности переменных за счет увеличения дисперсии факторных нагрузок по каждой переменной [164].

Варимакс. Является модификацией метода квартимакс. Х. Кайзер [146] предложил идею упрощения фактора, которая не учитывает строки факторной матрицы. Простота фактора определяется дисперсией квадратов его нагрузок. Происходит минимизация сложности факторов за счет увеличения дисперсии факторных нагрузок по каждому фактору [164]. Нахождение максимума функции

$$m\sum_{l=1}^{r}\sum_{i=1}^{m}\left(\frac{b_{il}}{h_i}\right)^4-\sum_{l=1}^{r}\left[\sum_{i=1}^{m}\frac{b_{il}^2}{h_i^2}\right]^2\rightarrow max \qquad (2.21)$$

позволяет определить ортогональное положение системы координат. Данный критерий является модифицированным. Это необходимо для того, чтобы при определении осей координат нормированные переменные имели равные веса.

Эквимакс. Одновременное упрощение и фактора, и характеристики. Данный метод используется, когда точно определено количество факторов.

Косоугольный метод. Факторы не всегда бывают некоррелируемыми. Выделенные факторы, оказывающие влияние на исходные характеристики, взаимосвязаны между собой. Тогда предлагается косоугольная система координат [146]. В работах Р. Каттелла и К. Дикмана [146] показано, что косоугольное факторное отображение лучше отражают взаимосвязь между характеристиками. Л. Тэрстоун, К. Иберла также поддерживают косоугольное представление.

Максплейн. Схож с итеративными процедурами вращения графического способа.

Облимакс. Завышает коэффициенты корреляции между факторами в сравнении с исходными значениями, то есть величина угла между осями занижается, и решение характеризуется наибольшей косоугольностью [146].

Квартимин. Минимизация функции без наложения ограничений

$$\sum_{l=1}^{r}\sum_{i=1}^{m}(a_{ik}a_{il})^2\rightarrow min. \qquad (2.22)$$

упрощает факторы, минимизируя сумму смешанных произведений столбцов структурной матрицы.

Облимин. Это ряд критериев, которые были предложены Дж. Кэрроллом. В основе их критериев взяты алгоритмы Квартимина и Коваримина. Упрощает факторы, минимизируя сумму смешанных произведений столбцов матрицы факторных нагрузок. Коваримин является аналогом косоугольного варианта варимакс – критерия.

Орто-косоугольное. Изменяет масштаб в пространстве факторов, чтобы они были ортогональны. Неперемасштабированные факторы могут коррелировать.

Промакс. Факторы, полученные при ортогональном вращении, вращаются вновь с учетом корреляции между ними.

Если конфигурация векторов, строго соответствующая корреляционной матрице, позволяет путем вращения достигнуть такого состояния, что значительное большинство переменных окажется на гиперплоскостях координат или вблизи них, то говорят о простой структуре [146]. Простая структура характеризует взаимосвязи между конфигурацией векторов и осей координат внутри пространства общих факторов.

Рассмотрим, как полученная модель факторного анализа, окажет влияние на разделение однородных партий алгоритмами кластеризации. В работе [16] показано, что задача выделения однородных партий может быть сведена к задаче кластерного анализа. Каждая группа (кластер) должна представлять однородную партию, изготовленную из одного вида сырья. Для решения задачи выявления однородных партий в работах [165, 166, 167] предложено применение алгоритма кластеризации k-средних. В [17] рассмотрен метод нечеткой кластеризации на основе EM-алгоритма. Предложена модель разделения однородных производственных партий на основе смеси гауссовых распределений [18]. В [168] рассмотрены вопросы использования ансамблей алгоритмов кластеризации (k-средних, k-медоид, k-медиан, EM, а также, их оптимизированных версий). В [106, 169] рассмотрено применение генетических алгоритмов с жадной эвристикой, а также модификаций EM-алгоритма для разделения однородных партий ЭРИ. Показано преимущество новых алгоритмов над классическими алгоритмами кластеризации для многомерных данных.

Основная цель кластерного анализа заключается в разделении объектов на группы по схожим характеристикам в общей выборке. В нашем случае, под разделением будем понимать, что каждый объект находится только в одной группе. Каждая группа называется кластером. Каждый кластер обладает такими свойствами как, плотность, дисперсия, размеры, форма и отделимость [146, 160]. Б. Эверитт предложил свое определение кластера [146] «…непрерывные области (некоторого) пространства с относительно высокой плотностью точек, отделенные от других таких же областей областями с относительно низкой плотностью точек». К методам кластерного анализа относятся иерархические агломеративные методы; иерархические дивизимные методы; итеративные методы группировки; методы поиска модельных значений плотности; факторные методы; методы сгущений; методы, использующие теорию графов [146]. Применяя данные методы к одним и тем же исходным данным, можно получить разный результат разделения объектов по группам.

Процедуры выделения объектов в группы в кластерном анализе состоит из следующих этапов. Первый этап заключается в обработке исходных данных. На данном этапе происходит предварительная очистка данных, а также нормирование характеристик, если в этом есть необходимость. На втором этапе определяется способ вычисления расстояния между группами объектов. На третьем этапе происходит разделение объектов на группы. На четвертом этапе – интерпретация результатов. И на заключительном пятом этапе определяется качество полученной группировки.

2.2 Пример практической задачи автоматической группировки промышленной продукции

Современный космический аппарат – сложнейшая конструкция, которая при пребывании в космосе должна осуществлять порученные ей задачи в течение одного – двух десятилетий [170]. Космический аппарат содержит от 100 до 200 тысяч электронных деталей. Значительная стоимость аппарата и отсутствие возможности ремонта электрорадиоизделий во время совершения космического

полета требует от электронной компонентной базы (ЭКБ) исключительного качества.

Промышленные партии электронной компонентной базы могут быть неоднородными и быть собранными из нескольких производственных партий сырья. По этой причине результаты тестовых испытаний на выборке ЭКБ распространять на всю поставленную партию деталей необходимо осторожно, как минимум, с убеждением в том, что партия изготовлена из одной партии сырья или, что разброс характеристик различных партий невелик. Это сопряжено с тем, что относительно незначительные изменения в производственном процессе могут кардинально повлиять на характеристики чувствительности. Следует понимать, из какого количества однородных групп собрана производственная партия электрорадиоизделий (ЭРИ) [70,168,171].

В случае если установленная совокупность компонентов состоит из нескольких различных групп, тогда для того, чтобы обладать аргументированной позицией о каждой из них, следует провести испытания для каждой группы, предварительно выявив их [172].

В первую очередь, следует предотвратить попадание в бортовую аппаратуру потенциально ненадежных ЭРИ. Для исключения попадания в бортовую аппаратуру КА потенциально ненадежных ЭРИ внедрен принцип комплектования аппаратуры через специализированные испытательные технические центры. Технический центр обеспечивает закупку ЭКБ у поставщиков, а затем проводит испытания входного контроля (ВК), дополнительного отбраковочного испытания (ДОИ), диагностического неразрушающего контроля (ДНК) с применением выборочного разрушающего физического анализа (РФА).

Все проводимые испытания делятся две группы:

1) сплошные испытания для всей партии элементов – ДОИ, ДНК. (Задачей ДОИ и ДНК ЭРИ является индивидуальная отбраковка элементов, имеющих скрытые дефекты изготовления.)

2) выборочные испытания для нескольких элементов из партии – РФА. РФА проводится с целью определения соответствия образцов ЭРИ требованиям

конструкции и технологического процесса изготовления и выявления нарушений этих требований.

В течение 20 лет эксплуатации космического аппарата, начиная с апреля 2000 года, на космическом аппарате «Sesat» [136], не выявлено существенных замечаний к ЭРИ, к которым был применен такой подход испытаний.

Во вторую очередь, необходимо обеспечить работоспособность ЭРИ при неблагоприятных внешних воздействиях, например, обеспечить устойчивость к радиационным нагрузкам. В литературе изложено, что радиационная стойкость любого ЭРИ из производственной партии известна и, главное, одинакова. На практике же характеристики ЭРИ, в том числе радиационная стойкость, внутри производственной партии различны, выявление таких различий является целью проведения РФА.

Для того, чтобы распространить результаты испытаний РФА на всю производственную партию изделий, необходимо быть уверенными в том, что мы имеем дело с партией ЭРИ, изготовленной из однородной партии сырья. Поэтому выявление однородных производственных партий из сборных партий ЭРИ является одним из важнейших этапов при проведении испытаний.

На практике проводится ряд тестов, от десятков до нескольких тысяч для каждого изделия, результаты сводятся в таблицу и служат данными для анализа.

Таким образом, формируется задача: разделение на однородные производственные партии на основе данных тестовых испытаний.

Для решения задачи данные предоставлены АО «Испытательным техническим центром - НПО ПМ» (АО «ИТЦ - НПО ПМ») г. Железногорск.

2.3 Применение факторной модели в задачах автоматической группировки промышленной продукции

Рассмотрим применение факторной модели на практической задаче группировки электрорадиоизделий по однородным производственным партиям [173].

Исходные данные в данной задаче представляют собой набор некоторых характеристик электрорадиоизделий, измеренных в ходе обязательных тестовых испытаний. В качестве примера в настоящей работе рассматривается две выборки.

Микросхемы 140UD25AS1VK (далее по тексту Выборка I). Данная выборка содержит 42 вектора данных (входные измеряемые характеристики). Выборка составлена из данных об изделиях, заведомо принадлежащих двум разным партиям: партия 1 содержит 30 изделий, партия 2 – 16 изделий. Из рассмотрения исключены характеристики с нулевыми значениями. В рассмотрении остается 9 входных характеристик.

Микросхемы 1526IE10_002 (далее по тексту Выборка II). Общее количество изделий составляет 3987 штук. Партия 1 содержит 71 изделие, партия 2 – 116, партия 3 – 1867, партия 4 – 1250, партия 5 – 146, партия 6 – 113, партия 7 – 424. Каждая партия содержит информацию о 205 входных измеряемых характеристиках продукта. Входные характеристики, для которых вектор данных содержит только нулевые значения или для которых количество ненулевых значений не превышает 10%, были исключены из рассмотрения. Для дальнейшего анализа в рассмотрении осталось 67 входных характеристик.

Некоторые основные статистики, а также подбор вероятностного распределения характеристик представлены для выборки I в таблице 2.1. Гистограммы наблюдаемых частот изучаемых характеристик и графики подгонки распределений отображены на рисунках 2.2 и 2.3 (Выборка I), 2.4-2.6 (Выборка II).

Измеренные в ходе тестов характеристики по двух выборкам разделяются на 3 группы:

а) группа характеристик, для которых гауссовский характер распределения четко прослеживается (для первой выборки (Выборка I) переменные In1 – In6, для второй выборки (Выборка II) – In21 – In28, In39 – In46, In92 – In107);

б) характеристики, для которых гауссовский характер распределения прослеживается на частотной гистограмме, но гистограмма содержит пропуски в

силу ограничения точности измерительного прибора (переменные In19 – In20 для Выборки I, In84 – In91для Выборки II);

в) характеристики, для которых частотная гистограмма имеет вид, не соответствующий гауссовскому распределению (переменная In7 для Выборки I, In 57 – In64, In75 – In82, In10 – In20 для Выборки II).

Характеристики последней группы можно четко отделить от остальных по критерию эксцесса – его значение для таких характеристик существенно больше, чем для характеристик, имеющих гауссовское распределение.

Таблица 2.1 – Подгонка вероятностных распределений характеристик Выборки I

Характеристика	Среднее	Стандартное отклонение	Критерий эксцесса	Вид распределения	Критерий Колмогорова-Смирнова	
					значение	p
In1	-6,6201	1,064412	-1,89865	Смешанное гауссовское	0,064207	0,963638
In2	6,7121	1,065861	-1,89644	Смешанное гауссовское	0,073631	0,899834
In3	-7,9303	1,017114	-1,73217	Смешанное гауссовское	0,089655	0,725008
In4	7,9159	1,035469	-1,65965	Смешанное гауссовское	0,078055	0,858188
In5	-0,5413	0,667941	1,89437	Нормальное	0,138952	0,209289
In6	0,5704	0,672340	1,26743	Нормальное	0,145728	0,167753
In7	0,2183	0,280165	6,04933	Логнормальное	0,139975	0,202556
In19	69,4429	0,052595	-0,90258	Смешанное гауссовское	0,237246	0,002947
In20	-69,271	0,173430	0,42352	Смешанное гауссовское	0,148851	0,150937

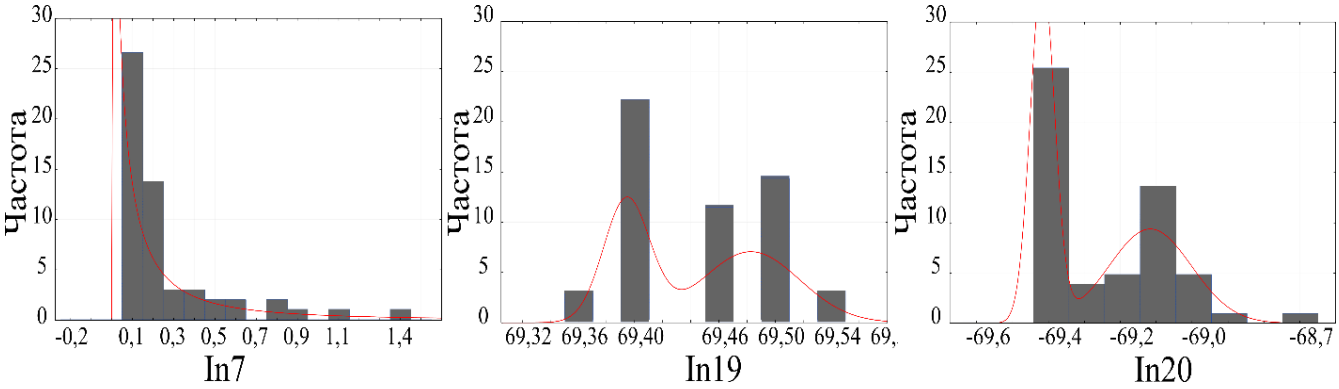


Рисунок 2.2 – Гистограммы наблюдаемых частот и графики подгонки распределений. Выборка I. Гистограммы не соответствуют гауссовскому распределению

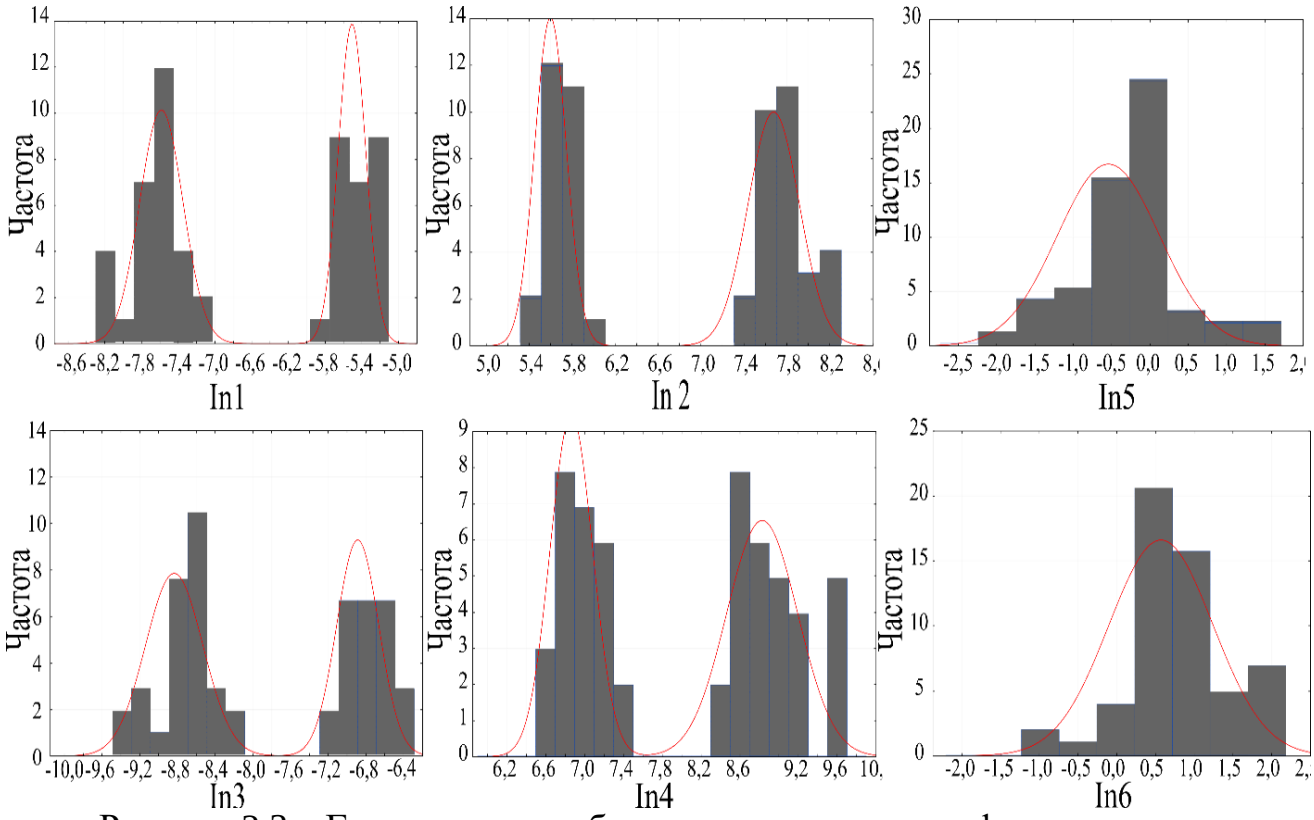


Рисунок 2.3 – Гистограммы наблюдаемых частот и графики подгонки распределений. Выборка I. Нормальное гауссовское распределение (In1-In4), с частотными промежутками (In5, In6)

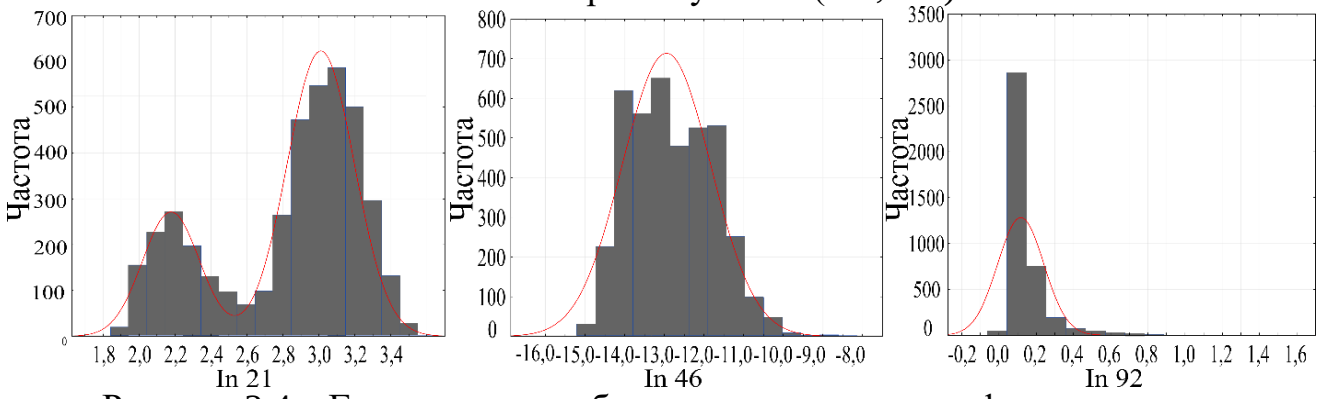


Рисунок 2.4 – Гистограммы наблюдаемых частот и графики подгонки распределений. Выборка II. Нормальное гауссовское распределение

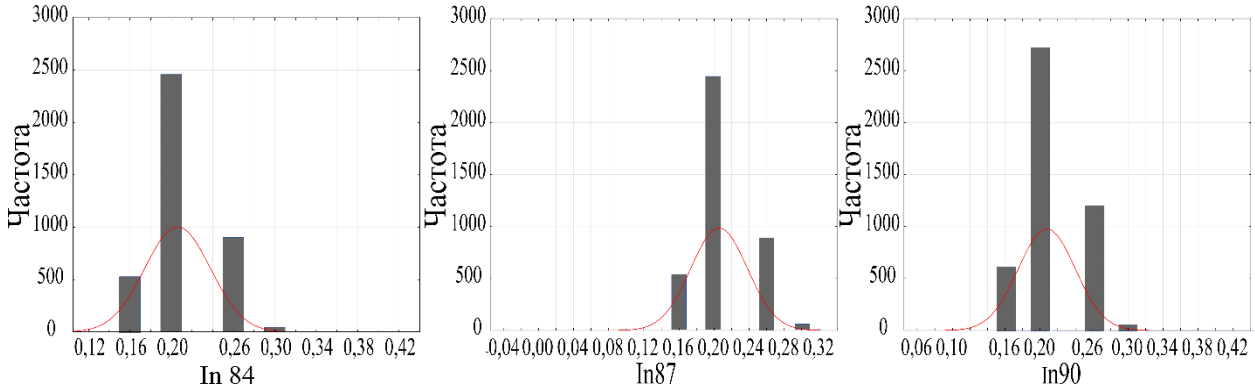


Рисунок 2.5 – Гистограммы наблюдаемых частот и графики подгонки распределений. Выборка II. Нормальное гауссовское распределение с частотными промежутками

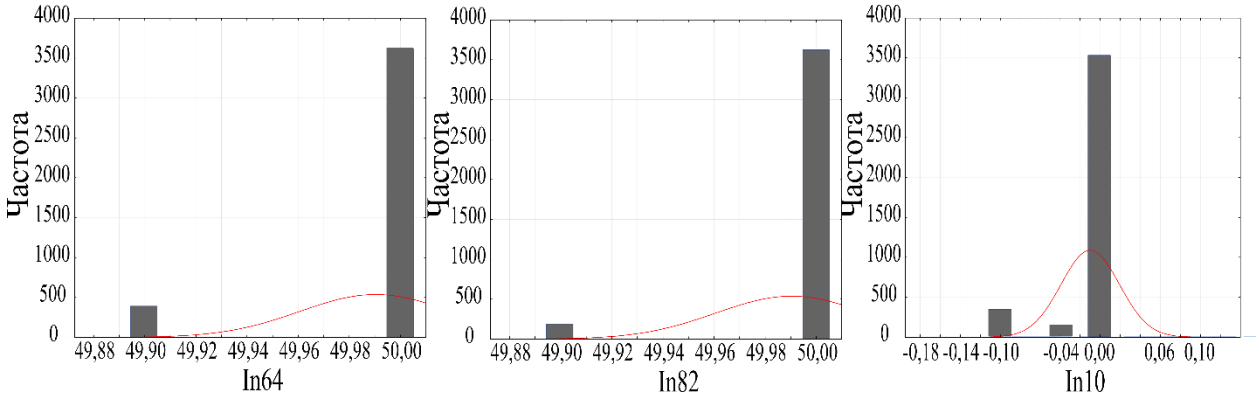


Рисунок 2.6 – Гистограммы наблюдаемых частот и графики подгонки распределений. Выборка II. Гистограмма не соответствует гауссовскому распределению

Нормирование характеристик производилось по формуле (2), предложенной в работе [16].

Проверка значимости исходной корреляционной матрицы

В целях понижения размерности входных данных для алгоритмов кластеризации был применен факторный анализ. Ряд популярных методов факторного анализа основаны на матрице парных корреляций характеристик [174]. Корреляция Пирсона предполагает нормальное распределение характеристик. Однако, ряд характеристик, в рассматриваемой задаче согласно рисункам 2.2 и 2.6 не имеют нормальное гауссовское распределение, поэтому также использована матрица корреляций Спирмена [175].

Первым этапом факторного анализа будет нахождение матрицы парных корреляций Пирсона и Спирмена, в данной работе покажем матрицы корреляций для Выборки I (таблицы 2.2, 2.3). В таблицах выделены корреляции, значимые на уровне p<0,05. Далее, делая предположение об ортогональности факторов, получим $R = A \cdot A^T$,

Таблица 2.2 – Корреляционная матрица Спирмена

	In1	In2	In3	In4	In5	In6	In7	In19	In20
In1	1,00								
In2	-0,99	1,00							
In3	0,96	-0,96	1,00						

In4	-0,98	0,98	-0,98	1,00					
In5	0,06	-0,05	-0,06	0,001	1,00				
In6	-0,06	0,05	0,04	0,02	-0,98	1,00			
In7	-0,15	0,14	-0,11	0,18	-0,33	0,35	1,00		
In19	-0,80	0,80	-0,74	0,76	-0,34	0,34	0,26	1,00	
In20	0,80	-0,80	0,74	-0,76	0,33	-0,30	-0,30	-0,80	1,00

Таблица 2.3 – Корреляционная матрица Пирсона

	In1	In2	In3	In4	In5	In6	In7	In19	In20
In1	1,00								
In2	-1,00	1,00							
In3	0,99	-0,99	1,00						
In4	-0,99	0,99	-1,00	1,00					
In5	0,09	-0,08	0,02	-0,01	1,00				
In6	-0,11	0,11	-0,04	0,04	-0,96	1,00			
In7	-0,31	0,31	-0,29	0,32	-0,30	0,42	1,00		
In19	-0,77	0,77	-0,75	0,75	-0,31	0,33	0,35	1,00	
In20	0,88	-0,88	0,85	-0,84	0,19	-0,19	-0,28	-0,72	1,00

На основе корреляционных матриц проведем проверку целесообразности применения факторной модели для наших исходных данных. Используя критерии Бартлетта-Уилкса, Лоули и Кайзера оценим значимость исходной корреляционной матрицы. Для этого найдем частные коэффициенты корреляции для двух выборок. В работе продемонстрированы частные коэффициенты для выборки I (таблицы 2.4 и 2.5).

А) Выборка I

По критерию Бартлетта-Уилкса для корреляционной матрицы Спирмена χ^2=1054,733 (R= 1,65656×10^{-9}), Пирсона χ^2= 1382,223 (R= 3,11034×10^{-12}) при n=56, m=9, число степеней свободы df=36, $p=0.000$, подтверждаем зависимость характеристик и значимость корреляционных матриц.

По критерию адекватности выборки Кайзера (MSA) для матрицы Спирмена факторный анализ не применим к выборке (MSA=0,357), Пирсона пороговое значение превышает 0,7 (MSA=0,755), что подтверждает приемлемый уровень адекватности применения факторной модели.

По критерию Лоули для матрицы Пирсона значение χ^2=1547,30 и Спирмена χ^2=1462,94 превышает табличные значения. Это еще раз подтверждает значимость корреляционной матрицы.

Из трех критериев подтверждается значимость исходной корреляционной матрицы по Спирмену. По матрице Пирсона значимость подтверждается по всем критериям.

Таблица 2.4 – Частные коэффициенты корреляции по матрице Пирсона. Выборка I

	In1	In2	In3	In4	In5	In6	In7	In19	In20
In1		-0,99906	-0,03229	0,407787	0,150212	0,073974	0,050392	-0,55876	-0,24256
In2	-0,99906		-0,05868	0,410896	0,144825	0,073682	0,052366	-0,55820	-0,26442
In3	-0,03229	-0,05868		-0,64578	-0,17890	-0,12286	0,345517	-0,10270	-0,19941
In4	0,407787	0,410896	-0,64578		-0,14743	-0,15991	0,354215	0,198651	0,174521
In5	0,150212	0,144825	-0,17890	-0,14743		-0,95233	0,424576	0,016082	0,149144
In6	0,073974	0,073682	-0,12286	-0,15991	-0,95233		0,512293	0,056871	0,137981
In7	0,050392	0,052366	0,345517	0,354215	0,424576	0,512293		0,091261	-0,01114
In19	-0,55876	-0,55820	-0,10270	0,198651	0,016082	0,056871	0,091261		-0,20406
In20	-0,24256	-0,26442	-0,19941	0,174521	0,149144	0,137981	-0,01114	-0,20406	

Таблица 2.5 – Частные коэффициенты корреляции по матрице Спирмена. Выборка I

	In1	In2	In3	In4	In5	In6	In7	In19	In20
In1		-0,98934	0,316176	0,54799	-0,26818	-0,30655	-0,27496	0,143326	0,133654
In2	-0,98934		0,293196	0,625438	-0,30917	-0,34541	-0,30938	0,182178	0,108469
In3	0,316176	0,293196		-0,62318	-0,14362	-0,0784	0,337111	-0,15786	0,066551
In4	0,54799	0,625438	-0,62318		0,283978	0,304526	0,467848	-0,22335	-0,02558
In5	-0,26818	-0,30917	-0,14362	0,283978		-0,97266	4,324377	0,054839	0,34494
In6	-0,30655	-0,34541	-0,0784	0,304526	-0,97266		0,002856	0,137541	0,291944
In7	-0,27496	-0,30938	0,337111	0,467848	4,324377	0,002856		0,143284	-0,16223
In19	0,143326	0,182178	-0,15786	-0,22335	0,054839	0,137541	0,143284		-0,25853
In20	0,133654	0,108469	0,066551	-0,02558	0,34494	0,291944	-0,16223	-0,25853	

Б) Выборка II

По критерию Бартлетта-Уилкса для корреляционной матрицы Спирмена χ^2=502250,1587 (R= 9,36119×10^{-56}), Пирсона χ^2= 595003,1468 (R= 6,44048×10^{-66}) при n=3987, m=67, число степеней свободы df=2244, p=0,000, подтверждаем зависимость характеристик и значимость корреляционных матриц.

По критерию адекватности выборки Кайзера (MSA) для матрицы Спирмена (MSA=0,9699), Пирсона (MSA=0,9677) пороговое значение превышает 0,9, достигнута безусловная адекватность применения факторной модели.

По критерию Лоули для матрицы Пирсона значение χ^2=2,283678×10^6 и Спирмена χ^2=2,096615×10^6 превышает табличные значения.

Все критерии подтверждают значимость корреляционной матрицы выборки II и по Пирсону, и по Спирмену.

Предварительная оценка числа выделяемых факторов

После проверки значимости корреляционной матрицы произведено сравнение решений методами главных компонент, главных факторов, главных осей, максимального правдоподобия и центроидным методом.

А) Выборка I

Максимальное число факторов составляет 9 (рисунки 2.7, 2.8). Собственные значения факторов, полученные при использовании перечисленных методов, приведены в таблицах 2.6, 2.7. В данном случае результаты отличаются незначительно, поэтому можно говорить о прохождении факторным решением теста на повторяемость.

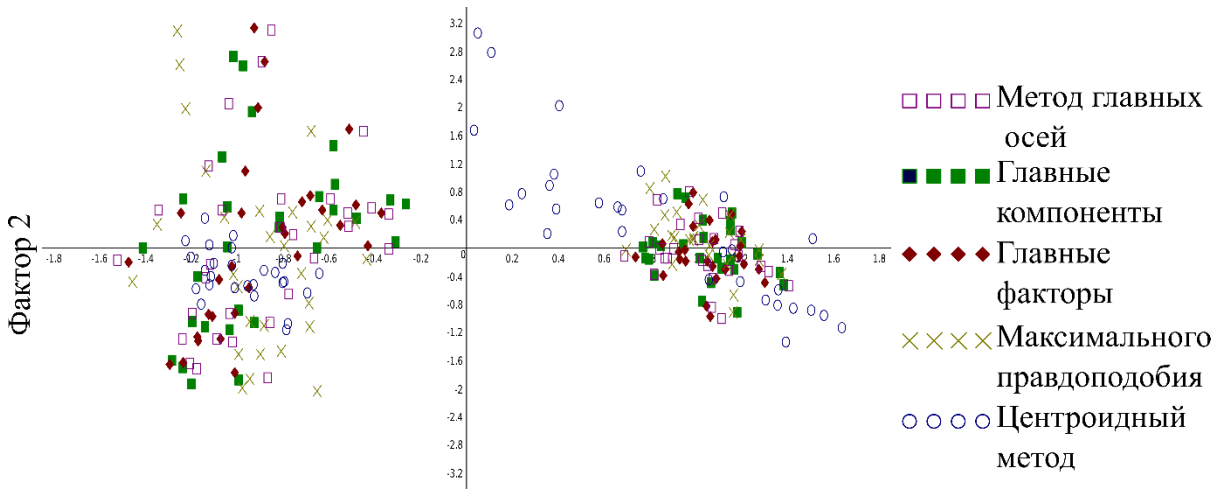


Рисунок 2.7 - Точки в факторном пространстве. Матрица Спирмена

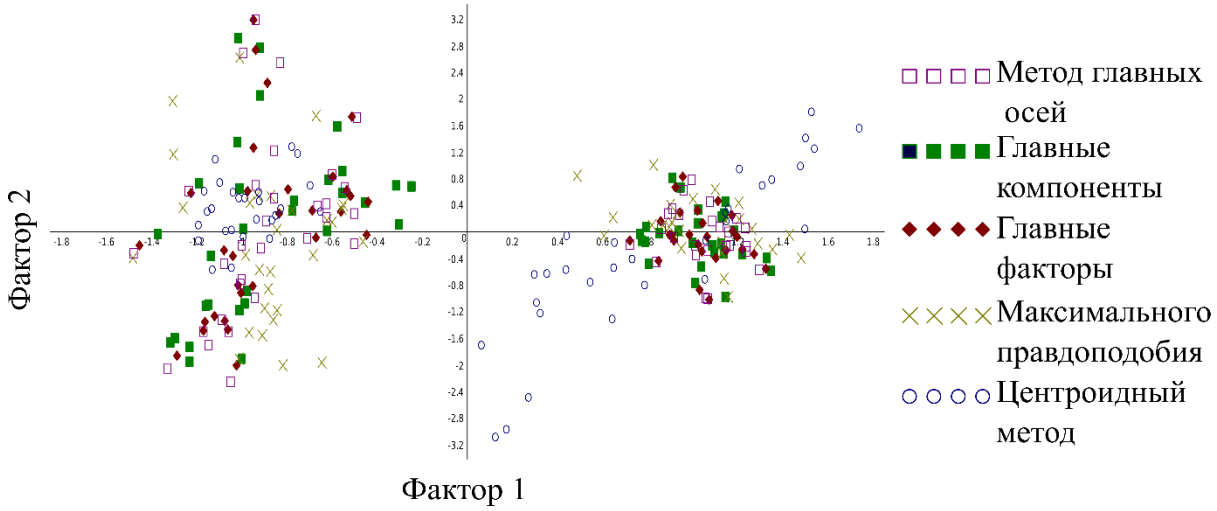


Рисунок 2.8 – Точки в факторном пространстве. Матрица Пирсона

Таблица 2.6 – Собственные значения факторов, выделенных различными методами. Матрица Спирмена

Факторы	Главные компоненты	Главные факторы	Максимального правдоподобия	Центроидный метод	Метод главных осей
1	**5,411839**	**5,340240**	**5,146639**	**5,097230**	**5,342110**
2	**2,257600**	**2,183460**	**2,247645**	**2,408946**	**2,188454**
3	0,793208	0,224988	0,120145	0,205037	0,227375
4	0,258923	0,108622	0,376758	0,169539	0,113668
5	0,200748	0,029687	0,015215	0,033856	0,030655
6	0,045831	0,008298	0,276859	0,017106	0,012273
7	0,016823	0,000196	0,082202	0,005791	
8	0,014736			0,004890	
9	0,000289			0,005584	

Таблица 2.7 – Собственные значения факторов, выделенных различными методами. Матрица Пирсона

Факторы	Главные компоненты	Главные факторы	Максимального правдоподобия	Центроидный метод	Метод главных осей
1	**5,604700**	**5,540853**	**5,372149**	**5,311326**	**5,538156**
2	**2,133308**	**2,061030**	**2,131905**	**2,270077**	**2,071828**
3	0,736465	0,236789	0,234625	0,235531	0,240618
4	0,294884	0,096440	0,480127	0,131754	0,096651
7	0,191385	0,083067	0,022954	0,082592	0,083589
6	0,025770	0,005323	0,022916	0,013550	0,003237
7	0,010411			0,008266	
8	0,003070			0,007345	
9	0,000008			0,004638	

Для оценки количества факторов, достаточных для отбора в факторное решение, будем использовать критерий Кайзера, Каттела. В таблицах 2.6 и 2.7 по критерию Кайзера предлагается отбирать факторы с собственными значениями, большими единицы. Этому соответствуют два первых фактора. Количество достаточных факторов в данном случае не зависит от метода, которым проведено выделение факторов.

Другим способом отбора является критерий отсеивания Каттела [163], использующий график «каменистой осыпи», на котором отражаются собственные значения всех факторов. По матрице Спирмена данный критерий рекомендует оставить в модели три первых фактора с максимальными собственными значениями (для метода главных компонент – 3 или 4 фактора) (рисунок 2.9).

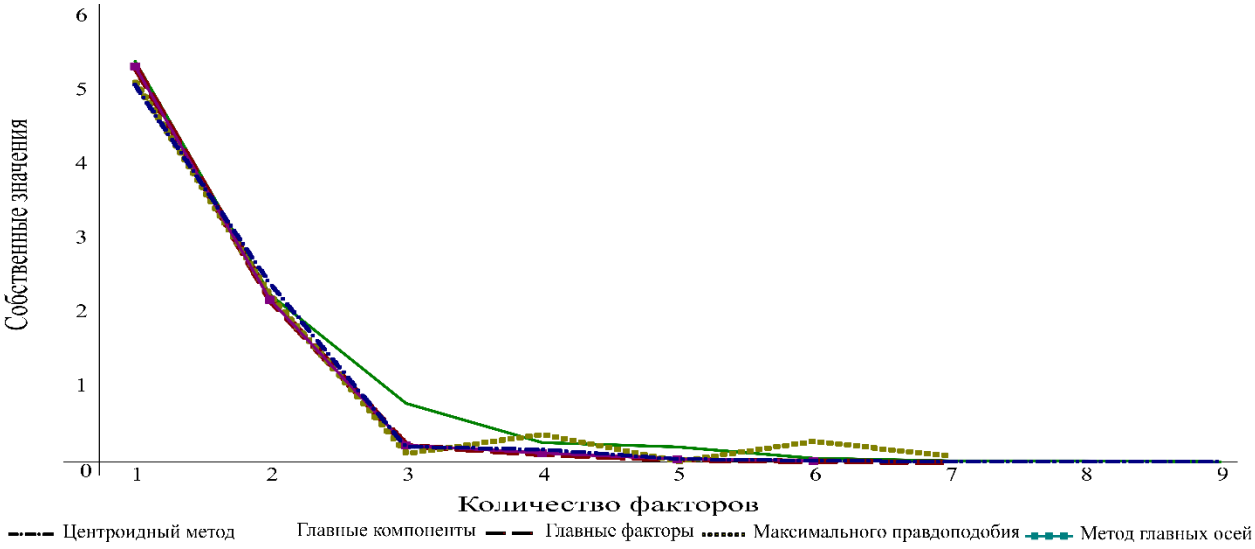


Рисунок 2.9 – Графики «каменистой осыпи». Матрица Спирмена

По матрице Пирсона критерий также независимо от метода выделения рекомендует оставить в модели три первых фактора с максимальными собственными значениями (рисунок 2.10).

Далее проведем процедуру вращения для нахождения в пространстве факторов одной из возможных систем координат, упрощающих факторную структуру. Вращение рассмотрим на примере факторов, выделенных методами главных компонент и главных осей. Использованы методы ортогонального вращения: варимакс с нормализацией по Кайзеру и квартимакс с нормализацией [146].

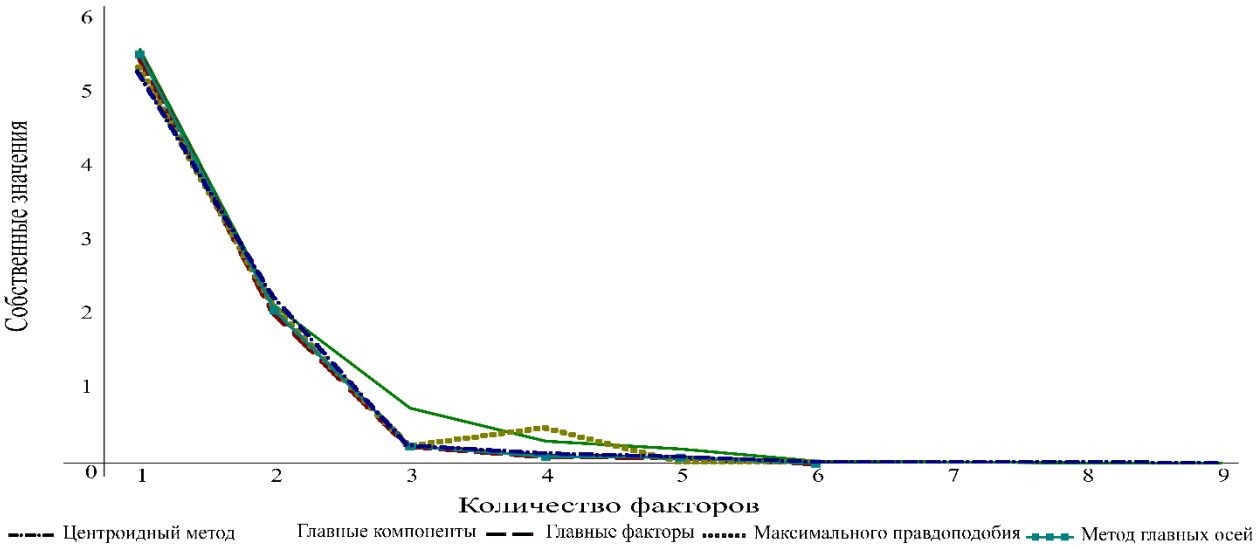


Рисунок 2.10 – Графики «каменистой осыпи». Матрица Пирсона

Варимакс-критерий максимизирует суммарную дисперсию квадратов нагрузок общих факторов по каждому исходному признаку. Квартимакс-критерий базируется на том факте, что при стремлении к нулю значений нагрузок будет убывать сумма квадратов попарных произведений элементов строк матрицы *A*.

Матрица Спирмена

В таблицах 2.8 и 2.9 приведены собственные значения факторов после вращения, дисперсия (в процентах), описываемая данными факторами по всем переменным, а также накопительная дисперсия по всем переменным.

Таблица 2.8 – Вращение (метод главных компонент). Матрица Спирмена

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	4,9002	5,2909	5,4118	54,447	58,7879	60,131	54,447	58,7879	60,132
2	2,1205	2,1607	2,2576	23,562	24,0074	25,084	78,008	82,7953	85,216
3	1,0271	0,9630	0,7932	11,412	10,7005	8,8134	89,420	93,4958	94,029
4	0,4420	0,2568	0,2589	4,9109	2,8529	2,8769	94,331	96,3487	96,906
5	0,4312	0,2496	0,2008	4,7914	2,7737	2,2305	99,122	99,1224	99,137
6	0,0400	0,0410	0,0458	0,4448	0,4554	0,5092	99,567	99,5778	99,646
7	0,0193	0,0189	0,0168	0,2139	0,2094	0,1869	99,781	99,7872	99,833
8	0,0194	0,0189	0,0147	0,2158	0,2097	0,1637	99,997	99,9969	99,997
9	0,0003	0,0003	0,0003	0,0032	0,0032	0,0032	100,00	100,0000	99,999

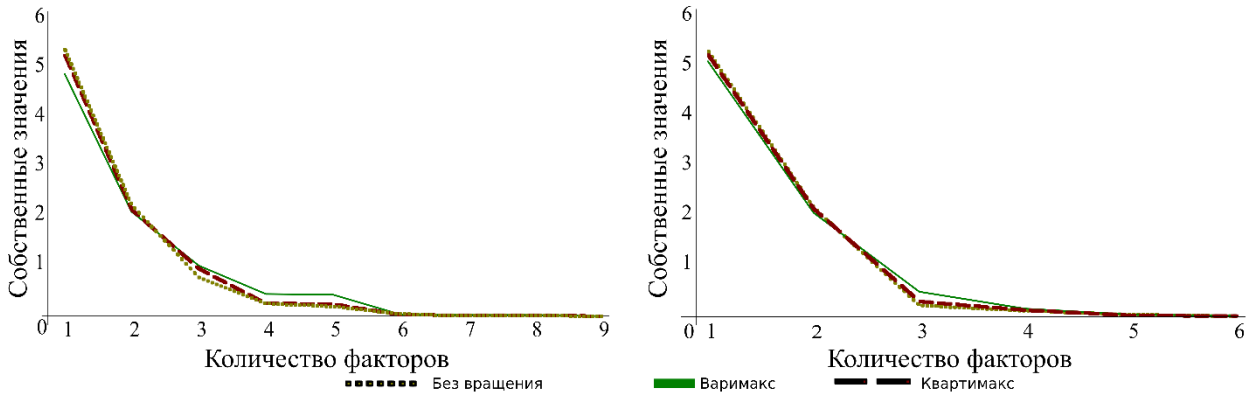
Таблица 2.9 – Вращение (метод главных осей). Матрица Спирмена

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	5,1343	5,2758	5,3421	57,046	58,6203	59,357	57,048	58,6203	59,357
2	2,0876	2,1645	2,1885	23,195	24,0503	24,316	80,243	82,6706	83,673
3	0,4962	0,3034	0,2274	5,5128	3,3711	2,5264	85,756	86,0417	86,199
4	0,1497	0,1259	0,1137	1,6634	1,3987	1,2630	87,419	87,4404	87,462
5	0,0302	0,0293	0,0307	0,3358	0,3255	0,3406	87,755	87,7659	87,803
6	0,0166	0,0156	0,0123	0,1844	0,1733	0,1364	87,939	87,9392	87,939

При варимакс-вращении факторов, выделенных методом главных компонент, собственное значение третьего фактора превышает единицу, а значит, также может быть включено в модель, что подтверждается графиком «каменистой осыпи» (рисунок 2.11); при этом суммарная дисперсия, описываемая всеми факторами, составляет при варимакс-вращении 89,42%, при квартимакс-вращении – 93,50%. Кумулятивная доля остальных факторов, не включенных в модель, при любом вращении не превышает 11% от общей дисперсии.

При вращении факторов, выделенных методом главных осей, выделение третьего фактора по рассмотренным выше критериям нецелесообразно (рисунок 2.11); при этом решение, полученное при варимакс-вращении, обеспечивает объяснение 80,24% общей дисперсии, при квартимакс-вращении – 82,67%.

Значения факторных нагрузок методом главных компонент и методом главных осей приведены в таблицах 2.10 и 2.11. Выделены нагрузки со значениями по модулю >0,7.



а) Главные компоненты б) Главные оси

Рисунок 2.11 – График «каменистой осыпи». Матрица Спирмена

Таблица 2.10 – Факторные нагрузки (метод главных компонент). Матрица Спирмена

	Без вращения			Варимакс нормализованных			Квартимакс нормализованных		
	Фактор 1	Фактор 2	Фактор 3	Фактор 1	Фактор 2	Фактор 3	Фактор 1	Фактор 2	Фактор 3
In1	0,9724	-0,1801	0,0396	-0,9819	-0,0313	0,0447	0,9939	0,0003	0,0008
In2	-0,9714	0,1879	-0,0399	0,9833	0,0245	-0,0418	-0,9944	0,0068	-0,0036
In3	0,9362	-0,2833	0,0201	-0,9785	0,0671	0,0255	0,9768	-0,1016	0,0169
In4	-0,9552	0,2225	0,0318	0,9849	-0,0225	-0,0957	-0,9844	0,0560	0,0517
In5	0,2154	0,9427	0,2228	-0,0020	-0,9812	0,1207	0,0572	0,9878	-0,0917
In6	-0,2172	-0,9395	-0,2032	0,0177	0,9833	-0,1407	-0,0618	-0,9864	0,1107
In7	-0,2716	-0,4819	0,8322	0,0904	0,2090	-0,9718	-0,1503	-0,2347	0,9602
In19	-0,8840	-0,1739	-0,0717	0,7190	0,2778	-0,1112	-0,8242	-0,2803	0,0700
In20	0,8861	0,1622	0,0054	-0,7186	-0,2521	0,1494	0,8259	0,2544	-0,1117

Таким образом, первому фактору соответствуют высокие нагрузки по переменным In1-In4, In19, In20 (рисунки 2.12, 2.13, значения по модулю). Этим фактором описывается 54,5-60,1% всей дисперсии (в зависимости от метода выделения и вращения). Второму фактору соответствуют высокие нагрузки по переменным In5, In6. Этим фактором описывается еще 23,2-25,1% всей дисперсии. При выделении факторов методом главных компонент третьему фактору соответствует высокая нагрузка по единственной переменной In7; этот фактор описывает еще 8,8-11,4% общей дисперсии всех переменных. Суммарно двухфакторное решение имеет кумулятивную долю дисперсии 80,2-83,7%, трехфакторное решение – 89,4-94,0%. Остальные факторы незначимы и в последующем анализе не учитываются.

Таблица 2.11 – Факторные нагрузки (метод главных осей). Матрица Спирмена

	Без вращения		Варимакс нормализованных		Квартимакс нормализованных	
	Фактор 1	Фактор 2	Фактор 1	Фактор 2	Фактор 1	Фактор 2
In1	0,9803	-0,1634	-0,9952	-0,0212	0,9951	-0,0163
In2	-0,9776	0,1705	0,9939	0,0144	-0,9933	0,0234
In3	0,9414	-0,2670	-0,9792	0,0777	0,9702	-0,1188
In4	-0,9611	0,2093	0,9774	-0,0474	-0,9777	0,0778
In5	0,2100	0,9618	-0,0249	-0,9716	0,0781	0,9839
In6	-0,2120	-0,9576	0,0287	0,9686	-0,0790	-0,9809
In7	-0,2369	-0,3421	0,1139	0,2403	-0,1732	-0,2852
In19	-0,8558	-0,1776	0,7858	0,2817	-0,8234	-0,2712
In20	0,8590	0,1645	-0,7811	-0,2437	0,8255	0,2416

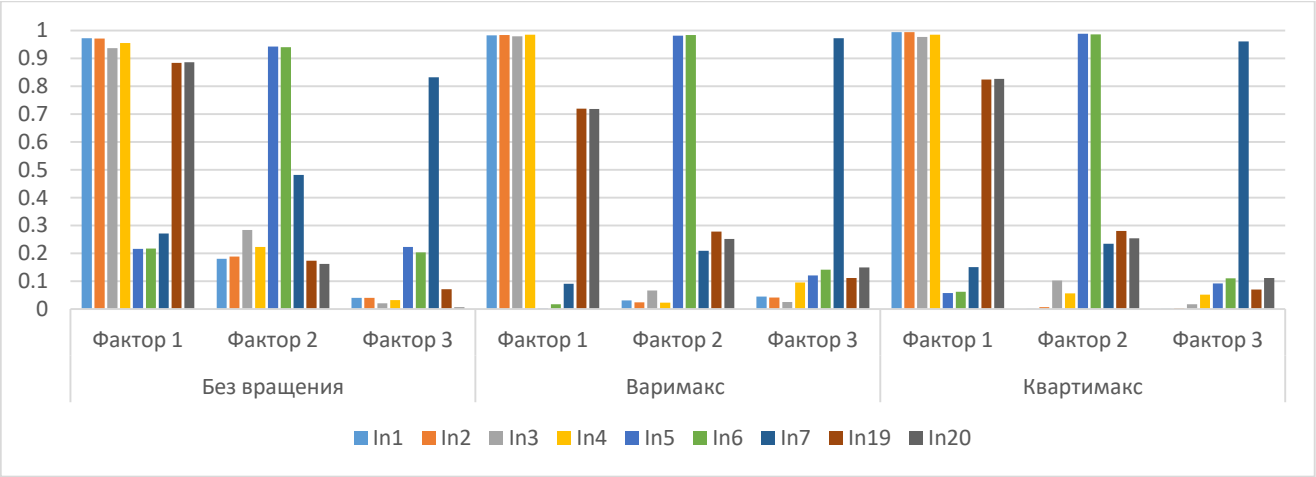


Рисунок 2.12 – Факторные нагрузки (метод главных компонент). Матрица Спирмена

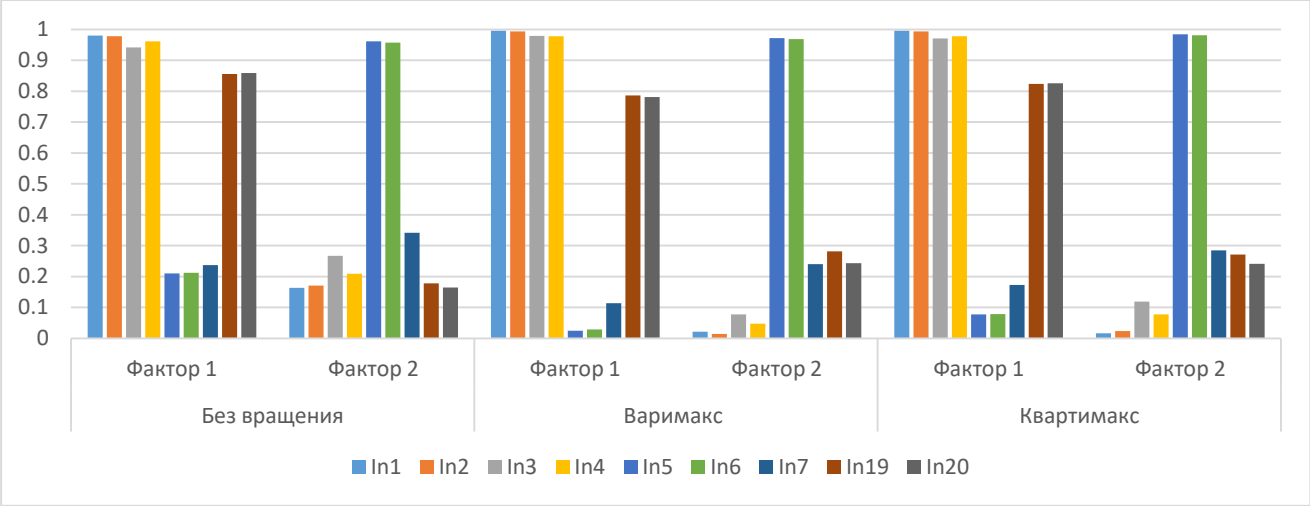


Рисунок 2.13 – Факторные нагрузки (метод главных осей). Матрица Спирмена

Как видно из рисунков 2.7 и 2.14, разделение между двумя партиями происходит исключительно по первому фактору.

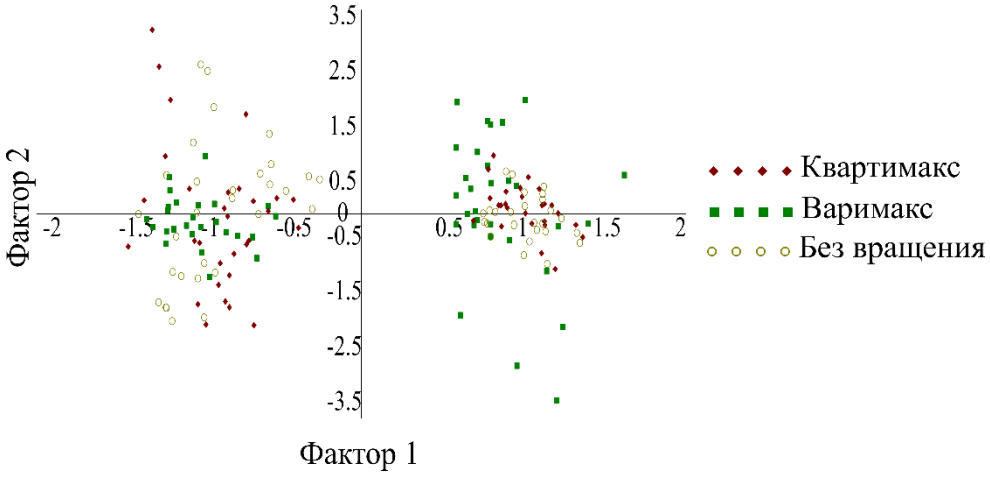


Рисунок 2.14 – Точки измерений методом главных компонент в факторном пространстве с разными вариантами вращения. Матрица Спирмена

Матрица Пирсона

В таблицах 2.12 и 2.13 приведены собственные значения факторов после вращения, дисперсия (в процентах), описываемая данными факторами по всем переменным, а также накопительная дисперсия по всем переменным.

Таблица 2.12 – Вращение (метод главных компонент). Матрица Пирсона

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	5,0305	5,4505	5,6047	55,894	60,561	62,274	55,894	60,561	62,274
2	2,0523	2,0788	2,1333	22,803	23,098	23,703	78,697	83,659	85,978
3	1,0307	0,9044	0,7365	11,452	10,049	8,183	90,150	93,707	94,161
4	0,5817	0,3250	0,2949	6,464	3,611	3,276	96,613	97,318	97,437
5	0,2649	0,2016	0,1914	2,943	2,240	2,127	99,556	99,558	99,564
6	0,0262	0,0261	0,0258	0, 291	0,290	0,286	99,847	99,848	99,850
7	0,0094	0,0096	0,0104	0,104	0,107	0,116	99,951	99,954	99,966
8	0,0044	0,0041	0,0031	0, 049	0,045	0,034	100,00	100,000	100,00
9	0,0000	0,0000	0,0000	0,000	0,000	0,000	100,00	100,000	100,00

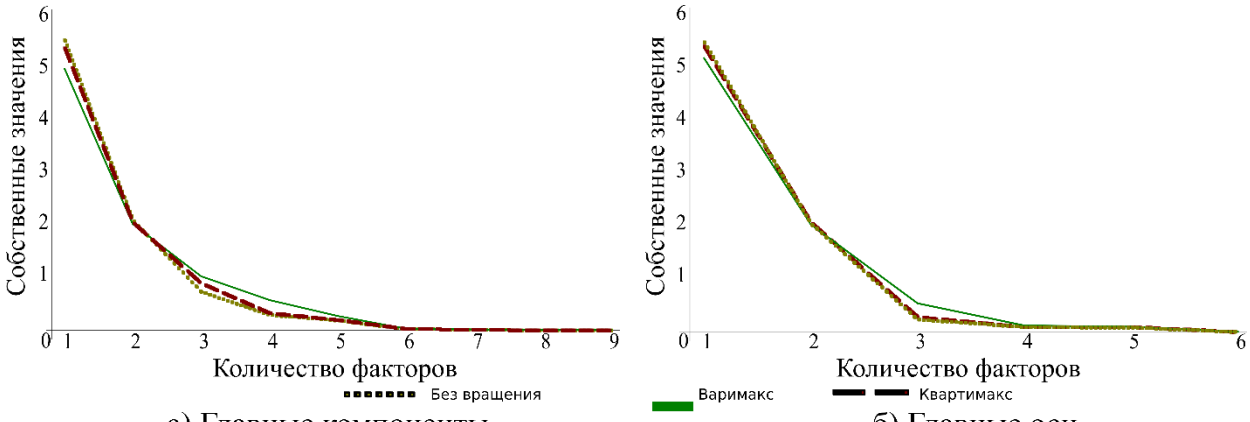
Таблица 2.13 – Вращение (метод главных осей). Матрица Пирсона

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	5,2215	5,4579	5,5382	58,017	60,644	61,535	58,017	60,644	61,535
2	2,0443	2,0944	2,0718	22,714	23,271	23,020	80,731	83,914	84,555
3	0,5438	0,2878	0,2406	6,042	3,198	2,674	86,773	87,112	87,229
4	0,1222	0,0972	0,0967	1,357	1,080	1,074	88,130	88,191	88,303
5	0,0990	0,0936	0,0836	1,100	1,040	0,093	89,231	89,231	89,232
6	0,0033	0,0033	0,0032	0,0369	0,037	0,036	89,268	89,268	89,268

При варимакс-вращении факторов, выделенных методом главных компонент, собственное значение третьего фактора превышает единицу, а значит, также может быть включено в модель (рисунок 2.15); при этом суммарная дисперсия, описываемая всеми факторами, составляет при варимакс-вращении 90,15%, при квартимакс-вращении – 93,71%. Кумулятивная доля остальных факторов, не включенных в модель, при любом вращении не превышает 10% от общей

дисперсии. Кроме того, при вращении из первого фактора исключается характеристика In19.

При вращении факторов, выделенных методом главных осей, выделение третьего фактора по рассмотренным выше критериям нецелесообразно (рисунок 2.13); при этом решение, полученное при варимакс-вращении, обеспечивает объяснение 80,73% общей дисперсии, при квартимакс – вращении – 83,91%.



а) Главные компоненты б) Главные оси

Рисунок 2.15 – График «каменистой осыпи». Матрица Пирсона

Значения факторных нагрузок приведены в таблицах 2.14 и 2.15. Выделены нагрузки со значениями >0,7.

Таблица 2.14 – Факторные нагрузки (метод главных компонент). Матрица Пирсона

	Без вращения			Варимакс нормализованных			Квартимакс нормализованных		
	Фактор 1	Фактор 2	Фактор 3	Фактор 1	Фактор 2	Фактор 3	Фактор 1	Фактор 2	Фактор 3
In1	0,9810	-0,1519	0,0360	-0,9809	-0,0485	0,1092	-0,9969	-0,0217	0,0349
In2	-0,9805	0,1553	-0,0341	0,9816	0,0450	-0,1099	0,9970	0,0181	-0,0357
In3	0,9650	-0,2192	0,0185	-0,9844	0,01934	0,1003	-0,9931	0,0473	0,0285
In4	-0,9644	0,2127	0,0199	0,9790	-0,0246	-0,1377	0,9890	-0,0515	-0,0665
In5	0,2157	0,9338	0,2523	-0,0192	-0,9879	0,0718	-0,0647	-0,9905	0,0373
In6	-0,2464	-0,9505	-0,1214	0,0330	0,9699	-0,1990	0,083965	0,9756	-0,1643
In7	-0,4146	-0,4389	0,7962	0,1884	0,2253	-0,9536	0,2729	0,2492	-0,9291
In19	-0,8482	-0,1543	-0,0843	0,6617	0,2431	-0,1323	0,7826	0,2497	-0,0709
In20	0,9061	-0,0337	0,1179	-0,8382	-0,1426	0,0776	-0,8946	-0,1256	0,0067

Таблица 2.15 – Факторные нагрузки (метод главных осей). Матрица Пирсона

	Без вращения		Варимакс нормализованных		Квартимакс нормализованных	
	Фактор 1	Фактор 2	Фактор 1	Фактор 2	Фактор 1	Фактор 2
In1	0,9864	-0,1358	-0,9856	-0,0423	-0,9961	-0,0075
In2	-0,9858	0,1394	0,9857	0,0386	0,9960	0,0038
In3	0,9719	-0,2053	-0,9864	0,0264	-0,9914	0,0624
In4	-0,9711	0,2006	0,9743	-0,0376	0,9872	-0,0670
In5	0,2114	0,9487	-0,0359	-0,9869	-0,0812	-0,9860
In6	-0,2415	-0,9614	0,0272	0,9509	0,0998	0,9718
In7	-0,3711	-0,3211	0,2200	0,2629	0,3031	0,3046
In19	-0,8175	-0,1542	0,7378	0,2628	0,7861	0,2508
In20	0,8926	-0,0138	-0,8727	-0,1499	-0,8887	-0,1172

Таким образом, первому фактору соответствуют высокие нагрузки по переменным In1-In4, In19, In20 (рисунки 2.16 и 2.17, значения по модулю). Этим фактором описывается 55,9-62,3% всей дисперсии (в зависимости от метода выделения и вращения). Второму фактору соответствуют высокие нагрузки по переменным In5, In6. Этим фактором описывается еще 22,7-23,7% всей дисперсии. При выделении факторов методом главных компонент третьему фактору соответствует высокая нагрузка по единственной переменной In7; этот фактор описывает еще 8,2-11,5% общей дисперсии всех переменных. Суммарно двухфакторное решение имеет кумулятивную долю дисперсии 78,7-86%, трехфакторное решение – 90,2-94,2%. Остальные факторы незначимы и в последующем анализе не учитываются.

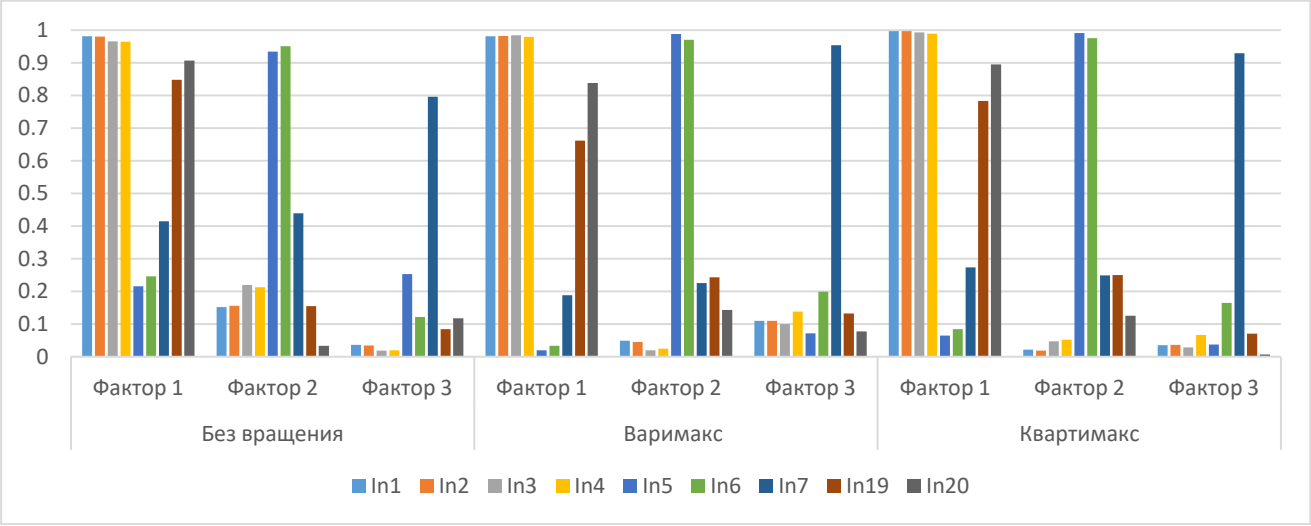


Рисунок 2.16 – Факторные нагрузки (метод главных компонент). Матрица Пирсона

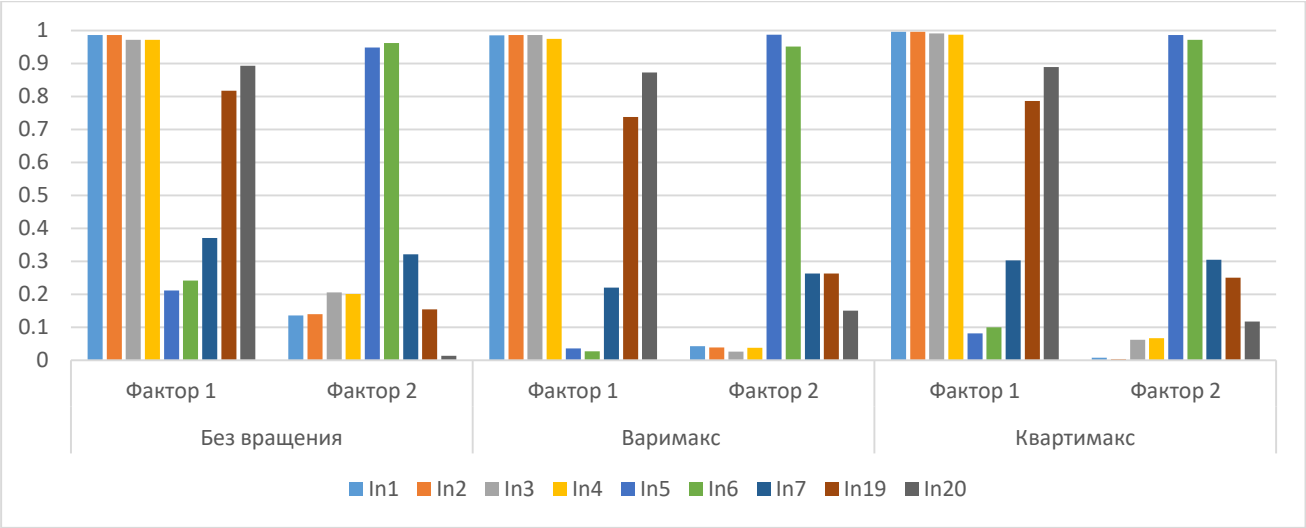


Рисунок 2.17 – Факторные нагрузки (метод главных осей). Матрица Пирсона

Как видно из рисунков 2.8 и 2.18, разделение между двумя партиями происходит исключительно по первому фактору.

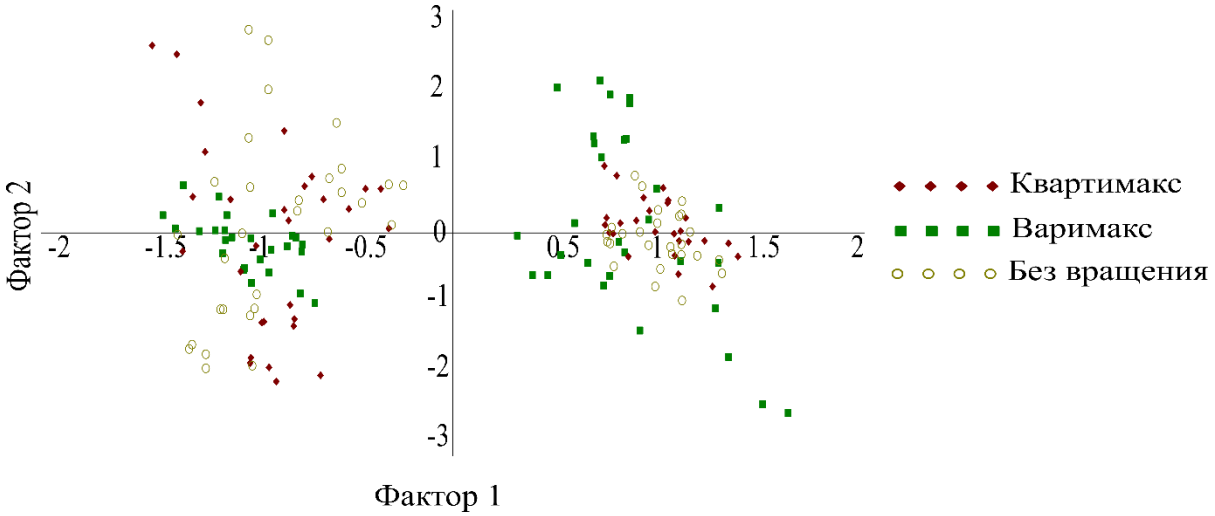


Рисунок 2.18 – Точки измерений методом главных компонент в факторном пространстве с разными вариантами вращения. Матрица Пирсона

Проведя сравнительный анализ результатов выделения общих факторов на основе матриц из коэффициентов корреляции Спирмена и Пирсона приходим к следующему заключению. Вне зависимости от способа получения коэффициентов исходной матрицы корреляций получаем один набор общих факторов при различных вариантах вращения. Согласно рисункам 2.12, 2.13, 2.16, 2.17 каждому фактору соответствует определенный набор переменных, которые оказывают высокие нагрузки на факторы. Также разделение на однородные партии происходит по одному общему первому фактору.

Б) Выборка II

Для предварительной оценки числа выделяемых факторов использовали корреляционную матрицу по Пирсону. Факторному анализу подвергались различные комбинации партий: полная выборка и частичная смешанная выборка из четырех, трех и двух партий. Полная выборка состоит из семи однородных партий. Смешанная выборка из четырех партий состоит из партии 1, партии 2, партии 5 и партии 6. Смешанная выборка из трех партий состоит из партии 1, партии 2 и партии 6. Смешанная выборка из двух партий состоит из партии 1 и партии 2.

Число факторов было определено по критерию Кайзера, и общая доля дисперсии, воспроизводимой этими факторами, должна составлять не менее 70%. Для извлечения факторов использовали метод главных компонент, метод главных факторов, метод главных осей, метод максимальных правдоподобий, метод MINRES и метод центроида. В дальнейшем мы использовали метод главных компонентов, поскольку он описывает максимальную дисперсию входных характеристик.

Проведены вычислительные эксперименты с различным сочетанием партий.

Полная выборка. Для полной выборки метод, основанный на критерии Кэттела, рекомендует выбрать 4 фактора в модели, и это число не изменяется при любой процедуре вращения (рисунок 2.19). В соответствии с критерием Кайзера, принимая во внимание общую дисперсию не менее 70%, выбирается пять факторов. К. Иберла [146] рекомендует в случаях спора выбирать большее число, поэтому мы выделяем 5 факторов для дальнейшего рассмотрения. Фактор 1 соответствует наибольшим нагрузкам для характеристик In92-In107. Этот фактор описывает 22,779-23,954% от общей дисперсии. Фактор 2 соответствует наибольшим нагрузкам по характеристикам In58-In64, In76-In82. Этот фактор описывает дополнительно 19,335-21,265% от общей дисперсии. Фактор 3 соответствует наибольшим нагрузкам на характеристики In39-In46. Этот фактор

описывает дополнительные 12 300-14,776% от общей дисперсии. Фактор 4 (характеристики In10, In11, In13, In 14, In18) описывает 9,003-9,375% от общей дисперсии. Фактор 5 (характеристики In21 – In28) описывает 6,781% (без вращения), 11,928% (варимакс) и 11,993% (квартамакс) от общей дисперсии. Независимо от метода вращения суммарный процент от общей дисперсии составляет 75,794% (таблица 2.16).

Таблица 2.16. Вращение. Полная выборка

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	15,262	15,3670	16,049	22,779	22,936	23,954	22,779	22,936	23,954
2	12,955	13,1069	14,248	19,335	19,562	21,265	42,115	42,498	45,219
3	8,2927	8,2411	9,8998	12,377	12,300	14,776	54,492	54,799	59,995
4	6,2810	6,0322	6,0424	9,375	9,003	9,018	63,867	63,802	69,013
5	7,9917	8,0351	4,5433	11,928	11,993	6,781	75,794	75,794	75,794

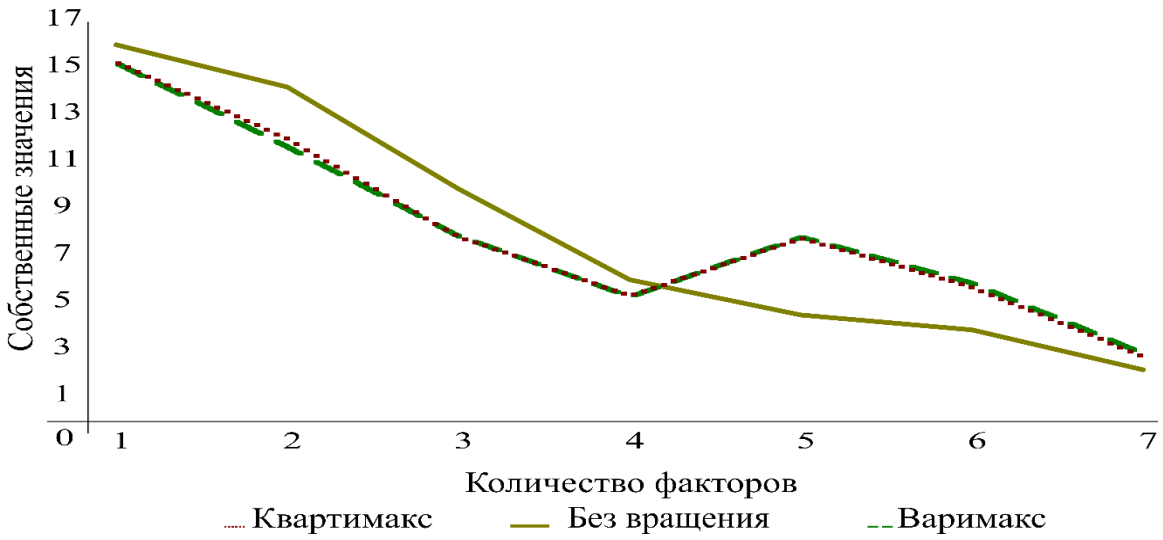


Рисунок 2.19 – График «каменистой осыпи». Полная выборка

Частичная смешанная выборка из четырех партий. Общее количество изделий в четырех партиях составляет 446. В этой комбинации мы исключаем еще пять входных характеристик, поскольку они имеют нулевые векторы данных или их число ненулевых значений не превышает 10%. Для дальнейшей обработки осталось 62 входных характеристики. Критерий Каттеля, независимо от способа

вращения, рекомендует выбрать 4 фактора в модели (рисунок 2.20), однако, согласно критерию Кайзера, с учетом общего процента отклонений не менее 70% мы выделяем 6 факторов. Фактор 1 показывает самые высокие нагрузки для характеристик In21 – In28, In39 – In46. Этот фактор описывает 23,304-38,622% от общей дисперсии. Фактор 2 показывает существенные нагрузки для характеристик In58-In 64, он описывает дополнительно 13 220-17,761% от общей дисперсии. Фактор 3 имеет существенные нагрузки для In91-In107, фактор 4 для In79 – In82, фактор 6 для In57, In78. Независимо от способа вращения суммарный процент от общей дисперсии составляет 70,364% (таблица 2.17).

Таблица 2.17. Вращение. Выборка из четырех партий

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения
1	15,069	16,9248	23,946	24,304	27,298	38,622	24,304	27,298	38,622
2	11,012	10,3553	8,1961	17,761	16,702	13,220	42,066	44,000	51,841
3	12,274	11,4133	6,9643	19,796	18,409	11,233	61,862	62,409	63,074
4	2,4193	2,0711	1,8951	3,902	3,340	3,057	65,764	65,749	66,131
5	1,4149	1,4754	1,3752	2,282	2,380	2,218	68,046	68,129	68,349
6	1,4373	1,3861	1,2497	2,318	2,236	2,016	70,364	70,364	70,364

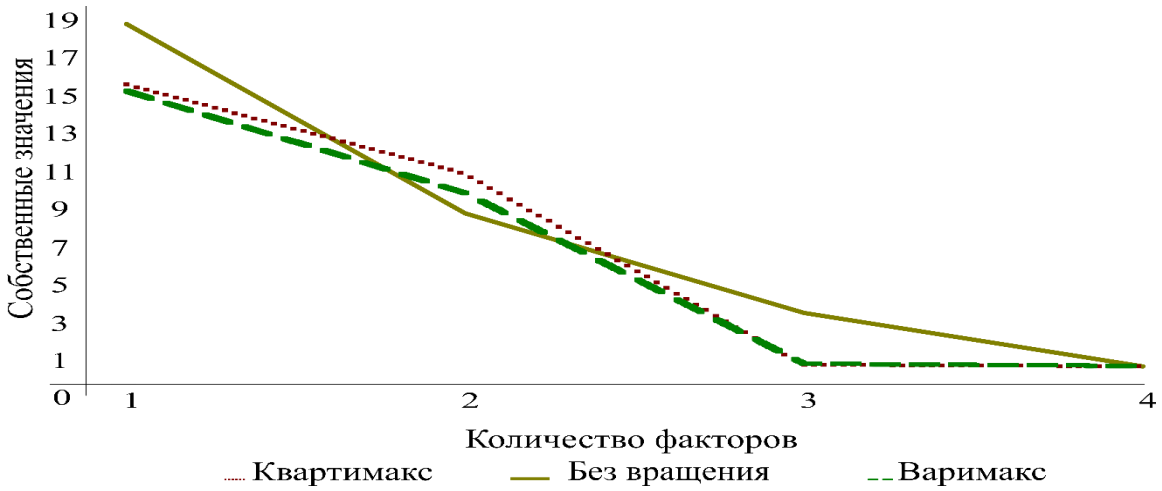


Рисунок 2.20 – График «каменистой осыпи». Выборка из четырех партий

Частичная смешанная выборка из трех партий. Общее количество изделий в трех партиях составляет 300. После исключения характеристик с

нулевыми векторами данных или векторов, заполненных данными менее чем на 10%, для дальнейшей обработки остается 41 характеристика. Критерий Кателя, независимо от способа вращения, рекомендует выбрать 3 фактора в модели (рисунок 2.21). Согласно критерию Кайзера, принимая во внимание общий процент отклонения не менее 70%, мы также выделяем 3 фактора. Фактор 1 показывает самые высокие нагрузки для характеристик In21 – In28, In39 – In46. Этот фактор описывает 37,089-46,382% от общей дисперсии. Фактор 2 имеет значительные нагрузки для In92-In107 и описывает 22 031-26,612% от общей дисперсии. Фактор 3 имеет существенные нагрузки для In84-In91 и описывает дополнительно 9,192-13,905% от общей дисперсии. Независимо от способа вращения суммарный процент от общей дисперсии составляет 77,606% (таблица 2.18).

Таблица 2.18. Вращение. Выборка из трех партий

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс
1	15,206	15,885	19,017	37,089	38,744	46,382	37,089	38,744	46,382
2	10,911	10,892	9,0329	26,612	26,566	22,031	63,701	65,311	68,413
3	5,7009	5,0409	3,7689	13,905	12,295	9,192	77,606	77,606	77,606

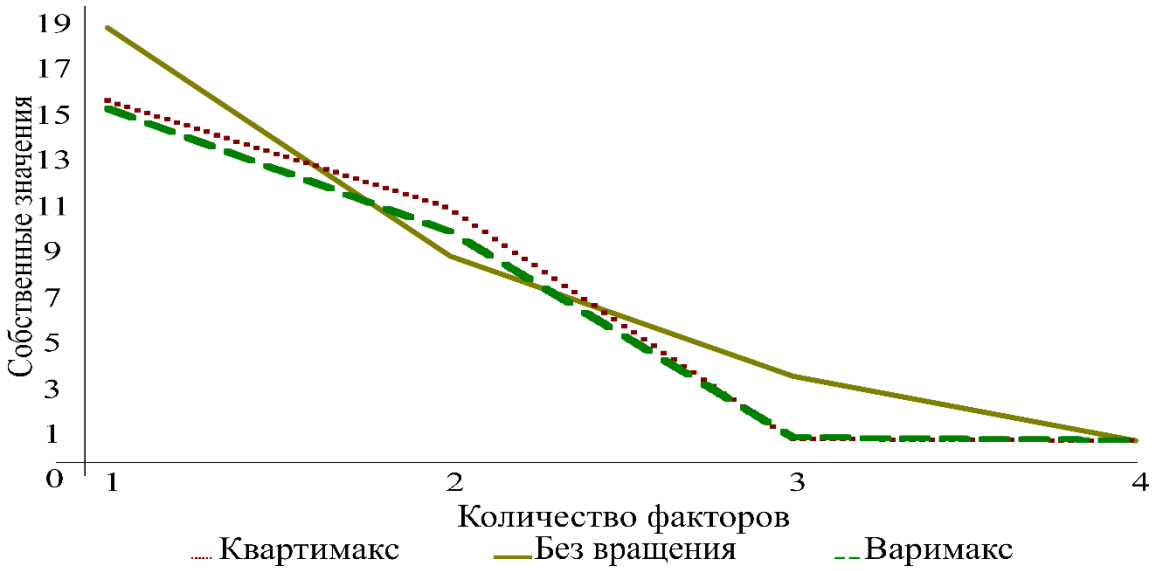


Рисунок 2.21 – График «каменистой осыпи». Выборка из трех партий

Частичная смешанная выборка из двух партий. Количество изделий в двух партиях составляет 187. После исключения характеристик с нулевыми векторами данных или векторов, заполненных данными менее чем на 10%, для дальнейшей обработки остается 41 характеристика. Критерий Кателя, независимо от способа вращения, рекомендует выбрать 2 фактора в модели (рисунок 2.22), и в соответствии с критерием Кайзера, учитывая общий процент отклонения не менее 70%, мы также выделяем 2 фактора. Фактор 1 показывает самые высокие нагрузки для характеристик In21 – In28, In39 – In46 и описывает 45,410-66,195% от общей дисперсии. Фактор 2 показывает самые высокие нагрузки для In92-In95, In100-In102, In106 и описывает дополнительно 7,282-28,067% от общей дисперсии. Независимо от способа вращения суммарный процент от общей дисперсии составляет 73,478% (таблица 2.19).

Таблица 2.19. Вращение. Выборка из трех партий

Факторы	Собственные значения			Доля от общей дисперсии (%)			Кумулятивная доля от общей дисперсии (%)		
	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс	Без вращения	Варимакс	Квартимакс
1	18,618	26,614	27,1401	45,410	64,912	66,195	45,410	64,912	66,195
2	11,508	3,5121	2,9858	28,067	8,566	7,282	73,478	73,478	73,478

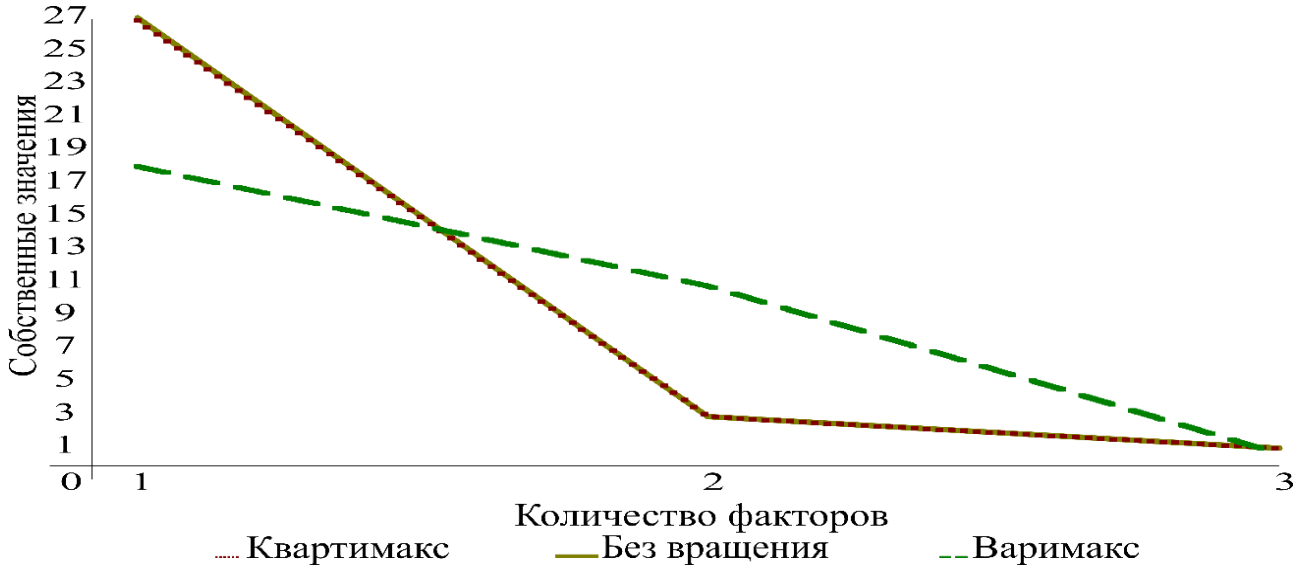


Рисунок 2.22 – График «каменистой осыпи». Выборка из двух партий

Проверка адекватности факторной модели

Проверка адекватности факторной модели сводится к проверке достижения простой структуры. Под простой структурой понимается такая конфигурация векторов, которая путем вращения позволяет достигнуть такого состояния, что значительное большинство векторов окажется на гиперплоскостях координат или вблизи них [143]. Кроме того, простая структура «контрастна»: факторные нагрузки высоки для переменных, определяющих данный фактор, и близки к нулю для всех остальных. При оценке достижения простой структуры пользуются тестом Баргмана [149] (2.10). Для каждого фактора необходимо рассчитать количество нулевых нагрузок.

Выборка I

Матрица Спирмена

В нашем случае (таблица 2.20) тест Баргмана для α=0,05 выполняется лишь в трех случаях. Однако, анализ количества переменных с нулевыми нагрузками в процентах от общего числа переменных показывает, что в общем случае при вращении факторная структура приближается к простой для факторов 2 и 3. При повышении уровня α до 0,25 тест Баргмана выполняется для факторов 2 и 3 при любом вращении.

Таблица 2.20 – Критерий Баргмана. Матрица Спирмена

			Табличное значение для α=0,05	Табличное значение для α=0,25	Количество нулевых нагрузок	Процент нулевых нагрузок
Главные компоненты	без вращения	Фактор 1	5	4	0	0%
		Фактор 2			0	0%
		Фактор 3			6	67%
	варимакс норм.	Фактор 1			3	33%
		Фактор 2			4	44%
		Фактор 3			4	44%
	квартимакс норм.	Фактор 1			2	22%
		Фактор 2			4	44%
		Фактор 3			6	67%

Продолжение таблицы 2.20

			Табличное значение для α=0,05	Табличное значение для α=0,25	Количество нулевых нагрузок	Процент нулевых нагрузок
Главные оси	без вращения	Фактор 1	4	3	0	0%
		Фактор 2			0	0%
	варимакс норм.	Фактор 1			2	22%
		Фактор 2			4	44%
	квартимакс норм.	Фактор 1			2	22%
		Фактор 2			3	33%

Матрица Пирсона

Тест Баргмана для α=0,05 выполняется лишь в четырех случаях. Однако, анализ количества переменных с нулевыми нагрузками в процентах от общего числа переменных показывает, что в общем случае при вращении факторная структура приближается к простой для факторов 2 и 3 (таблица 2.21). При повышении уровня α до 0,25 тест Баргмана выполняется для факторов 2 и 3 при любом вращении.

Таблица 2.21 – Критерий Баргмана. Матрица Пирсона

			Табличное значение для α=0,05	Табличное значение для α=0,25	Количество нулевых нагрузок	Процент нулевых нагрузок
Главные компоненты	без вращения	Фактор 1	5	4	0	0
		Фактор 2			1	11%
		Фактор 3			5	56%
	варимакс норм.	Фактор 1			2	22%
		Фактор 2			4	44%
		Фактор 3			3	33%
	квартимакс норм.	Фактор 1			2	22%
		Фактор 2			4	44%
		Фактор 3			**7**	78%
Главные оси	без вращения	Фактор 1	4	3	0	0%
		Фактор 2			1	11%
	варимакс норм.	Фактор 1			1	11%
		Фактор 2			4	44%
	квартимакс норм.	Фактор 1			2	22%
		Фактор 2			4	44%

Выборка II

Для полной выборки критерий Баргмана удовлетворяет 3 из 5 факторов без вращения факторной структуры и для всех факторов в случае вращения факторной структуры для α<=0,05. Для выборки из четырех партий тест удовлетворяет 3 из 6 факторов в случае без вращения структуры и 4 из 6 факторов в случае вращения факторной структуры для α<= 0,25. Для выборки, состоящей из трех партий, тест удовлетворяет только 1-му фактору без вращения факторной структуры и 2 факторам в случае вращения факторной структуры для α<= 0,25. Для выборки, состоящей из двух партий, тест удовлетворяет в одном случае для α <= 0,25 (таблица 2.22).

Таблица 2.22 – Критерий Баргмана

		Фактор	Табличное значение для α=0,05	Табличное значение для α<=0,25	Количество нулевых нагрузок	Процент нулевых нагрузок
Полная выборка (67 характеристик)	без вращения	Фактор 1	17	14	9	13%
		Фактор 2			6	8%
		Фактор 3			18	27%
		Фактор 4			31	46%
		Фактор 5			26	39%
	варимакс	Фактор 1	17	14	43	64%
		Фактор 2			26	39%
		Фактор 3			33	49%
		Фактор 4			33	49%
		Фактор 5			42	63%
	квартимакс	Фактор 1			41	61%
		Фактор 2			24	36%
		Фактор 3			33	49%
		Фактор 4			33	49%
		Фактор 5			43	64%
Четыре партии (62 характеристики)	без вращения	Фактор 1	20	17	3	5%
		Фактор 2			8	13%
		Фактор 3			14	23%
		Фактор 4			45	73%
		Фактор 5			36	58%
		Фактор 6			51	82%
	варимакс	Фактор 1			14	23%
		Фактор 2			10	16%
		Фактор 3			18	29%
		Фактор 4			44	71%
		Фактор 5			37	60%
		Фактор 6			51	82%

Продолжение таблицы 2.22

		Фактор	Табличное значение для α=0,05	Табличное значение для α<=0,25	Количество нулевых нагрузок	Процент нулевых нагрузок
Четыре партии (62 характеристики)	квартимакс	Фактор 1			14	23%
		Фактор 2			9	15%
		Фактор 3			17	27%
		Фактор 4			52	84%
		Фактор 5			36	58%
		Фактор 6			50	81%
Три партии (41 характеристика)	без вращения	Фактор 1	9	7	0	0%
		Фактор 2			0	0%
		Фактор 3			8	20%
	варимакс	Фактор 1			7	17%
		Фактор 2			0	0%
		Фактор 3			10	24%
	квартимакс	Фактор 1			7	17%
		Фактор 2			0	0%
		Фактор 3			15	37%
Две партии (41 характеристика)	без вращения	Фактор 1	6	4	0	0%
		Фактор 2			3	7%
	варимакс	Фактор 1			1	2%
		Фактор 2			0	0%
	квартимакс	Фактор 1			0	0%
		Фактор 2			5	12%

Анализ процента нулевых нагрузок показывает, что с увеличением количества партий и при любом повороте увеличивается число случаев, для которых выполняется тест Баргмана.

Значения факторов, полученные с помощью описанных выше ортогональных вращений, рассматриваются как входные данные для алгоритмов кластеризации. Кластеризация проводилась с помощью ПО Deductor Studio Academic [176]. Применен метод g-средних (с уровнем значимости 0,1%), метод EM (нижний порог правдоподобия 0,2, уровень точности модели 10^{-5}, максимальное количество итераций 100) и метод карт Кохонена (Self-organizing map, SOM) (начальная инициализация карты из собственных векторов, функция соседства ступенчатая, уровень значимости 0,1%) [177].

А) Выборка I

Матрица Спирмена

В случае применения алгоритма EM непосредственно к исходным данным, первая партия разбилась на три кластера, вторая – на два кластера. При этом изделия из разных партий не пересекались. При применении метода карт Кохонена, первая партия разбилась на два кластера, вторая партия целиком отнесена к третьему кластеру. При кластеризации двухфакторной модели алгоритм EM не смог разделить данные, алгоритм g-средних показал 100%-ный результат при любом вращении. В случае трехфакторной модели верного разбиения удалось добиться для партии 2. Партию 1 в большинстве случаев алгоритмы разбивали на два кластера (таблица 2.23).

Таблица 2.23 – Результаты кластеризации партий ЭРИ. Матрица Спирмена

	Партия 1 (30шт)		Партия 2 (26 шт)	
	Кластеризова-но верно, шт. (доля)	Кластеризова-но ошибочно, шт. (доля)	Кластеризова-но верно, шт. (доля)	Кластеризова-но ошибочно, шт. (доля)
	исходные данные			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	13 (0,43)	17 (0,57)	16 (0,62)	10 (0,38)
SOM	18 (0,60)	12 (0,40)	26 (1,00)	0 (0,00)
	2 фактора без вращения			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	-	-	-	-
SOM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
	2 фактора варимакс			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	-	-	-	-
SOM	11 (0,37)	19 (0,63)	26 (1,00)	5 (0,19)
	2 фактора квартимакс			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	-	-	-	-
SOM	13 (0,43)	17 (0,57)	26 (1,00)	0 (0,00)
	3 фактора без вращения			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
SOM	18 (0,60)	12 (0,40)	26 (1,00)	0 (0,00)
	3 фактора варимакс			
g-средних	19 (0,63)	11 (0,37)	26 (1,00)	0 (0,00)
EM	17 (0,57)	13 (0,43)	26 (1,00)	0 (0,00)
SOM	11 (0,37)	19 (0,63)	24 (0,92)	2 (0,80)

Продолжение таблицы 2.23

	Партия 1 (30шт)		Партия 2 (26 шт)	
	Кластеризова-но верно, шт. (доля)	Кластеризова-но ошибочно, шт. (доля)	Кластеризова-но верно, шт. (доля)	Кластеризова-но ошибочно, шт. (доля)
	3 фактора квартимакс			
g-средних	19 (0,63)	11 (0,37)	26 (1,00)	0 (0,00)
EM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
SOM	14 (0,47)	16 (0,53)	26 (1,00)	0 (0,00)

Матрица Пирсона

При применении алгоритма EM первая партия разбилась на три кластера, вторая – на два кластера. При этом изделия из разных партий не пересекались. Когда к исходным данным был применен метод карт Кохонена, первая партия разбилась на два кластера, вторая партия целиком отнесена к третьему кластеру (таблица 2.24).

При кластеризации двухфакторной модели алгоритм EM не смог разделить данные. В случае трехфакторной модели верного разбиения удалось добиться любым из трех методов.

Таблица 2.24 – Результаты кластеризации партий ЭРИ. Матрица Пирсона

	Партия 1 (30шт)		Партия 2 (26 шт)	
	Кластеризовано верно, шт. (доля)	Кластеризовано ошибочно, шт. (доля)	Кластеризовано верно, шт. (доля)	Кластеризовано ошибочно, шт. (доля)
	исходные данные			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	13 (0,43)	17 (0,57)	16 (0,62)	10 (0,38)
SOM	18 (0,60)	12 (0,40)	26 (1,00)	0 (0,00)
	2 фактора			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	-	-	-	-
SOM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
	3 фактора			
g-средних	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
EM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)
SOM	30 (1,00)	0 (0,00)	26 (1,00)	0 (0,00)

Б) Выборка II

Точность кластеризации для рассматриваемых смешанных партий с различными ортогональными вращениями представлена в таблице 2.25.

При варимакс-вращении в случае кластеризации полной выборки по алгоритму EM получаем точность 0,39, по SOM точность равна 0,59. В случае частичной смешанной выборки с четырьмя партиями точность по EM составила 0,97, по SOM точность равна 0,89. В случае смешанной выборки из трех партий EM дает нам 0,91 точности, по SOM точность равна 0,68. А для алгоритма EM с двумя партиями смешанной выборки точность составила 0,94, по SOM точность равна 0,83.

При квартимакс-вращении в случае кластеризации полной выборки по алгоритму EM точность составляем 0,45, по SOM точность равна 0,50. В случае частичной смешанной выборки с четырьмя партиями точность по EM составила 0,97, по SOM точность равна 0,67. В случае смешанной выборки из трех партий EM дает нам 0,93 точности, SOM точность равна 0,82. Для алгоритма EM с двумя партиями смешанной выборки точность составила 0,94, SOM точность равна 0,83.

Без вращения в случае кластеризации полной выборки по алгоритму EM получаем точность 0,48, по SOM точность равна 0,70. В случае кластеризации по четырем партиям смешанная выборки по EM составила 0,97, по SOM точность равна 0,79. В случае смешанной выборки из трех партий EM дает нам точность 0,97, по SOM точность равна 0,83. Для алгоритма EM с двумя партиями смешанной выборки точность составила 0,98, по SOM точность равна 0,98.

Таблица 2.25 – Кластеризация полной и частично смешанной выборок

		Без вращения		Варимакс		Квартимакс	
		EM	SOM	EM	SOM	EM	SOM
Две партии, шт. (доля)							
Партия 1 (n=71)	Кластеризовано верно	71 (1,00)	71 (1,00)	71 (1,00)	42 (0,59)	71 (1,00)	71 (1,00)
	Кластеризовано ошибочно	0 (0,00)	0 (0,00)	0 (0,00)	29 (0,41)	0 (0,00)	0 (0,00)
Партия 2 (n=116)	Кластеризовано верно	113 (0,97)	113 (0,97)	104 (0,90)	114 (0,98)	116 (0,100)	116 (0,100)
	Кластеризовано ошибочно	3 (0,03)	3 (0,03)	12 (0,10)	2 (0,02)	0 (0,00)	0 (0,00)
RI		0,98	0,98	0,94	0,91	1,00	1,00
Три партии, шт. (доля)							
Партия 1 (n=71)	Кластеризовано верно	71 (1,00)	36 (0,51)	71 (1,00)	41 (0,58)	71 (1,00)	40 (0,56)
	Кластеризовано ошибочно	0 (0,0)	35 (0,49)	0 (0,00)	30 (0,42)	0 (0,00)	31 (0,44)

Продолжение таблицы 2.25

		Без вращения		Варимакс		Квартимакс	
		EM	SOM	EM	SOM	EM	SOM
Партия 2 (n=116)	Кластеризовано верно	113 (0,94)	110 (0,95)	110 (0,95)	60 (0,52)	111 (0,96)	116 (1,00)
	Кластеризовано ошибочно	3 (0,06)	6 (0,05)	6 (0,05)	56 (0,48)	5 (0,04)	0 (0,00)
Партия 6 (n=113)	Кластеризовано верно	106 (0,94)	102 (0,90)	93 (0,82)	102 (0,90)	98 (0,87)	91 (0,81)
	Кластеризовано ошибочно	7 (0,06)	11 (0,10)	20 (0,18)	11 (0,10)	15 (0,13)	22 (0,19)
RI		0,98	0,93	0,94	0,86	0,95	0,93
Четыре партии, шт. (доля)							
Партия 1 (n=71)	Кластеризовано верно	70 (0,99)	71 (1,00)	70 (0,99)	71 (1,00)	70 (0,99)	71 (1,00)
	Кластеризовано ошибочно	1 (0,01)	0 (0,00)	1 (0,01)	0 (0,00)	1 (0,01)	0 (0,00)
Партия 2 (n=116)	Кластеризовано верно	108 (0,93)	108 (0,93)	108 (0,93)	116 (1,00)	108 (0,93)	86 (0,74)
	Кластеризовано ошибочно	8 (0,07)	8 (0,07)	8 (0,07)	0 (0,00)	8 (0,07)	30 (0,26)
Партия 5 (n=146)	Кластеризовано верно	146 (1,00)	68 (0,41)	146 (1,00)	116 (0,79)	146 (1,00)	38 (0,26)
	Кластеризовано ошибочно	0 (0,00)	78 (0,59)	0 (0,00)	20 (0,21)	0 (0,00)	108 (0,74)
Партия 6 (n=113)	Кластеризовано верно	107 (0,95)	107 (0,95)	107 (0,95)	107 (0,95)	108 (0,96)	103 (0,91)
	Кластеризовано ошибочно	6 (0,05)	6 (0,05)	6 (0,05)	6 (0,05)	5 (0,04)	10 (0,09)
RI		0,98	0,938	0,98	0,977	0,98	0,89
Полная выборка, шт. (доля)							
Партия 1 (n=71)	Кластеризовано верно	68 (0,96)	71 (1,00)	70 (0,99)	71 (1,00)	63 (0,89)	71 (1,00)
	Кластеризовано ошибочно	3 (0,04)	0 (0,00)	1 (0,01)	0 (0,00)	8 (0,11)	0 (0,00)
Партия 2 (n=116)	Кластеризовано верно	0 (0,00)	0 (0,00)	60 (0,52)	0 (0,00)	81 (0,70)	0 (0,00)
	Кластеризовано ошибочно	116 (1,00)	116 (1,00)	56 (0,48)	116 (1,00)	35 (0,30)	116 (1,00)
Партия 3 (n=1867)	Кластеризовано верно	487 (0,35)	297 (0,16)	618 (0,33)	298 (0,16)	699 (0,37)	649 (0,35)
	Кластеризовано ошибочно	1380 (0,65)	1570 (0,84)	1249 (0,67)	1569 (0,84)	1168 (0,63)	1218 (0,65)
Партия 4 (n=1250)	Кластеризовано верно	467 (0,37)	0 (0,00)	462 (0,37)	0 (0,00)	458 (0,37)	159 (0,13)
	Кластеризовано ошибочно	783 (0,63)	1250 (1,00)	788 (0,63)	1250 (1,00)	792 (0,63)	1091 (0,87)
Партия 5 (n=146)	Кластеризовано верно	121 (0,83)	67 (0,46)	135 (0,92)	102 (0,70)	133 (0,91)	73 (0,50)
	Кластеризовано ошибочно	25 (0,17)	79 (0,54)	11 (0,08)	44 (0,30)	13 (0,09)	73 (0,50)

Продолжение таблицы 2.25

		Без вращения		Варимакс		Квартимакс	
		EM	SOM	EM	SOM	EM	SOM
Партия 6 (n=113)	Кластеризовано верно	107 (0,95)	113 (1,00)	107 (0,95)	113 (1,00)	105 (0,93)	113 (1,00)
	Кластеризовано ошибочно	6 (0,05)	0 (0,00)	6 (0,05)	0 (0,00)	8 (0,07)	0 (0,00)
Партия 7 (n=424)	Кластеризовано верно	314 (0,74)	0 (0,00)	256 (0,60)	0 (0,00)	255 (0,60)	284 (0,70)
	Кластеризовано ошибочно	110 (0,26)	424 (1,00)	168 (0,40)	424 (1,00)	169 (0,40)	140 (0,30)
RI		0,74	0,38	0,72	0,38	0,74	0,65

Результаты кластеризации из трех и двух партий смешанных выборок представлены на рисунках 2.23 и 2.24 соответственно. Разделение партий происходит исключительно на Факторе 1 в обоих случаях.

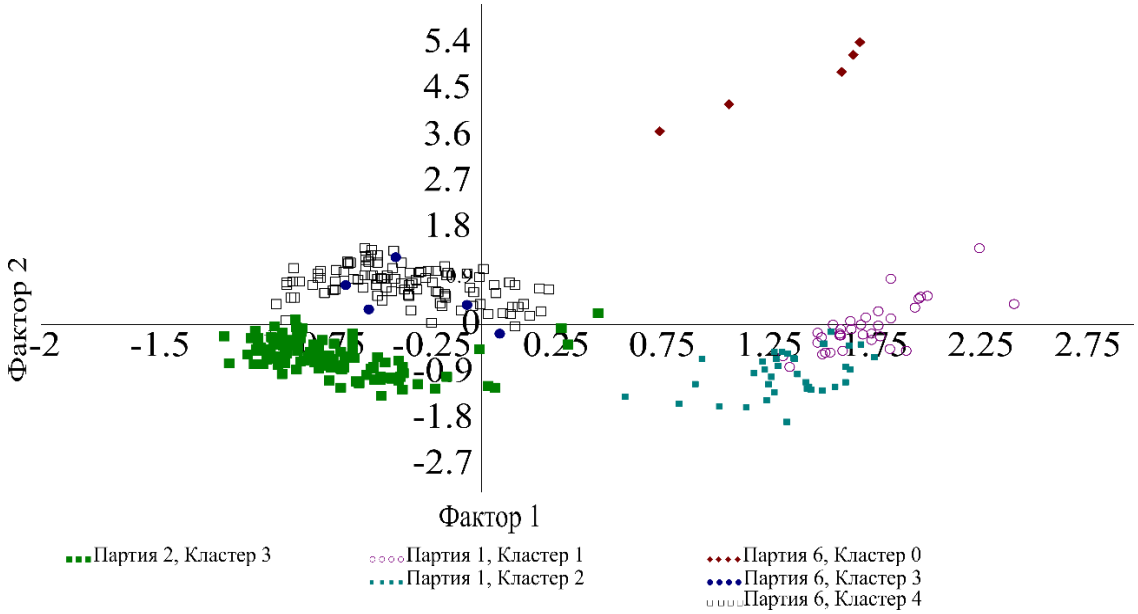


Рисунок 2.23 – Результаты кластеризации. Выборка из трех партий

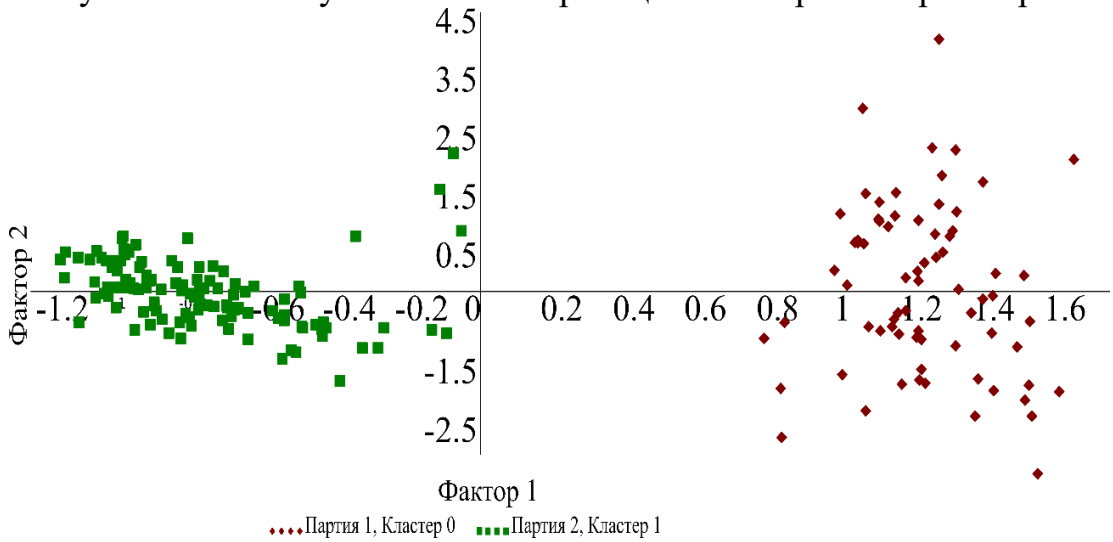


Рисунок 2.24 – Результаты кластеризации. Выборка из двух партий

В результате показано, что применение факторной модели действительно представляется целесообразным, по крайней мере – в случае разбиения набора данных на две однородные партии, так как повышает точность данного разбиения в случае использования различных алгоритмов кластеризации. Установлено, что две партии электрорадиоизделий могут быть с высокой точностью разделены с применением двухфакторной или трехфакторной модели на исходных данных матрицы корреляций по Спирмену. При этом, используя значения матрицы корреляций Пирсона, трехфакторная модель дала наилучшие результаты, показав 0% ошибок вне зависимости от используемого алгоритма кластеризации. Таким образом, в случае отклонения исходных характеристик от нормального распределения для факторной модели допустимо использование корреляционной матрицы Пирсона вместо корреляционной матрицы Спирмена.

Также было установлено, что оптимальное количество рассматриваемых факторов в выборке, состоящей из 3987 изделий, зависит от количества рассматриваемых устройств в смешанной партии, а также от входных измеренных характеристик устройства в данном образце. Независимо от типа ортогонального вращения точность кластеризации уменьшается с увеличением количества однородных партий в смешанной партии. В то же время невозможно получить универсальный набор с небольшим количеством признаков для разделения смешанной партии, состоящей из произвольного числа однородных партий. Таким образом, несмотря на то, что предлагаемый метод позволяет несколько уменьшить размерность данных, для надежного разделения однородных партий методами кластерного анализа использование многомерных данных неизбежно.

2.4 Результаты экспериментов по применению жадных эвристических алгоритмов с особыми функциями расстояний для задач автоматической группировки продукции

В данном разделе описано исследование эффективности применения жадных эвристических алгоритмов с особыми функциями расстояний для задач

автоматической группировки продукции [178,179]. Применены два метода кластерного анализа: метод k-средних [26-31,180], модифицированный метод k-VNS [171, 181].

Метод k-средних в задаче автоматической группировки многомерных данных хорошо зарекомендовал себя [67,71,171,172,180], его применение позволяет достигнуть достаточно высокой точности разбиения на однородные партии.

Метод k-VNS (Variable Neighborhood Search) предусматривает в качестве одного из вариантов организации локального поиска применение алгоритмов поиска с чередующимися окрестностями [44, 50]. Он является метаэвристическим методом для решения набора комбинаторной оптимизации и глобальных задач оптимизации. Этот алгоритм исследует отдаленные районы нынешнего действующего решения и переходит оттуда к новому, если было сделано улучшение [44, 50,181].

Исходные данные представляют собой определенный набор результатов тестовых воздействий на электрорадиоизделия по контролю вольт-амперных характеристик входных и выходных цепей микросхем. В качестве примера рассматриваются микросхемы 1526ТЛ1, микросхемы 1526ИЕ10 и диоды 3ОТ122А. Каждое ЭРИ составлено из данных об изделиях с определенным набором входных характеристик, заведомо принадлежащих разным однородным партиям. Микросхема 1526ТЛ1 состоит из трех однородных партий (количество изделий 625 штук), микросхемы 1526ИЕ10 и диоды 3ОТ122А состоят из пяти и трех однородных партий соответственно (количество изделий микросхемы 1526ИЕ10 составляет 870, диоды 3ОТ122А – 279). Из каждой выборки исключены характеристики с нулевыми значениями. Для анализа в рассмотрении микросхемы 1526ТЛ1 остается 4 входных характеристики, микросхемы 1526ИЕ10 и диоды 3ОТ122А – 6 входных характеристик [178].

Количество кластеров было выбрано в зависимости от количества однородных партий в выборке. Для каждого тестируемого алгоритма проведено по 30 запусков для каждой выборки. Фиксировались все результаты, достигнутые каждым из алгоритмов в каждой попытке, затем среди результатов были определены

минимальные (Min) и максимальные (Max) значения целевой функции по каждой выборке данных, а также среднее значение (Average) и среднеквадратичное отклонение (Std.Dev.) целевой функции (таблица 2.26).

Минимальное значение среднеквадратичного отклонения целевой функции показывает, что представленные алгоритмы методов жадных эвристик показывают высокую точность и стабильность решений. При этом классический алгоритм k-means уступает модифицированному алгоритму k-VNS, который при большинстве случаев запуска возвращал одинаковое рассчитанное значение целевой функции.

Таблица 2.26. Вычислительные результаты работы алгоритмов k-средних и k-VNS

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
Микросхема 1526ТЛ1 (три кластера)				
k-means в режиме мультистарта	581,484	592,637	583,637	5,252
k-VNS	**581,484**	**581,484**	**581,484**	0,000
Микросхема 1526ИЕ10 (пять кластеров)				
k-means в режиме мультистарта	1135,702	1151,079	1143,391	5,233
k-VNS	1101,165	1110,352	1105,759	3,356
Диоды 3ОТ122А (три кластера)				
k-means в режиме мультистарта	946125,23	946148,57	946136,9	3,835
k-VNS	**946116,91**	**946116,91**	**946116,91**	0,000

На следующем шаге проведена оценка влияния информативных признаков на качество автоматической группировки промышленных изделий по однородным партиям. Применены методы k-средних и k-медоид с разным набором информативных признаков для каждой выборки. Для каждого метода произведено 30 тестовых запусков программы с использованием трех мер расстояний: Евклидово расстояние [110,111,182], расстояние Махаланобиса [110, 112, 114], манхэттенское расстояние [111,182,183]. Использование корреляционных зависимостей может быть задействовано путем перехода от поиска в

пространстве с евклидовой или манхэттенской метрикой к поиску в пространстве с расстоянием Махаланобиса [110,112,114].

Микросхема 1526ТЛ1. В случае применения метрики евклидово расстояние точность кластеризации по k- средних составляет 0,824-0,864, по k- медоид – 0,810-0,851. Для метрики манхэттенское расстояние точность по k- средних составляет 0,969-0,984, по k- медоид – 0,971-0,976. Для метрики расстояние Махаланобиса точность по k- средних составляет 0,966-0,984, по k- медоид – 0,972-0,978 (таблица 2.27).

Таблица 2.27 – Результаты кластеризации микросхемы 1526ТЛ1

Алгоритм	Расстояние	Количество информативных признаков		
		2	3	4
		Кластеризовано верно, шт. (доля)		
k- средних	евклидово	540 (0,86)	527 (0,84)	515 (82,4)
	манхэтенское	615 (0,98)	613 (0,98)	606 (0,97)
	Махаланобиса	615 (0,98)	612 (0,98)	604 (0,97)
k- медоид	евклидово	532 (0,85)	513 (0,82)	506 (0,81)
	манхэтенское	610 (0,98)	608 (0,97)	607 (0,97)
	Махаланобиса	611 (0,98)	610 (0,98)	608 (0,97)

Микросхемы 1526ИЕ10. В случае применения евклидовой метрики точность кластеризации по k-means составляет 0,669-0,756, по k- медоид – 0,662-0,740. Для манхэттенской метрики точность по k-means составляет 0,812-0,849, по k- медоид – 0,805-0,839. Для расстояния Махаланобиса точность по k-means составляет 0,862-0,881, по k- медоид – 0,856-0,875 (таблица 2.28).

Диоды 3ОТ122А. В случае применения метрики евклидова расстояния точность кластеризации по k-means составляет 0,720-0,756, по k-медоид – 0,706-0,746. Для метрики манхэттенское расстояние точность по k-средних составляет 0,588-0,642, по k-медоид – 0,566-0,631. Для метрики Расстояние Махаланобиса точность по k-средних составляет 0,785-0,817, по k- медоид – 0,774-0,799 (таблица 2.29).

Таблица 2.28 – Результаты кластеризации микросхемы 1526ИЕ10

Алгоритм	Расстояние	Количество информативных признаков		
		2	4	6
		Кластеризовано верно, шт. (доля)		
k- средних	евклидово	658 (0,756)	632 (0,726)	582 (0,669)
	манхэтенское	739 (0,849)	722 (0,829)	706 (0,812)
	Махаланобиса	766 (0,881)	763 (0,877)	750 (0,862)
k- медоид	евклидово	644 (0,740)	629 (0,723)	576 (0,662)
	манхэтенское	730 (0,839)	715 (0,822)	700 (0,805)
	Махаланобиса	761 (0,875)	759 (0,872)	745 (0,856)

Таблица 2.29 – Результаты кластеризации –диоды 3ОТ122А

Алгоритм	Выбор метрики	Количество информативных признаков		
		2	4	6
		Кластеризовано верно, шт. (доля)		
k-means	евклидова	211 (0,756)	204 (0,731)	201 (0,720)
	манхэттенская	179 (0,642)	168 (0,602)	164 (0,588)
	Махаланобиса	228 (0,817)	221 (0,792)	219 (0,785)
k-medoid	евклидова	208 (0,746)	199 (0,713)	197 (0,706)
	манхэттенская	176 (0,631)	165 (0,591)	158 (0,566)
	Махаланобиса	223 (0,799)	218 (0,781)	216 (0,774)

Анализ полученных результатов кластеризации показал, что с увеличением количества информативных признаков точность кластеризации уменьшается. Также было установлено, что независимо от алгоритмов кластеризации, при использовании меры расстояния Махаланобиса точность кластеризации увеличивается, а вот, при использовании метрики евклидова расстояния точность кластеризации уменьшается.

2.5 Выбор оптимизационной модели для задачи автоматической группировки на основе модели k-средних.

В вычислительном плане задача k-средних, в которой в качестве целевой минимизирумой функции выступает сумма квадратов расстояний, удобнее по

сравнению с моделью к-медиан, использующей сумму расстояний, поскольку при использовании суммы квадратов расстояний центральная точка кластера – центроид – совпадает с усредненным значением координат всех объектов кластера. При переходе к сумме квадратов расстояний Махаланобиса данное свойство сохраняется.

Тем не менее, применение расстояния Махаланобиса в задаче автоматической группировки промышленных изделий во многих случаях приводит к снижению точности в сравнении с результатами, достигаемыми с евклидовой метрикой [184,185] (Приложение А, таблица А.1). Фрагмент результатов эксперимента приведен в таблице 2.30. В таблицах 2.30 и 2.31 представлены количество и доля верно кластеризованных объектов в каждой партии. Данные для проведения эксперимента описаны в разделе 2.2.

Таблица 2.30 – Результаты кластеризации с различными мерами расстояний

Партии	Квадрат евклидова расстояния	Квадрат расстояния Махаланобиса	Манхэттенское расстояние	Косинусное расстояние	Корреляционное расстояние
Частичная выборка, состоящая из четырех однородных партий					
Партия 1 (n=71)	70 (0,99)	47 (0,66)	71 (1,00)	70 (0,99)	70 (0,99)
Партия 2 (n=116)	78 (0,67)	83 (0,72)	64 (0,55)	78 (0,67)	84 (0,72)
Партия 5 (n=146)	96 (0,66)	88 (0,60)	105 (0,72)	96 (0,66)	104 (0,71)
Партия 6 (n=113)	44 (0,39)	91 (0,81)	50 (0,44)	44 (0,39)	38 (0,37)
среднее	**0,65**	**0,69**	**0,65**	**0,65**	**0,66**
Сумма расстояний	473,174	26146,35047	401,4	0,0012	0,0011
Полная выборка					
Партия 1 (n=71)	67 (0,94)	70 (0,99)	68 (0,96)	67 (0,94)	71 (1,00)
Партия 2 (n=116)	4 (0,03)	4 (0,03)	4 (0,03)	4 (0,03)	78 (0,67)
Партия 3 (n=1867)	578 (0,31)	223 (0,12)	558 (0,30)	578 (0,31)	0 (0,00)
Партия 4 (n=1250)	403 (0,32)	127 (0,11)	446 (0,36)	406 (0,33)	227 (0,18)
Партия 5 (n=146)	66 (0,45)	81 (0,55)	63 (0,43)	64 (0,44)	78 (0,53)
Партия 6 (n=113)	88 (0,78)	113 (1,00)	82 (0,73)	88 (0,78)	32 (0,28)

Продолжение таблицы 2.30

Партии	Квадрат евклидова расстояния	Квадрат расстояния Махаланобиса	Манхэттенское расстояние	Косинусное расстояние	Корреляционное расстояние
Партия 7 (n=424)	311 (0,73)	404 (0,95)	303 (0,72)	311 (0,73)	314 (0,74)
среднее	**0,38**	**0,26**	**0,38**	**0,38**	**0,20**
Сумма расстояний	5008,127	248808,6	1755,8	0,007	0,004

В данном эксперименте при использовании расстояния Махаланобиса ковариационная матрица *C* рассчитывалась по всей выборке, используемой в качестве обучающей (параметризующей) выборки.

Также эксперимент показал, что точность кластеризации k-средних с метрикой Махаланобиса с обучением выше в сравнении с метрикой Махаланобиса без обучения. Ковариационную матрицу *C* обучали на выборке из 6 партий. Используя полученную ковариационную матрицу, кластеризовали выборки, составленные из 2 партий во всех возможных комбинациях (с обучением в таблице 2.31). Затем полученный результат сравнили с традиционным способом кластеризации k-means с метрикой Махаланобиса на ковариационной матрице по 2 партиям (без обучения в таблице 2.31), а также с Евклидовым и манхеттенским расстояниями [186]. Для каждой модели проводились множественные попытки запуска экспериментов, усредненные результаты которых представлены в приложении А (таблица А.2). Фрагмент результатов эксперимента приведен в таблице 2.31.

Таблица 2.31 – Результаты кластеризации с обучающей выборкой

Партии	Квадрат расстояния Махаланобиса (с обучением)	Квадрат расстояния Махаланобиса (без обучения)	Квадрат евклидова расстояния	Манхэттенское расстояние
Партия 5+Партия 7 (n=570)				
Партия 7 (n=424)	424 (1,00)	218 (0,51)	380 (0,90)	385 (0,91)
Партия 5 (n=146)	136 (0,93)	85 (0,58)	146 (1,00)	146 (1,00)
среднее	**0,98**	**0,53**	**0,92**	**0,93**
Средняя сумма расстояний	34385	34282	1250	3202

Продолжение таблицы 2.31

Партии	Квадрат расстояния Махаланобиса (с обучением)	Квадрат расстояния Махаланобиса (без обучения)	Квадрат евклидова расстояния	Манхэттенское расстояние
Партия 1+ Партия 5 (n=217)				
Партия 1 (n=71)	71 (1,00)	39 (0,56)	70 (0,99)	71 (1,00)
Партия 5 (n=146)	84 (0,58)	80 (0,55)	99 (0,68)	108 (0,74)
среднее	**0,71**	**0,55**	**0,78**	**0,83**
Средняя сумма расстояний	22199	13004	223	841
Партия 5+ Партия 6 (n=259)				
Партия 5 (n=146)	96 (0,66)	81 (0,55)	146 (1,00)	146 (1,00)
Партия 6 (n=113)	113 (1,00)	66 (0,59)	105 (0,93)	109 (0,75)
среднее	**0,81**	**0,57**	**0,97**	**0,99**
Средняя сумма расстояний	23172	6564	512	1246

Точность кластеризации остается ниже по сравнению с результатами, достигаемыми с Евклидовым и манхэттенским расстояниями.

Предположение о том, что статистические зависимости между значениями характеристик проявляются в разных партиях ЭРИ схожим образом, имеет под собой экспериментальные основания (рисунки 2.25 – 2.30).

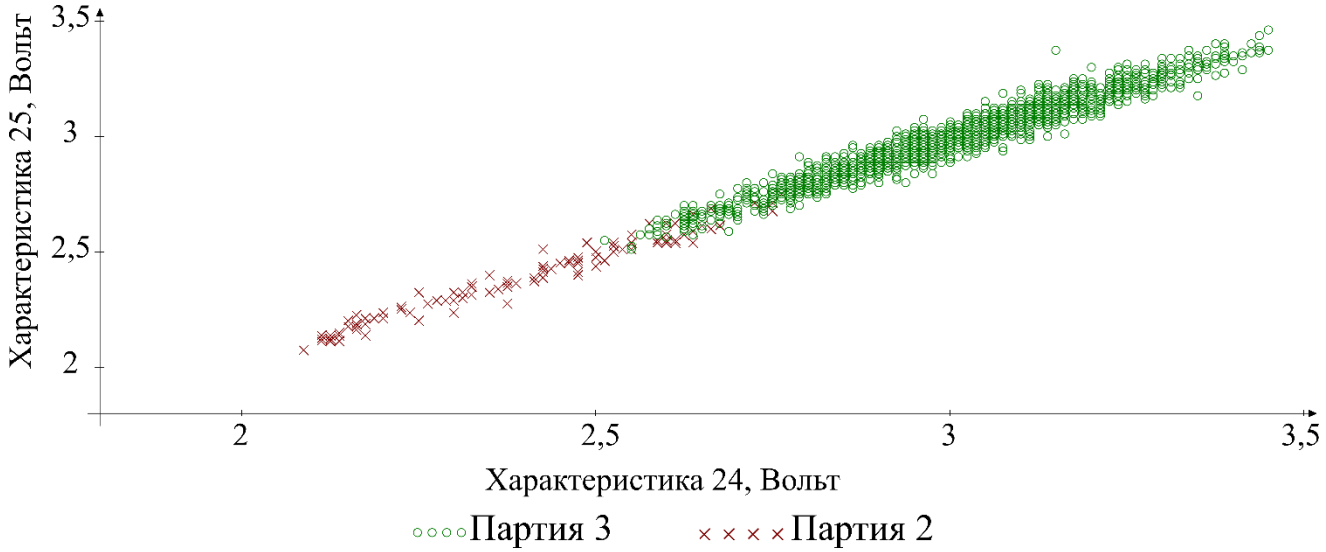


Рисунок 2.25 – Статистическая зависимость характеристик ЭРИ (Партия 2 и Партия 3)

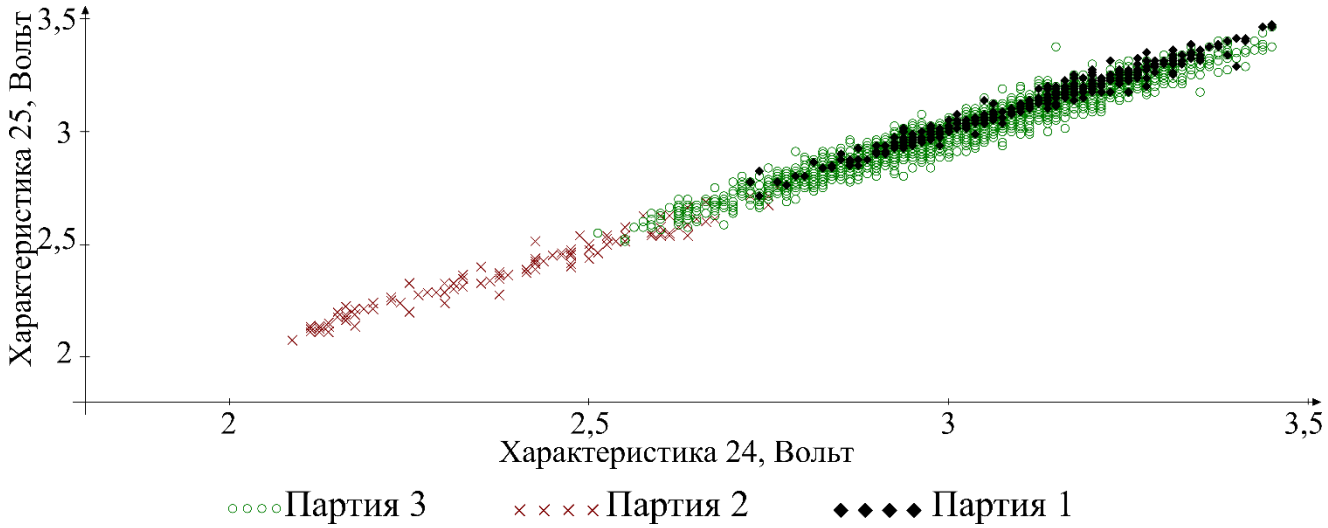


Рисунок 2.26 – Статистическая зависимость характеристик 24,25 ЭРИ (Партия1, Партия 2 и Партия 3)

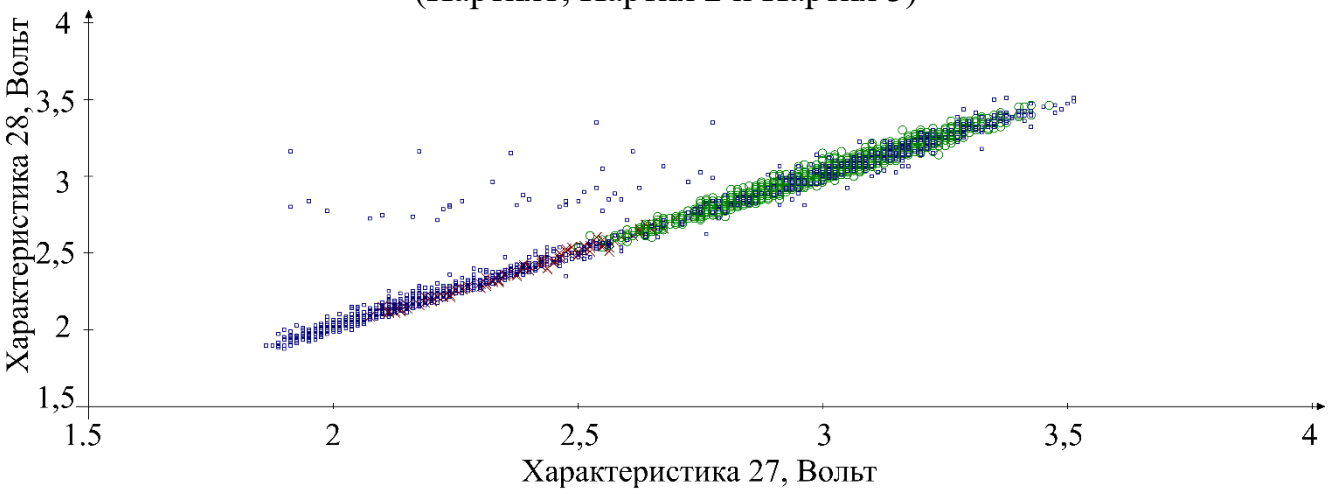


Рисунок 2.27 – Статистическая зависимость характеристик 27,28 ЭРИ (Партия2, Партия 3 и Партия 4)

Рисунок 2.28 – Статистическая зависимость характеристик 97,98 ЭРИ (Партия1, Партия 2 и Партия 3)

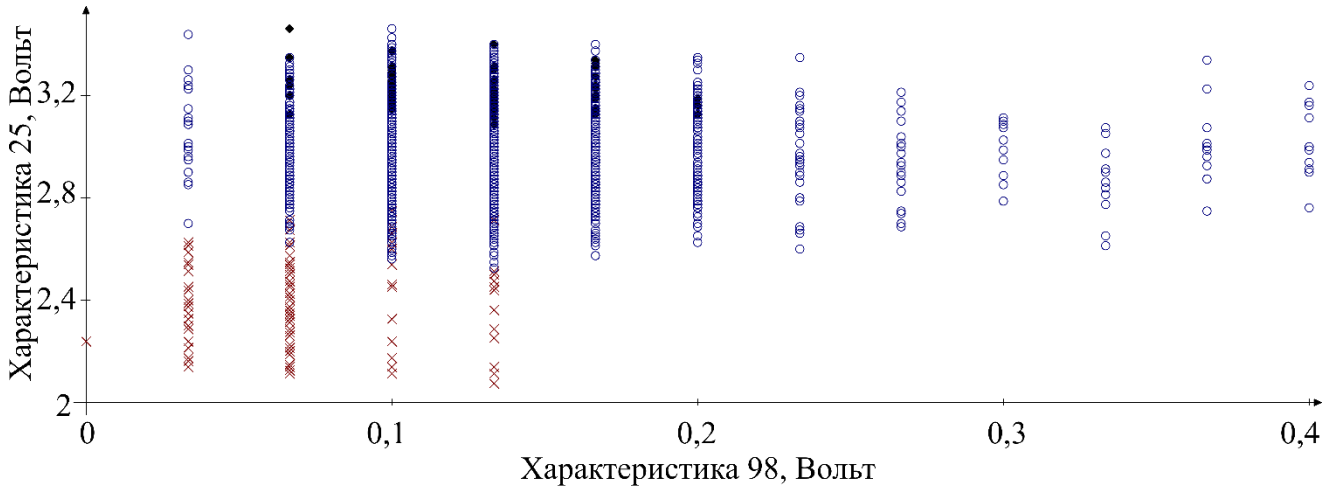


Рисунок 2.29 – Статистическая зависимость характеристик 25,98 ЭРИ (Партия1, Партия 2 и Партия 3)

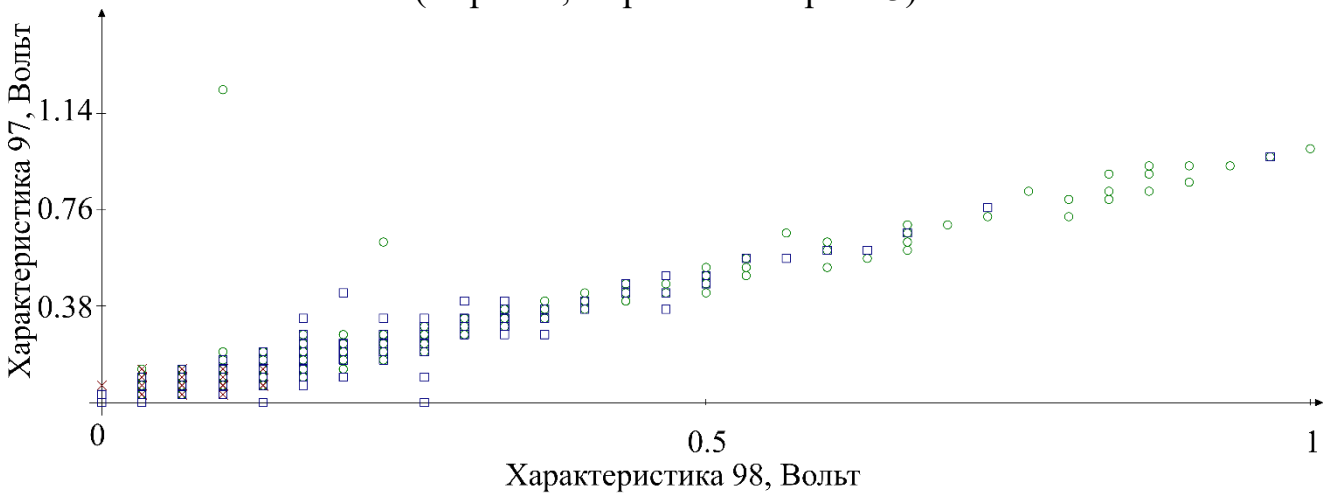


Рисунок 2.30 – Статистическая зависимость характеристик 97,98 ЭРИ (Партия2, Партия 3 и Партия 4)

Тем не менее, как видно из рисунка 2.25, размах и дисперсия характеристик разных партий существенно различаются. Как показывает рисунок 2.30, даже если величина размаха и дисперсии каких-либо характеристик существенно не различаются в партиях отдельных изделий, они существенно отличаются от размаха и дисперсии по выборке в целом.

Таким образом, ошибочно принимать дисперсию и коэффициенты ковариации в каждой из однородных партий равными значениям дисперсии и коэффициентов ковариации по выборке в целом.

При этом эксперименты с моделью автоматической группировки, основанной на модели разделения смеси гауссовых распределений путем

максимизации функции правдоподобия EM-алгоритмом [18], показывают относительно высокую адекватность модели только при использовании диагональных ковариационных матриц (т.е. некоррелированных распределений), причем одинаковых для всех распределений. При этом очевидно имеющиеся корреляции между характеристиками никак не учитываются.

Расстояние Махаланобиса инвариантно относительно масштаба [187]. Благодаря этому свойству нормализация данных не имеет значения в случае применения этой метрики. При этом специальный способ нормирования по границам дрейфов показал высокую эффективность. Если нормализация по размаху (0-1 нормализация) или по среднеквадратичному отклонению уравнивает значимость всех характеристик, неизбежно увеличивая значимость неинформативных характеристик, содержащих лишь шум, то привязка границ характеристик к границам, определяемым их физической природой, задает масштаб, пропорциональный допустимым колебаниям этих характеристик в условиях эксплуатации (границам дрейфа), без привязки к размаху и дисперсии этих значений в конкретной партии. При переходе к метрике Махаланобиса эти преимущества теряются. Решением проблемы сохранения масштаба могло бы стать использование метрики Махаланобиса с корреляционной матрицей R вместо ковариационной матрицы C:

$$D_M(X) = \sum_{i=1}^{n}(X-\mu)^T R^{-1}(X-\mu). \quad (2.23)$$

Каждый элемент матрицы R вычисляется следующим образом:

$$r_{XY} = \sum_{i=1}^{N} \frac{(X_i-\bar{X})(Y_i-\bar{Y})}{(N-1)S_X S_Y}, \quad (2.24)$$

где S_X и S_Y – стандартные отклонения.

Как показывают эксперименты, результаты которых приведены ниже, такой подход не показывает преимущества по сравнению с другими методами.

В некоторых случаях при решении задачи автоматической группировки мы располагаем выборкой, состав которой заведомо известен. Данная выборка может служить обучающей (параметризующей) выборкой, по которой можно рассчитать усредненную ковариационную матрицу для однородной партии [186]:

$$C = \frac{1}{N}\sum_{j=1}^{k} C_j\, n_j, \tag{2.25}$$

где n_j – количество объектов (изделий) в j-й партии, N-общий размер выборки, C_j – ковариационные матрицы отдельных партий изделий, каждая из которых может быть рассчитана по формуле:

$$\mathrm{C}_j = \mu\big[\big(X - \mu(X)\big)\big(Y - \mu(Y)\big)\big], \tag{2.26}$$

где μ – математическое ожидание.

В случае, если такой выборки нет, для выделения предположительно однородных партий могут быть использованы известные модели кластерного анализа. При таком подходе предположительно неоднородная партия может быть разбита на число предположительно однородных партий, определяемое критерием силуэта [61,127,128]. При этом сборная партия может быть разбита на большее число однородных партий, чем есть на самом деле: меньшие кластеры с большей вероятностью содержат данные одного класса, т.е. снижается вероятность ложного отнесения объектов разных классов в один кластер. Доля объектов одного класса, ложно отнесенных к разным классам, при этом не столь важна для оценки статистических характеристик однородных групп объектов.

Предложен алгоритм k-средних с использованием меры расстояния Махаланобиса с усредненной оценкой ковариационной матрицы [186]. Сходимость алгоритма k-средних с использованием расстояния Махаланобиса рассмотрена в [188]. Оптимальное значение k было найдено по критерию силуэта [61,127,128]:

Алгоритм 2.1 k-средних с мерой расстояния Махаланобиса с усредненной оценкой ковариационной матрицы

Шаг 1. Методом k-средних с евклидовыми расстояниями разделить выборку на некоторое число k кластеров (здесь k – некоторая экспертная оценка возможного числа однородных групп, не обязательно точная);

Шаг 2. Для каждого кластера рассчитать центроид μ_i (вектор размерностью M). Центроид определяется как среднее арифметическое всех точек в кластере C_i, i=1,...k:

$$\mu_i = \frac{1}{m_i} \sum_{j \epsilon C_i} X_j,$$

где m_i – количество точек в i-м кластере, X_j – вектор значений измеряемой характеристики размерностью M,

Шаг 3. Рассчитать усредненную оценку ковариационной матрицы (2.25). Если усредненная оценка ковариационной матрицы является вырожденной, то перейти к Шагу 4, в противном случае перейти к Шагу 5.

Шаг 4. Увеличить число кластеров (k + 1) и повторить Шаги 1 and 2. Сформировть новые кластеры, используя квадрат Евклидова расстояния:

$$D(X_j, \mu_i) = \sum_{i=1}^{M} (X_j - \mu_i)^2$$

где M - количество характеристик. Вернуться к Шагу 3 с новым обучающих примером (набором).

Шаг 5. Сопоставить каждый вектор данных ближайшему центроиду, используя квадрат расстояния Махаланобиса с усредненной оценкой ковариационной матрицы (2.25) для формирования новых кластеров.

Шаг 6. Повторить алгоритм с Шага 2 до тех пор, пока кластеры не перестанут изменяться.

В качестве меры точности качества кластеризации использован индекс Рэнда (Rand Index, RI) [129], определяющий долю объектов, для которых эталонное и полученное разбиение по кластерам схоже.

Серия экспериментов проведена над данными микросхемы 1526IE10_002, описание которых представлено в разделе 2.2.

Данная сборная партия удобна тем, что ее состав заранее известен, что позволяет оценивать точность применяемых моделей кластеризации. При этом партия сложна для группировки известными моделями: некоторые однородные партии в ее составе практически неотличимы друг от друга, и точность известных моделей кластеризации на этой партии невысока [189,190].

Для обучения модели с усредненной метрикой Махаланобиса из компонент сборной партии составлены новые комбинации выборок, содержащие изделия, относящиеся к разным однородным партиям. Новые комбинации составлены из 2-7 однородных партий.

В данной работе представлены результаты трех групп экспериментов.

Первая группа. Обучающая выборка соответствует рабочей выборке, для которой проводилась кластеризация.

Вторая группа. Обучающая и рабочая выборки не совпадают. На практике в качестве обучающей выборки тестовый центр может использовать ретроспективные данные поставок и тестирования того же типономинала.

Третья группа. Обучающая и рабочая выборки также не совпадают, но в качестве обучающей выборки использовались результаты работы автоматической группировки электрорадиоизделий (k-средних в режиме мультистарта с евклидовой метрикой).

В каждой из групп экспериментов для каждой рабочей выборки алгоритм к-средних запущен по 30 раз с каждой из пяти исследуемых моделей кластеризации:

Модель DM1. К-средних с метрикой Махаланобиса, ковариационная матрица рассчитана для всей обучающей выборки. Целевая функция определяется как сумма квадратов расстояний;

Модель DC. К-средних с метрикой, аналогичной расстоянию Махаланобиса, но используется корреляционная матрица вместо ковариационной матрицы. Целевая функция определяется как сумма квадратов расстояний;

Модель MD2. К-средних с метрикой Махаланобиса, с усредненной ковариационной матрицей (2.24). Целевая функция определяется как сумма квадратов расстояний;

Модель DR. К-средних с манхэттенским расстоянием Целевая функция определяется как сумма расстояний;

Модель DE. К-средних с евклидовым расстоянием. Целевая функция определяется как сумма квадратов расстояний.

Для каждой модели рассчитаны минимальное (Min), максимальное (Max), среднее значения (Average), среднеквадратичное отклонение (Std.Dev) индекса Рэнда (RI) и целевой функции, а также значения коэффициентов вариации (V) и размаха(R) целевой функции.

Первая группа.

Проведено пять серий экспериментов. *Серия I.* Новая выборка составлена из комбинации ЭРИ, относящихся к Партиям 1-2 (Приложение А, таблица А.3). *Серия II.* Выборка составлена из комбинации ЭРИ, относящихся к Партиям 1-3 (Приложение А, таблица А.4). *Серия III.* Выборка составлена из комбинации ЭРИ, относящихся к Партиям 1-5 (Приложение А, таблица А.5). *Серия IV.* Выборка составлена из ЭРИ, относящихся к Партиям 3-7 (Приложение А, таблица А.6). *Серия V.* Выборка составлена из ЭРИ, относящихся к семи партиям (таблица 2.32).

Таблица 2.32 – Результаты эксперимента Серия V. Первая группа

Серия V	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7549	0,6564	0,8221	0,73896	0,74475	255921	38431	264541	18902	6007,84
Min	0,5601	0,6443	0,7323	0,70230	0,70392	250558	37063	260024	17785	5009,97
Average	0,6270	0,6504	0,7711	0,71616	0,72142	253041	372289	261582	18225	5298,31
Std dev	0,0509	0,0034	0,0244	0,01044	0,00903	1178,1	261,01	989,392	433,12	290,276
V						0,4656	0,7011	0,37824	2,3766	5,47865
R						5363,2	1368,3	4516,87	1116,9	997,872

Вторая группа.

В каждой серии экспериментов в обучающей выборке представлено не более семи однородных партий, в свою очередь рабочая выборка представлена новой комбинацией ЭРИ, относящихся к разным однородным партиям. *Серия I.* Рабочая выборка составлена из изделий Партий 1 и 2, обучающая выборка состоит из Партий 3-7 (Приложение А, таблица А.7). *Серия II.* Рабочая выборка

составлена из ЭРИ Партий 1 и 2, обучающая выборка состоит из Партий 1-7 (Приложение А, таблица А.8). *Серия III.* Рабочая выборка составлена из ЭРИ Партий 1-3, обучающая выборка состоит из Партий 4-7 (Приложение А, таблица А.9). *Серия IV.* Рабочая выборка составлена из ЭРИ Партий 1-3, обучающая выборка состоит из Партий 1-7 (Приложение А, таблица А.10). *Серия V.*Рабочая выборка составлена из ЭРИ Партий 1-5, обучающая выборка состоит из Партий 1-7 (таблица 2.33). *Серия VI.* Рабочая выборка составлена из ЭРИ Партий 3-7, обучающая выборка состоит из Партий 1-7 (Приложение А, таблица А.11).

Третья группа

В каждой серии экспериментов обучающая выборка состоит из 10 партий, являющиеся в свою очередь результатом применения алгоритма k-средних к обучающей выборке, содержащей всю сборную партию целиком, рабочая выборка представлена новой комбинацией ЭРИ, относящихся к разным однородным партиям.Серия I.Рабочая выборка составлена из ЭРИ Партий 1-2 (Приложение А, таблица А.12). *Серия II.*Рабочая выборка составлена из ЭРИ Партий 1-3 (Приложение А, таблица А.13). *Серия III.*Рабочая выборка составлена из ЭРИ Партий 1-5 (Приложение А, таблица А.14). *Серия VI.*Рабочая выборка составлена из ЭРИ Партий 3-7 (Приложение А, таблица А.15). *Серия V.*Рабочая выборка составлена из ЭРИ Партий 1-7 (таблица 2.34)

Таблица 2.33 – Результаты эксперимента Серия V. Вторая группа

Серия V	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7490	0,6454	0,8524	0,73366	0,73567	254822	38704	263405	20509	9194,61
Min	0,4312	0,6314	0,7470	0,69559	0,68932	249355	37856	257534	19408	6554,1
Average	0,5660	0,6362	0,8117	0,70799	0,71919	251694	37982	259689	19674	7119,85
Std dev	0,0519	0,0034	0,0324	0,01538	0,01002	1462,8	203,55	1502,09	289,63	571,119
V						0,5812	0,5359	0,57842	1,4722	8,02151
R						5466,8	847,84	5870,96	1101,6	2640,51

Таблица 2.34 – Результаты эксперимента Серия V. Группа 3

Серия V	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7672	0,6579	0,7489	0,73969	0,73456	255886	379167	281265	18897	6494,62
Min	0,5618	0,6453	0,6958	0,70286	0,70466	250839	36997	274506	17785	5009,42
Average	0,6317	0,6499	0,7246	0,71359	0,71935	252877	37178	277892	18240	5249,95
Std dev	0,0468	0,0032	0,0160	0,00814	0,00630	1164,5	152,84	2358,92	452,73	366,499
V						0,4605	0,4111	0,84886	2,4820	6,98101
R						5047,1	919,81	6758,85	1112,2	1485,20

Согласно представленным результатам на рисунках 2.31-2.33 в большинстве случаев коэффициент вариации V значений целевой функции выше у модели DE, где используется евклидова метрика. Также показано, что коэффициент размаха значений целевой функции выше у модели DM2, где используется метрика Махаланобиса, с усредненной ковариационной матрицей. Следовательно, получение устойчивых значений целевой функции требует многократных попыток запуска алгоритма к-средних или использование других алгоритмов, основанных на модек-средних, таких как j-means [191] или алгоритмов метода жадных эвристик [109].

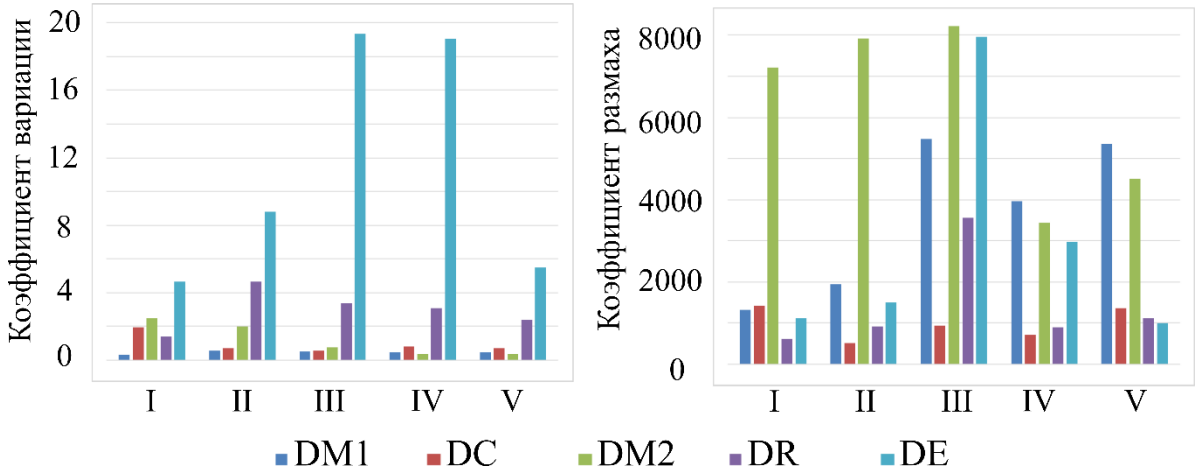


Рисунок 2.31 – Первая группа

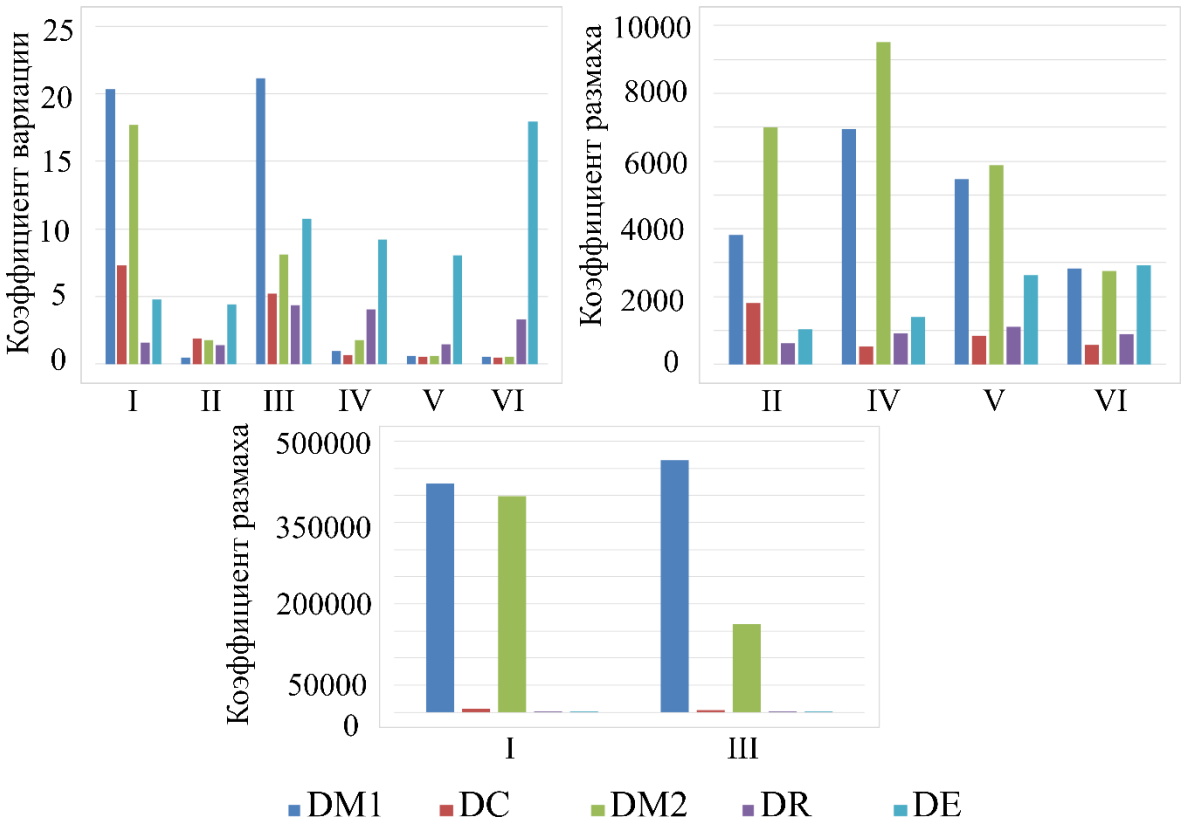


Рисунок 2.32 – Вторая группа

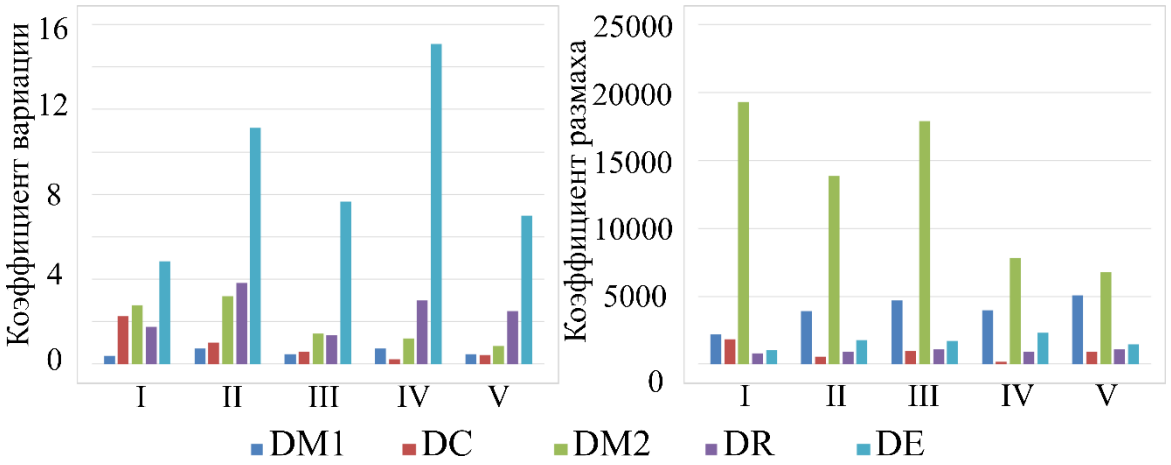


Рисунок 2.33 – Третья группа

Также выявлено, что модель DM2 в первой группе экспериментов показывает наибольшую точность кластеризации по индексу Рэнда RI относительно остальных представленных моделей в данной работе (рисунок 2.34 а)). Во второй группе экспериментов (2, 4, 5 из 6 серий) и третьей группе экспериментов (3, 4 из 5 серий) показано, что точность кластеризации у модели DM2 выше остальных моделей (рисунки 2.34 б), 2.34 в)). Также во всех сериях экспериментов модель DM2 превосходит Модель DE, где используется Евклидова метрика.

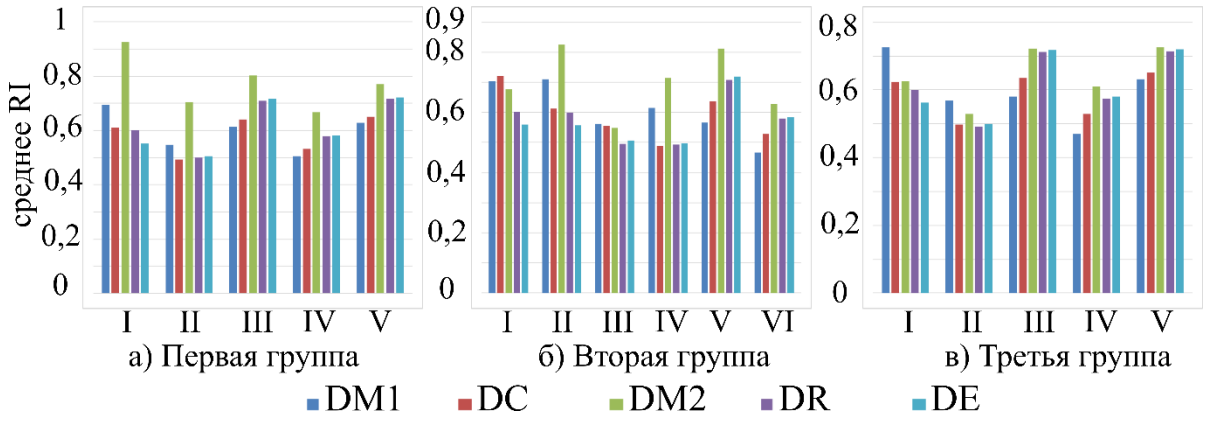


Рисунок 2.34 – Среднее RI

С увеличением количества ЭРИ в обучающей и рабочей выборках точность кластеризации остается постоянной, что подтверждается, например, рисунками 2.35б) и из Приложения Б (рисунки Б.2 (серия VI), Б.5 (серия V, VI), Б.6 (серия IV, V)). Также было выявлено, что в серии экспериментов, в которых была использована модель DM1, отсутствует корреляции между значениями целевой функции и значениями индекса RI (рисунок 2.35а), Приложение Б, рисунки Б.1-Б.3). При этом для результатов с расстоянием DM2 прослеживается обратная корреляция между достигаемым значением целевой функции и точностью кластеризации RI при небольшой выборке (рисунок 2.36а) (серия I), приложение Б, рисунок Б.7 (серия II)), но с увеличением количества элементов в выборке ни прямая, ни обратная корреляция не прослеживается (Приложение Б, рисунки Б.7 (серия IV); Приложение Б, рисунки Б.8 (серия IV); Приложение Б, рисунки Б.9 (серия V)).

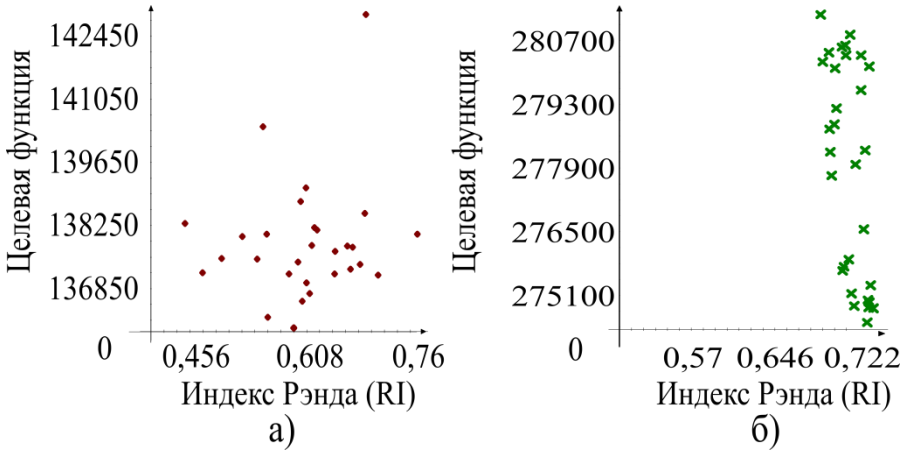


Рисунок 2.35 – Зависимость индекса Рэнда RI от значения целевой функции

а) Модель DM1 (Вторая группа, серия IV); б) Модель DM2 (Тертья группа, серия III)

Кроме того, внимания заслуживает тот факт, что при применении евклидовой метрики лучшие (меньшие) значения целевой функции не соответствуют лучшим (большим) значениям точности кластеризации (RI) (рисунок 2.36б), Приложение Б, рисунки Б.13-Б.15). Данный факт показывает, что модель с евклидовой метрикой не вполне адекватна – наиболее компактные кластеры не вполне соответствуют однородным партиям ЭРИ.

При этом для результатов с расстоянием DM2, где используется мера расстояния Махаланобиса с усредненной оценкой ковариационной матрицы, прослеживается обратная корреляция между достигаемым значением целевой функции и точностью кластеризации RI, либо же ни прямая, ни обратная корреляция не прослеживается (Приложение Б, рисунки Б.7 – Б.9).

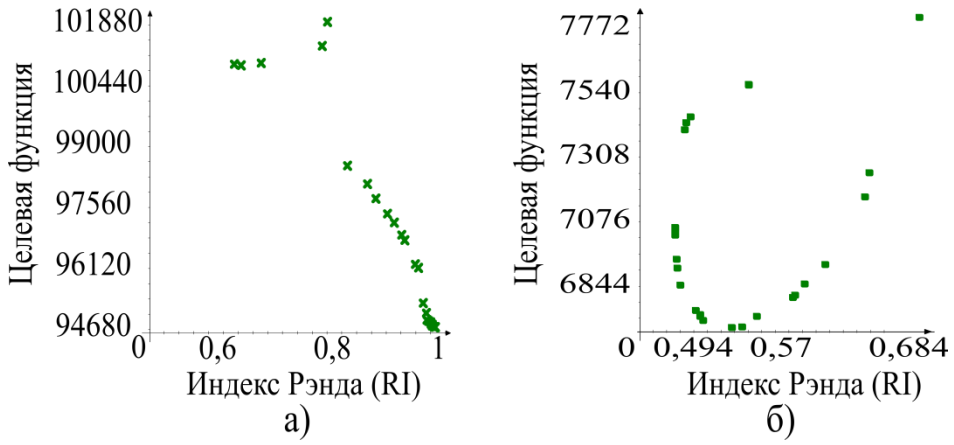


Рисунок 2.36 - Зависимость индекса Рэнда RI от значения целевой функции
a) DM2 model (Первая группа, Серия I); b) DE model (Первая группа, Серия I)

Таким образом, учитывая более высокое среднее значение RI, модель автоматической группировки ЭРИ по электронным партиям на основе модели k-средних с усредненной (средневзвешенной) ковариационной матрицей имеет преимущество перед моделями с евклидовой и манхэттенской метриками по точности.

Результаты Главы 2

Исследовано практическое применение факторного анализа для решения задачи автоматической группировки электрорадиоизделий по производственным партиям. Выполнено сравнение различных методов выделения факторов, включая рассмотрение двух вариантов ортогонального вращения.

Методы факторного анализа не позволяют существенно сократить размерность пространства без потери точности. Тем не менее, в некоторых случаях точность разбиения (доля объектов, корректно отнесенных к «своему» кластеру, представляющему однородную партию промышленной продукции) на однородные партии может быть существенно повышена, особенно для выборок, состоящих из значительного количества (более 2-3) однородных партий.

Исследована возможность применения жадных эвристических алгоритмов для задач разделения смеси распределений с особыми мерами расстояний. Рассмотрены варианты использования особых мер расстояний в задачах автоматической группировки промышленной продукции при небольшом объеме входных данных. Произведено сравнение разных мер расстояний. Было установлено, что расстояние Махаланобиса повышает точность разделения изделий независимо от алгоритмов кластеризации (k-means, k-medoid).

Предложенная модель кластеризации, использующая модель k-средних с расстоянием Махаланобиса и усредненную (средневзвешенную) оценку ковариационной матрицы, сравнивались с моделью k-средних с евклидовыми и манхэттенскими расстояниями при решении задачи автоматической группировки промышленной продукции по однородным производственным партиям. Принимая во внимание более высокое среднее значение индекса Рэнда, предложенная модель и алгоритм оптимизации, применяемые для кластеризации пормышленной продукции на однородные производственные партии, имеет преимущество перед моделями с традиционно используемыми евклидовыми и манхэттенскими расстояниями.

Представленная модель и алгоритм для решения задачи автоматической группировки продукции на основе модели k – средних с мерой расстояния Махаланобиса и средневзвешенной ковариационной матрицей используется в деятельности АО «Испытательный технический центр – НПО ПМ» (Приложение В).

Показано, что коэффициент размаха значений целевой функции выше у новой *модели DM2*, где используется мера расстояния Махаланобиса с усредненной оценкой ковариационной матрицы. Следовательно, для получения устойчивых значений целевой функции требуются множественные попытки запуска алгоритма k-средних или использование других алгоритмов, основанных на модели k-средних, таких как j-means, или же алгоритмов метода жадных эвристик. Разработке усовершенственного алгоритма посвящена Глава 3 работы.

ГЛАВА 3. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧИ K-СРЕДНИХ

Глава посвящена разработке генетического алгоритма с перекрестной мутацией для задачи k-средних. Новый алгоритм позволяет повысить точность решения задачи k-средних (1.1) и стабильность результата за фиксированное ограниченное время выполнения.

В данной главе под точностью алгоритма понимается исключительно достигнутая величина целевой функции. Не учитываются другие важные вопросы в области кластерного анализа, такие как адекватность модели (1.1) и соответствие результатов алгоритма фактическому (реальному) разделению объектов, если таковое известно.

3.1 Известные генетические алгоритмы для задачи k – средних

Классические алгоритмы кластеризации, например, k-средних, реализуют поиск решения в некотором небольшом подмножестве допустимых решений при определённых ограничениях, например, ограничение на количество групп. При этом не гарантируется нахождение строго оптимального решения. То есть данные алгоритмы осуществляют локальный поиск, улучшая предыдущий результат. Результат работы алгоритма зависит от первоначального выбранного решения. Следовательно, для поиска оптимального решения требуются множественные попытки запуска алгоритмов с процедурами случайного выбора начальных решений или случайного выбора локального поиска [192,193], либо использование данных процедур в алгоритме одновременно, а также можно применить более сложные алгоритмы, такие как генетические алгоритмы [21,95], нейронные сети [95], алгоритм имитации отжига [93]. Данные алгоритмы основаны на идее моделирования природных процессов [21]. Авторами работы «Генетический алгоритм k-средних» [86] показано преимущество генетических алгоритмов перед классическими алгоритмами.

Генетический алгоритм (ГА) является представителем эволюционных алгоритмов оптимизации (поиска оптимального решения). Алгоритм основан на

принципах генетических процессов эволюции живой природы, в связи с этим в эволюционных алгоритмах используются понятия и принципы эволюции. Приведем понятия «популяция» и особых используемых для нее процедур, называемых «эволюционные операторы». Популяция представляет собой набор различных возможных вариантов решения, называемых также хромосомами, которые в свою очередь состоят из набора характеристик, называемых генами, а их значения аллелями. Для задачи кластеризации под популяцией понимается возможные варианты группировки. Особые процедуры позволяют получить новые решения, называемые потомками. Особыми процедурами являются процедуры селекции, процедуры скрещивания, процедуры мутации. Процедура селекции предназначена для отбора наиболее сильных родительских хромосом и передачи их генов следующим поколениям потомков. Процедура скрещивания служит для обмена генами родителей для получения возможных новых последовательностей. Например, за одну перестановку для каждой пары родительских хромосом образуется новая пара хромосом-потомков. Процедура мутации изменяет последовательность генов внутри хромосомы.

В своей работе Дж. Холланд вводит понятие функции приспособленности [95], которая предназначена для определения лучшего решения поставленной задачи. Под функцией приспособленности понимается целевая функция. Целью скрещивания генов является достижение наилучшего значения целевой функции.

В работе В.Б. Берикова и Г.С. Лбова «Современные тенденции в кластерном анализе» [21] представлены следующие шаги генетического алгоритма для задачи автоматической группировки.

Шаг 1. Случайным образом формируется набор возможных вариантов решения (популяция). Каждый вариант решения представляет собой последовательность целых чисел определенной длины. Каждое число соответствует номеру кластера (группы). Для каждого решения определяем значение целевой функции.

Шаг 2. Используя эволюционные операторы, формируется новый набор вариантов решения, то есть новая популяция. Для отбора лучших решений

(родительских решений) по значению целевой функции используется процедура селекции, а применив процедуру скрещивания для выбранных лучших решений, формируем новую последовательность целых чисел (решения-потомки). Например, с помощью перестановки одного (или нескольких) сегментов у родительских решений. С помощью процедуры мутации изменяется последовательность целых чисел по заданному правилу, например, заменяется один номер кластера другим. Как правило, изменение происходит случайным образом. Затем вычисляются значения целевой функции для новой популяции.

Шаг 2 повторяется до тех пор, пока не выполнится условие остановки.

Существуют базовые способы кодирования решений: битовое, целочисленное и вещественное. В классическом битовом представлении решение ГА, предложенное Дж. Холландом [95], состоит из битовой строки. Каждый ген соответствует одной характеристике. В отличие от битового целочисленное кодирование может представить значения генов целыми числами в заданном интервале. Для решения задач в непрерывных пространствах используется вещественное кодирование, где каждый ген представлен вещественным числом.

Также используются иные способы кодирования решений, например, в виде множества индексов группируемых объектов, выбранных в качестве центров групп в k-медоидной задаче [194], а также выбранных в качестве начальных центров в p-медианной задаче на сети, которые в дальнейшем изменяются с помощью процедур локального поиска [87].

Анализ литературы показал, что практически отсутствует систематизация используемых подходов [195–198] для алгоритмов с кодированием решений вещественными числами.

Существуют множество вариантов процедур селекции, например, такие как рулеточная селекция, турнирный отбор [199]. В рулеточной селекции определяется вероятность выбора решения, как отношение значения целевой функции данного решения к сумме значений всех целевых функций возможных решений популяции. В турнирном отборе из популяции на первом этапе

выбирается лучшее из пары решение, затем на следующем этапе происходит отбор лучших среди лучших для процедуры скрещивания.

В процедурах скрещивания для целочисленного кодирования применяют, например, 1-точечный, 2-точечный и однородные операторы скрещивания (кроссовера). Для 1-точечного кроссовера выбирается одна точка разрыва и происходит обмен частями родительской хромосомы после точки, для 2-точечного кроссовера выбирается две точки разрыва, и обмен частями родительских хромосом происходит между точками. В однородных операторах кроссовера для целочисленного кодирования выбранный случайно ген родительского решения наследуется гену потомка, по следующему правилу. Каждому выбранному гену определяется случайное число в пределах от 0 до 1, если число больше 0,6, то ген унаследуется первому потомку от первого родительского решения, иначе второму потомку. А для бинарного кодирования точкой разрыва является разряд хромосомы. Для вещественного кодирования в литературе представлены различные кроссоверы [200], например, такие как 2-точечный и дискретный кроссовер. 2- точечный кроссовер аналогичен целочисленному, с разницей, что точка разрыва ставится между генами. В дискретном кроссовере новое решение получается из случайных взятых генов родительских решений.

В процедурах мутации для целочисленного кодирования используют, например, точечную мутацию и генную мутацию для бинарного кодирования, инверсию и перестановку для целочисленного кодирования. В точечной мутации происходит замена одного разряда гена. В генной мутации происходит замена полного гена. В инверсии происходит полная или частичная замена значений генов. В перестановках допускается замена значений гена (или генов) фрагментарно. Для вещественного кодирования оператор мутации изменяет значение гена с некоторой вероятностью.

Появление первого ГА для решения дискретной задачи p-медианы, предложенного С. Хосаджем и М. Гудчайлдом [201], предшествовало ГА для задачи k – средних. В своих исследованиях Ю. Чиоу и Л. Лян [202] использовали

эволюционные операторы для поиска узлов сети, наиболее подходящих в качестве элементов решения (центров), при этом остальные узлы могут быть решениями (центрами), которые позволяют улучшить значение целевой функции.

Авторы работы [203] представили ГА с кодированием решений в виде множества индексов узлов сети, выбираемых в качестве центра, используя несколько операторов скрещивания. Алгоритм, дал более точные результаты с очень медленной сходимостью. В своей работе О. Алп, Э. Эркут, Ц. Дрезнер [204] представили простой и более быстрый генетический алгоритм с особой процедурой скрещивания – жадная (агломеративная) эвристическая процедура, которая также дает точные результаты для сетевой задачи p-медианы. Идея данной процедуры заключается в объединении родительских множеств индексов узлов, которые выбраны в качестве центров, в одно недопустимое решение с избыточным числом центров. Затем последовательно удаляется тот центр, удаление которого показывает наименьшее ухудшение значения целевой функции. Данная идея унаследована и представлена в дальнейшем в работах А.Н. Антамошкина и ЛА. Казаковцева для решения задач автоматической группировки на основе моделей теории размещения [105,106]. В дальнейшем алгоритмы с жадной агломеративной процедурой были адаптированы для решения задачи k-средних.

В генетических алгоритмах для k-средних и аналогичных задач, которые используют традиционное двоичное кодирование хромосом, можно использовать многие методы мутации. Например, в работе [205] авторы представляют хромосому двоичными строками, составленными из двоично-закодированных характеристик (координат) центров. Оператор мутации произвольно изменяет один или несколько генов выбранной хромосомы. В некоторых ГА используется двоичное кодирование, даже если поиск центров осуществляется в непрерыном пространстве [87, 105, 206]. В алгоритме k-средних обычно начальное решение является подмножетвом исходных данных. В таком кодоровании решения 1 означает, что соответствующий вектор данных рассматривается в качестве начального центра и должен быть выбран, а 0 ставится для тех данных, которые

не рассматриваются в качестве начального центра. В этом случае, алгоритм k-средних или аналогичный ему алгоритм используется на каждой итерации ГА для оценки конечного значения (локального минимума) целевой функции (1.1). В работе [207] авторы называют свой алгоритм «Эволюционные k-средние». Однако они фактически решают альтернативную задачу k-средних, направленную на повышение устойчивости кластеризации вместо минимизации (1.1). Этот алгоритм работает с бинарными матрицами и использует два типа операторов мутации: кластерное разбиение и кластерное слияние (агломеративное). В [208] хромосомы представляют собой цепочки целых чисел, представляющих номер кластера для каждого из кластерных объектов, и авторы решают задачу k-средних с одновременным определением количества кластеров на основе критерия силуэта [61,127,128] и критериев Дэвида-Боулдина [209] (аналогичный подход используется в гораздо более простом алгоритме X-Means [210]), которые используются в качестве целевой функции. Таким образом, авторы работы [208] решают задачу с математической постановкой, отличной от (1.1), и используют пересчет кластеров в соответствии с (1.1) в качестве оператора мутации. Подобное кодирование используется в [86], где авторы предлагают оператор мутации, который изменяет назначение отдельных объектов данных кластерам. А. В. Еремеев описал оператор мутации как процедуру, которая гарантирует разнообразие популяции [211]. Для задачи k-средних и p-медианы процедура мутации, как правило, изменяет одно или несколько решений, заменяя некоторые центры [85, 86].

Процедуры скрещивания и мутации являются наиболее важными процедурами в генетических алгоритмах. Как описано выше, процедура скрещивания стремится сохранить свойства родительских решений в решениях потомков, в свою очередь, процедура мутации производит небольшие изменения внутри решения. Процедура мутации обычно рассматривается как вторичный оператор с низкой вероятностью ω [95].

Многие исследования показали [212-215], что эволюционные алгоритмы без процедуры скрещивания могут работать лучше, чем стандартный генетический

алгоритм, если процедура мутации сочетается с эффективной процедурой отбора. Процедура мутации выполняется в единственном родительском решении. В [85] авторы кодируют решения в своем генетическом алгоритме как наборы центроидов, представленные их координатами (векторами действительных чисел) в d-мерном пространстве. Генетические алгоритмы с жадным эвристическим оператором скрещивания используют тот же принцип [216].

Таким образом, различные генетические алгоритмы для k-средних и аналогичных задач можно разделить на три группы в соответствии со способом кодирования решения (хромосомы). При целочисленном кодировании каждый ген решения представлен вектором данных $A_1,\ldots,A_N$, значением является номер его кластера. Такие алгоритмы предназначены для решения задач k-средних и p-медианы, при этом могут использоваться другие целевые функции, отличные от (1.1). Иногда поиск локального минимума (1.1) объявляется процедурой мутации. При целочисленном или бинарном кодировании каждый ген соответствует центроиду и описывает индекс вектора данных $\overline{1,N}$, выбранный как начальный центроид для метода локального поиска. Эти алгоритмы могут использовать широкое разнообразие операторов скрещивания и мутации. При веществеом кодировании каждый ген представляет собой центроид, закодированный своими координатами.Такие алгоритмы способны демонстрировать наиболее точные результаты. Как правило, они не используют никаких процедур мутации [87,105,203].

Для различных методов кодирования хромосом были разработаны различные операторы мутации: инверсия битов для двоичного кодирования [95], обмен, вставка, инверсия и перестановка смещения [217] для хромосом переменной длины, мутация по Гауссу [218] и полиномиальная мутация для вещественного кодирования. [219,220]. Некоторые исследования предлагают комбинацию операторов мутации [221] или операторов самоадаптивной мутации [222–224]. Эффективность различных операторов мутации зависит от параметров ГА [214,225,226] и типа задачи [227,228]. Следует отметить, что в литературе для генетических алгоритмов при решении задачи k-средних с вещественным

кодированием известно очень ограниченное множество возможных операторов мутации. ГА для сетевой проблемы p-медианы, описанной в [229], включает оператор гипермутации, который заключается в попытке заменить каждый ген в хромосоме каждым геном из набора генов, которые изначально не были частью обработанной хромосомы. После каждой замены алгоритм проверяет улучшение значения целевой функции. Оператор требует больших вычислительных ресурсов из-за многочисленных проверок целевой функции и фактически аналогичен принципу локального поиска, встроенному в алгоритм j-means [190]. В [230] алгоритм гипермутации получил дальнейшее развитие как алгоритм ближайших четырех соседей. Идея состоит в том, чтобы снизить вычислительные затраты за счет сокращения набора генов, используемых для замены, до ближайших соседей заменяемого гена. В работах [86,231,232] авторы предлагают использовать алгоритм k-средних в качестве оператора мутации. Каждый из этих алгоритмов объявляет локальный поиск как оператор мутации. Структура ГА позволяет нам использовать широкий спектр вариантов генетических операторов. Однако локальный поиск предназначен для улучшения произвольного решения, преобразовывая его в локальный оптимум и тем самым уменьшая, а не увеличивая разнообразие хромосом (решений).

Авторы работы «Метод кластеризации на основе генетического алгоритма» [85] кодируют решения (хромосомы) в своих ГА как наборы центроидов, представленных их координатами (векторами действительных чисел) в многомерном пространстве. Каждая хромосома подвергается мутации с фиксированной вероятностью ω Процедура (оператор) мутации выглядит следующим образом (равномерная случайная мутация).

Алгоритм 3.1 Процедура исходной мутации ГА для задачи k-средних

Шаг 1. Генерация случайного числа $b \in (0,1]$ с равномерным распределением;

Шаг 2. ЕСЛИ $b < \omega$, ТО хромосома мутирует. Если положение текущего центроида υ, то после мутации оно становится:

$$\upsilon \leftarrow \begin{cases} \upsilon \pm 2 \times b \times \upsilon, \upsilon \neq 0, \\ \upsilon \pm 2 \times b, \upsilon = 0. \end{cases}$$

Знаки «+» и «-» имеют одинаковую вероятность. Координаты центроида смещены случайным образом.

Координаты центроида сдвигаются случайным образом. Однако локальные минимумы, распределенные по пространству поиска, не являются однородными [211]: новые локальные минимумы (1.1) могут быть найдены с большей вероятностью в некоторой окрестности известного локального минимума, чем в окрестности случайно выбранной точки. Таким образом, объединение двух локальных минимумов (например, смешение центроидов из двух известных локальных минимумов) должно превосходить случайный сдвиг координат центроидов. Идея объединения локальных минимумов является основной идеей генетических алгоритмов с жадными агломеративными эвристическими процедурами кроссовера [87,104] и другими алгоритмами [233]. Идея жадной агломеративной эвристической процедуры кроссовера для задачи дискретной p-медианы, предложенная в [204] и адаптированная для задач непрерывной p-медианы и k-средних в [87,105], была использована в генетических алгоритмах без какой-либо процедуры мутации. Такие алгоритмы демонстрируют очень точные результаты по сравнению с другими алгоритмами на практических задачах. Одна из наиболее важных проблем генетических алгоритмов - это схождение всей популяции в некую узкую область (вырождение популяции) вокруг некоторого локального минимума. В алгоритмах поиска переменных окрестностей с рандомизированными окрестностями, образованными применением жадных агломерационных процедур к известному решению и случайно сгенерированному решению [73,81], используется аналогичная идея. Алгоритмы, предложенные в [73,81], демонстрируют лучшие результаты, чем аналогичные генетические алгоритмы. Новые случайно сгенерированные решения (локальные минимумы) используются для формирования рандомизированной окрестности вокруг наиболее известного решения. В этом случае применение жадных агломеративных эвристических процедур к наиболее известному решению в сочетании со случайно сгенерированными локальными минимумами обеспечивает разнообразие окрестностей.

3.2 Генетический алгоритм с новой процедурой перекрестной мутации и его вычислительный эксперимент для задачи k- средних

Генетический алгоритм, представленный в работе «Метод кластеризации на основе генетического алгоритма» использует для выбора первоначального решения один из видов селекции – рулеточная селекция, а также 1-й оператор скрещивания.

В своей работе мы заменили эту процедуру мутации для задачи k-средних следующей процедурой [234,235].

Алгоритм 3.2 Процедура перекрестной мутации ГА для задачи k-средних

Шаг 1. Генерация случайного начального решения $S = \{X_1 \dots X_k\}$;

Шаг 2. Применить алгоритм k-средних к S для получения локального оптимума S';

Шаг 3. Применить простую перекрестную процедуру для мутированного индивида S' из популяции и S для получения нового решения S'';

Шаг 4. Применить алгоритм k-средних к S'' для получения локального оптимума S'';

Шаг 5.ЕСЛИ $F(S'')<F(S')$, **ТО** $S' \leftarrow S''$.

Предложенная процедура используется с вероятностью мутации равной 1, после каждого оператора скрещивания.

Проведен вычислительный эксперимент для набора данных о расположении пользователей Mopsi-Joensuu (6014 векторов данных размерности 2) [235,236]. Результаты запуска исходного Алгоритма 3.1, описанного с вероятностью мутации 0,01, и его версия с Алгоритмом 3.2 в качестве оператора мутации представлены в таблице 3.1 и на рисунке 3.1 (численность популяции N_{POP} = 20).

Таблица 3.1 – Результаты вычислительного эксперимента для набора данных Mopsi-Joensuu (6014 векторов данных размерности 2), 300 кластеров, ограничение по времени 3 минуты

Поколение	ГА k-средних с исходной мутацией	ГА k-средних с перекрестной мутацией	k-средних с режиме мультистарта (лучшее значение)
0	1706,07	1706,07	1859,06
10	1697,29	1667,95	
20	1682,37	1664,78	
30	1681,06	1664,78	
40	1679,58	1664,78	
50	1679,58	1664,78	
60	1672,29	1664,78	
70	1667,27	1664,78	
80	1666,17	1664,78	
90	1666,03	1664,78	
100	1665,01	1664,78	
150	1664,81	1664,78	
200	1664,78	1664,78	

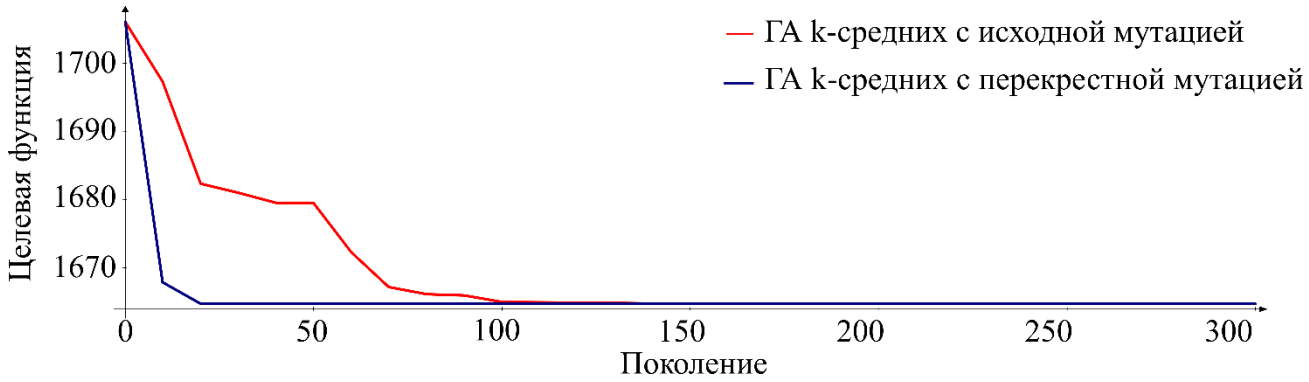


Рисунок 3.1 – Результаты для набора данных Mopsi-Joensuu (6014 векторов данных размерностью 2), 300 кластеров, 3 минуты

Новая процедура мутации является быстрой и эффективной по сравнению с исходной мутацией генетического алгоритма, показана высокая скорость сходимости целевой функции.

3.3 Подход к повышению разнообразия в популяции генетических алгоритмов с жадной агломеративной эвристикой для задачи k-средних

Генетические алгоритмы метода жадных эвристик и многие другие эволюционные алгоритмы для задачи k-средних, обходятся без мутации. Идея жадной агломеративной эвристической процедуры заключается в объединении

двух известных решений в одно промежуточное недопустимое решение с избыточным числом центроидов и далее последовательно уменьшается количество центроидов. На каждой итерации удаляется центроид, который показывает наименьшее увеличение значения целевой функции (1.1).

Алгоритм 3.3 Базовая жадная агломеративная эвристическая процедура

Дано: начальное число кластеров K, необходимое количество кластеров k, $k<K$, начальное решение $S = \{X_1, \dots, X_K\}$, где $|S| = K$.

ШАГ 1. Улучшить начальное решение алгоритмом k-средних

ПОКА $K>k$

ЦИКЛ для каждого $i' \in \{\overline{1, K}\}$ выполнить:

ШАГ 2. $S' \leftarrow S \backslash \{X_{i'}\}$. Улучшить решение S' алгоритмом k-средних и сохранить соответствующие полученные значения целевой функции (1.1), как переменные $F'_{i'} \leftarrow F\{S'\}$.

КОНЕЦ ЦИКЛА

ШАГ 3. Выбрать подмножество S_{elim} из центров n_{elim}, $S_{elim} \in S, S_{elim} \vee n_{elim}$ с минимальным значением соответствующих переменных $F'_{i'}$. $n_{elim} = max\{1, 0.2 \cdot (|S| - k)\}$.

ШАГ 4. Получить новое решение $S \leftarrow S \backslash S_{elim}$; $K = K - 1$. Улучшить решение алгоритмом k-средних.

КОНЕЦ ПОКА

Также начальное решение S можно получить объединением двух известных решений. Алгоритмы 3.4 и 3.5 являются известными эвристическими процедурами [81,203,233], которые реализуют первый шаг жадного эвристического оператора кроссовера и выполняющий затем запуск Алгоритма 3.3. Алгоритмы 3.4 и 3.5 модифицируют начальное решение вторым известным решением. Фактически Жадная процедура 1 дополняет первое множество

поочередно каждым элементом из второго множества. Жадная процедура 2 объединяет оба множества.

Алгоритм 3.4 Жадная процедура 1 с частичным объединением

Дано: Два набора центров кластеров $S' = \{X'_1, \dots, X'_k\}$ and $S'' = \{X''_1, \dots, X''_k\}$

Цикл: для каждого $i' \in \{\overline{1, K}\}$

Шаг 1. Объединить S'и один элемент из набора S'': $S \leftarrow S \cup \{X''_{i'}\}$.

Шаг 2. Запустить Алгоритм 3.3 с решением S и сохранить полученный результат.

Конец Цикла

Шаг 3. Вернуть лучшие решения, сохраненные на Шаге 2.

Алгоритм 3.5 Жадная процедура 2 с полным объединением множеств

Дано: Два набора центров кластеров $S' = \{X'_1, \dots, X'_k\}$ и $S'' = \{X''_1, \dots, X''_k\}$

Шаг 1. Объединить два набора центров кластеров $S \leftarrow S' \cup S''$.

Шаг 2. Запустить Алгоритм 3.3 с решением *S*.

Данные алгоритмы могут быть включены в различные стратегии глобального поиска. Решения, полученные из первого решения S'и сформированные путем объединения его элементов с элементами второго решения S'', а затем запущенные алгоритмом k-средних, используются в качестве окрестностей, в которых ищется лучшее решение. Таким образом, второе решение S'' является параметром окрестности [81]. Общая структура ГА для k-средних и аналогичных задач может быть описана следующим образом:

Алгоритм 3.6 ГА с алфавитом действительных чисел для задачи k-средних

Дано: Начальная численность популяции N_{POP}

ШАГ 1.Выбрать N_{POP} начальных решений $S_1, \dots, S_{N_{POP}}$, где $|S_i| = k$, и $\{S_1, \dots, S_{N_{POP}}\}$ – случайно выбранное подмножество набора векторов данных. Улучшить каждое начальное решение алгоритмом k-средних и сохранить

соответствующие полученные значения целевой функции (1.1), как переменные $f_k \leftarrow F(S_k), k = \overline{1, N_{POP}}$.

ЦИКЛ

ШАГ 2. ЕСЛИ условие остановки выполнено, **ТОГДА** СТОП. Вернуть решение S_i , $i \in \{\overline{1, N_{POP}}\}$ с минимальным значением f_i .

ШАГ 3. Случайным образом выбрать 2 индекса $k_1, k_2 \in \{\overline{1, N_{POP}}\}$, $k_1 \neq k_2$.

ШАГ 4. Запустить процедуру скрещивания: $S_C \leftarrow Crossingover(S_{k_1}.S_{k_2})$.

ШАГ 5. Запустить процедуру мутации: $S_C \leftarrow Mutation(S_C)$.

ШАГ 6. Запустить выбранную процедуру отбора для изменения набора популяции.

КОНЕЦ ЦИКЛА

На ШАГЕ 6 использован следующий алгоритм:

Алгоритм 3.7 Процедура отбора (Алгоритм 3.6, шаг 6)

ШАГ 1.Случайным образом выбрать 2 индекса $k_4, k_5 \in \{\overline{1, N_{POP}}\}, k_4 \neq k_5$.

ШАГ 2. ЕСЛИ $f_{k_4} > f_{k_5}$ **ТОГДА** $S_{k_4} \leftarrow S_c, f_{k_4} \leftarrow F(S_c)$, **ИНАЧЕ** $S_{k_5} \leftarrow S_c, f_{k_5} \leftarrow F(S_c)$.

Такие алгоритмы работают с небольшой популяцией, а использование других процедур отбора не позволяет улучшить значительно результаты [105,204,208]. В генетических алгоритмах с жадным эвристическим кроссовером [105,206] Алгоритм 3.4 или Алгоритм 3.5 используется в качестве генетического оператора скрещивания. Эти процедуры требуют больших вычислительных ресурсов из-за многократного выполнения алгоритма k-средних. В случае крупномасштабных задач и очень строгих ограничений по времени ГА с жадным эвристическим кроссоверооператором выполняют только несколько итераций. Размер популяции

обычно небольшой, 10-25 хромосом. Динамично растущие популяции [105,206] способны улучшить результаты.

ГА с жадной эвристикой для задач p-медиан и k-средних можно описать следующим образом.

Алгоритм 3.8. ГА с жадной эвристикой для задач p-медиан и k-средних (модификации GA-FULL, GA-ONE и GA-MIX)

Дано: Численность популяции N_{POP}.

Шаг 1. Установить $N_{iter} \leftarrow 0$. Выбрать набор начальных решений $\{S_1, \dots, S_{N_{POP}}\}$, где$|S_i| = k$. Улучшить каждое начальное решение алгоритмом k-средних и сохранить соответствующие полученные значения целевой функции (1.1), как переменные $f_k \leftarrow F(S_k), k = \overline{1, N_{POP}}$. В данной работе начальное значение популяции N_{POP}=5.

Цикл

Шаг 2. ЕСЛИ условие остановки выполнено, **ТОГДА** СТОП. Вернуть решение S_i ,$i \in \{\overline{1, N_{POP}}\}$ с минимальным значением f_i , **ИНАЧЕ** установить численность популяции следующим образом: $N_{iter} \leftarrow N_{iter} + 1$; $N_{POP} \leftarrow max\{N_{POP}, \lceil\sqrt{1 + N_{iter}}\rceil\}$; **ЕСЛИ** N_{POP}изменился, **ТОГДА** сгенерировать новый $S_{N_{POP}}$, как описано в Шаге 1.

Шаг 3. Случайным образом выбрать 2 индекса $k_1, k_2 \in \{\overline{1, N_{POP}}\}, k_1 \neq k_2$.

Шаг 4. Запустить Алгоритм 3.4 (для GA-ONE*) или Алгоритм 3.5 (для GA-FULL*) с решениямиS_{k_1}и S_{k_2}. Для GA-MIX*, Алгоритм 3.4 или Алгоритм 3.5 выбираются случайным образом с равной вероятностью. Получить новое решениеS_c.

Шаг 5. $S_C \leftarrow Mutation(S_C)$. По умолчанию процедура мутации не используется.

Шаг 6. Запустить Алгоритм 3.7

КОНЕЦ ЦИКЛА

---

*GA-ONE – генетический алгоритм с жадной эвристикой с частичным объединением, GA-FULL–генетический алгоритм с жадной эвристикой с полным объединением, GA-MIX–случайный выбор алгоритмов 3.4 или 3.5

Этот алгоритм использует динамически растущую популяцию. В нашей новой версии шага 5 оператор перекрестной мутации выглядит следующим образом.

Algorithm3.9 Оператор перекрестной мутации для Шага 5 Алгоритма 3.8 (модификации GA-FULL-MUT, GA-ONE-MUT и GA-MIX-MUT)

Шаг 1. Запустить алгоритм k-средних для случайно выбранного начального решения, чтобы получить решение S'.

Шаг 2.Запустить Алгоритм 3.4 (для GA-ONE) или Алгоритм 3.5 (для GA-FULL) с решениями S_cи S''. Получить новое решениеS'_c.

Шаг 3. ЕСЛИ $F(S'_c) < F(S_c)$, **ТОГДА** $S_c \leftarrow S'_c$.

3.4 Результаты вычислительных экспериментов с перекрестной мутацией для задачи k-средних

Проведены вычислительные эксперименты с классическими наборами данных из репозиториев UCI (Machine Learning Repository) [237], Basic Benchmark repositories [236] а также с данными образцов промышленной продукции [173].

Europe (169309 векторов данных размерностью 2) - классификация европейских навыков, компетенций, квалификаций и профессий;

Ionosphere (351 векторов данных размерностью 35) - классификация радиолокационных возвратов из ионосферы (прогнозирование высокоэнергетических структур в атмосфере по данным антенны);

Результаты неразрушающих тестовых испытаний сборных производственных партий электрорадиоизделий (ЭРИ), проведенных в специализированном тестовом центре АО «ИТЦ - НПО ПМ» (г. Железногорск) для комплектации бортовой аппаратуры космических аппаратов, состав которых заранее известен: набор интегральных схем 5514ВС1Т2-9А5 (10 партий, 91 векторов данных

размерностью 173); набор интегральных схем 1526TLI (10 партий, 1234 векторов данных размерностью 157).

Новые модификации трех ГА (GA-FULL-MUT, GA-ONE-MUT и GA-MIX-MUT) сравнивались с известными алгоритмами j-means и k-средних (в режиме мультистарта), ГА без мутации (GA-FULL, GA-ONE и GA-MIX), алгоритмы АГ для задачи k-средних с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями, образованными применением жадных агломеративных эвристических процедур (k-GH-VNS1, k-GH-VNS2, k-GH-VNS3), а также для задачи j-means (j-means GH-VNS1, j-means GH-VNS2). Для всех наборов данных выполнено 30 попыток запуска каждого алгоритма. По каждому алгоритму рассчитаны минимальное (Min), максимальное (Max), среднее (Average) значения и среднеквадратичное отклонение (Std.Dev.) целевой функции.

Лучшие значения новых алгоритмов (*) выделены жирным шрифтом, лучшие значения известных алгоритмов указаны курсивом, подчеркнуты наиболее оптимальные значения целевой функции (таблицы 3.2-3.6). Для подтверждения статистической значимости преимущества (↑⇑) и недостатка (↓⇓) новых алгоритмов над известными алгоритмами использованы U-критерий Манна-Уитни (↑↓↕) [238] и t-критерий Стьюдента (⇑⇓⇳) [239]. Уровень значимости 0,99.

Таблица 3.2 – Результаты вычислительных экспериментов для набора данных Europe (169309 векторов данных размерностью 2), 30 кластеров, ограничение по времени 4 часа

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
j-means	7,51477E+12	7,60536E+12	7,56092E+12	29,764E+9
k-means	7,54811E+12	7,57894E+12	7,56331E+12	13,560E+9
k-GH-VNS1	***7,49180E+12***	7,49201E+12	***7,49185E+12***	***0,073E+9***
k-GH-VNS2	7,49488E+12	7,52282E+12	7,50082E+12	9,989E+9
k-GH-VNS3	***7,49180E+12***	7,51326E+12	7,49976E+12	9,459E+9
j-means-GH-VNS1	***7,49180E+12***	7,49211E+12	***7,49185E+12***	0,112E+9
j-means-GH-VNS2	7,49187E+12	7,51455E+12	7,4962E+12	8,213E+9
GA-FULL-MUT*	7,49293E+12	7,49528E+12	7,49417E+12	0,934E+9
GA-MIX-MUT*	***7,49177E+12***	7,49211E+12	***7,49186E+12***	0,117E+9
GA-ONE-MUT*	***7,49177E+12***	7,49188E+12	***7,49182E+12***	***0,042E+9***

Таблица 3.3 – Результаты вычислительных экспериментов для набора микросхем 5514ВС1Т2-9А5 (91 векторов данных размерностью 173), группировка в 10 однородных партий (кластеров), ограничение по времени 2 минуты

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
j-means	7060,45	7085,67	7073,55	8,5951
k-means	7046,33	7070,83	7060,11	8,8727
k-GH-VNS1	***7001,12***	7009,53	7004,48	4,3453
k-GH-VNS2	***7001,12***	7010,59	7002,26	2,9880
k-GH-VNS3	***7001,12***	7009,53	7003,01	3,1694
j-means-GH-VNS1	***7001,12***	7001,12	***7001,12***	***0,0000***
j-means-GH-VNS2	***7001,12***	7011,94	7003,88	4,4990
GA-FULL-MUT*	***7001,12***	7001,27	***7001,24***	***0,0559***
GA-MIX-MUT*↕⇳	***7001,12***	7001,12	***7001,12***	***0,0000***
GA-ONE-MUT*↕⇳	***7001,12***	7001,12	***7001,12***	***0,0000***

Таблица 3.4 – Результаты вычислительных экспериментов для набора данных Ionoshere (351 векторов данных размерностью 35), 10 кластеров, ограничение по времени 1 минута, расстояние Махаланобиса

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
k-means	9253,2467	9304,2923	9275,3296	13,5569
k-GH-VNS1	***9083,6662***	9153,0192	9121,0728	20,6875
k-GH-VNS2	***9085,8065***	9144,3779	9112,2959	14,6803
k-GH-VNS3	9090,5465	9128,4111	9109,8492	***10,2740***
GA-FULL	9117,5695	9175,1517	9142,8457	15,3522
GA-FULL-MUT*	9098,8748	9157,0265	9136,6556	16,1343
GA-MIX	9095,1540	9141,6417	9114,1280	13,0638
GA-MIX-MUT*	9102,8695	9138,2243	9113,4571	***8,6361***
GA-ONE	***9078,6460***	9115,9342	***9099,7687***	***10,1104***
GA-ONE-MUT*↕⇳	***9073,1919***	9120,1842	***9101,6286***	12,8542

Таблица 3.5 – Результаты вычислительных экспериментов для набора интегральных схем 5514ВС1Т2-9А5 (91 векторов данных размерностью 173), группировка в 10 однородных партий (кластеров), ограничение по времени 2 минуты, расстояние Махаланобиса

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
k-means	7289,7935	7289,8545	7289,8296	0,0153
k-GH-VNS1	7289,7366	7289,8024	7289,7648	***0,0147***
k-GH-VNS2	7289,7472	7289,8021	7289,7792	0,0153
GA-FULL	7289,7474	7289,8134	7289,7742	0,0175
GA-FULL-MUT*	***7289,7062***	7289,7754	7289,7508	0,0181
GA-MIX	7289,7319	7289,7771	7289,7501	***0,0133***
GA-MIX-MUT*	7289,7227	7289,7772	***7289,7496***	***0,0149***
GA-ONE	7289,7228	7289,7796	***7289,7494***	0,0159
GA-ONE-MUT*↕⇳	***7289,7147***	7289,7752	***7289,7466***	0,0165

Таблица 3.6 – Результаты вычислительных экспериментов для набора интегральных схем 1526TLI (1234 векторов данных размерностью 157), группировка в 10 однородных партий (кластеров), ограничение по времени 2 минуты

Алгоритм	Значение целевой функции			
	Min	Max	Average	Std.Dev.
j-means	***43841,97***	43843,51	43842,59	0,4487
k-means	43842,10	43844,66	43843,38	0,8346
k-GH-VNS1	***43841,97***	43844,18	43842,34	0,9000
k-GH-VNS2	***43841,97***	43844,18	43843,46	1,0817
k-GH-VNS3	***43841,97***	43842,10	***43841,99***	***0,0424***
j-means-GH-VNS1	***43841,97***	43841,97	***43841,97***	***0,0000***
j-means-GH-VNS2	***43841,97***	43844,18	43842,19	0,6971
GA-FULL-MUT*	***43841,97***	45009,09	44620,29	569,14
GA-MIX-MUT*	***43841,97***	45009,09	44542,31	591,74
GA-ONE-MUT*↓⇓	***43841,97***	45009,09	44363,83	583,63

Проведенные вычислительные эксперименты показывают, что ГА с жадным агломеративным оператором скрещивания с новой идеей процедуры мутации превосходят ГА без мутации по полученному значению целевой функции.

3.5 Результаты экспериментов с параллельной реализацией алгоритмов

CUDA (Compute Unified Device Architecture) - платформа параллельных вычислений и модель программирования, специально разработанная фирмой NVIDIA для общих вычислений на графических процессорах (GPU), которая позволяет существенно увеличить вычислительную производительность [231]. Параллельные (CUDA) реализации алгоритмов k-средних известны [240,241], этот подход использован в экспериментах данной работы [242] (Приложение Г). Все остальные алгоритмы были реализованы на центральном процессоре, для экспериментов использовались классические наборы данных из репозиториев UCI (Machine Learning Repository) [237], Basic Benchmark repositories [236] а также с данными образцов промышленной продукции [173].

Individual Household Electric Power Consumption (IHEPC, 2075259 векторов данных размерностью 9) - данные об индивидуальном домашнем потреблении

электроэнергии за несколько лет, нормализованные данные (0–1), столбцы «дата» и «время» удалены;

SUSY (5000000 векторов данных размерностью 18) –данные для задачи классификации, заключающейся в том, чтобы различить сигнальный процесс, который производит суперсимметричные частицы, и фоновый процесс, который этого не делает. Нормализованные данные (0–1). Здесь мы не принимаем во внимание истинную маркировку, предоставляемую базой данных, и используем этот набор данных для поиска внутренней структуры данных;

Chess (Король-Ладья против Короля-Пешки, 3196 логических векторов данных размерностью 36);

BIRCH3 [10] (100000 векторов данных размерностью 2) - группы точек случайного размера на плоскости;

Europe (169309 векторов данных размерностью 2) - классификация европейских навыков, компетенций, квалификаций и профессий;

Mopsi-Joensuu (6014 векторов данных размерностью 2) - данные о расположении пользователей.

Тестовая система состояла из процессора Intel Core 2 Duo E8400, 16 ГБ ОЗУ, графического процессора NVIDIA GeForce GTX1050ti с 4096 МБ ОЗУ, производительность с плавающей запятой 2138 Гфлопс. Для всех наборов данных выполнено 30 попыток запуска каждого алгоритма (таблицы 3.7-3.12).

Новые модификации трех ГА с новым оператором мутации (третья группа): k-GA-FULL-MUT, k-GA-ONE-MUT и k-GA-MIX-MUT сравнивались с известными алгоритмами, включая генетические алгоритмы с жадным эвристическим кроссовером (первая группа): j-means и k-средних (в режиме мультистарта), ГА без мутации (k-GA-FULL, k-GA-ONE и k-GA-MIX), алгоритмы автоматической группировки для задачи k-средних с комбинированным применением алгоритмов поиска с чередующимися рандомизированными окрестностями, образованными применением жадных агломеративных эвристических процедур (k-GH-VNS1, k-GH-VNS2, k-GH-VNS3). Для алгоритмов, запущенных в режиме мультистарта (j-means и k - средних), фиксировались

только лучшие результаты, достигнутые в каждой попытке. А также результаты эксперимента сравнивались с генетическими алгоритмами с жадным эвристическим кроссовером и известными операторами мутации (вторая группа): k-GA-xxx-m1 для равномерной случайной мутации и k-GA-xxx-m2 для исправления ошибок [243], где ген заменяется на случайно выбранную точку данных. Эксперименты проведены с различными значениями вероятности мутации μ.

По каждому алгоритму рассчитаны минимальное (Min), максимальное (Max), среднее (Average) и медианное (Median) значения и среднеквадратичное отклонение (Std.Dev.) целевой функции. Первоначальный размер популяции для всех генетических алгоритмов составлял $N_{POP} = 10$ хромосом.

В каждой группе алгоритмов подчеркнуты наилучшее среднее и медианное значения целевой функции (1.1). Лучшие алгоритмы второй и третьей групп сравнивались с лучшим алгоритмом первой группы. Для подтверждения статистической значимости преимущества (↑⇑) и недостатка (↓⇓) использованы U-критерий Манна-Уитни [238] (↑↓↕) и t-критерий Стьюдента [239] (⇑⇓⇳). Уровень значимости 0,99.

При сравнительном анализе эффективности алгоритмов выбор единицы времени играет важную роль. Астрономическое время, затрачиваемое алгоритмом, сильно зависит от особенностей его реализации, способности компилятора оптимизировать программный код и способности оборудования выполнять код определенного алгоритма. Алгоритмы часто оцениваются путем сравнения количества выполненных итераций (например, количества поколений населения для ГА) или количества оценок целевой функции. В нашем случае некоторые алгоритмы не являются эволюционными, поэтому сравнение количества поколений недопустимо. Сравнение вычислений целевой функции также не совсем корректно. Во-первых, алгоритм k-средних, который потребляет почти все время процессора, не вычисляет (1.1) напрямую. Во-вторых, во время работы жадного оператора агломерационного кроссовера количество центроидов изменяется (уменьшается с $2k$ до k или с $k + 1$ до k), а также изменяется время,

затрачиваемое на вычисление целевой функции. Поэтому мы все же выбрали астрономическое время в качестве шкалы для сравнения алгоритмов. Более того, во всех алгоритмах использована одна и та же реализация алгоритма k-средних, запускаемого при одинаковых условиях.

Таблица 3.7 – Результаты вычислительных экспериментов для набора данных SUSY (5000000 векторов данных размерностью 18), 25 кластеров, ограничение по времени 500 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-means (в режиме мульстарта)	924115,13	925244,06	924548,63	924317,69	419,9655
j-means (в режиме мульстарта)	926428,75	928250,50	927328,31	927256,00	741,0420
k-GH-VNS1	924227,38	924317,56	924276,83	924265,25	41,0833
k-GH-VNS2	924115,13	925141,50	924822,84	924994,22	410,9538
k-GH-VNS3	924712,69	925243,88	924997,23	925141,50	222,6291
k-GA-FULL	925134,50	925243,88	925198,99	925243,81	56,0092
k-GA-MIX	924227,38	925141,50	924635,38	924265,31	473,6178
k-GA-ONE	924115,13	924317,50	924251,27	924265,25	63,1124
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	924712,69	925243,81	925109,47	925141,50	181,4919
k-GA-FULL-m1,μ=0,01	924265,25	925243,88	925032,71	925141,50	344,9659
k-GA-FULL-m1,μ=0,1	924265,25	925243,88	925015,63	925141,50	341,6232
k-GA-FULL-m2,μ=0,001	924227,38	925243,81	925021,54	925141,50	352,6434
k-GA-FULL-m2,μ=0,01	924877,00	925141,56	925096,58	925141,50	98,6034
k-GA-FULL-m2,μ=0,1	924115,13	925243,88	924864,72	925141,50	485,5295
k-GA-MIX-m1,μ=0,001	924265,25	925243,88	924714,60	924877,00	424,5925
k-GA-MIX-m1,μ=0,01	924115,06	925141,50	924457,88	924227,38	461,0992
k-GA-MIX-m1,μ=0,1	924227,38	925243,81	924539,75	924317,50	448,1161
k-GA-MIX-m2,μ=0,001⇓	924115,13	925243,81	924516,25	924265,25	467,2019
k-GA-MIX-m2,μ=0,01	924115,06	925051,50	924357,08	924317,50	319,4935
k-GA-MIX-m2,μ=0,1↑	924115,06	924265,25	924200,90	924265,25	80,2563
k-GA-ONE-m1,μ=0,001⇳	924227,38	924317,63	924282,27	924265,25	35,6455
k-GA-ONE-m1,μ=0,01	924115,13	925864,13	924626,94	924317,50	661,5964
k-GA-ONE-m1,μ=0,1⇳	924115,06	924317,56	924231,88	924265,25	85,8271
k-GA-ONE-m2,μ=0,001	924227,38	925925,75	924933,53	924876,94	709,7990
k-GA-ONE-m2,μ=0,01	924115,06	925243,81	924518,29	924317,50	466,6610
k-GA-ONE-m2,μ=0,1	924265,25	924891,75	924384,63	924317,50	224,9720
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT	924115,13	924994,00	924381,57	924227,38	383,1728
k-GA-MIX-MUT	924227,38	925243,88	924804,63	925140,75	486,4890
k-GA-ONE-MUT↑⇑	924115,06	924265,25	924184,66	924227,38	66,4470

Таблица 3.8 – Результаты вычислительных экспериментов для набора данных IHEPC, 100 кластеров, ограничение по времени 1000 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-Means (multistart)	3609,0322	3767,3699	3682,9703	3702,0381	66,0620
j-Means	3222,0024	3469,1904	3330,8628	3308,4663	95,1764
k-GH-VNS1	3194,3730	3218,3042	3200,5752	3194,9126	10,2875
k-GH-VNS2	3192,3491	3193,9951	3193,4711	3193,2563	0,7125
k-GH-VNS3	3192,9253	3195,6731	3194,7855	3195,6541	1,2618
k-GA-FULL	3194,5671	3196,0510	3195,5192	3195,6592	0,5586
k-GA-MIX	3193,8230	3194,9165	3194,6958	3194,9133	0,4879
k-GA-ONE	3194,2986	3194,9141	3194,7907	3194,9136	0,2751
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	3195,1061	3196,8482	3196,0125	3196,2146	0,8129
k-GA-FULL-m1,μ=0,01	3195,1055	3195,6616	3195,3321	3195,1191	0,2999
k-GA-FULL-m1,μ=0,1	3195,1050	3197,3811	3196,2253	3196,3418	0,8776
k-GA-FULL-m2,μ=0,001	3194,2142	3197,0143	3195,4257	3195,5142	0,9382
k-GA-FULL-m2,μ=0,01	3194,3091	3197,0327	3195,5808	3195,6631	1,0298
k-GA-FULL-m2,μ=0,1	3194,0044	3196,3257	3195,2396	3195,1074	0,8542
k-GA-MIX-m1,μ=0,001	3194,0125	3196,0512	3195,0221	3194,9287	0,7481
k-GA-MIX-m1,μ=0,01	3193,8232	3196,1799	3194,9487	3194,9133	0,8346
k-GA-MIX-m1,μ=0,1	3193,9868	3195,0732	3194,3911	3194,2983	0,4045
k-GA-MIX-m2,μ=0,001	3194,0418	3195,9175	3194,9819	3194,7105	0,8723
k-GA-MIX-m2,μ=0,01↓⇓	3193,0942	3194,9136	3194,0628	3193,8240	0,8194
k-GA-MIX-m2,μ=0,1	3194,2979	3194,9136	3194,6118	3194,4761	0,2839
k-GA-ONE-m1,μ=0,001	3194,7247	3196,4271	3195,5692	3195,1289	0,7018
k-GA-ONE-m1,μ=0,01	3194,5459	3196,2983	3195,3459	3195,0737	0,7263
k-GA-ONE-m1,μ=0,1	3194,5449	3213,1353	3198,7382	3195,0747	8,0666
k-GA-ONE-m2,μ=0,001	3194,3081	3195,4285	3194,8905	3194,8857	0,4765
k-GA-ONE-m2,μ=0,01	3194,2979	3195,0737	3194,7310	3194,9126	0,3341
k-GA-ONE-m2,μ=0,1	3194,2986	3194,9141	3194,7027	3194,9126	0,2949
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT	3195,0874	3196,7456	3195,8983	3195,6616	0,6483
k-GA-MIX-MUT↕⇕	3193,2725	3194,9143	3193,4996	3193,8240	0,7473
k-GA-ONE-MUT	3194,9131	3195,0774	3195,0426	3195,0737	0,0724

Таблица 3.9 – Результаты вычислительных экспериментов для набора данных Chess (3196 векторов данных размерностью 36), 50 кластеров, ограничение по времени 10 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-Means (multistart)	6926,22	6958,36	6941,13	6939,64	11,2781
j-Means	6938,97	6987,53	6962,71	6961,96	18,6573
k-GH-VNS1	6851,11	6855,66	6853,08	6853,21	1,5482
k-GH-VNS2	6851,07	6857,08	6853,96	6854,35	2,4912
k-GH-VNS3	6851,15	6859,06	6854,82	6855,94	3,5286
k-GA-FULL	6864,33	6867,14	6865,68	6865,66	1,2282

Продолжение таблицы 3.9

Алгоритм	Значение целевой функции				
	Min	Min	Min	Min	Min
k-GA-MIX	6851,41	6858,32	6854,64	6854,56	2,9540
k-GA-ONE	6851,44	6860,75	6856,01	6856,16	4,0019
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	6860,04	6870,22	6867,09	6867,79	2,6577
k-GA-FULL-m1,μ=0,01	6858,87	6870,85	6866,83	6867,13	3,3960
k-GA-FULL-m1,μ=0,1	6861,89	6871,42	6868,57	6868,84	2,3854
k-GA-FULL-m2,μ=0,001	6863,43	6872,63	6868,37	6867,97	2,4871
k-GA-FULL-m2,μ=0,01	6865,95	6871,79	6868,40	6868,34	1,9573
k-GA-FULL-m2,μ=0,1	6863,01	6869,63	6867,21	6867,82	1,9583
k-GA-MIX-m1,μ=0,001	6851,29	6856,30	6853,51	6853,35	1,6152
k-GA-MIX-m1,μ=0,01	6851,67	6860,16	6854,67	6854,23	2,7174
k-GA-MIX-m1,μ=0,1	6851,31	6859,33	6853,83	6853,31	2,3047
k-GA-MIX-m2,μ=0,001	6852,11	6859,69	6855,23	6854,35	2,4314
k-GA-MIX-m2,μ=0,01	6851,68	6858,65	6854,49	6853,68	1,9113
k-GA-MIX-m2,μ=0,1	6851,23	6860,69	6854,17	6853,34	2,7578
k-GA-ONE-m1,μ=0,001	6851,35	6859,78	6853,79	6852,97	2,4321
k-GA-ONE-m1,μ=0,01	6851,18	6857,68	6853,48	6853,41	1,7505
k-GA-ONE-m1,μ=0,1↕⇑	6851,17	6858,90	6852,82	6852,02	2,0071
k-GA-ONE-m2,μ=0,001	6851,19	6860,02	6854,86	6854,32	3,2739
k-GA-ONE-m2,μ=0,01	6851,35	6857,28	6854,09	6853,67	1,9327
k-GA-ONE-m2,μ=0,1	6851,26	6854,55	6852,87	6853,18	1,0231
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT	6852,40	6857,80	6855,43	6855,78	1,7635
k-GA-MIX-MUT↑⇑	6851,12	6853,19	6851,61	6851,18	0,8080
k-GA-ONE-MUT	6851,11	6853,81	6852,04	6851,34	1,0307

Таблица 3.10 – Результаты вычислительных экспериментов для набора данных BIRCH3 (100000 векторов данных размерностью 2), 100 кластеров, ограничение по времени 10 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-Means (multistart)	7,83252E+13	9,52953E+13	8,93284E+13	9,02011E+13	52,9723E+11
j-Means	4,03770E+13	6,59148E+13	4,78407E+13	4,69827E+13	58,1576E+11
k-GH-VNS1	3,71492E+13	3,74385E+13	3,72471E+13	3,72063E+13	0,76473E+11
k-GH-VNS2	3,71530E+13	3,72899E+13	3,72045E+13	3,71994E+13	0,27407E+11
k-GH-VNS3	3,71558E+13	3,73439E+13	3,72163E+13	3,72011E+13	0,50354E+11
k-GA-FULL	3,72086E+13	3,73018E+13	3,72411E+13	3,72413E+13	0,25040E+11
k-GA-MIX	3,71538E+13	3,72083E+13	3,71998E+13	3,72071E+13	0,16590E+11
k-GA-ONE	3,71526E+13	3,72083E+13	3,71925E+13	3,72072E+13	0,24657E+11
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	3,72566E+13	3,77255E+13	3,74320E+13	3,74355E+13	1,10195E+11
k-GA-FULL-m1,μ=0,01	3,72566E+13	3,79192E+13	3,74764E+13	3,74532E+13	1,61443E+11

Продолжение таблицы 3.10

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
k-GA-FULL-m1,μ=0,1	3,72893E+13	3,84583E+13	3,76484E+13	3,74674E+13	3,91344E+11
k-GA-FULL-m2,μ=0,001	3,72397E+13	3,85824E+13	3,76392E+13	3,75036E+13	3,95201E+11
k-GA-FULL-m2,μ=0,01	3,72859E+13	3,75769E+13	3,74216E+13	3,74359E+13	0,78992E+11
k-GA-FULL-m2,μ=0,1	3,72555E+13	3,78012E+13	3,74676E+13	3,74355E+13	1,56028E+11
k-GA-MIX-m1,μ=0,001	3,71491E+13	3,73835E+13	3,72213E+13	3,72072E+13	0,52620E+11
k-GA-MIX-m1,μ=0,01	3,71967E+13	3,72074E+13	3,72039E+13	3,72070E+13	0,03821E+11
k-GA-MIX-m1,μ=0,1	3,71466E+13	3,72074E+13	3,72007E+13	3,72072E+13	0,15455E+11
k-GA-MIX-m2,μ=0,001	3,71530E+13	3,72072E+13	3,71979E+13	3,72011E+13	0,14772E+11
k-GA-MIX-m2,μ=0,01	3,71466E+13	3,72083E+13	3,72015E+13	3,72072E+13	0,15465E+11
k-GA-MIX-m2,μ=0,1	3,71592E+13	3,72072E+13	3,71980E+13	3,72072E+13	0,16092E+11
k-GA-ONE-m1,μ=0,001	3,72009E+13	3,72102E+13	3,72053E+13	3,72072E+13	0,03291E+11
k-GA-ONE-m1,μ=0,01↕⇳	3,71472E+13	3,72074E+13	3,71938E+13	3,72009E+13	0,22370E+11
k-GA-ONE-m1,μ=0,1	3,71969E+13	3,72869E+13	3,72107E+13	3,72072E+13	0,21359E+11
k-GA-ONE-m2,μ=0,001	3,71681E+13	3,72430E+13	3,72062E+13	3,72072E+13	0,14321E+11
k-GA-ONE-m2,μ=0,01	3,71968E+13	3,73260E+13	3,72133E+13	3,72072E+13	0,31388E+11
k-GA-ONE-m2,μ=0,1	3,71528E+13	3,72077E+13	3,71984E+13	3,72072E+13	0,18669E+11
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT	3,71908E+13	3,73862E+13	3,72381E+13	3,72108E+13	0,61387E+11
k-GA-MIX-MUT↕⇳	3,71977E+13	3,72072E+13	3,72045E+13	3,72071E+13	0,03486E+11
k-GA-ONE-MUT	3,71976E+13	3,72074E+13	3,72052E+13	3,72072E+13	0,03436E+11

Таблица 3.11 – Результаты вычислительных экспериментов для набора данных Europe (169309 векторов данных размерностью 2), 30 кластеров, ограничение по времени 10 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-Means (multistart)	7,54173E+12	7,57541E+12	7,55414E+12	7,54853E+12	11,2422E+9

Продолжение таблицы 3.11

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
j-Means	7,55697E+12	7,71002E+12	7,64233E+12	7,64924E+12	47,5626E+9
k-GH-VNS1	7,49695E+12	7,50850E+12	7,50287E+12	7,50346E+12	3,4668E+9
k-GH-VNS2	7,50035E+12	7,51500E+12	7,50727E+12	7,50753E+12	4,3317E+9
k-GH-VNS3	7,50346E+12	7,51458E+12	7,50803E+12	7,50919E+12	3,7463E+9
k-GA-FULL	7,50359E+12	7,51505E+12	7,51020E+12	7,50998E+12	2,9113E+9
k-GA-MIX	7,50036E+12	7,50752E+12	7,50350E+12	7,50360E+12	1,5822E+9
k-GA-ONE	7,49831E+12	7,55141E+12	7,50696E+12	7,50372E+12	12,4626E+9
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	7,50359E+12	7,51488E+12	7,51099E+12	7,51182E+12	2,8380E+9
k-GA-FULL-m1,μ=0,01	7,50359E+12	7,51504E+12	7,51057E+12	7,51065E+12	2,7895E+9
k-GA-FULL-m1,μ=0,1	7,50362E+12	7,51504E+12	7,51020E+12	7,51065E+12	3,1105E+9
k-GA-FULL-m2,μ=0,001	7,50359E+12	7,51505E+12	7,51003E+12	7,50919E+12	3,4939E+9
k-GA-FULL-m2,μ=0,01	7,50077E+12	7,51503E+12	7,50926E+12	7,51108E+12	4,6759E+9
k-GA-FULL-m2,μ=0,1	7,49595E+12	7,51417E+12	7,50831E+12	7,50919E+12	4,5349E+9
k-GA-MIX-m1,μ=0,001	7,50102E+12	7,50723E+12	7,50468E+12	7,50472E+12	2,0215E+9
k-GA-MIX-m1,μ=0,01	7,50066E+12	7,50652E+12	7,50321E+12	7,50361E+12	1,9696E+9
k-GA-MIX-m1,μ=0,1	7,50091E+12	7,50704E+12	7,50382E+12	7,50375E+12	1,9825E+9
k-GA-MIX-m2,μ=0,001	7,49812E+12	7,50851E+12	7,50478E+12	7,50528E+12	2,7402E+9
k-GA-MIX-m2,μ=0,01	7,49506E+12	7,50520E+12	7,50257E+12	7,50353E+12	2,5055E+9
k-GA-MIX-m2,μ=0,1↕⇳	7,49573E+12	7,50677E+12	7,50256E+12	7,50251E+12	3,4842E+9
k-GA-ONE-m1,μ=0,001	7,49840E+12	7,55417E+12	7,50577E+12	7,50353E+12	13,5436E+9
k-GA-ONE-m1,μ=0,01	7,49234E+12	7,54085E+12	7,50941E+12	7,50363E+12	15,8707E+9
k-GA-ONE-m1,μ=0,1	7,50089E+12	7,55332E+12	7,51161E+12	7,50372E+12	17,5433E+9
k-GA-ONE-m2,μ=0,001	7,49731E+12	7,55152E+12	7,50997E+12	7,50270E+12	16,7714E+9
k-GA-ONE-m2,μ=0,01	7,50075E+12	7,55416E+12	7,50721E+12	7,50374E+12	13,1445E+9
k-GA-ONE-m2,μ=0,1	7,49523E+12	7,55026E+12	7,51145E+12	7,50363E+12	18,1598E+9
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT	7,50034E+12	7,50777E+12	7,50361E+12	7,50376E+12	2,4491E+9

Продолжение таблицы 3.11

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
k-GA-MIX-MUT↑⇑	7,49859E+12	7,50399E+12	7,50211E+12	7,50213E+12	1,6438E+9
k-GA-ONE-MUT	7,49966E+12	7,50373E+12	7,50222E+12	7,50233E+12	1,4000E+9

Таблица 3.12 – Результаты вычислительных экспериментов для набора данных Mopsi-Joensuu (6014 векторов данных размерностью 2), 300 кластеров, ограничение по времени 10 секунд

Алгоритм	Значение целевой функции				
	Min	Max	Average	Median	Std.Dev.
известные алгоритмы					
k-Means (multistart)	5,3229	7,5711	6,6616	6,8050	0,6727
j-Means	2,7867	6,0623	4,2377	3,8380	1,1752
k-GH-VNS1	1,9960	3,4027	2,4989	2,3928	0,5308
k-GH-VNS2	3,1070	9,1468	6,9344	7,4990	2,2980
k-GH-VNS3	0,1432	0,2974	0,1836	0,1592	0,0582
k-GA-FULL	0,1912	1,0615	0,4290	0,3564	0,2299
k-GA-MIX	0,7039	2,5733	1,4348	1,2930	0,6968
k-GA-ONE	2,1265	7,3622	4,4011	4,3184	1,3787
генетические алгоритмы с новыми комбинациями известных генетических операторов					
k-GA-FULL-m1,μ=0,001	0,1499	1,0124	0,6256	0,5817	0,2681
k-GA-FULL-m1,μ=0,01	0,1492	0,9951	0,3741	0,2949	0,2330
k-GA-FULL-m1,μ=0,1	0,1440	0,4354	0,2632	0,2699	0,0807
k-GA-FULL-m2,μ=0,001	0,1302	0,3872	0,2017	0,1589	0,0811
k-GA-FULL-m2,μ=0,01	0,1239	0,3441	0,1969	0,1552	0,0726
k-GA-FULL-m2,μ=0,1↕⇕	0,1123	0,2823	0,1622	0,1384	0,0535
k-GA-MIX-m1,μ=0,001	0,41571	5,8679	7,1372	5,9456	5,6937
k-GA-MIX-m1,μ=0,01	0,40612	0,5874	8,7251	6,3025	7,1967
k-GA-MIX-m1,μ=0,1	0,26781	0,2416	3,4529	3,3582	2,7349
k-GA-MIX-m2,μ=0,001	0,6825	7,2932	4,8789	3,9325	2,1523
k-GA-MIX-m2,μ=0,01	0,6373	6,7522	3,2538	3,3473	1,5760
k-GA-MIX-m2,μ=0,1	0,13661	5,2759	3,2085	1,5904	4,4927
k-GA-ONE-m1,μ=0,001	2,1527	8,2427	3,5267	3,2028	1,7428
k-GA-ONE-m1,μ=0,01	2,1336	9,1164	3,9120	3,6403	1,9144
k-GA-ONE-m1,μ=0,1	2,1199	5,9004	3,5209	3,4819	0,9954
k-GA-ONE-m2,μ=0,001	2,51371	4,8735	7,4719	3,8462	4,8952
k-GA-ONE-m2,μ=0,01	2,30102	0,5685	8,3449	4,9896	6,2828
k-GA-ONE-m2,μ=0,1	2,8133	8,4065	4,0472	3,5393	1,5569
генетические алгоритмы с новым оператором мутации					
k-GA-FULL-MUT↑⇑	0,1457	1,0108	0,3529	0,3077	0,2326
k-GA-MIX-MUT	0,2647	3,8119	2,3451	2,6905	1,1910
k-GA-ONE-MUT	2,0543	8,2719	4,2152	4,0904	1,6552

В вычислительных экспериментах ограничение по времени использовалось в качестве условия остановки для всех алгоритмов. Преимущество новых алгоритмов сохраняется независимо от выбранного лимита времени.

Разброс значений во всех таблицах невелик, тем не менее, в некоторых случаях различия статистически значимы. Во всех случаях новые алгоритмы с жадной эвристической мутацией превосходят известные или демонстрируют примерно такую же эффективность (разница в результатах статистически незначима). Более того, новые алгоритмы демонстрируют стабильность результатов (узкий диапазон значений целевой функции). В большинстве случаев наилучшие результаты были достигнуты генетическими алгоритмами с операторами мутации.

Результаты Главы 3

Предложены новые генетические алгоритмы для задачи k-средних с кодированием хромосомы действительными числами, где одна и та же процедура используется как в качестве оператора скрещивания, так и в качестве оператора мутации. Эксперименты показывают, что генетические алгоритмы с жадным агломеративным оператором скрещивания, построенные в соответствии с этой идеей, превосходят генетические алгоритмы без какой-либо процедуры мутации и алгоритмы с равномерной случайной мутацией по полученному значению целевой функции.

При параллельной реализации (с использованием архитектуры CUDA) вычислительные эксперименты показывают, что способы поддержания популяционного разнообразия, такие как генетический оператор мутации, улучшают характеристики генетических алгоритмов с жадным эвристическим оператором скрещивания для крупномасштабной задачи k-средних. Более того, лучшие результаты могут показать алгоритмы с оператором мутации на основе жадного эвристического оператора скрещивания со случайно сгенерированной хромосомой (новая жадная эвристическая мутация).

Новый генетический алгоритм для задачи k-средних с оригинальной идеей использования одной процедуры в качестве оператора скрещивания и оператора мутации демонстрирует более точный и стабильный результат значения целевой функции за фиксированное время выполнения. Представленный генетический алгоритм используется в деятельности АО «Испытательный технический центр – НПО ПМ» (Приложение В).

ГЛАВА 4. АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ ПРОДУКЦИИ НА ОСНОВЕ СИГМОИДАЛЬНОЙ НЕЙРОННОЙ СЕТИ С РЕГУЛЯРИЗАЦИЕЙ

Описанные в предыдущих главах методы и алгоритмы автоматической классификации позволяют с достаточно высокой точностью производить разделение образцов промышленной продукции на однородные партии.

В данной главе предлагается следующая идея организации работы тестового центра: при поступлении в тестовый центр первых закупленных партий изделий путем проведения автоматической классификации выделяются однородные партии, каждая из которых сертифицируется. Обучающее множество (размеченная выборка) формируется из сертифицированных однородных партий. Затем новые партии изделий выделяются с помощью обученного классификатора – искусственной нейронной сети (рисунок 4.1).

Рисунок 4.1 – Схема организации работы тестового центра

Глава посвящена разработке алгоритма обучения двухслойной сигмоидальной нейронной сети с регуляризацией, а также новому подходу к нахождению начального приближения ИНС и сохранению его положительных свойств при обучении с избыточностью описания в виде чрезмерно большого числа нейронов и одновременной недостаточной аппроксимацией в определенных частях области данных.

4.1 Известные подходы к решению задач классификации объектов

Задача классификации является задачей с учителем, это значит, что в ходе обучения испытуемая система принудительно обучается с помощью коллекции примеров «стимул-реакция». Имея такую коллекцию данных, необходимо автоматически расставлять метки на произвольных объектах того же типа, что были в изначальной коллекции.

Задача классификации может возникнуть практически в любой дисциплине, связанной с многомерным анализом данных, в том числе в практических производственных задачах, где требуется высокая точность разделения на однородные партии промышленных продуктов. Одна из таких отраслей - производство электронных компонентов для использования в космической отрасли.

В настоящее время на практике применяется большое количество методов классификации.

Классификация K-ближайших соседей (K-Nearest Neighbors Classification) (KNN) [244] является непараметрическим алгоритмом обучения. В отличие от других алгоритмов обучения, которые позволяют отбрасывать обучающие данные после построения модели, KNN сохраняет все обучающие примеры в памяти. Он начинает работать только во время фазы тестирования / оценки, чтобы сравнить данные тестовых наблюдений с ближайшими тренировочными, что займет значительное время при сравнении каждой точки данных теста. Следовательно, этот метод неэффективен для больших данных; Кроме того, производительность ухудшается при большом количестве переменных из-за «проклятия размерности».

Наивный байесовский классификатор (NBA) относится к вероятностным методам классификации [245-247]. В основе алгоритма лежит предположение о том, что атрибуты классифицируемого объекта независимы. NBA определяется как [248, 249]:

$$c = arg\ max_{k\in\{1,\dots n\}} P(c_k) \prod_{i=1}^{m} P(x_i|c_k),$$

где x_i - характеристики объекта, C=$\{c_1,\dots c_k\}$ – множество классов. Данный классификатор определяет класс c_k с наибольшей условной вероятностью для заданного набора характеристик x_i объекта. В основе данного классификатора лежит теорема Байеса. NBA чаще всего используется для задач с большим числом входных характеристик (большой размерностью исходных данных).

В линейном дискриминантном анализе (LDA) [250] процесс соотнесения объекта к конкретному (существующему) классу состоит из двух этапов [244,245]. На первом шаге анализируется входная информация об объектах в каждом классе, затем выявляется (определяется) набор дискриминантных переменных (характеристики объекта), который позволяет определить различия между классами. На втором шаге осуществляется процесс соотнесения нового объекта к существующему классу. Задача дискриминации классов сводится к построению линейных канонических дискриминантных функций следующего вида [246,251]:

$$c_{ji} = c_{\mathrm{j0}} + c_{\mathrm{j1}}X_{\mathrm{1i}} + c_{\mathrm{j2}}X_{\mathrm{2i}} + \cdots + c_{\mathrm{jm}}X_{\mathrm{mi}} = c_{\mathrm{j0}} + \sum_{i=1}^{m} c_{jm}bX_{mi} \qquad (4.2)$$

где X_i – характеристики объекта, i=1…m, c_{ij} – коэффициенты дискриминантной функции, j = 1,...,min(k−1,m), k - количество классов. Коэффициенты c_{ij} подбираются так, чтобы центроиды (центры) классов были различны. Цель дискриминантного анализа заключается в нахождении оптимальной комбинации дискриминантных переменных, которая наилучшим образом разделила бы существующие классы. Затем объект классифицируется в класс, для которого показатель функции классификации выше. Одно из достоинств данного метода – его применимость для малых размеров выборок.

Метод опорных векторов (SVM) является линейным методом классификации [246]. Основная идея метода заключается в построении оптимальной гиперплоскости, которая разделяет объекты на классы [248,249]. Оптимальная гиперплоскость - плоскость, которая определяет максимальный отступ (margin) между классами. Задача оптимизации (квадратичной оптимизации) определяется как [252]:

$$\|w\|^2 \to min,$$

$$c_i(w \cdot x_i - b) \geq 1, 1 \leq i \leq n.)$$

В основе метода лежит построение линейного порогового классификатора

$$f(x_1, \ldots, x_n) = sign(\sum_{i=1}^{n} w_i x_i - w_0),$$

где x_i – характеристики объекта, w_i – параметры гиперплоскости.

В случае мультикласса на практике часто применяется решающее правило, основанное на разделении задачи на бинарные по схеме One-vs-Rest:

$$f(x) = argmax_{y \in Y}(w_y x + b_y),$$

где $Y=(y_1, \ldots, y_n)$ – метки класса.

Bayes Point Machines (BPM) [253] представляют собой тип алгоритма линейной классификации. Основная идея BPM состоит в том, чтобы идентифицировать «средний» классификатор, который способен эффективно и результативно аппроксимировать теоретическое оптимальное среднее по Байесу нескольких линейных классификаторов (на основе их способности обобщать). «Средний» классификатор известен как Bayes point.

Искусственные нейронные сети (ANN) [254, 255] представляют собой вычислительные структуры, которые моделируют простые биологические процессы, ассоциируемые с процессами человеческого мозга [112].

The Learning Vector Quantization algorithm (LVQ) [249, 258]. Алгоритм искусственной нейронной сети, который позволяет выбрать, скольких обучающих экземпляров нужно придерживаться, и точно узнать, как эти экземпляры должны выглядеть.

Логистическая регрессия (LR) относится к линейным методам классификации [246, 249]. С помощью логистической регрессии можно спрогнозировать вероятность наступления события по значениям характеристик объекта (класс исследуемого объекта). Каждому объекту присваивается значение зависимой переменной $y = k$, (k = 1, ..., K), где K - количество классов:

$$\mathrm{P}\{\mathrm{y} = k|\mathrm{X}; \mathrm{W}\} = \frac{e^{W\,X(y=k)}}{\sum_{j=1}^{k} e^{W\,X(y=j)}},$$

$$f(x, w) = \frac{1}{1+e^{-z(x,w)}},$$

$$z(x, w) = w_{p+1} + \sum_{i=1}^{p} x_i w_i,$$

где x_i – компоненты вектора $x \in R^p$ (характеристики объекта), $w=(w_1,w_2,\ldots,w_p,w_{p+1})^T$ набор неизвестных параметров, который можно оценить с помощью метода наименьших квадратов (МНК). Логистическую регрессию можно представить в виде однослойной нейронной сети с сигмоидальной функцией активации, веса которой есть коэффициенты логистической регрессии, а вес поляризации — константа регрессионного уравнения [254].

Decision tree algorithms [246, 249] представляют собой иерархические методы, которые работают путем итеративного разделения набора данных на основе определенных статистических критериев. Цель Decision tree algorithms - максимизировать дисперсию между различными узлами в дереве и минимизировать дисперсию в каждом узле. Некоторые из наиболее часто используемых алгоритмов дерева решений включают в себя Iterative Dichotomizer 3 (ID3), C4.5 и C5.0 (successors of ID3), Automatic Interaction Detection (AID), Chi Squared Automatic Interaction Detection (CHAID), and Classification and Regression Tree (CART).

Boosted decision trees [249] являются формой ансамблевых моделей. Как и другие ансамблевые модели, Boosted decision trees используют несколько деревьев решений для создания предикторов. Каждое из отдельных деревьев решений может быть слабым предиктором. Однако, в сочетании они дают хорошие результаты.

Random forest [249] является ансамблем деревьев решений. В задаче классификации итоговое решение принимается голосованием «по большинству». Все деревья строятся независимо по следующей схеме. Выбирается подвыборка обучающей выборки, по которой строится дерево. Для построения каждого расщепления в дереве просматриваем определенное количество случайных признаков. Выбираем наилучшие признак и расщепление по нему по заранее заданному критерию. Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса). Выбор большего количества деревьев ведет к повышению качества классификации, однако время работы алгоритма также увеличивается. [244].

4.2 Искусственные нейронные сети в задачах классификации

Впервые в своей работе «Логическое исчисление идей, относящихся к нервной активности (A logical calculus of the ideas immanent in nervous activity)» У.С. Мак-Каллок и В. Питтис[257] вводят понятие искусственной нейронной сети. Н.Винер, являющийся автором книги «Кибернетика: Или Контроль и Коммуникация у Животных и Машин (Cybernetics: Or Control and Communication in the Animal and the Machine)», заложил теоретические основы искусственного интеллекта, нейронауки [258]. Д.О. Хеббом предложен первый алгоритм обучения искусственной нейронной сети [259]. Ф. Розенблаттом предложена одна из первых моделей нейронных сетей Perzeptron (Перцептрон, Персептрон) с одним скрытым слоем [260], а также в дальнейших своих исследованиях Ф. Розенблатт представил модели многослойных перцептронов с перекрестными и обратными связями, привел ряд теорем о сходимости перцептрона [261]. Авторы работы «Персептроны (Perceptrons)» [262] М. Минский и С. Пайперт представили ряд своих доказательств о теореме сходимости перцептрона. Большой вклад в развитие искусственных нейронных сетей внес Т. Кохонен. Им предложен класс нейронных сетей, главным элементом которого является слой Кохонена (сети векторного квантования сигналов [263], самоорганизующиеся карты Кохонена (Self-organizing map, SOM) [92], сети векторного квантования с учителем [264]). На примере решения задач в области экологии и геологии Б.В. Хакимов предлагает нелинейную модель нейрона на основе сплайнов [265]. А.И. Галушкиным [266] и П.Дж. Вербосом [267] предложен метод обратного распространения ошибки для снижения доли ошибки работы многослойного перцептрона, а в дальнейшем развит Д. И. Румельхартом, Дж. Е. Хинтом, Р.Дж. Вильямсом [268], С. И. Барцевым, В.А. Охониным [269]. К. Фукусима предложил новую самоорганизующуюся многослойную нейронную сеть Когнитрон [270], а позже модель самоорганизующейся нейронной сети для механизма распознавания образов Неокогнитрон [271]. Дж. Дж. Хопфилд предложил нейронную сеть с обратными связями (сеть Хопфилда) [272,273]. Дж. Хинтоном разработаны

алгоритмы глубокого обучения многослойных нейронных сетей, с использованием машины Больцмана [274], изобретенной совместно с Т. Сейновски. А.Н. Горбанем, А. Зиновьевым и А. Питенко представлена новая технология визуализации данных – метод «Упругих карт» [275-280]. В отличие от SOM Кохонена метод позволяет точнее регулировать геометрические свойства карты.

Искусственный нейрон составляет основу искусственной нейронной сети. Функционирование нейрона определяется его текущим состоянием, подобно тому, как это происходит в нервной системе, где нервные клетки могут находиться в возбужденном или заторможенном состоянии. У каждого нейрона есть группа синапсов (входных связей), соединенных с выходами других нейронов, ядро нейрона составляет ячейка нелинейного преобразователя, а на выходе – аксон, с которого сигнал (возбуждения или торможения) поступает на входные синапсы следующих нейронов. Общая структура нейрона приведена на рисунке 4.2.

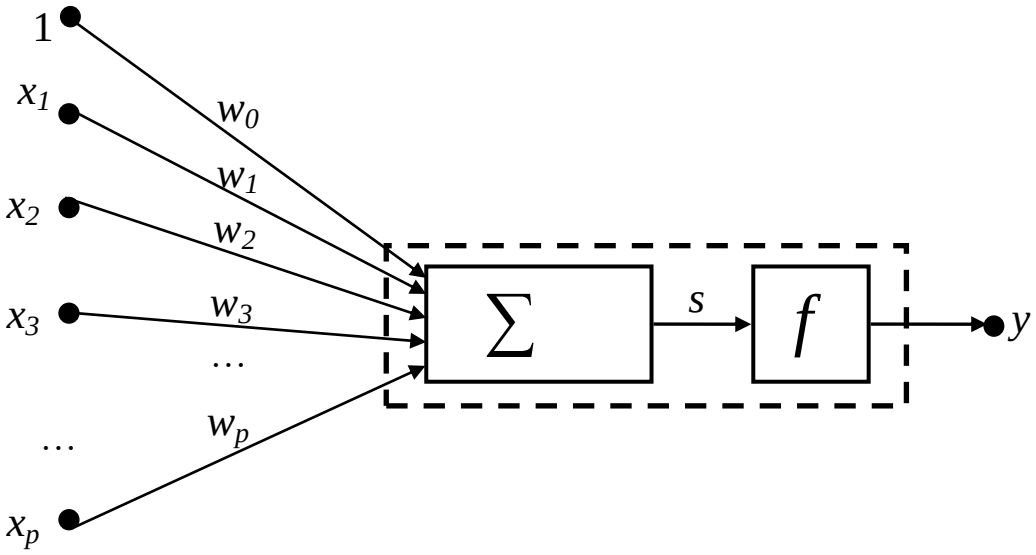


Рисунок 4.2 – Искусственный нейрон

Функционирование синапсов характеризуется величиной синаптической связи или весом синапса w_i. Синапс вычисляет произведение входного сигнала x_i на синаптический вес и передает получившееся произведение на выход. Далее

сигналы попадают на вход сумматора (∑). Соответственно, текущее состояние нейрона определяется, как взвешенная сумма его входов:

$$s = w_0 + \sum_{j=1}^{p} x_j w_j \ .$$

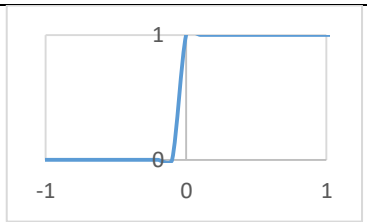
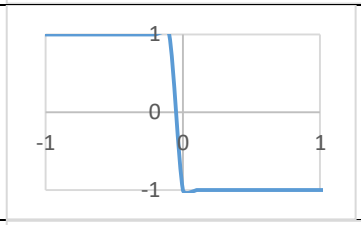
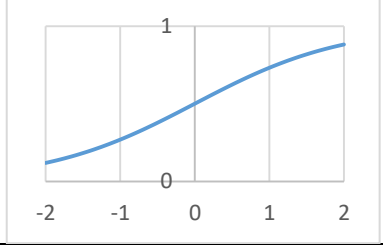
Следующий элемент нейрона – нелинейный преобразователь (*f*). Соответственно, выход нейрона есть функция его состояния $y = f(s)$.

Математическое описание модели искусственного нейрона может быть записано в виде:

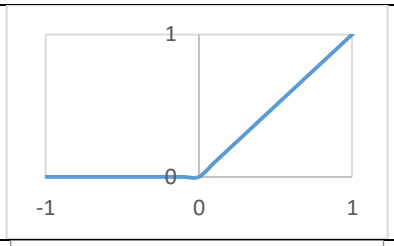
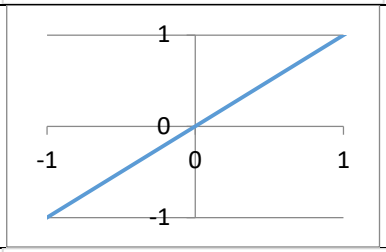
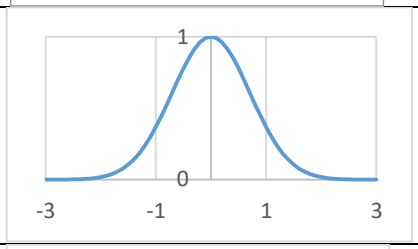
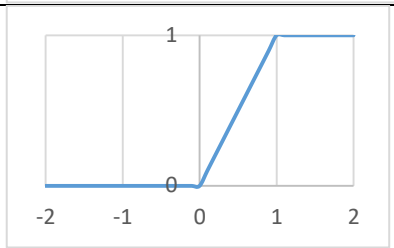
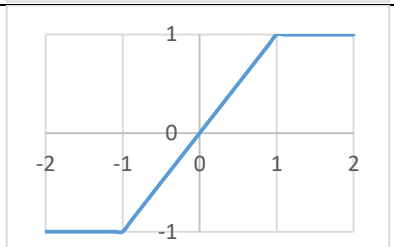
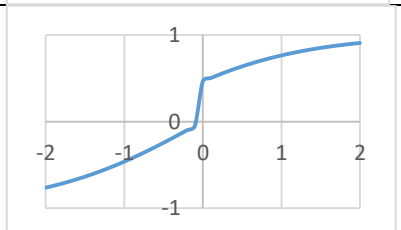
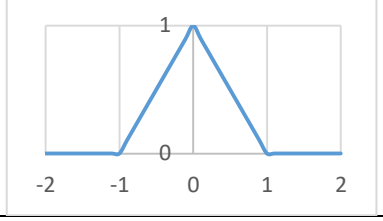
$$y = f(w_0 + \sum_{j=1}^{p} x_j w_j \ ).$$

Функция нелинейного преобразователя называется функцией активации нейрона или его передаточной функцией. Функция активации может иметь различный вид (таблица 4.1) [116, 281]. Вид функции во многом определяет способность ИНС решать прогностические и классификационные задачи при малом числе элементов. Кроме того, существуют ограничения в применимости некоторых функций активации в ИНС к наиболее эффективным методам обучения на базе алгоритмов обратного распространения ошибки.

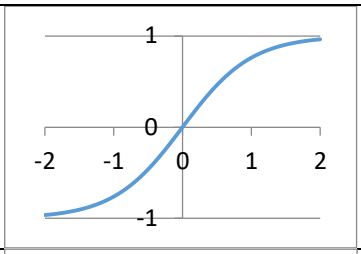
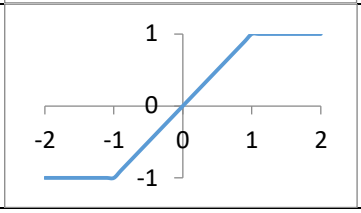
Таблица 4.1 – Примеры функции нелинейного преобразователя

Название функции	Формула	Область значений	График
Пороговая	$f(s) = \begin{cases} 0, s < 0, \\ 1, s \geq 0 \end{cases}$	[0,+1]	
Сигнатурная (знаковая)	$f(s) = \begin{cases} 1, s > 0, \\ -1, s \leq 0 \end{cases}$	[-1,+1]	
Сигмоидальная (логистическая)	$f(s) = \frac{1}{1 + e^{-as}}$	(0,+1)	

Продолжение таблицы 4.1

Название функции	Формула	Область значений	График
Полулинейная	$f(s) = \begin{cases} s, s > 0, \\ 0, s \leq 0 \end{cases}$	[0,+∞)	
Линейная	$f(s) = s$	(-∞;+∞)	
Радиальная базисная (гауссова)	$f(s) = e^{-s^2}$	(0,+1]	
Полулинейная с насыщением	$f(s) = \begin{cases} 0, s \leq 0, \\ s, 0 < s < 1, \\ 1, s \geq 1 \end{cases}$	(0,+1]	
Линейная с насыщением	$f(s) = \begin{cases} -1, s \leq -1, \\ s, -1 < s < 1, \\ 1, s \geq 1 \end{cases}$	[-1,+1]	
Гиперболический тангенс (сигмоидальная)	$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$	(-1,+1)	
Треугольная	$f(s) = \begin{cases} 1 - \lvert s \rvert, \lvert s \rvert \leq 1, \\ 0, \lvert s \rvert > 1 \end{cases}$	[0,+1]	

Продолжение таблицы 4.1

Название функции	Формула	Область значений	График
Гиперболическая тангенциальная	$f(s)=\frac{2}{1+e^{-2s}}-1$	[-1,+1]	
Симметричная линейная с ограничениями	$f(s)=\begin{cases}-1, s<-1,\\ s, -1\le s\le 1,\\ 1, s>1\end{cases}$	[-1,+1]	

Наиболее распространенной является сигмоидальная функция, функция *S*-образного вида:

$$f(s)=\frac{1}{1+e^{\alpha s}}.$$

При уменьшении α сигмоида становится более пологой, в пределе при α=0 вырождаясь в горизонтальную линию на уровне 0,5, при увеличении α сигмоида приближается по внешнему виду к функции единичного скачка с порогом в точке *s*=0. Значение выходного сигнала нейрона лежит в диапазоне (0,+1). Сигмоидальная функция усиливает слабые сигналы и предотвращает насыщение от больших сигналов [116]. Важным преимуществом сигмоидальной функции активации является ее дифференцируемость на всей области определения, что необходимо как раз для применения алгоритма обратного распространения ошибки и разработанных на его основе градиентных методов обучения ИНС.

Основная особенность любой ИНС – параллельная обработка сигналов. Параллелизм ИНС достигается путем объединения большого числа нейронов в слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой. Расчет выходных сигналов элементов ИНС (синапсов, сумматоров, нелинейных преобразователей) выполняется по слоям.

По архитектуре связей ИНС делят на сети прямого распространения и сети с обратной связью [273,282]. В сетях прямого распространения сигнал

распространяется только от входного слоя к выходному слою [273,282]. К сетям прямого распространения относят однослойный персептрон [116,273], многослойный персептрон [116,273,283], сеть радиальных базисных функций [116,273,284-287]. Они решают такой класс задач, как прогнозирование, кластеризация и распознавание. В свою очередь, в сетях обратного распространения сигнал выходного слоя может распространяться к входному слою. К таким сетям с обратной связью относят соревновательные сети [273], самоорганизующееся карты Кохонена (SOM), сеть Хопфилда [272,273], и модели ART [273].

На рисунке 4.3 представлена однослойная нейронная сеть прямого распространения. На p входов поступают сигналы, преобразуемые в синапсах, и поступающие на сумматоры на m нейронов, образующие единственный слой и выдающие m сигналов:

$$y_i = f\left(w_{i0} + \sum_{j=1}^{p} x_j w_{ij}\right), \text{где } i = 1,..,m$$

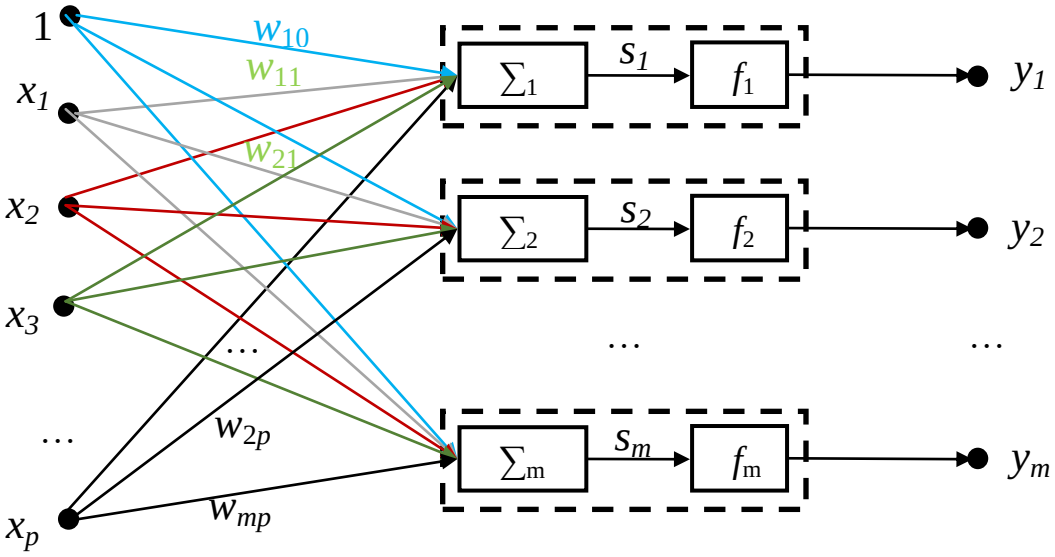


Рисунок 4.3 – Однослойная нейронная сеть

Многослойные нейронные сети прямого распространения состоят из входного, выходного и расположенных между ними одного или нескольких скрытых слоев нейронов [273]. На примере двухслойной нейронной сети рассмотрим принцип ее работы (рисунок 4.4). На p входов поступают сигналы,

преобразуюемые в синапсах и поступающие на сумматоры на m нейронов, образующие скрытый первый слой и выдающие m сигналов. Затем полученные m сигналов, преобразуемые в синапсах и поступающие на сумматоры на k нейронов, образуют скрытый второй слой и выдают k сигналов:

$$y_i = f\left(w_{i0}^{(2)} + \sum_{j=1}^{m} f(s_j) w_{ij}^{(2)}\right), \text{где } i = 1,..,k,$$

$$s_j = w_{i0}^{(1)} + \sum_{j=1}^{p} x_j w_{ij}^{(1)}, \quad j = 1,\ldots,m.$$

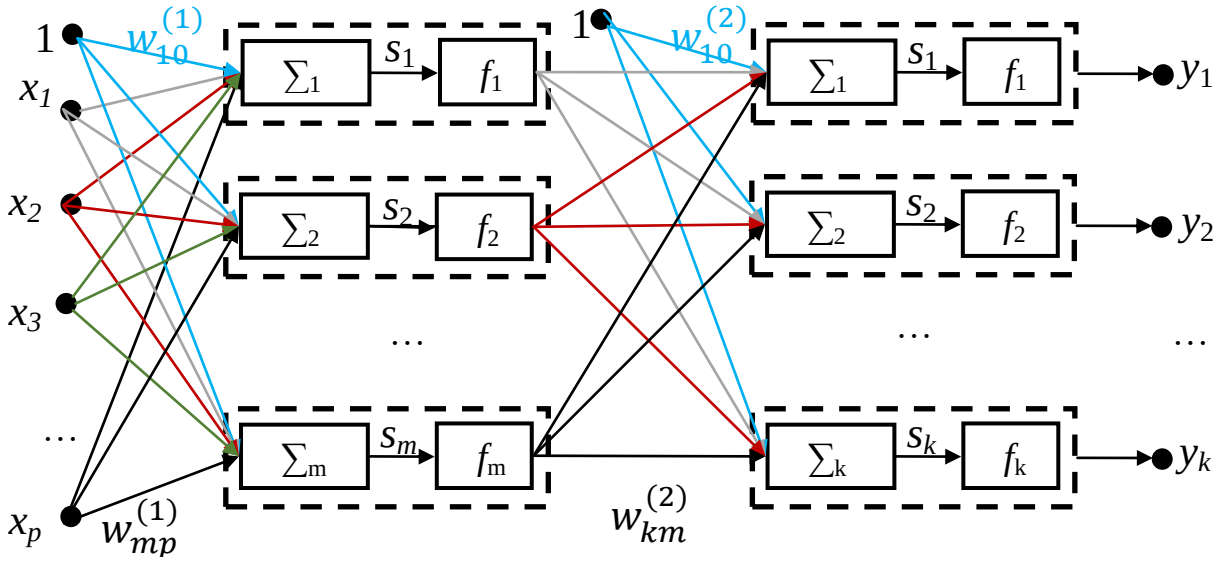


Рисунок 4.4 – Двухслойная нейронная сеть

Обучение нейронной сети

В теории нейронных сетей существуют две актуальных проблемы, одной из которых является выбор оптимальной структуры нейронной сети, а другой построение эффективного алгоритма обучения нейронной сети. Оптимизация нейронной сети направлена на уменьшение объёма вычислений при условии сохранения точности решения задачи на требуемом уровне. Параметрами оптимизации в нейронной сети могут быть: размерность и структура входного сигнала нейросети; синапсы нейронной сети. Вторая проблема заключается в разработке качественных алгоритмов обучения нейросети. Обучение нейронных

сетей – это подбор весовых коэффициентов нейронов. Тип и характер обучения определяются, прежде всего, объемом априорной и текущей информации о среде, в которую «погружена» сеть, а также критерием качества (целевой функцией), при котором свободные параметры нейронной сети адаптируются в результате ее непрерывной стимуляции внешним окружением.

Можно выделить два основных вида обучения: обучение с «учителем», и обучение «без учителя».

Алгоритм обратного распространения ошибки

Обучение многослойного персептрона чаще всего происходит с помощью алгоритма обратного распространения ошибки или его модификации.

Обратное распространение ошибки – системный метод обучения ИНС. В основу метода положено обучение ИНС по образцам с помощью примеров обучающей выборки. Для каждого примера из обучающей выборки считается известным требуемое (целевое) значение выхода ИНС, то есть метод обратного распространения ошибки представляет собой обучение с учителем.

Такое обучение можно представить как решение оптимизационной задачи на минимизацию функции ошибки (невязки) E на всей обучающей выборке путем подстройки обучаемых параметров ИНС – значений весов синапсов w, параметров активационных функций.

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i^o - y_i)^2,$$

где y_i^o – ожидаемое значение выхода на i-м образце выборки, y_i – рассчитываемое значение искусственной нейронной сетью, n – число образцов обучающей выборки.

Минимизация функционала ошибки E в методе обратного распространения ошибки осуществляется с помощью одного из градиентных методов оптимизации, например, «наискорейшего спуска» или его модификаций. При этом веса синапсов изменяются в направлении, обратном направлению наибольшего возрастания функции ошибки – в направлении антиградиента E:

$$W(t+1) = W(t) - \eta \frac{\partial E}{\partial W},$$

где $0<\eta\leq 1$ – «шаг обучения», определяемый пользователем параметр, определяющий скорость обучения, в ряде методов он подбирается итерационно.

Метод обратного распространения ошибки реализуют в двух вариантах. В первом варианте обучаемые параметры пересчитываются после подачи всей обучающей выборки, и ошибка имеет вид:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i^o - y_i)^2.$$

Во втором варианте ошибка пересчитывается после каждого примера:

$$E_i = \frac{1}{2}(y_i^o - y_i)^2.$$

Установим

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}},$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t),$$

Тогда

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_{ij}} * \frac{\partial y_i}{\partial S_j} * \frac{\partial S_j}{\partial w_{ij}}$$

$$y_j = f(S_j) = \frac{1}{1+e^{-S_j}},$$

где y_j –функция активации.

$$f'(S_j) = \frac{\partial}{\partial S_j}\left(\frac{1}{1+e^{-S_j}}\right) = \frac{1}{(1+e^{-S_j})^2}(-e^{-S_j}) = \frac{1}{1+e^{-S_j}} * \frac{-e^{-S_j}}{1+e^{-S_j}} = y_j(1-y_j)$$

$$S_j = \sum W_{ij} * y_i^{n-1} \quad \frac{\partial S_i}{\partial W_{ij}} = y_i^{(n-1)},$$

$$\frac{\partial E}{\partial y_i} = \sum_k \frac{\partial E}{\partial y_k} * \frac{dy_k}{dS_k} * \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{dy_k}{dS_k} * W_{jk}^{(n+1)},$$

при этом суммирование по *k* идет среди нейронов *n*-го слоя.

Введем новое обозначение:

$$\delta_j = \frac{\partial E}{\partial y_k} * \frac{dy_j}{dS_j},$$

$$\delta_j^{n-1} = \left[\sum_k \delta_k^n * W_{jk}^n\right] * \frac{dy_j}{dS_j}.$$

Для внутреннего нейрона

$$\delta_j^n = (d_j^n - y_j^n) * \frac{dy_j}{dS_j}.$$

Для внешнего нейрона

$$\Delta W_{ij}^n = -\eta \delta^{(n)} * y_j^{n-1},$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}.$$

Обучение по методу обратного распространения ошибки производится в соответствии со следующим алгоритмом:

Шаг 1. Производится инициализация начальных значений обучаемых параметров. Обычно начальные значения составляют малые случайные величины.

Шаг 2. На входы ИНС подается один из примеров обучающей выборки. В режиме обычного функционирования ИНС (сигналы распространяются от входов к выходам) производится расчет значений выходов всех нейронов .

Шаг 3. Рассчитываются значения δ_j^n для нейронов выходного слоя по формуле для внешнего нейрона.

Шаг 4. Рассчитываются значения δ_j^n для всех внутренних нейронов.

Шаг 5. Вычисляются приращения весовых коэффициентов ΔW_{ij}^n.

Шаг 6. Скорректировать веса синапсов $W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}$.

Шаг 7. Повторить шаги 2-6 для каждого примера обучающей выборки до тех пор, пока значение функционала ошибки *E* не станет достаточно малым.

Альтернативным условием останова цикла на шаге 7 является невозможность обучить ИНС до приемлемой точности. Это решение принимается исходя из того, что приращения весовых коэффициентов ΔW_{ij}^n станут слишком малы и количество итераций в этой точке станет больше заданного.

Такая ситуация может возникнуть в следующих двух случаях:

1. Алгоритм оптимизации попал в локальный экстремум функционала ошибки *E*. Для того, чтобы исключить данную возможность, необходимо стартовать обучение из другой стартовой конфигурации обучаемых параметров.

2. Архитектура ИНС недостаточна для решения задачи. Если многократный запуск обучения сети все-таки не привел к успешному обучению, то, скорее всего, следует усложнить сеть – увеличить количество нейронов или даже количество слоев.

4.3 Искусственные нейронные сети с регуляризацией

Регуляризация в машинном обучении является методом добавления дополнительных ограничений к условию с целью решить некорректно поставленную задачу [287] (неустойчивость решения [288] и невозможность получения точного решения [280]) или предотвратить переобучение. Чаще всего эта информация имеет вид штрафа за сложность модели [116]. Главная идея регуляризации заключается в стабилизации решения с помощью некоторой вспомогательной неотрицательной функции, которая несет в себе информацию о решении, полученную ранее (априорная информация). Наиболее общей формой априорной информации является предположение о гладкости выходной функции искомого в том смысле, что одинаковый входной сигнал соответствует одинаковому выходному. Включение в алгоритм такой информации называется регуляризацией решения [281,282].

В работах [292-294] А.Н. Тихоновым предложен универсальный способ построения регуляризирующего алгоритма (метод регуляризации), основанный на введении так называемого сглаживающего функционала. Теория регуляризации Тихонова в своем изначальном виде использует два слагаемых.

Слагаемое стандартной ошибки E_S. Первое слагаемое описывает ошибку между ожидаемым значением выхода на i-м образце выборки y_i^o и рассчитываемым значением y_i. Например, ошибку можно определить, как:

$$E_S = \frac{1}{2}\sum_{i=1}^{n}(y_i^o - y_i)^2 = \frac{1}{2}\sum_{i=1}^{n}(y_i^o - F(x))^2,$$

где n – число образцов обучающей выборки; $F(x)$ – функция аппроксимации.

Слагаемое регуляризации E_C. Второе слагаемое зависит от геометрических свойств функции аппроксимации $F(x)$. Ее можно определить следующим образом:

$$E_C = \frac{1}{2}\|DF(x)\|,$$

где D – линейный дифференциальный оператор. || • || – норма в функциональном пространстве, к которому принадлежит *DF(x)*.

Априорная информация о форме решения (функция отображения *F(x)*), включенная в дифференциальный оператор D, обеспечивает его зависимость от конкретной задачи. Оператор D иногда еще называют стабилизатором, так как в задаче регуляризации он стабилизирует решение, делая его гладким и, таким образом, удовлетворяющим свойству непрерывности.

Таким образом, в теории регуляризации необходимо минимизировать величину, состоящую из двух слагаемых (E_C и E_S):

$$E(F(x)) = E_S + \alpha E_C = \frac{1}{2}\sum_{i=1}^{n}(y_i^o - F(x))^2 + \frac{1}{2}\alpha\|DF(x)\|,$$

где α – положительное действительное число, называемое параметром регуляризации; *E(F(x))* — функционал Тихонова. Функционал отображает функции (определенные в соответствующем функциональном пространстве) на ось действительных чисел. Параметр регуляризации α можно рассматривать как индикатор достаточности обучающего набора данных для определения решения. Для $\alpha \to 0$ задача является безусловной и имеет решение, целиком зависящее от набора данных. Для $\alpha \to \infty$ предполагается, что самого априорного ограничения на гладкость, представленного дифференциальным оператором D, достаточно для определения решения. В практических приложениях параметр регуляризации α принимает среднее значение между двумя крайними случаями. Этим определяется влияние на решение как априорной информации, так и данных обучающей выборки. Таким образом, слагаемое регуляризации E_C представляет собой функцию штрафа за сложность модели, влияние которой на окончательное решение определяется параметром регуляризации α.

Считается, что теория регуляризации обеспечивает практическое решение дилеммы смещения и дисперсии. В частности, оптимальный выбор параметра регуляризации α позволяет обеспечить приемлемое соотношение между

смещением и дисперсией модели при решении задачи обучения с учетом некоторой априорной информации.

В работах [22,289,290,292,295-303] рассмотрены различные методы регуляризации

Существуют следующие виды регуляризаций.

1. Квадратичная регуляризация А.Н. Тихонова (Quadratic Tikhonov regularization (L2)). В случае аппроксимации линейной моделью используют регуляризатор Тихонова [292]:

$$L2(W) = \sum_{i=1}^{k} w_{ij}^2, \quad (4.1)$$

где присутствуют параметры линейной части модели. Регуляризатор *L2* в основном используется для подавления больших компонентов вектора W для предотвращения переобучения модели.

2. Модульно линейная регуляризация (Лассо Тибширани) (Modular linear regularization (Lasso) (L1)). Регуляризатор, предложенный в работах [22,297], в основном используется для подавления больших и малых компонентов вектора W

$$L1(W) = \sum_{i=1}^{k} |w_{ij}|. \quad (4.2)$$

3. Негладкая регуляризация (Lγ2) [297-300], которая определяется как:

$$L\gamma 2(W) = \sum_{i=1}^{k} \left(|w_{ij}| + \varepsilon\right)^{\gamma} (\varepsilon = 10^{-6}, \gamma = 0.7). \quad (4.3)$$

Использование *Lγ2* приводит к подавлению в большей мере малых («слабых») компонент вектора *W*. Подобное свойство *Lγ2* позволяет сводить к нулю слабые компоненты, несущественные для описания данных.

Регуляризаторы *L1* и *Lγ2* предназначены для подавления незначимых компонент модели.

4.4 Построение составного алгоритма обучения искусственной нейронной сети

Наряду с моделью k-средних искусственные нейронные сети (ИНС, ANN) представляют собой простые и легко интерпретируемые модели, при этом

обеспечивающие высокое качество решения задач классификации (по критерию AUC ROC-кривой, площадь под кривой ошибок).

Предварительное обучение производится посредством минимизации по W квадратичной ошибки и негладкого сглаживающего функционала:

$$W = argmin_W E(\alpha, W, D), \tag{4.4}$$

$$E(\alpha, W, D) = \sum_{x,y\epsilon D}\big(y - f(x, W)\big)^2 + \alpha L(W),$$

где $D = \{(x^i, y_i) \vee x^i \in R^p, y_i \in R^1, i = 1, \dots, N\}$ – данные наблюдения, α – параметры регуляризации, *f(x,W)* – аппроксимирующая функция, $x \in \mathrm{R}^p$ – вектор данных, $W \in \mathrm{R}^n$ – вектор настраиваемых параметров, p и n – их размерности. $L(W)$ – регуляризатор. Регуляризатор $L(W)$ подавляет компоненты вектора W.

В задаче классификации с помощью ИНС использованы следующие виды регуляризации:

1. Квадратичная регуляризация А.Н. Тихонова $L2(W)$;
2. Модульная линейная регуляризация (Лассо Тибширани) $L1(W)$;
3. Смешанная регуляризация [304], представленная двумя видами регуляризаторов $L\gamma 2(W)$ и $L2(W)$.

$$L\gamma(W, a_1, a_2) = a_1 \times L\gamma 2(W) + a_2 \times L2(W). \tag{4.5}$$

Эти же виды регуляризации представлены в логистической регрессии.

В данной работе обучили двухслойную сигмоидальную нейронную сеть следующего вида:

$$f(x, W) = w_{i0}^{(2)} + \sum_{i=1}^{m} w_{ij}^{(2)} \varphi(s_i), \tag{4.6}$$

где $\varphi(s_i)$ – функция активации нейрона вида

$$\varphi(s_i) = 1/\big(1 + exp(-s_i)\big), \tag{4.7}$$

$$s_i = w_{i0}^{(1)} + \sum_{j=1}^{p} x_j w_{ij}^{(1)}, i = 1, \dots, m. \tag{4.8}$$

где x_j – компоненты вектора $x \in R^p$, $W=((w_{ij}^{(2)}, i=0,\dots,m), (w_{ij}^{(1)}, j=0,\dots,p, i=1,\dots,m))$ – набор неизвестных параметров, которые оцениваются с помощью (4.13), m – число нейронов.

Предлагается составной алгоритм обучения ИНС. Алгоритм ІА предназначен для выбора начального приближения параметров ИНС. Алгоритм основан на

первоначальной оценке параметров нейрона, связанных с выбранными центрами [301]. Это обеспечивает равномерное покрытие области данных рабочими областями нейронов. Наличие регуляризации, даже с чрезмерным количеством параметров по сравнению с объемом данных, позволяет получить приемлемое решение на этом этапе.

Алгоритм 4.1. Алгоритм нахождения начального приближения ИНС (Алгоритм IA)

Требуется: исходные данные $D = \{(x^i, y_i) \vee x^i \in R^p, y_i \in R^1, i = 1, \dots, N\}$, начальное количество нейронов m (m<N).

Шаг 1. Разделить набор исходных данных D на обучающее (DO) и тестовое множество (DT). Принять D = DO.

Шаг 2. Выбрать регуляризатор (4.1), (4.2), (4.3) и установить его начальные параметры;

Шаг3. На наборе данных D установить начальные центры рабочих областей нейронов c_i, $i = 1, \dots, m$. Центры рабочих областей c_i определяются одним из алгоритмов кластеризации. Это гарантирует, что нейроны будут находиться в областях с высокой плотностью данных

Шаг 4. Исключаем свободный член нейрона $w_{ij}^{(0)}$ из (4.8) и используем выражение:

$$s_i = \sum_{j=1}^{p}(x_j - c_i)w_{ij}^{(1)}, i = 1, \dots, m. \quad (4.9)$$

Шаг 5. Определить начальные параметры $w_{ij}^{(1)}$ для (4.9) и $w_i^{(2)}$ для (4.6) с помощью датчика равномерно распределенных случайных чисел для каждого нейрона.

Шаг 6. Решить задачу (4.4) для (4.6), (4.7), (4.9) с фиксированными центрами, используя регуляризатор (4.1), (4.2), (4.3). Это позволяет на первом этапе покрыть всю область данных рабочими областями.

Шаг 7. Вернуться к исходному описанию нейронной сети (4.6), (4.7), (4.8) для определения свободного члена нейрона $w_{ij}^{(0)}$, оставив параметры без изменений:

$$w_{i0}^{(1)} = -\sum_{j=1}^{p} c_i w_{ij}^{(1)}$$

Шаг 8. Окончательный набор параметров обозначаем W^0.

В основном алгоритме поочередно выполняется удаление малозначимых нейронов [303], затем выполняется обучение ИНС:

Алгоритм 4.2. Алгоритм обучения ИНС

Дано: исходные данные $D = \{(x^i, y_i) \vee x^i \in R^p, y_i \in R^1, i = 1, \dots, N\}$, начальное количество нейронов m (m<N).

Шаг 1. Выбрать начальное приближение параметров сети W^0, применив Алгоритм 4.1 (IA)

Шаг 2. Выбрать модель окончательного приближения ИНС. Установить h=m.

Цикл для k=0,1,…,m-1

Шаг 3. Решить задачу (4.4) для (4.6), (4.7), (4.8)

Шаг 4. Установить $S_k = S(W^k, D)$. $S(W^k, D)$ величина среднеквадратичной погрешности определяется:

$$S(W^k, D) = \sqrt{\sum_{x,y \in D} \left(y - f(x, W^k)\right)^2 / N}.$$

Шаг 5. ЕСЛИ $S_k < (1 + \varepsilon ps)S_0$, где $\varepsilon ps = 0.1$

ТОГДА удалить нейрон для которого выполняется $S_k < (1 + \varepsilon ps)S_0$, h=h-*1*

Шаг 6. ЕСЛИ не произошло удаление нейронов**, ТОГДА** удаляем нейрон, который приводит к наименьшему росту S), h=h-*1*.

Шаг 7. ЕСЛИ $h \leq q$**, ТОГДА** выход из цикла

Конец Цикла

Шаг 8. Выбираем $f(x, W^k)$ с числом параметров, не превосходящих N, и имеющих наименьшее значение показателя S_k, в качестве модели окончательного приближения и решаем задачу (4.4) для (4.6), (4.7), (4.8)**.**

4.5 Результаты обучения искусственной нейронной сети с регуляризацией

Для проведения экспериментов использовались данные о кредитных операциях из репозитория UCI Machine Learning Repository [305,306], а также данные образцов промышленной продукции (Микросхемы 1526IE10_002 [173]), описанные в разделе 2.2. С данными наборами проведено три серии экспериментов.

I серия. В экспериментах данной серии участвуют замеренные характеристики микросхемы 1526IE10_002, представленные в разделе 2.2. Исследована смешанная партия промышленной продукции космического назначения, в состав которой входит семь однородных производственных партий. Общее количество изделий во всей выборке составляет 3987 изделий. Каждое изделие содержит информацию о 205 измеряемых характеристиках. Для обучения классификаторов в рассмотрении у каждого изделия остается 68 входных характеристик (признаков). Из 7 однородных партий были составлены следующие комбинации: полная выборка и три вида смешанных партий, состоящих из 4, 3 и 2 однородных партий.

В данном случае в каждой смешанной партии изделий сформировано обучающее и тестовое множество (выборка) в соотношении 2/3 и 1/3.

Для решения поставленной задачи необходимо выбрать методы классификации. Были рассмотрены проведенные исследования в области сравнения методов классификации наборов данных в различных практических приложениях [252,307-311]. Большинство из них выделяют методы SVM, искусственных нейронных систем (ANN) and logistic regri логистической регрессии (LR) как обеспечивающие наиболее высокое качество классификации.

Также, с целью выбора методов классификации в нашей работе проведена предварительная оценка описанных выше алгоритмов классификации на данных промышленной продукции [173] в программной версии MatLab R2019b [312]. По критерию точности классификации (доле верно классифицированных объектов) получены следующие результаты (таблица 4.2).

Таблица 4.2 – Точность предварительной классификации

Метод	Decision Tree	LDA	NBA	SVM	ANN
Точность	0,796	0,851	0,855	0,930	0,842

Обобщив результаты сравнительных обзоров классификационных методов и результатов предварительной классификации, было решено провести исследование с использованием следующих методов: линейный дискриминантный анализ (LDA), наивный байесовский алгоритм (NBA), метод опорных векторов (SVM), логистическая регрессия (LR), искусственные нейронные сети (ANN). При построении моделей логистической регрессии и искусственной нейронной сети для уменьшения эффекта переобучения использованы следующие виды регуляризации: Квадратичная регуляризация А.Н. Тихонова (L2), Модульно линейная регуляризация (Лассо Тибширани) (L1), смешанная регуляризация (Lγ), а также, рассмотрены варианты без регуляризации (NR).

Результаты классификации представлены в таблицах 4.3, 4.4, 4.7. Приведены доли верно классифицированных объектов.

Таблица 4.3 – Точность классификации, доля верно классифицированных изделий. Серия I. Полная выборка (общее количество изделий 3987, обучающая выборка 2567, тестовая выборка 1330)

Партия Модель	1 n=24	2 n=39	3 n=623	4 n=417	5 n=48	6 n=37	7 n=142	Среднее
LDA	1,00	0,95	0,74	0,29	1,00	1,00	0,64	0,57
NBA	0,98	0,95	0,70	0,00	0,92	1,00	0,88	0,53
SVM	0,98	0,97	0,85	0,73	0,95	0,97	0,89	0,83
ANN_Rγ	0,98	0,67	0,89	0,72	0,94	0,95	0,90	0,83
ANN_R2	0,94	0,77	0,86	0,67	0,90	0,97	0,89	0,80
ANN_NR	0,66	0,68	0,82	0,72	0,94	0,82	0,89	0,79
ANN_R1	0,98	0,73	0,87	0,61	0,96	0,97	0,89	0,79
LR_Rγ	0,98	0,97	0,85	0,82	0,95	0,97	0,89	0,87
LR_R2	0,98	0,96	0,85	0,72	0,95	0,98	0,90	0,82
LR_NR	0,80	0,86	0,85	0,78	0,85	0,55	0,90	0,82
LR_R1	0,98	0,97	0,85	0,71	0,95	0,97	0,89	0,82

Таблица 4.4 – Точность классификации. Доля верно классифицированных изделий Серия I. Смешанная выборка из 4 партий изделий (общее количество изделий 446, обучающая выборка 299, тестовая выборка 147) и Смешанная выборка из 3 партий (общее количество изделий 300, обучающая выборка 201, тестовая выборка 99)

	Смешанная выборка из 4 партий					Смешанная выборка из 3 партий			
Модель	1 n=23	2 n=38	5 n=49	6 n=37	Среднее	1 n=23	2 n=39	6 n=37	Среднее
LDA	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
NBA	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
SVM	1,00	0,90	0,99	1,00	0,97	1,00	1,00	0,99	0,99
ANN_Rγ	1,00	0,94	0,99	0,99	0,97	1,00	1,00	1,00	1,00
ANN_R2	1,00	0,97	0,99	1,00	0,99	1,00	1,00	1,00	1,00
ANN_NR	0,98	0,94	1,00	1,00	0,98	1,00	1,00	0,98	0,99
ANN_R1	0,84	0,74	0,67	0,75	0,73	0,77	1,00	0,62	0,80
LR_Rγ	0,97	0,90	0,99	0,96	0,95	1,00	0,96	0,94	0,96
LR_R2	1,00	0,94	0,99	1,00	0,98	1,00	0,99	1,00	0,99
LR_NR	1,00	0,95	0,99	1,00	0,98	1,00	1,00	0,98	0,99
LR_R1	0,97	0,90	0,99	0,98	0,95	1,00	0,99	0,96	0,98

Результаты показали, что в случае с небольшим количеством однородных партий в выборке (2-4) практически все рассмотренные методы классификации позволяют решать задачу классификации с высокой точностью (0.95-1.00). С ростом числа классов (однородных партий) в выборке проявляются преимущества модели классификации на основе логистической регрессии с негладкой регуляризацией, а также метода опорных векторов (SVM).

Серия II. Модели обучены на характеристиках микросхемы, представленной в I серии экспериментов. Обучающая и тестовая выборка формируется также в соотношении 2/3 и 1/3, но не в порядке поступления в тестовый центр, а случайным образом (таблицы 4.5, 4.6, 4.7).

Таблица 4.5 – Точность классификации. Доля верно классифицированных изделий Серия II. Полная выборка (общее количество изделий 3987, обучающая выборка 2567, тестовая выборка 1330)

Партия Модель	1 n=24	2 n=39	3 n=623	4 n=417	5 n=48	6 n=37	7 n=142	Среднее
LDA	0,91	0,87	0,86	0,59	0,98	0,92	0,94	0,80
NBA	1,00	0,95	0,76	0,31	0,94	0,95	0,92	0,66
SVM	0,83	0,98	0,97	0,96	0,88	0,92	0,97	0,96
ANN_Rγ	0,83	0,95	0,93	0,83	0,93	0,95	0,99	0,90
ANN_R2	0,83	0,95	0,93	0,98	0,87	0,95	0,97	0,95

Продолжение таблицы 4.5

Партия Модель	1 n=24	2 n=39	3 n=623	4 n=417	5 n=48	6 n=37	7 n=142	Среднее
ANN_NR	0,77	0,86	0,90	0,88	0,55	0,93	0,88	0,88
ANN_R1	0,87	0,98	0,93	0,99	0,87	0,92	0,93	0,95
LR_Rγ	0,80	0,98	0,97	0,97	0,87	0,92	0,97	0,96
LR_R2	0,81	0,98	0,98	0,94	0,88	0,92	0,98	0,96
LR_NR	0,76	0,94	0,76	0,96	0,83	0,89	0,68	0,83
LR_R1	0,76	0,98	0,97	0,96	0,88	0,89	0,97	0,96

Таблица 4.6 – Точность классификации. Доля верно классифицированных изделий. Серия II. Смешанная выборка из 4 партий изделий (общее количество изделий 446, обучающая выборка 298, тестовая выборка 148) и Смешанная выборка из 3 партий (общее количество изделий 300, обучающая выборка 201, тестовая выборка 99)

	Смешанная выборка из 4 партий					Смешанная выборка из 3 партий			
Модель	1 n=23	2 n=38	5 n=49	6 n=37	Среднее	1 n=23	2 n=39	6 n=37	Среднее
LDA	1,00	1,00	1,00	0,97	0,99	1,00	1,00	1,00	0,98
NBA	1,00	1,00	1,00	0,97	0,99	1,00	1,00	0,99	0,99
SVM	1,00	0,96	1,00	0,98	0,98	1,00	0,98	0,99	0,98
ANN_Rγ	1,00	0,99	1,00	0,99	0,99	1,00	1,00	0,99	0,99
ANN_R2	1,00	0,99	1,00	0,99	0,99	1,00	0,99	1,00	0,99
ANN_NR	1,00	0,97	1,00	0,99	0,99	1,00	0,99	0,99	0,98
ANN_R1	0,84	0,74	0,67	0,75	0,74	0,78	1,00	0,63	0,80
LR_Rγ	0,86	0,94	1,00	0,98	0,95	1,00	0,97	0,98	0,98
LR_R2	1,00	0,97	1,00	0,98	0,98	1,00	0,99	0,99	0,98
LR_NR	1,00	0,99	1,00	0,98	0,99	1,00	0,99	1,00	0,99
LR_R1	0,88	0,96	0,99	0,97	0,95	1,00	0,98	0,99	0,98

Таблица 4.7 – Точность классификации. Доля верно классифицированных изделий. Смешанная выборка из двух партий (общее количество изделий 187, обучающая выборка 125, тестовая выборка 62)

Модель	Смешанная выборка из 2 партий					
	Серия I			Серия II		
	1 n=23	2 n=39	Среднее	1 n=23	2 n=39	Среднее
LDA	1,00	1,00	1,00	1,00	1,00	1,00
NBA	1,00	1,00	1,00	1,00	1,00	1,00
SVM	1,00	1,00	1,00	1,00	1,00	1,00
ANN_Rγ	1,00	1,00	1,00	1,00	1,00	1,00
ANN_R2	1,00	1,00	1,00	1,00	1,00	1,00
ANN_NR	1,00	1,00	1,00	1,00	1,00	1,00
ANN_R1	0,63	0,63	0,63	0,63	0,63	0,63
LR_Rγ	1,00	1,00	1,00	1,00	1,00	1,00
LR_R2	1,00	1,00	1,00	1,00	1,00	1,00
LR_NR	1,00	1,00	1,00	1,00	1,00	1,00
LR_R1	1,00	1,00	1,00	1,00	1,00	1,00

Результаты показали, что высокая точность классификации также достигается при любом методе классификации, если количество однородных партий в выборке не превышает 4. Для выборки, содержащей все однородные партии, лучшими оказываются модели классификации на основе логистической регрессии с регуляризациями, а также метод опорных векторов (SVM).

Серия III. В третьей серии исследования используются данные о кредитных операциях из репозитория UCI Machine Learning Repository [237]. Количество записей в выборке «German Credit Data» (German) составляет 1000 строк, каждая запись имеет 24 признака (атрибута). Выборка «Australian Credit Data» (Australion) содержит 690 строк и 15 атрибутов. В таблице 4.8 представлены результаты классификации со следующими моделями классификации: LDA, NBA, SVM, LR, ANN. Нейронные сети рассмотрены в конфигурации с 1,2,5 и 10 нейронами. Модели логистической регрессии и нейронных сетей представлены с разными видами регуляризации (NR, R2, R1, Rγ).

В данном исследовании проявляется преимущество модели SVM. Кроме того, высокую точность классификации показали модели на основе логистической регрессии с регуляризацией R1 и Rγ на данных скоринга немецких банков и модели классификации на основе нейронных сетей с 10 нейронами на данных скоринга австралийских банков.

Таблица 4.8. – Точность классификации данных из репозитория машинного обучения UCI, доля верно классифицированных изделий

Модель	German	Australian	Модель	German	Australian
LDA	0,74	0,86	ANN_2_Rγ	0,75	0,86
NBA	0,74	0,88	ANN_2_R2	0,75	0,85
SVM	0,87	0,91	ANN_2_NR	0,73	0,84
ANN_1_Rγ	0,76	0,87	ANN_2_R1	0,70	0,67
ANN_1_R2	0,75	0,86	ANN_5_Rγ	0,76	0,85
ANN_1_NR	0,74	0,85	ANN_5_R2	0,71	0,87
ANN_1_R1	0,73	0,83	ANN_5_NR	0,70	0,83
LR_Rγ	0,76	0,86	ANN_5_R1	0,70	0,62
LR_R2	0,75	0,87	ANN_10_Rγ	0,76	0,86
LR_NR	0,76	0,86	ANN_10_R2	0,71	0,87
LR_R1	0,76	0,86	ANN_10_NR	0,66	0,80
			ANN_10_R1	0,70	0,87

Для оценки качества алгоритмов классификации применили количественный показатель AUC (area under the curve) ROC-кривой (receiver operating characteristic). ROC-кривая визуализирует результат работы алгоритма (модели) классификации (таблицы 4.9 – 4.10).

Таблица 4.9 – Значения AUC для классификационных моделей серий I и серий II

Модель	Серия I				Серия II			
	Полная выборка	Смешанная выборка			Полная выборка	Смешанная выборка		
		4 партии	3 партии	2 партии		4 партии	3 партии	2 партии
LDA	0,990	1	1	1	0,960	1	1	1
NBA	0,980	1	0,990	1	0,950	0,990	1	1
SVM	0,928	0,997	0,998	1	0,954	0,746	0,992	1
ANN_Rγ	0,260	0,997	1	1	0,981	1	0,995	1
ANN_R2	0,911	1	1	1	0,989	1	0,997	1
ANN_NR	0,905	1	0,957	1	0,996	1	0,993	1
ANN_R1	0,914	0,935	0,998	0,992	0,988	0,923	0,907	1
LR_Rγ	0,926	0,993	0,998	1	0,948	1	0,992	1
LR_R2	0,926	0,997	1	1	0,954	0,997	0,992	1
LR_NR	0,925	1	0,998	1	0,968	1	0,997	1
LR_R1	0,928	0,992	0,998	1	0,951	0,992	0,992	1

Чем выше значение показателя AUC от значения 0,5, тем процесс классификации прошел качественнее. Если значение показателя AUC близко к 0,5, то классификатор непригоден. Кривые ROC широко используются, потому что они относительно просты для понимания, они охватывают более одного аспекта классификации и позволяют визуально и с минимальными усилиями сравнивать производительность различных моделей.

Таблица 4.10. Значения AUC для классификационных моделей серий III

Модель	German	Australian	Модель	German	Australian
LDA	0,790	0,930	ANN_2_Rγ	0,875	0,915
NBA	0,750	0,900	ANN_2_R2	0,842	0,894
SVM	0,770	0,960	ANN_2_NR	0,713	0,840
ANN_1_Rγ	0,846	0,921	ANN_2_R1	0,801	0,910
ANN_1_R2	0,804	0,900	ANN_5_Rγ	0,951	0,964
ANN_1_NR	0,732	0,859	ANN_5_R2	0,920	0,925
ANN_1_R1	0,796	0,893	ANN_5_NR	0,784	0,858
LR_Rγ	0,801	0,899	ANN_5_R1	0,800	0,892
LR_R2	0,792	0,887	ANN_10_Rγ	0,995	0,990
LR_NR	0,791	0,889	ANN_10_R2	0,982	0,947
LR_R1	0,793	0,889	ANN_10_NR	0,725	0,917
			ANN_10_R1	0,801	0,893

Сравнительные результаты классификации и качества обученных моделей показаны на рисунке 4.5 для I и II серии экспериментов, на рисунках 4.6, 4.7 для III серии экспериментов.

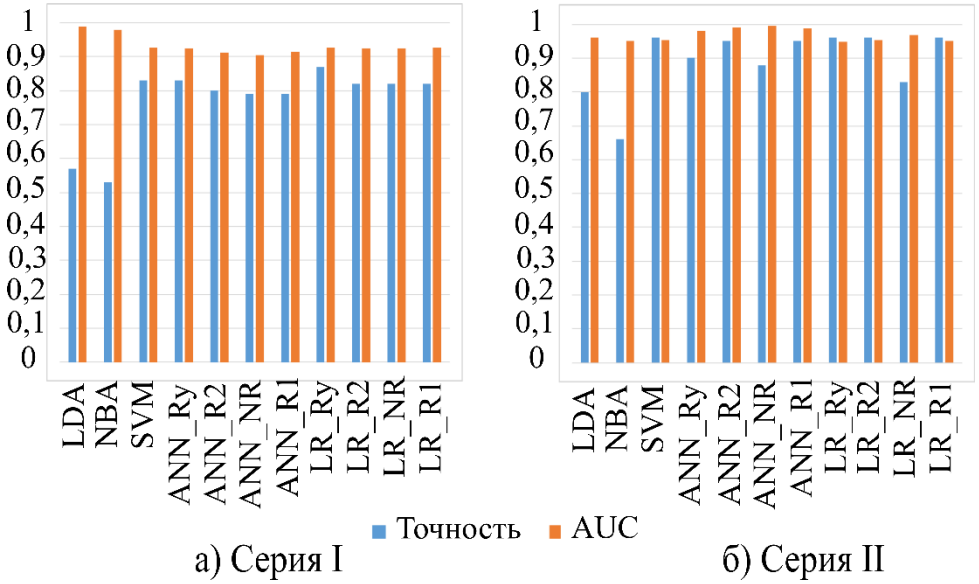


Рисунок 4.5 – Точность и значение AUC для a) Серии I; b) Серии II

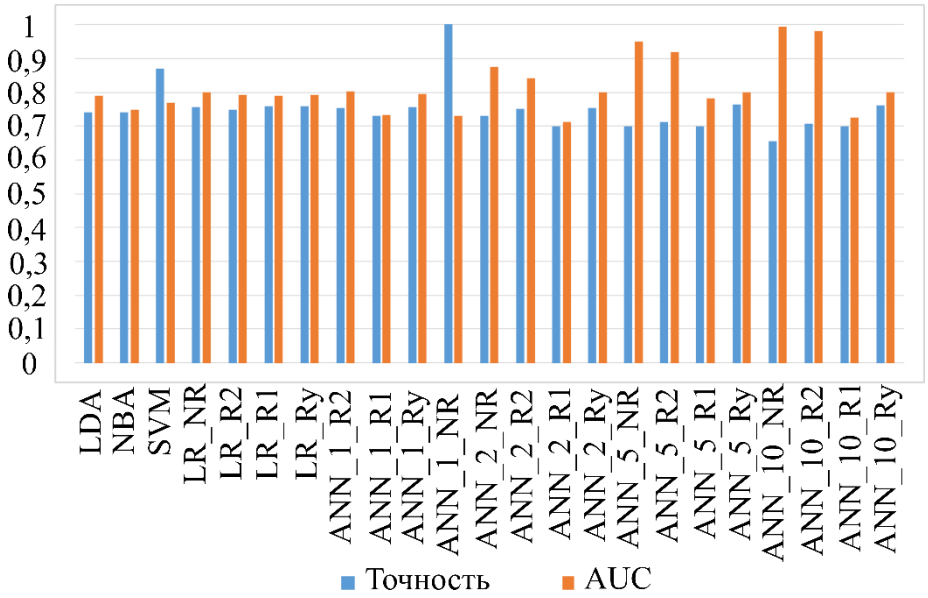


Рисунок 4.6 – Точность и значение AUC для Серии III (German Credit Data)

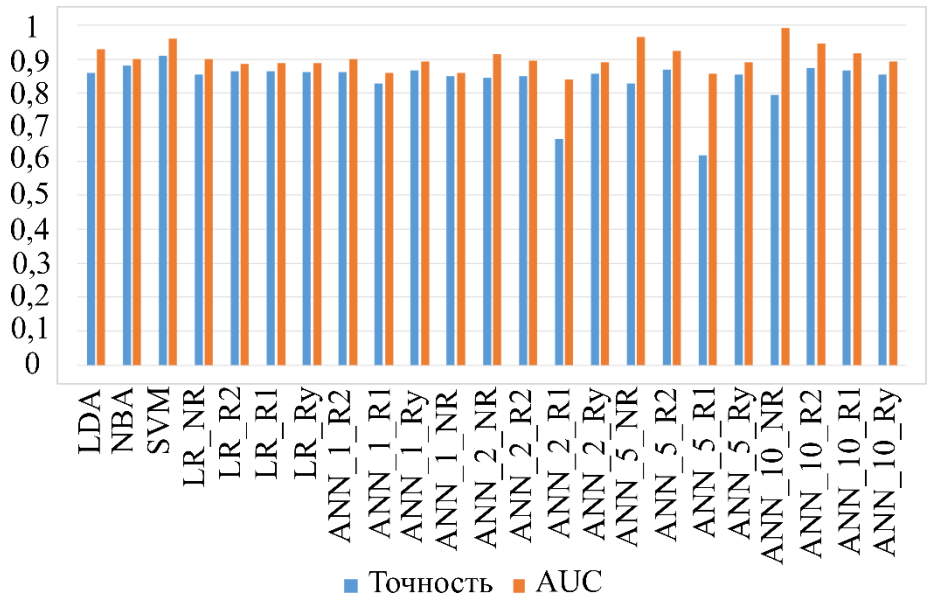


Рисунок 4.7 – Точность и значение AUC для серии III (Australian Credit Data)

Результаты вычислительного эксперимента над данными промышленной продукции свидетельствуют об эффективности предложенного алгоритма. В случае с небольшим количеством однородных партий в выборке (2-4) практически все рассмотренные методы классификации позволяют решать задачу классификации с высокой точностью (0,95-1,00). Показано, что при использовании нейросетевых моделей классификации и моделей логистической регрессии наилучшие результаты достигаются с применением предложенных в работе алгоритмов с регуляризацией.

Результаты Главы 4

Разработан алгоритм обучения двухслойной сигмоидальной искусственной нейронной сети с регуляризацией. Предложен новый подход нахождения начального приближения искусственной нейронной сети и сохранение его положительных свойств при обучении с избыточностью описания в виде чрезмерно большого числа нейронов. Предложен метод автоматической группировки, основанный на классификации с помощью моделей, обученных на сертифицированных производственных партиях промышленной продукции.

Сертифицированные партии при этом формируются с помощью алгоритмов кластеризации.

Рассмотрены различные виды моделей классификации, в том числе с применением различных типов регуляризации.

Результаты показали, что в случае с небольшим количеством однородных партий в выборке (2-4) практически все рассмотренные методы классификации позволяют решать задачу классификации с высокой точностью (0,95-1,00). С ростом числа классов (однородных партий) в выборке проявляются преимущества модели классификации на основе логистической регрессии с регуляризацией, также высокую точность классификации показала модель опорных векторов. Выявлено, что метод классификации на основе логистической регрессии остается лучшим по критерию AUC независимо от способа деления множества данных на обучающую и тестовую выборку. Показано, что при использовании нейросетевых моделей классификации и моделей логистической регрессии наилучшие результаты достигаются с применением предложенных в работе алгоритмов с регуляризацией.

ЗАКЛЮЧЕНИЕ

В диссертационной работе предложены модели и алгоритмы, которые позволяют повысить точность (по индексу Рэнда) автоматической группировки промышленной продукции и получить более стабильные (по коэффициенту вариации) значения целевой функции в сравнении с известными моделями и алгоритмами автоматической группировки.

Цель диссертации достигнута путем решения поставленных задач, а именно:

1. Сравнительный анализ известных алгоритмов решения задач автоматической группировки объектов с повышенными требованиями к точности разделения, показал, что существующие решения не позволяют одновременно получать высокую точность разделения объектов, получать стабильные результаты при многократных запусках алгоритмов, а также иметь высокую скорость работы алгоритмов. В области классификации существует проблема, связанная с точностью классификации в условиях малого количества данных и большого количества признаков.

2. Построена новая модель автоматической группировки объектов на основе модели k-средних с расстояниями Махаланобиса и обучаемой ковариационной матрицей. Предложенный новый алгоритм, основанный на оптимизационной модели k-средних с расстояниями Махаланобиса и обучаемой ковариационной матрицей, позволяет снизить долю ошибок (повысить индекс Рэнда) при выявлении однородных производственных партий продукции. А также продемонстрировано, что модель с применением методов факторного анализа позволяет уменьшить размерность исходных данных и уменьшить долю ошибок кластеризации до нуля для задачи разбиения на две однородные партии.

3. Разработан новый генетический алгоритм для задачи k-средних с оригинальной идеей использования одной процедуры в качестве оператора скрещивания и оператора мутации, который демонстрирует более точный и стабильный результат значения целевой функции за фиксированное время выполнения.

4. Разработан алгоритм обучения двухслойной сигмоидальной искусственной нейронной сети с регуляризацией. Предложен новый подход нахождения начального приближения искусственной нейронной сети и сохранение его положительных свойств при обучении с избыточностью описания в виде чрезмерно большого числа нейронов. Полученные результаты демонстрируют высокую точность классификации с применением предложенных в работе алгоритмов регуляризации при наличии малоинформативных признаков.

Таким образом, в диссертации решена задача повышения точности работ систем классификации промышленной продукции за счет применения усовершенственных математических моделей и алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

1. Reinsel, D. The Digitization of the World From Edge to Core. An IDC White Paper / D. Reinsel, J. Gantz, J. Rydning // – IDC. – 2018 [Электронный ресурс] Режим доступа: URL https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (дата обращения 28.10.2020).

2. Tabachnick, B.G. Using Multivariate Statistics, fifth ed. / B.G.Tabachnick, L.S. Fidell // Boston: Allyn and Bacon. – 2007. – P.980.

3. Duda, R. Pattern Classification, second ed./ R. Duda., P. Hart, D. Stork// New York: John Wiley and Sons. – 2001. – P. 680.

4. Воронцов, К.В. Алгоритмы кластеризации и многомерного шкалирования [Электронный ресурс]/ К.В. Воронцов// Курс лекций.– МГУ.– 2007.– Режим доступа: http://www.ccas.ru/voron/ download/Clustering.pdf.

5. Апраушева, Н.Н. Статистическая система автоматической классификации / Н.Н. Апраушева, И.А. Горлач, Д.Е. Иванов // Вычислительный центр РАН, Москва. – 1999. – С. 27.

6. Гильфанов, А.Ф. Применение алгоритма кластеризации для определения безопасных моделей пассажирских самолетов / А.Ф. Гильфанов // Всероссийской научно-практической конференции с международным участием «Новые технологии, материалы и оборудование российской авиакосмической отрасли». – 2018. – С.571 – 575.

7. Зотин, А.Г. Обнаружение опухоли мозга на основе мрт с применением метода нечеткой кластеризации с-средних / А.Г. Зотин, Ю.А. Хамад, С.В. Кириллова, М.А. Курако, К.В. Симонов // Медицина и высокие технологии. – 2018. – №.1. – С.20 – 28.

8. Луценко, Е.В. Агломеративная когнитивная кластеризация симптомов и синдромов в ветеринарии / Е. В. Луценко // Политематический сетевой электронный научный журнал кубанского государственного аграрного университета. – 2018. – №139. – С.99 – 116.

9. Левенков, К.О. Классификации пациентов с хроническим пиелонефритом по степени тяжести заболевания на основе кластерного анализа /

К.О. Левенков, А.С. Турбин, Е.Н. Коровин // Материалы международной научной конференции «Наука россии: цели и задачи». – 2017. – С. 14 – 17.

10. Шамсутдинова, Т. М. Технологии интеллектуального анализа статистических данных (на примере кластерного анализа показателей сельскохозяйственного производства субъектов РФ) / Т.М. Шамсутдинова // Материалы международной научно-практической конференции «Современные научно-практические решения в АПК». –2017. – С.467 – 473.

11. Мулянова, Ю.Н. Кластерный анализ в сельском хозяйстве / Ю.Н. Мулянова, С.Н. Косников // ECONOMICS. – №5(37). – С.47 – 53.

12. Klyuchko, O.M. Cluster analysis in biotechnology / O.M. Klyuchko // Biotechnologia Acta. – 2017. – V.10 №5 .– P.5 – 18.

13. Лепский, А.И. Сравнительный анализ алгоритмов кластеризации лейкоцитов по FS и SS параметрам при цитофлуориметрическом исследовании крови // А.И. Лепский // Информационные технологии. – 2020. – Т.26 №1. – С.56-61.

14. Фомина, Е.Е. Методы многомерной статистики в социологических и социально-экономических исследованиях. Учебное пособие / Е. Е. Фомина // Тверь: Издательство: Тверской государственный технический университет.–2019. – 112 с.

15. Фомина, Е.Е. Возможности метода деревьев классификации при обработке социологической информации / Е.Е. Фомина // Гуманитарный вестник. – 2018. – №11(73). – С. 1-10.

16. Орлов, В.И. Алгоритмическое обеспечение поддержки принятия решений по отбору изделий микроэлектроники для космического приборостроения: монография / В. И. Орлов, Л. А. Казаковцев, И. С. Масич, Д. В. Сташков; Сиб. гос. аэрокосмич. ун-т. – Красноярск. – 2017. – 250 с.

17. Orlov, V.I. Fuzzy clustering of EEE components for space industry/ V.I. Orlov, D.V. Stashkov, L.A. Kazakovtsev, A.A.Stupina // IOP Conference Series: Materials Science and Engineering. – 2016. – Vol.155. [Электронный ресурс] Режим

доступа: URL http://iopscience.iop.org/article/10.1088/1757-899X/155/1/012026/pdf (дата обращения 31.10.2020)

18. Kazakovtsev, L.A. Improved model for detection of homogeneous production batches of electronic components/ L.A. Kazakovtsev, V.I. Orlov, D.V. Stashkov, A.N. Antamoshkin, I.S. Masich // IOP Conference Series: Materials Science and Engineering. – 2017, DOI: 10.1088/1757-899x/255/1/012004.

19. Хараханов, В.А. Исследование способов кластеризации деталей машиностроения на основе нейронных сетей / В.А. Харахинов, С.С. Сосинская // Программная Инженерия. – 2017. – Т. 8 № 4. – С.170 – 176.

20. Кабанов, С.И. Акустико-эмиссионный контроль авиационных конструкций: монография / С.И. Кабанов, А.Е. Кареев, Е.Ю. Лебедев, В.Л. Кожемякин, И.С. Рамазанов, Б.М. Харламов // Научно-техническое издательство «Машиностроение». – Москва. – 2008. – 439 с.

21. Бериков, В.Б. Современные тенденции в кластерном анализе / В.Б.Бериков, Г.С. Лбов // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». Новосибирск: Институт математики им. С.Л. Соболева СО РАН.- 2008.- 26 с. [Электронный ресурс] Режим доступа: URL https://dl.booksee.org/genesis/369000/166eb0bcf7998f38939c06615eedffe9/_as/[Berikov_V.S._Lbov_G.S.]_Sovremennuee_tendencii_v(BookSee.org).pdf (дата обращения 28.10.2020).

22. Hastie, T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. / T. Hastie, R. Tibshirani, J. Friedman // 2nd ed. – Springer-Verlag. – 2009.

23. Conforti, M. Integer Programming / M. Conforti, G. Cornuéjols, G. Zambelli // SpringerLink. – 2004.

24. Cohen, M.B. Geometric median in nearly linear time/ M.B. Cohen, Y.T. Lee, G. Miller, J. Pachocki, A. Sidford // Proceedings of the forty-eight annual ACM symposium on Theory of Computing (STOC’16). – 2016. – P.9 – 21.

25. Cockayne, E. J. Euclidean constructability in graph minimization problems / E. J.Cockayne, Z. A. Melzak // Mathematics Magazine. – 1969. –№ 42(4). – P. 206 – 208.

26. Youguo, Li. A Clustering Method Based on K-Means Algorithm / L. Youguo, W. Haiyan // Physics Procedia. – 2012. – №25. – P. 1104 – 1109.

27. Ansari, S. A. Using K-Means Clustering to Cluster Provinces in Indonesia / S. A. Ansari, D. Darmawan, R. Robbi, H. Rahmat // Journal of Physics Conference. – 2018. – №1028. – P. 521 – 526.

28. Hossain, Md. A dynamicK-means clustering for data minin / Md. Hossain, Md. Akhtar, A. Nasim, M. Rahman // Indonesian Journal of Electrical Engineeringand Computer Science. – 2019. – № 13 (521). – P. 521 – 526.

29. Perez-Ortega, J. Balancing effort and benefit of Kmeans clustering algorithms in Big Data realms / J. Perez-Ortega, N.N. Almanza-Ortega, D. Romero //. PLoS ONE. – 2018. – № 13 (9), DOI: https://doi.org/10.1371/journal.pone.0201874.

30. Patel, V.R. Modified k-Means Clustering Algorithm / V.R. Patel, R.G. Mehta // In: V.V. Das, N. Thankachan (eds) Computational Intelligence and Information Technology. CIIT2011. Communications in Computer and Information Science. – Springer, Berlin, Heidelberg. – 2011. – P. 250.

31. Na, S. Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm / S. Na, L. Xumin, G. Yong // Third International Symposium on Intelligent Information Technology and Security Informatics. – 2010. – P. 63–67.

32. Steinhaus, H. Sur la divisiondes corps materiels en parties / H. Steinhaus // Bull. Acad. Polon. Sci.–1956.–Cl. III, vol IV.–P.801-804.

33. Weiszfeld, E. Sur le point pour lequel la somme des distances de n points donnes est minimum / E. Weiszfeld // Tohoku Mathematical Journal. – 1937. – P. 355 – 386.

34. Cooper, L. The weber problem revisited / L. Cooper and I. N. Katz // Computers and Mathematics with Applications. – 1981. – № 7(3). – P.225 – 234.

35. Kuhn, H.W. A note on fermat’s problem / H. W. Kuhn. // Mathematical Programming. – 1973. – № 4(1). – P.98 – 107.

36. Lawrence, M. Ostresh. On the convergence of a class of iterative methods for solving the weber location problem / Ostresh Lawrence M. // Operations Research. – 1978. – №26 (4). – P.597 – 609.

37. Plastria, F. and Elosmani M. On the convergence of the weiszfeld algorithm for continuous single facility location allocation problems / F. Plastria and M. Elosmani // TOp. – 2008. – №16 (2). – P.388 – 406.

38. Vardi, Y. The multivariate l1-median and associated data depth / Y. Vardi, Cun-Hui Zhang // Proceedings of the National Academy of Sciences. – 2000. – № 97(4). – P. 1423 – 1426.

39. Badoiu, M. Approximate clustering via core-sets / M. Badoiu, S. Har-Peled, and P. Indyk // In Proceedings on 34th Annual ACM Symposium on Theory of Computing, Montréal, Québec, Canada. – 2002. – P. 250 – 257.

40. Kuhn, H. W. An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics / H. W. Kuhn, R. E. Kuenne // Journal of Regional Science. – 1962. – Вып. 4. – P. 21 – 34.

41. Kernighan, B. W. An efficient heuristic procedure for partitioning graphs / B. W. Kernighan, S. Lin // Bell Syst. Tech. J. – 1970. – V. 49. – P. 291 – 307.

42. Nicholson, T. A. J. A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems / T. A. J. Nicholson // J. Inst. Math. Appl. – 1965. – V. 13. – P. 362-375.

43. Page, E. S. On Monte Carlo methods in congestion problems. I: Searching for an optimum in discrete situations / E.S. Page // Oper. Res. – 1965. – V. 13 (2). – P. 291-299.

44. Hansen, P. Variable neighborhood search: principles and applications / P.Hansen, N.Mladenovic // Eur. J. Oper. Res. – 2001. – Vol. 130. – P. 449 – 467.

45. Hansen, P. Development of variable neighborhood search / P. Hansen, N. Mladenovic // Essays and surveys in metaheuristics. Dordrecht: Kluwer Acad. Publ. – 2002. – P. 415-440.

46. Mladenovic, N. Variable neighborhood search/ N. Mladenovic, P. Hansen// Comput. Oper. Res. – 1997. – Vol. 24. – P. 1097 – 1100.

47. Кочетов, Ю.А. Локальный поиск с чередующимися окрестностями / Ю.А. Кочетов, Н. Младенович, П. Хансен // Дискретн. анализ и исслед. опер. сер.2. – 2003. – Т. 10 № 1. – С. 11 – 43.

48. Brimberg, J. Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem / J. Brimberg, P. Hansen, N. Mladenovic, E. Taillard// Oper. Res. – 2000. – Vol. 48 № 3. – P. 444 – 460.

49. Hansen, P. Variable neighborhood search for weighted maximum satisfiability problem / P. Hansen, B. Jaumard, N. Mladenovic, A. Parreira// Les Cahiers du GERAD G-2000-62. Monreal. Canada. – 2000.

50. Hansen, P. Variable neighborhood decomposition search / P. Hansen, N. Mladenovic, D. Perez-Brito // J. Heuristics. – 2001. – Vol. 7 № 4. – P. 335 – 350.

51. Lopez, F. G. The parallel: variable neighborhood search for the p-median problem / F.G. Lopez, B.M. Batista, J. Moreno-Perez, M. Moreno-Vega // Res. Rep. Univ. of La Laguna, Spain. – 2000.

52. Feo, T. Greedy randomized adaptive search / T. Feo, M. Resende // J. Global Optim. – 1995. – V. 6. – P. 109 – 133.

53. Glover, F. Tabu search- Part I / F. Glover // ORSA Journal on Computing. – 1989. – V.1. – P. 190 – 206.

54. Lloyd, S.P. Least Squares Quantization in PCM / S.P. Lloyd // IEEE Transactions on Information Theory. – 1982. – Vol. 28. – P. 129-137.

55. MacQueen, J.B. Some Methods of Classification and Analysis of Multivariate Observations / J.B.MacQueen // Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability. – 1967. – Vol.1. – P. 281 – 297.

56. Садовский, М. Г. Выявление связи структуры и таксономии геномов хлоропластов методом динамических ядер / М. Г. Садовский, А. И. Чернышова // Фундаментальные исследования. – 2014. – № 11(3). – С. 545 – 549.

57. Садовский, М.Г. О фундаментальной связи геномов митохондрий с геномами организмов-носителей / М.Г. Садовский // Фундаментальные исследования. – 2014. – № 9(4). – С. 781 – 783.

58. Kaufman, L. Clustering by means of Medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods/ L. Kaufman, P.J. Rousseeuw// Springer US. – 1987. – P. 405 – 416.

59. Jain, A. K. Algorithms for Clustering Data / A.K. Jain, R.C. Dubes // Prentice-Hall: New Jersey, USA. – 1988. –P. 96 – 101.

60. Bradley, P. S. Clustering via Concave Minimization in Advances in Neural Information Processing Systems / P.S. Bradley, O.L. Mangasarian, W.N. Street // Mozer, M. C., Jordan, M. I., Petsche, T. Eds.; MIT Press: Cambridge, Massachusetts, USA. – 1997. – vol. 9. – P. 368 – 374.

61. Kaufman, L. Finding groups in data: an introduction to cluster analysis / L. Kaufman, P.J. Rousseeuw // New York:Wiley. – 1990. – P. 368.

62. Ng, R. T. CLARANS: A Method for Clustering Objects for Spatial Data Mining. / R. T. Ng, J. Han //IEEE Transactions on knowledge and data engineering. – 2002. – Vol. 14 № 5. – P.1003-1016.

63. Dunn, J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters / J.C. Dunn // Journal of Cybernetics. – 1973. – V. 3 № . – P. 32. – 57.

64. Zhang, T. BIRCH: An Efficient Data Clustering Method for Very Large Databases / T. Zhang, R. Ramakrishnan, M. Linvy // In Proc. ACM SIGMOD Int. Conf. on Management of Data. ACM Press, New York. – 1996. – P. 103 – 114.

65. Ester, M. A density-based algorithm for discovering clusters in large spatial databases with noise / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). – 1996. – P. 226 – 231.

66. Ankerst, M. OPTICS: ordering points to identify the clustering structure / M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander // In A. Delis, C. Faloutsos, S. Ghandeharizadeh (Eds.), Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. Philadelphia: ACM. – 1999. –Vol. 28 № 2. –P. 49 – 60.

67. Королёв, В.Ю. EM-алгоритм, его модификации и их применение к задаче разделения смесей вероятностных распределений. Теоретический обзор / В.Ю. Королёв // ИПИРАН. М. – 2007. – С. 94.

68. Celeux, G. A classification EM algorithm for clustering and two stochastic versions / G.Celeux, G.Govaert // Computational Statistics and Data Analysis, – 1992. – Vol. 14. – P. 315 – 332.

69. Казаковцев, Л.А. Эвристические алгоритмы разделения смеси распределений: монография / Л.А. Казаковцев, Д.В. Сташков, В.И. Орлов // под общ.ред.В.И.Орлова; СибГУ им.М.Ф.Решетнева. – Красноярск. – 2018. – 164 с.

70. Казаковцев, Л. А. Разработка алгоритмического обеспечения анализа однородности партий электрорадиоизделий для комплектации РЭА КА: монография / Л.А. Казаковцев, И.С. Масич, В.И. Орлов и др. // М-во образования и науки Рос. Федерации, Сиб. гос. аэрокосм. ун-т им. М. Ф. Решетнева. – Красноярск: СибГАУ. – 2016. – 161 с.

71. Kazakovtsev, L. Algorithms with Greedy Heuristic Procedures for Mixture Probability Distribution Separation / L. Kazakovtsev, D. Stashkov, M. Gudyma, V. Kazakovtsev // Yugoslav Journal of Operations Research. – 2019. – Vol.29. – P. 51–67.

72. Drezner, Z. New heuristic algorithms for solving the planar p-median problem / Z. Drezner, J. Brimberg, N. Mladenovic, S. Salhi // Computers & Operations Research. – 2015.– № 62. – P. 296 – 304.

73. Mishra, N. Sublinear time approximate clustering / N. Mishraслова, D. Oblinger, L. Pitt // 12th SODA. – 2001. – P. 439. – 447.

74. Eisenbrand, F. Approximating connected facility location problems via random facility sampling and core detouring / F. Eisenbrand, F. Grandoni, T. Rothvosz, G. Schafer // Proceedings of SODA’2008, New York: ACM. – 2008. – P. 1174 – 1183.

75. Jaiswal, R. A. Simple D2-Sampling Based PTAS for k-Means and Other Clustering Problems / R.A. Jaiswal, A. Kumar, S. Sen // Algorithmica. – 2014. – № 70 (1). – P. 22-46.

76. Avella, P. An Aggregation Heuristic for Large Scale p-median Problem / P. Avella, M. Boccia, S. Salerno, I. Vasilyev // Computers & Operations Research. – 2012. – № 39(7). – P. 1625 – 1632.

77. Francis, R. L. Aggregation error for location models: survey and analysis / R. L. Francis, T.J. Lowe, M.B. Rayco, A. Tamir // Ann. Oper. Res. – 2009. – Т. 167 № 1. – P.171 – 208.

78. Mladenovic, N. The p-median problem: A survey of metaheuristic approaches / N. Mladenovic, J. Brimberg, P. Hansen, J. A. Moreno-Perez // European Journal of Operational Research. – 2007. – № 179. – P. 927 – 939.

79. Arthur, D. k-Means++: The Advantages of Careful Seeding / D. Arthur, S. Vassilvitskii // Proceedings of SODA'07, SIAM. – 2007. – P. 1027 – 1035.

80. Drezner, Z. Solving the planar p-median problem by variable neighborhood and concentric searches / Z. Drezner, J. Brimberg, N. Mladenovic, S. Salhi // J. Glob. Optim. – 2015. – № 63. – P. 501 – 514.

81. Rozhnov, I. P. VNS-Based Algorithms for the Centroid-Based Clustering Problem / I.P. Rozhnov, V.I. Orlov, L.A. Kazakovtsev // Facta Universitatis Series: Mathematics and Informatics. – 2019. – № 34(5). – P. – 957-972.

82. Still, S. Geometric Clustering using the Information Bottleneck method / S. Still, W. Bialek, L. Bottou // Advances In Neural Information Processing Systems 16, Cambridge:MIT Press. – 2004.

83. Sun, Zh. A Parallel Clustering Method Combined Information Bottleneck Theory and Centroid Based Clustering / Zh. Sun, G. Fox, W. Gu, Zh. Li //The Journal of Supercomputing. – 2014. – № 69 (1). – P. 452 – 467.

84. Houck, C. R. Comparison of genetic algorithms, random restart and two-opt switching for solving large location-allocation problems / C. R. Houck, J.A. Joines, M.G. Kay // Computers & Operations Research. – 1996. – № 23(6) . – P. 587. – 596.

85. Maulik, U. Genetic Algorithm-Based Clustering Technique / U. Maulik, S. Bandyopadhyay // Pattern Recognition. – 2000. – № 33 (9). –P. 1455. – 1465.

86. Krishna, K. Genetic K-Means algorithm / K. Krishna, M.M. Murty // IEEE Transactions on Systems. Man. and Cybernetics. Part B (Cybernetics). – 1999. – № 29 (3). – P.433 – 439.

87. Neema, M. N. New Genetic Algorithms Based Approaches to Continuous p-Median Problem / M.N. Neema, K.M. Maniruzzaman, A. Ohgai // Networks and Spatial Economics. – 2011. – № 11. – P. 83. – 99.

88. Kazakovtsev, L. Application of Algorithms with Variable Greedy Heuristics for k-Medoids Problems / L. Kazakovtsev, I. Rozhnov // Informatica. – 2020. – № 44 (1). – P. 55. – 61.

89. Vidyasagar, M. Statistical learning theory and randomized algorithms for control / M. Vidyasagar // IEEE Control Systems. – 1998. – № 12. – P. 69 – 85.

90. Граничин, О. Н. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах / О.Н. Граничин, Б.Т. Поляк // М.: Наука. – 2003. – С. 291.

91. Goldberg, D.E. Genetic algorithm in search, optimization and machine learning / D.E. Goldberg // MA: Addison-Wesley. – 1989. – P. 432.

92. Kohonen, T. Self-Organization and Associative Memory, 3rd ed. / T. Kohonen // Springer information sciences series. New York: Springer-Verlag. – 1989. – P. 312.

93. Kirkpatrick, S. Optimization by simulated annealing / S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi // Science. – 1983. – Vol. 220(4598). – P. 671 – 680.

94. Лбов, Г.С. Выбор эфективной системы зависимых признаков / Г.С. Лбов // Вычислительные системы. – 1965. – Вып. 19. – С. 21 – 34.

95. Holland, J. Adaptation in Natural and Artificial Systems. Ann / J. Holland // Arbor: University of Michigan Press. – 1975. – P. 183.

96. Еремеев, А.В. Генетический алгоритм с турнирной селекцией как метод локального поиска / А.В. Еремеев // Дискретный анализ и исследование операций. – 2012. – Т. 19 № 2. – С. 41 – 53.

97. Akhmedova, Sh.A. New optimization metaheuristic based on co-operation of biology related algorithms / Sh.A. Akhmedova, E.S. Semenkin // Vestnik SibGAU. – 2013. – № 4 (50). – P. 92 – 99.

98. Семенкин, Е.С. Коэволюционный генетический алгоритм решения сложных задач условной оптимизации / Е.С. Семенкин, Р.Б. Сергиенко // Вестник СибГАУ. – 2009. – №2. – С. 17 – 21.

99. Muhlenbein, H. The Equation for Response to Selection and Its Use for Prediction / H. Muhlenbein // Evolutionary Computation. – 1998. – Vol. 5 № 3. – P. 303 – 346.

100. Семенкин, Е.С. Эволюционные методы моделирования и оптимизации сложных систем: Конспект лекций /Е.С. Семѐнкин, М.Н. Жукова, В.Г. Жуков, И.А. Панфилов, В.В. Тынченко. – Красноярск: СФУ. – 2007. – С. 515.

101. Кочетов, Ю.А. Методы локального поиска для дискретных задач размещения: дис. … доктора физ.-мат. Наук: 05.13.18: защищена 19.01.2010. – Новосибирск: Институт математики им.Соболева. – 2010. – С.259.

102. Reeves, C.R. Genetic algorithms for the operations researcher / C.R. Reeves // INFORMS Journal of Computing. – 1997. – Vol. 9 issue 3. – P. 231 – 250.

103. Agarwal, C.C. Optimized crossover for the independent set problem/ C.C.Agarwal, J.B.Orlin, R.P. Tai // Operations research. – 1997. – Vol. 45 issue 2. – P. 226 – 234.

104. Eremeev, A.V. Optimal recombination in genetic algorithms for combinatorial optimization problems, part 1 / A.V. Eremeev, J.V. Kovalenko // Yugoslav Journal of Operations Research. – 2014. – Vol. 24 issue 1. – P. 1 – 20.

105. Kazakovtsev, L. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems/ L.A. Kazakovtsev, A.N. Antamoshkin // Informatica. – 2014. – Vol. 38 № 3. – P. 229 – 240.

106. Казаковцев, Л.А. Метод жадных эвристик для систем автоматической группировки объектов: дис... докт. техн. наук/ Л.А. Казаковцев// Красноярск. – 2016. – 429 с.

107. Семенкин, Е.С. Метод обобщенного адаптивного поиска для оптимизации управления космическими аппаратами: дис... д-ра техн. наук / Е.С. Семенкин // САА. Красноярск. – 1997. – 400 с.

108. Семенкин, Е.С. Об эволюционных алгоритмах решения сложных задач оптимизации/ Е.С.Семенкин, А.В.Гуменникова, М.Н.Емельянова, Е.А.Сопов// Вестн. Сиб. гос. аэрокосмич. ун-та им.акад. М.Ф.Решетнева: сб. науч. тр./ под ред. проф. Г.П.Белякова Сиб. гос. аэрокосмич. ун-т. вып.5. Красноярск. – 2003.-С.14–23.

109. Сошникова, Л.А. Многомерный статистический анализ в экономике: Учебное пособие для вузов / Л.А. Сошникова, В.Н. Тамашевич, Г. Усбе, М. Шефер; Подред. В.Н. Тамашевича. - М.: БНИТИ – Дана. – 1999. – 598 с.

110. Хачумов, М.В. Расстояния, метрики и кластерный анализ / М.В. Хачумов// Искусственный интеллект и принятие решений. – 2012. – №1. – С.81 – 89.

111. Деза, Е. Глава 19. Расстояния на действительной и цифровой плоскостях. 19.1. Метрики на действительной плоскости / Е. Деза, М. Мари Деза // Энциклопедический словарь расстояний. М: Наука. – 2008. – С. 276.

112. Крутиков, В.Н. Анализ данных: учебное пособие [Электронный ресурс]/ В.Н. Крутиков, В.В. Мешечкин. // Кемеровский государственный университет. – Кемерово: Кемеровский государственный университет. – 2014. – 138 с.: ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=278426 (дата обращения: 27.09.2020). – Библиогр. в кн. – ISBN 978-5-8353-1770-7. – Текст: электронный.

113. Federal standard 1037C (Telecommunications: Glossary of Telecommunication Terms) [Электронный ресурс] Режим доступа: URL https://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm (дата обращения 31.10.2020).

114. McLachlan, G.J. Mahalanobis Distance / McLachlan G.J. // Resonance. – 1999 – № 4. – P. 20-26.

115. Антамошкин, А.Н. Алгоритм для задачи размещения с метрикой МОСКВЫ-КАРЛСРУЭ / А.Н. Антамошкин, Л.А. Казаковцев // Системы управления и информационные технологии. – 2012. – № 3-1 (49). – С. 111 – 115.

116. Соколов, Е. Семинары по метрическим методам классификации /Е. Соколов // [Электронный ресурс]. – 2013. – Режим доступа URL: http://www.machinelearning.ru/wiki/images/9/9a/Sem1_knn.pdf (дата обращения 30.10.2020)

117. Сивоголовко, Е.В. Методы оценки качества четкой кластеризации / Е.В. Сивоголовко // Компьютерные инструменты в образовании. – 2011. – № 4. – С. 14 – 31.

118. Calinski, R.B. A dendrite method for cluster analysis / R.B. Calinski, J. Harabasz. // Comm. in Statistics. – 1974. – Volume 3 Issue 1. – P. 1 – 27.

119. Dunn, J.C. Well separated clusters and optimal fuzzy-partitions / J.C. Dunn. // Journal of Cybernetics. – 1974. – № 4. – P. 95 – 104.

120. Davies, D.L. A cluster separation measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysisand Machine Intelligence. – 1979. – Vol. 1 № 2.

121. Chou, C. H. A new cluster validity measure and its application to image compression / C.H. Chou, M.C. Su, E. Lai. // Pattern Analysis and Applications. – 2004. – №7. P. 205 – 220.

122. Chou, C.H. Symmetry as a new measure for cluster validity / C.H. Chou, M.C. Su, L. Eugene // 2nd WSEASInternational Conference of scientific on Computation and Soft Computing. – 2002. – P. 209 – 213.

123. Krzanowski, W. A criterion for determining the number of group s in a dataset using sum of squares clustering / W. Krzanowski, Y. Lai // Biometrics. – 1985. – № 44. – P.23-34

124. Halkidi, M. Quality scheme assessment in the clustering process / M. Halkidi, M. Vazirgiannis, Y. Batistakis // In: Zighed D.A., Komorowski J., Żytkow J. (eds) Principles of Data Mining and Knowledge Discovery. PKDD 2000. Lecture Notes in Computer Science. – 2000. – Vol. 1910.

125. Halkidi, M. Clustering validity assessment: Finding the optimal partitioning of a data set / M. Halkidi, M. Vazirgiannis. // Proceedings 2001 IEEE International Conference on Data Mining, San Jose, CA, USA. – 2001. – P. 187 – 194.

126. Sharma, S. Applied multivariate techniques / S. Sharma // John Wiley and Sons, Inc. – 1996. – P. 512.

127. Rousseeuw, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis / P. Rousseeuw // Journal of Computational and Applied Mathematics. – 1987. – Vol. 20. – P. 53 – 65.

128. Golovanov, S.M. Recursive clustering algorithm based on silhouette criterion maximization for sorting semiconductor devices by homogeneous batches / S.M. Golovanov, V.I. Orlov, L.A. Kazakovtsev // IOP Conf. Series: Materials Science and Engineering. – 2019. – № 537. DOI: doi:10.1088/1757-899X/537/2/022035.

129. Rand, W.M. Objective Criteria for the Evaluation of Clustering Methods / W.M. Rand // Journal of the American Statistical Association. – 1971. – Vol. 66 № 336. – P. 846 – 850.

130. Hubert, L. Comparing partitions / L. Hubert, P. Arabie // Journal of classification. . – 1985. – №2(1). – P.193–218.

131. D’Ambrosio, A. Adjusted Concordance Index: an Extensionl of the Adjusted Rand Index to Fuzzy Partitions / A. D’Ambrosio, S. Amodio, C. Iorio et al. // Journal of Classification Classif. – 2020 DOI: https://doi.org/10.1007/s00357-020-09367-0

132. Theodoridis, S. Pattern recognition / S. Theodoridis, K. Koutroumbas. // Academic Press, San Diego, CA. – 1999. –P. 711.

133. Bradley, A. The use of the area under the ROC curve in the evaluation of machine learning algorithms / A. Bradley // Pattern Recognition. – 1997, – № 3. –P. 1145–1159.

134. Логистическая регрессия и ROC-анализ математический аппарат / компания Loginom Company [Электронный ресурс] Режим доступа: https://loginom.ru/blog/logistic-regression-roc-auc (дата обращения 30.10.2020)

135. Федосов, В.В. Минимально необходимый объем испытаний изделий микроэлектроники на этапе входного контроля / В.В. Федосов, В.И. Орлов // Известия высшихучебных заведений. Приборостроение. – 2011. – Т.54 № 4. – С.58 – 62.

136. Федосов, В.В. Технический отчет. Космический аппарат «SESAT» со сроком активного функционирования 10 лет. Принципы, методы и результаты комплектации аппаратуры электрорадиоизделиями / В.В.Федосов, В.И. Куклин, В.И. Орлов, Ш.Н. Исляев и др .// ФГУП «НПО ПМ им. академика Решетнева». – 1999. – С.408.

137. Федосов, В.В. Вопросы обеспечения работоспособности электронной компонентной базы в аппаратуре космических аппаратов: учеб. пособие / В.В.Федосов // Сиб. гос. аэрокосмич. Ун-т. – Красноярск. – 2015. – С. 68.

138. Орлов, В.И. Усовершенствованный метод формирования производственных партий электронной компонентной базы с особыми требованиями качества / В.И.Орлов, Д.В. Сташков, Л.А. Казаковцев, И.П. Рожнов, И.Р. Насыров, О.Б. Казаковцева // Современные наукоемкие технологии. – 2018. – № 1. – С. 37 – 42.

139. Сташков, Д.В. Алгоритм для серии задач разделения смеси распределений / Д.В.Сташков, М.Н.Гудыма, Л.А.Казаковцев, И.П.Рожнов, В.И. Орлов // Решетневские чтения. – 2017. – Т. 1. № 21. – С. 327 – 328.

140. Рожнов, И.П. Усовершенствованный алгоритм разделения смеси распределений для данных большой размерности / Л.А. Казаковцев, Д.В. Сташков, О.Б. Казаковцева, И.П. Рожнов, А.В. Медведев// Лесной и химический комплексы -проблемы и решения. (7 декабря 2017). Красноярск. – 2017. – С. 502 – 505.

141. Орлов, В.И. Выявление однородных партий изделий космической радиоэлектроники на основе разделения смеси сферических гауссовых распределений / В. И. Орлов, Д. В. Сташков, Л. А. Казаковцев, И. Р. Насыров, А. Н. Антамошкин // Вестник СибГАУ. – 2017. – Том 18 № 1. – С. 69 – 77.

142. Сташков, Д. В. Применение ЕМ-алгоритма со сферическим гауссовым распределением к задаче классификации промышленной продукции /Д.В. Сташков, В.И. Орлов, И.Р. Насыров, Л.А. Казаковцев // Экономика и менеджмент систем управления. – 2017. – №1-1 (23). –.С.185 – 193.

143. Мазуров, В. Д. Факторный анализ и смысл факторов как функция смыслов признаков / В. Д. Мазуров // XI Международная конференция «Российские регионы в фокусе перемен». Екатеринбург, 17-19 ноября 2016 г.: сборник докладов. – Екатеринбург: Издательство УМЦ УПИ, 2016. – Ч. 1. – С. 563 – 569.

144. Кадневский, В. М. Ф. Гальтон - учёный-энциклопедист, один из создателей научного метода тестов (к 190-летию со дня рождения) / В.М. Кадневский В.М., В.В. Лемиш // Вестник Омского университета. – Омск: Издательство Омский государственный университет им. М.Ф. Достоевского. – 2012. – № 4. – С.276 – 280.

145. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space / K. Pearson // Philosophical Magazine Series 6. – 1901. – Т 2 (11). – P.559 – 572.

146. Иберла, К. Факторный анализ / К. Иберла // Пер. с нем. В.М. Ивановой; предисл. А.М. Дуброва. – М.: Статистика. – 1980. – С.398 (Uberla, K. Factorenanalyse / K.Uberla // Berlin: Springer-Verlag. –1977. – P.399.)

147. Thurstone, L.L. Multiple factor analysis / L.L. Thurstone // Chicago. – 1961. – P. 250.

148. Елисеев, О. П. Экспериментальная психология личности: учебное пособие для бакалавриата и магистратуры / О. П. Елисеев // Москва: Юрайт. – 2017. – 389 с.

149. Bargmann, R. Signifikanz untersuchungen der einfachenStruktur in der Faktoren-Analyse / R. Bargmann // Mitteilungsblatt für Mathematische Statistik. – 1955. –vol. 1. – P. 1-24.

150. Машин, В.А. Методические вопросы использования факторного анализа на примере спектральных показателей сердечного ритма / В.А. Машин // Экспериментальная психология. – 2010. – № 4. – С. 119 – 138.

151. Цугунян, А.М. Применение элементов факторного анализа при анализе эффективности использования трудовых ресурсов организации / А.М. Цугунян, А.С. Ломжинская // Проблемы современной науки. – 2016. – № 16. – С. 88–96.

152. Тороп, Е.А. Корреляционный и факторный анализы признаков урожайности озимой ржи / Е.А. Тороп, А.А Тороп, В.В. Чайкин // Современное экологическое состояние природной среды и научно-практические аспекты рационального природользования. I Международная научно-практическая Интернет-конференция, посвященная 25-летию ФГБНУ «Прикаспийский научно-исследовательский институт аридного земледелия». – 2016. – С. 2730 – 2740.

153. Бугримов, В.А. Использование факторного анализа для оптимизации размера заказываемых партий запасных частей станцией техобслуживания / В.А. Бугримов, А.В. Кондратьев, В.И. Сарбаев, В.В. Бородулин // Научное обозрение. – 2017. – № 2. – С.65 – 71.

154. Лушкин, А.М. Методология вероятностного оценивания текущего уровня аварийности по результатам факторного анализа авиационных событий / А.М. Лушкин // Научный вестник МГТУ ГА. – 2015. – №218. – С.25 – 28.

155. Орлов, А.И. Разработка системы прогнозирования уровня безопасности полетов и поддержки принятия решений на основе факторного анализа показателей / А.И. Орлов, В.Д. Шаров // Проблемы управления безопасностью сложных систем. Труды XXI Международной конференции. – 2013 – С. 360 – 363.

156. Хайдаров, Р.А. Алгоритм факторного анализа показателей безопасности полетов от причин-факторов / Р.А. Хайдаров // Научный вестник МГТУ ГА. – 2011. – № 174. – С. 96 – 99.

157. Harman, H. Modern factor analysis / H. Harmann // The university of Chicago press, Chicago. – 1967. – P. 474

158. Митина, О.В., Михайловская И.Б.: Факторный анализ для психологов / О.В. Митина, И.Б. Михайловская // М.: Учебно-методический коллектор «Психология». – 2001. – С. 169.

159. Rencher, A. C. Methods of multivariate analysis / A.C. Rencher // Wiley. – 2002. – P. 739.

160. Ким, Дж.-О. Факторный, дискриминантный и кластерный анализ / Дж.-О. Ким, Ч. У. Мьюллер, У. Р. Клекка, М. С. Олдендерфер, Р. К. Блэшфилд // М.: Финансы и статистика. – 1989. – С.215.

161. Открытая медицинская библиотека: [Электронный ресурс]. URL: http://medic.oplib.ru/random/view/63214/ (Дата обращения 30.10.2020)

162. Kaiser, H.F. Sample and population score matrices and sample correlation matrices from an arbitrary population correlation matrix / H.F. Kaiser, K. Dickman // Psychometrika. – 1962. – Vol.27, issue 2. – P.179-181.

163. Cattel, R.B. The scree test the number of factors / R.B. Cattel //Multivar. Behave. Res. – 1966. – Vol. 1. – P.245-276.

164. Лоули, Д. Факторный анализ как статистический метод / Д. Лоули, А. Максвелл // Перевод с английского Ю.Н. Благовещенского. Мир, Москва. – 1967. – С. 145 (Lawley, D., Maxwell, A.: Factor analysis as a statistical method / D. Lawley, A. Maxwell // Butterworths, London. – 1963. – P. 153).

165. Ackermann, M. R. StreamKM++: A Clustering Algorithm for Data Streams / M.R. Ackermann, Ch. Lammersen, M. Märtens, Ch. Raupach // Journal of Experimental Algorithmics. – 2010. – № 17 (1). – P. 173 – 187.

166. Kanungo, T. Computing nearest neighbors for moving points and applications to clustering / T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu // Proc. of the tenth annual ACM-SIAM symp. on Discrete algorithms (Society for Industrial and Applied Mathematics). – 1999. – P. 931 – 932.

167. Kazakovtsev, L A, Stupina A A and Orlov V I 2014 Modification of the genetic algorithm with greedy heuristic for continuous location and classification problems / L.A. Kazakovtsev, A.A. Stupina, V.I. Orlov // Sistemy upravleniya iinformatsionnye tekhnologii. – 2014.– №2(56). – P. 31–34.

168. Rozhnov, I. Ensembles of clustering algorithms for problem of detection of homogeneous production batches of semiconductor devices/ I.Rozhnov, V.Orlov, L.Kazakovtsev// В сборнике: CEUR Workshop Proceedings Сер. OPTA-SCL 2018 - Proceedings of the School-Seminar on Optimization Problems and their Applications.CEUR-WS. –2018. –Vol. 2098. –P. 338–348.

169. Казаковцев, Л.А. Метод жадных эвристик для задач размещения / Л.А.Казаковцев, А.Н. Антамошкин // Вестник СибГАУ.– 2015.– №2.– С.317 – 325.

170. Орлов, В.И. О непараметрической диагностике и управлении процессом изготовления электрорадиоизделий / В.И.Орлов, Н.А. Сергеева // Вестник СибГАУ. – 2013. – Вып. 2(48). – С. 70 – 75.

171. Орлов, В.И. Алгоритм поиска в чередующихся окрестностях для задачи выделения однородных производственных партий электрорадиоизделий / В.И. Орлов, И.П. Рожнов, В.Л. Казаковцев, М.Н. Гудыма // Решетневские чтения. Красноярск. – 2018. – Т. 1 № 22. –С. 315 – 316.

172. Сташков, Д.В. Моделирование сборной партии электрорадиоизделий в виде смеси вероятностных распределений / Д.В. Сташков Д.В, И.Р. Насыров, А.М. Попов, В.И. Орлов // Экономика и менеджмент систем управления. – 2017. – № 2.2 (24) – С. 282 – 289.

173. Orlov, V.I., Fedosov, V.V.: ERC clustering dataset (2016) http://levk.info/Data1526 7parts.csv

174. Shkaberina, G. Sh. Correlation between the time taken to master the competency and the rank of competence evaluated on the basic educational programme G. Shkaberina, L. Baranovskaya // Journal of Economics and Social Sciences. — 2016. – № 8. – P. 3.

175. Шкаберина, Г.Ш. Факторный анализ с использованием матрицы Спирмена в задаче автоматической группировки электрорадиоизделий по производственным партиям / Г.Ш. Шкаберина, В.И. Орлов, Е.М. Товбис, Л.А. Казаковцев // Системы управления и информационные технологии. – 2019.– №1(75). – С.91-96.

176. Deductor. Руководство аналитика. Версия 5.3/ Компания BaseGroupTM Labs, 1995-2013. – С. 219 [Электронный ресурс] Режим доступа URL: https://basegroup.ru/deductor/manual/guide-analyst-530 (дата обращения 29.10.2020)

177. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps / T. Kohonen // Biological Cybernetics. – 1980. – № 43 (1). – С. 59 – 69.

178. Шкаберина, Г.Ш. Применение алгоритмов кластеризации с особыми мерами расстояния для задачи автоматической группировки электрорадиоизделий / В.И. Орлов В.И, Г.Ш. Шкаберина, И.П. Рожнов, А.А. Ступина, Л.А. Казаковцев// Системы управления и информационные технологии.– 2019.– № 3(77). – С. 42 – 46.

179. Shkaberina, G. Sh. Detection of homogeneous production batches of semiconductor devices by greedy heuristic clustering algorithms with special distance metrics / G. Sh.Shkaberina, I. P.Rozhnov, V. P.Popov, L.A.Kazakovtsev,E. V. Lapunova // IOP Conference Series: Materials Science and Engineering 2019. – 2020.–. Vol. 734. –Article ID 012104. – 5 P. DOI: 10.1088/1757-899X/734/1/012104.

180. Cheung Y.M. k-Means: A new generalized k-means clustering algorithm / Y.M. Cheung // Pattern Recognition Letters. – 2003. – Vol. 24, Issue 15. – P.2883 – 2893.

181. Рожнов, И.П. Алгоритм для задачи к-средних с рандомизированными чередующимися окрестностями / И.П. Рожнов, Л.А. Казаковцев, М.Н. Гудыма, В.Л. Казаковцев // Системы управления и информационные технологии. – 2018. – № 3 (73). – С. 46 – 51.

182. Казаковцев, Л. А. Выбор метрики для системы автоматической классификации электрорадиоизделий по производственным партиям / Л.А. Казаковцев, В.И. Орлов, А.А. Ступина // Программные продукты и системы. – 2015. – № 2. – С. 124 – 129.

183. Айвазян, С. А. Прикладная статистика: классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин // М.: Финансы и статистика. – 1989. – С. 607.

184. Шкаберина, Г.Ш. Оптимизационная модель для автоматической группировки многомерных данных / Г.Ш. Шкаберина // Системы управления и информационные технологии. – 2020. – № 3(81). – С. 94 – 99.

185. Shkaberina, G. Sh. Efficiency of distance measures in the automatic grouping of electronic radio devices by k-means algorithm / G. Sh.Shkaberina, E. M.Tovbis, L. A.Kazakovtsev, A. O. Shemyakov, A. P.Romanov, N. V.Lukonin // IOP Conference Series: Materials Science and Engineering2019.– 2020.– Vol. 734. – Article ID012136.– 5P. DOI: 10.1088/1757-899X/734/1/012136.

186. Shkaberina, G. Sh. On optimization models for automatic grouping of industrial products by homogeneous production batches / G. Sh.Shkaberina, V. I.Orlov, E. M.Tovbis, L.A.Kazakovtsev // In: Kochetov Y., Bykadorov I., Gruzdeva T. (eds) Mathematical Optimization Theory and Operations Research. MOTOR 2020. Communications in Computer and Information Science.–2020. –Vol. 1275.– P.421 – 436. DOI: 10.1007/978-3-030-58657-7_33.

187. Шумская, А. О. Оценка эффективности метрик расстояния Евклида и расстояния Махаланобиса в задачах идентификации происхождения текста / А.О. Шумская // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2013. – № 3(29). – С.141 – 145.

188. Lapidot, I. Convergence problems of Mahalanobis distance-based k-means clustering / I. Lapidot // IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE). – 2018. – P. 1 – 5.

189. Shkaberina, G.S. Orlov V.I., Tovbis E.M., Kazakovtsev L.A. Identification of the Optimal Set of Informative Features for the Problem of Separating of Mixed Production Batch of Semiconductor Devices for the Space Industry / G.S. Shkaberina, V.I. Orlov, E.M. Tovbis, L.A. Kazakovtsev // Mathematical Optimization Theory and Operations Research. MOTOR 2019. Communications in Computer and Information Science. – 2019. – Vol. 1090. – P. 408-421.

190. Shkaberina,G.Sh. Estimation of the impact of semiconductor device parameters on the accuracy of separating a mixed production batch/ G.Sh.Shkaberina,

V.I.Orlov, E.M.Tovbis, E.V.Sugak, L.A.Kazakovtsev // IOP Conf. Series: MIP: Engineering.-2019.-Vol. 537.

191. Hansen, P, Mladenovic N. J-means: a new local search heuristic for minimum sum of squares clustering / P. Hansen, N. Mladenovic // Pattern recognition. – 2001. – Vol.34, Issue 2. – P. 405-413.

192. Граничин, О.Н. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах/ О.Н.Граничин, Б.Т.Поляк// М.Наука. -2003.-С. 291.

193. Vidyasagar, M. Statistical learning theory and randomized algorithms for control/ M. Vidyasagar// IEEE Control Systems. -1998.-No. 12.-P.69-85.

194. Sheng, W. A genetic k-medoids clustering algorithm/ W.Sheng, X.Liu// Journal of Heuristics.-2006.-Vol. 12,-No. 6.-P. 447-466.

195. Zeebaree, D.Q. Combination of K-means clustering with Genetic Algorithm: A review / D.Q. Zeebaree, H. Haron, A.M. Abdulazeez, S.R.M. Zeebaree // International Journal of Applied Engineering Research. – 2017. – Vol.12, №24. – P.14238–14245.

196. Hruschka, E. A Survey of Evolutionary Algorithms for Clustering / E. Hruschka, R. Campello, A. Freitas, A. de Carvalho // IEEE Transactions on. Systems, Man, and Cybernetics, Part C: Applications and Reviews. – 2009. – 39 (2). – P. 133 – 155.

197. Freitas, A. A. A Review of Evolutionary Algorithms for Data Mining / A.A. Freitas // Data Mining and Knowledge Discovery Handbook. – 2009. – P. 371–400.

198. Bandyopadhyay, S. Genetic Algorithms for Clustering and Fuzzy Clustering / S. Bandyopadhyay // Wires Data Min. Knowl. – 2011. – № 1. – P. 524 – 531.

199. Глава 9. Генетический алгоритм [Электронный ресурс] Режим доступа: URL http://www.qai.narod.ru/Publications/tsoy_chapterGA.pdf (дата обращения 30.10.20)

200. Herrera, F. Hybrid Crossover Operators for Real-Coded Genetic Algorithms: An Experimental Study / F.Herrera, M.Losano, A.M.Sanches // Soft Computing. – 2005. – № 9. – P. 280 – 298.

201. Hosage, C. M. Discrete Space Location-Allocation Solutions from Genetic Algorithms / C. M. Hosage, M.F. Goodchild // Annals of Operations Research. – 1986. – №6. –P. 35 – 46.

202. Chiou, Y. Genetic clustering algorithms/ Y.Chiou, L.W.Lan// European Journal of Operational Research. – 2001. – Vol. 135. – P. 413-427.

203. Bozkaya, B. A. Genetic Algorithm for the p-Median Problem/ B.A. Bozkaya, J. Zhang, E. Erkut // Facility Location: Applications and Theory / Z.Drezner, H.Hamacher [eds.].-New York: Springer. – 2002. – P. 179 – 205.

204. Alp, O. An Efficient Genetic Algorithm for the p-Median Problem / O. Alp, E.Erkut, Z. Drezner // Annals of Operations Research. – 2003. – 122 (1-4). – P. 21 – 42.

205. Kim, K. A recommender system using GA K-means clustering in an online shopping market / K. Kim, H.Ahn // Expert Systems with Applications. – 2008. – №34. – P. 1200 – 1209

206. Kwedlo, W. Using Genetic Algorithm for Selection of Initial Cluster Centers for the K-Means Method / W. Kwedlo, P. Iwanowicz // ICAISC 2010: Artificial Intelligence and Soft Computing. –2010. – P. 165 – 172.

207. He, Zh. Clustering stability-based Evolutionary K-Means / Zh. He, Ch. Yu // Soft Computing. – 2019. – № 23(1). – P. 305 – 321.

208. Pizzuti, C. A K-means Based Genetic Algorithm for Data Clustering / C. Pizzuti, N. Procopio // International Joint Conference SOCO16-CISIS16-ICEUTE16. – 2016. – № 527. – P. 211 – 222.

209. Davies, D.L. A cluster separation measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysis and Machine Intelligence. –1979. – №1(2). – P. 224 – 227.

210. Pelleg, D. X-means: Extending K-means with Efficient Estimation of the Number of Clusters / D. Pelleg, A. Moore // In Proceedings of the 17th International conference on Machine Learning. – 2000. – P. 727 – 734.

211. Eremeev, A.V. Genetic algorithm with tournament selection as a local search method / A.V. Eremeev // Discrete analysis and operations research. – 2012. – Vol.19. – № 2. – P. 41 – 53.

212. Fogel, D. B. Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems / D.B. Fogel, J. Atmar // Biol. Cybern. –1990. – № 63. – P. 111 – 114.

213. Liu, C. On designing genetic algorithms for solving small - and medium - scale traveling salesman problems / C. Liu, A. Kroll // EC 2012, SIDE 2012. Lecture Notes in Computer Science. – 2012. – Vol.7269. –P. 283 – 291.

214. Osaba, E. Crossover versus mutation: a comparative analysis of the evolutionary strategy of genetic algorithms applied to combinatorial optimization problems / E. Osaba, R. Carballedo, F. Diaz, E. Onieva, I. de la Iglesia, A. Perallos // The Scientific World Journal. – 2014. – Vol.2014. – P.22.

215. Walkenhorst, J. Multikriterielle optimierungsverfahren fur pickup-anddelivery-probleme / J. Walkenhorst, T. Bertram // In:Proceedings of 21. Workshop computational intelligence, Dortmund, Germany. – 2011. – P. 61 – 76.

216. Kazakovtsev, L.A. Greedy Heuristic Method for Location Problems / L.A. Kazakovtsev, A.N. Antamoshkin // Vestnik SibGAU. – 2015. – Vol.16, №2. – P. 317 – 325.

217. Larranaga, P. Genetic algorithms for the travelling salesman problem: a review of representations and operators/ P.Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, S. Dizdarevic // Artif. Intell. Rev. – 1999. – 13(2). – P. 129 – 170.

218. Sarangi, A. Design of linear phase FIR high pass filter using PSO with gaussian mutation / A. Sarangi, R. Lenka, S.K. Sarangi // In: Swarm, Evolutionalry, and Memetic Computing, SEMCCO 2014. – 2015. – Vol. 8947. – P.471 – 479.

219. Deb, D. Investigation of mutation schemes in real-parameter genetic algorithms / D. Ded, K.Ded// LNCS. – 2012. – № 7677. – P. 1 – 8.

220. Deep, K. A new mutation operator for real coded genetic algorithms / K. Deep, M. Thakur// Appl. Math. Comput. – 2007. – № 193(1). – P. 211 – 230.

221. Deep, K. Combined mutation operators of genetic algorithm for the travelling salesman problem / K. Deep, H. Mebrahtu // Int. J. Combin. Opt. Probl. Inf. – 2011. – № 2(3). – P. 1 – 23

222. Hong, T P. Simultaneously applying multiple mutation operators in genetic algorithms / T.P. Hong, H. S. Wang, W.C. Chen // J. Heuristics. – 2000. – №6(4). – P. 439 – 455,

223. McGinley, B. Maintaining healthy population diversity using adaptive crossover, mutation, and selection / B. McGinley, J. Maher, C. O'Riordan, F. Morgan // IEEE Trans. Evol. Comput. – 2011. – № 15(5). – P. 692 – 714.

224. Serpell, M. Self-adaptation of mutation operator and probability for permutation representations in genetic algorithms / M. Serpell, J.E. Smith //Evol. Comput. – 2010. – № 18(3). – P.491 – 514.

225. Brizuela, C. A. Experimental genetic operators analysis for the multiobjective permutation flowshop / C.A. Brizuela, R. Aceves// LNCS. – 2003. – № 2632. – P.578 – 592,

226. Wang, L, Zhang L (2006) Determining optimal combination of genetic operators for flow shop scheduling / L. Wang, L. Zhang// Int. J. Adv. Manuf. Technol. – 2006. – № 30(3-4). – P. 302 – 308.

227. Hasan, B. H. F. Evaluating the effectiveness of mutation operators on the behavior of genetic algorithms applied to non-deterministic polynomial problems / B. H. F. Hasan, M. S. M. Saleh //Informatica. – 2011. – №35(4). – P.513 – 518

228. Karthikeyan, P. Improved genetic algorithm using different genetic operator combinations (GOCs) for multicast routing in ad hoc networks / P. Karthikeyan, S. Baskar, A. Alphones // Soft. Comput. – 2013. – №17(9). – P. 1563 – 1572.

229. Correa, E. S. A Genetic Algorithm for the P-median Problem / E.S. Correa, M.T.A. Steiner, A.A. Freitas, C. Carnieri // Proc. GECCO-2001. – 2001. – P. 1268 – 1275.

230. Alkhalifah, Y. A genetic algorithm applied to graph problems involving subsets of vertices / Y. Alkhalifah, R. L. Wainwright // Proceedings of the 2004 Congress on Evolutionary Computation. – 2004. – № 1. – P. 303 – 308,

231. Lu, Y. FGKA: A fast genetic K-means clustering algorithm / Y. Lu, S. Lu, F. Fotouhi, Y. Deng, S.J. Brown// Proceedings of the 2004 ACM Symposium on Applied Computing (SAC). –2004. Режим доступа DOI: 10.1145/967900.968029

232. Cheng, S. S. A Prototypes-Embedded Genetic K-means Algorithm / S.S. Cheng, Y. H. Chao, H.M. Wang, H. C. Fu// 18th International Conference on Pattern Recognition (ICPR'06). – 2006. – P. 724 – 727.

233. Brimberg, J. A New Local Search for Continuous Location Problems / J. Brimberg, Z. Drezner, N. Mladenovic, S. Salhi// European Journal of Operational Research. – 2014. – № 232(2). – P. 256-265

234. Shkaberina, G. Genetic Algorithms with the Crossover-Like Mutation Operator for the k-Means Problem. / L. Kazakovtsev, G. Shkaberina, I. Rozhnov, R. Li,V. Kazakovtsev // In: Kochetov Y., Bykadorov I., Gruzdeva T. (eds) Mathematical Optimization Theory and Operations Research. MOTOR 2020. Communications in Computer and Information Science. –2020. –Vol. 1275.– P. 350-362.

235. Шкаберина, Г.Ш. Модели и алгоритмы автоматической группировки объектов на основе модели k- средних / Г.Ш. Шкаберина, Л.А. Казаковцев, Ж. Ли // Сибирский журнал науки и технологий. – 2020. – Т. 21, № 2. – С. 1 –11.

236. Clustering basic benchmark [http://cs.joensuu.fi/sipu/datasets] (обращение 30.10.2020)

237. Machine Learning Repository [https://archive.ics.uci.edu/ml/index.php] (обращение 30.10.2020)

238. Mann, H. B., Whitney D. R. On a test of whether one of two random variables is stochastically larger than the other. / H.B. Mann, D.R. Whitney// Annals of Mathematical Statistics. – 1947. – №18. – P. 50– 60.

239. Student. The probable error of a mean. / Student // Biometrika. –1908. – № 6 (1). –P. 1-25.

240. Nvidia Developer.https://developer.nvidia.com/cuda-zone.

241. Zechner, M. Accelerating K-Means on the Graphics Processor via CUDA / M. Zechner, M. Granitzer// 2009 First International Conference on Intensive Applications and Services. –2009. –P. 7-15. DOI: 10.1109/INTENSIVE.2009.19.

242. Орлов, В.И. Программный комплекс решения задач автоматической группировки объектов с применением массивно-параллельных систем / В.И. Орлов, Л.А. Казаковцев, И.П. Рожнов, Г.Ш. Шкаберина, И.С.Масич // М.: РОСПАТЕНТ.- 2020. Свидетельство о государственной регистрации программы для ЭВМ № 2020663109 от 22.10.2020.

243. Luebke, D. How gpus work/ D. Luebke, G. Humphreys// Computer. –2007. № 40(2). –P.96 – 100

244. Dangeti, P. Statistics for Machine Learning / P. Dangeti // Birmingham, Packt Publishing. –2017. – P. 438.

245. Dudoit, S. Comparison of Discrimination methods for the classification of tumors using gene expression data / S. Dudoit, J. Fridlyand, T. Speed // Journal of the American Statistical Association. – 2002. –Vol.97 №457. – P.77-87.

246. McLachlan, G. J. Discriminant Analysis and Statistical Pattern Recognition / G.J. McLachlan // New York: Wiley. – 1992. –P. 526.

247. Zhang, H. Naive Bayesian Classifiers for Ranking / H. Zhang, J. Su // In: Boulicaut JF., Esposito F., Giannotti F., Pedreschi D. (eds) Machine Learning: ECML 2004. ECML 2004. Lecture Notes in Computer Science. – 2004. – Vol. 3201 DOI:10.1007/978-3-540-30115-8_46

248. Boser, B. E. A training algorithm for optimal margin classifiers / B. E. Boser, I. M.Guyon, V. N. Vapnik // In COLT ’92: Proceedings of the fifth annual workshop on Computational learning theory. – 1992. –P. 144–152.

249. Brownlee, J. Master Machine Learning Algorithms / J. Brownlee//Edition, v1.1.– 2016. – P.162.

250. Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems / R.A. Fisher // Annals of Eugenics. – 1936. – Vol.7. – P.179-188.

251. Hauck, T. Scikit-learn Cookbook / T. Hauck// Birmingham: Packt Publishing. – 2014. –P.214.

252. Bartlett, P. Generalization performance of support vector machines and other pattern classifiers / P. Barlett, J. Shawe-Taylor // Advances in Kernel Methods. MIT Press, Cambridge, USA. – 1998.

253. Herbrich, R., Graepel, T., Campbell, C. Bayes Point Machines / R. Herbrich, T. Graepel, C. Campbell // Journal of Machine Learning Research. –2001. – Vol.1. –P. 245-279.

254. Kim, P. Matlab Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence / P. Kim // Seoul: Apress. – 2017. –P.151.

255. Neapolitan, R., Jiang, X. Artificial Intelligence with an Introduction to Machine Learning/ R. Neapolitan, X. Jiang// Chapman and Hall/CRC Press. – 2018. – P.480.

256. Schneider, P., Hammer, B., Biehl, M. Adaptive relevance matrices in learning vector quantization / P. Schneider, B. Hammer, M. Biehl // Neural Computation. –2009. – Vol. (10). – P. 3532 – 3561.

257. McCalloch, W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity / W.S. McCalloch// Bull. Math. Biophys. – 1943. – v.5. – P.115–133.

258. Винер, Н. Кибернетика / Н. Винер// 2-е изд. –1961.

259. Hebb, D. O.. The organization of behavior. A neuropsychological theory / D. O. Hebb // New York, Wiley. –1949.– P.305-319.

260. Rosenblatt, F. The Perceptron: A probabilistic model for information storage and organization in the brain / F. Rosenblatt //Psychological Review. – 1958. – 65(6). –P. 386-408.

261. Rosenblatt, F. Principles of Neurodynamic: Perceptrons and the Theory of Brain Mechanisms / F. Rosenblatt //Spartan Books.– 1962. – C. 616.

262. Минский, М. Персептроны / М. Минский, С. Пейперт // М.: «Мир». – 1971. –С. 263.

263. Hecht-Nielsen, R. Neurocomputing /R. Hecht-Nielsen// Addison-Wesley Publishing. –1990. – P.433.

264. Kohonen, T. Learning Vector Quantization/ T. Koxonen // Neural Networks. –1988. –. suppl 1. –P. 303

265. Хакимов, Б.В. Моделирование корреляционных зависимостей сплайнами: На примере в геологии и экологии / Б. В. Хакимов // Изд-во Моск. ун-та ; СПб. : Нева. –.2003. –.Р.141

266. Галушкин, А. И. Синтез многослойных систем распознавания образов/ А.И. Галушкин//М.: Энергия. – 1974. –Р. 366.

267. Werbos, P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences / P.J. Werbos// Ph.D. thesis. Harvard University, Cambridge, MA. . –1974.

268. Rumelhart, D.E. Learning Internal Representations by Error Propagation / D.E. Rumelhart, G.E. Hinton, R.J. Williams // In: Parallel Distributed Processing. – 1986. –vol. 1. – P. 318 – 362.

269. Барцев, С. И. Адаптивные сети обработки информации / С.И. Барцев, В.А. Охонин// Красноярск: Ин-т физики СО АН СССР. – 1986. – С.20

270. Fukushima, K. Cognitron: A self–organizing multilayered neural network/ K. Fukushima // Biological Cybernetics. – 1975. – № 20. – P.121–36.

271. Fukushima, K. Neocognitron: A self–organizing neural network model for a mechanism of pattern recognition uneffected by shift in position / K. Fukushima // Biological Cybernetics. – 1980. – № 36(4). – P. 193–202.

272. Hopfield, J.J. Learning algorithms and probability distributions in feed-forward and feed-back networks / J. J. Hopfield // Biophysics. –1987. –Vol. 84. – P. 8429-8433.

273. Лазарев, В.М. Нейросети и нейрокомпьютеры: монография / В.М. Лазарев, А.П. Свиридов //Московский государственный технический университет радиотехники, электроники и автоматики. –Москва. – 2011. – 131 с.

274. Ackley, D. H. A Learning Algorithm for Boltzmann Machines / D.H. Ackley, H. David, G.E. Hinton, T. J.Sejnowski //Cognitive Science. – 1985. – №9 (1). – С.147 – 169.

275. Зиновьев, А.Ю. Визуализация произвольных данных методом упругих карт / А.Ю. Зиновьев, А.А. Питенко // Радіоелектроніка, інформатика, управління. – 2000. – № 1. – Р. 76-85.

276. Горбань, А.Н. Визуализация данных методом упругих карт / А.Н. Горбань, А.А. Зиновьев // Информационные технлогии. – 2000. – №6. – С. 26-36

277. Gorban, A.N. Vizualization of any data using elastic map method / A.N. Gorban, A.A. Pitenko, A.Y. Zinov'ev, D.C. Wunsch // Smart Engineering Sysytem Design. – 2001. – V.11. – P. 363 – 368.

278. Gorban, A.N. Method of Elastic Maps and Applications in Data Visualization and Data Modeling / A.N. Gorban, A. Yu. Zinovyev // International Journal of Computing Anticipatory Systems, CHAOS. 201. – V12. – P.353 – 369.

279. Горбань, А.Н. Визуализация данных. Метод упругих карт / А. Н. Горбань, А.Ю. Зиновьев, А.А. Питенко. // Нейрокомпьютер. – 2002. – № 4. – С.19 – 30.

280. Зиновьев, А. Ю. Метод упругих карт для визуализации данных: Алгоритмы, программное обеспечение и приложения в биоинформатике: дис... канд. техн. наук /А.Ю. Зиновьев //Красноярск. – 2001. – 164 с.

281. Носиров, И. С. Построение системы управления электроприводными системами металлорежущих станков с нейронными сетями: дис... канд. техн. наук / И.С. Носиров // Санкт-Петербург. – 2019. – 113 с.

282. Манусов, В.З. Нейронные сети: прогнозирование электрической нагрузки и потерь мощности в электрических сетях. От романтики к прагматике: монография / В.З. Манусов, С.В. Родыгина // Новосибирский государственный технический университет. – Новосибирск. – 2018. – 303 с.

283. Джейн, А.К. Введение в искусственные нейронные сети / А.К. Джейн, Ж.Мао , К.М. Моиуддин // Открытые системы. СУБД. – 1997. – №4. – С.16

284. Осовский, С. Нейронные сети для обработки информации/ С. Осовский // Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика. – 2002. – С. 344

285. Broomhead, D.S. Multivariable functional interpolation and adaptive networks / D.S. Broomhead, D. Lowe // Complex Systems. – 1988. – Vol.2. – P.321 – 355.

286. Алкезуини, М. М. Совершенствование алгоритмов обучения сетей радиальных базисных функций для решения задач аппроксимации / М.М. Алкезуини, В.И. Горбаченко // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 123-138.

287. Федотов, А.М. Некорректные задачи со случайными ошибками в данных / А.М. Федотов // Новосибирск: Наука. – 1982. – С. 280

288. Севастьянов, А.А. Решение обратных некорректных задач в прикладной спектроскопии с помощью вейвлет-анализа и нейронных сетей: дис…канд. физ. – мат. наук / А.А. Севастьянов // Казань. – 2004. – С.124.

289. Kharintsev, S.S. Inverse problems in the restoration of signal with fractal Gaussian noise in applied spectroscopy / S.S. Kharintsev, R.R. Nigmatullin, M.Kh. Salakhov // Asian Journal of Spectroscopy. – 1999. – V.3 (2). – P.49-67.

290. Севастьянов, А.А. Нейросетевая регуляризация решения обратных некорректных задач прикладной спектроскопии / А. А. Севастьянов С. С. Харинцев М. Х. Салахов // электронный журнал «Исследовано в России». – 2003. – Р. 2254-2266.

291. Тихонов, А.Н. О решении некорректно поставленных задач и методе регуляризации / А. Н. Тихонов // Докл. АН СССР. – 1963. – Т. 151, № 3. – С. 501–504

292. Тихонов, А. Н. Методы решения некорректных задач / А. Н. Тихонов, В.Я. Арсенин // М.: Наука. – 1979. – С. 283.

293. Тихонов, А. Н. О некорректных задачах линейной алгебры и устойчивом методе их решения / А.Н. Тихонов // ДАН СССР. – 1965. – Т. 163, № 3. – С. 591 – 594.

294. Тихонов, А. Н. Регуляризирующие алгоритмы и априорная информация / А.Н. Тихонов, А.В. Гончарский, В.В. Степанов, А.Г. Ягола // М.: Наука. – 1983. – С.198.

295. Bishop, Ch.M. Pattern Recognition and Machine Learning / Ch. M. Bishop // Springer Science+Business Media, LLC. – 2006. – P. 738.

296. Boyd, S. Convex Optimization / S. Boyd, L. Vandenberghe // UK: Cambridge University Press. – 2004. – P. 716.

297. Tibshirani, R. J. Regression shrinkage and selection via the lasso / R. J. Tibshirani // Journal of the Royal Statistical Society. Series B (Methodological). – 1996. – Vol. 58(1). – P.267–288.

298. Krutikov, V.N. Subgradient learning methods of neural networks research / V.N. Krutikov, D.V. Arishev // Vestnik KemGU. – 2004. – Vol. 3(3). P.119-123

299. Krutikov, V.N. Training of sigmoidal neural networks in conditions of interference on the example of the credit scoring data approximation problem / V. N. Krutikov, N.S. Samoilenko, L.A. Kazakovtsev, M.N. Zhalnin// Economy and management of control systems. – 2018.– Vol. 28 (2-3). – P.388-399.

300. Krutikov, V.N. On the applicability of non-smooth regularization in construction of radial artificial neural networks / V.N. Krutikov, N.S. Samoilenko, I.R. Nasyrov, L.A. Kazakovtsev // Control systems and information technologies. – 2018. – Vol 2(72). – P.70-75.

301. Shkaberina, G. Sh. New Method of Training Two-layer Sigmoid Neural Networks Using Regularization / V. N.Krutikov, L. A.Kazakovtsev, G. Sh. Shkaberina,V. L.Kazakovtsev // IOP Conference Series: Materials Science and Engineering 2019. – 2019. – Vol. 537, Issue 4. – Article ID 042055. – 6 P. DOI:10.1088/1757-899X/537/4/042055.

302. Shkaberina, G. S. New Methods of Training Two-Layer Sigmoidal Neural Networks with Regularization / V. N.Krutikov, G. S.Shkaberina, M. N.Zhalnin,L. A. Kazakovtsev // 2019 International Conference on Information Technologies (InfoTech),St. St. Constantine and Elena resort (near the city of Varna), Bulgaria, 2019.– 2019.–P. 1– 4. DOI: 10.1109/InfoTech.2019.8860890.

303. Shkaberina, G. Sh. Regularization methods for neural network models and logistic regression models in the problem of classifying industrial products into homogeneous batches / Krutikov V. N., Shkaberina G. Sh., Tovbis E. M., Kazakovtsev L.A. // 2020 International Conference on Information Technologies (InfoTech),Varna, Bulgaria2020. –2020.–P. 1-4. DOI: 10.1109/InfoTech49733.2020.9211013.

304. Krutikov, V. Algorithms for Reduction of Input Space Dimensionality in Regression-Based Classification Models / V. Krutikov, G. Shkaberina, L. Kazakovtsev // 2019 International Russian Automation Conference (RusAutoCon). Sochi, Russia. – 2019. – P. 1-5.

305. Hofmann, H. German Credit Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/

306. Australian Credit Approval. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/australian/

307. Maroco, J. Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests / J. Maroco, D. Silva, A. Rogrigues, M. Guerreiro, I. Santana, A. Mendonca // BMC Research Notes. – 2011. – № 4(1). – P. 299.

308. Kruger, M. Experimental comparison of ad hoc methods for classification of maritime vessels based on real-life AIS data / M. Kruger // IEEE 2018 21st Conference on Information Fusion (FUSION). – 2018. –P. 672 – 679.

309. McKenzie, D. Classification by similarity: an overview of statistical methods of case-based reasoning / D. McKenzie, R. Forsyth // Computers in human behavior. –1995. – Vol.11(2). – P.273-288.

310. Batuta, T.V. Automatic text classification methods / T.V. Batuta // Software products and systems. – 2017. – Vol.30(1). – P. 85-99.

311. Yuan, Y. A Comparative Analysis of SVM, Naive Bayes and GBDT for Data Faults Detection in WSNs / Y. Yuan, S. Li, Z. Xingjian, J. Sun // 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). – 2018. – P. 394-399.

312. MatLab and Simulink official page https://www.mathworks.com/products/matlab.html

ПРИЛОЖЕНИЕ А. Результаты решения задач автоматической группировки

Таблица А.1 – Результаты автоматической группировки с различными мерами расстояний (данные Микросхемы 1526IE10_002)

Партии	Квадрат евклидова расстояния	Квадрат расстояния Махаланобиса	Манхэттенское расстояние	Косинусное расстояние	Корреляционное расстояние
Частичная выборка, состоящая из двух однородных партий, шт. (доля)					
Партия 1 (n=71)	71 (1,00)	49 (0,69)	71 (1,00)	71 (1,00)	71 (1,00)
Партия 2 (n=116)	112 (0,97)	74 (0,64)	114 (0,98)	58 (0,50)	58 (0,50)
среднее	**0,98**	**0,66**	**0,99**	**0,69**	**0,69**
Сумма расстояний	325,6793	7288,697	251,35	0,0056	0,0058
Частичная выборка, состоящая из трех однородных партий, шт. (доля)					
Партия 1 (n=71)	71 (1,00)	28 (0,39)	71 (1,00)	71 (1,00)	71 (1,00)
Партия 2 (n=116)	89 (0,77)	34 (0,29)	70 (0,60)	57 (0,49)	55 (0,47)
Партия 6 (n=113)	33 (0,29)	47 (0,42)	47 (0,42)	62 (0,55)	70 (0,62)
среднее	**0,64**	**0,36**	**0,63**	**0,63**	**0,65**
Сумма расстояний	348,3259	11555,83	237,95	0,0062	0,0071
Частичная выборка, состоящая из четырех однородных партий, шт. (доля)					
Партия 1 (n=71)	70 (0,99)	47 (0,66)	71 (1,00)	70 (0,99)	70 (0,99)
Партия 2 (n=116)	78 (0,67)	83 (0,72)	64 (0,55)	78 (0,67)	84 (0,72)
Партия 5 (n=146)	96 (0,66)	88 (0,60)	105 (0,72)	96 (0,66)	104 (0,71)
Партия 6 (n=113)	44 (0,39)	91 (0,81)	50 (0,44)	44 (0,39)	38 (0,37)
среднее	**0,65**	**0,69**	**0,65**	**0,65**	**0,66**
Сумма расстояний	473,174	26146,35047	401,4	0,0012	0,0011
Полная выборка, шт. (доля)					
Партия 1 (n=71)	67 (0,94)	70 (0,99)	68 (0,96)	67 (0,94)	71 (1,00)
Партия 2 (n=116)	4 (0,03)	4 (0,03)	4 (0,03)	4 (0,03)	78 (0,67)
Партия 3 (n=1867)	578 (0,31)	223 (0,12)	558 (0,30)	578 (0,31)	0 (0,00)
Партия 4 (n=1250)	403 (0,32)	127 (0,11)	446 (0,36)	406 (0,33)	227 (0,18)
Партия 5 (n=146)	66 (0,45)	81 (0,55)	63 (0,43)	64 (0,44)	78 (0,53)
Партия 6 (n=113)	88 (0,78)	113 (1,00)	82 (0,73)	88 (0,78)	32 (0,28)

Продолжение таблицы А.1

Партии	Квадрат евклидова расстояния	Квадрат расстояния Махаланобиса	Манхэттенское расстояние	Косинусное расстояние	Корреляционное расстояние
Партия 7 (n=424)	311 (0,73)	404 (0,95)	303 (0,72)	311 (0,73)	314 (0,74)
среднее	**0,38**	**0,26**	**0,38**	**0,38**	**0,20**
Сумма расстояний	5008,127	248808,6	1755,8	0,007	0,004

Таблица А.2 – Результаты кластеризации с обучающей выборкой

Партии	Квадрат расстояния Махаланобиса (с обучением)	Квадрат расстояния Махаланобиса (без обучения)	Квадрат евклидова расстояния	Манхэттенское расстояние
Партия 4+Партия 7 (n=1674) , шт. (доля)				
Партия 4 (n=1250)	850 (0,68)	685 (0,55)	741 (0,59)	895 (0,72)
Партия 7 (n=424)	390 (0,92)	256 (0,60)	228 (0,54)	423 (1,00)
среднее	**0,74**	**0,56**	**0,58**	**0,79**
Средняя сумма расстояний	94467	100898	7119	12272
Партия 1+Партия 7 (n=495) , шт. (доля)				
Партия 7 (n=424)	253 (0,60)	Вырожденная матрица	416 (0,98)	415 (0,98)
Партия 1 (n=71)	71 (1,00)		71 (1,00)	71 (1,00)
среднее	**0,65**	**-**	**0,98**	**0,98**
Средняя сумма расстояний	17551	-	1233	2795
Партия 6+Партия 7 (n=537) , шт. (доля)				
Партия 7 (n=424)	223 (0,53)	Вырожденная матрица	244 (0,58)	282 (0,67)
Партия 6 (n=113)	113 (1,00)		92 (0,81)	84 (0,75)
среднее	**0,63**	**-**	**0,63**	**0,68**
Средняя сумма расстояний	18190	-	1396	3300
Партия 2+Партия 7 (n=540) , шт. (доля)				
Партия 7 (n=424)	216 (0,51)	Вырожденная матрица	217 (0,51)	274 (0,65)
Партия 2 (n=116)	116 (1,00)		95 (0,82)	97 (0,84)
среднее	**0,62**	**-**	**0,58**	**0,69**
Средняя сумма расстояний	18190	-	1123	3090
Партия 5+Партия 7 (n=570) , шт. (доля)				
Партия 7 (n=424)	424 (1,00)	218 (0,51)	380 (0,90)	385 (0,91)

Продолжение таблицы А.2

Партии	Квадрат расстояния Махаланобиса (с обучением)	Квадрат расстояния Махаланобиса (без обучения)	Квадрат евклидова расстояния	Манхэттенское расстояние
Партия 5 (n=146)	136 (0,93)	85 (0,58)	146 (1,00)	146 (1,00)
среднее	**0,98**	**0,53**	**0,92**	**0,93**
Средняя сумма расстояний	34385	34282	1250	3202
Партия 1+Партия 4 (n=1321) , шт. (доля)				
Партия 1 (n=71)	71 (1,00)	47 (0,66)	71 (1,00)	71 (1,00)
Партия 4 (n=1250)	471 (0,38)	653 (0,52)	772 (0,62)	642 (0,51)
среднее	**0,41**	**0,53**	**0,64**	**0,54**
Средняя сумма расстояний	82458	79599	7237	11120
Партия 6+Партия 4 (n=1363) , шт. (доля)				
Партия 4 (n=1250)	410 (0,33)	648 (0,52)	735 (0,59)	570 (0,46)
Партия 6 (n=113)	102 (0,90)	59 (0,52)	67 (0,59)	85 (0,75)
среднее	**0,38**	**0,52**	**0,59**	**0,48**
Средняя сумма расстояний	82649	82054	5452	10014
Партия 2+Партия 4 (n=1366) , шт. (доля)				
Партия 4 (n=1250)	412 (0,33)	622 (0,50)	769 (0,62)	485 (0,39)
Партия 2 (n=116)	98 (0,85)	69 (0,59)	76 (0,66)	96 (0,82)
среднее	**0,37**	**0,51**	**0,62**	**0,43**
Средняя сумма расстояний	82693	82318	5410	9996
Партия 4+Партия 5 (n=1396) , шт. (доля)				
Партия 4 (n=1250)	953 (0,76)	772 (0,62)	772 (0,62)	873 (0,70)
Партия 5 (n=146)	91 (0,62)	91 (0,62)	146 (1,00)	146 (1,00)
среднее	**0,75**	**0,62**	**0,66**	**0,73**
Средняя сумма расстояний	99605	83963	6689	11619
Партия 1+Партия 6 (n=184)				
Партия 1 (n=71)	71 (1,00)	Вырожденная матрица	71 (1,00)	71 (1,00)
Партия 6 (n=113)	111 (0,98)		113 (0,100)	113 (1,00)
среднее	**0,99**	-	**1,00**	**1,00**
Средняя сумма расстояний	6500	-	354	797

Продолжение таблицы А.2

Партии	Квадрат расстояния Махаланобиса (с обучением)	Квадрат расстояния Махаланобиса (без обучения)	Квадрат евклидова расстояния	Манхэттенское расстояние
Партия 1+Партия 2 (n=187) , шт. (доля)				
Партия 1 (n=71)	71 (1,00)	Вырожденная матрица	71 (1,00)	71 (1,00)
Партия 2 (n=116)	116 (1,00)		112 (0,97)	114 (0,98)
среднее	**1,00**	-	**0,98**	**0,99**
Средняя сумма расстояний	6481	-	325	747
Партия 1+ Партия 5 (n=217) , шт. (доля)				
Партия 1 (n=71)	71 (1,00)	39 (0,56)	70 (0,99)	71 (1,00)
Партия 5 (n=146)	84 (0,58)	80 (0,55)	99 (0,68)	108 (0,74)
среднее	**0,71**	**0,55**	**0,78**	**0,83**
Средняя сумма расстояний	22199	13004	223	841
Партия 2+ Партия 6 (n=229) , шт. (доля)				
Партия 2 (n=116)	91 (0,78)	Вырожденная матрица	89 (0,77)	70 (0,60)
Партия 6 (n=113)	87 (0,77)		37 (0,33)	48 (0,42)
среднее	**0,78**	-	**0,55**	**0,52**
Средняя сумма расстояний	7319	-	282	903
Партия 5+ Партия 6 (n=259) , шт. (доля)				
Партия 5 (n=146)	96 (0,66)	81 (0,55)	146 (1,00)	146 (1,00)
Партия 6 (n=113)	113 (1,00)	66 (0,59)	105 (0,93)	109 (0,75)
среднее	**0,81**	**0,57**	**0,97**	**0,99**
Av. Средняя сумма расстояний of distances	23172	6564	512	1246
Партия 5+ Партия 2 (n=262) , шт. (доля)				
Партия 2 (n=116)	116 (1,00)	67 (0,57)	108 (0,93)	109 (0,94)
Партия 5 (n=146)	78 (0,54)	80 (0,55)	146 (1,00)	146 (1,00)
среднее	**0,74**	**0,56**	**0,97**	**0,97**
Средняя сумма расстояний	23070	15710	458	1175

Таблица А.3 – Результаты эксперимента Серия I. Первая группа

Серия I	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,8805	0,7903	0,9971	0,61628	0,71506	94860	148422	101945	12196	7810,8
Min	0,5009	0,5549	0,6464	0,59146	0,49968	93548	134171	947394	11572	6694
Average	0,6951	0,6119	0,9245	0,60079	0,55219	93851	13469	96622	11652	7062,09
Std dev	0,1390	0,0412	0,1042	0,00648	0,05949	304,42	259,52	2401,46	160,77	329,42
V						0,3244	1,9268	2,48542	1,3798	4,66462
R						1312,3	1425,2	7205,97	623,63	1116,48

Таблица А.4 – Результаты эксперимента Серия II. Первая группа

Серия II	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7042	0,5941	0,8606	0,5231	0,54539	101596	136963	109317	104604	5615,56
Min	0,4548	0,4775	0,5899	0,48239	0,44481	99650	13181	101409	9542,1	4114,01
Average	0,5474	0,4915	0,7051	0,50017	0,50492	100402	13229	102373	9939,9	4534,36
Std dev	0,0759	0,0262	0,0857	0,02003	0,02334	536,14	91,974	2060,45	462,72	399,977
V						0,5339	0,6952	2,01269	4,6552	8,82103
R						1946,2	515,16	7907,82	918,17	1501,56

Таблица А.5 – Результаты эксперимента Серия III. Первая группа

Серия III	индексРэнда					Целеваяфункция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,8359	0,6545	0,8583	0,7343	0,7382	249093	37573	260042	22958	14505,5
Min	0,4727	0,6326	0,7378	0,6810	0,6729	243613	36628	251830	19408	6557,18
Average	0,6129	0,6400	0,8034	0,7094	0,7158	246082	36754	254004	19733	7319,47
Std dev	0,0920	0,0043	0,0360	0,0172	0,0112	1199,4	209,27	1886,97	660,51	1418,11
V						0,4874	0,5694	0,74289	3,3472	19,3746
R						5473,6	944,97	8211,23	3550,4	7948,27

Таблица А.6 – Результаты эксперимента Серия IV. Первая группа

Серия VI	индексРэнда					Целевыефункция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,6266	0,5579	0,7648	0,66202	0,66905	168235	19742	173310	12090	6203,87
Min	0,4601	0,5223	0,6163	0,54736	0,54869	164272	19018	169878	11197	3226,19
Average	0,5041	0,5332	0,6678	0,57916	0,58167	165533	19128	170719	11688	3797,11
Std dev	0,0435	0,0072	0,0410	0,03323	0,02955	736,82	149,96	635,930	361,71	724,181
V						0,4451	0,7839	0,37250	3,0946	19,0719
R						3963,3	723,90	3432,03	893,32	2977,68

Таблица А.7 – Результаты эксперимента Серия I. Вторая группа

Серия I	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,80022	0,83448	0,78595	0,61627	0,70760	105009	27926,0	104542	12196	7710,29
Min	0,61558	0,50384	0,61352	0,59145	0,49967	628609,	22251,3	647531,	11572,3	6694,78
Average	0,70323	0,7216	0,67758	0,60083	0,55920	812421,	25258,3	789831,	11677,2	7009,32
Std dev	0,06432	0,0884	0,0608	0,00730	0,05590	165378,	1844,86	139619,	187,318	335,365
V						20,3562	7,30398	17,6770	1,60412	4,78456
R						421487	5674,68	397890,2	623,6327	1015,517

Таблица А.8 – Результаты эксперимента Серия II. Вторая группа

Серия II	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,8881	0,8126	0,9929	0,61628	0,71039	138848	19383	147155	12196	7743,48
Min	0,5189	0,5008	0,5165	0,59146	0,49966	135032	17583	140162	11572	6695,54
Average	0,7103	0,6133	0,8247	0,60034	0,55766	136239	17663	143343	11656	6955,42
Std dev	0,1146	0,0465	0,1524	0,00676	0,05771	623,72	334,09	2557,39	160,44	306,932
V						0,4578	1,8915	1,78410	1,3764	4,41284
R						3816,6	1801,7	6993,62	623,63	1047,94

Таблица А.9 – Результаты эксперимента Серия III. Вторая группа

Серия III	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,6775	0,6638	0,5899	0,5231	0,53692	959764	21245	674623	10460	5772,89
Min	0,4789	0,5131	0,5141	0,48239	0,44482	495239	17250	511829	9542,1	4115,13
Average	0,5617	0,5543	0,5485	0,49504	0,50721	617115	17889	589969	9817,5	4545,46
Std dev	0,0407	0,0253	0,0176	0,01855	0,01814	130286	930,26	47828,9	427,85	487,675
V						21,112	5,2003	8,10702	4,3580	10,7289
R						464526	3994,7	162793	918,17	1657,76

Таблица А.10 – Результаты эксперимента Серия IV. Вторая группа

Серия VI	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7694	0,5632	0,8596	0,5231	0,53150	142927	181748	149957	10460	5514,94
Min	0,4526	0,4617	0,4884	0,48239	0,44639	135973	17657	140437	9542,1	4114,47
Average	0,6149	0,4897	0,7138	0,4921	0,49832	137826	17723	142801	9756,3	4544,81
Std dev	0,0762	0,0265	0,1115	0,01709	0,02445	1305,5	114,87	2542,11	394,93	417,729
V						0,9472	0,6481	1,78018	4,0479	9,19135
R						6953,8	516,83	9520,06	918,17	1400,47

Таблица А.11 – Результаты эксперимента Серия VI. Вторая группа

Серия VI	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,5215	0,5513	0,7942	0,60618	0,66961	122365	23083	128612	12090	6143,39
Min	0,4534	0,5151	0,5432	0,54791	0,53722	119529	22495	125855	11197	3228,05
Average	0,4674	0,5277	0,6289	0,57979	0,58326	120915	22634	126519	11594	3732,68
Std dev	0,0119	0,0064	0,0611	0,02549	0,03138	639,84	107,34	662,212	384,98	669,028
V						0,5292	0,4742	0,52341	3,3204	17,9236
R						2835,4	587,88	2756,93	893,39	2915,34

Таблица А.12 – Результаты эксперимента Серия I. Третья группа

Серия I	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,8805	0,8126	0,8126	0,61628	0,70761	137214	19385	178125	123395	7715,82
Min	0,5286	0,5598	0,5144	0,56747	0,49966	135010	17583	158809	11572	6694
Average	0,7252	0,6239	0,6261	0,60039	0,56219	136097	17691	160817	11668	7026,19
Std dev	0,1011	0,0519	0,0774	0,00859	0,06209	537,91	398,38	4450,14	204,48	339,920
V						0,3952	2,2518	2,76720	1,7525	4,83789
R						2204,1	1801,7	19316,1	766,59	1021,49

Таблица А.13 – Результаты эксперимента Серия II. Третья группа

Серия II	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,6985	0,5914	0,6508	0,5231	0,53981	140371	18173	165379	10460	5854,17
Min	0,4411	0,4585	0,4750	0,48239	0,44466	136494	17657	151543	9542,1	4113,69
Average	0,5690	0,4972	0,5288	0,49117	0,50013	137979	17785	155134	9725,7	4632,56
Std dev	0,0691	0,0387	0,0375	0,01613	0,02552	1040,9	177,75	4980,26	373,47	515,606
V						0,7544	0,9994	3,2103	3,8399	11,1300
R						3877,1	515,88	13837,2	918,17	1740,48

Таблица А.14 – Результаты эксперимента Серия III. Группа Третья группа

Серия III	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,7579	0,6444	0,7349	0,73427	0,74379	254255	38757	290440	20509	8267,99
Min	0,4512	0,6287	0,6733	0,69539	0,68172	249583	37812	272546	19408	6570,70
Average	0,5802	0,6355	0,7208	0,71094	0,71729	251524	37966	276636	19632	7298,20
Std dev	0,0779	0,0039	0,0142	0,01701	0,01231	1159,3	217,69	3934,93	268,48	558,889
V						0,4609	0,5734	1,42242	1,3676	7,65789
R						4671,8	945,12	17894,1	1101,6	1697,29

Таблица А.15 – Результаты эксперимента Серия IV. Третья группа

Серия VI	индекс Рэнда					Целевая функция				
	Модель					Модель				
	DM1	DC	DM2	DR	DE	DM1	DC	DM2	DR	DE
Max	0,5082	0,5333	0,7101	0,66213	0,64063	123456	22689	141280	12083	5524,42
Min	0,4516	0,5177	0,5525	0,54753	0,53868	119482	22525	133467	11197	3229,08
Average	0,4692	0,5281	0,6087	0,57451	0,57954	121272	22613	135964	11677	3738,96
Std dev	0,0127	0,0039	0,0445	0,03064	0,02759	882,97	54,569	1620,62	351,19	563,883
V						0,7281	0,2413	1,19195	3,0074	15,0813
R						3972,8	163,49	7813,62	886,29	2295,33

ПРИЛОЖЕНИЕ Б. Зависимость индекса Рэнда RI от значения целевой функции для моделей с различными мерами расстояний

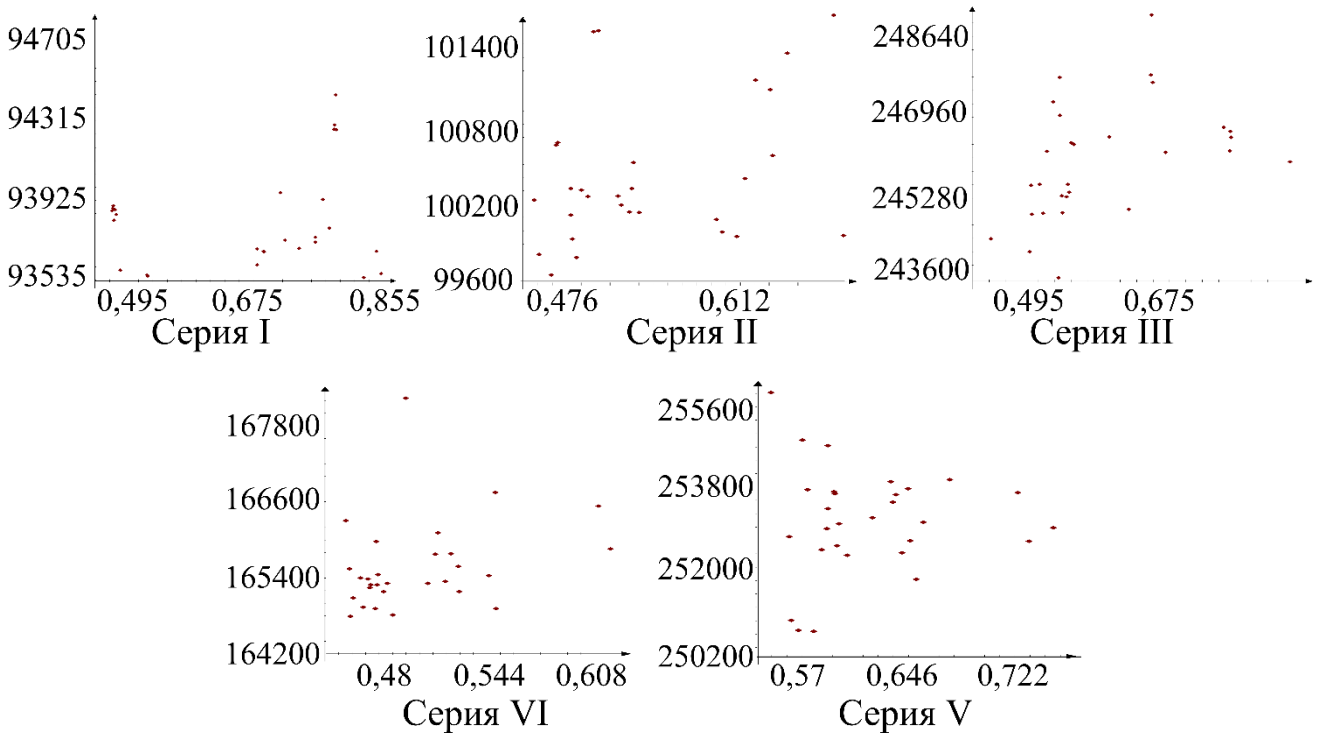


Рисунок Б.1 – Модель DM1. Первая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

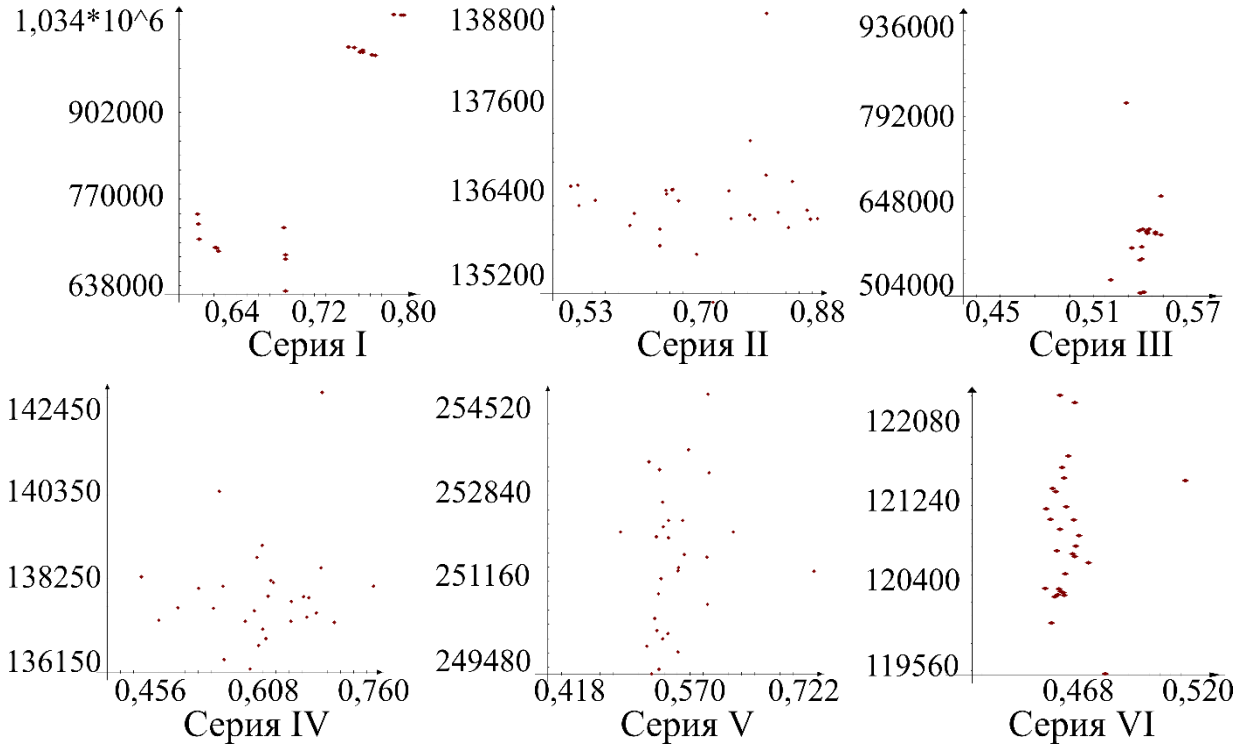


Рисунок Б.2. – Модель DM1. Вторая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

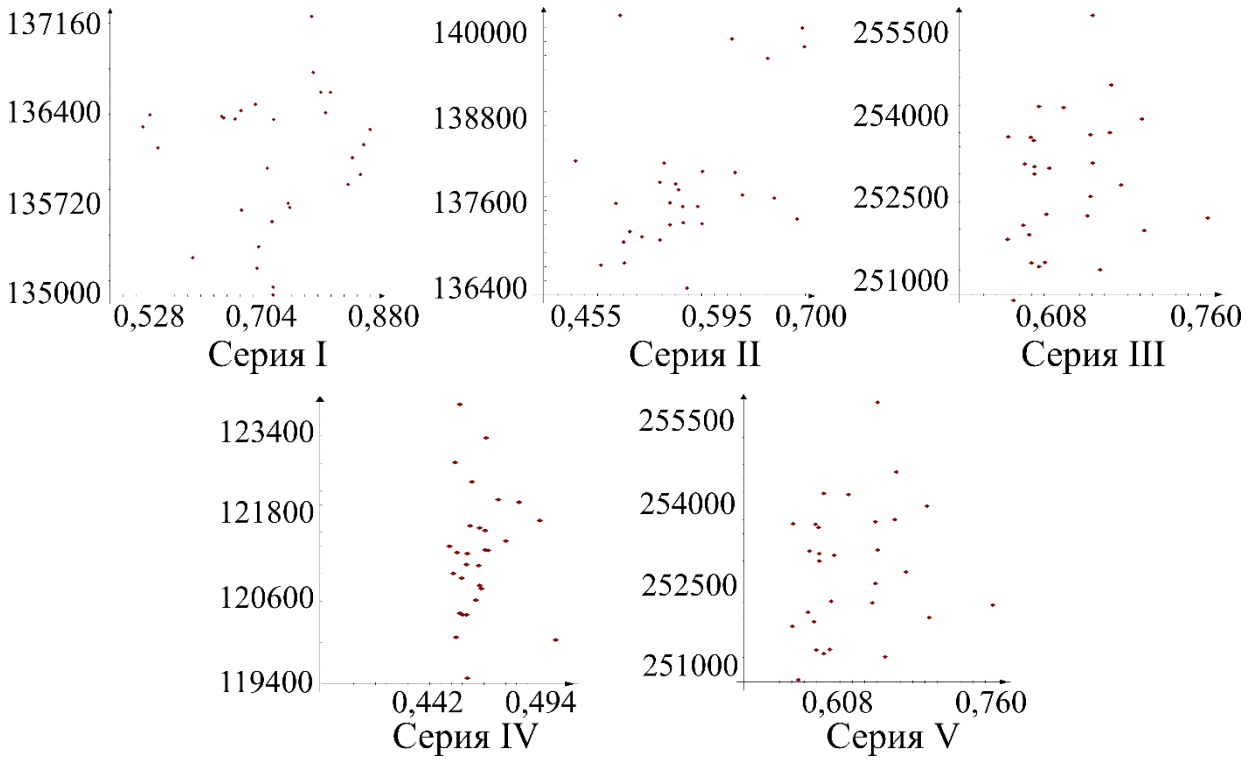


Рисунок Б.3 – Модель DM1. Третья группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

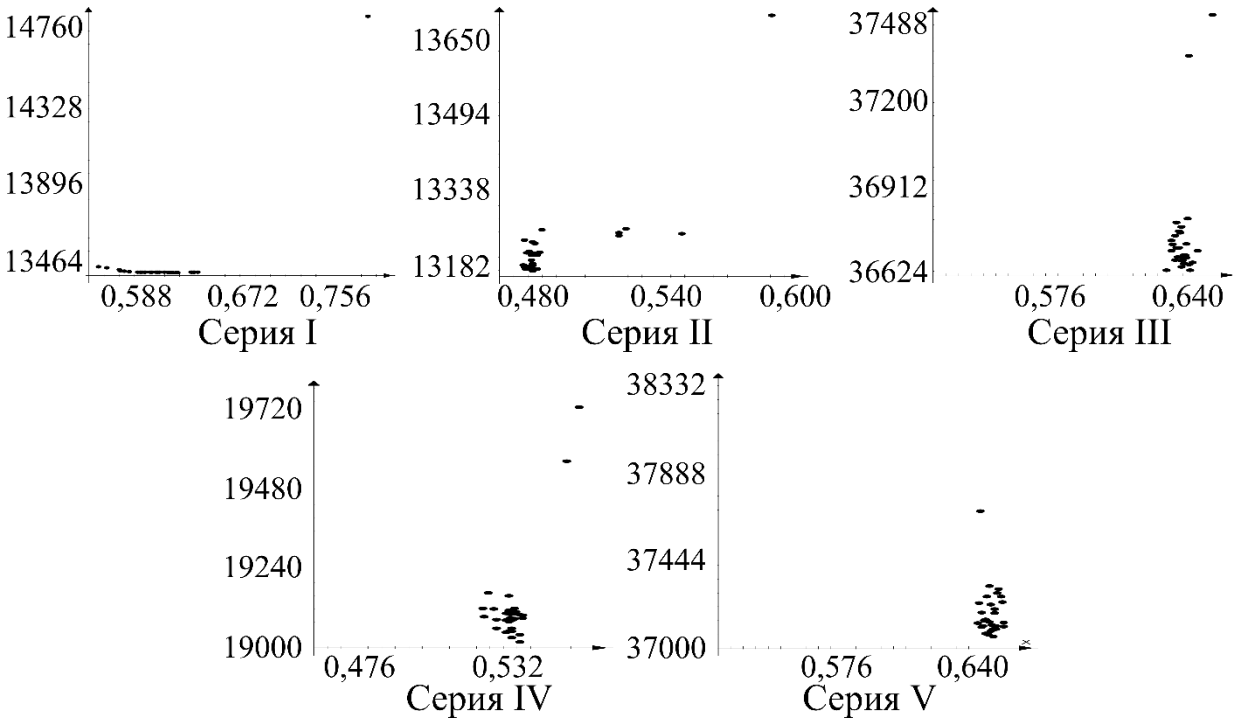


Рисунок Б.4– Модель DC. Перваягруппа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

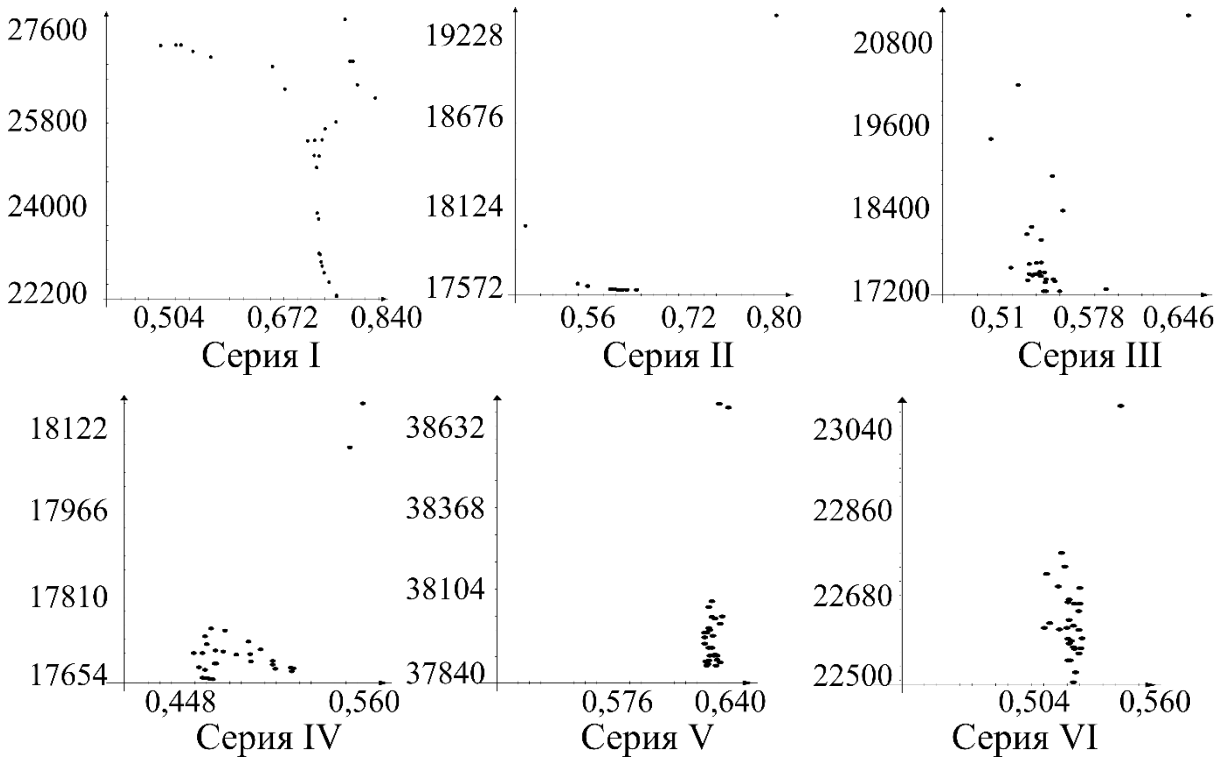


Рисунок Б.5 – Модель DC. Вторая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

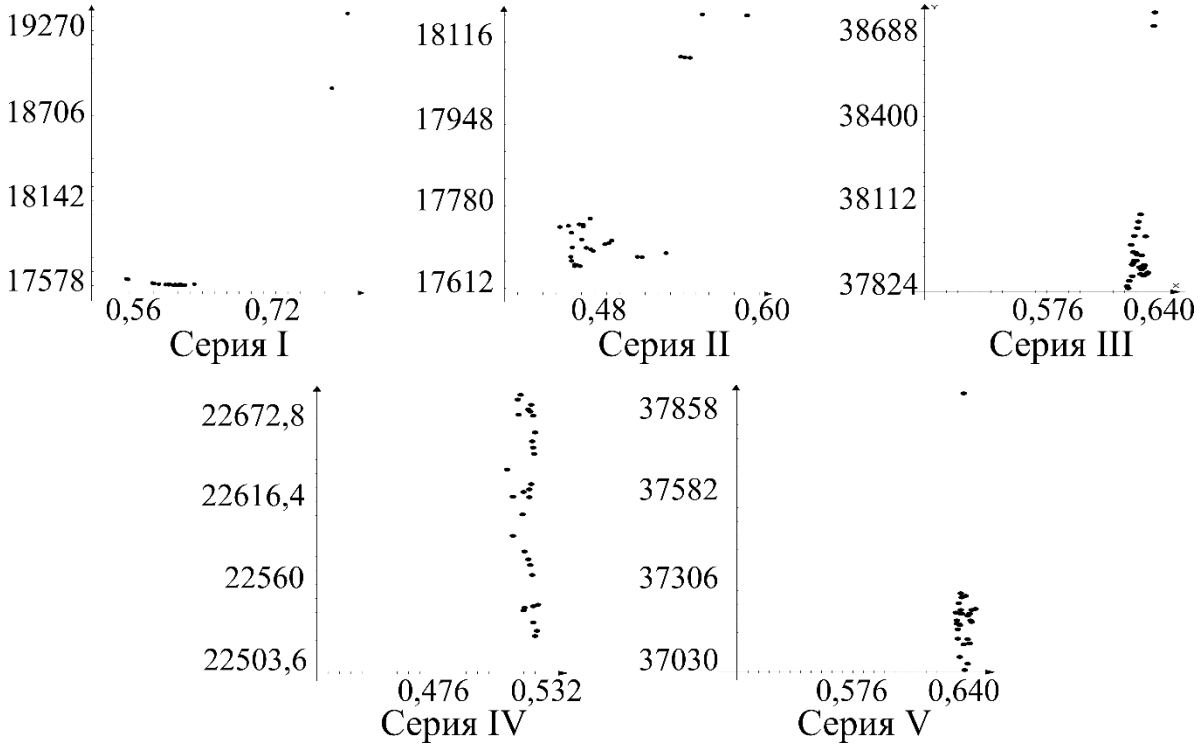


Рисунок Б.6 – Модель DC. Третья группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

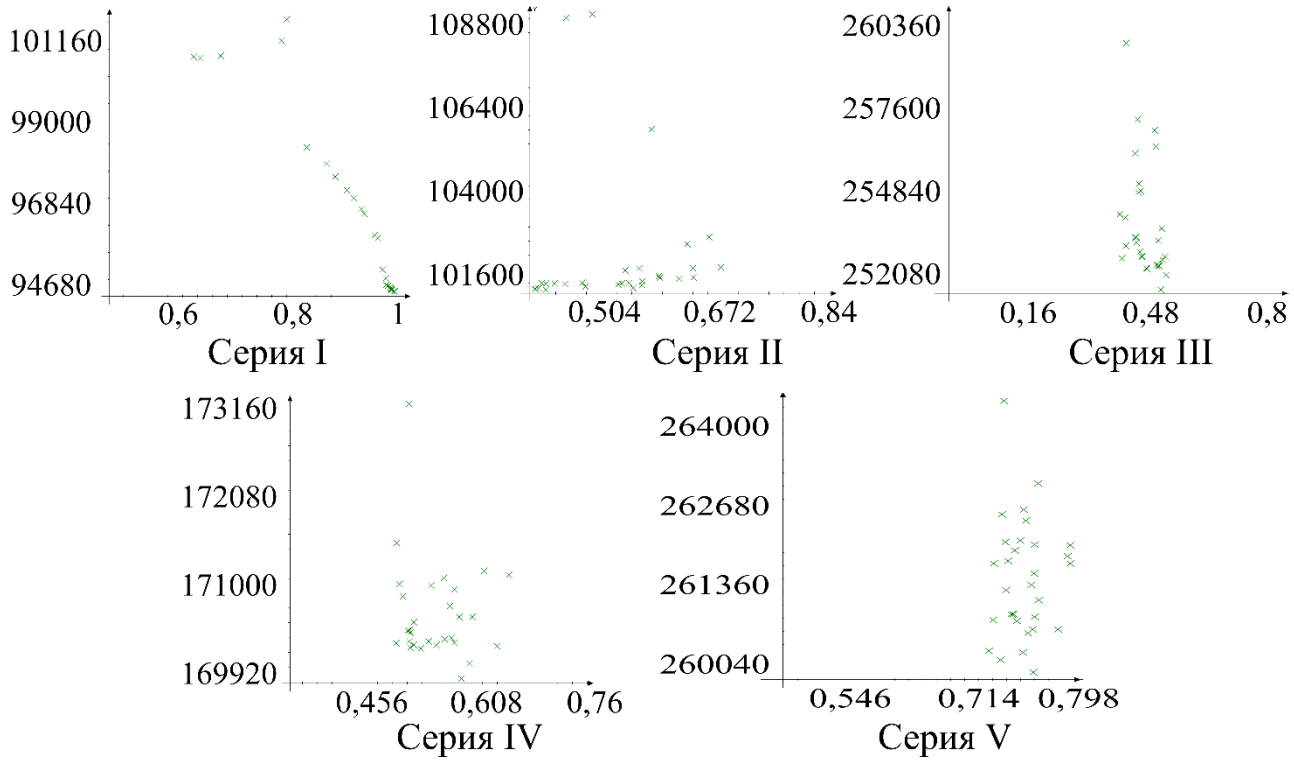


Рисунок Б.7 – Модель DM2.Первая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

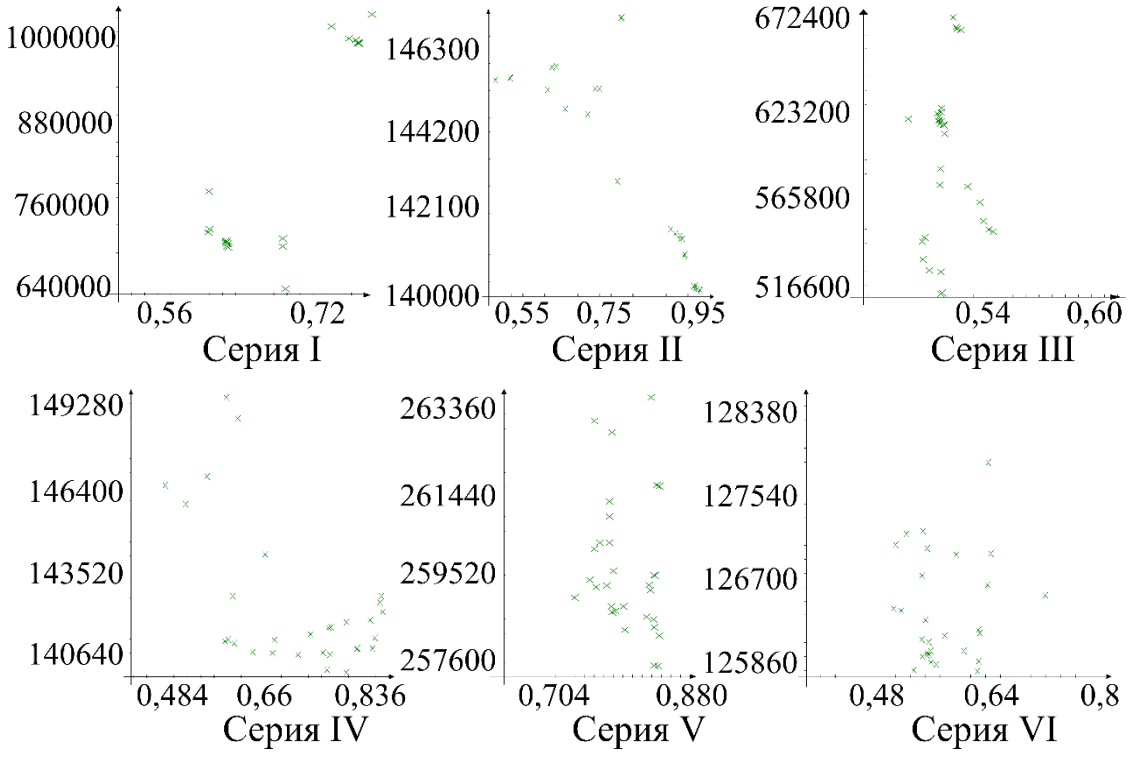


Рисунок Б.8 – Модель DM2.Вторая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

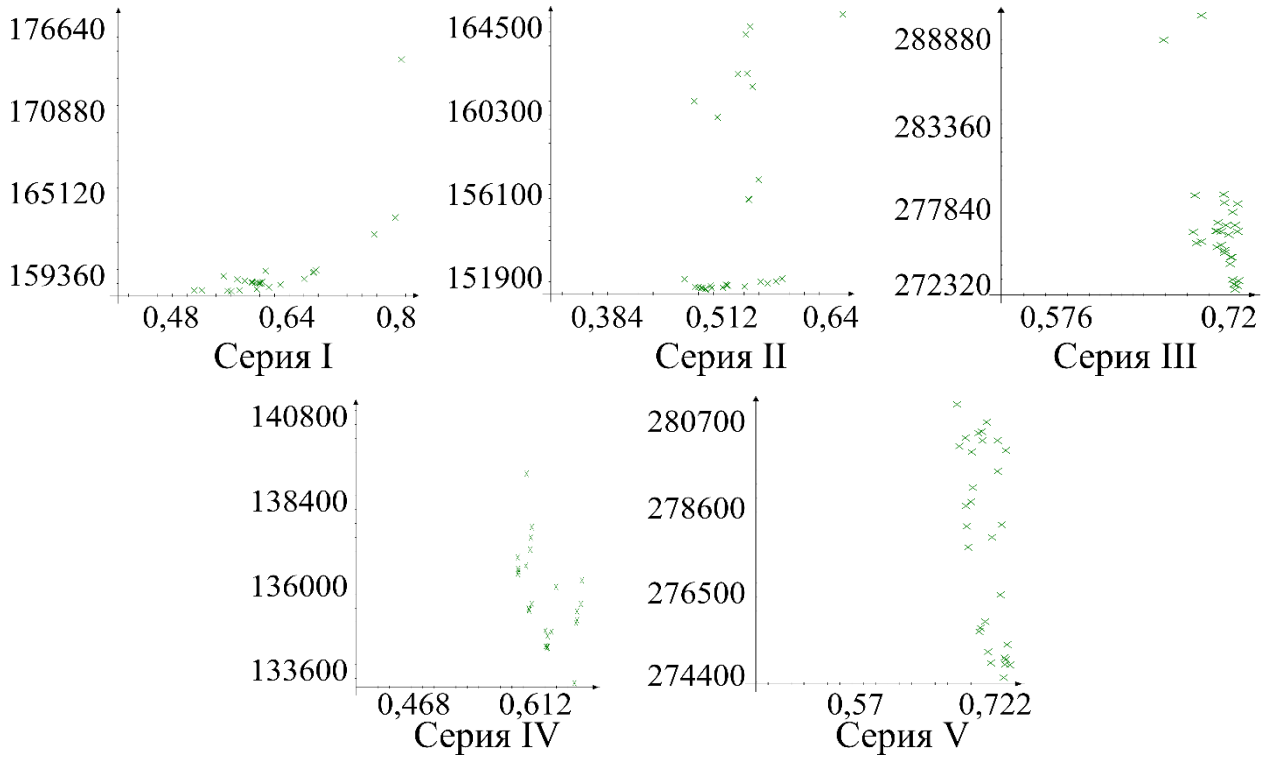


Рисунок Б.9 – Модель DM2.Третья группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

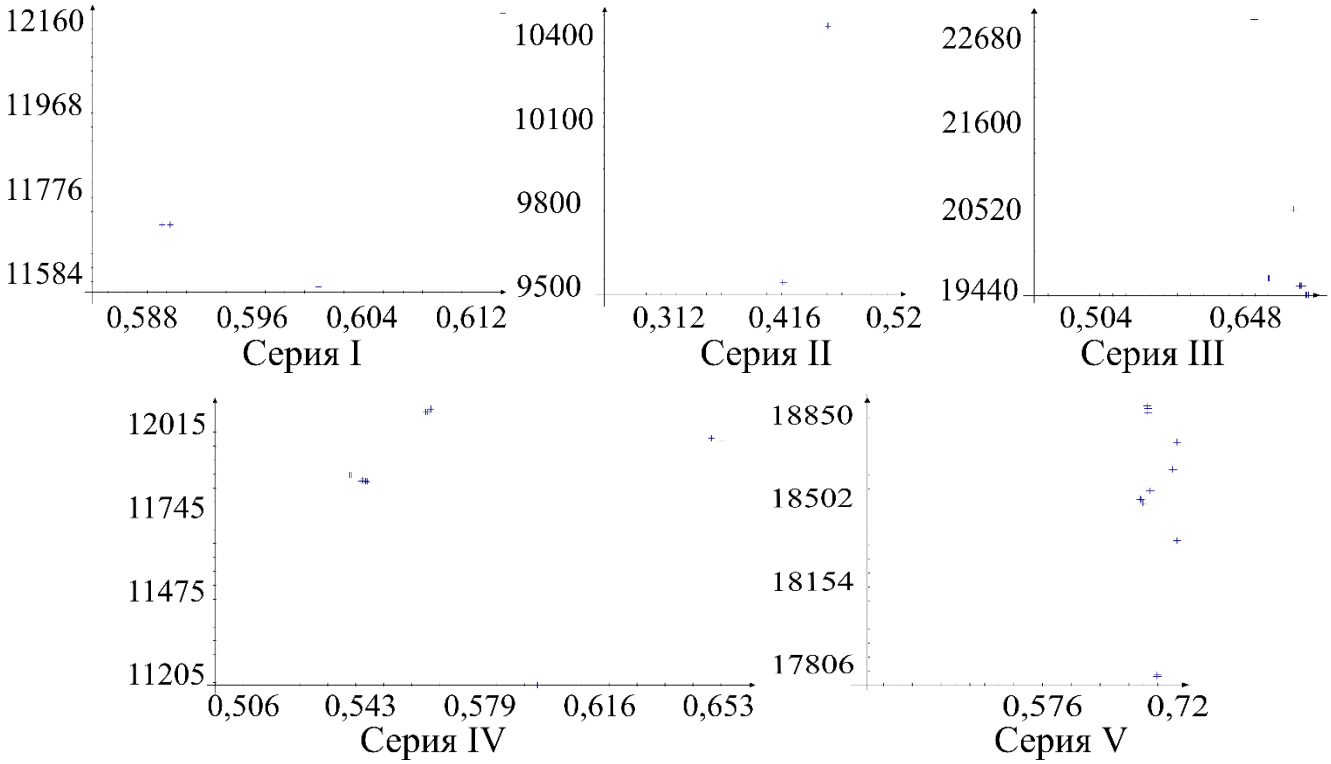


Рисунок Б.10 – Модель DR. Первая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

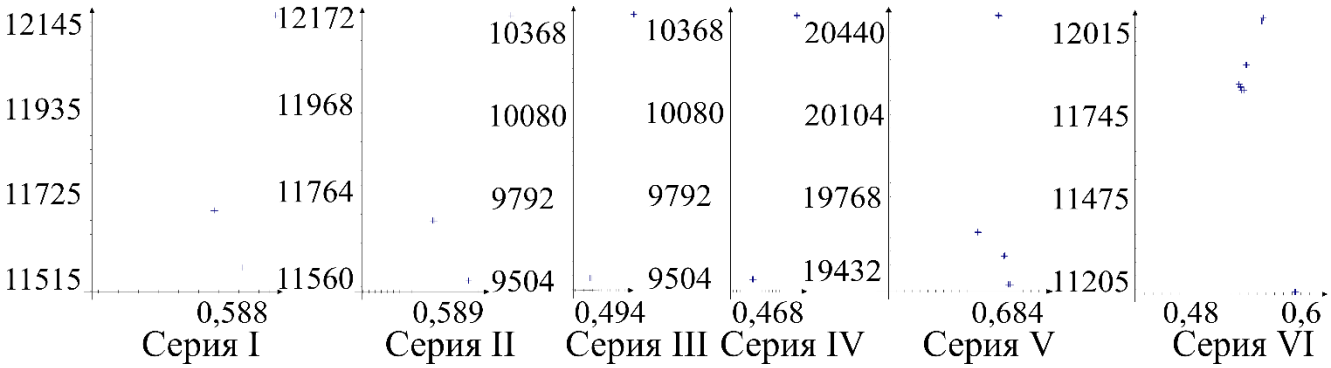


Рисунок Б.11 – Модель DR. Вторая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

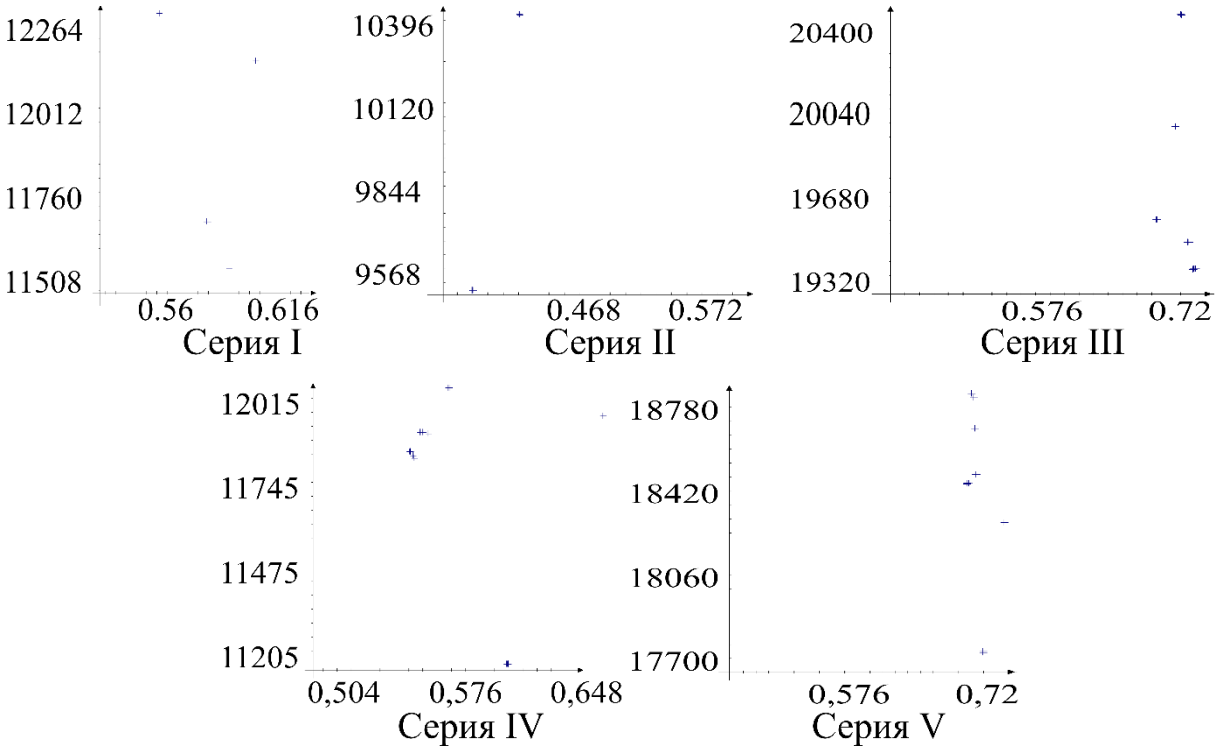


Рисунок Б.12 – Модель DR. Третья группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

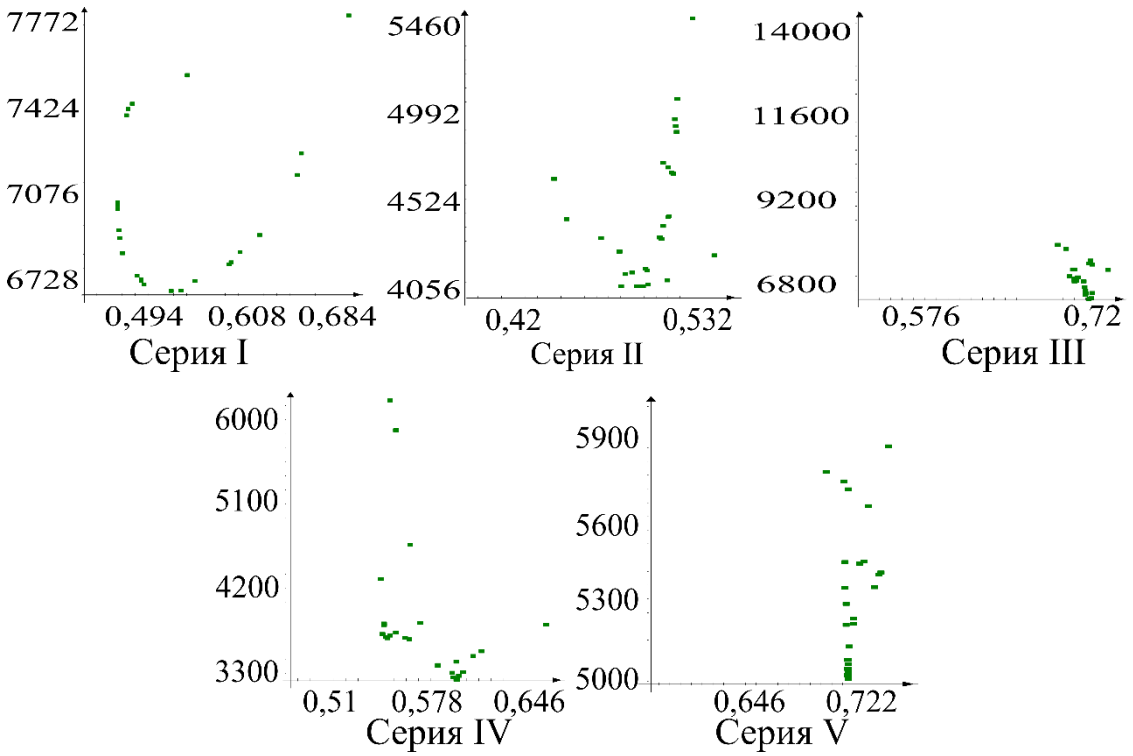


Рисунок Б.13 – Модель DE. Первая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

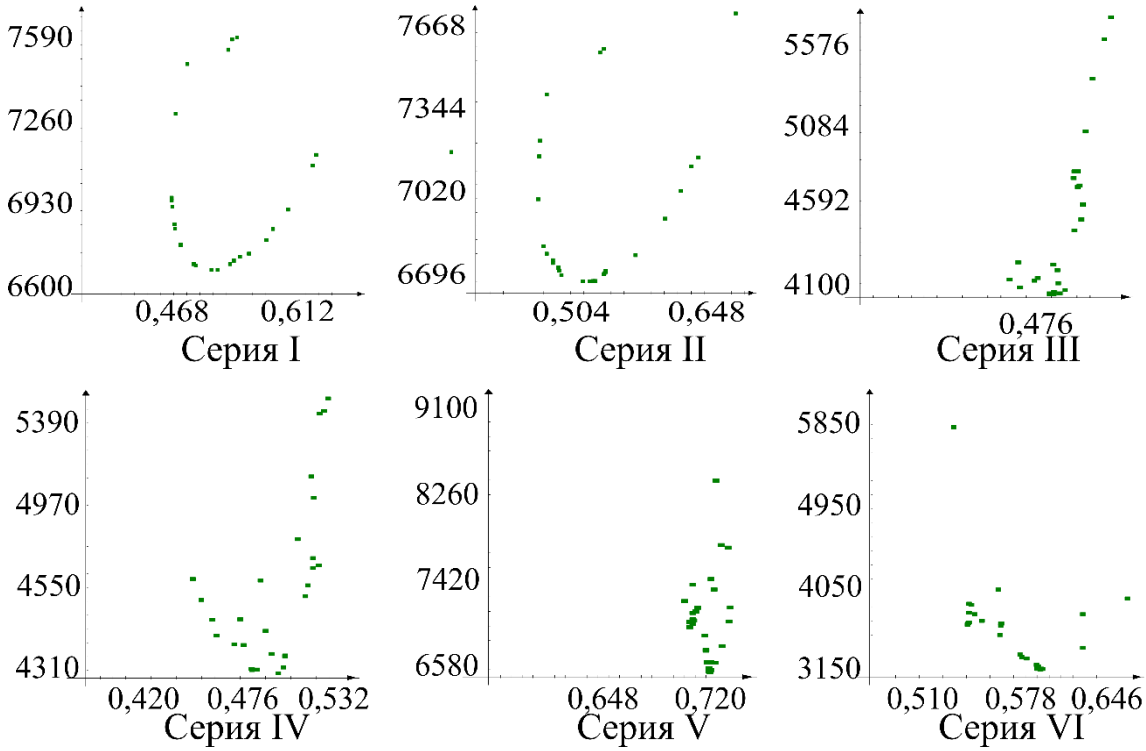


Рисунок Б.14 – Модель DE. Вторая группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

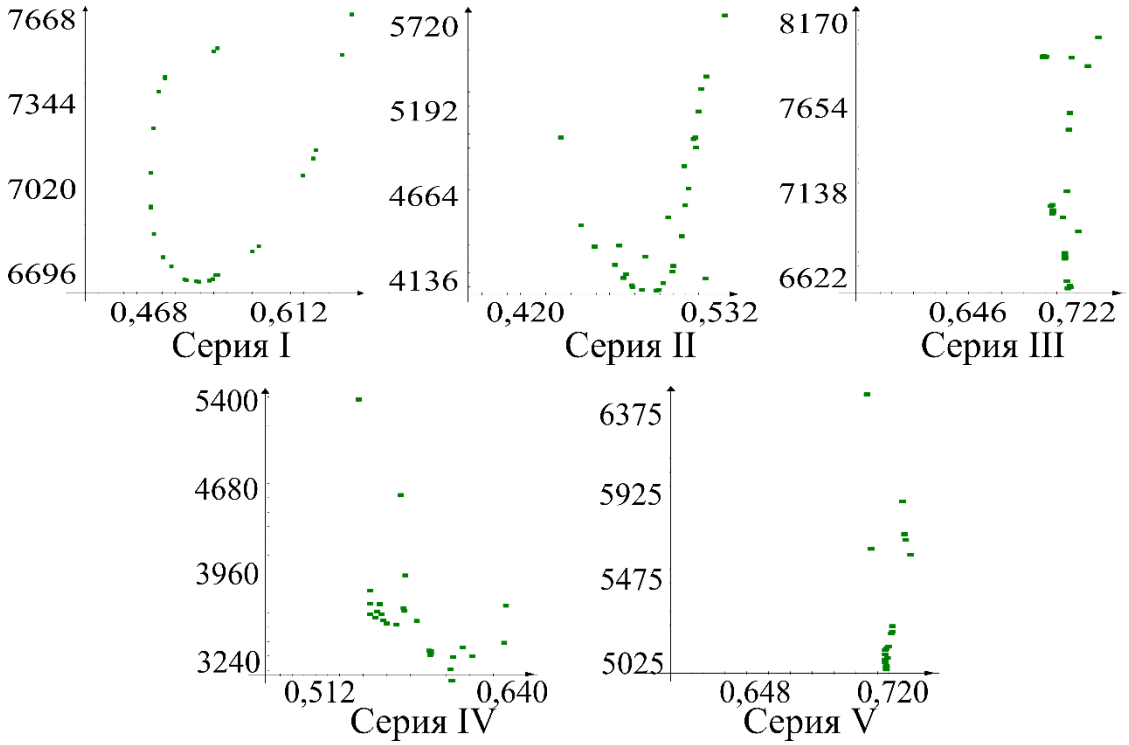


Рисунок Б.15 – Модель DE. Третья группа
(ось X – Индекс Рэнда RI, ось Y – Целевая функция)

ПРИЛОЖЕНИЕ В. Акт об использовании результатов диссертацинного исследования

Акционерное общество
«Испытательный технический центр–НПО ПМ»
ул. Молодежная, 20, г. Железногорск, ЗАТО Железногорск, Красноярский край, Россия, 662970
тел./факс (391-9)-74-97-45, e-mail: itcnpopm@atomlink.ru, ttc@krasmail.ru, http:/www.ttc-npopm.ru
ОКПО 51431138, ОГРН 1032401224272, ИНН 2452027379, ОКВЭД 72.19

АКТ

об использовании результатов диссертационного исследования

Шкабериной Гузели Шарипжановны

Настоящим актом подтверждается, что предложенная модель и алгоритм для решения задачи автоматической группировки продукции на основе модели k-средних с мерой расстояния Махаланобиса и средневзвешенной ковариационной матрицей, рассчитанной по предварительно размеченным партиям, а также генетический алгоритм с процедурой перекрестной мутации для задачи k-средних используются в деятельности АО «Испытательный технический центр – НПО ПМ» (г. Железногорск).

Применение указанных модели и алгоритмов, разработанных в рамках диссертационного исследования соискателя ученой степени кандидата технических наук Шкабериной Гузели Шарипжановны, позволило повысить точность решения задачи разделения сборных партий электрорадиоизделий космического применения на однородные партии. Решение данной задачи, в свою очередь, способствует повышению эффективности разрушающего физического анализа электрорадиоизделий за счет гарантированного отбора изделий из каждой поступившей однородной партии.

Директор АО «ИТЦ – НПО ПМ», канд. техн. наук В.И. Орлов

ПРИЛОЖЕНИЕ Г. Свидетельство о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020663109

Программный комплекс решения задач автоматической группировки объектов с применением массивно-параллельных систем

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева» (СибГУ им. М.Ф. Решетнева) (RU)***

Авторы: ***Орлов Виктор Иванович (RU), Казаковцев Лев Александрович (RU), Рожнов Иван Павлович (RU), Шкаберина Гузель Шарипжановна (RU), Масич Игорь Сергеевич (RU)***

Заявка № **2020661738**

Дата поступления **07 октября 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***22 октября 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев