Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева»

На правах рукописи

Гудыма Михаил Николаевич

АЛГОРИТМЫ РЕШЕНИЯ СЕРИИ ЗАДАЧ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ

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 – 2017

Оглавление

ВВЕДЕНИЕ

Настоящая работа посвящена разработке и исследованию алгоритмов автоматической группировки (кластеризации), широко используемой в системах интеллектуального анализа данных – поиска в данных скрытых нетривиальных и полезных закономерностей, позволяющих получить новые знания об исследуемых данных. Особенный интерес к методам анализа данных возник в связи с развитием методов сбора, передачи информации и хранения данных, позволившим накапливать большие объемы информации. Кластеризация, основываясь на установленном отношении схожести элементов, устанавливает подмножества (кластеры), в которые группируются входные данные. Одними из простейших и эффективнейших являются методы и модели, основанные на минимизации суммарных расстояний между объектами одной группы (кластера) или между объектами кластера и его центром, который может называться также центроидом, медоидом, медианой в зависимости от постановки конкретной задачи (задачи k-средних, k-медоид или дискретная p-медианная, непрерывная p-медианная). Данные модели имеют сходство, а иногда идентичны моделям теории размещения, в частности – p-медианной модели. Задача поиска центра кластера при этом связана с решением классической задачи теории размещения – задачи Вебера. Задачи автоматической группировки данных могут накладывать специфические требования по учету расстояний в пространствах этих данных. Поэтому назрела необходимость расширить перечень используемых моделей и алгоритмов размещения с различными мерами расстояния. Иной подход представляют модели, основанные на разделении смеси распределений. Предполагается, что параметры объектов порождены некими многомерными вероятностными распределениями, параметры которых неизвестны. Требуется определить, каким из распределений порожден тот или иной объект.

Все эти задачи NP-трудны, алгоритмы полиномиальной сложности разработаны лишь для некоторых частных случаев. Для их приближенного решения разработано большое количество в основном рандомизированных

алгоритмов, позволяющих получать результаты различной точности. В частности, алгоритмы метода жадных эвристик позволяют получать результаты, превосходящие по точности (выраженной значением целевой функции) и стабильности (воспроизводимости) результаты других известных методов при объеме данных до сотен тысяч векторов данных большой размерности.

Большинство алгоритмов для данных задач требуют указания числа групп (кластеров). Определение этого числа является отдельной задачей, решаемой с использованием различных критериев, таких как критерий силуэта, Хартигана, Акаике и т.д. Существуют методы сравниетельно невысокой точности, такие как X-means, совмещающие решение задачи определения числа групп с задачей собственно автоматической группировки. Эти методы требуют использования строго оговоренного критерия для определения числа групп.

Другим подходом является решение серии задач. Под серией понимается множество задач различающихся только числом групп (кластеров), на которые разбиваются объекты. Результаты решения этих задач в дальнейшем могут быть проанализированы с использованием любых критериев или же могут рассматриваться как Парето-оптимальное решение двухкритериальной задачи: критерий минимального суммарного расстояния и критерий числа групп. Примером такой задачи с неизвестным числом кластеров (групп) является задача разделения сборной партии электрорадиоизделий космического применения на однородные производственные партии, изготовленные из единой партии сырья по данным сотен неразрушающих тестовых испытаний в специализированных тестовых центрах. На выходе имеем массив результатов этих измерений, который, кроме основной задачи отсева некачественных изделий, может быть задействован для решения задачи выявления однородных партий изделий, число которых неизвестно. Данная задача в рамках производственного процесса тестирования должна быть решена сравнительно быстро, при этом результат должен быть таким, чтобы его трудно было улучшить любым известным методом без значительных вычислительных затрат.

Степень разработанности темы. Стоит отметить, что довольно долго

теория размещения и кластерный анализ развивались параллельно, и их методы схожи. В 1909 г. А.Вебер исследовал задачу о нахождении центра тяжести для взвешенных точек (задача Вебера или 1-медианная задача), являющуюся развитием задачи П. Ферма XVII века. В 1937 г. А.Вайсфельд предложил решение этой задачи градиентным спуском. В середине прошлого века С.Л.Хакими рассматривал задачу нахождения медианы графа, и определил возможность дискретизации непрерывной задачи Вебера. Позднее Хакими обобщил эту задачу до нахождения *p* медиан графа с минимальной суммой взвешенных расстояний.

Одной из наиболее популярных моделей кластерного анализа, модель k-средних, была предложена в 1957 г. Г.Штейнгаузом, алгоритм разработан С.Ллойдом тогда же, однако его работа была опубликована только в 1982 г. Она сходна с p-медианной задачей не только по своей постановке, но и по используемым традиционным подходам к ней – ALA-процедура (Л.Купер, 1964) и процедура k-средних построены по одной схеме. Подобную связь можно заметить и между агломеративными эвристическими процедурами, применяемыми в кластерном анализе, к примеру, методом информационного бутылочного горлышка (IBC – Information Bottleneck Clustering) и методом размещения складов, предложенным А.Куном и М.Хамбургером в 1963 г.

Среди ученых, в чьих трудах получили развитие теория размещения и автоматическая группировка объектов, в первую очередь необходимо упомянуть Ц.Дрезнера, С.Л.Хакими и Г.Весоловски. Весомый вклад внесли также В.А.Трубин, Н.Младенович, Дж.Бримберг и Р.Ф.Лав. Методы и модели теории размещения получили наиболее полное отражение в монографиях под редакцией Ц.Дрезнера и Х.Хамахера (2004), Р.Фарахани и М.Хекматфара (2009). В СССР исследованием вопроса размещения предприятий занимались В.Р.Хачатуров и В.П.Черенин. В Институте математики им. С.Л. Соболева СО РАН при разработке моделей стандартизации и унификации в качестве основы использовались модели размещения. Работы Э.Х.Гимади, В.Л.Береснева, А.А.Колоколова, а позже Ю.А.Кочетова, А.В.Еремеева, Г.Г.Забудского и др. заложили основу для разработки программно-математического аппарата решения этих задач.

В 2014 году Л.А.Казаковцевым и А.Н.Антамошкиным был предложен метод жадных эвристик [1]. Метод широко использует эволюционные подходы, большой вклад в развитие которых вносит Красноярская школа эволюционных алгоритмов (Е.С.Семенкин и др.).

Настоящая работа посвящена разработке алгоритмов одновременного решения серий задач автоматической группировки объектов с различным числом групп (кластеров), позволяющих получать наиболее точные (по значению целевой функции) результаты по сравнению с другими известными методами.

Основная **идея** настоящей диссертации состоит в разработке генетического алгоритма метода жадных эвристик для задач автоматической группировки с заранее неизвестным числом групп (кластеров), который одновременно оперирует смешанной популяцией, состоящей из решений с различным числом групп (кластеров). Эффективность алгоритма повышается за счет того, что решения, являющиеся локальными оптимумами задач с различным числом групп (кластеров), совместно участвуют в едином процессе рекомбинации и создания новых решений.

Объектом диссертационного исследования являются задачи автоматической группировки многомерных данных с неизвестным числом групп (кластеров), **предметом исследования** – алгоритмы их решения.

Целью исследования является повышение эффективности систем автоматической группировки объектов, к которым предъявляются высокие требования по точности результата, выраженного значением целевой функции.

В процессе достижения поставленной цели решались следующие **задачи:**

1) анализ проблем, возникающих при применении методов кластеризации, основанных на моделях теории размещения и моделях разделения смеси распределений, при заранее неизвестном числе кластеров (групп);

2) разработка алгоритмов одновременного решения серии задач автоматической группировки (кластеризации) данных большой размерности (до сотен измерений) и большого объема (до десятков тысяч векторов данных) на основе моделей k-средних, k-медоид и k-медиан, различающихся только числом

групп (кластеров), на которые разбиваются объекты, при заранее известном максимальном числе групп;

3) разработка алгоритма одновременного решения серии задач нечеткой кластеризации данных большой размерности на основе модели разделения смеси вероятностных распределений, различающихся только числом групп (кластеров, распределений), на которые разбиваются объекты, при заранее известном максимальном числе распределений;

4) разработка эффективных алгоритмов решения задач Вебера (1-медианных) с нестандартными мерами расстояния, востребованными при решении практических p-медианных задач.

Методы исследования. Методологической базой явились работы по методам кластеризации, в частности – по методу жадных эвристик для задач автоматической группировки объектов. Использован математический аппарат теории размещения, методы теории оптимизации, в частности – эволюционные методы оптимизации, а также методы системного анализа, исследования операций, аналитической геометрии и теории вероятностей.

Новые научные результаты и положения, выносимые на защиту:

1. Предложен новый генетический алгоритм с вещественным алфавитом для одновременного решения серий непрерывных и дискретных задач автоматической группировки объектов на основе моделей k-средних, k-медоид, k-медиан с различными мерами расстояния, основанный на комбинации особой модификации жадных агломеративных эвристических процедур, эволюционных методов оптимизации, методов локального поиска и методов решения задачи поиска центра группы (задачи Вебера с соответствующей мерой расстояния). Разработанный алгоритм обеспечивает лучшие (по значению целевой функции) результаты для серии задач с числом групп (кластеров) от 2 до заданного значения p_{max} за приемлемое время для задач с большим объемом входных данных в сравнении с известными методами. Алгоритм позволяет одновременно решать серию задач, благодаря чему повышается эффективность систем автоматической группировки объектов.

2. Впервые предложен генетический алгоритм с вещественным алфавитом для одновременного решения серий задач автоматической группировки объектов на основе моделей разделения смеси вероятностных распределений, основанный на комбинации особой модификации жадных агломеративных эвристических процедур и EM-алгоритма. Разработанный алгоритм обеспечивает результаты для серии задач с числом групп (кластеров) от 2 до заданного значения p_{max}, не уступающие результатам известных методов, позволяя, в отличие от них, одновременно решать за приемлемое время серию задач с большим объемом входных данных, за счет чего повышается эффективность работы систем автоматической группировки объектов.

3. Предложен новый алгоритм решения задачи Вебера с допустимыми зонами, ограниченными окружностями, который является составной частью алгоритма решения задачи автоматической группировки с особой мерой расстояния, ограниченной снизу. Экспериментально показано, что алгоритм позволяет получать более точные решения в сравнении с известными алгоритмами.

Значение для теории. Результаты исследования дополняют арсенал эффективных эвристических методов решения NP-трудных задач автоматической группировки и размещения с широким кругом используемых мер расстояний. Принцип скрещивания в единой популяции генетического алгоритма особей, представляющих собой решения различных задач, различающихся единственным параметром – числом групп, создает основу для синтеза новых эффективных алгоритмов.

Практическая ценность методов решения задач автоматической группировки и задач размещения обусловлена широким диапазоном их применения как в задачах кластерного анализа или автоматической группировки данных, так и непосредственно в практических задачах группировки физических объектов или оптимального размещения в пространстве. Разработанный метод позволяет повысить эффективность алгоритмов за счет одновременного решения за ограниченное время сразу серии задач с большим объемом входных данных.

Практическая реализация результатов: Программная реализация алгоритма решения серии задач автоматической группировки с прямоугольной метрикой позволила повысить эффективность СППР автоматической классификации электрорадиоизделий по производственным партиям ОАО ИТЦ – НПО ПМ (г.Железногорск) за счет возможности работы с данной СППР в интерактивном режиме и встроить классификацию электрорадиоизделий в производственный процесс проведения испытаний.

Апробация работы. Основные положения и результаты диссертационной работы докладывались и обсуждались на международных конференциях. В их числе: XIII Международная научно-практическая конференция «Перспективы развития информационных технологий» (2013 г., г. Новосибирск), XV Международная научно-практическая конференция «Актуальные вопросы науки» (2014 г., г. Москва), XVIII Международная научная конференция «Решетневские чтения» (2014 г., г. Красноярск), Международная научная конференция «Проблемы современной аграрной науки» (2013 г., г. Красноярск), 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, (2014 г., г. Санкт-Петербург), XX Международная научная конференция «Решетневские чтения» (2016 г., г. Красноярск). Работа в целом обсуждалась на научно-техническом семинаре «Электронная компонентная база космических систем» (2016 г., г. Железногорск).

Публикации. Основные теоретические и практические результаты диссертации опубликованы в 17 статьях (также имеются 3 свидетельства о государственной регистрации программ для ЭВМ), среди которых 7 работ в ведущих рецензируемых изданиях, рекомендуемых в действующем перечне ВАК, 2 – в международных изданиях, проиндексированных в системах цитирования Scopus.

Структура и объем диссертации. Диссертация состоит из введения, 3 глав и заключения. Она изложена на 201 листе машинописного текста, содержит список литературы из 289 наименований.

ГЛАВА 1. РАЗВИТИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ЗАДАЧ РАЗМЕЩЕНИЯ И АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ ОБЪЕКТОВ

1.1 Задачи автоматической группировки объектов и сферы их применения

Совершенствование технологий записи и хранения данных повлекло за собой необходимость переработки большого количества информации. Это, а также проникновение в сферу обработки данных идей, принадлежащих к области искусственного интеллекта, дало толчок стремительному прогрессу в области обработки информации, и возникновению нового направления под названием интеллектуальный анализ данных (Data Mining), которое определяется как процесс обнаружения в сырых данных ранее неизвестных нетривиальных практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности [2, 3].

Для анализа данных было предложено множество статистических методов, таких как корреляционный анализ, регрессионный анализ, частотный анализ, дискриминантный анализ, дисперсионный анализ, корреляционный анализ, многомерное шкалирование, факторный анализ, кластерный анализ [4].

Автоматическая группировка данных, известная также как кластерный анализ, имеет своей целью выделить в исходных многомерных данных такие однородные подмножества, чтобы объекты внутри групп были в известном смысле похожи друг на друга, а объекты из разных групп не похожи. Под «похожестью» (мерой сходства) понимается близость объектов в многомерном пространстве признаков и задача сводится к выделению в этом пространстве естественных скоплений объектов, которые считаются однородными группами [5].

Цель решения задачи автоматической группировки состоит в разработке алгоритма и/или автоматизированной системы, способных обнаруживать эти естественные скопления в не отмаркированных предварительно данных.

Более точно задачу можно определить следующим образом. Имеется

выборка $X_l = \{x_1, ..., x_l\} \subset X$ и функция расстояния между объектами $p(x, x')$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике p, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X_l$ приписывается метка (номер) кластера y_i.

Алгоритм кластеризации — это функция $a: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие метку кластера $y \in Y$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимально число кластеров, с точки зрения того или иного критерия качества кластеризации.

Решение задачи кластеризации принципиально неоднозначно по следующим причинам [6]:

- Наилучшего критерия качества кластеризации не существует и разбиение может различаться от случая к случаю (рис.1.1)
- Число кластеров, как правило, неизвестно заранее
- Результат существенно зависит от выбранной метрики p

Идеальную группу можно определить как компактный и изолированный ряд точек, которыми в некотором пространстве характеристик представлены объекты или элементы данных. В действительности группа – понятие субъективное, и ее определение может потребовать знаний в соответствующей области. В случае двух- и трехмерных данных человек вполне способен выделить обособленные группы, но в практических задачах число измерений данных может быть очень велико. Например, в прикладной задаче группировки электрорадиоизделий, которая, в частности, рассматривается в этой работе, размерность данных может варьироваться от нескольких десятков до тысяч измерений [7].

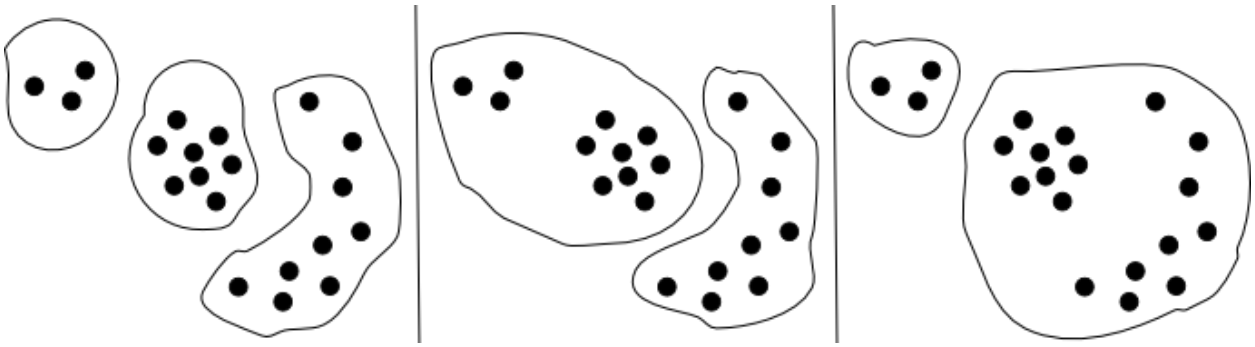


Рисунок 1.1 – группы (кластеры) различной формы

Многие дисциплины предполагают многомерный анализ данных и в любой из них могут возникать задачи автоматической группировки. К ним могут быть сведены, к примеру, такие задачи, как задачи распознавания печатного [8] и рукописного [9] текста, а также сегментация изображения для компьютерного зрения [10, 11, 12, 13], категоризация документов для обеспечения быстрого доступа и поиска [14, 15, 16, 17], работа с группами клиентов в CRM-системах [18, 19], географическая группировка клиентов точек сервисного обслуживания [20].

1.2 Пример актуальной практической задачи группировки

В России, в отличие от США и стран Западной Европы, отсутствуют специализированные производства электронной компонентной базы (ЭКБ) – электрорадиоизделий (ЭРИ) для космической отрасли, поэтому ЭРИ общего военного (неспециализированного) применения [21, 22] категорий качества «ВП» и «ОС» («ОСМ») должны подвергаться демонстрации возможности использования в аппаратуре космических аппаратов (КА). ЭКБ иностранного производства, которая находит все более широкое распространение в аппаратуре КА, также должна подвергаться квалификации по условиям применения и уровню качества, поскольку на настоящее время не существует никаких документов о гармонизации систем качества отечественной и импортной ЭКБ.

Поэтому демонстрация возможности использования ЭРИ в аппаратуре КА в

течение длительного времени (на основе разработки принципов и правил) включает разработку методологии обеспечения качества и работоспособности ЭКБ при воздействии факторов космического пространства.

Для исключения попадания в бортовую аппаратуру КА с длительными сроками активного существования потенциально ненадежных ЭРИ в последние годы внедряется новый принцип комплектования аппаратуры через специализированные испытательные технические центры [23, 24] с проведением операций сплошного входного контроля ЭРИ, дополнительных отбраковочных испытаний (ДОИ), диагностического неразрушающего контроля (ДНК) с применением выборочного разрушающего физического анализа (РФА). Задачей ДОИ и ДНК ЭРИ является, по существу, индивидуальная отбраковка элементов, имеющих скрытые дефекты изготовления. РФА проводится с целью определения соответствия образцов ЭРИ требованиям конструкции и технологического процесса изготовления и выявления нарушений этих требований.

Таким образом, все проводимые над ЭРИ испытания можно разделить на две группы:

1) сплошные испытания для всей партии элементов – ДОИ, ДНК.

2) выборочные испытания для нескольких элементов из партии – РФА.

Многие из эксплуатируемых до 2000 года КА, разработанных и изготовленных АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева», в первые дни, месяцы эксплуатации имели замечания по качеству функционирования: сбои, перерывы в связи, отказы, значительная часть которых по результатам анализа возникала из-за отказов ЭРИ. И только на эксплуатируемом с 18 апреля 2000 года КА «Sesat» не выявлено существенных замечаний к ЭРИ в течение более 15 лет эксплуатации. Одной из главных причин, по мнению многих специалистов, является то, что впервые в практике все 100% ЭРИ, комплектующих бортовую аппаратуру КА «Sesat», прошли ДОИ, ДНК и РФА [24]. То, что аналогичные результаты, а именно – отсутствие или существенное снижение количества замечаний к работе ЭРИ, прошедших данный набор испытаний (ДОИ+ДНК+РФА) [25, 26], позволяет сделать вывод о

состоятельности подхода к испытаниям, в состав которого входят именно эти три основных компонента.

Важное значение имеет разработка методов прогнозирования и обеспечения работоспособности ЭРИ при неблагоприятных внешних воздействиях. Одно из центральных мест при этом занимают методы обеспечения устойчивости к тепловым и радиационным нагрузкам [27, 28, 29].

Вопросы обеспечения радиационной стойкости (РС) БА изложены в обширной литературе, например в [30, 31, 32, 33], но она, в основном, посвящена применению ЭРИ в предположении, что стойкость любого ЭРИ из производственной партии известна и одинакова. На самом деле РС ЭРИ внутри производственной партии различна и зависит от содержащихся в каждом ЭРИ внутренних дефектов (дислокации, неконтролируемых примесей, других точечных дефектов) [34].

Собственно выявление наиболее существенных из таких дефектов в партиях изделий является целью проведения РФА. При этом распространять результаты проведенного РФА на всю поступившую партию изделий необходимо с большой осторожностью. Для этого нужно, как минимум, быть уверенными в том, что мы имеем дело действительно с единой партией ЭРИ, изготовленной из единой партии сырья. Поэтому выявление истинных производственных партий из предположительно сборных партий ЭРИ является одним из важнейших мероприятий при проведении испытаний.

Испытания состоят в контроле вольт-амперных характеристик входных и выходных цепей микросхем. Последствия электрических воздействий сводятся в таблицу и служат данными для анализа. Различия в полученных результатах означают различия в эксплуатационных характеристиках изделий и их принадлежность к разным производственным партиям. Весь массив данных диагностических испытаний используется для выявления в поступившей партии групп компонентов с однородными характеристиками. Разброс значений каждого параметра слишком узок для того, чтобы по нему отнести изделие к той или иной партии, однако по совокупности параметров это возможно.

1.3 Взаимосвязь задач автоматической группировки объектов и теории размещения

Приведенное выше определение задачи автоматической группировки объектов предполагает наличие некой меры подобия (сходства), либо наоборот — меры различия. Мера различия по сути является расстоянием между объектами в некотором дискретном либо непрерывном пространстве характеристик. Задача группировки оперирует положениями объектов в пространстве и расстояниями между ними, поэтому вполне очевидна ее связь с задачами теории размещения, которые в большинстве работ определяются как задачи, основными параметрами которых являются положения каких-либо объектов в пространстве и расстояния между ними [35, 36, 37, 38].

Задачи размещения можно классифицировать по зависимости целевой функции от расстояний между новыми и существующими объектами. Существуют два типа моделей – непрерывные и сетевые (дискретные) [39].

В непрерывных моделях новые объекты могут быть представлены, к примеру, как точки на эвклидовой плоскости или в трехмерном пространстве, и могут быть размещены где угодно в некоторой допустимой области. Также учитывается метрика пространства размещения. В сетевых моделях существующие объекты представлены узлами сети (вершинами графа), а новые объекты могут быть размещены только на вершинах или на вершинах и ребрах графа. Функцией расстояния служит кратчайший путь в сети.

Нужно отметить, что теория размещения и кластерный анализ используют одинаковые либо очень схожие методы, хотя долгое время развивались параллельно. Так, например, ALA-процедура (Alternating Location-Allocation – чередующееся распределение-размещение) для решения p-медианной задачи – одна из основных задач теории размещения [40] и процедура k-средних [41] – весьма распространенный алгоритм в кластерном анализе – построены по одной схеме.

Первые попытки сформулировать и решить подобные задачи были предприняты достаточно давно. Еще в XVII веке П. Ферма рассматривал простейший случай размещения. Для трех точек на плоскости следовало найти четвертую, называемую медианой, такую, чтобы сумма расстояний от нее до первых трех точек была минимальной [42]. Чуть позже, эта задача была частично решена Э.Торричелли [43] и Б.Кавальери [44]. Впоследствии, этой проблемой занимались и другие математики – Ф.Симпсон, Т.Хайнен и др.

В 1909 году немецкий экономист и социолог А.Вебер опубликовал исследование [45], посвященное влиянию основных факторов производства на размещение предприятий с целью минимизации издержек. В своей работе он исследовал более общую задачу о нахождении центра тяжести для трех взвешенных точек. Эта задача, упоминающаяся впоследствии как задача Вебера (Ферма-Вебера) или 1-медианная задача [46] послужила исходной точкой развития теории размещения, и ее решение является составной частью многих методов решения задач автоматической группировки.

В 1934 году В. Ярник и О. Кесслер сформулировали обобщение задачи Ферма, заменив три точки на произвольное конечное число [47].

В 1937 году А.Вайсфельд в своей работе [48] доказал теорему, сформулированную Штурмом [49] и в одном из доказательств вывел последовательность, которая сходилась к оптимальному решению задачи Вебера. Надо заметить, что по сути это являлось вариантом алгоритма градиентного спуска [50], хотя само понятие «градиентный спуск» появилось гораздо позже. Процедура Вайсфельда и ее модификации [51] по-прежнему является широко распространенным методом при решении задач размещения.

В начале 60-х американский математик С.Л.Хакими рассматривал задачу нахождения медианы графа и его вариант задачи для абсолютной медианы был схож с задачей Вебера. Хакими определил абсолютную медиану как точку на графе, сумма взвешенных расстояний от которой до вершин графа минимально. Он принял условием, что абсолютная медиана может располагаться в любом месте на ребрах графа, однако доказал, что оптимальным положением для нее

всегда является одна из вершин [52]. Таким образом, была определена возможность дискретизации непрерывной задачи Вебера.

В последующей работе [53] Хакими обобщил эту задачу до нахождения *p* медиан графа с минимальной суммой взвешенных расстояний и показал, что всегда есть множество вершин *p*, удовлетворяющее условию. Такое множество вершин называется p-медианой графа.

В практических задачах используется множество метрик и мер рсстояния. Наиболее обширное их описание дано в «Энциклопедии расстояний» и других работах М.Дезы и Е.Дезы [54, 55, 56]. Наиболее популярны модели с метриками, основанными на l_p-нормах Минковского:

$$L(X,Y) = l_p((x_1,x_2,\ldots,x_d),(y_1,y_2,\ldots,y_d)) = \left(\sum_{i=1}^{d}|x_i - y_i|^p\right)^{1/p}, p \geq 1$$

и ее частных случаях – евклидовой l_2 (при p=2), прямоугольной или манхэттенской l_1 (p=1), чебышевской l_∞ ($p = +\infty$), а также с квадратичными евклидовыми расстояниями (такая мера расстояния метрикой не является):

$$l_2^2(X,Y) = \sum_{i=1}^{d}(x_i - y_i)^2 .$$

Квадратичная евклидова мера весьма популярна из-за того, что задача Вебера с данной мерой решается крайне просто. Решением является точка

$$X^* = (x_1^*,\ldots,x_d^*);\ \ x_r^* = \frac{\sum_{i=1}^{N} w_i a_{i,r}}{\sum_{i=1}^{N} w_i}\ \ \forall r = \overline{1,d}.$$

В теории размещения изучены задачи в евклидовой метрике (l_2), а также в метриках l_1 (прямоугольной или манхэттенской) и l_∞ (чебышевской). Для них созданы универсальные алгоритмы, в том числе – на основе процедуры Вайсфельда, исследована их вычислительная сложность [57, 58]. В реальных условиях в пространстве размещения обычно есть преграды и запрещенные зоны, это сложно учесть при использовании евклидовой или прямоугольной метрики, они дают довольно грубое приближение [59, 60]. Задача Вебера с преградами и запрещенными зонами представляет собой гораздо более сложную задачу оптимизации. Многие авторы предлагали различные алгоритмы для специальных

случаев [61, 62, 63], применение специальных метрик, в частности, рассмотрено в Главе 3.

1.4 Задача автоматической группировки объектов на основе непрерывной задачи размещения

Как уже отмечалось выше, p-медианная задача является одной из классических моделей теории размещения. Общей целью непрерывной задачи размещения [35] является определение местоположения одной или нескольких точек (называемых центрами, центроидами, медоидами и т.д. – в зависимости от того, как конкретно поставлена задача) в непрерывном пространстве – континууме возможных местонахождений искомых точек. Задача p-медоид [64, 65], которую в литературе также называют задачей k-медоид или дискретной p-медианной задачей [66], относится к задачам промежуточного класса, которые по факту дискретны, но при этом оперируют понятиями, характерными для непрерывных задач.

Основными параметрами задач размещения являются координаты объектов и расстояния между ними [67, 68, 36]. В качестве примера непрерывной задачи размещения можно привести размещение складов [36] таким образом, чтобы расстояние от каждого потребителя до ближайшего склада было минимально, размещение узлов компьютерных и коммуникационных сетей, базовых станций беспроводных сетей [69, 70, 71]. Многие задачи автоматической группировки [72, 73, 74] можно рассматривать как задачи размещения [75] с квадратичными евклидовыми расстояниями [72, 76], евклидовыми [75] или другими метриками и мерами расстояния [77] и наоборот.

Целью непрерывной p-медианной задачи [67] является отыскание точек (центров, центроидов, медиан), таких, чтобы сумма взвешенных расстояний (которым приписан некий вес, пропорционально степени важности) от N известных точек, называемых векторами данных или точками-потребителями (в зависимости от постановки и предметной области задачи), до ближайшего из p

центров достигала минимума.

Непрерывные задачи размещения с евклидовой, манхэттенской (прямоугольной), чебышёвской метриками хорошо изучены (все эти метрики являются частными случаями метрик, основанных на l_p-нормах Минковского [56]), предложено множество алгоритмов для решения задачи Вебера для этих метрик [78]. В частности, известная процедура Вайсфельда [48] была обобщена для метрик, основанных на нормах Минковского.

В традиционном понимании, в случае евклидовой метрики $L(X_j,A_i)=\sqrt{\sum_{k=1}^{d}(x_{j,k}-a_{i,k})^2}$ мы имеем собственно p-медианную задачу. Здесь $X_j=(x_{j,1},...,x_{j,k})\ \ \forall j=\overline{1,p}$, $A_i=(a_{i,1},...,a_{i,k})\ \ \forall i=\overline{1,N}$. В случае квадратичной евклидовой метрики $L(X_j,A_i)=\sum_{k=1}^{d}(x_{j,k}-a_{i,k})^2$ при $w_i=1\forall i=\overline{1,N}$ мы имеем задачу k-средних.

Задачу k-средних можно считать частным случаем p-медианной задачи с квадратичными евклидовыми расстояниями. Многие авторы отмечают общность задачи k-средних и непрерывной p-медианной задачи [79, 80, 36, 81, 82, 83]. Подобную связь можно заметить и между агломеративными эвристическими процедурами, применяемыми в кластерном анализе, к примеру, методом информационного бутылочного горлышка (IBC – Information Bottleneck Clustering) и методом размещения складов [84], предложенным А.Куном и М.Хамбургером в 1963 г.

Задача Вебера [45, 67, 36] является простейшим случаем непрерывной задачи (в случае с p=1) и состоит в поиске такой точки, чтобы сумма взвешенных евклидовых расстояний от этой точки до заданных точек достигала минимума:

$$arg\min_{X\in\Re^2} F(X)=\sum_{i=1}^{N} w_i L(X,A_i).$$

Здесь $L()$ – функция расстояния (норма, метрика или иная мера – произвольная функция, для которой, возможно, справедливы условия симметрии и тождества: L(X,Y)=L(Y,X), L(X,X)=0), евклидова в случае классической задачи Вебера.

Для решения этой задачи (поиска центра множества точек) мы можем

использовать процедуру Вайсфельда [48] или ее улучшенные модификации [85, 51]. Случаи с метрикой Чебышева и манхэттенской метрикой рассмотрены в [60, 86, 87]. Для решения задачи Вебера с иными метриками и мерами расстояния [88, 89, 90] разработано множество алгоритмов. Такой задаче, в частности, посвящена Глава 3.

Собственно p-медианная задача является одним из возможных обобщений [36, 91] задачи Вебера:

$$arg\,min F(X_1,\ldots,X_p) = \sum_{i=1}^{N} w_i \min_{j\in\{1,p\}} L(X_j, A_i).$$

Здесь $\{A_i \mid i=\overline{1,N}\}$ – набор выбранных точек (векторов данных или точек-потребителей, в случае если задача размещения носит «геометрический» характер), $\{X_j \mid j=\overline{1,p}\}$ это набор новых размещаемых объектов, $L()$ – функция расстояния в непрерывном или дискретном пространстве [52, 36], w_i – весовой коэффициент i-й заданной точки. В простейшем случае, $L()$ вычисляется как евклидово расстояние. В этом случае на каждой итерации метода итеративного чередующегося расположения-распределения (ALA – alternating location-allocation – чередующееся размещение-распределение) [40, 92] процедура Вайсфельда осуществляется до p раз.

Задача k-средних [41, 73] является наиболее простой и популярной моделью автоматической группировки [93, 94, 73]. С допущением, о котором говорилось выше, ее можно сформулировать как p-медианную задачу, где $w_i = 1 \forall i = \overline{1,N}$ и $L()$ – квадратичное евклидово расстояние l_2^2: $L(X,Y) = \sum_{i=1}^{d} (x_1 - y_1)^2$, где $X = (x_1,\ldots,x_d) \in \Re^d$, $Y = (y_1,\ldots,y_d) \in \Re^d$.

При квадратичной евклидовой (l_2^2) мере расстояния решением задачи Вебера является точка (центроид) [36]:

$$x = \frac{\sum_{i=1}^{N} w_i a_i}{\sum_{i=1}^{N} w_i}$$

В данном случае мы считаем, что $X = (x_1,\ldots,x_d) \in \Re^d$, $A_i = (a_{1,d},\ldots,a_{i,d}) \forall i = \overline{1,N}$.

Целевая функция p-медианных задач с евклидовой (l_2) метрикой,

квадратичными евклидовыми расстояниями (l_2^2) или другими метриками [56] не является выпуклой [92] и они относятся к задачам общей оптимизации. Результат работы ALA-процедуры и ее аналогов зависит от исходных данных, они могут найти один локальный оптимум. Кроме того, метод полного или частичного перебора для больших задач использовать невозможно, из-за того что такие задачи глобальной оптимизации NP-трудны [95, 96, 57] как в непрерывном, так и в дискретном случае [97, 98, 99]. В теории вычислительной сложности NP-трудной называется задача, которая по крайней мере так же сложна, как любая задача в классе NP. Более точно, задача является NP-трудной, если другая задача, принадлежащая классу NP, полиномиально сводится к ней [100, 101]. NP-класс можно определить как класс задач, которые поддаются решению на недетерминированной машине Тьюринга [102, 103, 104] за время, не большее, чем полином от длины строки, представляющей входные данные.

Идея наиболее популярного алгоритма для задачи k–средних была предложена в 1956 г. Штейнгаузом [105], а алгоритм был разработан Ллойдом год спустя, хотя его работа [41] увидела свет только в 1982 г. После работы Маккуина [73] алгоритм стал известен как стандартная процедура k-средних или алгоритм Ллойда. Впоследствии было опубликовано много работ [106, 107, 108], где были описаны более быстрые варианты этого алгоритма для наборов и непрерывных потоков данных. ALA-процедура (а также процедура k-средних) являются алгоритмами, находящими локальный минимум, последовательно улучшая известное решение. При этом в строгом смысле алгоритмами локального поиска они не являются, поскольку область поиска нового решения не обязательно лежит в ε-окрестности имеющегося решения.

Многие авторы предлагают упрощать задачу, выбирая часть начального набора данных (случайным или детерминированным образом) и используя результаты работы алгоритма на частичном наборе в качестве начального решения ALA-процедуры на полном наборе данных [65, 109, 110, 111, 112, 113].

На сегодняшний день разработано множество эвристических методов [114] задания начальных центров ALA-процедуры, по большей части они относятся к

эволюционным методам или методам случайного поиска, как например, процедура k-means++ [115].

В попытке улчшить результаты локального поиска при решении задач автоматической группировки в непрерывном пространстве многие авторы [116, 117, 118, 119] используют генетические алгоритмы (ГА) и иные эволюционные подходы. В ходе своей работы многие из таких эволюционных алгоритмов проводят рекомбинацию начального решения, полученного ALA-процедурой.

В 2003 году Альп, Эркут и Дрезнер [120] предложили точный и быстрый алгоритм для p-медианной задачи на сети с особым способом рекомбинации, названным агломеративной эвристикой. Неэма в своей работе [119] адаптировал этот алгоритм для непрерывных задач, предложив использовать для создания начальных решений для ALA-процедуры жадную агломеративную эвристику наряду с другими рекомбинационными эвристиками. Надо сказать, что при больших p алгоритм работает очень медленно из-за того, что на каждой итерации многократно выполняется генерация начальных решений для процедуры локального поиска. Кроме того, Лим и Сю [121] в 2003 году, а также Шен и Лиу [122] в 2006 году предложили генетические алгоритмы, основанные на рекомбинации подмножеств центров фиксированной мощности. Эти алгоритмы менее точны, однако работают быстрее.

В работе [123] предложена идея жадной агломеративной эвристики, состоящей из трёх вложенных циклов (см. Рисунок 1.2) – итерации стратегии глобального поиска, собственно жадная эвристическая процедура и анализ результатов. После объединения двух «родительских» особей (решений) образуется новое промежуточное решение (решение-кандидат), в общем случае – недопустимое. Из промежуточного решения по одному исключаются элементы решения (центры, центроиды, медоиды), до тех пор пока количество элементов решения (количество групп, на которые разбивается множество векторов данных) не снизится до p и решение не станет допустимым. Исключению на каждом шаге алгоритма подвергаются элементы, дающие наименьший прирост целевой функции. В случае использования алгоритма для непрерывной задачи, он создает

начальное решение для ALA-процедуры, которая выполняется в ходе каждой итерации, чтобы оценить результат исключения каждого элемента из промежуточного решения.

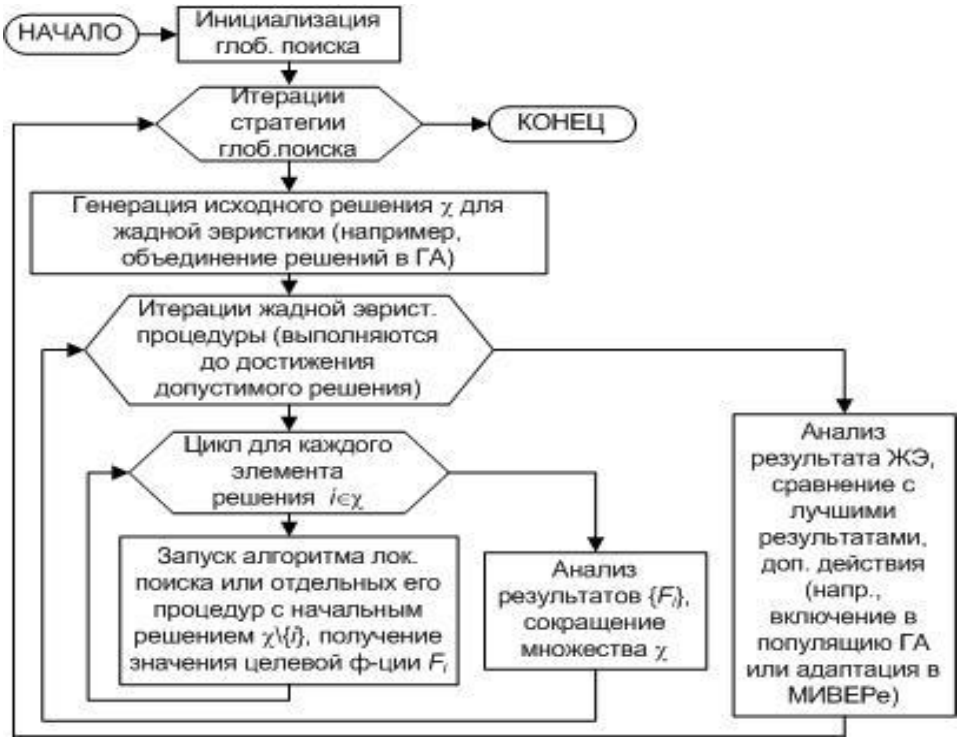


Рисунок 1.2. Общая схема алгоритма метода жадных эвристик [123].

Также на основе идеи алгоритма, предложенного Альпом и др. [120] для задач на сети (графе), предложен новый ГА, использующий вещественный алфавит (реализованный на ЭВМ как алфавит с плавающей запятой). В оригинальном алгоритме Альпа в «хромосомах» (промежуточных решениях) генетического алгоритма используется целочисленный алфавит (числами представлены номера вершин сети). В предложенном алгоритме кодирование элементов решения осуществляется вещественными числами. В качестве элементов «хромосом» данного ГА выступают координаты центров или центроидов промежуточных решений, измененные в ходе ALA-процедуры, которые исключаются до получения допустимого решения. Использование мутаций не предусмотрено. Сочетание ALA-процедуры и жадной агломеративной эвристики позволяет алгоритму получить более точные результаты.

В [124] предложена простая модификация жадной эвристики для решения

серии задач: после достижения требуемого числа кластеров p процесс исключения кластеров продолжается, и алгоритм продолжает фиксировать наилучшие известные значения целевой функции для каждого числа кластеров вплоть до 2 кластеров. Таким образом, можно получить решения серии задач с $p = \overline{2, p_{max}}$. При этом максимальное предполагаемое значение числа кластеров p_{max} все же должно быть известно.

Таблица 1.1 Сравнение алгоритмов на различных задачах группировки [124].

Набор данных, его параметры	p и мера расст-я	Алгоритм	Средний результат
Отбрак. испытания микросхемы 1526ТЛ1, N=1234, p_{max}=20	p=14, l_2^2	ALA ГА ФП+ALA ГА ЖЭС+ALA	150,124869801 149,954679652 149,78736565*
	p=10, l_2^2	ALA ГА ФП+ALA ГА ЖЭС+ALA	198,375350991 198,377650812 198,359747028*
	p=6, l_2^2	ALA ГА ФП+ALA ГА ЖЭС+ALA	362,70401636* 362,70401636* 362,704051312
UCI Mopsi Joensuu, N=6014, p_{max}=20	p=10, l_2	ALA ГА ФП+ALA ГА ЖЭС+ALA	359,680203232 359,545250068 359,410460803*
	p=4, l_2	ALA ГА ФП+ALA ГА ЖЭС+ALA	359,680203232 596,82520843* 596,825283111
BIRCH-3, N=100000, p_{max}=110	p=100, l_2^2	ALA ГА ФП+ALA ГА ЖЭС+ALA	$3{,}7513245 \cdot 10^{13}$ $3{,}7711179 \cdot 10^{13}$ $3{,}740432 \cdot 10^{13}$*
	p=20, l_2^2	ALA ГА ФП+ALA ГА ЖЭС+ALA	$3{,}305141 \cdot 10^{14}$ $3{,}303278 \cdot 10^{14}$* $3{,}3049972 \cdot 10^{14}$

Примечание: ALA – мультистарт процедуры k-средних (ALA-процедуры), ГАФП+ALA – ГА с рекомбинацией подмножеств фиксированной длины в комбинации с ALA-процедурой, ГАЖЭС+ALA – ГА с для решения серии задач. * - лучший результат.

При решении непрерывных p-медианных задач данным алгоритмом в сочетании с разными способами локального поиска были получены хорошие результаты для задач с большим числом кластеров. При снижении этого числа метод начинает давать худшие результаты в сравнении с другими методами.

Поэтому, ГА с жадной эвристикой авторы предлагают останавливать не при p=2, а раньше, примерно на значении $p=p_{max}/3$, а остальные решения получать, например, новым запуском алгоритма с меньшим значением p_{max}, либо модифицировать алгоритм так, чтобы обойти это ограничение.

Мы выдвинули гипотезу, что это ограничение порождено низкой вариативностью популяции (жадная эвристика, стартуя с достаточно большим фиксированным числом кластеров $2p_{max}$, дает очень близкие друг к другу результаты для небольшого числа кластеров практически независимо от начального множества центров/центроидов/медоидов). Поскольку генетические алгоритмы метода жадных эвристик не используют операторы мутации, без радикального изменения концепции, мы применили новые подходы в рамках метода жадных эвристик.

Далее в Главе 2 рассматривается наша модификация алгоритма с жадной эвристикой [125] в соответствии с выдвинутой гипотезой. Для модификации были использованы два подхода:

1. Динамический размер популяции.

2. Гетерогенная популяция – популяция, содержащая в своем составе решения с различным числом p.

1.5 Динамический размер популяции генетического алгоритма

Важным параметром эволюционных алгоритмов, включая генетические алгоритмы с жадной эвристикой, является размер популяции. В изначальном варианте генетического алгоритма с жадной эвристикой [120] для p-медианной задачи предлагается такой размер популяции, чтобы каждый объект являлся центром хотя бы одного кластера хотя бы в одном из решений-«особей». Такой подход приводит к формированию больших популяций, для которых формирование хотя бы второго и третьего поколений решений-«особей» требует очень продолжительного времени с учетом того, что жадная эвристика включает в себя запуск алгоритма локального поиска. В то же время, для очень больших задач

время выполнения единственной процедуры «скрещивания» может быть велико и трудно предсказуемо, и в этой связи для очень больших задач также работа не доходит даже до третьего поколения «особей». Для малых же задач такая популяция быстро вырождается – алгоритм раз за разом генерирует одни и те же решения из ограниченного множества возможных комбинаций.

Современная литература предлагает различные методы управления размером популяции.

В [126] описывается алгоритм GAVaPS (genetic algorithm with varying population size). Вводится понятие срока жизни хромосомы – количество генераций алгоритма, на протяжении которых хромосома остается в популяции. Срок жизни может быть константой, а может меняться в процессе вычислений для каждой хромосомы в зависимости от значения целевой функции. Способ изменения срока жизни *i*-ой особи (*lifetime*(*i*)) может быть

1) пропорциональным (в зависимости от значения целевой функции)

$$lifetime(i) = min(MinLT + \eta \frac{fitness(i)}{AvgFit}, MaxLT)$$

2) линейным (вычисляется из соотношения индивидуального значения целевой функции и лучшего значения, найденного на текущий момент, есть недостаток – рост популяции при большом количестве хороших хромосом)

$$lifetime(i) = MinLT + 2\eta \frac{fitness(i) - AbsFitMin}{AbsFitMax - AbsFitMin}$$

3) билинейным (учитываются максимальное, минимальное и среднее значения на текущий момент, позволяет заострить разницу между результатами близкими к лучшему)

$$lifetime(i) = \begin{cases} MinLT + \eta \frac{fitness(i) - MinFit}{AvgFit - MinFit}, \\ \qquad если\ AvgFit \geq fitness(i) \\ \frac{1}{2}(MinLT + MaxLT) + \eta \frac{fitness(i) - AvgFit}{MaxFit - AvgFit}, \\ \qquad если\ AvgFit < fitness(i) \end{cases}$$

где *AvgFit*, *MaxFit* и *MinFit* среднее, максимальное и минимальное значение

целевой функции в текущей популяции, *AbsFitMax* и *AbsFitMin* максимальное и минимальное значение целевой функции, найденные на текущий момент, *MaxLT* и *MinLT* заданные максимально и минимально допустимые значения срока жизни, $\eta = \frac{1}{2}(MaxLT - MinLT)$.

В [127] билинейный подход из [126] сочетается с неслучайным скрещиванием в двух вариантах.

1)Предотвращение инцеста – для каждой особи есть родовая таблица, где записаны предки и родственные особи, при скрещивании таблицы особей сравниваются, и, если степень родства превышает определенную величину, потомство не образуется.

2) Ассортативное скрещивание – после выбора одного родителя, из некоторого набора выбирается второй, наименее похожий на первого.

С.Веллев в [128] описывает адаптивный генетический алгоритм с динамическим размером популяции, где размер популяции зависит от коллективной вероятности выживания популяции (увеличиться или уменьшиться) и персональной вероятности выживания для каждой особи.

Персональная вероятность выживания для каждой особи $\xi_i \in \xi$ определяется как $p_i = \varphi(\xi_i)/\varphi^*$, где $\varphi(\xi_i)$значение целевой функции для ξ_i, φ^*максимальное значение целевой функции в популяции.

Размер популяции *N* может уменьшиться (даже до единицы, если выживает одна особь с максимальным значением целевой функции), остаться прежним и увеличиться (до 3*N*). Желательно чтобы размер популяции не становился ниже начального и увеличение было обратно пропорционально уровню сходимости особей.

Ожидаемый размер популяции s^E может быть вычислен как сумма вероятностей выживания всех особей популяции.

Желательный размер s^D

$$s^D = s^0 c + 3N(1-c),$$

где s^0 начальный размер популяции, *N* текущий размер популяции, с уровень

сходимости $c=\varphi'/\varphi^*$, φ' среднее значение целевой функции, φ^* максимальное значение целевой функции.

Коллективная вероятность выживания p^* определена как нормализованное отношение между ожидаемым размером популяции и желаемым

$p^*= s^D/(s^D+s^E)$

В [129] описана соревновательная модель регулирования размера популяции. Популяция делится на S (от 2 до 8) групп-субпопуляций, которые в дальнейшем соревнуются между собой по значению целевой функции лучшей на протяжении ω генераций особи в группе.

Более приспособленные группы увеличиваются, менее приспособленные уменьшаются с учетом заданного коэффициента потерь. Уменьшение популяции L_i^t определяется

$$L_i^t = \begin{cases} 0 & : i = \omega^* \\ N_i^t \cdot k & : i \neq \omega^* \wedge N_i^t(1-k) \geq N_i^{min} \\ N_i^t - N_i^{min} & : i \neq \omega^* \wedge N_i^t(1-k) < N_i^{min} \end{cases}$$

где i =1,..,S, ω^* индекс самой качественной группы, $k \in [0,1]$ коэффициент потерь, N_i^t размер группы i в генерации t.

Победители соревнования аккумулируют в себе потери проигравших. Разность размера популяции задается формулой:

$$\Delta N_i^t = \begin{cases} \sum_{j=1}^{S} L_j^t : i = \omega^* \\ -L_i^t : i \neq \omega^* \end{cases}$$

Кроме того, учитывается фактор потребления, отражающий сокращающиеся ресурсы, потребляемые популяцией, что приводит к постепенному сокращению общего размера популяции.

В алгоритме PRoFIGA [130] размер популяции меняется в зависимости от изменения лучшего значения целевой функции в популяции.

а. Если значение улучшается, размер популяции растет на X пропорционально улучшению и количеству шагов, оставшихся до максимума.

$$X = incFactor \cdot (mxEvalNum - currEvalNum) \cdot \frac{mxFit_{new} - mxFit_{old}}{initMxFit}$$

где *incFactor* заданный параметр в интервале (0,1), *mxEvalNum* и *currEvalNum* максимальное количество поколений и номер текущего, $mxFit_{new}$, $mxFit_{old}$ и *initMxFit* лучшее значение целевой функции в текущем поколении, в предыдущем и в начальном.

б. Если значение не улучшается в течение некоторого количества шагов, популяция все равно растет. В статье величина роста аналогична первому шагу.

в. Иначе размер популяции падает на определенный процент (1-5%).

Рост популяции идет за счет клонирования или случайной генерации новых особей.

Алгоритм APOGA [131] это гибрид GAVaPS [126] и PRoFIGA [130], если значение не улучшается, уменьшение популяции производится за счет особей с минимальным оставшимся временем жизни. У остальных особей оставшееся время сокращается, кроме лучших.

Йен и Лю в [132] предлагают динамический многокритериальный эволюционный алгоритм (DMOEA). Многокритериальная задача оптимизации конвертируется в двухкритериальную – минимизировать ранг Парето при сохранении плотности. Для контроля выживания особи есть три индикатора – здоровье, плотность и возраст. Прирост популяции осуществляется за счет новых особей, лучших, чем родители и не попавших в запретные зоны. Сокращение происходит за счет особей с худшими значениями индикаторов и превысивших заданный предельный возраст.

В [133] предложен генетический алгоритм без параметров (parameter-less genetic algorithm). Начиная с небольшой популяции, каждый раз, когда популяция сходится, генерируется новая популяция удвоенного размера. Параллельно с этим идет гонка между популяциями – если бóльшая популяция имеет лучшее значение целевой функции, чем меньшая, то меньшая уничтожается.

Смородкина и Тауриц предложили [134] эволюционный алгоритм с жадной регуляцией размера популяции, основанный на идее из [133]. Две популяции P_i и

P_{i+1}, P_{i+1} в два раза больше P_i. На каждой итерации в каждой популяции проводится определенное количество генетических операций, такое чтобы хватило на появление нового поколения в P_{i+1}, в P_i появляется два новых поколения из-за разницы размеров. Это продолжается, пока популяция не утратит силу (либо среднее значение целевой функции P_{i+1} превысит среднее значение P_i на 5%, либо максимальное значение P_i достигнет плато). Тогда счетчик i прибавляется и генерируется новая популяция P_{i+1}.

Самоприспосабливающийся метод для контроля размера популяции через систему голосования предложен в [135]. Каждая особь в популяции имеет ген, кодирующий ее голос по размеру популяции; размер популяции определяется подсчетом голосов всех особей.

Более сложный способ контроля размера популяций применяется в коэволюционном алгоритме [136, 137]. В коэволюционном алгоритме задействовано сразу несколько генетических алгоритмов (подпопуляций) с различными настройками основных параметров, действующих параллельно и конкурирующих за ресурс, который в процессе работы алгоритма перераспределяется в пользу более эффективного.

При классической схеме коэволюционного алгоритма перераспределение ресурсов происходит путем сокращения неэффективной популяции на фиксированный процент (статический штраф) и увеличения наиболее эффективной популяции на сумму потерь сокращенных. Штрафы могут быть динамические [138], адаптивные [139], смертельные [140].

Разработан и другой метод распределения ресурсов, турнирный [141] – перераспределение ресурсов происходит в каждой паре подпопуляций, таким образом, вместо определения единственного самого эффективного алгоритма, происходит ранжирование всех подпопуляций по эффективности.

В [142] рассматривается регулирование размера популяции в зависимости от стадии работы алгоритма. Период работы алгоритма делится на три стадии – начало, середина и конец. Размер популяции также делится на три типа – [2, *psize*/3] малый, *(psize*/3, 2**psize*/3] средний, и (2**psize*/3, *psize*] большой. Испытано

5 схем изменения – фиксированный размер, случайное изменение, возрастание (малый-средний большой), убывание, схема горы (средний-большой средний) и схема долины (большой-средний-большой). Эксперимент показал, что все схемы имеют преимущество перед фиксированной.

Коэволюционные алгоритмы предполагают наличие некого набора принципиально различных способов эволюции в генетическом алгоритме, каждый из которых применяется к той или иной части популяции. Метод жадных эвристик предполагает использование единственной схемы формирования дочерних особей, основанной на объединении множеств-особей с последующим сокращением мощности полученного множества. Такая схема показала свою эффективность [123]. Данная схема может иметь некоторые вариации [123, 143], выбор способа скрещивания определяется параметрами задачи, в частности –числом групп (кластеров) в решении, поэтому использование коэволюционных подходов к нашим задачам автоматической группировки вряд ли позволить существенно повысить эффективность алгоритма.

Более того, генетические алгоритмы метода жадных эвристик предполагают работу с популяцией очень малого размера (в [123] предлагается популяция в 15-20 особей). Это ставит под сомнение саму возможность разбиения такой малой популяции на еще меньшие популяции, а также использование других подобных сложных схем регулирования размера популяции для повышения эффективности алгоритма.

При этом следует отметить, что для очень больших задач даже при такой малой популяции сложность жадной агломеративной процедуры скрещивания, используемой в генетических алгоритмах метода жадных эвристик, не позволяет дойти даже до третьего-четвертого поколений особей за разумное время, предполагающее решение задач в диалоговом режиме. Поэтому выбор оптимального размера популяции для конкретной задачи, либо разработка алгоритма с самонастраивающимся (динамическим) размером популяции, является важной задачей, которая решена в Главе 2.

1.6 Генетические алгоритмы с переменной длиной хромосом

Традиционно при оптимизации длина хромосом (т.е. длина строки, которой кодируется решение) фиксируется априори и не может меняться с эволюцией последующих поколений.

Этот традиционный подход имеет некоторые недостатки при решении сложных задач. Во-первых, если решения кодируются традиционным для генетических алгоритмов образом – L-битной строкой, то при фиксированной длине хромосом наилучшая достижимая пригодность по своей природе ограничена хромосомной длиной. Следовательно, асимптота пригодности, которая обычно наблюдается в генетической оптимизации, является в значительной степени следствием ограничений числа переменных проектирования и их разрешений. Для генетического алгоритма с вещественным алфавитом данное свойство не характерно. При этом практическая реализация такого алгоритма, конечно, сводится к кодированию вещественных чисел типами данных float или double, точность которых ограничена (а следовательно – ограничена и достижимая пригодность), но данную проблему следует скорее отнести к разряду проблем архитектуры вычислительных систем, а не алгоритмов.

Во-вторых, проблема в том, что мы априори не знаем, насколько нужна свобода выбора и, следовательно, насколько длинными должны быть хромосомы. Если используются короткие хромосомы, невозможно получить хорошие решения из-за отсутствия свободы выбора. С другой стороны, если длина хромосомы является чрезмерной для конкретной проблемы, это вызовет высокую вычислительную нагрузку без большой выгоды для работы.

Для генетических алгоритмов метода жадных эвристик для решения задач автоматической группировки с вещественным алфавитом длина кода естественным образом определяется числом групп (кластеров) в решении: особь (хромосома) такого алгоритма представляет собой множество из k точек в d-мерном пространстве, где k – число групп (кластеров), на которые разбиваются объекты, а d – размерность пространства. В то же время, в некоторых случаях

[123, 143] в скрещивании участвует не вся хромосома, а лишь ее часть (подмножество), выбираемая случайным образом. При этом случайный характер выбора части хромосомы обоснован лишь эмпирически, при этом сравнение производилось лишь с генетическим алгоритмом, в котором хромосомы использовались при скрещивании полностью, а также с другими (не генетическими) алгоритмами. Это наталкивает на мысль о том, что в популяции, в принципе, могли бы использоваться особи с хромосомами различной длины.

Для обзора можно поделить генетические алгоритмы с переменной длиной хромосомы на три категории – с фиксированным размером решения, ограниченным размером и неограниченным размером.

а) Фиксированный размер решения

Решение с фиксированным размером – это такое решение, которое требует заданного количества переменных для полноты или оптимальности, при малом числе значений переменных решение не может быть проверено или оптимальное решение не может быть найдено. Первые ГА были разработаны с учетом этого типа задачи. Гены или биты в хромосоме ГА соответствовали значениям переменных для потенциального решения, основанным на их положении в хромосоме. Применительно к нашим задачам в качестве набора переменных традиционно выбираются координаты центров/центроидов/медоидов – главных точек.

Самое раннее и наиболее известное представление алгоритма с переменной длиной хромосом – это messyGA [144]. Этот ГА был разработан для устранения битовых позиционных зависимостей в стандартном ГА. Это представление позволяет ГА эволюционировать как битовые значения, так и битовые местоположения на хромосоме во время выполнения ГА. Цель messyGA состоит в том, чтобы найти тесную связь между битами, а также хорошими битными значениями одновременно.

Каждое значение бита в беспорядочной хромосоме помечено именем (целое число, указывающее положение бита). Перед оценкой пригодности биты значения

извлекаются из хромосомы и переупорядочиваются на основании их соответствующих им имен. Биты внутри генотипической последовательности больше не находятся в фиксированных позициях и могут перемещаться по хромосоме, чтобы получить лучшие блоки для строительства.

MessyGA был реализован с использованием оператора cut-and-splice, а не стандартного одноточечного кроссовера. Оператор производит хромосомы переменной длины. Иногда воспроизведение может привести к тому, что у потомков будет несколько битовых значений для заданного имени (чрезмерная спецификация) или отсутствие битовых значений для данного имени (недостаточная спецификация). Экземпляры этих двух ошибок должны быть рассмотрены до того, как полное решение может быть отправлено в функцию пригодности для оценки.

Для устранения чрезмерной спецификации из-за повторяющихся или конкурирующих значений битов для данного имени используется приоритет слева направо. Значение бита от первого беспорядочного гена с заданным именем используется во время оценки пригодности. Все другие беспорядочные гены с тем же именем затем игнорируются.

Недостаточная спецификация возникает, когда не существует беспорядочного гена для заданного бита, требуемого в предварительном решении. Это условие обрабатывается с помощью конкурентных шаблонов. Битовые значения, не найденные в геноме, извлекаются из шаблона, строки, содержащей локально оптимальные биты предыдущего поколения. Использование конкурентных шаблонов позволяет messyGA построить комплексное решение для тестирования, даже если отдельная хромосома не кодирует решение в целом.

Цель messyGA заключалась в том, чтобы позволить ГА развиваться быстрее и находить лучшие решения для задач с фиксированным размером решения. Ограничением оригинального messyGA является его сосредоточенность на битах. Полное решение (без учета недостаточной или чрезмерной спецификации) для задачи с n переменных с каждой переменной, требующей l битов, требует $l*n*(\text{floor}(\log_2(l*n-1))+1)$ дополнительных битов в геноме для идентификации всех

его составных битов. Для очень больших задач этот объем служебных данных может быстро превысить 90% бит в хромосоме.

Другим вариантом алгоритма с переменной длиной хромосомы для решений фиксированного размера является Virtual Virus или VIV [145]. Целью этой работы было создание генетического алгоритма, который бы более точно отражал биологию. Авторы описали ряд ключевых особенностей, обнаруженных в биологических системах, которые могут отсутствовать в стандартном ГА, включая:

- Геномы, которые различаются по длине в процессе эволюции,
- Расположение генов независимо от положения, идентифицированного кодонами остановки и запуска,
- Наличие некодирующих областей в геноме,
- Дублирующие или конкурирующие гены на одном и том же геноме,
- Использование перекрывающихся кадров считывания при извлечении генетической информации,
- Многосимвольный алфавит (A, T, C и G, представляющий нуклеотиды)
- Вырожденное картирование нуклеотидных триплетов (кодонов) в аминокислоты.

Каждая из этих функций была в некоторой степени включена в VIV. Использование многохарактерного алфавита и дегенеративного картирования позволило использовать в VIV избыточные представления генотипов. Более одного кодона (три буквы алфавита) могут трансформироваться в один и тот же фенотипический признак. Включение межгенных областей (некодирующая информация) и избыточных генов (одна и та же информация может появляться более одного раза) также позволяет VIV более точно следовать своему биологическому образцу.

Чтобы удалить локальные ограничения, каждый ген в VIV отмечен кодоном СТАРТ, состоящим из трех символов. Этот начальный тег аналогичен промоторной области ДНК. Конец гена определяется соответствующим кодоном

СТОП. Использование кодов СТАРТ и СТОП позволяет:

1) генам в VIV иметь вариабельную длину,

2) некодирующим областям появляться между кодами СТОП и СТАРТ;

3) перекрывать гены из-за трех кадров считывания.

Обработка хромосомы начинается с чтения генома слева направо. Если кодон СТАРТ найден в позиции *p*, любые последующие символы считываются как гены до тех пор, пока не будет достигнут кодон СТОП. Чтение генома возвращается в положение *p*+1, ища следующий кодон СТАРТ. Такая обработка хромосом позволяет VIV иметь внутри хромосомы перекрывающиеся гены. Перекрытие делает возможной эволюцию решений в очень компактном геноме.

Также нужно упомянуть пропорциональный генетический алгоритм (PGA) [146]. Этот алгоритм выдает решения с фиксированной длиной, в то же время не зависящие от локуса. В PGA используется многосимвольный алфавит. Каждый символ связан с конкретной переменной, которая может быть найдена в решении. Количество каждого из символов, присутствующих в геноме, используется для вычисления значения, которое затем присваивается соответствующей переменной.

В случае PGA1, значение вычисляется уравнением:

$$P_{PGA1}(V_i) = \frac{количество\ символов\ (V_i)\ в\ геноме}{длина\ особи},$$

где V_i это *i*-ая переменная в решении. Сумма всех переменных равна 1. Таким образом PGA1 хорошо приспособлен для задач распределения ресурсов.

Хромосома длиной 20 ABBDAFEDEFADBAAABFDD

Значения переменных

A = 6/20 = 0,30

B = 4/20 = 0,20

C = 0/20 = 0,00

D = 5/20 = 0,25

E = 2/20 = 0,10

F = 3/20 = 0,15

PGA использует как атомарные единицы (не биты или гены), а символы

алфавита. Цитируя [146], «PGA основан на идее, что значение имеет содержание, а не порядок кодируемой информации».

PGA является нашим последним примером генотипа переменной длины, который кодирует решение фиксированного размера. Количество символов в алфавите PGA определяется в начале на основании количества переменных в полном решении. Недостатком здесь является необходимость больших алфавитов, если задачи имеют большое количество переменных.

Основное преимущество PGA заключается в том, что излишняя и недостаточная спецификация больше не является проблемой. Все назначенные символы в геноме используются для вычисления пропорций для каждой переменной. Недостаточная спецификация (отсутствие символа) обрабатывается просто установкой значения, связанного с отсутствующим символом, в «0». Другим важным преимуществом этого алгоритма является отсутствие какой-либо «накладной» информации в геноме – не требуются идентификационные метки или начальные/конечные теги.

б) Ограниченный размер решения

Задача с ограниченным размером решения означает задачу с заданным максимальным количеством возможных переменных, но не требующую для создания полного тестируемого решения конкретизации всех переменных. Примером такой задачи может служить задача о рюкзаке 0-1.

Хорошо известным классом задач с ограниченным размером решения является выработка правил для микробеспилотников [147, 148]. В этих статьях ГА вырабатывают наборы правил для управления и навигации автономных транспортных средств. Существующие правила сопоставляются с входами датчиков из моделируемой среды. Наилучшие согласованные правила, влияющие на движение автономного транспортного средства выполняются.

Эти задачи имеют ограниченный размер решения, поскольку существует максимальное количество правил, которые могут быть сгенерированы при условии одного правила для каждого возможного условия. Например, каждый беспилотник в [149] имеет в общей сложности восемь датчиков, которые

находятся в состоянии «включено» или «выключено». Всего может существовать 2^8 возможных состояний среды или условий. Наибольшее возможное решение для этой проблемы будет содержать 256 условий/правил действия, предполагая, что мы связываем одно действие с каждой возможной комбинацией состояний датчиков. Число возможных условий можно рассматривать как верхнюю границу числа переменных для этой задачи.

Однако не каждое условие требует своего собственного специализированного правила для создания правильного и наиболее подходящего решения. Результаты экспериментов в указанных выше работах показывают, что для достижения хороших результатов часто бывает достаточно компактного набора общих правил (от 5 до 20 правил).

В ГА для этих задач недостаточная или избыточная спецификации непосредственно не учитывались. Вместо этого эти ситуации были разрешены путем подбора алгоритмов выбора правил в симуляторе беспилотника. Для недостаточной спецификации (конкретному условию нет соответствия), беспилотник выбирает правило, которое наиболее близко соответствует текущей среде. В случае избыточной спецификации (несколько действий для одного и того же условия) беспилотник выбирает случайным образом между конкурирующими правилами, чтобы определить действие, которое необходимо предпринять.

Другим примером ГА, для решений с ограниченным размером, является система SAMUEL [150]. Эта система использует ГА с переменной длиной в основе своего модуля обучения, чтобы изучать правила для последовательного принятия решений. Каждая особь в популяции представляет собой свод правил или «тактический план», который должен служить руководством для SAMUEL в его задачах. Планы состоят из переменного количества правил, которые загружаются в модуль SAMUEL для выполнения.

Правила выражаются в высокоуровневом *если/то* представлении, а не в двоичных строках. Кроссовер происходит между правилами и служит для объединения хороших правил для формирования лучшего тактического плана.

Создание новых правил – это функция мутации и нового оператора

специализации, который создает более специализированную версию ранее существовавшего правила. Для сокращения числа правил используются операторы обобщения и слияния.

В [150] SAMUEL учится направлять действия самолета, чтобы избежать зенитной ракеты. Действия по управлению курсом и скоростью полета применяются на основе условий, определенных набором из шести различных датчиков, отслеживающих информацию о среде вокруг самолета и ракете. Могут возникать свыше 25 миллионов возможных состояний и это является верхней границей размера полного тактического плана (решения). Это число велико, но представляет собой некоторое ограничение размера. Следует отметить, что авторы этой работы ограничились решениями с 32 или менее правилами – числом, существенно меньшим, чем верхняя граница. Никакое объяснение этому ограничению не приводится. Недостаточная или избыточная спецификации обрабатываются алгоритмом соответствия и выбора, включенным в систему SAMUEL.

B) Неограниченный размер решения

Третьей и последней категорией решений являются те, которые имеют неограниченный размер, это относится к задачам с неограниченной сложностью. В [151] утверждается, что генотипы должны быть неограниченными по длине, если мы хотим получить «структуру с произвольными и потенциально неограниченными возможностями». Неограниченные задачи, подобные тем, которые предлагаются в [151], часто встречаются в приложениях с искусственным интеллектом, где размер и сложность решения не могут быть определены с самого начала. Отраслями эволюционных вычислений, наиболее часто решающих проблемы такого типа, являются генетическое программирование и грамматическая эволюция.

ГА с хромосомами переменной длины также может решать те же самые проблемы. В качестве примера можно привести [152]. В этой работе авторы представляют новый подход к идентификации функций. Вместо того, чтобы искать глобальную функцию, охватывающую всю функциональную область,

функциональное пространство делится на области Вороного. Найдена локальная функция аппроксимации, которая обеспечивает хорошее приближение функции в пределах области. Локальный вектор кодирует параметры для идентификации одной области Вороного и ее локального аппроксиматора. Каждая особь в популяции представляет собой список переменной длины, составленный из локальных векторов. Комбинация всех локальных векторов в одном индивидууме охватывает все функциональное пространство и обеспечивает полную аппроксимацию.

Размер потенциальных решений для этого нового подхода к определению функций неограничен по двум причинам. Во-первых, размер самой области функций является неограниченным, что, возможно, требует большого количества областей Вороного для достижения хорошего приближения. Во-вторых, качество решения может зависеть от степени детализации самих регионов. Многие небольшие регионы могли бы дать лучшее решение, чем несколько крупных. Поэтому размер отдельной хромосомы должен допускать неограниченный рост числа локальных векторов, чтобы справиться с этими двумя проблемами.

Большинство представлений переменной длины, независимо от парадигмы эволюционных вычислений, в той или иной степени страдают от раздувания. В общем, раздувание представляет собой увеличение длины генома от одного поколения к другому, в первую очередь за счет роста посторонней генотипической информации. Эта информация является посторонней, поскольку не необходима для получения составлять действительного и/или оптимального решение.

Исследователи ГА разработали различные подходы к обработке раздувания в своих работах. В вышецитированных работах используются следующие методы:

- messyGA: нет явного механизма контроля раздувания; Однако авторы предполагают, что использование конкурентных шаблонов препятствует дублированию генов.

- VIV: линейный штраф длины добавляется к функции пригодности. Это изменяет первоначальную исходную пригодность каждой хромосомы, так что предпочтение отдаётся более коротким особям, при условии, что все исходные

значения пригодности равны.

- PGA: После кроссовера с любым потомком, длина которого превышает заданный предел применяется простейшее правое усечение. Все символы/гены, выходящие за этот предел, удаляются из хромосомы.

- SAMUEL: оператор кроссовера удаляет все повторяющиеся правила от того же самого потомка. Оператор специализации не может создавать новые правила, если максимальное число правил для плана уже достигнуто.

Алгоритмы с переменной длиной наиболее развиты в генетическом программировании. Генетическое программирование использует деревья разбора, которые кодируются как отдельные хромосомы. Было замечено, что генетическое программирование использует деревья, содержащие один или несколько разделов неиспользуемого кода, намного больше, чем необходимо [153, 154, 155]. Из всех парадигм эволюционных вычислений исследователи генетического программирования выпустили наибольшее количество работ, связанных с раздуванием, из-за обильного использования геномов переменной длины.

Методы контроля раздувания для генетического программирования более многочисленны, разнообразны и часто более сложны, чем те, которые описаны в литературе по ГА. Это может быть связано с популярностью генетического программирования и большим объемом работы по сравнению с ГА с переменной длиной или из-за более сложного характера задач генетического программирования.

Отметим, что генетические алгоритмы метода жадных эвристик для задач автоматической группировки предполагают указание некоторой верхней границы количества групп (кластеров), на которые разбиваются объекты, даже если точное количество кластеров априори неизвестно. Это ограничение в практических задачах обычно вытекает из характера группируемых объектов или данных, хотя для любой задачи такое ограничение существует: число групп никогда не превышает заранее известное число группируемых объектов. При этом особи с различной длиной хромосом могли бы естественным образом представлять собой решения задач с различным числом групп (кластеров).

1.7 Генетические алгоритмы со случайным выбором длины решения

Первое упоминание о случайном выборе размера хромосомы можно найти в [156]. Было отмечено, что при случайном отборе не происходит раздувания. Автор предположил, что раздувание было пропорционально давлению отбора – чем больше давление отбора, вызванное пригодностью, тем быстрее происходит увеличение размера хромосом. Было высказано предположение, что раздувание может быть дополнительно усилено силами дрейфа, который переносить некодирующие регионы наряду с полезными материалами для следующего поколения особей.

Эксперименты, выполненные в [157], подтверждают некоторые из этих результатов. Показано, что раздувание связано с давлением отбора. Отсутствие давления выбора останавливает рост размера программы. Авторы обнаружили «медленное сокращение размера программы» при случайном выборе. Это сокращение объясняется смещениями в операторе пересечения, которые требуют, чтобы потомки были меньше, чем предварительно заданное ограничение длины.

В [158] предлагается использовать два новых генетических оператора: одноточечный кроссовер и точечную мутацию. Новый кроссовер работает, выбирая точку на обоих родителях из общего региона. Общая область состоит из частей обоих родителей, начиная с корня с одинаковой арностью узлов и связанных ребер. Одна точка выбирается равномерно по краям внутри этой области, чтобы служить в качестве точки пересечения. Поддеревья ниже этой точки затем обмениваются между родителями, чтобы произвести двух новых потомков. Этот тип одноточечного кроссовера не увеличивает глубину потомков по отношению к обоим родителям, но поощряет смешивание поддеревьев. Использование этой специфической формы кроссовера позволило разработать свою новую схемы, которая математически моделирует «соревнования между программами с различной структурой и программами с одинаковой структурой».

Примерно в то же время в [159, 160] была разработана новая схема для ГА.

Главным вкладом этих работ было добавление реконструкции схемы через кроссовер и мутацию в оригинальную, более пессимистическую схему. Другой важный вывод [160] заключается в следующем:

"Мы также показали, что в общем случае предпочтение отдается коротким схемам низкого порядка. Фактически, если реконструкция схемы доминирует, верно обратное – обычно предпочтительны большие схемы. Только в обманчивых задачах оказывается, что предпочтительными будут короткие схемы, а затем только в абсолютно обманчивых задачах, поскольку система будет стремиться искать существующие необманные каналы "

В этой цитате использование терминов «короткий» и «большой» относится к определяющей длине схем. Это открытие важно, как возможный ключ к причине раздувания в ГА с переменной длиной. Без некоторого механизма контроля раздувания такие ГА быстро растут в размерах в течение первых поколений, поскольку они работают, чтобы комбинировать строительные блоки в более полные и лучшие решения. Тот факт, что большие схемы могут улучшить конструкцию строительных блоков, может объяснить, почему ГА растет так быстро.

Идея точной схемы была применена к новой теории схем в [161] для создания «макроскопической точной теоремы о схеме для генетического программирования с одноточечным кроссовером». В эту работу был включен пример, который вычислял общую вероятность передачи и эффективной пригодности для конкретной схемы в ограниченной популяции.

Вышеописанный пример послужил толчком для трех последующих работ: [162, 163, 164]. В этих работах авторы развивают точную теорию схем для линейных структур (подобных ГА с переменной длиной), состоящих исключительно из 1-арных функций и одного терминала. Результаты были получены для линейных структур как в описанном выше одноточечном кроссовере, так и в стандартном кроссовере. В стандартном кроссовере точки кроссовера выбираются для обоих родителей независимо друг от друга – никакой общей области не используется. Эта версия кроссовера эквивалентна той, которая

используется в большинстве ГА с переменной длиной.

Для одноточечного кроссовера, предполагая бесконечную совокупность и постоянную функцию пригодности (например, равномерную или случайную выборку), средний размер особей со временем остается постоянным – фиксированной точкой, равной среднему размеру первоначальной совокупности. Кроме того, распределение длин внутри популяции не изменяется с течением времени. Распределение остается фиксированной точкой, также равной распределению длины в исходной популяции. Одноточечный кроссовер не влияет на длину структуры.

Результаты при стандартном (как в ГА) кроссовере другие. Опять же, предполагая бесконечную популяцию и постоянную функцию пригодности, средний размер от одного поколения к следующему остается постоянным и фиксированным в точке, равной среднему размеру первоначальной совокупности. Однако распределение длин меняется со временем. Авторы показывают, что при постоянной пригодности структуры более короткие, чем в среднем, образцы отбираются чаще, чем более крупные. Со временем крупные особи становятся все больше, но их становится все меньше, в то время как более короткие особи сжимаются, но становятся более многочисленными. Таким образом, мы видим изменение в распределении длины, но не изменяем среднюю длину от одной популяции к другой. Заканчивая свою работу [163], авторы показали, что распределение длины в стандартном кроссовере не является фиксированной точкой, а представляет собой набор неподвижных точек во времени, определяемых семейством дискретных гамма-функций.

В [165] алгоритм выявляет конфликтные с точки зрения целевой функции гены и делает их точками разреза хромосомы для дальнейшего кроссовера и мутаций.

Алгоритм синаптического кроссовера переменной длины (SVLC) [166] использует биологический метод для выполнения кроссовера между геномами переменной длины. В отличие от других методов кроссовера с переменной длиной, которые рассматривают геномы как жесткие негибкие массивы и где

некоторые или все точки пересечения выбираются случайным образом, алгоритм SVLC считает, что геномы гибки и выбирает неслучайные точки пересечения на основе общего сходства родительских последовательностей. Алгоритм SVLC периодически «склеивает» или объединяет гомогенные генетические подпоследовательности. Это делается таким образом, что общие родительские последовательности автоматически сохраняются у потомства с заменой или удалением только генетических различий независимо от длины таких различий.

Изучив различные подходы к контролю (или методы контроля) размера хромосом, мы пришли к выводу, что, с одной стороны, нет необходимости в излишнем усложнении алгоритма, и, с другой стороны, есть смысл в сочетании различных подходов. Поэтому для контроля размера хромосом мы избрали случайный метод, однако, с ограничениями. Более подробно этот вопрос будет рассмотрен в Главе 2.

1.8 Модели автоматической группировки на основе разделения смеси распределений и EM-алгоритм

Модели группировки объектов могут быть основаны не только на минимизации суммарных расстояний между объектами группы, но и на разделении смесей вероятностных распределений. Совокупность параметров объектов одной группы является многомерной случайной величиной, распределенной по некоторому известному закону распределения, при этом параметры этого распределения неизвестны. Предполагается, что выборка содержит объекты разных групп, то есть параметры объектов порождены различными распределениями и нужно отделить объекты, предположительно порожденные одним из распределений в этой смеси, от других.

Одним из широко известных в аналитическом сообществе алгоритмов кластеризации, позволяющих эффективно решать оптимизационную задачу разделения смеси распределений, является EM-алгоритм. Его название происходит от слов "expectation-maximization", что переводится как "ожидание-

максимизация". Это связано с тем, что каждая итерация содержит два шага – вычисление математических ожиданий (expectation) и максимизацию (maximisation). Название впервые появилось в работе Демпстера, Лэрда и Рубина (A. P. Dempster, N. M. Laird, D. B. Rubin) [167], посвященной методике итеративного вычисления оценок максимального правдоподобия.

Практика показывает, что преимущества алгоритма заключается в надежной глобальной конвергенции, низкой стоимости итерации, экономии памяти и простоте программирования, а также в некоторой эвристической привлекательности. К сожалению, его сходимость может быть чрезвычайно медленной в простых задачах, которые часто встречаются на практике.

В контексте плотности смеси EM-алгоритм был получен и изучен по крайней мере с двух разных точек зрения несколькими авторами, многие из которых работают независимо. Хассельблад (V. Hasselblad) [168] получил EM-алгоритм для произвольной конечной смеси одномерных нормальных плотностей и сделал эмпирические наблюдения о его поведении. Далее, он описывал алгоритм для практически произвольных конечных смесей одномерных плотностей из экспоненциальных семейств в [169]. Алгебраический алгоритм работы [168] для одномерных нормальных смесей был вновь предложен Бехбудианом (J. Behboodian) [170], а Дэй (N. E. Day) [171] и Вулф (J. H. Wolfe) [172] сформулировали его соответственно для смесей двух многомерных нормальных плотностей с общей ковариационной матрицей и произвольными конечными смесями многомерных нормальных плотностей. Все эти авторы, по-видимому, самостоятельно получали EM-алгоритм, хотя Вулф [172] ссылался на Хассельблада [168]. Все они вывели алгоритм, установив частные производные функции логарифмического правдоподобия равными нулю и после некоторых алгебраических манипуляций получили уравнения, предлагающие алгоритм.

Следуя этим ранним выводам, EM-алгоритм был применен Таном и Чангом (W. Y. Tan, W. C. Chang) [173] к задаче смешения в генетике и использован Хосмером (D. W. Hosmer, Jr.) [174] в исследовании оценок максимального правдоподобия методом Монте-Карло. Дуда и Харт (R. O. Duda, P. E. Hart) [175]

применили EM-алгоритм для смесей многомерных нормальных плотностей и прокомментировали его поведение на практике. Хосмер [176] расширил EM-алгоритм для смесей двух одномерных нормальных плотностей, включив частично помеченные образцы. Петерс и Уокер (B. C. Peters, Jr., H. F. Walker) [177] предложили локальный анализ сходимости EM-алгоритма для смесей многомерных нормальных плотностей и предложили модификации алгоритма для ускорения сходимости. Петерс и Коберли (B. C. Peters, Jr., W. A. Coberly) [178] изучили EM-алгоритм для аппроксимации оценок максимального правдоподобия для пропорций по существу произвольной плотности смеси и дали локальный анализ сходимости алгоритма. Петерс и Уокер [179] обобщили результаты работы [178] на включение подмножеств пропорций смесей и анализ локальной конвергенции вдоль линий.

Все вышеупомянутые исследователи считали, что EM-алгоритм естественно возникает из частных форм, принимаемых частными производными функции логарифмического правдоподобия. Совершенно другая точка зрения на алгоритм была выдвинута Демпстером, Лэрдом и Рубином [167]. Они интерпретировали проблему оценки плотности смеси как проблему оценки, включающую неполные данные. При этом они не только связывали проблему плотности смеси с более широким классом статистических задач, но также показали, что EM-алгоритм для задач плотности смеси действительно является специализацией более общего алгоритма (также называемого EM-алгоритмом в [167]) для аппроксимации оценок максимального правдоподобия по неполным данным. Как видно в дальнейшем, этот более общий EM-алгоритм определяется таким образом, что он имеет некоторые желательные теоретические свойства по самому его определению. Ранее EM-алгоритм был определен независимо очень похожим образом Баумом (L. E. Baum) и др. [180] для очень общих задач оценки плотности смеси, Сундбергом (R. Sundberg) [181] для задач с неполными данных при участии экспоненциальных семейств и Хаберманом (S. J. Haberman) [182, 183, 184] для проблем, связанных с смесями, включая таблицы частот, полученные косвенным наблюдением. Хаберман также ссылается в [184] на версии своего алгоритма,

разработанные Кеппеллини, Синискалько и Смитом (R. Ceppelini, S. Siniscalco, C. A. B. Smith) [185], Ченом (T. Chen) [186] и Гудманом (L. A. Goodman) [187]. Кроме того, интерпретация задач смеси как задач с неполными данными была дана в кратком обсуждении смесей Орчардом и Вудбери (T. Orchard, M. A. Woodbury) [188]. Желательные теоретические свойства, автоматически используемые EM-алгоритмом, в свою очередь указывают на хорошую глобальную сходимость алгоритма, которая наблюдалось на практике многими исследователями. Теоремы, которые существенно подтверждают это поведение, были получены Реднером (R. A. Redner) [189], Варди (Y. Vardi) [190], Бойлсом (R. A. Boyles) [191] и By (C.-F. Wu) [192] и описаны в дальнейшем.

Классический EM-алгоритм относится к так называемым «жадным» алгоритмам, что приводит к «застоям» в локальном максимуме функции правдоподобия. Избежать этого позволяет рандомизация процесса оптимизации. Подобная методика использована в SEM-алгоритме (Stochastic EM-алгоритм), предложенном в работах Бронятовски, Целю и Диболта (M. Broniatowski, G. Celeux and J. Diebolt) [193, 194, 195]. Теоретические исследования свойств SEM-алгоритма были позднее осуществлены в работах Ипа (E. H. Ip) и Диболта [196, 197].

CEM-алгоритм (Classification EM-алгоритм), в котором вместо рандомизации (как в SEM-алгоритме) используется детерминированное правило, эквивалентное классификации по принципу максимума апостериорной вероятности, был разработан Целю и Говертом (G. Govaert) [198, 199].

Общность как постановок задач k-средних и задачи разделения смеси распределений, так и структуры алгоритмов k-средних и EM, позволяет предположить возможность применения аналогичных подходов к повышению точности и стабильности этих алгоритмов, а также способов организации решения серии задач с различным числом кластеров или распределений.

Выводы к Главе 1

Метод жадных эвристик в процессе работы оперирует уменьшающимся числом кластеров и, таким, образом, в нём уже заложена возможность решения серии задач, различающихся только числом кластеров. Однако, есть и трудности – при малых *p* алгоритм дает менее точные результаты в сравнении с другими известными алгоритмами. В этой главе мы выдвинули гипотезу о связи этой трудности с низкой вариативностью популяции, а также наметили пути решения этой проблемы – динамический контроль размера популяции и ее гетерогенный состав. Приведенный обзор работ в этой области дает обоснованную надежду, что рабочая гипотеза найдет экспериментальное подтверждение, а потенциал, заложенный в методе жадных эвристик, будет реализован в более полной мере, чему и посвящена следующая глава.

ГЛАВА 2. НОВЫЕ СЕРИЙНЫЕ АЛГОРИТМЫ МЕТОДА ЖАДНЫХ ЭВРИСТИК ДЛЯ РЕШЕНИЯ ЗАДАЧИ АВТОМАТИЧЕСКОЙ ГРУППИРОВКИ

В данной главе мы рассмотрим новый алгоритм решения задачи автоматической группировки, предложенный нами в рамках проверки гипотезы о повышении эффективности метода жадных эвристик путем увеличения вариативности популяции.

2.1 Генетический алгоритм решения задачи автоматической группировки с динамической популяцией

Как было сказано выше, такие задачи, как, например, задачу выявления однородных партий электрорадиоизделий можно свести к задаче k-средних. Модель k-средних – наиболее популярная модель кластерного анализа.

Для решения задача k-средних может быть использован одноименный алгоритм, называемый также ALA-алгоритмом (Alternating Location-Allocation – чередующееся размещение-распределение), включающий два чередующихся шага:

Алгоритм 2.1 ALA- процедура.

Дано: векторы данных $A_1 ... A_N$, k начальных центров кластеров $X_1 ... X_k$.

1. Для каждого центра X_i составить кластер (множество) C_i векторов данных, для которых этот центр является ближайшим.

2. Для каждого кластера C_i рассчитать новое значение центра X_i (т.е. решить задачу Вебера).

3. Повторить с шага 1, если шаги 1 и 2 привели к каким-либо изменениям хотя бы одного значения центра.

Этот же алгоритм может эффективно использоваться и для непрерывных p-медианных задач.

За исключением особых случаев, задачи k-средних и p-медианная являются NP-трудными, требующими глобального поиска [92].

Результат ALA-процедуры зависит от выбора начальных центров кластеров. Известный алгоритм k-means++ [115] имеет преимущество в сравнении со случайным выбором центров из множества векторов данных с равной вероятностью. Тем не менее, достигаемое им увеличение точности результата является недостаточным для многих практических задач, требующих более точного и, главное, стабильного результата. Для таких задач предложено множество алгоритмов, суть которых сводится к рекомбинации множеств точек, выбираемых в качестве начального решения ALA-процедуры или других процедур локального поиска [36].

Результат ALA-процедуры может быть улучшен с применением различных подходов. Например, процедуры сэмплинга [200] решают задачу для случайным образом выбираемого подмножества векторов данных, мощность которого в несколько раз меньше мощности множества всех векторов данных, после чего полученные центры кластеров используются в качестве начального решения исходной задачи с полным набором векторов данных.

Зависимость результатов ALA-процедуры от выбора начальных центров является серьезной проблемой с точки зрения обеспечения воспроизводимости результатов вычислений. В зависимости от выбранного случайным образом начального множества центров, результаты разбиения на кластеры могут сильно различаться. Для нашей задачи это означает, что два изделия, в зависимости от выбора начальных центров, могут быть в результате классифицированы как принадлежащие либо одной и той же, либо различным производственным партиям. Поскольку процедура классификации электронных компонентов, как и любой элемент технологического процесса в космической отрасли, должна быть строго регламентирована, предпочтение должно отдаваться методу, обеспечивающему не только очень точный, но и стабильный, т.е. воспроизводимый результат.

Довольно точный, но чрезвычайно медленный при больших объемах данных

метод информационного узкого места (IBC – Information Bottleneck Clustering) является детерминированным методом для задач кластеризации [201]. Алгоритм в начале своей работы считает каждый вектор данных отдельным кластером. Затем кластеры один за другим удаляются, а входящие в них векторы данных перераспределяются между оставшимися. Каждый раз удаляется тот кластер, удаление которого дает наименьший прирост целевой функции. Таким образом, этот алгоритм можно отнести к «жадным» алгоритмам. Так происходит до достижения требуемого числа кластеров.

Генетические алгоритмы (ГА) с жадной агломеративной эвристикой, изначально разработанные для p-медианной задачи на сети [120], являются компромиссными по точности результата и вычислительны затратам. В [119, 1] авторы предлагают подход к адаптации этих алгоритмов к непрерывным задачам размещения:

Алгоритм 2.2 ГА с жадной эвристикой и вещественным алфавитом.

Дано: Множество $V = (A_1,...,A_N) \in \Re^d$, число кластеров p, размер популяции N_p.

1: Создать N_p множеств координат $\chi_1,...,\chi_{N_p}$: $\chi_i \subset \Re^d$, $|\chi_k| = p \forall k = \overline{1, N_p}$, являющихся результатами выполнения ALA-процедуры. Таким образом, в каждом χ_i достигается локальный минимум задачи. Запомнить соответствующие значения целевой функции в переменных $F_1,...,F_{N_p}$.

2: Если достигнуты условия останова, перейти к Шагу 8.

3: Случайным образом выбрать два «родительских» множества χ_{k_1} и χ_{k_2}, $k_1, k_2 \in \left\{ \overline{1, N_p} \right\}, k_1 \neq k_2$. Запуская Алгоритм 2.3, получить «дочернее» множество координат χ_c, в котором достигается локальный минимум целевой функции. Сохранить значение целевой функции в переменной F_c.

4: Если $\exists k \in \left\{ \overline{1, N_p} \right\}: \chi_k = \chi_c$, то перейти к Шагу 2.

5: Выбрать индекс $k_{worst} = \arg\max_{k=\overline{1,N_p}} F_k$. Если $F_{worst} < F_c$, то перейти к Шагу 2.

6: Случайным образом выбрать два индекса k_1 и k_2, $k_1 \neq k_2$; выбрать $k_{worst} = \arg\max_{k \in \{k_1, k_2\}} F_k$.

7: Поменять местами значения $\chi_{k_{worst}}$ и χ_c, сохранить $F_{k_{worst}} = F_c$ и перейти к Шагу 2.

8: ОСТАНОВ. Результатом является множество χ_k^*, $k^* = \arg\min_{k=\overline{1,N_p}} F_k$.

Модификация жадной агломеративной эвристики для этого алгоритма следующая:

Алгоритм 2.3 Жадная процедура кроссинговера для Алгоритма 2.2.

Дано: Множество $V = (A_1,...,A_N) \in \Re^d$, , количество кластеров p, два "родительских" множества центров χ_{k_1} и χ_{k_2}, значения σ_e и L_{min}.

1: Объединить множества $\chi_c = \chi_{k_1} \bigcup \chi_{k_2}$. Запустить ALA-процедуру для $|\chi_c|$ кластеров, начиная с решения χ_c. Сохранить результат в χ_c.

2: Если $|\chi_c| = p$, то запустить ALA-процедуру из начального решения χ_c, затем ОСТАНОВ, результатом является χ_c.

2.1: Вычислить расстояния от каждого из векторов данных до ближайшего элемента множества χ_c.

$$d_i = \min_{X \in \chi_c} L(X, A_i) \forall i = \overline{1, N}.$$

Для каждого из векторов данных определить ближайший центр χ_c.

$$C_i = \arg\min_{X \in \chi_c} L(X, A_i) \forall i = \overline{1, N}.$$

Вычислить расстояние от каждого из векторов данных до второго ближайшего к нему элемента (центра) из множества χ_c.

$$D_i = \min_{Y \in (X_c \setminus \{C_i\})} L(Y, A_i).$$

3: Для каждого $X \in \chi_c$ вычислить $\delta_X = F(\chi_c \setminus \{X\}) = \sum_{i:C_i = X} (D_i - d_i)$.

4.1: Вычислить $n_\delta = \max\{[(|\chi_c| - p) * \sigma_e], 1\}$.

Упорядочить значения δ_X и выбрать подмножество $\chi_{e\lim} = \{X_1,...,X_{n_\delta}\} \subset \chi_c$ из n_δ точек, которым соответствуют минимальные значения δ_X.

4.2: Для каждого $j \in \{\overline{2,|\chi_{e\lim}|}\}$, если $\exists k \in \left\{ \overline{1,j-1} \right\}: L(X_j, X_k) < L_{\min}$, то удалить X_j из $\chi_{e\lim}$.

4.3: Присвоить $\chi_c = \chi_c \setminus \chi_{e\lim}$.

4.4: Перераспределить векторы данных между ближайшими к ним центрами.

$$C_i = \arg\min_{X \in \chi_c} L(X, A_i) \forall i = \overline{1,N}.$$

4.5. Для каждого $X \in \chi_c$, если $\exists i \in \left\{ \overline{1,N} \right\}: C_i = X$ и $C_i^* \neq X$, то пересчитать центр X* кластера $C_X^{clust} = \{A_i \mid C_i^* = X, i = \overline{1,N}\}$. Присвоить $\chi_c = (\chi_c \setminus \{X^*\}) \bigcup \{X\}$.

5: Перейти к Шагу 2.

Экспериментальным путем авторами [1] определены оптимальные значения параметров σ_e=0.25 и $L_{\min} = \min_{X \in \chi_c} \{\max\{L(X, X_j), L(X, X_k)\}\}$.

Большинство алгоритмов для решения задачи k-средних и p-медианной задачи, таких как ALA-процедура или весьма эффективный ГА с рекомбинацией подмножеств фиксированной длины [202] требуют, чтобы число кластеров p было известно. Другие алгоритмы, как например X-means [203], выбирают наилучшее значение p в соответствии со специальным критерием. Выбор адекватного критерия – отдельная сложная задача. В [124] предложена модификация жадной эвристической процедуры. Данная процедура последовательно исключает элементы из решения. Элементами в данном случае являются центроиды/центры/медоиды в зависимости от постановки задачи, общее название – главные точки, вокруг которых формируются кластеры – подмножества элементов, для которых одна из главных точек является ближайшей. После достижения требуемого числа кластеров p процесс исключения кластеров продолжается, и алгоритм продолжает фиксировать наилучшие известные значения целевой функции для каждого числа кластеров вплоть до 2 кластеров. Таким образом, можно получить решения серии задач с $p = \overline{2, p_{max}}$. При этом максимальное предполагаемое значение числа кластеров $p_{\max}$ все же должно быть

известно.

Алгоритм 2.4 Генетический алгоритм с жадной эвристикой для решения серии задач с $p \in \{2, p_{max}\}$

1. Инициализация популяции из N_{pop} особей. Каждая особь является множеством из p_{max} центров (обозначим их X_i). Присвоить $F_{new,j} = +\infty$ для каждого $j \in \{1, N_{pop}\}$ Инициализировать массивы значений целевой функции $F_k^* = +\infty$ и лучших решений $X_k^* = \{\}$ для каждого $k \in \{2, p_{max}\}$.

2. Выбрать случайным образом $j_1, j_2 \in [1, N_{pop}]$, $j_1 \neq j_2$

3. $X_{new} = X_{j_1} \bigcup X_{j_2}$; //(получение объединенного решения)

4. Пока $|X_{new}| > p_{max}$:

4.1. Выбрать элемент j, такой, чтобы его исключение давало наименьший прирост целевой функции: $j = \arg\min_{j \in X_{new}} F(X_{new} \setminus \{j\})$

4.2. $X_{new} = X_{new} \setminus \{j\}$. Следующая итерация 4.

5. Присвоить $F_{new} = 0$; $X^* = X_{new}$.

6. Пока $|X_{new}| > 2$:

6.1. Присвоить $F_{new} = F_{new} + f(X_{new})$; $k = |X_{new}|$; $F_k = f(X_{new})$; если $F_k < F_k^*$, то присвоить $F_k^* = F_k$;

6.2. Выполнить шаги 4.1 и 4.2 для X_{new}. Следующая итерация цикла 6.

7. Выбрать j_3 с использованием турнирного замещения по значению $F_{new,j}$. Присвоить $F_{j_3} = F_{new}$; $X_{j_3} = X^*$, $F_{new.j_3} = F_{new}$.

8. Проверить условия останова, перейти к шагу 2.

Как показано в [204], такой алгоритм способен давать весьма точные результаты решения задач k-средних и p-медианной задачи сразу для серии задач.

В данном алгоритме, в отличие от других ГА, рассмотренных выше, функция полезности F_{new} отличается от целевой функции задачи. Отметим, что, поскольку решается сразу серия задач, то количество целевых функций соответствует количеству решаемых задач. Мы складываем целевые функции для каждого значения p и получаем новую функцию полезности. Тем не менее, преимущества данного алгоритма наблюдаются лишь для решений с числом

кластеров, близким к p_{max}.

Для непрерывных p-медианных задач в сочетании с разными способами локального поиска также получены хорошие результаты для задач с большим числом кластеров, близким к p_{max}. Для задач с числом кластеров, значительно отличающимся от p_{max}, метод начинает давать худшие результаты в сравнении с другими методами. Авторы [204] предлагают ГА с жадной эвристикой останавливать не при p=2, а раньше, примерно на значении $p=p_{max}/3$, а остальные решения получать, например, новым запуском алгоритма с меньшим значением p_{max}. Следует отметить, что и в этом случае результаты существенно уступают результатам ГА с жадной эвристикой для решения единственной задачи для $p<p_{max}/1{,}5$.

В [125] нами предложена простая модификация Алгоритма 2.4 с размером популяции, монотонно растущим с номером итерации.

Алгоритм 2.5 ГА с жадной эвристикой и динамической популяцией для решения серии задач с $p\in\{2, p_{max}\}$

1. Инициализация начальной популяции начального размера из $N_{\text{рорнач}}$ особей. Каждая особь является множеством из p_{max} центров (обозначим их X_i). Присвоить $F_{new,j}= +\infty$ для каждого $j\in\{1, N_{рорнач}\}$ Инициализировать массивы значений целевой функции $F^*_k = +\infty$ и лучших решений $X^*_k=\{\}$ для каждого $k\in\{2, p_{max}\}$. N_{iter}=0.

2. $N_{iter}=N_{iter}+1$; $N_{pop} = \max\{N_{рорнач};\lceil\sqrt{1+N_{iter}}\rceil + 2\}$; Если N_{pop} изменилось, то инициализировать особь X_{Npop} аналогично шагу 1. Выбрать случайным образом $j_1, j_2\in [1, N_{pop}]$, $j_1\neq j_2$;

3. $X_{new} = X_{j_1} \cup X_{j_2}$;

4. Пока $|X_{new}| > p_{max}$:

4.1. Выбрать элемент j, такой, чтобы его исключение давало наименьший прирост целевой функции: $j = \arg\min_{j\in X_{new}} F(X_{new}\setminus\{j\})$

4.2. $X_{new}=X_{new}\setminus\{j\}$. Следующая итерация 4.

5. Присвоить F_{new}=0; $X^*=X_{new}$.

6. Пока $|X_{new}|>2$:

6.1. Присвоить $F_{new}=F_{new}+f(X_{new})$; $k=|X_{new}|$; $F_k=f(X_{new})$; если $F_k<F_k^*$, то присвоить $F_k^*=F_k$;

6.2. Выполнить шаги 4.1 и 4.2 для X_{new}. Следующая итерация цикла 6.

7. Выбрать j_3 с использованием турнирного замещения по значению $F_{new,j}$.

Присвоить $F_{j_3}=F_{new}; X_{j_3}=X^*, F_{new.j_3}=F_{new}$.

8. Проверить условия останова, перейти к шагу 2.

В Таблице 2.1 приведены результаты работы алгоритмов с полным объединенным (АПОР) и частичным объединенным решением (АЧОР) [123] для разных размеров популяции по данным отбраковочных испытаний ИС 2Д522Б (N=3711, d=10, метрика l_1, p=10, p_{max}=20, t=1 мин), испытаний ИС 140УД25 (N=523 d=42, метрика l_1, p=10, p_{max}=20, t=1 мин) и классического набора данных MissAmerica (N=6480, d=16, метрика l_2^2, p=75, p_{max}=100, t= 2,5 мин).

Таблица 2.1. Результаты работы алгоритмов с полным и частичным объединенным решением для разных размеров популяции.

Размер попул.	Пар-р	2Д522Б (АПОР)	140УД25 (АПОР)	MissAmerica (АПОР)	2Д522Б (АЧОР)	140УД25 (АЧОР)	MissAmerica (АЧОР)
5	*F(x)*	11778,05	980,67	755822,05	11777,99	979,37	755845,47
	σ	0,134	4,757	1597,645	0,127	5,030	591,256
10	*F(x)*	11778,02	976,64	755506,27	11778,01	976,77	755929,18
	σ	0,137	2,641	995,725	0,134	3,546	1466,276
20	*F(x)*	11777,97	975,29	756066,50	11777,95	976,15	756506,73
	σ	0,117	1,419	898,331	0,168	2,453	519,380
50	*F(x)*	11778,00	974,81	756960,20	11777,93	974,76	756969,27
	σ	0,119	0,239	1155,526	0,080	0,283	825,028
100	*F(x)*	11791,46	984,98	758777,26	11789,00	989,38	759518,74
	σ	17,304	6,496	1897,579	13,680	8,012	1601,17
лучшее	*F(x)*	11777,97	974,81	755506,27	11777,93	974,76	755845,47
	σ	0,117	0,239	995,725	0,080	0,283	591,256
среднее	*F(x)*	11780,70	978,478	756626,456	11780,176	979,286	756953,878
	σ	0,1248	1,859	1128,5904	0,1178	2,319	798,6392

Примечание: F(x) значение целевой функции, σ – стандартное отклонение

В Таблице 2.2 приведены результаты работы предложенного алгоритма с динамической популяцией в сравнении с лучшими результатами работы алгоритмов с полным объединенным и частичным объединенным решением. Для сравнения в таблицу включены результаты полученные мультистартом процедуры

k-средних (ALA-процедуры) и генетическим алгоритмом с рекомбинацией подмножеств фиксированной длины, результаты испытаний ИС 1526ТЛ1 (N=1234, d=120, метрика такси $L(X,Y)=\max\{||X-Y||,1\}$, описанная в Главе 3, p=10, p_{max}=10, t=2 мин), а также набор данных Ionosphere из репозитория UCI (N=351, d=34, метрика l_2^2, p=3, p_{max}=10, t=40 сек).

Таблица 2.2. Сравнение результатов работы алгоритмов.

Задача		АПОР лучшее	АЧОР лучшее	АПОР среднее	АЧОР среднее	АДП
2Д522Б, p=10	$F(x)$	11777,97	11777,93	11780,70	11780,176	11778,04
	σ	0,117	0,080	0,1248	0,1178	0,131
2Д522Б, p=15	$F(x)$	9452,69	9449,48	9460,06	9461,34	9453,94
	σ	3,395	3,870	6,092	12,532	1,958
140УД25, p=10	$F(x)$	974,81	974,76	978,478	979,286	974,77
	σ	0,239	0,283	1,859	2,319	0,283
140УД25, p=20	$F(x)$	660,17	659,91	667,92	671,85	659,84
	σ	1,016	0,416	6,0017	6,712	0,243
MissAmer p=75	$F(x)$	755506,3	755845,47	756626,456	756953,878	756019,80
	σ	995,725	591,256	1128,5904	798,6392	566,549
MissAmer p=50	$F(x)$	827604,5	827866,55	828233,87	829137,28	827904,01
	σ	925,538	621,880	1069,390	875,159	366,072
1526ТЛ1, p=10	$F(x)$	3440,06	3440,15	3446,73	3456,26	3442,64
	σ	1,849	1,569	6,990	14,833	3,881
Ion, p=3	$F(x)$	8453,95	8452,32	8463,86	8467,78	8454,93
	σ	4,533	3,292	9,678	10,837	4,747

Примечание: АПОР – лучшие результаты алгоритма с полным объединенным решением, АЧОР – лучшие результаты алгоритма с частичным объединенным решением (см. Таблицу 2.1), АДП – алгоритм с динамической популяцией ($F(x)$ значение целевой функции, σ – стандартное отклонение)

Как показывают эксперименты (Таблицы 2.1 и 2.2), такой подход позволяет не задумываться об оптимальном размере популяции, но никак не решает проблему низкой точности результатов при $p<p_{max}/1{,}5$.

Идею для дальнейшего развития этого подхода мы почерпнули из нашего алгоритма для решения задачи псевдобулевой оптимизации о загрузке производственных мощностей литейно-прокатного производства.

2.2. Использование генетического алгоритма с жадной эвристикой с хромосомами переменной длины на примере литейного производства

Рассмотрим другую задачу [205], не связанную с автоматической группировкой объектов, в качестве примера эффективного использования

генетических алгоритмов метода жадных эвристик с переменной длиной хромосомы.

В производственном цикле современного промышленного предприятия наряду с заказами массового или крупносерийного производства имеют место и заказы мелкосерийного и даже единичного характера, носящие нерегулярный характер.

Нерегулярность заказов и временные затраты перенастройки оборудования под конкретный заказ вызывают необходимость непрерывного процесса составления плана загрузки оборудования предприятия. Это можно рассмотреть на примере литейного производства.

Каждое литейное отделение имеет специализацию по видам продукции. Существует план по выполнению определенного количества заказов к заданному сроку. Каждый заказ характеризуется видом продукции, объемом и сроком выполнения. Предприятие работает в три смены. Перенастройка оборудования на другой вид продукции занимает одну смену.

Необходимо сформировать график загрузки производственных мощностей так, чтобы выполнить все заказы при минимальном числе перенастроек оборудования. При этом должна быть обеспечена минимальная общая загрузка производственного комплекса в течение каждых суток составляемого производственного графика. Также необходимо учитывать жесткие требования для затрат временных и вычислительных ресурсов на расчет.

Подобную задачу логично решать, как задачу размещения на сети, а именно как p-медианную задачу.

В [206] рассматриваемая задача решается как задача псевдо-булевой оптимизации (задача с булевыми переменными). Применена следующая модель. Пусть имеется K производственных линий для выпуска L видов продукции. Производительность всех производственных линий одинакова, для заданного l-го вида продукции линия может произвести V_l единиц продукции за смену при трех сменах в сутки. Требуется построить график с указанием вида продукции, производимой каждой производственной линией.

Для некоторые производственных линий доступен лишь ограниченный ассортимент продукции. Вводится матрица Z булевых констант z_{kl}, k=1...K, l=1...L, равных 1, если k-я линия может производить l-й вид продукции и 0 — в противном случае. Для каждого вида продукции установлен производственный план в объеме W_l единиц (l=1...L), который должен быть выполнен за T_l суток. Кроме того, установлена минимальная суммарная загруженность производственного комплекса в сутки в объеме W_{min} единиц продукции.

Виды продукции объединены в C классов M_c, $c = 1...C$. Смена класса продукции на производственной линии требует технологических операций по перенастройке продолжительностью в одну смену, в ходе которых выпуск продукции невозможен. Возможность безостановочной смены продукции с вида l на вид r описывается симметричной булевой матрицей $C_{l,r}$ размерности LxL: значение $C_{l,r}$ =1 означает необходимость останова производственной линии при смене продукции с вида l на вид r.

Вводятся булевы переменные $y_{i,k,l}$, которые принимают значение 1, если график предусматривает выпуск l-го вида продукции k-й производственной линией на i-е сутки.

График на I суток требуется составить так, чтобы при условии выполнения плана выпуска по видам продукции и срокам, с учетом требования минимальной загруженности, требовалось минимальное число изменений видов продукции. Таким образом, в булевых переменных задача формулируется следующим образом:

$$min \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} y_{i,k,l}(1-y_{(i-1),k,l}); \qquad (2.1)$$

$$V_l \sum_{i=1}^{T_l}\sum_{k=1}^{K}(3\cdot y_{i,k,l} - y_{i,k,l}\cdot(1-y_{(i-1),k,l})\cdot \cdot \sum_{r=1}^{L} y_{(i-1),k,r} C_{r,l}) \geq W_l \ \forall l = \overline{1,L}; \qquad (2.2)$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L} V_l (3 \cdot y_{i,k,l} - y_{i,k,l} \cdot (1 - y_{(i-1),k,l}) \cdot \cdot \sum_{r=1}^{L} y_{(i-1),k,r} C_{r,l}) \geq W_{min} \forall i = \overline{1,I}; \quad (2.3)$$

$$\sum_{l=1}^{L} y_{i,k,l} \leq 1 \forall i = \overline{1,I}, k = \overline{1,K}; \quad (2.4)$$

$$y_{i,k,l} \leq z_{k,l} \ \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}; \quad (2.5)$$

$$y_{i,k,l} \in \{0,1\} \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}. \quad (2.6)$$

Приведенная выше модель описывает ситуацию, когда все виды продукции относятся к разным классам, и для любой замены продукции требуется одна смена. Для учета возможности замены выпускаемой продукции без остановки производства в выражения (2.2) и (2.3) требуется добавить дополнительный множитель $\sum_{r=1}^{L} y_{(i-1),k,r} C_{r,l}$.

Модель также предполагает, что в начальный момент производство на всех линиях остановлено ($y_{0,k,l} = 0 \ \forall k = \overline{1,K}, l = \overline{1,L}$). В реальных задачах в условиях регулярного планирования непрерывного производства такая ситуация невозможна. Любая производственная линия в любой момент времени настроена на выпуск какой-либо продукции, за исключением случаев аварий, ремонтов и прочих ситуаций, когда данная линия исключается из производственного графика. Следовательно, константы $y_{0,k,l}$ не равны нулю.

В дополнение к переменным $y_{i,k,l}$ авторами [206] введены дополнительные переменные $x_{i,k,l}$:

$$x_{i,k,l} = y_{i,k,l}(1 - y_{(i-1),k,l}) \ \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}.$$

Значения переменных $y_{i,k,l}$ могут быть определены из значений $x_{i,k,l}$ с помощью специального алгоритма:

$$y_{i,k,l} = \begin{cases} x_{i,k,l}, & \sum_{r=1}^{L} x_{i,k,r} = 0, \\ y_{(i-1),k,l}, & \sum_{r=1}^{L} x_{i,k,r} > 0, \end{cases} \quad \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}.$$

Математически p-медианная задача [207] представляет собой следующее. Есть некоторая сеть (связный граф) $G = (V, E)$, где V множество узлов (вершин), E множество ребер $E_{i,j}, i, j \in V$ соединяющих их попарно. Для каждого ребра определены: его номер, длина $L_{i,j}$, мера расстояния $D(i, j)$ между любой парой узлов i и j определена как минимальный путь между этими узлами (под длиной пути между двумя узлами сети будем понимать сумму длин ребер этого пути). Цель задачи – выбрать множество узлов сети (вершин графа) S с мощностью p:

$$\arg \min_{S \subset V, |S| = p} f_G(S) = \arg \min_{S \subset V, |S| = p} \sum_{i \in V} \min_{j \in S} D(i, j). \quad (2.7)$$

Представим производственный график как трехмерную решетку в дискретных координатах с осями i, k, l. отложим дни по оси i, производственные линии по оси k, виды продукции по оси l.

Каждый узел такой сети описывается тремя координатами (i, k, l). Матрица булевых координат $[x_{i,k,l}, i = \overline{i,I}, k = \overline{i,K}, l = \overline{i,L}]$ связана с множеством χ узлов сети-решетки, которая описывает график:

$$x_{i,k,l} = \begin{cases} 1, & (i,k,l) \in \chi, \\ 0, & (i,k,l) \notin \chi. \end{cases}$$

Задача состоит в выборе множества χ узлов сети (вершин графа) с минимальной мощностью и удовлетворяющего условиям (2.4) – (2.6). Выбранному узлу соответствует значение булевой переменной $x_{i,k,l}$ =1. Таким образом, задача состоит в выборе минимально мощного множества точек переключения в графике производственного планирования.

Для экономии оперативной памяти, снижения времени доступа и упрощения вычислений имеет смысл перейти к целочисленным переменным, учитывая условие (2.4).

Введем переменные $y'_{i,k}$ принимающие значения от 0 до L. Значение $y'_{i,k} = l$

означает производство l-го типа продукции на k-ой производственной линии в i-ый день ($y'_{i,k} = 0$ означает остановку производства). Далее, $y'_{0,k}, k = \overline{1,K}$ целочисленные константы в диапазоне от 0 до L, показывающие тип выпускаемой продукции, на который настроена каждая из K производственных линий в начальный момент времени. Таким же образом введем дополнительные переменные $x'_{i,k}$:

$$x'_{i,k} = \begin{cases} y'_{i,k}, & y'_{i,k} \neq y'_{(i-1),k}, \\ 0, & y'_{i,k} = y'_{i-1,k}, \end{cases} \quad \forall i = \overline{1,I}, k = \overline{1,K}.$$

Значения $y'_{i,k}$ могут быть получены из $x'_{i,k}$:

$$y'_{i,k} = \begin{cases} y'_{(i-1),k}, & x'_{i,k} = 0, \\ x'_{i,k}, & x'_{i,k} \neq 0 \end{cases} \quad \forall i = \overline{1,I}, k = \overline{1,K}. \qquad (2.8)$$

Значения исходных булевых переменных могут быть получены:

$$x_{i,k,l} = \begin{cases} 1, & x'_{i,k} = l, \\ 0, & x'_{i,k} \neq l, \end{cases} \quad \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}, \qquad (2.9)$$

$$y_{i,k,l} = \begin{cases} 1, & y'_{i,k} = l, \\ 0, & y'_{i,k} \neq l, \end{cases} \quad \forall i = \overline{1,I}, k = \overline{1,K}, l = \overline{1,L}. \qquad (2.10)$$

При этом условие (2.4) становится ненужным, а условия (2.2)–(2.3) могут быть записаны в более удобной форме:

$$V_l \sum_{i=1}^{T_l} \sum_{k=1}^{K} y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) \geq W_l \ \forall l = \overline{1,L}, \quad (2.11)$$

$$\sum_{k=1}^{K} \sum_{l=1}^{L} V'_{y'_l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}}) >= W_{min} \ \forall i = \overline{1,I}. \quad (2.12)$$

Здесь $C'_{l,r}$ это матрица с булевыми значениями $C_{l,r}$, дополненная строкой и столбцом с нулевыми индексами ($l, r \in \{\overline{0,L}\}$):

$$C'_{l,r} = \begin{cases} C_{l,r}, & l > 0, r > 0, \\ 0, & r = 0, \\ 1, & l = 0, r > 0, \end{cases}$$

V' вектор норм производства за смену V с дополнительным нулевым элементом:

$$V'_l = \begin{cases} V_l, & l > 0 \\ 0, & l = 0. \end{cases}$$

Предложенный генетический алгоритм не сохраняет значения булевых переменных $x_{i,k,l}$ и $y_{i,k,l}$. При необходимости они определяются "на лету", согласно (2.9) и (2.10).

Согласно нотации принятой для p-медианных задач, промежуточные решения представлены как множества χ выбранных узлов (медиан) сети. Эти множества также удобно представлять как матрицы целочисленных переменных:

$$x'_{i,k} = \begin{cases} l, & (i,k,l) \in \chi, \\ 0, & (i,k,l) \notin \chi. \end{cases} \qquad (2.13)$$

Таким образом, включение узла (i, k, l) в множество сводится к присвоению элементу матрицы $x'_{i,k}$ значения l, а исключение к присвоению нулевого значения.

Идея генетического алгоритма с жадной эвристикой для p-медианной задачи на сети [208] состоит в следующем. Есть некоторый массив ("популяция") решений задачи, каждое из которых представляет собой множество из p выбранных узлов сети. Необходимо случайным образом выбрать два решения (родительские особи). Множество узлов, в которое объединяются родительские множества, рассматривается как промежуточное решение. Затем, мы по одному исключаем узлы, так чтобы прирост целевой функции (2.7) был минимален, до тех пор, пока не останется p узлов.

Алгоритм 2.6 Генетический алгоритм с жадной эвристикой для p-медианной задачи.

Дано: количество узлов p, множество всех узлов сети V, целевая функция (метрика расстояния) $f_{\mathsf{G}}(\cdot)$

Шаг 1. Сгенерировать начальный массив множеств узлов $A = \{\chi_j\} = \{\mathsf{V}_{m_1},...,\mathsf{V}_{m_p}\}$, $j = \overline{1, N}$, p вершин в каждом множестве. Здесь N количество особей (размер популяции) в алгоритме.

Шаг 2. Случайным образом выбрать два индекса элементов массива A (родительские особи) $j_1, j_2 = \overline{1, N}$, $j_1 \neq j_2$. Случайным образом выбрать $j_3 \in w$. Здесь w некоторое множество индексов особей (элементов массива A), которые оценены как "плохие". В [208] всегда выбирается особь с худшим (максимальным) значением целевой функции: $\chi_w = \{\arg\min_{S \in A} f_{\mathsf{G}}(S)\}$.

Шаг 3. Присвоить $\chi_{j_3} = \chi_{j_1} \cup \chi_{j_2}$;

Шаг 4. Пока $|\chi_{j_3}| > p$ выполнить:

Шаг 4.1. Присвоить $f_{best} = +\infty$, $FOUND = 0$.

Шаг 4.2. Для каждого узла $\mathsf{V} \in \chi_{j_3}$ выполнить:

Шаг 4.2.1. $\xi = \chi_{j_3} \setminus \{\mathsf{V}\}$.

Шаг 4.2.2. Если $f_{\mathsf{G}}(\xi) < f_{best}$, то присвоить $f_{best} = f_{\mathsf{G}}(\xi)$, $\chi_{best} = \xi$, $FOUND = 1$.

Шаг 4.2.3. Следующая итерация цикла 4.2.

Шаг 4.3. Присвоить $\chi_{j_3} = \chi_{best}$. Следующая итерация цикла 4.

Шаг 5. Проверить условие останова, если не выполнено, то перейти к Шаг 2.

Несмотря на определенное сходство с p-медианной задачей на сети, наша задача имеет несколько существенных отличий, таких как:

Свойство 1. Целевая функция p-медианной задачи монотонно убывает, функция (2.1) в нашей задаче монотонно возрастает (относительно переменных $x_{i,k,l}$ и $y_{i,k,l}$).

Свойство 2. Решая p-медианную задачу [208], мы имеем дело с безусловной оптимизацией. Здесь мы рассматриваем условную оптимизацию. Левые части ограничений (2.2) и (2.3) – монотонно возрастающие функции переменных $y_{i,k,l}$, а

относительно $x_{i,k,l}$ в общем случае они не монотонны: $x_{i,k,l}$ могут меняться от 1 до 0 (что соответствует исключению узла) так, что значения левых частей некоторых неравенств (2.2) и (2.3) могут уменьшаться, а других – возрастать.

Свойство 3. Число вершин p в p-медианной задаче известно, а наша цель – минимизировать число вершин.

Благодаря Свойству 1, исключение любой вершины из множества вершин (Шаг 4.2.1 Алгоритма 2.6) влечет за собой гарантированное уменьшение значения целевой функции и проверка в Шаге 4.2.2 становится ненужной. В тоже время, благодаря Свойству 2, проверка промежуточного решения на соответствие ограничениям остается актуальной. Благодаря Свойству 3 условия цикла 4 Алгоритма 2.6 так же становится ненужным. Исключение вершин из промежуточного решения следует повторять до тех пор, пока эта операция дает некоторые улучшения относительно соответствия промежуточного решения ограничениям.

Ограничения (2.4) и (2.5) могут быть учтены на этапе генерации начальных и промежуточных решений. Ограничения (2.2) и (2.3) должны быть преобразованы в штрафную функцию.

В рассматриваемом здесь алгоритме предлагается замена ограничения (2.2), которому для целочисленных переменных соответствует ограничение (2.11) на следующее ограничение:

$$f_2(\chi) = -\sum_{l=1}^{L} max\{0, W_l - V_l \sum_{i=1}^{T_l}\sum_{k=1}^{K} y_{i,k,l}(3 - C'_{y'_{(i-1),k}, y'_{i,k}})\} = 0, \quad (2.14)$$

а ограничение (2.3), которому соответствует ограничение (2.12), на ограничение:

$$f_3(\chi) = -\sum_{i=1}^{I} max\{0, W_{min} - \sum_{k=1}^{K}\sum_{l=1}^{L} V'_{y'_l}(3 - C'_{y'_{(i-1),k}, y'_{i,k}})\} = 0. \quad (2.15)$$

Функции $f_2(\cdot)$ и $f_3(\cdot)$ определены как функции множества χ. Значения целочисленных переменных $y'_{i,l}$ и булевых переменных $y_{i,k,l}$ определяются по формулам (2.13), (2.8), (2.10).

Также необходимо определить целевую функцию

$$f(\chi) = |\chi| = \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{L} x_{i,k,l},$$

значения булевых переменных $x_{i,k,l}$ определяются по формулам (2.13), (2.8), (2.9).

Экономический смысл функции $f_2(\chi)$ – общее количество продукции, выпускаемой с отставанием от плана, функции $f_3(\chi)$ – суммарное недовыполнение суточного минимума выпускаемой продукции.

Функции $f_2(\chi)$ и $f_3(\chi)$ имеют схожий экономический смысл, одинаковую размерность (число единиц продукции), поэтому можно применить единую штрафную функцию (аддитивной свертку этих функций) и соответствующего объединенного ограничения:

$$f_4(\chi) = f_2(\chi) + f_3\chi = 0. \quad (2.16)$$

Задачу поиска решения, удовлетворяющего ограничению (2.16) и, соответственно, ограничениям (2.2) и (2.3) исходной задачи, можно рассматривать как задачу минимизации функции $f_4(\chi)$.

Генетический алгоритм не требует единственного критерия для оценки решений. Решения могут оцениваться по двум критериям – значению целевой функции $f(\chi)$ и значению штрафной функции $f_4(\chi)$. Забегая вперед, отметим, что использование свертки (2.16) не дает заметных преимуществ перед использованием трех отдельных критериев $f(\cdot)$, $f_2(\cdot)$ и $f_3(\cdot)$, недостатки такого подхода также не выявлены. В то же время трехкритериальный подход к оценке и сравнению решений в популяции генетического алгоритма может быть применен к случаям, когда задача имеет штрафные функции $f_2(\cdot)$ и $f_3(\cdot)$ с различной размерностью.

Последовательное исключение узлов в основе алгоритма соответствует минимизации целевой функции $f(\cdot)$. Таким образом, нет нужды оценивать и сравнивать значения целевой функции в Шагах 4–4.3 при выполнении Алгоритма 2.6.

Общую схему алгоритма для задач (2.1)–(2.6) можно описать так.

Алгоритм 2.7 Генетический алгоритм с жадной эвристикой для планирования непрерывного производства

Шаг 1. Сгенерировать начальный массив множеств узлов сети, представленных тройками индексов (i, k, l): $A = \{\chi_j\} = \{(i_1, k_1, l_1), ..., (i_{p_j}, k_{p_j})\}$, $j = \overline{1, N}$. Здесь N число "особей" популяции генетического алгоритма. Количество узлов p_j может быть различно в каждом элементе массива A.

Шаг 2. Случайным образом выбрать два индекса родительских "особей" $j_1, j2 \in \{\overline{1, N}\}$, $j_1 \neq j_2$. Случайным образом выбрать $j_3 \in w$. Здесь w некоторое множество индексов особей (элементов массива A), которые оценены как "плохие". Методика оценки приведена ниже.

Шаг 3. Присвоить $\chi_{j_3} = \chi_{j_1} \cup \chi_{j_2}$.

Шаг 5. Для каждого узла $\mathsf{V} = (i_1, k_1, l_1) \in \chi_{j_3}$ выполнить:

Шаг 5.1. Если $\exists V_2 = (i_2, k_2, l_2) \in \chi_{j_3} : l_2 \neq l_1, i_2 = i_1, k_2 = k_1$ то случайным образом с равной вероятностью выбрать индекс $l \in \{l_1, l_2\}$, присвоить $\chi_{j_3} = \chi_{j_3} \setminus \{(i_1, k_1, l)\}$.

Шаг 5.2. Следующая итерация цикла 5.

Шаг 6. Вычислить матрицы булевых и целочисленных переменных $[x_{i,k,l}]$, $[y_{i,k,l}]$, $[x'_{i,k}]$ и $[y'_{i,k}]$, которые соответствуют множеству χ_{j_3}, используя формулы (2.13), (2.8)–(2.10). *Отметим, что этот шаг и аналогичные шаги ниже сводятся к вычислению* $[y'_{i,k}]$ *согласно выражению (2.8), потому что множества* χ_j *представлены матрицами целочисленных переменных.*

Шаг 7. Присвоить $FOUND = 0$.

Шаг 8. Случайным образом разместить узлы множества χ_{j_3}, для каждого узла $\mathsf{V} = (i', k', l') \in \chi_{j_3}$ выполнить:

Шаг 8.1. Присвоить $\xi = \chi_{j_3} \setminus \{\mathsf{V}\}$. Вычислить матрицы булевых и целочисленных переменных $[x^{\xi}_{i,k,l}]$, $[y^{\xi}_{i,k,l}]$, $[x'^{\xi}_{i,k}]$ и $[y'^{\xi}_{i,k}]$, которые соответствуют

множеству ξ. Если $f_4(\xi) < f_4(\chi_{j_3})$ то присвоить $\chi_{j_3} = \xi$, $FOUND = 1$, пересчитать соответствующие матрицы булевых и целочисленных переменных $[x_{i,k,l}]$, $[y_{i,k,l}]$, $[x'_{i,k}]$ и $[y'_{i,k}]$. Перейти к следующей итерации цикла 8. *Отметим, что исключение вершины* $V = (i', k', l')$ *из множества и перерасчет соответствующих целочисленных переменных сводится к обнулению* $x'_{i',k'}$ *и перерасчету* $y'_{i'',k'}$ *для* $i'' = \overline{i', T_l}$.

Шаг 8.2. выполнить процедуры локального поиска в окрестности узла V. Эти нижеописанные опциональные процедуры значительно повышают эффективность алгоритма.

Шаг 8.3. Следующая итерация цикла 7.

Шаг 9. Если $FOUND = 1$ то вернуться к Шагу 7 (начать цикл заново).

Шаг 10. Проверить условия останова, если они не выполняются, то перейти к Шагу 2.

Также как в Алгоритме 2.6, этот алгоритм последовательно исключает по одной вершине из множества χ_{j_3} (Шаг 4.2.1 Алгоритма 2.6 и Шаг 8.1 Алгоритма 2.7). В случае Алгоритма 2.6 мы заранее знаем конечное число вершин *p* (Шаг 4), а Алгоритме 2.7 мы исключаем вершины, пока это улучшает значение штрафной функции $f_4(\cdot)$ (см. Шаг 7, 8.1, 9). Шаги 5–5.2 гарантируют выполнение условия 4. Представляя множества x_{j_1}, x_{j_2}, x_{j_3} как матрицы целочисленных переменных $[x'^{j_1}_{i,k}]$, $[x'^{j_2}_{i,k}]$, $[x'^{j_3}_{i,k}]$ мы сводим эти шаги к копированию:

$$x'^{j_3}_{i,k} = \begin{cases} max\{x'^{j_1}_{i,k}, x'^{j_2}_{i,k}\}, & min\{x'^{j_1}_{i,k}, x'^{j_1}_{i,k}\} = 0, \\ x'^{j_r}_{i,k}, & min\{x'^{j_1}_{i,k}, x'^{j_1}_{i,k}\} > 0. \end{cases}$$

Здесь $r \in \{1,2\}$ случайным образом выбирается для каждой пары $(i,k) : min\{x'^{j_1}_{i,k}, x'^{j_1}_{i,k}\} > 0$.

Для обеспечения соответствия каждого решения ограничениям (2.4)–(2.6) используется следующий алгоритм генерации начальной популяции:

Алгоритм 2.8 Создание особи для начальной популяции генетического

алгоритма.

Дано: число узлов сети *p*.

Шаг 1. Присвоить $x'_{i,k} = 0 \; \forall i = \overline{1, I}, k = \overline{1, K}$.

Шаг 2. Для $j = \overline{1, p}$ выполнить:

Шаг 2.1. Случайным образом выбрать $k \in \{\overline{1, K}\}$ (*здесь и ниже случайный выбор целого цисла из определенного множества предполагает равное математическое ожидание выбора каждого числа*).

Вычислить $Z_k = \sum_{l=1}^{L} z_{k,l}$, количество видов продукции, доступное для производства на *k*-ой производственной линии. Случайным образом выбрать $m \in \{\overline{1, Z_k}\}$. Найти индекс *m*-го ненулевого элемента *k*-го столбца матрицы $Z = [z_{k,l}]$, сохранить его в переменной *l*. Случайным образом выбрать $i \in \{\overline{1, T_l}\}$. Если $x'_{i,k} = 0$ то присвоить $x'_{i,k} = l$. Иначе повторить Шаг 2.1.

Шаг 2.2. Следующая итерация цикла 2.

Шаг 3. Вернуть матрицу целочисленных переменных $[x_{i,k}]$, которая представляет множество узлов сети χ.

В качестве параметра этого алгоритма выступает количество узлов сети *p*. На практике видно, что если *p* в различных вариантах исходной популяции варьируется в широких пределах (а диапазоне $\{\overline{L, K \cdot I/2}\}$), то достигаются наилучшие результаты. Есть возможность либо случайного выбора *p* из определенного множества для каждого генерируемого экземпляра начальной популяции, либо определения *p* по формуле $p = [L + j(K \cdot I/2 - L)/N]$. Результаты статистически равнозначны. Здесь *j* номер индекса генерируемого экземпляра, [·] – целая часть числа.

Размер массива (количество "особей") популяции генетического алгоритма это важный параметр. Недостаточное количество снижает качество решений. Мы эмпирически установили достаточность значения $N = I + K + L$, дальнейшее увеличение не улучшает результаты и приводит лишь к увеличению времени

вычисления.

Мы использовали процедуры локального поиска в окрестности узла V, в качестве вспомогательных процедур (Шаг 8.2 Алгоритма 2.7) для ускорения поиска допустимых решений. Под окрестностью в данном контексте мы понимаем множество узлов, отличающихся от V значением координаты l (соответствует переключению вида продукции) или отличающихся значением координаты i не более чем на 1 (перенос начала производства на день раньше или позже). Мы использовали три процедуры:

Процедура 1. (Шаг 8.2 of Алгоритма 2.7). Проводится для вершины $V = (i,k,l)$ при условии $V_l \sum_{i=1}^{T_l} \sum_{k=1}^{K} y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}} > W_l$ (перевыполнение плана по l-му виду продукции).

Шаг 1. Составить множество индексов видов продукции, удовлетворяющий условию $V_l \sum_{i=1}^{T_l} \sum_{k=1}^{K} y_{i,k,l} (3 - C'_{y'_{(i-1),k}, y'_{i,k}} < W_l$ (перевыполнение плана).

Шаг 2. Случайным образом выбрать индекс l_2 из этого множества.

Шаг 3. Вычислить $\xi = (\chi_{j_3} \setminus \{V\}) \cup \{(i,k,l_2)\}$. Если $f_4(\xi) < f_4(\chi_{j_3})$ то присвоить $\chi_{j_3} = \xi$, $FOUND = 1$.

Процедура 2. (Шаг 8.2 Алгоритма 2.7). Проводится для вершины $V = (i,k,l)$ при условии $i > 1$ и $(i-1,k,l) \notin \chi_{j_3}$.

Вычислить $\xi = (\chi_{j_3} \setminus \{V\}) \cup \{(i-1,k,l)\}$. Если $f_4(\xi) < f_4(\chi_{j_3})$ то присвоить $\chi_{j_3} = \xi$, $FOUND = 1$.

Процедура 3. (Шаг 8.2 Алгоритма 2.7). Проводится для вершины $V = (i,k,l)$ при условии $i < T_l$ $(i+1,k,l) \notin \chi_{j_3}$.

Вычислить $\xi = (\chi_{j_3} \setminus \{V\}) \cup \{(i+1,k,l)\}$. Если $f_4(\xi) < f_4(\chi_{j_3})$ то присвоить $\chi_{j_3} = \xi$, $FOUND = 1$.

Необходимо заметить, что включение этих процедур в алгоритм заметно

снижает время поиска допустимых решений. Эти процедуры проводят докальный поиск в окрестности каждого узла сети и заменяют узел V на узел из окрестности, если замена позволит улучшить значение штрафной функции $f_4(\cdot)$. Процедура 1 для узла $\mathrm{V} = (i,k,l)$ выполняет поиск в множестве узлов $\{(i,k,l') \mid l' = \overline{1,L}\}$, а Процедуры 2 и 3 в множестве узлов $\{(i',k,l) \mid i' = i \pm 1, 1 \le i' \le T_l\}$.

Применение локального поиска снижает разнообразие популяции. Это может быть частично скомпенсировано введением процедуры мутации, которая отсутствует в Алгоритме 2.6.

Процедура 4 (Шаг 4 Алгоритма 2.7). Мутация множества χ_{j_3}. Выполняется с некоторой вероятностью p_m.

Сгенерировать множество ξ с помощью Алгоритма 2.8 при $p = |\chi_{j_3}|$. Присвоить $\chi_{j_3} = \chi_{j_3} \cup \xi$.

Экспериментами выявлены оптимальные значения: $p_m \in [0{,}005, 0{,}03]$ если в Алгоритм 2.7 включены только Процедуры 2 и 3 и $p_m \in [0{,}01, 0{,}05]$ при наличии в Алгоритме 2.7 всех процедур локального поиска (Процедуры 1, 2, 3).

Алгоритм 2.7 создает новые "особи" (промежуточные решения – множества узлов) χ_{j_3}, заменяющие собой существующие "особи" с индексом j_3, которые случайным образом выбираются из множества "плохих" решений w. При определении этого множества нужно учитывать значения целевой функции $f(\cdot)$ и штрафной функции $f_4(\cdot)$, вычисленные согласно (2.14)–(2.16). Используется следующая процедура.

Процедура 5. Определение множества w.

Шаг 1. Присвоить $S_n = 0 \; \forall n = \overline{1,N}$.

Шаг 2. Присвоить $F_n = |\chi_n|$.

Шаг 3. Отсортировать массив F в порядке возрастания.

Шаг 4. Найти медиану $f' = F_{[N/2]}$.

Шаг 5. Для $n \in \{\overline{1,N}\} : | \chi_n | \leq f'$ присвоить $S_n = S_n + 1$.

Шаг 6. Присвоить $F_n = f_4(\chi_n) \; \forall n = \overline{1,N}$.

Шаг 7. Отсортировать массив F порядке возрастания.

Шаг 8. Найти медиану $f_4' = F_{[N/2]}$.

Шаг 9. Для $n \in \{\overline{1,N}\} : f_4(\chi_n) \leq f_4'$ присвоить $S_n = S_n + 2$.

Шаг 10. Для $n \in \{\overline{1,N}\} : f_2(\chi_n) = 0$ присвоить $S_n = S_n + 1$.

Шаг 11. Для $n \in \{\overline{1,N}\} : f_3(\chi_n) = 0$ присвоить $S_n = S_n + 1$.

Шаг 12. Найти наименьший индекс $n \in \{\overline{1,N}\} : \; f_4(\chi_n) = 0$ и $| \chi_n | = \underset{q \in \{1,N\}}{min} | \chi_q |$.

Если индекс, удовлетворяющий условиям существует, то присвоить $S_n = S_n + 10$.

Шаг 13. Скопировать массив S в массив S'. Отсортировать массив S' в порядке возрастания.

Шаг 14. Найти медиану $s' = S'_{[N/2]}$.

Шаг 15. Вернуть результат – множество индексов $w = \{n \in \{\overline{1,N}\} \mid S_n \geq s'\}$.

Процедура осуществляет принцип подсчета очков: при значении целевой функции лучше медианы, решение получает 1 очко; при значении штрафной функции $f_4(\cdot)$ лучше медианы – 2 очка; за нулевые значения штрафных функций $f_2(\cdot)$ и $f_3(\cdot)$ решение получает 1 очко; лучшее решение, удовлетворяющее ограничениям получает дополнительные 10 очков. Решения с числом очков не больше медианного значения считаются "плохими".

Процедура требует много вычислительных ресурсов, даже если значения целевой функции $f(\cdot)$ и штрафной функции $f_4(\cdot)$ хранить в массивах, а не вычислять каждый раз для каждой "особи". Поэтому, в предложенном алгоритме процедура выполняется на Шаге 2 после генерации начальной популяции и каждые $N/4$ шагов (определяет кандидатов на выбывание после смены четверти популяции). Когда в популяцию включается новая особь, мы полагаем, что $j_3 \notin w$

(новопоявившиеся особи не являются кандидатами на выбывание).

Алгоритм 2.7 является алгоритмом метода жадных эвристик [123]. При этом особи-решения в данном алгоритме представляют собой множества точек переключения видов производимой продукции, и эти множества могут иметь различную мощность. Таким образом, данный алгоритм является генетическим алгоритмом с жадной эвристикой, при этом являясь алгоритмом с переменной длиной хромосомы (решения). Высокая сравнительная эффективность данного алгоритма показывает перспективность применения алгоритмов с переменной длиной хромосомы в алгоритмах метода жадных эвристик. В следующем подразделе предлагается такой алгоритм для задач автоматической группировки.

2.3 Алгоритм с гетерогенной популяцией для задачи автоматической группировки объектов

Значение термина «генетический алгоритм с хромосомами переменной длины» часто зависит от точки зрения. Например, алгоритм дельта-кодирования [209] упоминался в литературе совместно с ГА с переменной длиной слова. Этот алгоритм начинается с выполнения стандартного ГА, пока совокупность не сходится к множеству сходных решений. Лучший индивидуум из этой первоначальной популяции сохраняется как промежуточное решение. Новая популяция инициализируется с использованием меньшего количества бит для каждой переменной. Оценка пригодности происходит после того, как гены от новых особей добавляются или вычитаются из тех, что были в промежуточном растворе. Если найдено лучшее решение, оно станет новым промежуточным решением. Этот процесс повторяется для ряда дельта итераций до тех пор, пока не будет найдено наилучшее возможное решение или не будет выполнено какое-либо другое условие остановки.

Изменение количества бит в переменной позволяет алгоритму дельта-кодирования искать различные пространства для решения одной задачи. Но алгоритм дельта-кодирования действительно представляет собой ГА с

фиксированной длиной хромосомы, который использует разные длины строк для всех хромосом в каждом прогоне (дельта-итерация). Он не изменяет размеры строк в ходе выполнения и не допускает различные длины строк в пределах популяции.

Генетический алгоритм Промотор/Терминатор или ptGA в [210] – еще один пример ГА с фиксированной длиной, который иногда маркируется как алгоритм с переменной длиной. Этот ГА разработан для проверки использования некодирующих областей (названных интронами). Хромосомы состоят из битовых строк фиксированной длины. Отдельные гены встроены в каждую хромосому, идентифицируются тегами и окружены последовательностями битов промотора и терминатора. Общая длина (размер) хромосомы больше, чем требуется для хранения всех генов. ГА использует как содержание генов, так и их расположение на хромосоме (без перекрывающихся кадров считывания). Допускаются дублированные гены, а также некодирующие биты, количество которых зависит от общего размера хромосомы с фиксированной длиной – чем больше хромосома, тем больше свободного места для некодирующих битов или дублирующих генов. Автор утверждает, что вставка некодирующих битов в ГА с фиксированной длиной ускоряет нахождение хороших решений.

Сюда также можно отнести и генетический алгоритм со скрытыми генами (HGGA), приведенный в [211]. При фиксированной длине хромосом в силу особенностей целевой функции на конкретной итерации некоторые гены считаются неэффективными, становятся скрытыми и не участвуют в генетических операциях. Аналогичным образом, в [123, 143], в некоторых вариантах генетического алгоритма метода жадных эвристик для задач автоматической группировки в скрещивании участвует лишь часть хромосомы, правда, в данном случае эта часть выбирается случайным образом.

Алгоритмы 2.2 и 2.4 оперируют популяцией решений («особей»), каждое из которых состоит из одинакового числа p_{max} элементов – центров (центроидов, медоидов) кластеров. В отличие от них, "особи" Алгоритма 2.7 – множества троек *(i,k,l)* различной мощности.

Практика показывает, что наилучшие результаты достигаются, если p в различных экземплярах исходной популяции варьируется в широких пределах – в диапазоне $\{\overline{L, K \cdot I / 2}\}$.

Эффективность алгоритмов метода жадных эвристик с частичным объединенным решением для некоторых задач обусловлена тем, что в этих алгоритмах в промежуточном решении число кластеров меньше, чем при полном объединении двух родительских особей. Развитием этой идеи является скрещивание в единой популяции решений с различным числом кластеров.

В [125] нами предложен следующий алгоритм, оперирующий популяцией с различными значениями p.

Алгоритм 2.9 ГА с гетерогенной популяцией для р-медианной задачи.

1. Инициализация начальной популяции начального размера из $N_{рорнач}$ особей. Каждая особь является множеством из p_{max} центров (обозначим их X_i). Присвоить $F_{new,j}= +\infty$ для каждого $j \in \{1, N_{рорнач}\}$ Инициализировать массивы значений целевой функции $F_k^* = +\infty$ и лучших решений $X_k^*=\{\}$ для каждого $k \in \{2, p_{max}\}$. $N_{iter}=0$.

2. $N_{iter}=N_{iter}+1$; $N_{pop} = \max\{N_{рорнач}; \lceil \sqrt{1+N_{iter}} \rceil + 2\}$; Если N_{pop} изменилось, то инициализировать особь X_{Npop} аналогично шагу 1. Выбрать случайным образом $j_1, j_2 \in [1, N_{pop}]$, $j_1 \neq j_2$;

3. ; $X_{new} = X_{j_1} \cup X_{j_2}$

4. Пока $|X_{new}| > p_{max}$;

4.1. Выбрать элемент j, такой, чтобы его исключение давало наименьший прирост целевой функции: $j = \arg\min_{j \in X_{new}} F(X_{new} \setminus \{j\})$

4.2 $X_{new}=X_{new}\setminus\{j\}$ Следующая итерация 4.

5. Выбрать случайным образом $p_{child} \in \{2, p_{max}\}$. Если $p_{child}>|X_{new}|$, то $p_{child}=|X_{new}|$;

6. $f_{child,\ |Xnew|}=f(X_{new})$;

7. Пока $|X_{new}|> p_{child}$

7.1. Выбрать элемент j, такой, чтобы его исключение давало наименьший прирост целевой функции: $j = \arg\min_{j \in X_{new}} f_{child}(X_{new} \setminus \{j\})$

7.2. $X_{new}=X_{new}\backslash\{j\}$. Следующая итерация 7.

8. Пока $|X_{new}|>2$:

8.1. Присвоить $f_{child,\ |Xnew|}= F(X_{new})$; $k=|X_{new}|$; $f_{k,\ |Xnew|}=f(X_{new})$; если $f_{k,\ |Xnew|}<F^*_k$, то присвоить $F^*_k = f_{k,\ |Xnew|}$;

8.2. Выполнить шаги 4.1 и 4.2 для X_{new}. Следующая итерация 8.

9. Выбрать $j_3 \in \{1, N_{pop}\}$ с использованием турнирного замещения (Алгоритм 2.10). Присвоить $X_{j_3} = X_{new}$; $f_{j_3,k} = f_{child}$

10. Проверить условия останова, перейти к шагу 2.

Алгоритм 2.10 Турнирное замещение для Алгоритма 2.9.

1. Выбрать случайно $j_1 \in \{1, N_{pop}\}$, $j_2 \in \{1, N_{pop}\}$, $j_1 \neq j_2$; Выбрать случайно $p_{compar} \in \left\{2, \min\left\{|X_{j_1}|, |X_{j_2}|\right\}\right\}$; если $f_{j_1,p_{compar}} < f_{j_2,p_{compar}}$ то $j_3=j_2$, иначе $j_3=j_1$.

Возвращаясь к спектру методов контроля размеров хромосом, упомянутых в Главе 1, можно сказать, что данный алгоритм представляет собой сочетание подхода со случайной длиной хромосомы и подхода с ограниченной длиной. Ограничение вытекает из сути практической задачи автоматической группировки электрорадиоизделий, ведь очевидно, что количество производственных партий в сборной партии не может превышать определенного, эмпирически установленного количества, которое опять-таки очевидно меньше общего количества группируемых изделий.

2.4 Вычислительные эксперименты

В целях сравнения новые и известные алгоритмы запускались по 30 раз. Классические алгоритмы для решения единственной задачи запускались со всеми возможными вариантами числа кластеров от 2 до p_{max}.

В Таблице 2.3 приведены результаты работы предложенного алгоритма с гетерогенной популяцией в сравнении с лучшими результатами работы алгоритмов с полным объединенным и частичным объединенным решением и

результатами алгоритма с динамической популяцией. Для сравнения в таблицу включены результаты полученные мультистартом процедуры *k*-средних (ALA-процедуры) и генетическим алгоритмом с рекомбинацией подмножеств фиксированной длины, результаты испытаний ИС 1526ТЛ1 (N=1234, d=120, метрика такси[1] $L(X,Y)=\max\{||X-Y||,1\}$, описанная в Главе 3, p=10, p_{max}=10, t=2 мин), а также набор данных Ionosphere из репозитория UCI (N=351, d=34, метрика l_2^2, p=3, p_{max}=10, t=40 сек).

Таблица 2.3. Сравнение результатов работы алгоритмов.

Задача	П-тр	АПОР	АЧОР	АДП	АГП	ALA	АРПФД
2Д522Б, p=10	$F(x)$	11777,97	11777,93	11778,04	11776,48	11777,89	11775,56
	σ	0,117	0,080	0,131	1,879	0,023	2,559
2Д522Б, p=15	$F(x)$	9452,69	9449,48	9453,94	9456,17	9456,83	9454,98
	σ	3,395	3,870	1,958	2,621	4,817	4,817
140УД25, p=10	$F(x)$	974,81	974,76	974,77	974,51	974,76	977,11
	σ	0,239	0,283	0,283	0,370	0,283	5,461
140УД25, p=20	$F(x)$	660,17	659,91	659,84	661,23	667,68	663,32
	σ	1,016	0,416	0,243	1,081	1,188	0,763
MissAmer p=75	$F(x)$	755506,3	755845,47	756019,80	755422,09	765261,8	761278,34
	σ	995,725	591,256	566,549	1502,533	502,817	1755,331
MissAmer p=50	$F(x)$	827604,5	827866,55	827904,01	828391,37	833266,8	831822,57
	σ	925,538	621,880	366,072	717,021	1434,037	792,091
1526ТЛ1, p=10	$F(x)$	3440,06	3440,15	3442,64	3442,62	3454,09	3448,64
	σ	1,849	1,569	3,881	2,492	17,296	5,360
Ion, p=3	$F(x)$	8453,95	8452,32	8454,93	8450,95	8451,55	8451,54
	σ	4,533	3,292	4,747	0	2,444	2,447

Примечание: АПОР – лучшие результаты алгоритма с полным объединенным решением, АЧОР – лучшие результаты алгоритма с частичным объединенным решением, АДП – алгоритм с динамической популяцией, АГП – алгоритм с гетерогенной популяцией, ALA – ALA-алгоритм, АРПФД – генетический алгоритм с рекомбинацией подмножеств фиксированной длины. (F(x) значение целевой функции, σ – стандартное отклонение)

В более развернутой форме сравнение результатов работы алгоритмов приведено в Приложении А.

Как видно из таблиц, новый алгоритм с динамическим размером популяции (Алгоритм 2.5) в большинстве случаев не уступает лучшему из вариантов решения алгоритмов со статическим размером популяции.

Алгоритм 2.9 с гетерогенной популяцией, в отличие от серийного

[1] В практических задачах часто определена некая минимальная стоимость, подобно тому, как при поездке на такси есть начальная цена, включающая в себя стоимость поездки на 1-5 км.

Алгоритма 2.4, с уменьшением числа кластеров p дает решения, не уступающие решениям алгоритма для решения единственной задачи. Это позволяет получать результаты одновременно для серии задач, не уступающие по точности и стабильности результата известным алгоритмам решения единственной задачи, при этом позволяя одновременно решать сразу серию задач с различным числом кластеров. За счет этого повышается эффективность работы систем автоматической группировки без снижения требований к получаемому значению целевой функции.

Программная реализация алгоритмов включена в состав системы автоматизированного формирования и контроля спецпартий электрорадиоизделий космического применения и прошла опытную эксплуатацию на ОАО «Испытательный технический центр – НПО ПМ» (см. Приложение Б).

2.5 Стабильность получаемых решений

Как показано в [124], Алгоритм 2.4 вполне конкурентоспособен по точности и имеет преимущество по стабильности получаемого значения целевой функции. Однако, стабильность получаемого значения целевой функции не доказывает стабильного разбиения на одни и те же кластеры. Для исследования алгоритмов на предмет стабильности получаемого разбиения был проведен следующий эксперимент [212].

Задачи автоматической группировки ЭРИ по производственным партиям были решены различными алгоритмами с различным числом кластеров p от 2 до 9. Для каждой задачи каждый алгоритм запускался 10 раз. Полученные результаты фиксировались, а затем попарно сравнивались. Среди 10 результатов определялось максимальное количество полностью совпадающих результатов, а также среднее взвешенное расстояние между ними, исчисленное с использованием меры (метрики) Жаккара [213].

Мера Жаккара является мерой различия двух множеств S_1 и S_2:

$$J(S_1,S_2)=\frac{|S_1\cup S_2|-|S_1\cap S_2|}{|S_1\cup S_2|}.$$

Каждое из имеющихся решений задачи автоматической группировки является множеством p кластеров. Кластеры $C_{k,j}$ являются множествами векторов данных, непересекающимися подмножествами j-го решения D_j.

$$D_j=\bigcup_{i=1}^{p}C_{i,j},|D_j|=N,C_{k,j}\subset\{A_i\}\forall k=\overline{1,p}\ ,$$

$$C_{k,j}\cap C_{l,j}=\varnothing\bullet\forall k,l=\overline{1,p}.$$

В качестве меры различия двух решений можно было бы выбрать суммарное взвешенное расстояние в метрике Жаккара между соответствующими кластерами двух решений, использовав в качестве весового коэффициента количества элементов кластера:

$$dist(D_1,D_2)=\frac{\sum_{k=1}^{p}\left(J(C_{k,1},C_{k,2})|C_{k,1}|\right)}{N}.$$

Но, поскольку в разных решениях одни и те же кластеры могут быть пронумерованы по-разному, вначале нужно найти их попарные соответствия. Логично считать кластер $C_{l,2}$ второго решения соответствующим кластеру $C_{k,1}$ первого решения, если во втором решении нет кластера, который бы меньше отличался по составу от $C_{k,1}$, чем $C_{l,2}$. Поскольку отличия составов кластеров измеряются мерой Жаккара, имеем:

$$dist'(D_1,D_2)=\frac{\sum_{k=1}^{p}min_{j=\overline{1,p}}\left(J(C_{k,1},C_{j,2})|C_{k,1}|\right)}{N}.$$

Тогда в качестве меры стабильности результата алгоритма по составу получаемых кластеров будем использовать значение (для K запусков):

$$stability=\frac{\sum_{i=1}^{K}\sum_{j=i+1}^{K}dist'(D_1,D_2)}{K^2/2}.$$

Данную меру мы использовали для оценки стабильности наших результатов. Данные для трех различных задач приведены в таблицах 2.4-2.6.

В качестве меры стабильности в таблице используется средневзвешенное попарное расстояние Жаккара между соответствующими кластерами, N_p – число

кластеров, отклонение – среднеквадратичное отклонение целевой функции; 1 метод – мультистарт процедуры k-средних (ALA-процедуры), 2 метод – ГА с жадной эвристикой с вещественным алфавитом для решения серии задач в комбинации с ALA-процедурой, 3 метод – ГА с рекомбинацией подмножеств фиксированной длины в комбинации с ALA-процедурой в качестве средства локального поиска. Выбор используемых метрик в подобных задачах рассмотрен в [214].

Таблица 2.4. Сравнительная оценка стабильности результатов, получаемых различными рандомизированными алгоритмами для отбраковочных испытаний ИС 140УД25АС1ВК, (сборная партия из 2 партий), метрика l_1, N=56, d=42, t=2 с.

N_p	Метод	Усредн. значение целевой ф-ции	Отклонение	Мера стабильности	Макс. кол-во совпад. решений
2	1 метод	1369.897	0	0	10
	2 метод	1369.897	0	0	10
	3 метод	1369.897	0	0	10
3	1 метод	1237.608	0	0	10
	2 метод	1237.608	0	0	10
	3 метод	1237.608	0	0	10
4	1 метод	1143.206	0	0	10
	2 метод	1143.206	0	0	10
	3 метод	1143.206	0	0	10
5	1 метод	1072.397	0	0	10
	2 метод	1072.397	0	0	10
	3 метод	1072.397	0	0	10
6	1 метод	1019.872	1.2894	0.0935	6
	2 метод	1018.866	0	0	10
	3 метод	1018.866	0	0	10
7	1 метод	970.511	2.4889	0.14642	3
	2 метод	966.469	0	0	10
	3 метод	968.503	0.9518	0.05272	6
8	1 метод	923.380	1.9143	0.08961	2
	2 метод	921.063	0.0704	0.01039	8
	3 метод	920.847	0.5595	0.0509	3
9	1 метод	884.727	1.0276	0.2457	1
	2 метод	879.840	0.2503	0.0428	4
	3 метод	880.404	0.7341	0.0876	2

Таблица 2.5. Сравнительная оценка стабильности результатов, получаемых различными рандомизированными алгоритмами для отбраковочных испытаний ИС 1526ИЕ10 (сборная партия из 5 партий), метрика l_1, N=3987, d=206, t=7c.

N_p	Метод	Усредн. значение целевой ф-ции	Отклонение	Мера стабильности	Макс. кол-во совпад. решений
2	1 метод	351267.7	0	0	10
	2 метод	351267.7	0	0	10
	3 метод	351267.7	0	0	10
3	1 метод	311094.7	0	0	10
	2 метод	311094.7	0	0	10
	3 метод	311094.7	0	0	10
4	1 метод	296432.8	6.00473	0.0125	5
	2 метод	296419.6	0	0	10
	3 метод	296420.4	5.38030	0.0106	3
5	1 метод	284075.6	0.63262	0.0143	3
	2 метод	282493.6	0.34591	0.0012	9
	3 метод	282494.6	2.87659	0.0020	5
6	1 метод	273014.2	176.276	0.0998	2
	2 метод	272948.3	0	0	10
	3 метод	272949.1	1.23927	0.0146	4
7	1 метод	264571.7	842.773	0.1338	2
	2 метод	264050.8	6.5887	0,0091	6
	3 метод	265001.1	1142.34	0.2141	1
8	1 метод	258009.4	382.007	0.2387	1
	2 метод	257902.7	317.865	0.1005	5
	3 метод	258173.9	884.061	0.2714	1
9	1 метод	252365.3	606.091	0.2765	1
	2 метод	252069.3	306.065	0.1832	4
	3 метод	253246.8	1573.42	0.3735	1

Таблица 2.6. Сравнительная оценка стабильности результатов, получаемых различными рандомизированными алгоритмами для отбраковочных испытаний ИС 1526ТЛ1 (сборная партия из 3 партий), метрика l_2^2, d=120, N=1234, t=5c.

N_p	Метод	Усредн. значение целевой ф-ции	Откло нение	Мера стабиль ности	Макс. кол-во совпад. решений
2	1 метод 2 метод 3 метод	130834.6 130834.6 130834.6	0 0 0	0 0 0	10 10 10
3	1 метод 2 метод 3 метод	86599.78 86599.78 86599.78	0 0 0	0 0 0	10 10 10
4	1 метод 2 метод 3 метод	73923.03 73923.03 73923.03	0 0 0	0 0 0	10 10 10
5	1 метод 2 метод 3 метод	63337.30 63337.30 63337.30	0 0 0	0 0 0	10 10 10
6	1 метод 2 метод 3 метод	57571.80 57416.15 57416.15	83.4220 0.0295 0.0295	0.0718 0.0007 0.0007	4 6 6
7	1 метод 2 метод 3 метод	52785.07 52647.33 52647.26	72.5019 0.3044 0.36416	0.0659 0.0012 0.0021	3 7 6
8	1 метод 2 метод 3 метод	48617.58 48617.65 48617.59	0.0312 0.0312 0.0311	0.0002 0.0002 0.0002	9 9 9
9	1 метод 2 метод 3 метод	46051.49 45932.67 46347.68	237.1046 0.1815 271.7344	0.0539 0.0011 0.0689	1 9 3

Из таблиц видно, что в большинстве случаев наиболее стабильный результат достигается именно с помощью генетического алгоритма метода жадных эвристик. При многократных запусках алгоритм дает результаты, практически не различающиеся между собой. Кроме того, в большинстве случаев алгоритм дает как минимум 6 одинаковых результатов после 10 запусков. При увеличении времени счета показатель стабильности только увеличивается.

2.6 Нормировка данных испытаний электрорадиоизделий для задачи автоматической группировки

Для задачи автоматической группировки электрорадиоизделий используются данные, полученные в результате их испытаний на испытательных стендах. Это большой многомерный массив данных и для корректной работы алгоритма необходима нормировка данных, полученных в результате испытаний – приведение их к единой шкале.

Наиболее распространенные способы нормировки – 0-1-нормировка по минимальному и максимальному значению, когда минимальное значение показателя в выборке принимается равным 0, максимальное – равным 1, соответственно, остальные показатели располагаются в интервале (0;1) [215]. Другой способ – по среднеквадратичному отклонению:

$$a_{i,k} = \frac{a^*_{i,k} - \overline{a^*_k}}{\sigma(a^*_k)}.$$

Здесь $a^*_{i,k}$ – значение k-го показателя i-го вектора ненормированных данных, $\overline{a^*_k}$ – среднее значение k-го показателя по выборке, $\sigma(a^*_k)$ – его среднеквадратичное отклонение.

Указанные выше способы нормировки не сильно отличаются друг от друга, и оба они имеют существенный изъян, который проявляет себя при обработке реальных данных. Изделия разных партий одного и того же типа могут иметь различный разброс некоторого параметра. Так, для одной партии этот параметр может изменяться в широком диапазоне, и значение этого параметра на самом деле влияет на группировку изделий. А для изделий другой партии тот же параметр может быть стабилен, и фактически он не должен тогда влиять на группировку, но в соответствие с приведенными выше способами нормировки малейшее его изменение может повлиять на результат группировки. Для того чтобы этого избежать, требуется выявить, насколько изменения каждого параметра рассматриваемой партии являются значимыми.

Нами предложен новый способ нормировки [216], позволяющий получать гораздо более точное разбиение на производственные партии по данным тестовых испытаний электрорадиоизделий. Способ основан на использовании установленных границ дрейфа параметров изделий.

Дрейф – изменение параметров ЭРИ в ходе эксплуатации, а также изменение этих параметров в ходе проведения ДНК, имитирующего жесткие условия эксплуатации. Основным элементом ДНК [204] является электротермотренировка – имитация эксплуатации ЭРИ под нагрузкой в условиях высоких температур в течение нескольких часов или суток [217]. После электротермотренировки проводится повторный замер параметров и сравнение с их исходными значениями. Разность значений после ДНК и исходных значений мы и будем называть дрейфом в данной работе.

Дрейф параметров на заводах-изготовителях ЭРИ не оценивается. При проведении электротермотренировки фиксируются случаи выхода параметров за нормы ТУ. Фактически это означает, что отбраковываются ЭРИ с дрейфом параметров, превышающим границы X_{TY}. В качестве параметра, характеризующего запас параметрической надежности по величине дрейфа параметров, предлагается использовать коэффициент, определяемый соотношением:

$$k_{Д} = 1/\left[(X_{TY} - M(x_{TY}) / (X_{D} - M(y) \right],$$

где $M(X_{TY})$ – оценка математического ожидания параметра в соответствии с ТУ;

$M(y)$ – оценка математического ожидания дрейфа параметра в конкретной производственной партии;

Y_D – граница дрейфа параметра, установленная по выборке;

X_{TY} – верхняя или нижняя граница параметра по ТУ.

Математическое ожидание параметра по ТУ, фактически, является номинальным значением параметра ЭРИ по ТУ и может быть приближенно оценено по формуле:

$$M(X_{TY}) = \frac{1}{2}(X_{TY1} - X_{TY2}),$$

где X_{TY1} и X_{TY2}– соответственно верхняя и нижняя граница параметра по ТУ.

В качестве параметра, характеризующего нормы на границы дрейфа, установленные по выборке, принимается величина:

$$Y_D = M(y) \pm k \cdot s_D,$$

где s_D – оценка среднеквадратического отклонения дрейфа параметра;

k – толерантный коэффициент.

$M(y)$ определяется как $M(y) = \frac{1}{m}\sum_{i=1}^{M} Y_i$.

Соответственно, $s_D = \sqrt{1/m\sum_{i=1}^{m}[Y_i - M(y)]^2}$.

Оценка допустимых границ дрейфа производится по данным испытаний достаточно крупной партии ЭРИ.

Установленные таким образом нормы параметров в дальнейшем используются при проведении отбраковочных испытаний. Но также они являются важными ориентирами при оценке значимости разброса параметров конкретной партии изделий и могут быть использованы для более точной нормировки данных.

Итак, в основе нового способа нормировки исходных данных лежит использование допустимых границ дрейфа характеристик (показателей). В ходе электротермотренировки – одного из способов испытаний электрорадиоизделий – измеряемые показатели практически неизбежно меняются под воздействием высоких температур и значительных нагрузок. Для каждого из показателей установлены предельные значения этого изменения, называемого дрейфом. Обозначим нижнюю и верхнюю границу дрейфа *k*-го показателя соответственно δ_k^{min} и δ_k^{max}. Тогда предварительная обработка данных будет осуществляться согласно следующему выражению:

$$a_{i,k} = \frac{a_{i,k}^* - \overline{a_k^*}}{\delta_k^{max} - \delta_k^{min}}.$$

Как показали вычислительные эксперименты, такой способ нормировки по

границам дрейфа дает разбиение по производственным партиям с гораздо меньшим количеством ошибок. Это позволяет производить более точную классификацию электрорадиоизделий, что повышает эффективность алгоритмов автоматической группировки. При этом применение серийного алгоритма (Алгоритм 2.4) позволяет выполнять решение серии задач за один прогон алгоритма, благодаря чему такой алгоритм можно применять в интерактивном режиме, встраивая его в технологический процесс в качестве нового вида испытаний.

2.7 Применение генетического алгоритма с гетерогенной популяцией для задач разделения смеси распределений

В [218] описаны генетические алгоритмы метода жадных эвристик (в т.ч. генетический алгоритм для решения серии задач) содержащие в своем составе модифицированный EM-алгоритм.

EM-алгоритм для разделения смеси сферических распределений может быть описан следующим образом. Дан набор данных $S \subset R^n$, EM-алгоритм для смеси k нормальных распределений с общей сферической матрицей ковариации стартует с начальными стартовыми значениями параметров $\mu_i\langle 0\rangle$, $\alpha_i\langle 0\rangle$, $\sigma_i\langle 0\rangle$, которые в дальнейшем обновляются в соответствии со следующей двухшаговой процедурой (здесь t – номер итерации).

Алгоритм 2.11 EM-алгоритм.

Шаг 1. Пусть $\tau_i \sim N(\mu_i^{\langle t\rangle}, \sigma_i^{\langle t\rangle 2} I_n)$ является плотностью i-го гауссова распределения: $\tau_i(x) = \frac{1}{(2\pi)^{n/2}\sigma_i^{\,n}} exp(-\frac{\|x-\mu_i\|^2}{2\sigma_i^{\,2}})$. Для каждого вектора исходных данных $x \in S$ и каждого $1 \le i \le k$ по формуле Байеса [219] вычислим условную вероятность того, что x относится к i-му распределению с учетом текущих параметров: $p_i^{\langle t+1\rangle}(x) = \frac{\alpha_i^{\langle t\rangle}\tau_i(x)}{\sum_j \alpha_j^{\langle t\rangle}\tau_j(x)}$.

Шаг 2. Производится адаптация параметров распределений. Пусть N – количество векторов данных. $\alpha_i^{\langle t+1\rangle} = \frac{1}{N}\sum_{x\in S} p_i^{\langle t+1\rangle}(x)$; $\mu_i^{\langle t+1\rangle} = \frac{\sum_{x\in S} x p_i^{\langle t+1\rangle}(x)}{N\alpha_i^{\langle t+1\rangle}}$;

$$\sigma_i^{\langle t+1\rangle 2} = \frac{1}{d}\sum_{x\in S}\left\|x-\mu_i^{\langle t+1\rangle}\right\|^2 p_i^{\langle t+1\rangle}(x) \qquad (2.17)$$

Повторять с шага 1.

Условием останова в таком алгоритме является останов приращения целевой функции, в качестве которой принимается логарифмическая функция правдоподобия

$$L^{<t+1>} = \sum_{x\in S}\sum_{i=1}^{k} ln(\tau_i p_i^{\langle t+1\rangle}(x)). \qquad (2.18)$$

Использовано комбинированное условие останова: $t>t_{max}$ или $L\langle t+1\rangle - L\langle t\rangle<0.0001$.

В случае некоррелированных гауссовых распределений алгоритм в целом имеет тот же вид, за исключением того, что он оперирует вектором среднеквадратичных отклонений для каждого кластера (ковариационная матрица некоррелированного гауссова распределения – диагональная, и квадраты среднеквадратичных отклонений как раз и образуют эту диагональ). Использовался подход с разделением смесей некоррелированных гауссовых распределений с одинаковыми для всех кластеров векторами среднеквадратичных отклонений. В этом случае σ_j – среднеквадратичное отклонение (одинаковое для всех кластеров) по *j*-му измерению, и его перерасчет вместо (2.17) осуществляется следующим образом: $\sigma_j^{\langle t+1\rangle 2} = \sum_{i=1}^{k}\sum_{x\in S}\left\|x-\mu_i^{\langle t+1\rangle}\right\|^2 p_i^{\langle t+1\rangle}(x)/(N\alpha_i)$.

Отметим, что в случае данных большой размерности (*d*>100) плотность отдельных распределений $\tau_i(x)$ может принимать как очень малые значения ($\tau_i(x)$<1*10-300), так и очень большие значения, что может потребовать специальных механизмов при реализации алгоритма.

В работах [220, 221, 222, 223, 224, 225] предложен подход к повышению точности и стабильности результата решения задач k-медиан, k-медоид, k-

средних, основанный на применении жадных агломеративных эвристических процедур в комбинации с различными метаэвристиками и методами локального поиска. Идея жадной агломеративной эвристики основана на последовательном исключении кластеров из решения. Каждый раз удаляются те кластеры, удаление которых дает наименьший прирост целевой функции (данные задачи являются задачами минимизации). В [218] использован аналогичный подход с EM-алгоритмом.

Алгоритм 2.12 Базовая агломеративная эвристика для разделения гауссовых распределений. Дано: начальное число гауссовых распределений (кластеров) K, требуемое число кластеров k, $K>k$.

1.Выбрать начальное решение с K кластерами, т.е. выбрать случайным образом начальные параметры пары множеств распределений и их весовых коэффициентов $\langle D,W\rangle = \left\langle \left\{N(\mu_i^{\langle 0\rangle},\sigma_i^{\langle 0\rangle 2} I_n)\right\}, \left\{\alpha_i^{\langle 0\rangle}\right\}, i=\overline{1,K}\right\rangle$.

2.Выполнить Алгоритм 2.11, получить новое (улучшенное) решение задачи, представленное $\langle D,W\rangle$.

3.Если $K=k$, то останов.

4.Для каждого $i' \in \left\{\overline{1,K}\right\}$ выполнять:

4.1.Получить пару усеченных множеств $\langle D'',W''\rangle = \left\langle D \setminus \left\{N(\mu_{i'}^{\langle 0\rangle},\sigma_{i'}^{\langle 0\rangle 2} I_n)\right\}, \quad W \setminus \{\alpha_{i'}^{\langle 0\rangle}\}\right\rangle$.

4.2.Запустить Алгоритм 2.11 с начальными значениями параметров распределений, представленных усеченным $\langle D',W'\rangle$. При этом Алгоритм 2.11 ограничивается одной итерацией. Для полученного EM-алгоритмом решения рассчитать целевую функцию L согласно (2.18), сохранить ее значение в переменной $L_{i'}'$. Следующая итерация цикла 4.

5. Найти индекс $i'' = arg\ \underset{i'=1,k}{max} L_{i'}$. Получить пару усеченных $< D'',W'' >=< D \setminus \left\{N(\mu_{i''}^{\langle 0\rangle},\sigma_{i''}^{\langle 0\rangle 2} I_n)\right\}, \quad W \setminus \{\alpha_{i''}^{\langle 0\rangle}\} >$. Запустить для этой пары усеченных множеств Алгоритм 2.11, затем перейти к шагу 3.

Алгоритм 2.12 был использован в составе Алгоритма 2.13:

Алгоритм 2.13 Жадная процедура с частичным объединением №1.

Дано: пары множеств распределений $\langle D',W' \rangle = \left\langle \left\{ N(\mu'^{\langle 0 \rangle}_i ,\sigma'^{\langle 0 \rangle 2}_i I_n) \right\}, \left\{ \alpha'^{\langle 0 \rangle}_i \right\}, i = \overline{1,K} \right\rangle$ и

$\langle D'',W'' \rangle = \left\langle \left\{ N(\mu''^{\langle 0 \rangle}_i ,\sigma''^{\langle 0 \rangle 2}_i I_n) \right\}, \left\{ \alpha''^{\langle 0 \rangle}_i \right\}, i = \overline{1,K} \right\rangle$.

1. Для каждого $i' \in \{\overline{1,k}\}$ выполнять:

1.1. Объединить поэлементно множества в парах $\langle D',W' \rangle$ и $\langle D'',W'' \rangle$:

$\langle D,W \rangle = \left\langle D' \cup \left\{ N(\mu''^{\langle 0 \rangle}_{i'} ,\sigma''^{\langle 0 \rangle 2}_{i'} I_n) \right\}, W' \cup \{ \alpha''^{\langle 0 \rangle}_{i'} \} \right\rangle$.

1.2. Запустить Алгоритм 2.12 с этими парами объединенных множеств $\langle D,W \rangle$ в качестве начального решения. Полученный результат (пару полученных множеств, а также значение целевой функции) сохранить.

3. Возвратить в качестве результата лучшее (по значению целевой функции) из решений, полученных на шаге 1.2.

Данная процедура фактически дополняет один из «родительских» вариантов решения задачи разделения смесей, представленный парой $\langle D',W' \rangle$ поочередно с каждым из элементов второго «родительского» решения, представленного соответствующей парой множеств $\langle D'',W'' \rangle$, затем к этой паре объединенных множеств применяет жадную эвристику (Алгоритм 2.12) и фиксирует лучший из полученных результатов.

Более простой, но более требовательный в плане вычислительных ресурсов вариант аналогичного алгоритма представлен ниже.

Алгоритм 2.14 Жадная процедура с полным объединением.

Дано: см. Алгоритм 2.13

1.Объединить поэлементно множества $\langle D',W' \rangle$ и $\langle D'',W'' \rangle$:

$< D,W >=< D' \cup D'' , \quad W' \cup W'' >$.

2. Запустить Алгоритм 2.12 с этими объединенными множествами в качестве начального решения.

Как видно, здесь выполняется полное объединение двух «родительских» вариантов решения задачи разделения смесей распределений, представленных парами множеств $\langle D',W'\rangle$ и $\langle D'',W''\rangle$ соответственно. Наконец, возможен промежуточный вариант, в котором множества объединяются частично, при этом первое множество берется полностью, а из второго множества выбирается случайным образом случайное число элементов. Подобный подход хорошо зарекомендовал себя при решении задач автоматической группировки с применением моделей k-средних, k-медоид, k-медиан [226, 143].

Алгоритм 2.15 Жадная процедура с частичным объединением №2.

Дано: см. Алгоритм 2.13

1. Выбрать случайное $r'\in[0;1)$. Присвоить $r=[(k/2-2)\,r'2]+2$. Здесь [.] – целая часть числа.

2. Повторять *k-r* раз:

2.1. Сформировать случайно выбранное подмножество D''' элементов множества D'' мощности r и подмножество W''' соответствующих элементов множества W''' (также мощности r). Объединить множества $\langle D,W\rangle=\langle D'\bigcup D''',\ W'\bigcup W'''\rangle$. Запустить Алгоритм 2.12 с этими объединенными множествами в качестве начального решения.

3. Возвратить в качестве результата лучшее (по значению целевой функции) из решений, полученных на шаге 2.1.

Данные эвристические процедуры, являющиеся (не в строгом смысле) алгоритмами локального поиска в окрестности известного («родительского») решения, представленного множествами $\langle D',W'\rangle$, могут использоваться в составе различных стратегий глобального поиска. При этом в качестве окрестностей, в которых производится поиск решения, используются решения, производные («дочерние») по отношению к решению $\langle D',W'\rangle$, образованные комбинированием его элементов с элементами решения $\langle D'',W''\rangle$ и применением жадной агломеративной эвристики (Алгоритма 2.12).

Одной из хорошо зарекомендовавших себя стратегий глобального поиска является применение эволюционных (генетических) алгоритмов.

Сложности кодирования решений, традиционно представляемых в классических генетических алгоритмах L-битными строками [227, 228], в алгоритмах метода жадных эвристик [223, 1, 229] решены применением так называемого генетического алгоритма с вещественным алфавитом, в котором «особи» - промежуточные решения задач k-медиан или k-средних – представлены непосредственно множествами точек в пространстве $\Re^d$ (то есть непосредственно множествами медиан или центроидов). Аналогичный подход с кодированием промежуточных решений в виде множеств векторов вещественных чисел мы применили и при построении генетического алгоритма для решения задач разделения смесей распределений. В нашем алгоритме промежуточные решения представлены парами множеств $\langle D_m, W_m \rangle = \langle \{N(\mu_{m,i}^{\langle 0 \rangle}, \sigma_{m,i}^{\langle 0 \rangle 2} I_n)\}, \{\alpha_{m,i}^{\langle 0 \rangle} = 1/k\}, i = \overline{1,K}, m = \overline{1,N_{POP}} \rangle$ где N_{POP} – размер популяции алгоритма, т.е. количество «особей» - промежуточных решений, которым он оперирует.

Важнейшим параметром генетических алгоритмов, включая генетические алгоритмы с жадной эвристикой, является размер популяции N_{pop}. В изначальном варианте генетического алгоритма с жадной эвристикой для p-медианной задачи на сети [120] предлагается такой размер популяции, чтобы каждый узел сети являлся центром хотя бы одного кластера хотя бы в одном из решений-«особей». Такой подход приводит к формированию больших популяций, для которых формирование хотя бы второго и третьего поколений решений-«особей» требует очень продолжительного времени с учетом того, что жадная эвристика включает в себя запуск алгоритма локального поиска (в нашем случае – запуск EM-алгоритма). С другой стороны, в работах [223, 1, 229] предлагается использовать фиксированный небольшой размер популяции, порядка 15-25 «особей». В то же время, для очень больших задач время выполнения единственной процедуры «скрещивания» (кроссинговера) может быть велико и трудно предсказуемо, и в этой связи для очень больших задач также работа не доходит даже до третьего

поколения «особей». Для малых же задач такая популяция быстро вырождается – алгоритм раз за разом генерирует одни и те же решения из очень ограниченного множества возможных комбинаций.

Предложен следующий простой вариант определения динамически меняющегося размера популяции: размер зависит от номера итерации в соответствии со следующим соотношением:

$$N_{POP} = arg\ max\left(N_{POP(нач.)}, (\sqrt{1+N_{iter}} + 2)\right).$$

Алгоритм 2.16 Генетический алгоритм с жадной эвристикой (дано общее описание трех вариантов, названных GA-FULL, GA-ONE, GA-RAND). Дано: Начальный размер популяции N_{POP}.

1. Сгенерировать случайным образом N_{POP} начальных решений, представленных множествами распределений $D_m = \{N(\mu_{m,i}, \sigma_{m,i}{}^2 I_n), i = \overline{1,k}\}, m = \overline{1,N_{POP}}$ и соответствующими множествами весовых коэффициентов $W_m = \{\alpha_{l,i} = 1/k, i = \overline{1,k}\}, m = \overline{1,N_{POP}}$. Начальные значения среднеквадратичных отклонений устанавливаются равными для всех кластеров и вычисляются для всей выборки: $\sigma_i^2 = \frac{1}{d}\sum_{x \in S} \left\| x - \overline{x} \right\|^2$. Значения $\mu_{m,i}$ устанавливаются равными координатам случайно выбранных векторов данных.

Для каждого из начальных решений запускается Алгоритм 2.11, полученные значения целевой функции сохраняются в переменных $f_1, \ldots, f_{N_{POP}}$. Присвоить N_{iter}=0;

2. Проверка условий останова (например, максимальное время работы). Если условия достигнуты, возвращается решение, которому соответствует наилучшее (наибольшее) значение целевой функции $f_1, \ldots, f_{N_{POP}}$.

3. Присвоить N_{iter}=N_{iter}+1; $N_{POP} = max\{ N_{POP}; \lceil \sqrt{1+N_{iter}} \rceil + 2 \}$; если N_{POP} изменилось, то инициализировать новое решение (см. Шаг 1).

4. Выбрать случайным образом два индекса $k_1, k_2 \in \{\overline{1,N}\}, k_1 \neq k_2$. Для пары решений, представленных множествами D_{k_1}, D_{k_2} и W_{k_1}, W_{k_2}, выполнить

Алгоритм 2.15 (в варианте алгоритма GA-FULL – генетический алгоритм с жадной эвристикой с полным объединением), Алгоритм 2.14 (в варианте алгоритма GA-ONE – генетический алгоритм с жадной эвристикой с частичным объединением), либо выбрать один из двух этих алгоритмов случайным образом (вариант алгоритма GA-RAND).

5. Выбрать индекс $k_3 \in \{\overline{1,N_{POP}}\}$. Используем простое турнирное замещение: случайным образом выбираем $k_4, k_5 \in \{\overline{1,N_{POP}}\}$, если $f_{k_4} < f_{k_5}$ то $k_3=k_4$, иначе $k_3=k_5$.

6. Заменяем множества D_{k_3}, W_{k_3} и соответствующее значение целевой функции f_{k_3} новыми значениями, полученными на Шаге 4. Перейти к Шагу 2.

Общность EM-алгоритма и ALA-алгоритма позволяет применить общие подходы к повышению точности результатов. В работе [125] приведен эволюционный алгоритм с гетерогенной популяцией для p-медианной задачи. Здесь применен тот же подход для решения задачи разделения сферических и некоррелированных гауссовых распределений.

Если в задачах k-средних, k-медиан, k-медоид параметрами решения задачи являются исключительно множества координат центров кластеров $X_i=(x_{i,1},\ldots,x_{i,d})$, то в задаче о разделении смеси распределений решениями являются множества параметров распределений, дополненные значениями их весовых коэффициентов – априорных вероятностей распределений $\alpha_i^{\langle t\rangle}$, т.е. в этом случае каждое решение X – это множество троек $(\mu_i^{\langle t\rangle}, \sigma_i^{\langle t\rangle}, \alpha_i^{\langle t\rangle})$. Также учитывается, что задача разделения смеси распределений – это задача максимизации.

Алгоритм 2.17 Алгоритм с гетерогенной популяцией для задачи разделения смеси распределений.

1. Сгенерировать случайным образом $N_{POPнач}$ начальных решений, представленных парой множеств распределений $\langle D_m, W_m\rangle = \langle \{N(\mu_{m,i}^{\langle 0\rangle}, \sigma_{m,i}^{\langle 0\rangle 2})\}, \{\alpha_{m,i}^{\langle 0\rangle} = 1/k\}, i=\overline{1,k_m}, m=\overline{1,N_{POPнач}}\rangle$. Начальные значения среднеквадратичных отклонений устанавливаются равными для всех кластеров и

вычисляются для всей выборки: $\sigma_i^{\langle 0\rangle 2} = \frac{1}{d}\sum_{x\in S}\left\|x-\bar{x}\right\|^2$. Значения $\mu_{m,i}^{\langle 0\rangle}$ устанавливаются равными координатам случайно выбранных векторов данных. Для каждого из начальных решений запускается Алгоритм 2.11, полученные значения целевой функции сохраняются в переменных $f_1,...,f_{N_{POP}}$. Присвоить N_{iter}=0.

2. $N_{iter}=N_{iter}+1$; $N_{POP} = max\{ N_{POPнач}; \lceil\sqrt{1+N_{iter}}\rceil + 2\}$; Если N_{pop} изменилось, то инициализировать особь X_{Npop} аналогично шагу 1. Выбрать случайным образом $k_1,k_2 \in [1, N_{POP}]$, $k_1 \neq k_2$;

3. ; $\langle D_{new}, W_{new}\rangle = \langle D_{k_1} \cup D_{k_2}, W_{k_1} \cup W_{k_2}\rangle$

4. Пока $|D_{new}| > p_{max}$;

4.1. Выбрать элемент j, такой, чтобы его исключение давало наименьшее ухудшение целевой функции: $j = arg \max\limits_{i'\in 1,|D_{new}|} L_{i'}\left(D_{new} \setminus \{N(\mu_{i'},\sigma_{i'})\}, W_{new} \setminus \{\alpha_{i'}\}\right)$

4.2 $D_{new} = D_{new} \setminus \{N(\mu_j,\sigma_j)\}$, $W_{new} = W_{new} \setminus \{\alpha_j\}$ Следующая итерация 4.

5. Выбрать случайным образом $p_{child} \in \{2, p_{max}\}$. Если $p_{child} > |D_{new}|$, то $p_{child} = |D_{new}|$;

6. $f_{child,|D_{new}|} = L(\langle D_{new}, W_{new}\rangle)$;

7. Пока $|D_{new}| > p_{child}$

7.1. Выбрать элемент j, такой, чтобы его исключение давало наименьшее ухудшение целевой функции: $j = arg \max\limits_{j\in 1,|D_{new}|} L\left(D_{new} \setminus \{N(\mu_j,\sigma_j)\}, W_{new} \setminus \{\alpha_j\}\right)$

7.2. $D_{new} = D_{new} \setminus \{N(\mu_j,\sigma_j)\}, W_{new} = W_{new} \setminus \{\alpha_j\}$ Следующая итерация 7.

8. Пока $|D_{new}| > 2$:

8.1. Присвоить $f_{child,|D_{new}|} = L(D_{new}, W_{new}); k = |D_{new}|; f_{k,|D_{new}|} = L(D_{new}, W_{new})$; если $f_{k,|D_{new}|} < F_k^*$, то присвоить $F_k^* = f_{k,|D_{new}|}$;

8.2. Выполнить шаги 4.1 и 4.2 для D_{new}. Следующая итерация 8.

9. Выбрать $j_3 \in \{1, N_{POP}\}$ с использованием турнирного замещения.

Присвоить $D_{j_3} = D_{new}; W_{j_3} = W_{new}; f_{j_3,k} = f_{child}$

10. Проверить условия останова, перейти к шагу 2.

Для тестирования алгоритмов мы использовали как результаты тестирования электрорадиоизделий, полученные ОАО «ИТЦ НПО-ПМ» (эти данные нормированы специальным образом – по границам дрейфа), так и классические наборы данных из репозитория UCI (www.cs.uci.edu/mlearn/mlrepository.html).

Результаты (Таблицы 2.7-2.10) показывают, что новые алгоритмы не имеют преимущества перед мультистартом классического EM-алгоритма при очень малом числе кластеров (2-3 кластера). С ростом числа кластеров и объема выборки сравнительная эффективность повышается, и для многих больших задач новые алгоритмы имеют безусловное преимущество. Также новые алгоритмы оказались неэффективны для разбиения наборов булевых данных (см.набор данных Chess). При этом невозможно отдать однозначное предпочтение одной из версий нового алгоритма.

Для большинства наборов данных было выполнено по 30 попыток запуска каждого из алгоритмов, для наборов данных KDDCUP04Bio и Europe – по 10 попыток. Фиксировались только лучшие результаты, достигнутые в каждой попытке, затем эти результаты были усреднены. Результаты работы EM-алгоритма в режиме мультистарта и его модификаций обозначены EM, CEM, SEM. Результаты работы модификаций ГА обозначены как GA1 (GA-ONE), GA2 (GA-FULL), GA3 (GA-RAND)

Таблица 2.7. Сравнительные результаты для генетического алгоритма с жадной эвристикой и модификацией EM-алгоритмов

Набор данных, число вектор., размерн.	Chess (UCI), N=3197, d=50, булевы,		ИС 140УД17АВК, N=51, d=46,		Ionosphere, N=351, d=35		Mopsi (UCI), N=6014, d=2,	
Тип распр., время,число кластеров *k*	сфер., 3 мин., 10 кластеров		сфер., 0,5 мин., 2 кластера		нормир.,сфер., 1 мин., 10 кластеров		сфер.,15 мин., 20 кластеров	
	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.
Рез-ты EM	-30525∗	0	3790,1∗	104	-871,4	15,8	39268,7	9967,6
Рез-ты CEM	-30564	32,3			-893,4	7,9	48424,2	237,3
Рез-ты SEM	-30560	46,4			-880	15,2	36272,2↓	9619,6
Рез-ты GA1	-30581↓	1,8	3665,6↓	0	-960,3↓	23,5	50443,4	115,3
Рез-ты GA2	-30532	19,6	3697,8	83,4	-824,8∗	4,5	49288,9	390,5
Рез-ты GA3	-30554	28,7	3707,7	88	-832,8	14,8	50499,9∗	60,9

Таблица 2.7. (продолжение)

Набор данных, число векторов, размерность	Europe, (UCI), N=169308, d=2		BIRCH-3 (UCI), N=100 000, d=2		KDDCUP04Bio(UCI), N=145751, d=74	
Тип распр., время,число кластеров *k*	некорр.,60 мин., 30 кластеров		сфер., 60 мин., 100 кластеров		нормир.,сфер., 90 мин.,30 кластеров	
	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.
Рез-ты EM	-3653148,6	2634,7	-2567483	4351,1	-12513229	1256
Рез-ты CEM	-	-	-2603151	5170,3	-12512717	343
Рез-ты SEM	-3654493↓	4348,6	-2728547↓	3869,7	-12514336↓	683,6
Рез-ты GA1	-3643277∗	2122,9	-2553038	8014	-12512517∗	159,1
Рез-ты GA2	-3651917	3465,7	-2348371∗	588278	-12512817	410,9
Рез-ты GA3	-3648896,2	5151,5	-	-	-12512811	639,1

Примечания: ∗-лучший результат; ↓ - худший результат.

Таблица 2.8. Сравнительные результаты для генетического алгоритма с жадной эвристикой и модификацией EM-алгоритмов для разных значений кластеров

Набор данных, число векторов, размерность	ИС 1526ЛЕ2, N=3987, d=206					
Тип распр., время,число кластеров *k*	некорр.,0,5 мин., 2 кластера		некорр.,0,5 мин., 5 кластеров		некорр.,0,5 мин., 10 кластеров	
	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.
Рез-ты EM	16534776*	1466117	15537106*	2662488	8979298	1412448
Рез-ты CEM	1785868↓	157424	1886941↓	36348	8899298	79384
Рез-ты SEM	1551508	547055	910053	686384	8777561↓	615624
Рез-ты GA1	15719410	3364381	15038109	3448334	11069633*	3246532
Рез-ты GA2	16374013	2341266	10910351	4557883	10692948	457343
Рез-ты GA3	16327088	1486014	12550856	5196064	9162226	5146053

Примечания: ∗-лучший результат; ↓ - худший результат.

Таблица 2.9. Сравнительные результаты для генетического алгоритма с жадной эвристикой и модификацией EM-алгоритмов для разных значений кластеров

Набор данных, число векторов, размерность	Ionosphere, N=351, d=35					
Тип распр., время,число кластеров *k*	некорр.,2 мин., 3 кластера		некорр.,2 мин., 7 кластеров		некорр.,2 мин., 10 кластеров	
	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.	ср.знач. целев.ф.	сркв. откл.
Рез-ты EM	115933*	23345	101108*	19622	80564	8552
Рез-ты CEM	6647	3251	5895	487	4140	2963
Рез-ты SEM	2306↓	2304	2467↓	3193	-264↓	79
Рез-ты GA1	84520	16238	99297	21598	90455*	27938
Рез-ты GA2	113184	18074	83461	26362	85573	24927
Рез-ты GA3	103015	18216	83723	26856	89224	27609

Примечания: ∗-лучший результат; ↓ - худший результат.

Таблица 2.10. Сравнительные результаты серийного алгоритма с гетерогенной популяцией (Алгоритм 2.17) решения задач разделения смеси распределений

Набор данных, число вектор., размерность, тип распр-я, время прогона.	Число класт. *k*	Алгоритм	Ср. рез-т (лог. ф-ция правдоподоб.)	Среднеквадр. откл. результатов
Europe (UCI), *N*=169308, *d*=2 сферич, 1.5 часа.	40	Новый	-3625694,1*	20,148
		VNS2	-3625691,7*	14,769
		VNS3	-3625748,7	15,402
		EM	-3625957,3	49,561
		CEM	-3625779,0	25,064
		SEM	-3625740,2	29,064
Тесты ИС 1526ТЛ1, *N*=1234, *d*=120 сферич., 5 сек.	5	Новый	3673,671*	44,043
		VNS2	3543,974	144,691
		VNS3	3591,539	139,988
		EM	3598,160	32,160
Ionosphere, N=351, d=35, некоррелир. гауссово 15 сек	2	Новый	75843,08437*	24676,43567
		CEM	-5828,573294	0
		SEM	-5393,426806	1253,776554
		GA-FULL	55448,7001	22721,87905
		GA-ONE	75632,07793	25539,29628
		GA-RAND	57364,73187	17857,83098
	3	Новый	113933,1276*	23345,78651
		CEM	6647,436212	3251,042019
		SEM	2306,30573	2304,756938
		GA-FULL	84520,38226	16238,23724
		GA-ONE	113184,3001	18074,55099
		GA-RAND	103015,3048	18216,28923
	5	Новый	93858,7735	24413,94829
		CEM	5652,379689	1204,741017
		SEM	3387,095016	3551,680318
		GA-FULL	94775,08338	30510,29304
		GA-ONE	97281,21672*	25385,65362
		GA-RAND	89543,48932	20627,21983
	7	Новый	99301,7553*	19621,61142
		CEM	5894,534598	486,8482242
		SEM	2467,007219	3193,351329
		GA-FULL	99297,18151	21597,60173
		GA-ONE	83460,86312	26361,85196
		GA-RAND	83722,74318	26856,09099
	10	Новый	87563,64881	8552,051049
		CEM	4140,14943	2963,387714
		SEM	-264,2750616	79,39635456
		GA-FULL	90455,05474*	27938,49599
		GA-ONE	85572,87652	24927,24082
		GA-RAND	89224,39439	27608,79678

Таблица 2.10. (продолжение)

Набор данных, число вектор., размерность, тип распр-я, время прогона.	Число класт. *k*	Алгоритм	Ср. рез-т (лог. ф-ция правдоподоб.)	Среднеквадр. откл. результатов
ИС 1526ЛЕ2, N=3987, d=206 некоррелир. гауссово 1 мин	2	Новый	16534776,89*	1466117,877
		CEM	1785868,286	157424,1577
		SEM	1551508,52	547055,7822
		GA-FULL	15719410,09	3364381,708
		GA-ONE	16374013,71	2341266,143
		GA-RAND	16327088,79	1486014,162
	5	Новый	14537106,21	2662488,511
		CEM	1886941,353	36347,99832
		SEM	910053,0728	686384,0116
		GA-FULL	15038109,11*	3448334,02
		GA-ONE	10910351,48	4557883,571
		GA-RAND	12550856,62	5196064,777
	10	Новый	11094337,87*	5001166,509
		CEM	1899198,003	59578,20495
		SEM	777561,1466	538958,1976
		GA-FULL	11069633,09	5233339,264
		GA-ONE	10692948,18	6081784,966
		GA-RAND	7162226,308	5187965,408
	15	Новый	12748308,73*	4823811,16
		CEM	1790617,35	174865,2597
		SEM	1050428,115	581007,7463
		GA-FULL	9595665,575	4341400,435
		GA-ONE	6506382,825	5696807,758
		GA-RAND	7057633,7	4575160,897

Примечания: ∗-лучший результат;

Таким образом, с одной стороны, метод жадных эвристик [223, 1] может быть успешно применен для построения эффективных алгоритмов решения задач разделения смеси распределений. При этом сохраняется важное свойство алгоритмов, полученных с применением данного подхода: высокая точность получаемых результатов. Но полученные алгоритмы не обладают свойством высокой стабильности получаемых решений, характерной для алгоритмов метода жадных эвристик, что связано с особенностями решаемых задач: известные алгоритмы локального поиска, такие как SEM-алгоритм, позволяют получать гораздо более стабильный, но при этом стабильно плохой результат, стабильно уступающий новым алгоритмам. В то же время, для некоторых практических

задач, в качестве примера которых можно привести задачи автоматической группировки электрорадиоизделий [214, 230, 231], сформулированные в виде задач разделения смеси гауссовых распределений [232] результатов тестовых испытаний [21, 233], новые в ходе нескольких (не более 10) попыток запуска позволяют найти, вероятно, точный результат задачи или, по крайней мере, результат, который не получается превзойти с применением известных алгоритмов, благодаря чему данный недостаток новых алгоритмов нивелируется.

Результаты главы 2

Таким образом, разработаны новые алгоритмы метода жадных эвристик со стратегией глобального поиска, реализованной особым генетическим алгоритмом с гетерогенной (смешанной) популяцией решений, содержащей решения задач с различным числом групп (кластеров) и особой модификацией жадных эвристических процедур. При этом в качестве алгоритмов локального поиска, предусмотренных методом жадных эвристик, могут использоваться различные эффективные алгоритмы локального поиска для задач с соответствующей мерой расстояния. Экспериментально доказано, что разработанные алгоритмы позволяют в случае большой размерности практических задач получать результаты одновременно для серии задач, не уступающие по точности и стабильности результата известным алгоритмам решения единственной задачи, при этом позволяя одновременно решать сразу серию задач с различным числом кластеров. За счет этого повышается эффективность работы систем автоматической группировки без снижения требований к получаемому значению целевой функции.

Также разработан генетический алгоритм одновременного решения серии задач нечеткой кластеризации данных большой размерности на основе модели разделения смеси вероятностных распределений с различным предполагаемым числом распределений, при заранее известном максимальном числе распределений. Экспериментально доказано, что разработанный алгоритм

позволяет в случае большой размерности задач получать результаты одновременно для серии задач, практически не уступающие по достигаемому значению целевой функции и стабильности результата известным алгоритмам решения единственной задачи разделения смеси распределений, при этом позволяя одновременно решать сразу серию задач с различным числом распределений в смеси.

ГЛАВА 3. РЕШЕНИЕ ЗАДАЧИ С МЕРОЙ РАССТОЯНИЯ, ОГРАНИЧЕННОЙ СНИЗУ

3.1. Постановка задачи

Диагностические испытания электрорадиоизделий на испытательных стендах ИТЦ служат источником данных для алгоритмов автоматической группировки электрорадиоизделий. Однако, любые измерительные приборы имеют погрешность измерения и это влияет на результаты испытаний.

В задаче группировки определяются сходства и различия объектов и учет погрешностей представляет собой достаточно сложную задачу, так как элементарные арифметические операции с величинами погрешностей и отклонений не могут дать необходимый результат в случае многомерного пространства признаков (так, к примеру, известно, что при увеличении размерности диагональ единичного гиперкуба растет как корень значения размерности) [234].

Для задачи автоматической группировки в евклидовом пространстве в первую очередь имеет значение погрешность при определении координат объектов (значений характеристик) и расстояний между ними. Если два вектора $X_0 = (x_1, x_2, ..., x_p)$ и $Y_0 = (y_1, y_2, ..., y_p)$ размерности p известны с погрешностями $\varepsilon = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_p)$ и $\delta = (\delta_1, \delta_2, ..., \delta_p)$, то фактически мы имеем дело с векторами $X = X_0 + \varepsilon, \quad Y = Y_0 + \delta.$ Тогда

$$\rho^2(X,Y) = \rho^2(X_0,Y_0) + 2\sum_{i\in(1,p)}(x_i - y_i)(\varepsilon_i - \delta_i) + \sum_{i\in(1,p)}(\varepsilon_i - \delta_i)^2. \quad (3.1)$$

Пусть все характеристики объектов измерены с некими предельными значениями разбросов $|\varepsilon_i| \le \Delta, |\delta_i| \le \Delta, \quad i = 1,2,...,n.$ Предварительная стандартизация (нормировка) характеристик делает такое предположение вполне

естественным. При $p\Delta^2 \to 0$ последнее слагаемое в (3.1) не превышает $4p\Delta^2$, соответственно им можно пренебречь, и норма евклидова расстояния принимает вид:

$$N_{p^2}(X_0, Y_0) = 4 \sum_{i \in (1,p)} |x_i - y_i| \Delta$$

При условии, что математические ожидания величин $|x_i - y_i|$ одинаковы (после стандартизации переменных), то существует константа *C*, такая что

$$N_{p^2}(X_0, Y_0) = Cp\Delta$$

с точностью до бесконечно малых более высокого порядка при малых Δ, больших *p* и $p\Delta^2 \to 0$.

Из вышеприведенного следует, что

$$\rho(X, Y) = \rho(X_0, Y_0) + \theta \frac{Cp\Delta}{2\rho(X_0, Y_0)}, \quad \theta \in (-1, 1) \qquad (3.2)$$

Уравняв слагаемые в (3.2) и зная предельную погрешность отнормированных данных, мы можем определить минимально различимое расстояние при решении задачи автоматической группировки:

$$\rho_{\min} = \frac{Cp\Delta}{2\rho_{\min}}, \quad \rho_{\min} = \sqrt{\frac{Cp\Delta}{2}} \qquad (3.3)$$

Таким образом, расстояние между точками, меньшее чем $\rho_{\min}$, можно считать нулевым, а точки, которые разделены этим расстоянием неразличимыми. При нормальном распределении x_i и y_i и дисперсии равной 1, $\mu\{|x_i - y_i|\} = 2\sqrt{\pi}$, и, соответственно, $C \approx 4{,}51$. Следовательно в нашем случае $\rho_{\min} = 1{,}5\sqrt{p\Delta}$.

Из (3.3) можно заключить, что хотя возрастание размерности приводит к увеличению длины диагонали единичного куба, в пределах которого расположены

переменные, вместе с ней увеличивается и естественный квант расстояния – порог неразличимости $\rho_{\min}$. Следовательно, добавляя новые характеристики объектов, т.е. увеличивая размерность пространства нельзя добиться улучшения применения методов автоматической группировки [234].

Если на производстве установлен регламент проведения испытаний изделий, предписывающий не принимать во внимание разницу в показаниях измерительных приборов меньшую определенной величины, то в задаче автоматической группировки электрорадиоизделий имеет смысл использовать меру расстояния, пренебрегающую неким минимальным расстоянием, то есть ограниченную снизу. Однако задача Вебера (решение которой предполагается на втором шаге ALA-процедуры) для такой меры проработана недостаточно и было решено исследовать этот вопрос.

Задача Вебера это одна из четырех классических задач теории размещения [36, 67]. Эта простая задача оптимизации поставлена Альфредом Вебером в 1909 году [45, 67] как задача оптимального размещения объектов $X^* \in \Re^n$ на плоскости

$$X^* = arg \min_{X \in \Re^n} f(X) = arg \min_{X \in \Re^n} \sum_{i=1}^{N} w_i \| A_i - X \|. \qquad (3.4)$$

Здесь $A_i \in \Re^n, i \in \overline{\{1, N\}}$ некоторые заданные точки, называемые точками-потребителями, $w_i \in \Re, w_i \geq 0$ это их весовые коэффициенты, $\| - \|$ это некоторая норма $\Re^n \to \Re$. Первоначальная задача Вебера это задача неограниченной оптимизации. Вебер сформулировал её для евклидовой нормы $(\| - \| = l_2(-))$, позднее стали использоваться и другие нормы [235, 90, 88, 236, 237].

Примеры использования задачи Вебера включают в себя такие задачи как задача размещения склада [238, 36], задачи размещения базовых станций беспроводных сетей и других объектов. В число значимых областей применения входят также кластерный анализ, аппроксимация и решение статистических задач.

Эффективное решение существует лишь для специальных случаев [239].

Например, вариант задачи с l_1-нормой вместо евклидовой может быть решен за линейное время. Это также справедливо, если сумма квадратов евклидовых расстояний сведена к минимуму.

В 1937 Вайсфельд предложил алгоритм для решения задачи Вебера [48]. Позднее, Варди и Жанг модифицировали алгоритм Вайсфельда [85], а Сегеди частично расширил его для более общей задачи [240]. Однако, алгоритм Вайсфельда в некоторых случаях имеет очень медленную сходимость [239]. Это также справедливо и для более сложных непрерывных задач размещения.

В реальных условиях в пространстве размещения обычно есть преграды и запрещенные зоны. Задача Вебера с преградами и запрещенными зонами представляет собой гораздо более сложную задачу оптимизации. Многие авторы предлагали различные алгоритмы для специальных случаев [62, 70, 63, 241]. В [242, 243] авторы предлагают методы для задачи Вебера с ограничениями и минимаксной задачи соответственно. Также в [242, 244] предложены алгоритмы решения задачи Вебера с многогранными допустимыми зонами.

Некоторые варианты задачи Вебера представляют собой задачу невыпуклой оптимизации, которую сложно решить точно [245], потому что невыпуклая задача оптимизации, как правило включает в себя множество допустимых зон размещения и множество локальных оптимальных точек для каждой зоны [246]. Поэтому найти глобальное решение невыпуклой задачи оптимизации очень трудно. Самыми удобными в этом смысле мерами расстояния являются квадрат евклидова расстояния (не является метрикой) и прямоугольная метрика – в этом случае задача Вебера решается неитеративно алгоритмом линейной сложности.

Если условием решения задачи размещения или автоматической группировки является обязательное пренебрежение небольшими расстояниями, не превышающими некоторого предела (будем считать, что предел равен 1, чего можно достичь нормировкой), имеем меру расстояния $L(X,Y) = \max\{\|X - Y\|,1\} \forall X,Y$. В работе П.Станимировича и Л.А.Казаковцева показано, что задача с такой мерой расстояния может быть разложена на серию задач размещения с евклидовой метрикой, где область допустимых решений

ограничена окружностями. Каждая задача имеет область допустимых решений эквивалентную области пересечения окружностей с центрами в точках-потребителях.

Применение процедуры Вайсфельда, типичной для задачи Вебера без ограничений, может привести к образованию новой точки за пределами зоны допустимых решений, заданной ограничениями. Алгоритм, предложенный Л.А.Казаковцевым и П.Станимировичем, основан на замене точки на ближайшую точку в области допустимых решений.

Это единственный известный алгоритм для решения задач оптимизации для такой меры расстояния и, в поиске альтернативы, мы обратились к одному из перспективных направлений оптимизации – алгоритмам роевого интеллекта, в частности, алгоритму пчелиной колонии.

В тех случаях, когда оптимальное решение найти трудно, для ускорения поиска могут быть использованы эвристические и метаэвристические методы. Эвристические методы предложенные различными авторами позволяют получить обнадеживающие результаты решения задачи Вебера в отношении качества решений и затрат вычислительных ресурсов [247, 248, 249, 250].

В течение нескольких последних десятилетий в научном сообществе наблюдается стремление решать сложные задачи оптимизации с помощью метаэвристических алгоритмов [251]. В отличие от эвристических методов эти методы не зависят от специфичности задач. Кроме того, было показано, что эти алгоритмы могут обеспечить гораздо более эффективные решения по сравнению с классическими алгоритмами [252, 253, 254, 255, 256].

Одними из наиболее интересных и широко используемых видов метаэвристик являются алгоритмы роевого интеллекта, основанные на коллективном интеллекте муравьиных колоний, термитников, пчелиных роев, птичьих стай [257, 258] и т.д. Причина их успешности заключается в обмене информацией между агентами, таким образом, самоорганизация, совместная эволюция и обучение во время рабочих циклов приводят к качественным результатам. Хотя не все алгоритмы роевого интеллекта являются успешными,

некоторые доказали свою эффективность и, таким образом, приобрели важное значение в качестве инструмента для решения реальных задач [259]. Наиболее эффективные и хорошо изученные алгоритмы это муравьиный алгоритм [260], метод роя частиц [261], алгоритм пчелиной колонии (АПК) [262], светлячковый алгоритм (СА) [263] и алгоритм кукушки [264]. После изобретения этих алгоритмов они были доработаны для общих и специальных задач [265, 266, 267, 268, 269, 270, 271, 272, 273, 274].

Целью данной работы является изучение производительности некоторых известных методов роевого интеллекта для решения задачи Вебера с ограничениями – с допустимыми зонами, ограниченными дугами. Этот вариант задачи Вебера имеет невыпуклое допустимое множество, заданное ограничениями, что делает поиск глобального оптимума с помощью детерминированных алгоритмов оптимизации гораздо сложнее. Следовательно, метаэвристические алгоритмы могут быть использованы для получения качественных результатов.

В нашем исследовании [275] для решения ограниченной задачи Вебера используются четыре алгоритма роевого интеллекта: алгоритм пчелиной колонии для ограниченной оптимизации [276], кроссоверный алгоритм пчелиной колонии для ограниченной оптимизации (КО-АПК) [277], светлячковый алгоритм для ограниченной оптимизации [278] и улучшенный светлячковый алгоритм (УСА) [279]. КО-АПК и УСА это недавно предложенные улучшенные варианты АПК и СА для решения задач оптимизации с ограничениями соответственно. Кроме того, в [249] предложен эвристический алгоритм для решения задачи Вебера с ограничениями. Для этих пяти алгоритмов проведены эксперименты по решению задачи Вебера с допустимыми зонами, ограниченными дугами, и содержащими до 500 ограничений. Эксперименты проведены на случайно сгенерированных наборах данных.

Ограниченная задача Вебера с допустимыми зонами, ограниченными дугами [249] на непрерывном пространстве сформулирована в (3.4). Допустимые зоны определены как множество ограничений:

$$\| X - A_i \| - 1 \le 0, \forall i \in S_<, \quad (3.5)$$

и

$$1 - \| X - A_i \| \le 0, \forall i \in S_>, \quad (3.6)$$

где $S_<, S_> \in \{\overline{1, N}\}, S_< \bigcap S = \varnothing_>$ это подмножества множества индексов известных точек (точек-потребителей), а N общее количество точек-потребителей.

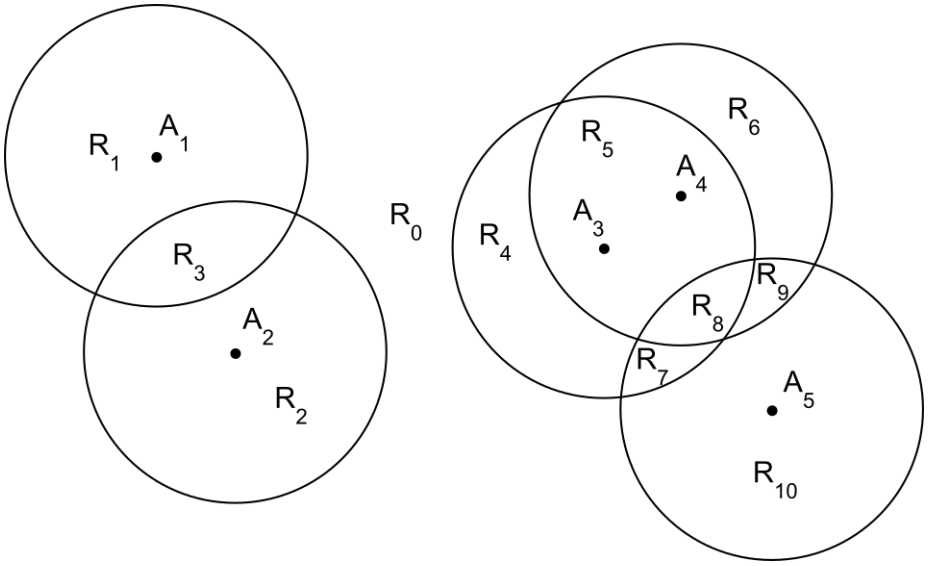


Рис. 3.1 Иллюстрация задачи (3.4), с ограничениями заданными (3.5)-(3.6). Каждая зона R_i имеет свое множество ограничений.

Метрика, используемая в практических задачах, зависит от многих факторов, в том числе от свойств транспортных средств [249]. В случае с системой общественного транспорта, цена обычно зависит от расстояния. Однако, обычно определена некая минимальная цена. Например, начальная стоимость поездки в такси может включать в себя стоимость поездки на 1-5 км. После перемасштабирования расстояний, так что расстояние, включенное в начальную цену было бы равно 1, можно определить функцию цены d_P как

$$d_P(X, Y) = max\{\|X - Y\|, 1\} \forall X, Y \in \Re^2. \quad (3.7)$$

где $\| - \|$ это некоторая норма.

В случае, если функция расстояния определена (3.7), задача может быть разложена на серию ограниченных задач размещения с евклидовой метрикой, где область допустимых решений ограничена дугами [280]. Каждая задача имеет

область допустимых решений эквивалентную области пересечения кругов (рис. 3.1) с центрами в точках-потребителях.

3.2. Модифицированная процедура Вайсфельда

Процедура Вайсфельда для решения задачи Вебера с заданным отклонением ε может быть описана [238] как:

Алгоритм 3.1. Процедура Вайсфельда

Требуется: Координаты и веса точек-потребителей $A_i = (a_1^i, a_2^i), w_i, i = \overline{1, N}$, предопределенное отклонение ε.

Шаг 1: Вычислить начальную точку

$$X^* = (x_1^*, x_2^*) : x_r = \frac{\sum_{i=1}^{N} a_r^i w_i}{\sum_{i=1}^{N} w_i} \; \forall\, r \in \{1,2\};\; n_{iter} = 0.$$

Шаг 2: Пока $\Delta > \varepsilon$ цикл:

Шаг 2.1: $n_{iter} = n_{iter} + 1;\; d_{denom} = \sum_{i=1}^{N} w_i / \| A_i - X^* \|_2$.

Шаг 2.2: $x_r^{**} = \sum_{i=1}^{N} \frac{x_r^* w_i}{\| X^* - A_i \|_2 \cdot d_{denom}} \; \forall i \in \{1,2\}\;;\; X^{**} = (x_1^{**}, x_2^{**})$.

Шаг 2.3: $\Delta = \left\| X^* - X^{**} \right\|; X^* = X^{**}$.

Шаг 2.4: Продолжить Шаг 2.

Шаг 3: ОСТАНОВ, возврат X^{**}.

В [21] предложен алгоритм для решения задачи Вебера, заданной (3.4) с ограничениями (3.5) и (3.6), основанный на процедуре Вайсфельда. Множество допустимых решений для данной задачи невыпуклое, а целевая функция *f*(*X*), заданная в (3.4) выпуклая [244]. Решение задачи оптимизации с ограничениями и выпуклой целевой функцией совпадает с решением задачи без ограничений или лежит на границе запрещенной зоны [243]. Таким образом, если X^* является решением задачи, заданной (3.4) с ограничениями (3.5) и (3.6), тогда X^* решение и неограниченной задачи (3.4) или $\exists i \in \{\overline{1, N}\} : \| A_i - X^* \|_2 = 1$.

Шаг 2.2 Алгоритма 3.1 может привести к образованию новой точки X^{**} за

пределами зоны допустимых решений, заданной ограничениями (3.5) и (3.6). Опишем эту область как R_f, предполагая, что $R_f \neq \varnothing$.

Опишем для произвольной точки $X \in \Re^2$ ближайшую точку в R_f:

$$C(X) = \arg\min_{X' \in R_f} \| X - X' \|_2 =$$

$$\begin{cases} X, & X \in R_f, \\ \arg\min_{X' \in R_f} \| X - X' \|_2, & X \notin R_f. \end{cases}$$

Алгоритм, предложенный в [21] основан на замене точки X^{**}, сгенерированной в шаге 2.2 Алгоритма 3.1 на ближайшую точку в области допустимых решений:

Алгоритм 3.2 Модифицированная процедура Вайсфельда [249]

Требуется: Координаты и веса точек-потребителей $A_i = (a_1^i, a_2^i), w_i, i = \overline{1, N}$, предопределенное отклонение ε, ограничения (3.5) and (3.6), определенные множествами $S_<$ и $S_>$.

Шаг 1: Вычислить начальную точку $X^* \in R_f$ (здесь R_f это допустимое множество, ограниченное ограничениями); $X^* = C(X^*)$ $\Delta = +\infty$.

Шаг 2: Пока $\Delta > \varepsilon$ цикл:

Шаг 2.1: $n_{iter} = n_{iter} + 1$; $d_{denom} = \sum_{i=1}^{N} w_i / \| A_i - X^* \|_2$.

Шаг 2.2: $x_r^{**} = \sum_{i=1}^{N} \frac{x_r^* w_i}{\| X^* - A_i \|_2 \cdot d_{denom}} \; \forall r \in \{1,2\}$.

Шаг 2.3: Если $X^{**} \notin R_f$ тогда $X^{**} = C(X^{**})$.

Шаг 2.4: $\Delta = \| X^* - X^{***} \|$; $X^* = X^{***}$.

Шаг 2.5: Продолжить Шаг 2.

Шаг 3: ОСТАНОВ, возврат X^{**}.

3.3. Метаэвристические алгоритмы

3.3.1. Алгоритм пчелиной колонии (АПК) для задачи оптимизации с ограничениями

Алгоритм пчелиной колонии, предложенный Д.Карабога, имитирует пищевое поведение роя медоносных пчел. Половину популяции виртуальных пчел составляют пчелы-рабочие, вторая половина состоит из наблюдателей и разведчиков. Каждый источник пищи – это возможное решение. Для каждого источника пищи есть своя рабочая пчела, и каждая рабочая пчела ищет пищу в окрестности этого источника пищи. По окончании поиска рабочие передают информацию о качестве источника пищи наблюдателям. Те, основываясь на полученной информации, летят к наиболее качественным источникам и ищут пищу в их окрестности. Рабочие с неперспективных источников превращаются в разведчиков и летят на поиски новых источников.

АПК был разработан для численных задач оптимизации без ограничений. Позднее он был расширен для решения задач с ограничениями [281, 276]. Для работы с ограничениями АПК использует правила Деба. В правилах Деба используется турнирная селекция, т.е. сравниваются два решения за раз по определенным критериям [282]:

* Любое допустимое решение, удовлетворяющее всем ограничениям, предпочтительнее любого недопустимого решения, нарушающего какое-либо из ограничений

* Среди двух возможных решений, имеющее лучшее значение целевой функции является предпочтительным

* Если оба решения недопустимы, решение с меньшей суммой нарушений ограничений предпочтительно. Сумма нарушений ограничений определяется как:

$$CV(X) = \sum_{j=1}^{q} max(0, g_i(X)) + \sum_{j=q+1}^{m} |h_j(X)|$$

где $g_i(X)$ ограничения неравенств, $h_j(X)$ – ограничения равенств, q –

количество ограничений равенств, m – общее количество ограничений в данной задаче.

Для задачи Вебера с ограничениями и допустимыми зонами, ограниченными дугами, мы задействовали версию АПК с плавающей точкой. Наша реализация алгоритма пытается получить оптимальный двумерный вектор [x_1, x_2], сводящий (3.4) к минимуму и удовлетворяющий ограничениям (3.5) и (3.6). Сначала в пространстве поиска случайным образом генерируется популяция особей, содержащая векторы решения, закодированные в числах с плавающей точкой, затем пригодность каждого решения оценивается. После инициализации, популяция пошагово улучшается через фазы рабочих, наблюдателей и разведчиков. Уравнение поиска решения в фазах рабочих и наблюдателей описывается как:

$$v_{ij} = \begin{cases} x_{ij} + \varphi_j \cdot (x_{ij} - x_{kj}), & если\ R_j < MR \\ x_{ij}, & если\ R_j \geq MR \end{cases} \quad (3.8)$$

где φ_j – случайное число в диапазоне [-1, 1], x_k – другое решение, случайно выбранное из популяции, R_j – случайно выбранное вещественное число из [0, 1), j = 1,2,...D (D – размерность задачи, в случае задачи Вебера D=2). MR (modification rate) – уровень модификации, управляющий параметр алгоритма, определяющий параметры оптимизации в уравнении поиска.

Процесс заканчивается, когда механизм обработки ограничений применяется к созданному новому решению v_i и осуществлен выбор по правилам Деба между x_i и v_i. Механизм обработки ограничений задан как:

$$v_{ij} = \begin{cases} l_j, & если\ v_{ij} < l_j \\ u_j, & если\ v_{ij} > u_j \end{cases} \quad (3.9)$$

где l_j и u_j нижняя и верхняя границы j-ой переменной в решении-кандидате v_i.

В алгоритме пчелиной колонии для задачи оптимизации с ограничениями

есть пять управляющих параметров. Наряду с обычными для алгоритмов роевого интеллекта, такими как размер популяции (*SP*), максимальное количество циклов (MCN) и вышеупомянутый *MR*, есть еще два – *limit* и период разведки (*SPP*). Они используются в фазе разведчиков.

Для каждого решения в популяции алгоритм вычисляет величину $failure_i$, которая представляет собой количество неудачных попыток улучшить решение x_i. Если величина $failure_i$ окажется равна или превысит значение параметра limit, то решение x_i будет отброшено. Параметр *SPP* используется в фазе разведчиков, чтобы определить заданный период циклов воспроизводства пчел-разведчиков.

Алгоритм 3.3 Псевдокод для алгоритма пчелиной колонии для задачи оптимизации с ограничениями.

Инициализация управляющих параметров SP, MCN, MR, limit и SPP

Инициализация в пространстве поиска популяции решений x_{ij}, $i = 1\ldots, SP/2$,

$j = 1\ldots, D$ случайным образом

Вычислить каждое x_i, $i = 1\ldots, SP/2$

Инициализация параметра $failure_i = 0$, $i = 1\ldots, SP/2$

cycle = 1

repeat

/*Фаза пчел-рабочих*/

for i = 1 to SP/2 do

Создать новое решение v_i для пчел-рабочих x_i согласно формуле (3.8)

Применить ограничения на созданное решение v_i согласно формуле (3.9)

Произвести выбор согласно критериям Деба между решениями x_i и v_i

if (решение x_i не улучшено) then

$failure_i = failure_i + 1$

else

$failure_i = 0$

end if

end for

/*Фаза пчел-наблюдателей*/

for i = 1 to SP/2 do

Вычислить величины вероятностей Pi для каждого решения x_i:

P_i = 0.9 * ($fitness_i$/maxfit) + 0.1 /* maxfit это лучшее значение целевой функции в популяции*/

end for

t = 1

i = 1

while (t < SP/2) do

if (S_i < P_i) then

/* S_i случайно выбранное вещественное число в диапазоне [0,1)*/

t = t + 1

Создать новое решение v_i для наблюдателей x_i согласно формуле (3.8)

Применить ограничения на созданное решение v_i согласно формуле (3.9)

Произвести выбор согласно критериям Деба между решениями x_i и v_i

if (решение x_i не улучшено) then

$failure_i$ = $failure_i$ + 1 else

$failure_i$ = 0

end if

end if

i = i + 1

if (i = SP/2) then

i = 1

end if

end while

/*Фаза пчел-разведчиков*/

если (cycle mod SPP = 0) then

Каждое решение $failure_i$ которого больше или равен значению *limit*

заменяется случайно созданным решением

end if

Запомнить лучшее решение на данный момент

cycle = cycle + 1

until cycle = MCN

3.3.2 Кроссоверный алгоритм пчелиной колонии (КО-АПК) для задачи оптимизации с ограничениями

В последние годы было предложено несколько версий алгоритма пчелиной колонии для решения задачи оптимизации с ограничениями [283, 284, 285, 286, 277]. Одним из наиболее интересных вариантов является недавно разработанный кроссоверный алгоритм пчелиной колонии, которому удалось превзойти другие алгоритмы на базе АПК, а также 11 других современных алгоритмов по качеству результатов, надежности и скорости сходимости [277]. В нашем исследовании этот вариант АПК так же применяется к задаче Вебера с допустимыми зонами, ограниченными дугами, для сравнения с другими метаэвристическими алгоритмами.

Основное дополнение в КО-АПК связано с операторами поиска в каждой фазе алгоритма, используемыми для улучшения распространения информации.

В фазе пчел-рабочих КО-АПК использует измененное по сравнению с АПК уравнение поиска:

$$v_{ij} = \begin{cases} x_{ij} + \varphi \cdot (x_{ij} - x_{kj}), & если\ R_j < MR \\ x_{ij}, & если\ R_j \geq MR \end{cases} \quad (3.10)$$

где φ – случайное число в диапазоне [-1, 1], x_k – другое решение, случайно выбранное из популяции, R_j – случайно выбранное вещественное число из [0, 1), j = 1,2,...D (D – размерность задачи, в случае задачи Вебера D=2). Как видно из (3.10), КО-АПК использует одно и то же число φ для каждого параметра j. Вдобавок, КО-АПК вместо фиксированного значения MR использует линейно возрастающее в начальных итерациях от 0.1 до заданного MR_{max} и равное MR_{max}

в оставшихся итерациях.

Также в фазе пчел-рабочих КО-АПК используется новое уравнение поиска для лучшего исследования окрестности качественного решения. Это уравнение:

$$v_{ij} = x_{ij} + \varphi \cdot (x_{ij} - x_{kj})$$

где φ – постоянное случайное число в диапазоне [-1, 1], x_i и x_k – два случайно выбранных решения, R_j – случайно выбранное вещественное число из [0, 1), j = 1,2,...D

В фазе разведчиков в КО-АПК для генерации новых решений в зоне, представляющей интерес, применяется единообразный кроссоверный оператор. Таким образом после каждой SPP-ой итерации каждое решение x_i которое превышает limit заменяется новым, создаваемым по:

$$v_{ij} = \begin{cases} y_j, & если\ R_j < 0.5 \\ x_{ij}, & если\ R_j \geq 0.5 \end{cases}$$

где y_j это j-ый элемент лучшего глобального решения, найденного на текущий момент, R_j – случайно выбранное вещественное число из [0, 1), j = 1,2,...D.

Также важно упомянуть, что в КО-АПК используется другой механизм обработки ограничений, отличный от АПК. Этот механизм позволяет сохранять разнообразие популяции и описан в [287]:

$$v_{ij} = \begin{cases} 2 \cdot l_j - v_{ij}, & если\ v_{ij} < l_j \\ 2 \cdot u_j - v_{ij}, & если\ v_{ij} > u_j \\ v_{ij}, & иначе \end{cases}$$

где v_{ij} – переменная j решения-кандидата i, l_j и u_j это нижняя и верхняя границы параметра v_{ij}.

3.3.3 Светлячковый алгоритм для оптимизации с ограничениями

Светлячковый алгоритм (СА) это алгоритм роевого интеллекта, основанный на брачном поведении светлячков. Он быстро приобрел известность, после того, как с его помощью было решено много задач оптимизации в различных областях [272, 273]. Поведение светлячков характеризуется испускаемым ими светом. Основная функция этого света привлекать брачных партнеров [257]. При увеличении расстояния до источника света интенсивность света падает, также влияние оказывает поглощение света воздухом [273]. Светлячковый алгоритм моделирует поведение светлячков таким образом, что интенсивность света пропорциональна целевой функции оптимизируемой задачи. Для построения светлячкового алгоритма используются три правила [273]:

1. Все светлячки одного пола
2. Их привлекательность пропорциональна их яркости
3. Яркость светлячка связана с целевой функцией

Светлячковый алгоритм первоначально был предложен для решения численных задач оптимизации без ограничений [263], а затем был расширен для решения задач с ограничениями путём включения в него штрафных функций [278]. Использование штрафных функций позволяет преобразовать задачу с ограничениями в задачу без ограничений [288]. Формула алгоритма следующая:

$$\varphi(X) = f(X) + p(X)$$

где $\varphi(X)$ – расширенная целевая функция для оптимизации, а $p(X)$ – штрафное значение, вычисляемое по:

$$p(X)=\sum_{j=1}^{q} r_j \cdot max(0, g_j(X))^2 + \sum_{j=q+1}^{m} c_j \cdot \left| h_j(X) \right|,$$

где r_j и c_j это положительные константы.

В светлячковом алгоритме для задачи Вебера с ограничениями и допустимыми областями, ограниченными дугами, группа из SP светлячков ищет хорошие решения на каждой итерации. Поисковый оператор представляет собой движение светлячка *i* к другому более яркому и привлекательному светлячку *j* и задается:

$$x_{ik} = x_{ik} + \beta \cdot (x_{ik} - x_{jk}) + \alpha \cdot S_k \cdot (rand_k - \frac{1}{2}) \qquad (3.11)$$

где второе слагаемое отвечает за привлекательность, а третье вносит элемент случайности.

Во втором слагаемом формулы (3.11) параметр β это привлекательность светлячков [257] вычисляемая как монотонно убывающая функция:

$$\beta = \beta_0 - e^{-\gamma \cdot r_{ij}^2} \quad (3.12)$$

где r_{ij} это расстояние между светлячком x_i и светлячком x_j, а β_0 и γ это заданные параметры алгоритма : максимальная привлекательность и коэффициент поглощения, соответственно.

Расстояние r_{ij} между светлячками x_i и x_j определяется как:

$$r_{ij} = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2}$$

Управляющий параметр β_0 определяет привлекательность в случае если два светлячка находятся на расстоянии r=0. Управляющий параметр γ определяет изменение привлекательности с ростом расстояния.

В третьем слагаемом формулы (3.11), $\alpha \in [0,1]$ параметр для рандомизации,

S_k это параметр для масштабирования, а $rand_k$ случайное число от 0 до 1. Параметр масштабирования S_k вычисляется как:

$$S_k = |u_k - l_k| \quad (3.13)$$

где l_k и u_k верхняя и нижняя границы k-ой переменной решения x_i.

В дополнение к этому в [278] упоминается возможность повысить качество решений путем уменьшения рандомизационного параметра α в геометрической прогрессии как в алгоритме имитации отжига, по схеме заданной:

$$\alpha(t) = \alpha(t-1) \cdot \theta^{\frac{1}{MCN}} \quad (3.14)$$

где MCN это максимальное количество циклов, t номер текущей итерации, а θ это параметр, вычисляемый по:

$$\theta = \frac{10.0^{-4.0}}{0.9} \quad (3.15)$$

Псевдокод светлячкового алгоритма для задачи оптимизации с ограничениями приведен в Алгоритме 3.4

Алгоритм 3.4. Псевдокод светлячкового алгоритма для задачи оптимизации с ограничениями

Случайным образом инициализировать популяцию решений в пространстве поиска

Вычислить каждый x_i , i = 1,2, . . . , SP

Инициализовать управляющие параметры SP, MCN, α_0, γ и β_0

t=0

while $t < MCN$ do

for i = 1 to SP do

for j = 1 to i do

if (x_i имеет большее значение целевой функции,

чем x_j, т е. x_j ярче чем x_i) then

for k = 1 to D do

$$g_k = x_{ik} + \beta \cdot (x_{ik} - x_{jk}) + \alpha \cdot S_k \cdot (rand_k - \frac{1}{2})$$

{где β вычислено по (3.12) и S_k вычислено по (3.13)}

end for

end if

end for

end for

Вычислить новое значение α по (3.14)

Проранжировать светлячков и запомнить лучшее решение на текущий момент

$t = t + 1$

end while

3.3.4 Улучшенный светлячковый алгоритм для оптимизации с ограничениями

Не так давно для решения задач оптимизации с ограничениями был предложен улучшенный светлячковый алгоритм (УСА) [279]. Он был протестирован на четырёх структурных задачах оптимизации и результаты показывают, что он весьма конкурентоспособен и превосходит оригинальный светлячковый алгоритм. Также он был применен к решению задачи Вебера с допустимыми зонами, ограниченными дугами, для сравнения с АПК, КО-АПК и СА.

Дл улучшения производительности в алгоритме было произведено две модификации. Первая касается использования для обработки ограничений правил Деба вместо штрафных функций. Механизм отбора на правилах Деба при создании нового решения применяется дважды. Первый раз при применении трех

правил допустимости вместо жадного отбора в процессе решения, какой светлячок ярче. Эти правила также применяются каждый раз после применения (3.11) для определения, будет ли дополняться решение. Использование этих правил снижает разнородность популяции [276], и, следовательно, процесс поиска повышает эксплуатационную способность CA [289].

Вторая модификация заключается в применение геометрической прогрессии к уменьшению параметра масштабирования S_k. В [279] был сделан вывод, что дальнейшее повышение качества решений и скорости сходимости CA может быт достигнуто уменьшением каждого параметра S_k, k=1,2,...,D, с помощью той же схемы, что использована для уменьшения параметра α. Схема описывается как:

$$S_k(t) = S_k(t-1) \cdot \theta^{\frac{1}{MCN}}$$

где MCN это максимальное количество циклов, t номер текущей итерации, а θ это параметр, вычисляемый по (3.15).

3.4 Эксперименты

Для экспериментов использован компьютер PC Intel Core i5-3300@3GHz с 4GB RAM. Так же для сравнения с метаэвристическими алгоритмами, в список алгоритмов включена модифицированная процедура Вайсфельда.

3.4.1 Эталонные функции

Производительность четырех метаэвристических и одного эвристического алгоритмов испытывается на 18 тестовых заданиях, каждое задание это однообъектная задача Вебера со связанными допустимыми зонами, ограниченными дугами равного радиуса. Задачи сгенерированы случайным образом, согласно алгоритму, приведенному в [249], реализованы в вариантах для 5, 10, 50, 100, 250 и 500 точек. Следовательно, эти тестовые задания включают в

себя множество ограничений с количеством членов от 5 до 500.

Шесть задач под названиями P1, P4, P7, P10, P13 и P16 для 5, 10, 50, 100, 250 и 500 точек соответственно изображены на Рисунке 3.2. На каждом рисунке допустимая зона закрашена серым, красным крестом обозначено решение-результат работы эвристического алгоритма [249].

Кроме того, расчеты в метрике такси проведены на реальных данных для задачи автоматической группировки электрорадиоизделий, результаты приведены в таблице 2.2.

3.4.2 Параметры

Для корректного сравнения четырех метаэвристических алгоритмов базовые параметры были ограничены до 20 решений и 200 итераций для всех алгоритмов. Кроме того, в метаэвристических алгоритмах, приведенных в разделе 3, предусмотрены управляющие параметры, влияющие на производительность. Величины этих параметров были взяты из [278], [279], [276] и [277], где эти алгоритмы были предложены для решения задач оптимизации с ограничениями.

В случае с АПК [276] величины управляющих параметров таковы: уровень модификации MR=0.8, период разведки $SPP = 0.5 * sn * D$, $limit = 0.5 * sn * D$, где $D = 2$ (размерность задачи), sn размер пчелиной колонии.

Для КО-АПК [277] управляющие параметры установлены как: первичное значение уровня модификации P=0.3, конечное занчение уровня модификации MR_{max}=0.9, период разведки SPP=350, $limit$=1

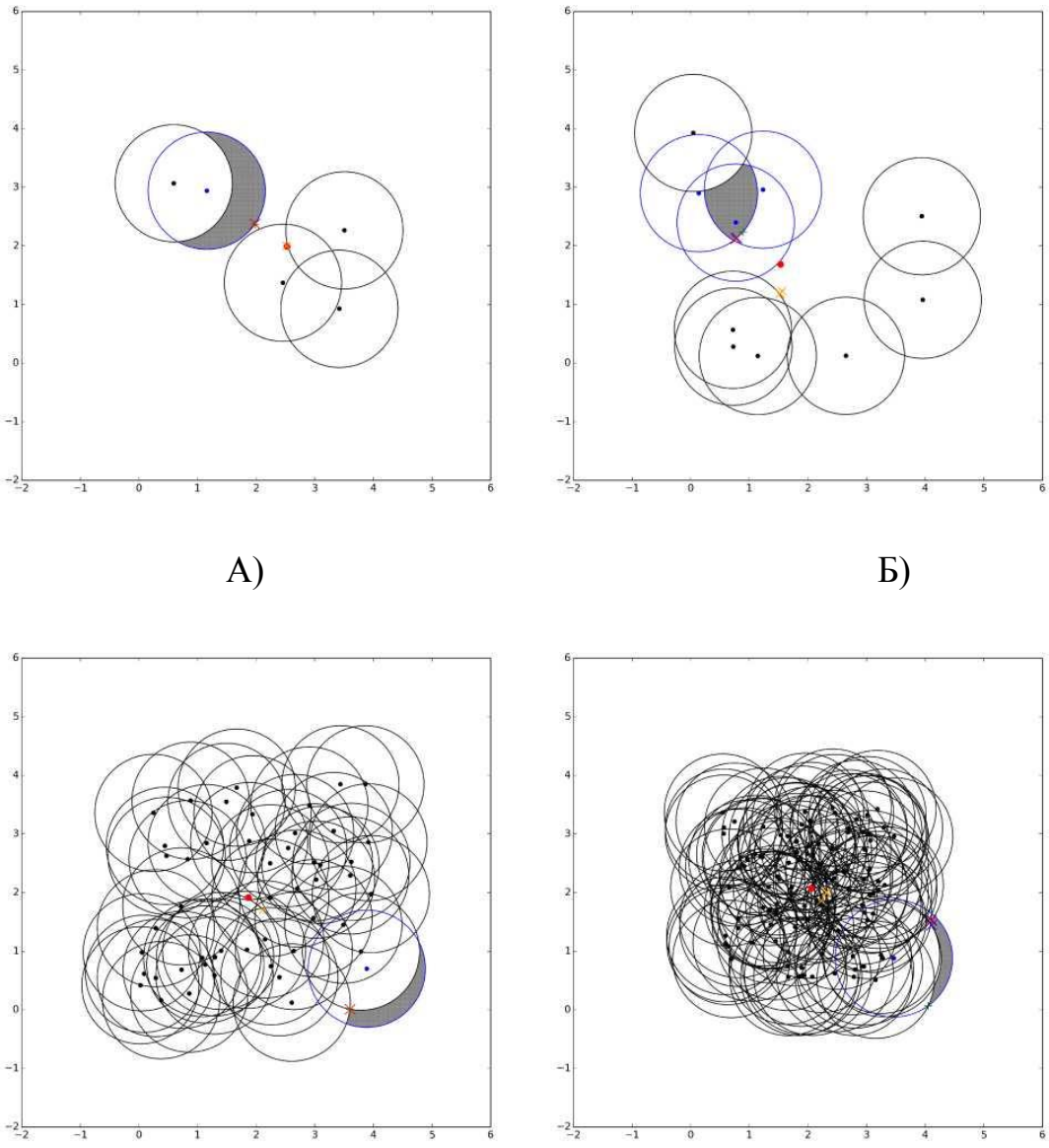
Для СА [278] установлены следующие значения управляющих параметров: коэффициент поглощения γ=1, начальное значение привлекательности β_0=1, начальное значение α=0.5.

Для УСА [279] величины управляющих параметров заданы как: коэффициент поглощения γ=1, начальное значение β=1.5, начальное значение α=0.9.

Для каждого эксперимента произведено 30 запусков.

3.4.3. Анализ полученных решений

Координаты решения, соответствующее значение целевой функции и процессорное время (в секундах) для эвристического алгоритма приведены в Таблице 3.1. Для анализа качества решений, полученных с помощью метаэвристических алгоритмов, используются наилучшие и средние значения, а также стандартные отклонения за 30 запусков. Лучшие результаты приведены в Таблице 3.2, средние значения и стандартные отклонения в Таблице 3.3.



А) Б)

В) Г)

Рис 3.2. Примеры задач: А) с 5 точками (P1), Б) с 10 точками (P4), В) с 50 точками (P7), Г) со 100 точками (P10)

В [249] сходимость эвристического алгоритма подтверждена экспериментально на случайным образом сгенерированных задачах. Следовательно, наилучшие результаты метаэвристических алгоритмов можно сравнивать с результатами эвристического для того, чтобы показать способность метаэвристик достичь ближайшего оптимального результата. Кроме того, средние значения и значения стандартных отклонений показывают надежность метаэвристических подходов.

Таблица 3.1. Точка решения, значение целевой функции и вычислительное время эвристического алгоритма

	К-во точек	Точка решения	Значение целевой функции	Время (сек)
P1	5	{1.97900484015, 2.37038768628}	38.6308975914	4.61E-05
P2	5	{2.511104376, 2.36822367838}	35.6524439425	3.34E-05
P3	5	{3.29440267812,1.91712629136}	32.8210021279	3.45E-05
P4	10	{0.768148485664,2.11871223533}	18.9069002076	0.001
P5	10	{1.35331974755,1.96480428579}	18.4514161651	0.001
P6	10	{2.40607643306,1.99679256415}	106.1627295736	0.001
P7	50	{3.60036240333,0.00927504530255}	723.0047301353	0.002
P8	50	{2.14432492473,0.0678576078541}	114.6249767471	0.007
P9	50	{1.61875017517,0.463308206334}	455.1344852622	0.002
P10	100	{4.10439880225,1.51968295711}	231.0855570171	0.003
P11	100	{2.85555816286,4.68224060226}	297.6045429275	0.009
P12	100	{2.81772843636,4.5680021125}	286.7386225628	0.005
P13	250	{0.861049499813,0.377164213194}	549.9465893405	0.008
P14	250	{0.794900743781,0.459653174322}	543.7610648263	0.007
P15	250	{0.844305238343,0.460727996564}	536.8203916176	0.005
P16	500	{0.816903951623,0.164582180338}	6194.3817254403	0.002
P17	500	{0.478124601239,0.957209048476}	1019.3526027819	0.001
P18	500	{0.95138220166,0.698523640367}	4667.4820432420	0.006

Как видно из Таблицы 3.2, каждый из метаэвристических алгоритмов выдал лучшие результаты, очень близкие к результатам, полученным с помощью эвристического алгоритма, для каждой из тестовых задач. Точнее, светлячковый алгоритм выдал 9 результатов (P2, P4, P5, P6, P8, P11, P16, P17 и P18), которые лучше, чем выданные эвристическим и 9 хуже. Улучшенный светлячковый алгоритм выдал 11 результатов (P1, P2, P4, P5, P6, P8, P9, P11, P16, P17 и P18) лучше, чем эвристический алгоритм и 7 немного хуже. Среди результатов,

выданных алгоритмом пчелиной колонии 9 результатов (P2, P4, P5, P6, P8, P9, P10, P11, P12, P16, P17 и P18) лучше, чем результаты эвристического алгоритма, один равный (P3) и 7 худших. Недавно предложенная улучшенная версия алгоритма пчелиной колонии (КО-АПК) позволила найти лучшие или равные результаты по сравнению с эвристическим алгоритмом для всех задач, за исключением P7, где был получен чуть худший результат.

Таблица 3.2. Сравнение наилучших решений СА, УСА, АПК и КО-АПК для 18 задач за 30 запусков

	К-во точек	СА	УСА	АПС	КО-АПС
P1	5	38.6308976454	38.6308975913	38.6309142364	**38.6308975913**
P2	5	32.6409720077	32.6409719926	32.6409728972	**32.6409719926**
P3	5	32.8210029289	32.8210021282	**32.8210021279**	**32.8210021279**
P4	10	18.75284843731	18.7528484372	18.7528484934	**18.7528484372**
P5	10	18.3972444323	18.3972444317	18.3972444324	**18.3972444317**
P6	10	105.9917830210	105.9917830154	105.9917834898	**105.9917830153**
P7	50	723.0047503821	723.0047449142	**723.0047448967**	**723.0047448967**
P8	50	108.9894177207	108.9894138823	**108.9894138815**	**108.9894138815**
P9	50	455.1345178665	455.1344639824	**455.1344639631**	**455.1344639631**
P10	100	231.0855621813	231.0855570229	231.0855570166	**231.0855570165**
P11	100	297.2586360361	297.2586211215	297.2586217396	**297.2586211143**
P12	100	286.7386328552	286.7386225682	**286.7386225626**	**286.7386225626**
P13	250	549.9466236751	549.9465893481	549.9476076576	**549.9465893411**
P14	250	543.7610990888	543.7610648322	543.7611199289	**543.7610648263**
P15	250	536.8204096701	536.8203916405	536.8203922596	**536.8203916167**
P16	500	6193.7933237011	6193.7927948468	**6193.7927947450**	**6193.7927947450**
P17	500	1017.8088815506	1017.8088230560	1017.8091304405	**1017.8088230326**
P18	500	4667.0499616218	4667.0492697179	4667.0515567517	**4667.0492696773**

С точки зрения наилучших результатов, из Таблицы 3.2 видно, что КО-АПК позволяет получить лучшие или равные результаты по сравнению со всеми остальными метаэвристическими алгоритмами, участвующими в рассмотрении. Далее, каждый улучшенный алгоритм, УСА и КО-АПК, в большинстве случаев выдает лучшие результаты, нежели оригинальная версия. Если сравнить производительность оригинальных АПК и СА, можно заключить, что оба алгоритма показывают сходную способность достигать близки к оптимальным результатов, т.е. АПК выдал 9 чуть лучших и 9 чуть худших результатов, чем СА.

Из Таблицы 3.3 можно заметить, что средние значения и стандартные

отклонения, достигнутые с помощью КО-АПК, гораздо лучше результатов, полученных с помощью других метаэвристических алгоритмов. КО-АПК последовательно сходился к одному и тому же решению с тем же значением целевой функции и с очень низким стандартным отклонением.

Таблица 3.3. Сравнение средних значений и стандартных отклонений

	К-во точек	Знач	СА	УСА	АПС	КО-АПС
P1	5	Средн	38.6308981842	38.6308975917	38.6310183806	**38.6308975913**
		Откл	6.00E-7	2.34E-10	1.56E-4	**2.08E-14**
P2	5	Средн	32.6409721528	32.6409719927	32.6409910473	**32.6409719926**
		Откл	1.20E-7	5.42E-11	2.80E-5	**1.21E-14**
P3	5	Средн	32.8210078851	32.8210021309	32.8210021279	**32.8210021279**
		Откл	2.87E-6	1.38E-9	4.49E-11	**1.20E-12**
P4	10	Средн	18.75284850491	18.7528484373	18.7528528692	**18.7528484372**
		Откл	6.53E-8	3.85E-11	5.62E-6	**1.02E-14**
P5	10	Средн	18.3972444696	18.3972444318	18.3972444546	**18.3972444317**
		Откл	3.40E-8	1.98E-11	1.91E-8	**3.04E-15**
P6	10	Средн	105.9917848861	105.9917830164	105.9917978053	**105.9917830153**
		Откл	2.48E-6	6.08E-10	1.54E-5	**1.66E-14**
P7	50	Средн	725.6469298429	723.0047450412	723.0047455780	**723.0047448968**
		Откл	14.227	8.17E-8	3.67E-6	**2.0E-10**
P8	50	Средн	108.989441317	108.9894138938	108.9894138815	**108.9894138815**
		Откл	1.36E-5	5.49E-9	3.06E-12	**2.32E-12**
P9	50	Средн	455.1346305574	455.1344640687	455.1344639631	**455.1344639631**
		Откл	7.52E-5	5.60E-8	**1.49E-13**	8.79E-13
P10	100	Средн	231.0856064139	231.0855570402	231.0855570526	**231.0855570166**
		Откл	2.48E-5	1.09E-08	1.07E-8	**1.30E-11**
P11	100	Средн	297.2586897515	297.2586211470	297.2586932542	**297.2586211143**
		Откл	3.11E-5	1.38E-8	2.01E-4	**2.39E-11**
P12	100	Средн	286.7386885019	286.7386225859	286.7386225626	**286.7386225626**
		Откл	3.69E-5	1.25E-8	1.02E-11	**1.10E-12**
P13	250	Средн	549.9466950535	549.9465893780	549.9913513828	**549.94658934110**
		Откл	4.98E-5	2.53E-08	0.0527	**3.91E-11**
P14	250	Средн	543.7611832652	543.7610648806	543.7655305827	**543.7610648263**
		Откл	6.06E-5	2.72E-8	0.00540	**2.51E-12**
P15	250	Средн	536.8205484167	536.8203916892	536.8204579818	**536.8203916167**
		Откл	6.43E-5	3.28E-8	1.08E-4	**2.83E-12**
P16	500	Средн	6193.7956664708	6193.792795605	6193.792794758	**6193.7927947450**
		Откл	0.0012	4.44E-7	2.26E-8	**1.42E-11**
P17	500	Средн	1017.8092208616	1017.808823222	1017.811042737	**1017.8088230326**
		Откл	2.25E-4	9.49E-8	0.0043	**2.35E-11**
P18	500	Средн	4667.0521716693	4667.049270252	4667.089016694	**4667.0492696773**
		Откл	0.0011	2.61E-7	0.0588	**1.33E-11**

Если сравнить надежность оставшихся трех алгоритмов, то можно заключить, что УСА превосходит СА и АПК. СА выдал 10 лучших средних значений (P1, P2, P4, P6, P9, P11, P13, P14, P17 и P18) и 10 лучших стандартных отклонений по сравнению с АПК. Остальные средние значения и стандартные отклонения лучше в случае с АПК. Следовательно, можно заключить, что СА и АПК имеют сходный уровень надежности.

Таблица 3.4. Среднее процессорное время за 30 запусков

Prob	К-во точек	СА	УСА	АПС	КО-АПС
P1	5	0.011	0.017	0.004	0.004
P2	5	0.010	0.022	0.005	0.004
P3	5	0.011	0.020	0.004	0.004
P4	10	0.012	0.030	0.005	0.006
P5	10	0.011	0.030	0.006	0.006
P6	10	0.012	0.031	0.004	0.005
P7	50	0.028	0.113	0.013	0.016
P8	50	0.026	0.112	0.014	0.014
P9	50	0.028	0.126	0.014	0.010
P10	100	0.045	0.345	0.035	0.043
P11	100	0.045	0.366	0.034	0.043
P12	100	0.043	0.272	0.022	0.025
P13	250	0.100	0.495	0.044	0.044
P14	250	0.108	0.658	0.048	0.051
P15	250	0.097	0.523	0.045	0.042
P16	500	0.188	1.091	0.082	0.093
P17	500	0.180	0.995	0.080	0.091
P18	500	0.187	1.064	0.081	0.094

Согласно результатам из Таблиц 3.2 и 3.3 можно заключить, что улучшенные версии светлячкового алгоритма и алгоритма пчелиной колонии превосходят свои оригинальные версии в решении задачи Вебера со связанными допустимыми зонами, ограниченными дугами. Далее, из этих результатов, а также из приведённых в Таблице 3.1, очевидно, что КО-АПК превосходит по качеству результатов и эвристический алгоритм, и все три метаэвристических алгоритма. Хотя КО-АПК обладает превосходством, тем не менее, надо заметить, что все четыре метаэвристических алгоритма дают равные или лучшие по качеству результаты по сравнению с эвристическим подходом для большинства тестовых

задач.

3.4.4 Анализ затрат времени

В Таблице 3.4 приведено среднее процессорное время за 30 запусков для каждого алгоритма. Данные показывают линейный рост времени для каждого алгоритма с ростом число ограничений.

При сравнении вычислительного времени для алгоритмов видно, что АПК и КО-АПК примерно в 2 раза быстрее, чем СА и примерно в 10 раз быстрее, чем УСА, во всех тестовых задачах. Вычислительное время для АПК и КО-АПК не слишком отличается друг от друга и для числа ограничений равного 500, разница составляет менее 0.1 секунды. Вычислительное время для УСА примерно в 5 раз больше, чем для СА, и для задачи с числом ограничений равным 500, разница составляет около 1 секунды.

Сравнивая эти данные с вычислительными затратами эвристического алгоритма, приведенными в Таблице 3.1, можно заметить, что затраты эвристического алгоритма меньше, чем у всех метавэвристических. Однако, вычислительное время метаэвристических алгоритмо вполне приемлемо, в большинстве случаев разница составляет менее секунды и ее можно считать несущественной.

Результаты главы 3

Задача Вебера с допустимыми зонами, ограниченными дугами, представляет собой задачу невыпуклой оптимизации. Из-за наличия множества локально оптимальных точек сложно глобальный оптимум. Метаэвристические подходы вполне годятся для решения подобных задач, давая качественные результаты за приемлемое время. Проведено сравнение производительности двух выдающихся представителей семейства алгоритмов роевого интеллекта, алгоритма пчелиной колонии (АПК) и светлячкового алгоритма (СА), а также их недавно разработанных улучшенных версий, кроссоверного алгоритма пчелиной

колонии (КО-АПК) и улучшенного светлячкового алгоритма (УСА). Кроме того, в сравнение также был включен эвристический алгоритм, основанный на модифицированной процедуре Вайсфельда.

Четыре метаэвристических алгоритма были протестированы на восемнадцати тестовых задачах с числом ограничений до 500. Результаты показывают, что все четыре метаэвристических алгоритма превосходят эвристический по точности, при заметном преимуществе алгоритма КО-АПК. По вычислительному времени АПК и КО-АПК превосходят СА и УСА. Хотя все эти четыре алгоритма требуют более высоких затрат вычислительных ресурсов, чем эвристический, тем не менее их процессорное время приемлемо и с ростом числа ограничений растет линейно.

Наконец, выявилось, что алгоритм КО-АПК превосходит остальные метаэвристические алгоритмы, участвовавшие в сравнении, как по качеству, так и по надежности результатов.

По результатам этого исследования можно заключить, что метаэвристические подходы могут успешно применяться для решения задач с максимальными и минимальными лимитами расстояния. Кроме того, это исследование способствует применению метаэвристических алгоритмов для решения ряда других сложных задач оптимизации практического применения. Программная реализация алгоритма для задач с мерой расстояния, ограниченной снизу, включена в состав системы автоматизированного формирования и контроля спецпартий электрорадиоизделий космического применения и прошла опытную эксплуатацию на ОАО «Испытательный технический центр – НПО ПМ» (см. Приложение Б).

ЗАКЛЮЧЕНИЕ

В диссертации разработаны новые алгоритмы метода жадных эвристик для одновременного решения серии задач автоматической группировки и размещения объектов с использованием различных метрик и мер расстояния, а также основанных на модели разделения смеси вероятностных распределений, позволяющие эффективно решать широкий круг задач автоматической группировки, к результатам которых предъявляются высокие требования точности и стабильности результата (по значению целевой функции) при большом объеме входных данных и заранее неизвестном числе групп (кластеров).

Цель диссертации достигнута путем решения поставленных задач, а именно:

1. Проведен анализ существующих методов и алгоритмов решения задач автоматической группировки, размещения и иных задач оптимизации. Выявлено, что методы, предполагающие одновременное с решением задачи автоматической группировки решение задачи определения числа групп в данных, обладают достаточно низкой точностью (по значению целевой функции).

2. Разработаны новые алгоритмы метода жадных эвристик со стратегией глобального поиска, реализованной особым генетическим алгоритмом с гетерогенной (смешанной) популяцией решений, содержащей решения задач с различным числом групп (кластеров) и особой модификацией жадных эвристических процедур. При этом в качестве алгоритмов локального поиска, предусмотренных методом жадных эвристик, могут использоваться различные эффективные алгоритмы локального поиска для задач с соответствующей мерой расстояния. Экспериментально доказано, что разработанные алгоритмы позволяют в случае большой размерности практических задач получать результаты одновременно для серии задач, не уступающие по точности и стабильности результата известным алгоритмам решения единственной задачи, при этом позволяя одновременно решать сразу серию задач с различным числом кластеров.

За счет этого повышается эффективность работы систем автоматической группировки без снижения требований к получаемому значению целевой функции.

3. Впервые разработан генетический алгоритм одновременного решения серии задач нечеткой кластеризации данных большой размерности на основе модели разделения смеси вероятностных распределений с различным предполагаемым числом распределений, при заранее известном максимальном числе распределений. Экспериментально доказано, что разработанный алгоритм позволяет в случае большой размерности задач получать результаты одновременно для серии задач, практически не уступающие по достигаемому значению целевой функции и стабильности результата известным алгоритмам решения единственной задачи разделения смеси распределений, при этом позволяя одновременно решать сразу серию задач с различным числом распределений в смеси. За счет этого повышается эффективность работы систем автоматической группировки без снижения требований к получаемому значению целевой функции правдоподобия.

4. Разработан новый эвристический алгоритм решения задачи Вебера с допустимыми зонами, ограниченными окружностями, который может эффективно применяться в составе алгоритма решения p-медианной задачи с мерой расстояния, ограниченной снизу. Показано, что новый алгоритм позволяет быстро получать значительно более точные результаты в сравнении с единственным известным алгоритмом.

СПИСОК ЛИТЕРАТУРЫ

[1]. Kazakovtsev, L.A. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems / L. A. Kazakovtsev, A.N. Antamoshkin. // Informatica.– 2014.– Vol. 38, No. 3.– P.229-240.

[2]. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. From data mining to knowledge discovery in databases: 3 // AI magazine.– 1996.– Vol. 17, № 3.– P. 37.

[3]. Дюк, В. А., Флегонтов, А. В., Фомина, И. К. Применение технологий интеллектуального анализа данных в естественнонаучных, технических и гуманитарных областях // Известия РГПУ им. А.И. Герцена.– 2011.– №138.– С.77-84.

[4]. Tabachnick, B.G. Using Multivariate Statistics, fifth ed. / B.G.Tabachnick, L.S. Fidell.– Boston:Allyn and Bacon.– 2007.– P.980.

[5]. Мандель, И.Д. Кластерный анализ. М.: Финансы и статистика.– 1988.– 176 с.

[6]. Воронцов, К.В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций.– МГУ.– 2007.– Режим доступа: http://www.ccas.ru/voron/download/Clustering.pdf

[7]. Лукьяненко, М.В. Надежность изделий электронной техники в аппаратуре космических аппаратов: учеб. пособие / М. В. Лукьяненко, Н. П. Чурляева, В. В. Федосов ; Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2016. – 188 с.

[8]. Славин, О.А. Алгоритмы распознавания шрифтов в печатных документах // Информационные технологии и вычислительные системы.– 2010.– №3.– С. 27-38.

[9]. Connell, S.D. Writer adaptation for online handwriting recognition / S.D. Connell, A.K. Jain//IEEE Trans. Pattern Anal. Machine Intell.– 2002.– Vol.24, issue 3, P.329–346.

[10]. Jain, A.K. Image segmentation using clustering /A.K. Jain, P. Flynn // Advances in Image Understanding.– IEEE Computer Society Press.– 1996.– P.65–83.

[11]. Арлазаров, В.В. Структурный анализ текстовых полей в системах потокового ввода оцифрованных документов /В.В. Арлазаров, В.М. Кляцкин, О.А.

Славин // Труды ИСА РАН.– 2015.– Т. 65, вып. 1.– С.75-81.

[12]. Shi, J. Normalized cuts and image segmentation / J. Shi, J. Malik // IEEE Trans. Pattern Anal. Machine Intell.– 2000.– Vol.22.– P. 888–905.

[13]. Борисенко, В.И. Сегментация изображения (состояние проблемы) / В.И.Борисенко, А.А.Златопольский, И.Б. Мучник// Автомат. и телемех.– 1987.– вып. 7.– С.3-56.

[14]. Iwayama, M. Cluster-based text categorization: A comparison of category search strategies / M. Iwayama, T. Tokunaga // Proc. 18th ACM Internat. Conf. on Research and Development in Information Retrieval.– 1995.– P. 273–281.

[15]. Барахнин, В.Б. Кластеризация текстовых документов на основе составных ключевых термов / В.Б. Барахнин, Д.А.Ткачев// Вестник НГУ. Серия: Информационные технологии.– 2010.– Т.8, вып.2.– С. 5-14.

[16]. Барахнин, В.Б. О задании меры сходства лоя кластеризации текстовых документов / В.Б. Барахнин, В.А. Нехаева, А.М. Федотов // Вестник НГУ. Серия: Информационные технологии.– 2008.– Т.6, вып.1.– С.3-9.

[17]. Bhatia, S. Conceptual clustering in information retrieval / S. Bhatia, J. Deogun // IEEE Trans. Systems Man Cybernet.– 1998.– Vol. 28 (B).– P. 427–436.

[18]. Berry, M.J.A. Data Mining techniques: for marketing, sales, and customer relationship management, 2nd ed. /Berry M.J.A., Linoff G.S.– [s.l.]: Wiley.– 2004.– P.464.

[19]. Галямов, А.Ф. Управление взаимодействием с клиентами коммерческой организации на основе методов сегментации и кластеризации клиентской базы / А.Ф. Галямов, С.В. Тархов // Вестник УГАТУ.– 2014.– Т.18, № 4(65).– С.149-156.

[20]. Hu, J. Statistical methods for automated generation of service engagement staffing plans / J. Hu, B.K. Ray, M. Singh // IBM J. Res. Dev.– 2007.– Vol. 51, issue 3.– P. 281–293.

[21]. Федосов, В.В., Орлов, В.И. Минимально необходимый объем испытаний изделий микроэлектроники на этапе входного контроля // Известия высших учебных заведений. Приборостроение.– 2011.– т.54. № 4.– С.58-62.

[22]. Харченко, В.С., Юрченко, Ю.Б. Анализ структур отказоустойчивых бортовых комплексов при использовании электронных компонентов Industry // Технология и конструирование в электронной аппаратуре.– 2003.– №2.– С.3-10.

[23]. Куклин, В.И, Орлов, В.И., Федосов, В.В. Результаты работ по обеспечению качества электрорадиоизделий отечественного производства для комплектования бортовой аппаратуры космических аппаратов за период 01.2008г.– 06.2009г. // VIII Российская научно-техническая конференция. Электронная компонентная база космических систем.– М.: 2009.– С.64-66.

[24]. Федосов, В.В., Куклин, В.И., Орлов, В.И., Исляев, Ш.Н. и др. Технический отчет. Космический аппарат «SESAT» со сроком активного функционирования 10 лет. Принципы, методы и результаты комплектации аппаратуры электрорадиоизделиями // ФГУП «НПО ПМ им. академика Решетнева» .– 1999.– 408 с.

[25]. Перечень ЦК-1/96. Изделия электронной техники, допускаемые для применения в аппаратуре космического аппарата «Ямал» с 10-летним сроком активного существования // АО ИТЦ «Циклон».– 1997.– 90 с.

[26]. Решение № SST-TP-97006 о квалификации электрорадиоизделий на соответствие требованиям космического аппарата с 10-летним сроком активного существования (Редакция 1-97) // АО ИТЦ «Циклон».– 1997.– 108 с.

[27]. Модель околоземного космического пространства: В 3-х т. Т.3 / Под ред. академика С.Н. Вернова. Издание седьмое // М.: МГУ.– 1983.– 133 с.

[28]. Стойкость изделий электронной техники к воздействию факторов космического пространства и электрических импульсных перегрузок: Справочник. Т. XII. 4-е изд. Книга 2. Термовакуумные и электрические воздействия // ВНИИ «Электронстандарт».– 1990.– 162 с.

[29]. Пиз, Р.Л, Джонстон, А.Х., Азаревич, Дж.Л. Радиационные испытания полупроводниковых приборов для космической электроники // ТИИЭР.– 1988.– Т.76, № 11.– С.126-145.

[30]. Радиационная стойкость бортовой аппаратуры и элементов космических аппаратов // I Всесоюзная научно-техническая конференция.

Материалы конференции. Томск.– 1991.– 257с.

[31]. Радиационная стойкость материалов радиотехнических конструкций: Справочник. Под ред. Н.А Сидорова, В.К. Князева // М.: Советское радио.– 1976.– 567 с.

[32]. Малышев, М.М., Малинин, В.Г., Куликов, И.В., Торгашов, Ю.Н, Ужегов, М.В. Методология оценки радиационной надежности ИЭТ в условиях низкоинтенсивных ионизирующих излучений. / В сб. Радиационно-надежностные характеристики изделий электронной техники в экстремальных условиях эксплуатации. Под редакцией Ю.Н. Торгашова // СПб.: Издательство РНИИ «Электронстандарт».– 1994.– 96с.

[33]. Мырова, Л.О., Чепиженко, А.З. Обеспечение стойкости аппаратуры связи к ионизирующим и электромагнитным излучениям. 2-е изд., перераб. и доп. // М.: Радио и связь.– 1988.– 296 с.

[34]. Кононов, В.К., Малинин, В.Г., Оспищев, Д.А., Попов, В.Д. Отбраковка потенциально-ненадежных интегральных микросхем с использованием радиационно-стимулирующего метода / В сб. Радиационно-надежностные характеристики изделий электронной техники в экстремальных условиях эксплуатации. Под редакцией Ю.Н. Торгашова // СПб.: Издательство РНИИ «Электронстандарт».– 1994.– 96 с.

[35]. Drezner, Z. Facility location: applications and theory / Z. Drezner, H. Hamacher.– Berlin:Springer-Verlag.– 2004.– P.460.

[36]. Farahani, R. Facility location: Concepts, models, algorithms and case studies / R. Z. Farahani and M. Hekmatfar (eds.)// Berlin Heidelberg:Springer-Verlag.– 2009.– 549 p.

[37]. Бельц, Е.А. Оптимизация размещения предприятий с учетом минимально допустимых расстояний / Е.А. Бельц, А.А. Колоколов // Вестн. Ом. ун-та.– 2012.– No 4.– С.13-16.

[38]. Кочетов, Ю.А. Двухуровневые задачи размещения / Кочетов Ю.А. // Труды ИВМ и МГ / Серия Информатика.– Новосибирск: Изд. ИВМиМГ СО РАН.– 2007.– Вып. 7.– С. 97–104.

[39]. Pfeiffer, B. A unified model for Weber problem with continuous and network distance / B. Pfeiffer, Klamroth. K. // Computers and OR.– Vol. 35, No. 2.– 2008.– P.312-326.

[40]. Cooper, L. The transportation-location problem //Oper. Res.– 1972.– Vol.20, No.1.– P. 94-108.

[41]. Lloyd, S.P. Least Squares Quantization in PCM // IEEE Transactions on Information Theory.– 1982.– Vol. 28.– P. 129-137.

[42]. Fermat P. de (1643), Ed. H.Tannery, ed., «Oeuvres», vol. 1, Paris 1891, Supplement: Paris 1922, cc. 153.

[43]. Torricelli, E. Opere de Evangelista Torricelli / E.Torricelli, G.Loria, G.Vassura // English edition.–Vol I.– Part 2.– Faenza.– 1919.– P. 90-97.

[44]. Kirszenblat, D. Dubins networks: Thesis / D. Kirszenblat.– Melbourne: Department of Mathematics and Statistics of the University of Melbourne.– 2011.– 56p.

[45]. Weber, A. Ueber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes/ A. Weber, [Verlag von] J. C. B. Mohr.– Tuebingen: Mohr.– 1922.– 209 P.

[46]. Hale, T.S. Location science research: a review / T. S. Hale, C. R. Moberg // Annals of Operations Research.– 2003.– Vol. 123.– P.21-35.

[47]. Jarník, V., Kössler, O. «O minimálních grafech obsahujících n daných bodu» // Ĉas, Pêstování Mat. (Essen).–1934.– T. 63.– 223-235.

[48]. Weiszfeld, E. Sur le point sur lequel la somme des distances de n points donnes est minimum/ E. Weiszfeld // Tohoku Mathematical Journal.– 1937.– Vol. 43, No.1.– P.335–386.

[49]. Sturm, R. Ueber den Punkt kleinster Entfernungssumme von gegebenen Punkten / R. Sturm // J. Rein. Angew. Math.– 1884.– Vol.97.– P. 49–61.

[50]. Beck, A. Weiszfeld's Method: Old and New Results / A. Beck, S. Sabach // J. Optim. Theory Appl.– 2015.– Vol.164, Iss.1.– P.1-40 DOI 10.1007/s10957-014-0586-7.

[51]. Drezner, Z. The fortified Weiszfeld algorithm for solving the Weber problem / Z. Drezner // IMA Journal of Management Mathematics.– 2013.– Vol.26.– P.1-9. DOI: 10.1093/imaman/dpt019

[52]. Hakimi, S.L. Optimum locations of switching centers and the absolute centers and medians of a graph / L.Hakimi. S. // Operations Research.– 1964.– Vol. 12, issue 3.– P.450–459.

[53]. Hakimi, S.L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems / S.L. Hakimi // Operations Research.– 1965.– Vol. 13, No. 3.– P.462–475.

[54]. Deza, M.M. Encyclopedya of Distances / M. M. Deza, E. Deza. // Berlin, Heidelberg: Springer-Verlag.– 2009.– 590 p.

[55]. Deza, M. Distances in pattern recognition / Deza M.– 2007.– Режим доступа: www.picb.ac.cn /pattern07/ SLIDES/ 2007_ summerschool_ Shanghai_ Deza.pdf

[56]. Deza, M.M. Metrics on Normed Structures / M. M. Deza, E. Deza. // Encyclopedia of Distances.– Berlin Heidelberg:Springer.– 2013.– P.89-99, DOI: 10.1007/978-3-642-30958-85.

[57]. Masuyama, S. The Computational Complexity of the m-Center Problems on the Plane / S. Masuyama, T. Ibaraki, T. Hasegawa //The Transactions of the Institute of Electronics and Communication Engineers of Japan.– 1981.– Vol. 64E.– P. 57-64.

[58]. Megiddo, N. On the Complexity of Some Common Geometric Location Problems / N. Megiddo, K. Supowit // SIAM Journal on Computing.– 1984.– Vol. 13.– P. 182-196.

[59]. Morris, J.G. Convergence of the Weiszfeld algorithm for Weber problems using a generalized ”distance” function / J. G. Morris // Operations Research.– 1981.– Vol.29.– P.37-48.

[60]. Wesolowsky, G.O. A Nonlinear Approximation Method for Solving a Generalized Rectangular Distance Weber Problem / G.O. Wesolowsky, R.F. Love // Management Science.– 1972.– Vol. 18, No.11.– P. 656-663.

[61]. Забудский, Г.Г. Решение задачи размещения в евклидовом пространстве с запрещенной областью / Г.Г. Забудский, И.В. Нежинский // Вестник Омского университета.– 1999.– Т.2.– С. 17-19.

[62]. Klamroth, K. Single-facility location problems with barriers // Springer

Verlag, Berlin, Heilderberg.– 2002.– 201 p.

[63] Pilotta, E.A., Torres, G.A. A projected Weiszfeld algorithm for the box-constrained Weber location problem // Applied Mathematics and Computation.– 2011.– Vol.218, Issue 6.– P.2932-2943.

[64]. Struyf, A. Clustering in an Object-Oriented Environment / A. Struyf, M. Hubert, P. Rousseeuw // Journal of Statistical Software.– 1997.– issue 1 (4).– P. 1-30.

[65]. Kaufman, L. Finding groups in data: an introduction to cluster analysis / L. Kaufman, P.J. Rousseeuw.– New York:Wiley.– 1990.– P.368.

[66]. Moreno-Perez, J.A. A Parallel Genetic Algorithm for the Discrete p-Median Problem / J.A. Moreno-Perez, J.L. Roda Garcia, J.M. Moreno-Vega // Studies in Location Analysis.– 1994.– issue 7.– P.131-141.

[67]. Wesolowsky, G. The Weber problem: History and perspectives / G. Wesolowsky // Location Science.– 1993.– No. 1.– P.5-23.

[68]. Drezner, Z. Trajectory Method for the Optimization of the Multifacility Location Problem with lp Distances / Z. Drezner, Wesolowsky G.O.A // Management Science.– 1978.– V.24.– P.1507–1514.

[69]. Reza, A.W. A Comprehensive Study of Optimization Algorithm for Wireless Coverage in Indoor Area / A.W. Reza, K. Dimyati, K.A.Noordin, A.S.M.Z. Kausar, Md.S. Sarker // Optimization Letters.– September 2012 [Электронный ресурс] Режим доступа DOI: 10.1007/s11590-012-0543-z .

[70]. Kazakovtsev, L. A. Wireless Coverage Optimization Based on Data Provided by Built-in Measurement Tools / L. A. Kazakovtsev // WASJ.– 2013.– Special Issue on Techniques and Technologies.– P. 8–15.

[71]. Казаковцев, Л.А., Гудыма, М.Н., Ступина, А.А., Кириллов, Ю.И. Задача выбора оптимального размещения элементов беспроводной сети // Современные проблемы науки и образования. – 2013.– № 3.– Режим доступа: https:// www.science-education.ru/ru/article/view?id=9551

[72]. Duran, B. S. Cluster Analysys: a Survey / B. S. Duran, P. L. Odell // Berlin-Heidelberg-New York:Springer-Verlag.– 1977.– 140 P.

[73]. 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.

[74]. Rui, X. Survey of Clustering Algorithms / X. Rui, D.Wunsch // IEEE Transactions on Neural Networks.– 2005.– Vol. 16, issue 3.– P.645-678, doi: 10.1109/TNN.2005.845141.

[75]. Meira, L.A.A. A Continuous Facility Location Problem and Its Application to a Clustering Problem / L.A.A. Meira, Miyazawa. F.K. // Proceedings of the 2008 ACM symposium on Applied computing (SAC '08).–New York: ACM.– 2008.– P.1826-1831, doi:10.1145/1363686.1364126.

[76]. Hansen, P. Analysis of global k-means, an Incremental Heuristic for Minimum Sum of Squares / P. Hansen, E. Ngai, B. Cheung, N. Mladenovic // Journal of Classifcation.– 2005.– Vol. 22(3).– P. 287-310.

[77]. Still, S. Geometric Clustering using the Information Bottleneck method / S. Still, W. Bialek, L. Bottou // Advances In Neural Information Processing Systems 16 / Eds.:S. Thrun,L. Saul, and B. Scholkopf.– Cambridge:MIT Press.– 2004 [Электронный ресурс] Режим доступа URL http://papers.nips.cc/paper/2361-geometric-clustering-using-the-information-bottleneck-method.pdf

[78]. Казаковцев, Л.А., Антамошкин, А.Н., Гудыма М.Н. Параллельный алгоритм для р-медианной задачи // Системы управления и информационные технологии.– 2013.– № 2.1.– С.124-128.

[79]. Klastorin, T.D. The p-Median Problem for Cluster Analysis: A Comparative Test Using the Mixture Model Approach / T.D. Klastorin // Management Science.– 1985.– Vol.31, No.1.– P. 84-95.

[80]. Hansen, P. Variable Neighborhood Search / P. Hansen, N. Mladenovic // Search Methodology / E.K.Bruke, G.Kendall [eds.].– Springer US.– 2005.– P. 211-238, doi: 10.1007/0-387-28356-0_8.

[81]. Balcan, M.–F. Distributed k-means and k-median clustering on general communication topologies / M.–F. Balcan, S. Ehrlich, Y. Liang // Advances in Neural Information Processing Systems.– 2013.– P. 1995-2003.

[82]. Belacel, N. Fuzzy J-Means: a new heuristic for fuzzy clustering / N. Belacel,

P. Hansen, N. Mladenovic // Pattern Recognition.– 2002.– Vol.35.– P. 2193–2200.

[83]. Har-Peled, S. Coresets for k-Means and k-Median Clustering and their Applications / S. Har-Peled, S. Mazumdar // Proc. 36th Annu. ACM Sympos. Theory Comput.– 2003.– P. 291-300.

[84]. Kuehn, A.A. A heuristic program for locating warehouses/A.A.Kuehn, M.J.Hamburger// Management Science.– 1963.– 9(4).– P.643-666.

[85]. Vardi, Y. A modified Weiszfeld algorithm for the Fermat-Weber location problem / Y. Vardi, C.–H. Zhang // Mathematical Programming.– Vol. 90, No. 3.– 2001.– P. 559-566, DOI: 10.1007/s101070100222.

[86]. Трубин, В.А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой / В.А. Трубин // Кибернетика.– 1978.– № 6, С. 67-70, DOI:10.1007/BF01070282.

[87]. Cabot, A. V. A Network Flow Solution to a Rectilinear Distance Facility Location problem / A. V. Cabot, R. L. Francis, M. A. Stary // American Institute of Industrial Engineers Transactions.– 1970.– Vol. 2.– P. 132-141.

[88]. Гудыма, М.Н. Алгоритм решения задачи размещения для некоторых специальных метрик / М.Н. Гудыма // Системы управления и информационные технологии.– 2013.– № 4(54).– С. 20-23.

[89]. Kazakovtsev, L.A. Decomposition of the Continuous Weber Problem with French Metro Metric / L.A. Kazakovtsev, P.S. Stanimirovic, M. Ciric // Proceedings of the International Conference on Problems of Modern Agrarian Science Collected scientific works.– Krasnoyarsk: KrasGAU.– 2012.– 5 P. [Электронный ресурс] Режим доступа URL http://www.kgau.ru/img/konferenc/2012/e2.doc

[90]. Staminirovic, P.S. Single-facility Weber location problem based on the Lift metric / Predrag S. Staminirovic, Maria Ciric, Lev A. Kazakovtsev, Idowu A. Osinuga // Facta Universitatis (Nis), Ser. Math. Inform.– 2012.– Vol. 27, issue 2.– P. 175-190.

[91]. Cooper, L. An extension of the generalized Weber problem / L. Cooper // Journal of Regional Science.– 1968.– Vol.8, issue 2.– P. 181-197.

[92]. Cooper, L. Location-allocation problem / L. Cooper // Operations Research.– 1963.– Vol. 11.– P. 331-343.

[93]. Bailey, K. Numerical Taxonomy and Cluster Analysis / K. Bailey // Typologies and Taxonomies / K. Bailey. - [s.l.]: Sage Pubns.– 1994.– 89 P. DOI:10.4135/9781412986397.

[94]. Tan, P.–N. Cluster Analysis: Basic Concepts and Algorithms / P.–N. Tan, M. Steinbach, V. Kumar // Introduction to Data Mining.– [s.l.]:Addison-Wesley.– 2006.– P. 487–567.

[95]. Drineas, P. Clustering Large Graphs via the Singular Value Decomposition / P. Drineas, A. Frieze, R. Kannan, S. Vempala, V. Vinay // Machine learning.– 1999.– Vol. 56, No. 1-3.– P. 9-33.

[96]. Aloise, D. NP-Hardness of Euclidean Sum-of-Squares Clustering / D. Aloise, A. Deshpande, P. Hansen, P. Popat // Machine Learning.– 2009.– Vol. 75.– P. 245-249, DOI:10.1007/s10994-009-5103-0.

[97]. Resende, M.G.C. Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures / M.G.C. Resende // TutORials in Operations Research/ Zhi-Long Chen and S. Raghavan [eds.].– INFORMS.– 2008.– P. 295–319.

[98]. Resende, M.G.C. Scatter search and pathrelinking: Fundamentals, advances, and applications / M.G.C. Resende, C.C. Ribeiro, F.Glover, R. Marti // Handbook of Metaheuristics [2nd Edition] / M. Gendreau and J.–Y. Potvin [eds.].– [s.l.]: Springer.– 2010.– P. 87–107.

[99]. Hansen P. Solving large p-median clustering problems by primaldual variable neighborhood search / P. Hansen, J. Brimberg, D. Urosevic, N. Mladenovic // Data Mining and Knowledge Discovery.– 2009.– Vol.19, No. 3.– P. 351–375.

[100]. Bovet, D.P. Introduction to the Theory of Complexity / D.P. Bovet, D.P. Crescenzi // New York: Prentice Hall Intern.– 1994.– 282 p.

[101]. Leeuwen, J. van. Algorithms and complexity: Handbook of Theoretical Computer Science /J. van Leeuwen, [ed.].–Vol.A, . Amsterdam: Elsevier.– 1998.– 998 P.

[102]. Turing, A.M. On Computable Numbers, with an Application to the Entscheidungs problem / A.M.Turing // Proceedings of the London Mathematical Society.– 1937.– Vol. 42.– P.230-265.

[103]. Turing, A.M. On Computable Numbers, with an Application to the

Entscheidungsproblem: A correction / A.M. Turing //Proceedings of the London Mathematical Society.– 1937.– Vol. 43, issue 6.– P. 544.

[104]. The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma / J. Copeland [ed.] // Oxford: Clarendon Press (Oxford University Press).– 2004.– 622 P.

[105]. Steinhaus, H. Sur la division des corps materiels en parties // Bull. Acad. Polon. Sci.– 1956.– Cl. III, vol IV.– P.801-804.

[106]. Zhang, T. BIRCH: An Efficient Data Clustering Method for Very Large Databases / T. Zhang, R. Ramakrishnan, M. Livny // Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96), New York: ACM.– 1996.– P. 103-114, DOI: 10.1145/233269.233324.

[107]. O'Callaghan, L. Streaming-Data Algorithms for High-Quality Clustering / L. O'Callaghan, A. Meyerson, R. Motwani, N. Mishra, S. Guha // Proceedings 18th International Conference on Data Engineering.– 2002.– P. 685-694, DOI: 10.1109/ICDE.2002.994785.

[108]. Ackermann, M.R. StreamKM: A Clustering Algorithm for Data Streams / M.R.Ackermann et. al. // J. Exp. Algorithmics.– May 2012.– Vol.17, Article 2.4 [Электронный ресурс] Режим доступа DOI:10.1145/2133803.2184450.

[109]. Guha, S. CURE: An efficient clustering algorithm for large databases / S. Guha, R. Rastogi, K. Shim // SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data.– New York: ACM.– 1998.– P. 73-84.

[110]. Eisenbrand, F. Approximating connected facility location problems via random facility sampling and core detouring / F. Eisenbrand, F. Grandoni, T. Rothvosz, G. Schafer // Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008).– New York: ACM.– 2008.– P. 1174-1183.

[111]. Czumaj, A. Sublinear Time Approximation for Clustering Via Random Sampling / A. Czumaj, Sohler. C. // Automata, Languages and Programming, Lecture Notes in Computer Science.– Berlin Heidelberg: Springer.– 2004.– Vol. 3142.– P. 396-407, DOI: 10.1007/978-3-540-27836-8 35.

[112]. Jaiswal, R. A Simple D2-Sampling Based PTAS for k-Means and Other

Clustering Problems / R. Jaiswal, A. Kumar, S. Sen // Algorithmica.– 2014.– Vol. 70, issue 1.– P. 22-46, DOI: 10.1007/s00453-013-9833-9.

[113]. Phoungphol, P. Sample Size Estimation with High Confidence for Large Scale Clustering / P. Phoungphol, Y. Zhang // Proceedings of the 3rd International Conference on Intelligent Computing and Intelligent Systems.– 2011.– [Электронный ресурс] Режим доступа URL http://www.cs.gsu.edu/phoungphol1/paper/icis2011.pdf

[114]. Mladenovic, N. The p-median problem: A survey of metaheuristic approaches / N. Mladenovic, J. Brimberg, P. Hansen, . A. Moreno-Perez // European Journal of Operational Research.– 2007.– Vol. 179, issue 3.– P.927–939.

[115]. Arthur, D. k-Means++:R[14]C The Advantages of Careful Seeding/D. Arthur and S. Vassilvitskii//Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms [SODA ’07].– SIAM.– 2007.– P.1027–1035.

[116]. 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, G.Kay. M. // Computers and Operations Research.– 1996.– Vol. 23.– P. 587-596.

[117]. Maulik, U. Genetic Algorithm-Based Clustering Technique / U. Maulik, S. Bandyopadhyay // Pattern Recognition.– 2000.– Vol. 33.– P. 1455-1465.

[118]. Krishna, K. Genetic K-means algorithm / K. Krishna, M. Murty // IEEE Transaction on System, Man and Cybernetics - Part B.– 1999.– Vol.29.– P. 433-439.

[119]. Neema, M.N. New Genetic Algorithms Based Approaches to Continuous p-Median Problem / M.N. Neema, K.M. Maniruzzaman, A. Ohgai // Netw. Spat. Econ.– 2011.– Vol.11.– P.83–99, DOI:10.1007/s11067-008-9084-5.

[120]. Alp, O. An Efficient Genetic Algorithm for the p-Median Problem / O. Alp, E. Erkut, Z. Drezner // Annals of Operations Research.– 122 (1-4).– 2003.– P. 21–42, doi 10.1023/A:1026130003508.

[121]. Lim, A. A fixed-length subset genetic algorithm for the p-median problem / A. Lim, Z. Xu // Lecture notes in computer science.– 2003.– Vol. 2724.– P. 1596-1597.

[122]. Sheng, W. A genetic k-medoids clustering algorithm / W. Sheng, X. Liu //Journal of Heuristics.– 2006.–Vol.12, No.6.– P. 447-466.

[123]. Казаковцев, Л. А. Метод жадных эвристик для систем автоматической группировки объектов: диссертация ... доктора технических наук: 05.13.01 / Казаковцев Лев Александрович; [Место защиты: Сибирский федеральный университет].– Красноярск, 2016.

[124]. Гудыма, М.Н., Казаковцев, Л.А., Антамошкин, А.Н. Решение серий задач автоматической группировки промышленной продукции // Экономика и менеджмент систем управления.– 2016.– Т. 22. № 4.– С. 80-87.

[125]. Гудыма, М.Н., Казаковцев, Л.А. Эволюционные алгоритмы решения серии задач автоматической группировки с динамическими и гетерогенными популяциями // Системы управления и информационные технологии, №2(68), 2017. – С. 33-38.

[126]. Arabas, J., Michalewicz, Z., Mulawka, J. GAVaPS – a genetic algorithm with varying population size. // Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ.– 1994.– P.73–78.

[127]. Fernandes, C., Rosa, A. A study on non-random mating and varying population size in genetic algorithms using a royal road function // Proceedings of the 2001 Congress on Evolutionary Computation.– vol. 1.– P.60-66.

[128]. Vellev, S. An Adaptive Genetic Algorithm with Dynamic Population Size for Optimizing Join Queries // International Conference: Intelligent Information and Engineering Systems, INFOS 2008, Varna, Bulgaria, June-July, 2008.

[129]. Schlierkamp-Voosen, D., Mühlenbein, H. Adaption of population sizes by competing subpopulations // Proceedings of the 1996 IEEE Conference on Evolutionary Computation.– 20-22 May 1996.– P.330-335.

[130]. Eiben, A.E., Marchiori, E., Valko, V.A. Evolutionary algorithms with on-the-fly population size adjustment / X. Yao et al., (eds), Parallel Problem Solving from Nature.– 2004.– PPSN VIII, LNCS 3242.– P.41–50.

[131]. Rajakumar, B.R., Aloysius, G. APOGA: An Adaptive Population Pool Size based Genetic Algorithm‖ // AASRI Procedia.– 2013.– vol. 4.– P.288-296.

[132]. Yen, G. G., Lu, H. Dynamic population size in multiobjective evolutionary algorithm // Proc. 9 th IEEE Cong. Evol. Comput.– 2002.– P.1648-1653.

[133]. Harik, G., Lobo, F. A parameter-less genetic algorithm // Proceedings of GECCO 1999: the Genetic and Evolutionary Computation Conference.– 1999.– volume 1.– P.258–265.

[134]. Smorodkina, E.A., Tauritz, D.R. Greedy Population Sizing for Evolutionary Algorithms // Proceedings of CEC 2007 - IEEE Congress on Evolutionary Computation.– 2007.– P.2181–2187.

[135]. Eiben, A., Schut, M., deWilde, A. Is self-adaptation of selection pressure and population size possible? // Proceedings of PPSN IX: the Ninth International Conference on Parallel Problem Solving from Nature.– 2006.– P.900-909.

[136]. 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.

[137]. Семенкин Е. С., Сергиенко Р. Б. Коэволюционный генетический алгоритм решения сложных задач условной оптимизации // Вестник СибГАУ.– 2009.– №2.– С.17-21.

[138]. Joines, J., Houck, C. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with gas. // Proceedings of the First IEEE International Conference on Evolutionary Computation.– 1994.– P.579-584.

[139]. Bean, J. C., Hadj-Alouane, A. B. A dual genetic algorithm for bounded integer programs. Technical Report TR 92-53 // Department of Industrial and Operations Engineering, The University of Michigan.– 1992.– 20 p.

[140]. Back, T., Hoffmeister, F., Schwefel, H.–P. A survey of evolution strategies. // R. K. Belew and L. B. Booker (Eds.), Proceedings of the 4th International Conference on Genetic Algorithms.– 1991.– P.2-9.

[141]. Сергиенко, Р. Б. Разработка турнирного метода перераспределения ресурсов между подпопуляциями в коэволюционном алгоритме / Р. Б. Сергиенко // Инновационные недра Кузбасса. IT-технологии: сб. науч. тр. Кемерово: ИНТ, 2007.– С. 401–404.

[142]. Ma, Z.M., Krings, A.W. Dynamic populations in genetic algorithms // Proceedings of the 2008 ACM symposium on Applied computing.– Fortaleza, Ceara, Brazil.– March 16-20, 2008.– P.1807-1811, doi>10.1145/1363686.1364119.

[143]. Kazakovtsev, L.A. Modified Genetic Algorithm with Greedy Heuristic for Continuous and Discrete p-Median Problems / L.A. Kazakovtsev, V.I. Orlov, A.A. Stupina, V.L. Kazakovtsev //FactaUniversitatis (Nis) Series Mathematics and Informatics.– 2015.– Vol. 30, No. 1.– P. 89-106.

[144]. Goldberg, D. E., Korb, B., Deb, K. Messy genetic algorithms: motivation, analysis, and first results // Complex Systems.– 1989.– 3(5).– P.493-530.

[145]. Burke, D. S., De Jong, K. A., Grefenstette, J. J., Ramsey, C. L., Wu, A. S. Putting more genetics into genetic algorithms // Evolutionary Computation.– 1998.– 6(4).– P.387-410.

[146]. Wu, A. S., Garibay, I. (2002). The proportional genetic algorithm: Gene expression in a genetic algorithm // Genetic Programming and Evolvable Hardware.– 2002.– 3(2).– P.157-192.

[147]. Bassett, J. K., De Jong, K. A. Evolving behaviors for cooperating agents // International Symposium on Methodologies for Intelligent Systems.– 2000.– P.157-165.

[148]. Wu, A. S., Schultz, A. S., Agah, A. Evolving control for distributed micro air vehicles. // Proceedings of IEEE Computational Intelligence in Robotics and Automation Engineers Conference.– 1999.– P.174-179.

[149]. Wu, A. S., Stringer, H. Learning using chunking in evolutionary algorithms. // Proceedings of the 11th Conference on Computer-Generated Forces and Behavior Representations.– 2002.– P.243-254.

[150]. Grefenstette, J. J., Ramsey, C. L., Schultz, A. C. (1990). Learning sequential decision rules using simulation models and competition // Machine Learning.– 1990.– Vol.5, Iss.4.– P.355-381.

[151]. Harvey, I. (1992a). Species adaptation genetic algorithms: a basis for a continuing SAGA. // Tettamanzi, A., editor, Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems, MIT Press,

Cambridge, MA.– 1992.– P.346-354.

[152]. Kavka, C., Schoenauer, M. Voronoi diagrams based function identification. // GECCO 2003: Proceedings of the Genetic and Evolutionary Computation Conference, LNCS 2723.– 2003.– P.1089-1100.

[153]. Blickle, T., Thiele, L. Genetic programming and redundancy. // Proceedings of Genetic Algorithms within the Framework of Evolutionary Computation.– 1994.– P.33-38.

[154]. Koza, J. R. Genetic Programming. On the Programming of Computers by Means of Natural Selection // MIT Press, Cambridge, MA.– 1992.– 840p.

[155]. Soule, T., Foster, J. A., Dickinson, J. Code growth in genetic programming. // Genetic Programming 1996: Proceedings of the First Annual Conference.– 1996.– P.215-223.

[156]. Tackett, W. A. Recombination, Selection, and the Genetic Construction of Computer Programs // Doctoral Dissertation, University of Southern California, Department of Electrical Engineering Systems.– 1994.– 302p.

[157]. Langdon, W. B., Poli, R. Fitness causes bloat. // Proceedings of the Second On-line World Conference on Soft Computing in Engineering Design and Manufacturing.– 1997.– P.13-22.

[158]. Poli, R., Langdon, W. B. A new schema theory for genetic programming with onepoint crossover and point mutation. // Genetic Programming 1997: Proceedings of the Second Annual Conference.– 1997.– P.278-285.

[159]. Stephens, C. R., Waelbroeck, H. Effective degrees of freedom in genetic algorithms and the block hypothesis. // Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97).– 1997.– P.34-40.

[160]. Stephens, C. R., Waelbroeck, H. Schemata evolution and building blocks // Evolutionary Computation.– 1999.– Vol.7(2).– P.109-124.

[161]. Poli, R. Exact schema theorem and effective fitness for GP with one-point crossover. // Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000).– 2000.– P.469-476.

[162]. McPhee, N. F., Poli, R. A schema theory analysis of the evolution of size in

genetic programming with linear representations. // Genetic Programming, Proceedings of EuroGP'2001, LNCS 2038.– 2001.– P.108-125.

[163]. Poli, R., McPhee, N. F. Exact schema theorems for GP with one-point and standard crossover operating on linear structures and their application to the study of the evolution of size. // Genetic Programming, Proceedings of EuroGP'2001, LNCS 2038.– 2001.– P.126-142.

[164]. Rowe, J. E., McPhee, N. F. The effects of crossover and mutation operators on variable length linear structures. // Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001.– 2001.– P.535-542.

[165]. Brie, A.H., Morignot, P. Genetic Planning Using Variable Length Chromosomes // Proceedings of the XV International Conference on Automated Planning and Scheduling (ICAPS 2005).– 2005.– P.320-330.

[166]. Hutt, B., Warwick, K. (2007) Synapsing Variable-Length Crossover: Meaningful Crossover for Variable-Length Genomes // IEEE Transactions on Evolutionary Computation.– 2007.– Vol.11, Iss.1.– P.118-131.

[167]. Dempster, A.P., Laird, N. M., Rubin, D.B. Maximum-likelihood from incomplete data via the EM algorithm // J. Royal Statist. Soc. Ser. B (methodological).– 1977.– Vol.39.– P.1-38.

[168]. Hasselblad, V. Estimation of parameters for a mixture of normal distributions // Technometrics.– 1966.– Vol.8.– P.431-444.

[169]. Hasselblad, V. Estimation of finite mixtures of distributions from the exponential family // J. Amer. Statist. Assoc.– 1969.– Vol.64.– P.1459-1471.

[170]. Behboodian, J. On a mixture of normal distributions // Biometrika.– 1970.– Vol. 57, Part 1.– P.215-217.

[171]. Day, N.E. Estimating the components of a mixture of normal distributions // Biometrika.– 1969.– Vol.56.– P.463-474.

[172]. Wolfe, J.H. Pattern clustering by multivariate mixture analysis // Multivariate Behavioral Res.– 1970.– Vol.5.– P.329-350.

[173]. Tan, W.Y., Chang, W.C. Convolution approach to genetic analysis of quantitative characters of self-fertilized population // Biometrics.– 1972.– Vol.28.–

P.1073-1090.

[174]. Hosmer(Jr.), D.W. On MLE of the parameters of a mixture of two normal distributions when the sample size is small // Comm. Statist.– 1973.– Vol.1.– P.217-227.

[175]. Duda, R.O., Hart, P.E. Pattern Classification and Scene Analysis // John Wiley, New York.– 1973.– 512 p.

[176]. Hosmer(Jr.), D.W. A comparision of iterative maximum-likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample // Biometrics.– 1973.– Vol.29.– P.761-770.

[177]. Peters(Jr.) B.C., Walker, H.F. An iterative procedure for obtaining maximum-likelihood estimates of the parameters for a mixture of normal distributions // SIAM J. Appl. Math.– 1978.– Vol.35.– P.362-378.

[178]. Peters(Jr.) B.C., Coberly, W.A. The numerical evaluation of the maximum-likelihood estimate of mixture proportions // Comm. Statist. Theor. Meth.– 1976.– Vol.5, Iss.12.– P.1127-1135.

[179]. Peters(Jr.) B.C., Walker, H.F. The Numerical evaluation of the maximum-likelihood estimate of a subset of mixture proportions // SIAM J. Appl. Math.– 1978.– Vol.35.– P.447-452.

[180]. Baum, L.E., Petrie, T., Soules, G., Weiss, N. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains // Ann. Math. Statist.– 1970.– Vol.41.– P.164-171.

[181]. Sundberg, R. An iterative method for solution of the likelihood equations for incomplete data from exponentialfamilies // Comm. Statist. Simulation Comput.– 1976.– Vol.5, Iss.1.– P.55-64.

[182]. Haberman, S.J. Log-linear models for frequency tables derived by indirect observations: Maximum-likelihood equations // Ann. Statist.– 1974.– Vol.2.– P.911-924.

[183]. Haberman, S.J. Iterative scaling procedures for log-linear models for frequency tables derived by indirect observation // Proc. Amer. Statist. Assoc. (Statist. Comp. Sect.).– 1976.– P.45-50.

[184]. Haberman, S.J. Product models for frequency tables involving indirect observation // Ann. Statist.– 1977.– Vol.5.– P.1124-1147.

[185]. Ceppelini, R., Siniscalco, S., Smith, C.A.B. The estimation of gene frequencies in a random- mating population // Ann. Human Genetics.– 1955.– Vol.20.– P.97-115.

[186]. Chen, T. Mixed-up frequencies in contingency tables // Ph.D. dissertation, Univ. of Chicago, Chicago.– 1972.– 392 p.

[187]. Goodman, L.A. The analysis of systems of qualitative variables when some of the variables are unobservable: Part I – A modified latent structure approach // Amer. J. Sociol.– 1974.– Vol.79.– P.1179-1259.

[188]. Orchard, T., Woodbury, M.A. A missing information principle: theory and applications // Proc. of the 6th Berkeley Symposium on Mathematical Statistics and Probability.– 1972.– vol. 1.– P.697-715

[189]. Redner, R.A. An iterative procedure for obtaining maximum likelihood estimates in a mixture model // Rep. SR-T1-04081, NASA Contract NAS9-14689, Texas A&M Univ., College Station, TX.– 1980.– 224 p.

[190]. Vardi, Y. Nonparametric estimation in renewal processes // Ann. Statist.– 1982.– Vol.10.– P.772-785.

[191]. Boyles, R.A. On the convergence of the EM algorithm // J. Royal Statist.– 1983.– Vol.45, No.1.– P.47-50.

[192]. Wu, C.–F. On the convergence of the EM algorithm // Ann. Statist.– 1983.– Vol.11, No.1.– P.95-103.

[193]. Broniatowski, M., Celeux, G., Diebolt, J. Reconnaissance de mélanges de densités par un algorithme d’apprentisage probabiliste // E. Diday, M. Jambu, L. Lebart, J.–P. Pag`es and R. Tomasone (Eds.) Data Analysis and Informatics, III, North Holland, Amsterdam.– 1983.– P.359-373.

[194]. Celeux, G., Diebolt, J. Reconnaissance de mélanges de densité et classification. Un algorithme d’apprentisage probabiliste: l’algorithme SEM // Rapport de Recherche de l’INRIA RR-0349. Centre de Rocquencourt.– 1984.– 75 p.

[195]. Celeux, G., Diebolt, J. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem // Computational Statistics Quarterly.– 1985.– Vol. 2, No. 1.– P.73-82.

[196]. Ip, E.H. A Stochastic EM Estimator in the Presence of Missing Data – Theory and Practice // PhD Dissertation, Stanford University.– 1994.– 127 p.

[197]. Diebolt, J., Ip E. H. S. Stochastic EM: method and application / W. R. Gilks, S. Richardson, D. J. Spiegelhalter (Eds.) // Markov Chain Monte Carlo in Practice.– Chapman and Hall, London.– 1996.– P.259-273.

[198]. Celeux, G., Govaert, G. A Classification EM Algorithm for Clustering and Two Stochastic Versions // Rapport de Recherche de l'INRIA RR-1364. Centre de Rocquencourt.– 1991.– 19 p.

[199]. Celeux, G., Govaert, G. A classification EM algorithm for clustering and two stochastic versions // Computational Statistics and Data Analysis.– 1992.– Vol. 14, Iss.3.– P.315-332.

[200]. Mishra, N., Oblinger, D., Pitt, L. Sublinear time approximate clustering // SODA '01 Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms.– 2001.– P. 439-447.

[201]. Sun, Zh. et al. A parallel clustering method combined information bottleneck theory and centroid-based clustering // The Journal of Supercomputing.– 2014.– Volume 69(1).– P.452-467.

[202]. Sheng, W. A., Liu, X. Genetic k-medoids clustering algorithm // Journal of Heuristics.– 2004.– Volume 12(6).– P.447-466, doi:10.1007/s10732-006-7284-z.

[203]. Pelleg, D., Moore, A. X-means: Extending k-means with efficient estimation of the number of clusters // ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning.– 2000.– P.727-734.

[204]. Kazakovtsev, L. A., Antamoshkin, A. N, Fedosov V.V. Greedy heuristic algorithm for solving series of EEE components classification problems // IOP Conference Series: Materials Science and Engineering.– 2016.– Vol. 122.– article ID 012011, DOI: 10.1088/1757-899X/122/1/012011.

[205]. Kazakovtsev, L.A., Gudyma, M.N., Antamoshkin, A.N. Genetic Algorithm with Greedy Heuristic for Capacity Planning // 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) – S.-Petersburg, 2014, 6-8 October .–2014.– P. 607-613.

[206]. Antamoshkin, A., Masich, I. Pseudo-Boolean Optimization in Case of an Unconnected Feasible Set / "Models and Algorithms for Global Optimization" // Optimization and Its Applications.– 2007.– Vol.4.– P.111-122.

[207]. Hakimi, S. L. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph// Operations Research. 1964. Issue 12(3), P.450-459.

[208]. Antamoshkin, A. N., Kazakovtsev, L.A. Random Search Algorithm for the p-Median Problem // Informatica.– 2013.– Vol.37(3).– P.267-278.

[209]. Mathias, K. E., Whitley, D. Initial performance comparisons for the Delta Coding algorithm // International Conference on Evolutionary Computation.– 1994.– P.433-438.

[210]. Mayer, H. A. ptGAs--genetic algorithms evolving noncoding segments by means of promoter/terminator sequences // Evolutionary Computation.– 1998.– Vol. 6(4).– P.361-386.

[211]. Gad, A. H. Space trajectories optimization using variable-chromosome-length genetic algorithms // PhD Dissertation, Michigan Technological University.– 2011.– 124 p.

[212]. Гудыма, М.Н., Орлов, В.И., Казаковцев, Л.А., Антамошкин, А.Н. Стабильность решений серийного алгоритма с жадной эвристикой для задачи автоматической группировки // Системы управления и информационные технологии.– 2016.– Т. 66.– № 4.1.– С.141-145.

[213]. Jaccard, P. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines // Bull. Soc. Vaudoise sci. Natur.– 1901.– V. 37. Bd. 140.– S.241-272.

[214]. Казаковцев, Л.А., Ступина, А.А., Орлов, В.И. Выбор метрики для системы автоматической классификации электрорадиоизделий по производственным партиям // Программные продукты и системы.– 2015.– № 2 (110) .– С.124-129.

[215]. Шенк, Х. Теория инженерного эксперимента. Пер. с англ. Е. Г. Коваленко под ред. Н. П. Бусленко // М.: Мир.– 1972.– 382 с.

[216]. Федосов, В.В., Казаковцев, Л.А., Гудыма, М.Н. Задача нормировки

исходных данных испытаний электрорадиоизделий космического применения для алгоритма автоматической группировки // Информационные технологии моделирования и управления.– 2016.– Т. 100.– № 4.– С.263-268.

[217]. Строгонов, А.В., Шацких, Д., Горлов, М.И. Технологические тренировки интегральных схем // Компоненты и технологии.– 2009.– №93.– С.196-199.

[218]. Сташков, Д.В., Гудыма, М.Н. Генетические алгоритмы метода жадных эвристик для серии задач разделения смеси распределений // Информационные технологии моделирования и управления.– 2017.– Т. 105, №3.– С.181-191.

[219]. Рубан, А.И. Методы анализа данных // Красноярск: ИПЦ КГТУ.– 2004.– 319 с.

[220]. Azizyan, M., Singh, A., Wasserman, L. A. Minimax theory for high-dimensional gaussian mixtures with sparse mean separation // NIPS.– 2013.– P.2139-2147.

[221]. Vempala, S., Wang, G. A spectral algorithm for learning mixtures of distributions // FOCS.–2002.– P.841-860.

[222]. Achlioptas, D., McSherry, F. On spectral learning of mixtures of distributions // COLT.– 2005.– P.458-469.

[223]. Казаковцев, Л.А., Антамошкин, А.Н. Метод жадных эвристик для задач размещения // Вестник СибГАУ.– 2015.– №2.– С.317-325.

[224]. Казаковцев, Л.А. Эволюционный алгоритм для задачи k-медоид // Системы управления и информационные технологии.– 2015.– №2(60).– С.36-40.

[225]. Казаковцев, Л.А., Ступина, А.А., Орлов, В.И. Модификация генетического алгоритма с жадной эвристикой для непрерывных задач размещения и классификации // Системы управления и информационные технологии.– 2014.– №.2(56).– С.35-39.

[226]. Казаковцев, Л.А., Масич, И.С., Орлов, В.И., Федосов, В.В. Быстрый детерминированный алгоритм для классификации электронной компонентной базы по критерию равнонадежности // Системы управления и информационные технологии.– 2015.– Т. 62, № 4.– С. 39-44.

[227]. Семенкин, Е.С. Коэволюционный генетический алгоритм решения сложных задач условной оптимизации / Е.С. Семенкин, Р.Б. Сергиенко // Вестник СибГАУ.– 2009.– №2.– С.17-21.

[228]. Семенкин, Е. С., Лебедев В. А. Метод обобщенного адаптивного поиска для синтеза систем управления сложными объектами // М.: МАКС-Пресс.– 2002.– 320 с.

[229]. Казаковцев, Л.А. Детерминированный алгоритм для задачи k-средних и k-медоид // Системы управления и информационные технологии.– 2015.– Т. 59, № 1.– С.95-99.

[230]. Kazakovtsev, L.A. Fast Deterministic Algorithm for EEE Components Classification / L.A.Kazakovtsev, A.N.Antamoshkin, I.S.Masich // IOP Conf. Series: Materials Science and Engineering.– 2015.– Vol.94.– article ID 012015.– 10 p., DOI: 10.1088/1757-899X/04/1012015.

[231]. Федосов, В.В., Казаковцев, Л.А., Масич, И.С. Метод нормировки исходных данных испытаний электрорадиоизделий космического применения для алгоритма автоматической группировки // Системы управления и информационные технологии.– 2016.– Т. 65, № 3.– С.92-96.

[232]. Orlov, V.I, Stashkov, D.V., Kazakovtsev, L.A., Stupina, A.A. Fuzzy clustering of EEE components for space industry // IOP Conference Series: Materials Science and Engineering.– 2016.– Vol. 155, No.1.– P.163-169.

[233]. Федосов В.В. Вопросы обеспечения работоспособности электронной компонентной базы в аппаратуре космических аппаратов: учеб.пособие // Сиб. гос. аэрокосмич. ун-т, Красноярск.– 2015.– 68 С.

[234]. Орлов, А.И. Теория принятия решений / А.И. Орлов.– М.: Издательство «Экзамен».– 2005.– 656 С.

[235]. Drezner, Z., Klamroth, K., Schobel, A., Wesolowsky, G. The Weber problem / Z. Drezner, H. Hamacher (Eds.), Facility Location: Applications and Theory // Springer-Verlag Berlin Heidelberg.– 2002.– P. 1-36.

[236]. Казаковцев, Л.А. Алгоритм для задачи размещения с неевклидовой метрикой, основанной на угловом расстоянии // Фундаментальные исследования.–

2012.– № 9-4.– С. 918-923.

[237]. Kazakovtsev, L.A., Stanimirović, P.S., Osinuga, I.A., Gudyma, M.N., Antamoshkin, A.N. Algorithms for location problems based on angular distances // Advances in Operations Research.– 2014.– vol. 2014.– Article ID 701267, doi:10.1155/2014/701267

[238]. Drezner, Z., Scott, C., Song, J. The central warehouse location problem revisited // IMA Journal of Managemeng Mathematics.– 2003.– Vol.14(4).– P.321-336.

[239]. Vygen, J. Approximation algorithms for facility location problems (lecture notes) // Technical report no. 05950, Research Institute for Discrete Mathematics.– 2005.– 59 p.

[240]. Szegedy, C. Some applications of the combinatorial Laplacian // Ph.D. thesis, University of Bonn.– 2005.– 146 p.

[241]. Забудский, Г.Г., Амзин, И.В. Сужение области поиска решения задачи Вебера на плоскости с прямоугольными запрещенными зонами // Автоматика и телемеханика.– 2012.– №5.– С.71-83.

[242]. Idrissi, H., Lefebvre, O., Michelot, C. A primal-dual algorithm for a constrained Fermat-Weber problem involving mixed norms // RAIRO – Operations Research – Recherche Operationnelle.– 1988.– Vol.22.– P.313-330.

[243]. Hansen, P., Peeters, D., Thisse, J. Constrained location and the Weber-Rawls problem // North-Holland Mathematics Studies.– 1981.– Vol.59.– P.147-166.

[244]. Hansen, P., Peeters, D., Thisse, J. An algorithm for a constrained Weber problem // Management Science.– 1982.– Vol.28, Iss.11.– P.1285-1295.

[245]. Hansen, P., Mladenović, N., Taillard, E. Heuristic solution of the multisource Weber problem as a p-median problem // Operations Research Letters.– 1998.– Vol.22, Iss. 2-3.– P. 55-62.

[246]. Gonzalez-Martin, S., Ferrer, A., Juan, A.A., Riera, D. Solving non-smooth arc routing problems throughout biased- randomized heuristics / J.F. de Sousa, R.Rossi (eds), Computer-based Modelling and Optimization in Transportation // Springer International Publishing.– 2014.– pp. 451-462.

[247]. Luis, M., Salhi, S., Nagy, G. Region-rejection based heuristics for the

capacitated multi-source Weber problem // Computers & Operations Research.– 2009.– Vol.36, Iss.6.– P.2007-2017.

[248]. Jianga, J.–L., Yuan, X.–M. A heuristic algorithm for constrained multi-source Weber problem the variational inequality approach // European Journal of Operational Research.– 2008.– Vol.187 (2).– P.357-370.

[249]. Kazakovtsev, L.A. Algorithm for a constrained Weber problem with feasible region bounded by arcs // Facta Universitatis, Ser. Math. Inform.– 2013.– Vol.28(3).– P.271-284.

[250]. Akyiiz, M.H., Oncan, T., Altnel, I.K. Beam search heuristics for the single and multi-commodity capacitated multi-facility Weber problems // Computers & Operations Research.– 2013.– Vol.40, Iss.12.– P.3056-3068.

[251]. Baykasoglu, A., Özbakı, L., Tapkan, P. Artificial bee colony algorithm and its application to generalized assignment problem / Chan F.T.S., Tiwari M.K. (eds) Swarm Intelligence, Focus on Ant and Particle Swarm Optimization // I-Tech Education and Publishing, Vienna, Austria.– 2007.– P.114-129.

[252]. Bischoff, M., Klamroth, K. An efficient solution method for Weber problems with barriers based on genetic algorithms // European Journal of Operational Research.– 2007.– Vol.177, Iss.1.– P.22-41.

[253]. Mohammadi, N., Malek, M.R., Alesheikh, A.A. A new GA-based solution for capacitated multi-source Weber problem // International Journal of Computational Intelligence Systems.– 2010.– Vol.3, Iss.5.– P.514-521.

[254]. Ghaderi, A., Jabalameli, M.S., Barzinpour, F., Rahmaniani, R. An efficient hybrid particle swarm optimization algorithm for solving the uncapacitated continuous location-allocation problem // Networks and Spatial Economics.– 2012.– Vol.12(3).– P.421-439.

[255]. Fallah-Jamshidi, S., Amiri, M., Karimi, N. Nonlinear continuous multi-response problems: a novel two-phase hybrid genetic based metaheuristic // Applied Soft Computing.– 2010.– Vol.10, Iss.4.– P.1274-1283.

[256]. Gharravi, H.G., Farham, M.S. Applying metaheuristic approaches on the single facility location problem with polygonal barriers // International Journal of

Metaheuristics.– 2014.– Vol.3, Iss.4.– P.348-370.

[257]. Yang, X.–S. Nature-Inspired Metaheuristic Algorithms // Luniver Press.– 2008.– 116 p.

[258]. Akhmedova, Sh.A. SVM-based classifier ensembles design with co-operative biology inspired algorithm // Вестник СибГАУ.– 2015.– № 1 (16).– С.22-27.

[259]. Fister(Jr.), I., Yang, X.–S., Fister, I., Brest, J., Fister, D. A brief review of nature-inspired algorithms for optimization // Elektrotehniski Vestnik.– 2013.– Vol.80, Iss.3.– P.1-7.

[260]. Dorigo, M., Maniezzo, V., Colorni, A. Ant system: Optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.–1996.– Vol.26, Iss.1.– P.29-41.

[261]. Kennedy, J., Eberhart, R.C. Particle swarm optimization // Proceedings of the 1995 IEEE International Conference on Neural Networks.– 1995.– Vol.6.– P.1942-1948.

[262]. Karaboga, D. An idea based on honey bee swarm for numerical optimization // Technical report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.– 2005.– 10 p.

[263]. Yang, X.S. Firefly algorithms for multimodal optimization // Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Springer.– 2009.– P. 169-178.

[264]. Yang, X.S., Deb, S. Cuckoo search via Levy flights // Proc. of World Congress on Nature & Biologically Inspired Computing.– 2009.– P.210-214.

[265]. Afshar, A., Massoumi, F., Afshar, A., Marino, M.A. State of the art review of ant colony optimization applications in water resource management // Water Resources Management.– 2015.– Vol.29,Iss.11.– P.3891-3904.

[266]. Mohan, B.C., Baskaran, R. A survey: Ant colony optimization based recent research and implementation on several engineering domain // Expert Systems with Applications.– 2012.– Vol.39, Iss.4.– P.4618-4627.

[267]. Neto, R.T., Filho, M.G. Literature review regarding ant colony optimization applied to scheduling problems: Guidelines for implementation and

directions for future research // Engineering Applications of Artificial Intelligence.– 2013.– Vol.26, Iss.1.– P.150-161.

[268]. Banks, A., Vincent, J., Anyakoha, C. A review of particle swarm optimization. Part I: background and development // Natural Computing.– 2007.– Vol.6, Iss.4.– P.467-484.

[269]. Banks, A., Vincent, J., Anyakoha, C. A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications // Natural Computing.– 2008.– Vol.7, Iss.1.– P.109-124.

[270]. Khare, A., Rangnekar, S. A review of particle swarm optimization and its applications in solar photovoltaic system // Applied Soft Computing.– 2013.– Vol.13, Iss.5.– P.2997-3006.

[271]. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications // Artificial Intelligence Review.– 2014.– Vol.42, Iss.1.– P.21-57.

[272]. Fister, I., Fister(Jr.), I., Yang, X.–S., Brest, J. A comprehensive review of firefly algorithms // Swarm and Evolutionary Computation.– 2013.– Vol.13.– P.34-46.

[273]. Fister(Jr.) I., Perc, M., Kamal, M., Fister, I. A review of chaos-based firefly algorithms: Perspectives and research challenges // Applied Mathematics and Computation.– 2015.– Vol.252.– P.155-165.

[274]. Fister(Jr.), I., Yang, X.–S., Fister, D., Fister I. Cuckoo Search: A Brief Literature Review, / Yang, X.–S. (ed.) Cuckoo Search and Firefly Algorithm: Theory and Applications // Springer International Publishing.– 2014.– P.49-62.

[275]. Гудыма, М.Н., Казаковцев, Л.А., Антамошкин, А.Н. Алгоритмы для задачи Вебера с допустимыми зонами, ограниченными окружностями // Системы управления и информационные технологии.– №1(67).– 2017.– С.4-9.

[276]. Karaboga, D., Akay, B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems // Applied Soft Computing Journal.– 2011.– Vol.11, Iss.3.– P.3021-3031.

[277]. Brajevic, I. Crossover-based artificial bee colony algorithm for constrained

optimization problems // Neural Computing and Applications.– 2015.– Vol.26, Iss.7.– P.1587-1601.

[278]. Gandomi, A.H., Yang, X.–S., Alavi, A. H. Mixed variable structural optimization using Firefly Algorithm // Computers & Structures.– 2011.– Vol.89, Iss.23-24.– P.2325-2336.

[279]. Brajevic, I., Ignjatovic, J. An enhanced firefly algorithm for mixed variable structural optimization problems // Facta Universitatis, Ser. Math. Inform.– 2015.– Vol.30, Iss.4.– P.401-417.

[280]. Kazakovtsev, L. A., Stanimirovic, P. S. Algorithm for Weber problem with a metric based on the initial fare // Journal of Applied Mathematics and Informatics.– 2015.– Vol.33, Iss.1-2.– P.157-172.

[281]. Karaboga, D., Basturk, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems // LNAI 4529: IFSA'07, Springer-Verlag.– 2007.– P.789-798.

[282]. Deb, K. An efficient constraint-handling method for genetic algorithms // Computer Methods in Applied Mechanics and Engineering.– 2000.– Vol.186, Iss.2-4.– P.311-338.

[283]. E. Mezura-Montes, O. Cetina-Dominguez, Empirical analysis of a modified artificial bee colony for constrained numerical optimization, Applied Mathematics and Computation 218 (22) (2012) 10943-10973.

[284]. Li, X., Yin, M. Self-adaptive constrained artificial bee colony for constrained numerical optimization // Neural Computing and Applications.– 2014.– Vol.24, Iss.3-4.– P.723-734.

[285]. Liang, Y., Wan, Z., Fang, D. An improved artificial bee colony algorithm for solving constrained optimization problems // International Journal of Machine Learning and Cybernetics.– 2017.– Vol.8, Iss.3.– P.739-754.

[286]. Sharma, T.K., Pant, M. Shuffled artificial bee colony algorithm // Soft Computing.–2016.– P.1-20.

[287]. Kukkonen, S., Lampinen, J. Constrained real-parameter optimization with generalized differential evolution // IEEE Congress on Evolutionary Computation 2006

(CEC 2006).– 2006.– P.207-214.

[288]. Mezura-Montes, E., Coello, C.A.C. Constraint-handling in nature-inspired numerical optimization: Past, present and future // Swarm and Evolutionary Computation.– 2011.– Vol.1, Iss.4.– P.173-194.

[289]. Cerpinsek, M., Liu, S.–H., Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey // ACM Computing Surveys.– 2013.– Vol.45, Iss.3.– P.1-33.

ПРИЛОЖЕНИЕ А. Сравнение работы различных алгоритмов для задачи автоматической группировки электрорадиоизделий

Мы применили разработанные алгоритмы со статическим размером популяции при различном количестве особей в популяции, а также новый алгоритм с динамической популяцией.

В Таблицах А.1, А.4, А.7, А.10, А.13, А.16, А.19, А.22, А.25, А.28, А.31, А.34 даны результаты работы алгоритмов в версии с полным объединенным решением (см. [222])

В Таблицах А.2, А.5, А.8, А.11, А.14, А.17, А.20, А.23, А.26, А.29, А.32, А.35 даны результаты работы алгоритмов в модификации с частичным объединенным решением (см. модифицированный алгоритм в [143]). Преимущества модификаций алгоритмов, приведенных в Таблицах А.2, А.5, А.8, А.11, А.14, А.17, А.20, А.23, А.26, А.29, А.32, А.35 над алгоритмами в Таблицах А.1, А.4, А.7, А.10, А.13, А.16, А.19, А.22, А.25, А.28, А.31, А.34 экспериментально доказаны лишь для отдельных задач. В таблицах А.2, А.5, А.8, А.11, А.14, А.17, А.20, А.23, А.26, А.29, А.32, А.35 даны результаты работы алгоритмов в модификации с частичным объединенным решением (см. модифицированный алгоритм в [143]. Тем не менее, преимущества новых алгоритмов с динамическими популяциями статистически доказаны, доказательства также приведены в таблицах.

Для доказательства мы использовали гипотезу о равенстве (неравенстве) оценок математических ожиданий (усредненных значений) целевой функции, полученных в результате работы алгоритмов. Обозначим через $\overline{m_1}$ и $\overline{m_2}$ вычисленные оценки математических ожиданий значений целевой функции, полученных разными алгоритмами при n запусках каждого из них, $x_1,\ldots x_n$ и $y_1,\ldots,y_n$ – значения целевой функции, полученные в каждом из запусков. Для проверки гипотезы $H:m_1=m_2$ вводим t-статистику [216] (критерий Стьюдента):

$$z = \frac{\overline{m_1} - \overline{m_2}}{\overline{\sigma}\sqrt{1/n_1 + 1/n_2}},$$

где

$$\overline{\sigma^2} = \frac{1}{n_1 + n_2 - 2}\left(\sum_{i=1}^{n} (x_i - \overline{m_1})^2 + \sum_{j=1}^{n} (y_j - \overline{m_2})^2 \right)$$

Для проверки гипотезы проверяем значения неравенства $|z| < t_{\text{альфа},2n-2}$, где $t_{\text{альфа},2n-2}$ – пороговое значение статистики Стьюдента. Из таблиц видно, что уровне значимости 1% для большинства задач различия математических ожиданий результата работы нового алгоритма с динамической популяции и наилучшего из вариантов алгоритма со статической популяцией статистически не значимы, т.е. «динамический» алгоритм работает не хуже, чем лучший из алгоритмов со статической популяцией. При этом число особей в алгоритмах, дающих лучшие результаты, в различных задачах сильно отличается, а различия в результатах алгоритмов с разным числом особей очевидны.

Для более развернутого подтверждения эффективности новых алгоритмов было решено осуществить полный перебор вариантов для некоторых задач. В таблицах А.28–А.36 приведены результаты обработки задач небольшой размерности, для которых полный перебор может быть осуществлен в течение приемлемого времени. В таблицах А.30, А.33 и А.36 для сравнения приведены результаты полного перебора.

Таблица А.1 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 2D522B_p5, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=15
Новый ГА серийн.	динамич.	среднее	14623,42232	11778,04243	9453,948833
		σ	1,87497E-12	0,131398416	1,958581264
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	14623,42232	11778,05911	9454,116983
		σ	1,87497E-12	0,134981965	3,355291928
	10	среднее	14623,42232	11778,02216	9454,220623
		σ	1,87497E-12	0,137330458	3,803505395
	20	среднее	14623,42232	11777,97862	9452,696734
		σ	1,87497E-12	0,117532026	3,395554166
	50	среднее	14623,42232	11778,0039	9454,599895
		σ	1,87497E-12	0,119906448	3,037425402
	100	среднее	14623,42232	11791,46149	9484,686237
		σ	1,87497E-12	17,30474663	16,86911638
	Лучший рез-т	среднее	14623,42232	11777,97862	9452,696734
			0,0000000%	-0,0005418%	-0,0132442%
		σ	1,87497E-12	0,117532026	3,395554166
			0,0000000%	-10,5529353%	73,3680511%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	-1,982637872	-1,749529555
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	14621,71883	11777,89821	9456,833545
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,03327E-06	0,0230770044	4,817093454
			200,00000%	-82,43738%	130,14664%
ГА Шена и Лиу		среднее	14621,71883	11775,55952	9454,981414
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,03327E-06	2,559214518	4,817093454
			55108488,9%	1847,67532%	145,94810%

Таблица А.2 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 2D522B_p5, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=15
Новый ГА серийн.	динамич.	среднее	14623,42232	11778,04243	9453,948833
		σ	1,87497E-12	0,131398416	1,958581264
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	14623,42232	11777,99527	9449,589523
		σ	1,87497E-12	0,127570141	3,551942983
	10	среднее	14623,42232	11778,01171	9449,483721
		σ	1,87497E-12	0,134235737	3,870172935
	20	среднее	14623,42232	11777,95233	9450,233667
		σ	1,87497E-12	0,168973669	3,379196556
	50	среднее	14623,42232	11777,93197	9449,976983
		σ	1,87497E-12	0,080704273	3,420332104
	100	среднее	14623,42232	11789,00802	9507,428186
		σ	1,87497E-12	13,68025463	48,44048223
	Лучший рез-т	среднее	14623,42232	11777,93197	9449,483721
			0,0000000%	-0,0001525%	-0,0144833%
		σ	1,87497E-12	0,080704273	3,870172935
			0,0000000%	-20,8145388%	60,5074856%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	-0,756635534	-1,644178373
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	14621,71883	11777,89821	9456,833545
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,03327E-06	0,0230770044	4,817093454
			200,000000%	-82,4373801%	130,146647%
ГА Шена и Лиу		среднее	14621,71883	11775,55952	9454,981414
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,03327E-06	2,559214518	4,817093454
			55108488,9%	1847,67532%	145,94810%

Таблица А.3 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 2D522B_p5, l_1, p-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_1		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=5	p=10	p=15
ГА серийн. с динамич. популяцией	среднее	14623,42232	11778,04243	9453,948833
	σ	1,87497E-12	0,131398416	1,958581264
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	14622,17142	11777,17512	9455,615251
		-0,0085540%	-0,0073638%	0,0176267%
	σ	0,673604395	1,319108446	3,820346886
		-	903,8998101%	95,0568484%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	14621,81975	11776,48489	9456,174602
		-0,0109589%	-0,0132241%	0,0235433%
	σ	0,41297911	1,879166169	2,6217001
		-	1330,12848%	33,8571010%
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-30,05826447	-6,404569502	5,268344472
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. значимо	да, преимущ. стат. значимо	нет, преимущ. не наблюдается
ALA	среднее	14621,71883	11777,89821	9456,833545
		-0,0116490%	-0,0210809%	0,0109219%
	σ	1,03327E-06	0,0230770044	4,817093454
		200,000000%	-82,4373801%	130,1466473%
ГА Шена и Лиу	среднее	14621,71883	11775,55952	9454,981414
		-0,0116490%	-0,0210809%	0,0109219%
	σ	1,03327E-06	2,559214518	4,817093454
		55108488,9%	1847,6753223%	145,9481025%

Таблица А.4 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 140УД25АСВК, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=20
Новый ГА серийн.	динамич.	среднее	1231,122314	974,7686159	659,8418289
		σ	0	0,283038419	0,243811938
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	1232,897866	980,6757122	669,2356784
		σ	4,295130824	4,757895618	7,437042504
	10	среднее	1231,122314	976,6443671	665,4255408
		σ	0	2,641981642	3,230847889
	20	среднее	1231,122314	975,2956158	662,6421798
		σ	0	1,419911637	1,517056162
	50	среднее	1231,122314	974,8109857	660,1708841
		σ	0	0,239211124	1,016516349
	100	среднее	1238,471624	984,9896083	682,1459574
		σ	8,561008611	6,496167884	16,8072303
	Лучший рез-т	среднее	1231,122314	974,8109857	660,1708841
			0,0000000%	0,0043467%	0,0498688%
		σ	0	0,239211124	1,016516349
			-	-15,4845745%	316,92640%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	0,626224291	1,724126532
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	1231,122314	974,7686159	667,6815854
			0,0000000%	0,0000000%	1,1881266%
		σ	0	0,283038419	2,1782161
			-	0,0000000%	793,400099%
ГА Шена и Лиу		среднее	1231,122315	977,1142667	663,3298912
			0,0000002%	0,2401745%	0,5241852%
		σ	1,43344E-06	5,461095122	0,763327235
			-	1829,4536573%	213,080335%

Таблица А.5 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 140УД25АСВК, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=20
Новый ГА серийн.	динамич.	среднее	1231,122314	974,7262461	659,577814
		σ	0	0,314936079	0,257323457
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	1232,931209	979,3739936	669,6112213
		σ	4,039369166	5,030576955	6,707968646
	10	среднее	1231,122314	976,7710974	666,6601732
		σ	0	3,546663134	5,346565495
	20	среднее	1231,122314	976,159598	662,2667519
		σ	0	2,453326594	1,910638986
	50	среднее	1231,122314	974,7686159	659,9177506
		σ	0	0,283038419	0,41641617
	100	среднее	1238,787504	989,3819904	700,8326212
		σ	6,191390966	8,012566216	19,18010234
	Лучший рез-т	среднее	1231,122314	974,7686159	659,9177506
			0,0000000%	0,0043468%	0,0515385%
		σ	0	0,283038419	0,41641617
			-	-10,1282966%	61,8259658%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	0,548065205	3,803637227
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	нет
ALA		среднее	1231,122314	974,7686159	667,6815854
			0,0000000%	0,0000000%	1,1881266%
		σ	0	0,283038419	2,1782161
			-	0,0000000%	793,400099%
ГА Шена и Лиу		среднее	1231,122315	977,1142667	663,3298912
			0,0000002%	0,2401745%	0,5241852%
		σ	1,43344E-06	5,461095122	0,763327235
			-	1829,4536573%	213,080335%

Таблица А.6 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 140УД25АСВК, p-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_1		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=5	p=10	p=20
ГА серийн. с динамич. популяцией	среднее	1231,122314	974,7686159	659,8418289
	σ	0	0,283038419	0,243811938
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	1231,122314	974,1754389	669,3605906
		0,0000000%	-0,0608531%	1,4425823%
	σ	0	2,34371E-13	2,376438072
		-	-100,0000%	874,7012749%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	1231,122314	974,5143972	661,2303079
		0,0000000%	-0,0260799%	0,2104260%
	σ	0	0,37058428	1,081299692
		-	30,9307341%	343,4974345%
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-	-4,222892058	9,702871682
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	рез-ты равны	да, преимущ. стат. значимо	нет, преимущ. не наблюдается
ALA	среднее	1231,122314	974,7686159	667,6815854
		0,0000000%	0,0000000%	1,1881266%
	σ	0	0,283038419	2,1782161
		-	0,0000000%	793,4000992%
ГА Шена и Лиу	среднее	1231,122315	977,1142667	663,3298912
		0,0000002%	0,2401745%	0,5241852%
	σ	1,43344E-06	5,461095122	0,763327235
		-	1829,45365%	213,080335%

Таблица А.7 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний 140УД25АСВК (с данными по дрейфу), p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	Метрика такси
Время работы алгоритмов	t=20 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=5
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	17,68436387	10,85750382	8,053434997
		σ	0	1,2882E-08	8,82506E-09
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	17,68436387	10,85750383	8,100383163
		σ	0	1,91868E-15	0,124213129
	10	среднее	17,68436387	10,85750383	8,053435007
		σ	0	1,91868E-15	7,41576E-09
	20	среднее	17,68436387	10,85750383	8,053435009
		σ	0	9,10894E-09	9,88755E-09
	50	среднее	17,68436387	10,85750382	8,053435004
		σ	0	1,2882E-08	7,96438E-09
	100	среднее	17,68436387	11,48610437	9,33501439
		σ	0	1,503291887	2,018636784
	Лучший рез-т	среднее	17,68436387	10,85750382	8,053435004
			0,0000000%	0,0000000%	0,0000001%
		σ	0	1,2882E-08	7,96438E-09
			-	0,0000000%	-9,7526476%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	1,035098186	3,44380909
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	17,68436369	10,91141991	8,053434987
			-0,0000010%	0,4965791%	-0,0000001%
		σ	0	0,142648538	0
			-	1107348349%	-100,00000%
ГА Шена и Лиу		среднее	17,68436369	10,98330802	8,053434992
			-0,0000389%	-0,0931749%	-0,0907079%
		σ	0	0,125804186	1,89114E-08
			-67,52116%	-99,9772015%	9,3043876%

Таблица А.8 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 140УД25АСВК (с данными по дрейфу), p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	Метрика такси
Время работы алгоритмов	t=20 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=5
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=10	p=20	p=50
Новый ГА серийн.	динамич.	среднее	17,68436387	10,85750382	8,053435002
		σ	0	1,2882E-08	7,27046E-09
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	17,68436387	11,43218829	9,288066238
		σ	0	1,520472164	2,048100118
	10	среднее	17,68436387	10,85750383	8,10038316
		σ	0	9,10894E-09	0,12421313
	100	среднее	17,68436387	10,96533599	8,888730703
		σ	0	0,184158471	0,615274122
	Лучший рез-т	среднее	17,68436387	10,85750383	8,10038316
			0,0000000%	0,0000001%	0,5829582%
		σ	0	9,10894E-09	0,12421313
			-	-29,2893219%	170846415%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	3,380616521	2,927700728
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	17,68436369	10,91141991	8,053434987
			-0,0000010%	0,4965791%	-0,0000001%
		σ	0	0,142648538	0
			-	1107348349%	-100,00000%
ГА Шена и Лиу		среднее	17,68436369	10,98330802	8,053434992
			-0,0000389%	-0,0931749%	-0,0907079%
		σ	0	0,125804186	1,89114E-08
			-67,52116%	-99,9772015%	9,3043876%

Таблица А.9 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий (диодов) 140УД25АСВК (с данными по дрейфу), p-медиан, без нормирования		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		Метрика такси		
Время работы алгоритмов		t=20 сек		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=5		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=10	p=20	p=50
ГА серийн. с динамич. популяцией	среднее	17,68436387	10,85750382	8,053434997
	σ	0	1,2882E-08	8,82506E-09
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	17,68436371	10,85750381	8,053435013
		-0,0000009%	0,0000000%	0,0000002%
	σ	1,06651E-08	1,45399E-08	0
		-	12,8698103%	-100,000000%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	17,68436375	10,85750381	8,053434997
		-0,0000007%	-0,0000001%	0,0000000%
	σ	8,01354E-08	9,10894E-09	1,17056E-08
		-	-29,2893219%	32,6399036%
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-	7,69277E-08	0,545809139
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	рез-ты равны	да, преимущ. стат. незначимо	нет, преимущ. не наблюдается
ALA	среднее	17,68436369	10,91141991	8,053434987
		-0,0000010%	0,4965791%	-0,0000001%
	σ	0	0,142648538	0
		-	1107348349%	-100,000000%
ГА Шена и Лиу	среднее	17,68436369	10,98330802	8,053434992
		-0,0000389%	-0,0931749%	-0,0907079%
	σ	0	0,125804186	1,89114E-08
		-67,52116%	-99,9772015%	9,3043876%

Таблица А.10 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 1526IE10 (с данными по дрейфу), p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	Метрика такси
Время работы алгоритмов	t=2 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=10
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=5	p=10
Новый ГА серийн.	динамич.	среднее	5971,322618	4677,09755	3442,648386
		σ	0	0	3,881828912
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	5971,322618	4677,09755	3443,196923
		σ	0	0	8,476019197
	10	среднее	5971,322618	4677,09755	3440,062494
		σ	0	0	1,849849104
	20	среднее	5971,322618	4677,143274	3443,09269
		σ	0	0,12097518	4,691046096
	50	среднее	5971,322618	4677,143274	3444,579295
		σ	0	0,12097518	6,325394548
	100	среднее	5972,04946	4687,823391	3462,764309
		σ	1,923043443	24,78940597	13,6082089
	Лучший рез-т	среднее	5971,322618	4677,09755	3440,062494
			0,0000000%	0,0000000%	-0,0751135%
		σ	0	0	1,849849104
			-	-	-52,345939%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	-	-3,293793132
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	нет
ALA		среднее	5971,322618	4677,09755	3454,093687
			0,0000000%	0,0000000%	0,3324563%
		σ	0	0	17,29694475
			-	-	345,58750%
ГА Шена и Лиу		среднее	5971,322618	4677,097548	3448,641505
			0,0000000%	0,0000000%	0,1742154%
		σ	0	1,70513E-06	5,360281756
			-	-	38,08650%

Таблица А.11 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 1526IE10 (с данными по дрейфу), p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	Метрика такси
Время работы алгоритмов	t=2 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=10
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=5	p=10
Новый ГА серийн.	динамич.	среднее	5971,322618	4677,09755	3439,579844
		σ	0	0	3,090399659
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	5971,322618	4677,09755	3442,090624
		σ	0	0	2,883107715
	10	среднее	5971,322618	4677,09755	3440,158582
		σ	0	0	1,569795617
	20	среднее	5971,322618	4677,09755	3444,945365
		σ	0	0	4,058877299
	50	среднее	5971,322618	4677,09755	3443,058082
		σ	0	0	3,558442779
	100	среднее	5971,322618	4677,905315	3511,096322
		σ	0	1,277871711	62,09544944
	Лучший рез-т	среднее	5971,322618	4677,09755	3440,158582
			0,0000000%	0,0000000%	0,0168258%
		σ	0	0	1,569795617
			-	-	49,204122%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-	-	0,914500681
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	5971,322618	4677,09755	3454,093687
			0,0000000%	0,0000000%	0,3324563%
		σ	0	0	17,29694475
			-	-	345,58750%
ГА Шена и Лиу		среднее	5971,322618	4677,097548	3448,641505
			0,0000000%	0,0000000%	0,1742154%
		σ	0	1,70513E-06	5,360281756
			-	-	38,08650%

Таблица А.12 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 1526IE10 (с данными по дрейфу), p-медиан, без нормирования		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		Метрика такси		
Время работы алгоритмов		t=2 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=10		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=3	p=5	p=10
ГА серийн. с динамич. популяцией	среднее	5971,322618	4677,09755	3442,648386
	σ	0	0	3,881828912
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	5971,322618	4677,188998	3445,921402
		0,0000000%	0,0019552%	0,0950726%
	σ	0	0,156178285	7,247920298
		-	-	86,7140583%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	5971,322618	4677,09755	3442,621774
		0,0000000%	0,0000000%	-0,0007730%
	σ	0	0	2,49270701
		-	-	-35,7852428%
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-	-	-0,04468368
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	рез-ты равны	рез-ты равны	нет, преимущ. не наблюдается
ALA	среднее	5971,322618	4677,09755	3454,093687
		0,0000000%	0,0000000%	0,3324563%
	σ	0	0	17,29694475
		-	-	345,5875089%
ГА Шена и Лиу	среднее	5971,322618	4677,097548	3448,641505
		0,0000000%	0,0000000%	0,1742154%
	σ	0	1,70513E-06	5,360281756
		-	-	38,0865019%

Таблица А.13 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 5503Р1, l_2^2, р-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2^2
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=10	p=15	p=20
Новый ГА серийн.	динамич.	среднее	43690,56546	42796,7003	37510,89378
		σ	2,8361902	10,77139382	8,592195824
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	43689,39858	42803,50444	37520,99859
		σ	4,998996763	12,93094671	28,92521153
	10	среднее	43692,37435	42800,1733	37517,96293
		σ	3,183222015	8,245792391	18,0850513
	20	среднее	43691,75423	42797,03781	37522,9813
		σ	1,840100266	8,27039459	22,64086663
	50	среднее	43689,69052	42803,81797	37517,1073
		σ	1,766533016	8,405516768	18,07858791
	100	среднее	43833,72317	43039,09892	37947,67676
		σ	180,376018	154,3701981	142,3718154
	Лучший рез-т	среднее	43689,39858	42797,03781	37517,1073
			-0,0026708%	0,0007886%	0,0165646%
		σ	4,998996763	8,27039459	18,07858791
			76,2574585%	-23,21890%	110,40707%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-1,572610475	0,136126351	1,700235858
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	43721,17202	42853,61001	38082,27052
			0,0700530%	0,1329769%	1,5232288%
		σ	10,89982485	19,67889931	83,7137333
			284,31219%	82,69594%	874,29964%
ГА Шена и Лиу		среднее	43701,45589	42800,99608	37786,40649
			0,0249252%	0,0100369%	0,7343488%
		σ	12,22001903	10,00090042	64,69889166
			330,86035%	-7,1531448%	652,99600%

Таблица А.14 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 5503Р1, l_2^2, р-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2^2
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=10	p=15	p=20
Новый ГА серийн.	динамич.	среднее	43690,07529	42808,33802	37457,08698
		σ	3,342760635	8,671969818	21,05653346
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	43691,18112	42810,65289	37441,35191
		σ	2,615918165	10,97779738	15,46428842
	10	среднее	43691,23874	42801,88462	37444,22992
		σ	3,879149275	6,797537961	12,53117923
	20	среднее	43693,45851	42804,66043	37460,74313
		σ	1,987839218	14,10532545	14,54814752
	50	среднее	43690,00546	42798,84023	37463,10579
		σ	4,000207443	10,68289272	31,83960321
	100	среднее	43757,30812	42960,43378	38237,94822
		σ	25,07805412	57,29036992	292,4897128
	Лучший рез-т	среднее	43690,00546	42798,84023	37441,35191
			-0,0001598%	-0,0221868%	-0,0420083%
		σ	4,000207443	10,68289272	15,46428842
			19,6677800%	23,1887673%	-26,5582417%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-0,103752385	-3,780738616	-3,298914403
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	нет	нет
ALA		среднее	43721,17202	42853,61001	38082,27052
			0,0700530%	0,1329769%	1,5232288%
		σ	10,89982485	19,67889931	83,7137333
			284,31219%	82,69594%	874,29964%
ГА Шена и Лиу		среднее	43701,45589	42800,99608	37786,40649
			0,0249252%	0,0100369%	0,7343488%
		σ	12,22001903	10,00090042	64,69889166
			330,86035%	-7,15314%	652,99600%

Таблица А.15 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 5503Р1, l_2^2, р-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_2^2		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=10	p=15	p=20
ГА серийн. с динамич. популяцией	среднее	43690,56546	42796,7003	37510,89378
	σ	2,8361902	10,77139382	8,592195824
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	43686,74504	42796,98473	37619,88565
		-0,0087443%	0,0006646%	0,2905606%
	σ	2,615479966	9,741168585	30,59284543
		-7,7819264%	-9,5644561%	256,0538663%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	43687,46822	42800,24444	37497,77774
		-0,0070890%	0,0082813%	-0,0349660%
	σ	3,61330287	6,148740051	10,56555064
		27,39987%	-42,91602%	22,96682%
Гипотеза о неравенстве мат. ожиданий Н:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-5,222883281	2,21343168	-7,460317812
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да,преимущ. стат. значимо	нет, преимущ. не наблюдается	да, преимущ. стат. значимо
ALA	среднее	43721,17202	42853,61001	38082,27052
		0,0700530%	0,1329769%	1,5232288%
	σ	10,89982485	19,67889931	83,7137333
		284,31219%	82,6959411%	874,2996437%
ГА Шена и Лиу	среднее	43701,45589	42800,99608	37786,40649
		0,0249252%	0,0100369%	0,7343488%
	σ	12,22001903	10,00090042	64,69889166
		330,86035%	-7,1531448%	652,99600%

Таблица А.16 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Europe, l_2, р-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2
Время работы алгоритмов	t=12 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=50
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=10	p=20	p=50
Новый ГА серийн.	динамич.	среднее	1099345405	830591331,9	554280750,1
		σ	186,8542612	471269,1629	174625,9313
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	1099345168	831038747,7	554129122,8
		σ	156,5310396	623737,5817	380981,1627
	10	среднее	1099345435	830564995,3	554342550,1
		σ	235,6520605	803873,3431	346193,4519
	20	среднее	1099345426	830431446,4	554346557,7
		σ	183,8731642	708606,5637	418310,7586
	50	среднее	1099345461	830459422,5	554380630,8
		σ	88,32283027	622231,399	262541,9073
	100	среднее	1099345570	830519942,3	554245485,2
		σ	621,6679308	751940,0495	239222,9571
	Лучший рез-т	среднее	1099345168	830431446,4	554129122,8
			-0,0000215%	-0,0192496%	-0,0273557%
		σ	156,5310396	708606,5637	380981,1627
			-16,22827%	50,36132%	118,16986%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-7,524901197	-1,02904659	-1,981642008
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	нет	да	да
ALA		среднее	1099345083	829817431,1	553786123,9
			-0,0000292%	-0,0931747%	-0,0892375%
		σ	41,28612187	90,89316634	181509,3122
			-77,9046399%	-99,9807131%	3,9417862%
ГА Шена и Лиу		среднее	1099344977	829817453,8	553777917,7
			-0,0000389%	-0,0931749%	-0,0907079%
		σ	60,68808679	107,4422917	190873,8048
			-67,52116%	-99,97720%	9,30438%

Таблица А.17 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Europe, l_2, p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2
Время работы алгоритмов	t=12 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=50
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=10	p=20	p=50
Новый ГА серийн.	динамич.	среднее	1099345334	830585098,7	554453074,9
		σ	185,79661	616407,6088	268065,6023
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	1099345413	829820264,5	554339554,5
		σ	193,5418725	2852,616483	180047,3299
	10	среднее	1099345424	830380681,7	554206279,8
		σ	500,4406147	472992,2279	178440,8147
	20	среднее	1099345398	830448333,3	553945709,3
		σ	258,8525967	618135,707	277029,2072
	50	среднее	1099345318	830264965,6	554202095,6
		σ	191,0608657	228120,7921	289259,4819
	100	среднее	1099345466	830688240	554677867
		σ	307,0897037	819361,4773	420984,8316
	Лучший рез-т	среднее	1099345318	829820264,5	553945709,3
			-0,0000014%	-0,0920838%	-0,0915074%
		σ	191,0608657	2852,616483	277029,2072
			2,8333432%	-99,5372191%	3,3438102%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-0,461698368	-6,796030357	-7,208847676
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	нет	нет
ALA		среднее	1099345083	829817431,1	553786123,9
			-0,0000292%	-0,0931747%	-0,0892375%
		σ	41,28612187	90,89316634	181509,3122
			-77,9046399%	-99,9807131%	3,9417862%
ГА Шена и Лиу		среднее	1099344977	829817453,8	553777917,7
			-0,0000389%	-0,0931749%	-0,0907079%
		σ	60,68808679	107,4422917	190873,8048
			-67,52116%	-99,97720%	9,30438%

Таблица А.18 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты испытаний тестовой задачи Europe, l_2, р-медиан, без нормирования		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_2		
Время работы алгоритмов		t=12 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=50		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=10	p=20	p=50
ГА серийн. с динамич. популяцией	среднее	1099345405	830591331,9	554280750,1
	σ	186,8542612	471269,1629	174625,9313
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	1099345300	830320711,8	554185097,7
		-0,0000096%	-0,0325816%	-0,0172570%
	σ	129,2424219	392644,3665	307900,5055
		-30,83249%	-16,68362%	76,32003%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	1099345241	830233820,6	554275668,7
		-0,0000149%	-0,0430430%	-0,0009167%
	σ	178,2426307	571199,226	340833,9155
		-4,6087418%	21,2044562%	95,1794404%
Гипотеза о неравенстве мат. ожиданий Н:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-4,909626798	-3,739652881	-0,102777032
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да,преимущ. стат. значимо	да,преимущ. стат. значимо	да, преимущ. стат. значимо
ALA	среднее	1099345083	829817431,1	553786123,9
		-0,0000292%	-0,0931747%	-0,0892375%
	σ	41,28612187	90,89316634	181509,3122
		-77,90463%	-99,98071%	3,9417862%
ГА Шена и Лиу	среднее	1099344977	829817453,8	553777917,7
		-0,0000389%	-0,0931749%	-0,0907079%
	σ	60,68808679	107,4422917	190873,8048
		-67,52116%	-99,97720%	9,30438%

Таблица А.19 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Ionosphere, $l_2{}^2$, р-медиан, без нормирования
Число векторов данных	351
Размерность пространства	34
Метрика или мера расстояния	$l_2{}^2$
Время работы алгоритмов	t=40 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=10
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=5	p=10
Новый ГА серийн.	динамич.	среднее	8454,936764	7282,506926	5981,229037
		σ	4,747388006	0,469155221	0,144604414
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	8476,678084	7283,948309	5981,179273
		σ	16,51483244	1,14671524	0,146988434
	10	среднее	8457,049144	7283,942069	5981,199627
		σ	5,167157833	1,00848066	0,136744055
	20	среднее	8455,349254	7283,284223	5981,257077
		σ	4,996301293	0,938492768	0,180780302
	50	среднее	8453,957118	7282,586506	5981,268713
		σ	4,533821425	0,760842738	0,173567814
	100	среднее	8476,2745	7291,206583	6070,214141
		σ	17,18082254	8,047493313	98,37238839
	Лучший рез-т	среднее	8453,957118	7282,586506	5981,179273
			-0,0115867%	0,0010928%	-0,0008320%
		σ	4,533821425	0,760842738	0,173567814
			-4,4986123%	62,1729235%	20,0294031%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-1,155952836	0,487633963	-1,206534437
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	8451,555361	7281,942022	6025,381871
			-0,0399932%	-0,0077570%	0,7381900%
		σ	2,444692884	0,20605035	10,99590865
			-48,50446%	-56,08055%	7504,13072%
ГА Шена и Лиу		среднее	8451,543671	7281,886573	5976,867801
			-0,0401215%	-0,0085167%	-0,0729157%
		σ	2,447232935	0,156509064	2,057939019
			-48,45096%	-66,64023%	1323,15090%

Таблица А.20 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Ionosphere, $l_2{}^2$, p-медиан, без нормирования
Число векторов данных	351
Размерность пространства	34
Метрика или мера расстояния	$l_2{}^2$
Время работы алгоритмов	t=40 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=10
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=5	p=10
Новый ГА серийн.	динамич.	среднее	8452,118955	7282,248866	5980,352738
		σ	3,097627349	0,461166419	1,548062823
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	8471,946573	7284,342307	5981,712153
		σ	17,35724626	1,011054388	2,774521141
	10	среднее	8470,139514	7283,222415	5980,506967
		σ	16,76199339	1,261325967	1,490404932
	20	среднее	8456,966944	7283,363556	5980,872837
		σ	4,894955693	1,072889122	0,018114489
	50	среднее	8452,319805	7282,21598	5980,029935
		σ	3,292603452	0,324820042	2,052907427
	100	среднее	8487,531144	7293,795324	6162,30479
		σ	11,8799981	19,40839367	191,9016926
	Лучший рез-т	среднее	8452,319805	7282,21598	5980,029935
			0,0023763%	-0,0004516%	-0,0053977%
		σ	3,292603452	0,324820042	2,052907427
			6,2943692%	-29,5655475%	32,6113770%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0,344146096	-0,319325227	-0,68764965
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	8451,555361	7281,942022	6025,381871
			-0,0399932%	-0,0077570%	0,7381900%
		σ	2,444692884	0,20605035	10,99590865
			-48,50446%	-56,08055%	7504,13072%
ГА Шена и Лиу		среднее	8451,543671	7281,886573	5976,867801
			-0,0401215%	-0,0085167%	-0,0729157%
		σ	2,447232935	0,156509064	2,057939019
			-48,45096%	-66,64023%	1323,15090%

Таблица А.21 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты испытаний тестовой задачи Ionosphere, $l_2{}^2$, р-медиан, без нормирования		
Число векторов данных		351		
Размерность пространства		34		
Метрика или мера расстояния		$l_2{}^2$		
Время работы алгоритмов		t=40 сек		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=10		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=3	p=5	p=10
ГА серийн. с динамич. популяцией	среднее	8454,936764	7282,506926	5981,229037
	σ	4,747388006	0,469155221	0,144604414
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	8450,95014	7281,831139	5980,777065
		-0,0471514%	-0,0092796%	-0,0075565%
	σ	0	2,81246E-12	0,910490796
		-100,000000%	-100,000000%	529,6424662%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	8450,973501	7281,941269	5980,980785
		-0,0468751%	-0,0077673%	-0,0041505%
	σ	0,065944874	0,246391672	0,802014281
		-98,6109230%	-47,4818438%	454,6264192%
Гипотеза о неравенстве мат. ожиданий $H: f^*_{неоднородн2} < f^*_{динамич}$	Статистика Стьюдента	-6,465942754	-8,268341112	-2,359609566
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. значимо	да, преимущ. стат. значимо	да, преимущ. стат. незначимо
ALA	среднее	8451,555361	7281,942022	6025,381871
		-0,0399932%	-0,0077570%	0,7381900%
	σ	2,444692884	0,20605035	10,99590865
		-48,5044643%	-56,0805591%	7504,130726%
ГА Шена и Лиу	среднее	8451,543671	7281,886573	5976,867801
		-0,0401215%	-0,0085167%	-0,0729157%
	σ	2,447232935	0,156509064	2,057939019
		-48,45096%	-66,64023%	1323,15090%

Таблица А.22 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Miss America, l_2^2, p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2^2
Время работы алгоритмов	t=2 мин 40 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=100
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=20	p=50	p=75
Новый ГА серийн.	динамич.	среднее	993282,7843	827904,0182	756019,8023
		σ	134,6026006	366,0723926	566,5497517
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	993384,0528	828017,1349	755822,0532
		σ	276,6849607	732,8621859	1597,645243
	10	среднее	993148,9383	828061,0948	755506,2732
		σ	188,1760561	878,036415	995,7256211
	20	среднее	993180,4638	827897,6058	756066,5051
		σ	97,74373393	1005,488671	898,3315063
	50	среднее	993187,4868	827604,4865	756960,2006
		σ	214,6237662	925,538092	1155,526327
	100	среднее	994461,3252	829589,0342	758777,2616
		σ	1602,589065	1805,02888	1897,579368
	Лучший рез-т	среднее	993148,9383	827604,4865	755506,2732
			-0,0134751%	-0,0361795%	-0,0679254%
		σ	188,1760561	925,538092	995,7256211
			39,8012039%	152,8292520%	75,7525475%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-4,481156077	-1,648343216	-2,455187262
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	нет	да	нет
ALA		среднее	993219,4583	833266,8292	765261,7524
			-0,0063754%	0,6477576%	1,2224481%
		σ	386,5734032	1434,037637	502,8179893
			187,19608%	291,7360791%	-11,24910%
ГА Шена и Лиу		среднее	992902,6196	831822,5739	761278,3463
			-0,0382787%	0,4732206%	0,6960292%
		σ	190,2351119	792,0912002	1755,331908
			41,3309334%	116,375569%	209,82837%

Таблица А.23 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Miss America, l_2^2, p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2^2
Время работы алгоритмов	t=2 мин 40 сек
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=100
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=20	p=50	p=75
Новый ГА серийн.	динамич.	среднее	993221,2199	827707,828	755573,6081
		σ	217,647801	711,6067404	1323,301354
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	993263,1527	827886,1377	755845,478
		σ	387,9112044	391,8714382	591,2560693
	10	среднее	993201,8745	828035,1668	755929,1861
		σ	151,383005	669,5741462	1466,276895
	20	среднее	993245,7286	827866,5579	756506,736
		σ	183,2359062	621,8805348	519,3800857
	50	среднее	993152,0972	827967,017	756969,2772
		σ	190,0580702	480,4190857	825,0280209
	100	среднее	995077,2038	830949,2911	759518,7404
		σ	1711,030146	1346,235515	1601,173801
	Лучший рез-т	среднее	993152,0972	827866,5579	755845,478
			-0,0069594%	0,0191770%	0,0359819%
		σ	190,0580702	621,8805348	591,2560693
			-12,67631%	-12,60895%	-55,31962%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-0,565216461	0,919950939	1,027397729
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	993219,4583	833266,8292	765261,7524
			-0,0063754%	0,6477576%	1,2224481%
		σ	386,5734032	1434,037637	502,8179893
			187,19608%	291,73607%	-11,24910%
ГА Шена и Лиу		среднее	992902,6196	831822,5739	761278,3463
			-0,0382787%	0,4732206%	0,6960292%
		σ	190,2351119	792,0912002	1755,331908
			41,3309334%	116,375562%	209,82837%

Таблица А.24 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты испытаний тестовой задачи Miss America, l_2^2, р-медиан, без нормирования		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_2^2		
Время работы алгоритмов		t=2 мин 40 сек		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=100		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=20	p=50	p=75
ГА серийн. с динамич. популяцией	среднее	993282,7843	827904,0182	756019,8023
	σ	134,6026006	366,0723926	566,5497517
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	993138,8463	827993,5051	755938,429
		-0,0144911%	0,0108089%	-0,0107634%
	σ	147,4644387	807,2509625	852,2193467
		9,5554158%	120,51675%	50,4226847%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	993143,3327	828391,3724	755422,0981
		-0,0140395%	0,0588660%	-0,0790593%
	σ	163,3881952	717,0211689	1502,53389
		21,3856155%	95,8686816%	165,2077572%
Гипотеза о неравенстве мат. ожиданий Н:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-4,909626798	-3,739652881	-0,102777032
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. значимо	нет, преимущ. не наблюдается	да, преимущ. стат. значимо
ALA	среднее	993219,4583	833266,8292	765261,7524
		-0,0063754%	0,6477576%	1,2224481%
	σ	386,5734032	1434,037637	502,8179893
		187,19608%	291,73607%	-11,24910%
ГА Шена и Лиу	среднее	992902,6196	831822,5739	761278,3463
		-0,0382787%	0,4732206%	0,6960292%
	σ	190,2351119	792,0912002	1755,331908
		41,3309334%	116,375562%	209,82837%

Таблица А.25 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением

Характеристика набора данных, задачи и алгоритма			Значение		
Наименование набора данных			Результаты испытаний тестовой задачи Mopsi-Joensuu, l_2, p-медиан, без нормирования		
Число векторов данных					
Размерность пространства					
Метрика или мера расстояния			l_2		
Время работы алгоритмов			t=4 мин.		
Число запусков алгоритмов			30		
Максимальное число кластеров			p_{max}=20		
Тип решаемой задачи			p-медианная		
Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=20
Новый ГА серийн.	динамич.	среднее	528,2776838	359,4104691	208,7595881
		σ	1,11273E-07	3,50024E-05	9,45727E-06
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	528,2776838	359,4104649	208,7595883
		σ	7,19583E-08	3,76075E-05	8,40894E-06
	10	среднее	528,2776837	359,4104417	208,7595912
		σ	2,3005E-07	3,13811E-05	8,11523E-06
	20	среднее	528,2776839	359,4104993	208,7595815
		σ	2,06551E-07	2,2757E-05	9,00574E-06
	50	среднее	528,2776839	359,410446	208,7595931
		σ	1,82603E-07	3,45515E-05	7,53147E-06
	100	среднее	528,2780818	360,6531331	211,3856346
		σ	0,001049194	0,599289632	2,137209376
	Лучший рез-т	среднее	528,2776837	359,4104417	208,7595815
			0,0000000%	-0,0000076%	-0,0000032%
		σ	2,3005E-07	3,13811E-05	9,00574E-06
			106,74391%	-10,3458076%	-4,7744197%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-2,741008966	-3,193072833	-2,761722968
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	нет	нет	нет
ALA		среднее	528,2776257	359,4104279	209,4204294
			-0,0000110%	-0,0000115%	0,3165562%
		σ	6,22167E-07	2,10886E-05	0,836084605
			459,13720%	-39,75087%	8840552,83%
ГА Шена и Лиу		среднее	528,277623	359,4103643	209,9823139
			-0,0000115%	-0,0000292%	0,5857100%
		σ	0	1,11269E-09	0,845391962
			-100,00000%	-99,99682%	8938967,64%

Таблица А.26 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты испытаний тестовой задачи Mopsi-Joensuu, l_2, p-медиан, без нормирования
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_2
Время работы алгоритмов	t=4 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=5	p=10	p=20
Новый ГА серийн.	динамич.	среднее	528,2776838	359,410448	208,7595892
		σ	2,28979E-07	4,26334E-05	7,92666E-06
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	528,2776836	359,4104721	208,7595884
		σ	1,13694E-07	6,06162E-05	1,04983E-05
	10	среднее	528,2776836	359,4104741	208,7595926
		σ	2,72128E-07	3,62631E-05	9,61845E-06
	20	среднее	528,2776836	359,4104521	208,7595869
		σ	1,81696E-07	3,14781E-05	1,15262E-05
	50	среднее	528,2776838	359,4104808	208,7595882
		σ	1,67529E-07	3,85716E-05	6,36461E-06
	100	среднее	528,278875	360,9723395	216,67501
		σ	0,001483968	0,790640281	5,492732668
	Лучший рез-т	среднее	528,2776836	359,4104521	208,7595869
			0,0000000%	0,0000011%	-0,0000011%
		σ	1,13694E-07	3,14781E-05	1,15262E-05
			-50,34737%	-26,1656123%	45,4111143%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	-5,380226237	0,421771049	-0,902285306
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	нет	да	да
ALA		среднее	528,2776257	359,4104279	209,4204294
			-0,0000110%	-0,0000115%	0,3165562%
		σ	6,22167E-07	2,10886E-05	0,836084605
			459,13720%	-39,7508732%	8840552,83%
ГА Шена и Лиу		среднее	528,277623	359,4103643	209,9823139
			-0,0000115%	-0,0000292%	0,5857100%
		σ	0	1,11269E-09	0,845391962
			-100,0000%	-99,99682%	8938967,64%

Таблица А.27 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты испытаний тестовой задачи Mopsi-Joensuu, l_2, p-медиан, без нормирования		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_2		
Время работы алгоритмов		t=4 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=5	p=10	p=20
ГА серийн. с динамич. популяцией	среднее	528,2776838	359,4104691	208,7595881
	σ	1,11273E-07	3,50024E-05	9,45727E-06
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	528,277659	359,4104199	208,7600102
		-0,0000047%	-0,0000137%	0,0002022%
	σ	2,46246E-05	2,7282E-05	1,01559E-06
		22029,956%	-22,05686%	-89,26127%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	528,2776756	359,410499	208,7600086
		-0,0000015%	0,0000083%	0,0002014%
	σ	1,95451E-05	2,42699E-05	1,47381E-06
		17465,043%	-30,66225%	-84,41611%
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-3,233970248	5,443508471	340,3236783
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. значимо	да, преимущ. стат. незначимо	нет, преимуществ не наблюдается
ALA	среднее	528,2776257	359,4104279	209,4204294
		-0,0000110%	-0,0000115%	0,3165562%
	σ	6,22167E-07	2,10886E-05	0,836084605
		459,13720%	-39,75087%	8840552,83%
ГА Шена и Лиу	среднее	528,277623	359,4103643	209,9823139
		-0,0000115%	-0,0000292%	0,5857100%
	σ	0	1,11269E-09	0,845391962
		-100,0000%	-99,99682%	8938967,64%

Таблица А.28 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением.

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 1526ТЛ1Н184, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=5
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	73948,43660	68310,74431	64762,53298
		σ	1,455191E-011	1,455191E-011	1,455191E-011
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	74093,44769	68521,04313	64936,3245648
		σ	266,50797	115,083625	148,87183
	10	среднее	74045,30447	68386,70471	64838,42247
		σ	256,28829	94,77178	136,13695
	20	среднее	73948,43660	68410,25174	64838,42247
		σ	1,57178E-011	93,11791	136,13695
	50	среднее	73948,43660	68310,74431	64762,53298
		σ	1,57178E-011	1,57178E-011	1,57178E-011
	100	среднее	74846,10778	69768,12428	66724,38111
		σ	1536,21822	2737,87414	3124,01546
	Лучший рез-т	среднее	73948,43660	68310,74431	64762,53298
			0,0000000%	0,0000000%	0,0000000%
		σ	1,57178E-011	1,57178E-011	1,57178E-011
			8,0123450%	8,0123450%	8,0123450%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	0	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	73948,43660	68310,74431	64762,53298
			0,0000000%	0,0000000%	0,0000000%
		σ	1,57178E-011	1,455191E-011	1,57178E-011
			8,0123450%	0,0000000%	8,0123450%
ГА Шена и Лиу		среднее	73948,43657	68310,74431	64762,53289
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,57178E-011	1,455191E-011	0
			8,0123450%	0,0000000%	-100,00000%

Таблица А.29 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 1526ТЛ1Н184, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	73948,43660	68310,74431	64762,53298
		σ	1,57178E-011	1,57178E-011	1,57178E-011
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	74008,27591	68600,53784	65094,39841
		σ	158,31992	289,99839	410,72974
	10	среднее	73948,43660	68504,25868	64762,53298
		σ	1,57178E-011	138,95517	1,57178E-011
	20	среднее	73948,43660	68335,35520	64762,53298
		σ	1,57178E-011	65,11430	1,57178E-011
	50	среднее	73948,43660	68310,74431	64762,53298
		σ	1,57178E-011	1,57178E-011	1,57178E-011
	100	среднее	74761,12911	69923,90632	67170,19260
		σ	376,42336	1154,19711	1606,87101
	Лучший рез-т	среднее	73948,43660	68310,74431	64762,53298
			0,0000000%	0,0000000%	0,0000000%
		σ	1,57178E-011	1,57178E-011	1,57178E-011
			0,0000000%	0,0000000%	0,0000000%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	0	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	73948,43660	68310,74431	64762,53298
			0,0000000%	0,0000000%	0,0000000%
		σ	1,57178E-011	1,455191E-011	1,57178E-011
			8,0123450%	0,0000000%	8,0123450%
ГА Шена и Лиу		среднее	73948,43657	68310,74431	64762,53289
			-0,0116490%	-0,0210809%	0,0109219%
		σ	1,57178E-011	1,455191E-011	0
			8,0123450%	0,0000000%	-100,00000%

Таблица А.30 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 1526ТЛ1Н184, l_1, p-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_1		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=3	p=4	p=5
ГА серийн. с динамич. популяцией	среднее	73948,43660	68310,74431	64762,53298
	σ	1,455191E-11	1,455191E-11	1,455191E-011
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	73948,43660	68310,74431	64762,53298
		0,0000000%	0,0000000%	0,0000000%
	σ	1,57178E-011	1,57178E-011	1,57178E-011
		8,0123450%	8,0123450%	8,0123450%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	73948,43660	68310,74431	64762,53298
		0,0000000%	0,0000000%	0,0000000%
	σ	1,57178E-011	1,57178E-011	1,57178E-011
		8,0123450%	8,0123450%	8,0123450%
Результат полного перебора		73948,4366	68310,74431	64762,53298
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	0	0	0
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. незначимо	да, преимущ. стат. незначимо	да, преимущ. стат. незначимо
ALA	среднее	73948,43660	68310,74431	64762,53298
		0,0000000%	0,0000000%	0,0000000%
	σ	1,57178E-11	1,455191E-11	1,57178E-11
		8,0123450%	0,0000000%	8,0123450%
ГА Шена и Лиу	среднее	73948,43657	68310,74431	64762,53289
		-0,0116490%	-0,0210809%	0,0109219%
	σ	1,57178E-011	1,455191E-11	0
		8,0123450%	0,0000000%	-100,00000%

Таблица А.31 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением.

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 5503, l_1, р-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=5
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	10	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	20	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	50	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	100	среднее	76860,52050	71053,25577	66948,31607
		σ	599,61933	777,68189	990,51374
	Лучший рез-т	среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	0	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-
ГА Шена и Лиу		среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-

Таблица А.32 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий 5503, l_1, р-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	10	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	20	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	50	среднее	76231,70405	70288,82164	65988,16397
		σ	0	0	0
	100	среднее	76830,69374	71304,58305	67884,22693
		σ	537,66492	769,74146	1519,41517
	Лучший рез-т	среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	0	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-
ГА Шена и Лиу		среднее	76231,70405	70288,82164	65988,16397
			0,00000%	0,00000%	0,00000%
		σ	0	0	0
			-	-	-

Таблица А.33 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий 5503, l_1, р-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_1		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=3	p=4	p=5
ГА серийн. с динамич. популяцией	среднее	76231,70405	70288,82164	65988,16397
	σ	0	0	0
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	76231,70405	70288,82164	65988,16397
		0,00000%	0,00000%	0,00000%
	σ	0	0	0
		-	-	-
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	76231,70405	70288,82164	66054,96461
		0,00000%	0,00000%	0,10123%
	σ	0	0	176,73789
		-	-	-
Результат полного перебора		76231,70405	70288,82164	65988,16397
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	-	-	2,92770
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	рез-ты равны	рез-ты равны	нет, премущ. не наблюдается
ALA	среднее	76231,70405	70288,82164	65988,16397
		0,00000%	0,00000%	0,00000%
	σ	0	0	0
		-	-	-
ГА Шена и Лиу	среднее	76231,70405	70288,82164	65988,16397
		0,00000%	0,00000%	0,00000%
	σ	0	0	0
		-	-	-

Таблица А.34 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с полным объединенным решением.

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий Ionosphere, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=5
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	8815,65413	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	8848,37447	8017,91326	7634,67334
		σ	22,35224	0	2,73387
	10	среднее	8835,56522	8017,91326	7633,20405
		σ	24,84201	0	1,96473E-012
	20	среднее	8822,19820	8017,91326	7633,20405
		σ	17,31397	0	1,96473E-012
	50	среднее	8815,65413	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
	100	среднее	8910,92293	8186,07924	8080,32589
		σ	172,61505	324,49723	386,35745
	Лучший рез-т	среднее	8815,65413	8017,91326	7633,20405
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	0	1,96473E-012
			0,00000%	-	0,00000%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	-	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	8815,65413	8017,91326	7633,20405
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	0	1,96473E-012
			0,00000%	-	0,00000%
ГА Шена и Лиу		среднее	8815,65412	8017,91326	7633,20403
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	2,46582E-012	9,82366E-012
			0,00000%	-	-50,00000%

Таблица А.35 – сравнительные результаты работы серийных ГА с жадной эвристикой для задач кластеризации и размещения: варианты алгоритма с частичным объединенным решением

Характеристика набора данных, задачи и алгоритма	Значение
Наименование набора данных	Результаты тестовых испытаний электрорадиоизделий Ionosphere, l_1, p-медиан, с нормированием
Число векторов данных	
Размерность пространства	
Метрика или мера расстояния	l_1
Время работы алгоритмов	t=1 мин.
Число запусков алгоритмов	30
Максимальное число кластеров	p_{max}=20
Тип решаемой задачи	p-медианная

Версия алгоритма	Объем популяции	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
			p=3	p=4	p=5
Новый ГА серийн.	динамич.	среднее	8815,65413	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
ГА серийн., абс. значения и в %% к рез. нового алгоритма	5	среднее	8835,28633	8017,91326	7633,67019
		σ	24,48565	0	1,23330
	10	среднее	8815,65413	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
	20	среднее	8815,6541369	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
	50	среднее	8815,65413	8017,91326	7633,20405
		σ	1,96473E-012	0	1,96473E-012
	100	среднее	9193,78113	8879,28350	8658,57864
		σ	331,37677	578,15993	700,82086
	Лучший рез-т	среднее	8815,65413	8017,91326	7633,20405
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	0	1,96473E-012
			0,00000%	-	0,00000%
	Гипотеза о равенстве мат. ожиданий с новым алг.	t-статистика	0	-	0
		Порог при уровне значимости 1%	2,3924		
		Принятие гипотезы	да	да	да
ALA		среднее	8815,65413	8017,91326	7633,20405
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	0	1,96473E-012
			0,00000%	-	0,00000%
ГА Шена и Лиу		среднее	8815,65412	8017,91326	7633,20403
			0,00000%	0,00000%	0,00000%
		σ	1,96473E-012	2,46582E-012	9,82366E-012
			0,00000%	-	-50,00000%

Таблица А.36 – сравнительные результаты работы алгоритмов серийного решения задач кластеризации и размещения с неоднородной популяцией

Наименование набора данных		Результаты тестовых испытаний электрорадиоизделий Ionosphere, l_1, p-медиан, с нормированием		
Число векторов данных				
Размерность пространства				
Метрика или мера расстояния		l_1		
Время работы алгоритмов		t=1 мин.		
Число запусков алгоритмов		30		
Максимальное число кластеров		p_{max}=20		
Тип решаемой задачи		p-медианная		
Версия алгоритма	Значение целевой функции	Число кластеров (центров/центроидов/медоидов)		
		p=3	p=4	p=5
ГА серийн. с динамич. популяцией	среднее	8815,65413	8017,91326	7633,20405
	σ	1,96473E-012	0	1,96473E-012
ГА с ЖЭ с неоднородн. популяцией, вар. 1	среднее	8815,65413	8017,91326	7633,20405
		0,00000%	0,00000%	0,00000%
	σ	0	0	0
		0,00000%	-	0,00000%
ГА с ЖЭ с неоднородн. популяцией, вар. 2	среднее	8815,65413	8017,91326	7633,67019
		0,00000%	0,00000%	0,10123%
	σ	1,96473E-012	0	1,23330
		0,00000%	0,00000%	0,00610%
Результат полного перебора		8815,65413	8017,91326	7633,20405
Гипотеза о неравенстве мат. ожиданий H:$f^*_{неоднородн2}<f^*_{динамич}$	Статистика Стьюдента	0	-	2,92770
	Порог статистики при α=1%	2,66328694		
	Состоятельность гипотезы	да, преимущ. стат. незначимо	рез-ты равны	нет, преимущ. не наблюдается
ALA	среднее	8815,65413	8017,91326	7633,20405
		0,00000%	0,00000%	0,00000%
	σ	1,96473E-012	0	1,96473E-012
		0,00000%	-	0,00000%
ГА Шена и Лиу	среднее	8815,65412	8017,91326	7633,20403
		0,00000%	0,00000%	0,00000%
	σ	1,96473E-012	2,46582E-012	9,82366E-012
		0,00000%	-	-50,00000%

ПРИЛОЖЕНИЕ Б. Акт об использовании результатов исследования

УТВЕРЖДАЮ
Директор
ОАО «Испытательный технический центр -
НПО ПМ» канд. техн. наук
В.И. Орлов

АКТ
о внедрении результатов диссертационного исследования
Гудымы Михаила Николаевича

Настоящим актом подтверждается, что алгоритмы метода жадных эвристик для решения серий задач автоматической группировки на основе генетического алгоритма с динамическими и неоднородными популяциями, а также усовершенствованный с их применением метод автоматической классификации электрорадиоизделий по классам качества и производственным партиям на основе данных входного контроля и дополнительных отбраковочных испытаний, внедрены в опытную эксплуатацию в составе системы автоматизированного формирования и контроля специальных партий электрорадиоизделий космического применения на ОАО «Испытательный технический центр – НПО ПМ» (г.Железногорск).

Применение указанных алгоритмов, разработанных в рамках диссертационного исследования соискателя ученой степени кандидата технических наук Гудымы Михаила Николаевича, позволило повысить стабильность результатов оценки степени однородности партий электрорадиоизделий, а также снизить затраты временных и вычислительных ресурсов. В результате этого появилась возможность проводить анализ однородности партий изделий в интерактивном режиме и встроить его в технологический процесс формирования партий электрорадиоизделий, предназначенных для космического приборостроения.

Зам директора «ОАО ИТЦ –НПО ПМ»
канд. техн. наук
В.В. Федосов