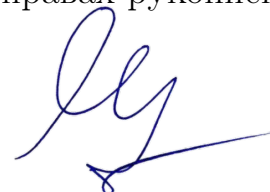


Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
науки и технологий имени академика М.Ф. Решетнева»

На правах рукописи



Малашин Иван Павлович

**БИНАРНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ
С ДИНАМИЧЕСКОЙ КОРРЕКЦИЕЙ ОБЛАСТИ
ПОИСКА ДЛЯ СЛОЖНЫХ ЗАДАЧ ОПТИМИЗАЦИИ
СО СМЕШАННЫМИ ПЕРЕМЕННЫМИ**

2.3.1 — Системный анализ, управление и обработка информации, статистика

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Сопов Евгений Александрович

Красноярск — 2026

Оглавление

	Стр.
Введение	4
1. Эволюционные метаэвристики для решения задач смешанной глобальной оптимизации «черного ящика»	10
1.1 Задачи смешанной глобальной оптимизации	10
1.2 Методы решения задач со смешанными переменными	14
1.3 Модификации эволюционных алгоритмов с использованием кластерного анализа	34
1.4 Выводы по главе 1	39
2. Метод динамической коррекции поискового пространства	41
2.1 Обоснование и разработка метода	41
2.2 Анализ работоспособности и визуализация на базовых тестовых функциях	61
2.3 Результаты исследования эффективности предложенного подхода на функциях СЕС2017	62
2.4 Результаты решения тестовых прикладных задач со смешанными переменными	76
2.5 Выводы по главе 2	83
3. Апробация метода динамической коррекции области поиска на практических задачах оптимизации	85
3.1 Оптимизация гиперпараметров моделей машинного обучения в системе классификации текстовых материалов госконтрактов	85
3.2 Задача оптимизации гиперпараметров модели бинарной классификации несбалансированных данных юридических лиц	97
3.3 Выводы по главе 3	111
Заключение	113
Список литературы	115

Приложение А. Результаты вычислительных экспериментов исследования метода динамической коррекции области поиска на эталонных тестовых функциях CEC2017	136
Приложение В. Акты об использовании результатов	139

Введение

Актуальность темы исследования. Задачи оптимизации приобретают все большее значение в различных областях науки и техники в связи с развитием систем интеллектуального анализа данных и усложнением процессов проектирования сложных систем. Методы эволюционных вычислений, в том числе генетические алгоритмы (ГА), широко применяются к решению подобных задач. Преимущества современных модификаций ГА заключаются в возможности решать задачи со смешанными переменными, выраженными в произвольных шкалах, путем представления решения в наиболее простой шкале – бинарной. Подобные алгоритмы позволяют решать сложные задачи глобальной оптимизации большой размерности, не накладывая жестких требований к свойствам целевой функции и функций ограничений. Однако бинарные генетические алгоритмы сталкиваются с проблемой выбора параметров кодирования вещественных переменных, таких как границы области поиска и шаг дискретизации. Слишком малый шаг сетки увеличивает размерность бинарного пространства поиска, что усложняет решение задачи. В то же время слишком крупный шаг дискретизации может привести к тому, что узлы сетки могут располагаться недостаточно близко к оптимумам в исходном вещественном пространстве. Эффективнее было бы использовать более плотную сетку в перспективных регионах вещественного пространства поиска и пропускать менее перспективные регионы. Эту проблему можно преодолеть, используя методы динамической коррекции границ области поиска, основанные на анализе плотности распределения решений популяции в пространстве поиска, в частности, подходы, использующие кластеризацию для выделения перспективных регионов пространства.

Степень разработанности темы. Исследованием и развитием теории и практики применения ГА активно занимаются отечественные и зарубежные ученые. Основоположником идеи генетических алгоритмов является Дж. Холланд, который заложил теоретический фундамент для всех последующих исследований. Дальнейшее развитие ГА связано с работами Д. Голдберга, К. Дэба и К. Де Йонга, которые предложили ряд новых операторов и схем применения ГА для бинарного пространства поиска, а также расширили область применения (многокритериальная оптимизация, задачи большой размерности и дру-

гие), включая бинарное кодирование. основополагающие работы по алгоритмам роя частиц и дифференциальной эволюции принадлежат Дж. Кеннеди и Р. Эберхарту. Данные алгоритмы М. Гонгом и А. Тьюсоном были адаптированы для бинарного поиска. В работах Д. Уитни предложены схемы перекодирования решений в ГА для повышения точности поиска, а в работах Н. Шраудольф и Р. Белью исследован ГА с динамическим кодированием решений. Среди российских исследователей необходимо выделить В.М. Курейчика, В.В. Курейчика и Л.А. Гладкова, работы которых посвящены развитию теории ГА. Работы А.В. Еремеева посвящены оценкам сложности ГА и разработке эффективных операторов поиска для различных представлений решений. В работах А.П. Карпенко представлены систематизация и анализ современного состояния области эволюционных вычислений. Работы научной школы Е.С. Семенкина посвящены созданию самоконфигурируемых и самонастраивающихся эволюционных и генетических алгоритмов. Работы Л.А. Демидовой и Ю.А. Скобцова развивают квантово-инспирированные генетические алгоритмы. В работах Е.А. Сопова исследованы способы бинарного кодирования для многих классов задач оптимизации.

Тем не менее, проблема адаптивной коррекции области поиска в бинарных ГА остается недостаточно изученной, а эффективные подходы для ее решения на текущий момент не предложены, что и подтверждает актуальность настоящего диссертационного исследования.

Объектом диссертационного исследования являются генетические алгоритмы, используемые для решения сложных задач глобальной оптимизации с алгоритмически заданными целевыми функциями и смешанными переменными; **предметом исследования** — метод динамической коррекции пространства поиска, применяемый в бинарных генетических алгоритмах.

Целью диссертационного исследования является повышение точности решения сложных задач оптимизации со смешанными переменными бинарным генетическим алгоритмом за счет динамической коррекции области поиска с сохранением размерности бинарного пространства.

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

1. Провести анализ текущего состояния развития методов эволюционной оптимизации и методов кодирования пространства поиска в бинарных генетических алгоритмах.

2. Обосновать и разработать метод динамической коррекции области поиска в генетическом алгоритме.

3. Обосновать и разработать алгоритм корректировки границ кластеров на основе относительной пригодности индивидов для адаптивного расширения или сужения области поиска в зависимости от качества решений на границах.

4. Обосновать и разработать процедуру динамической модификации схемы бинарного кодирования генотипов для объединения разрозненных областей поиска в едином бинарном представлении.

5. Провести экспериментальные исследования для оценки эффективности предложенного подхода на наборе эталонных тестовых задач.

6. Апробировать разработанный метод на практических задачах структурного синтеза моделей машинного обучения.

Научная новизна работы состоит в следующем:

1. Предложен новый метод динамической коррекции области поиска генетического алгоритма, отличающийся от известных процедурой итеративной кластеризации популяции, позволяющей за счет выявления перспективных и неперспективных кластеров определять необходимость сужения области поиска (пункт 4 паспорта специальности).

2. Впервые предложен алгоритм корректировки границ кластеров при динамической коррекции области поиска генетического алгоритма на основе относительной пригодности индивидов, обеспечивающий адаптивное расширение или сужение области поиска по каждой координате в зависимости от качества решений на границах (пункт 3 паспорта специальности).

3. Разработан новый способ бинарного кодирования вещественных переменных с помощью процедуры динамической модификации схемы кодирования, который, в отличие от известных способов бинарного кодирования вещественных переменных, объединяет идентифицированные перспективные области поиска в едином бинарном представлении, что обеспечивает повышение точности без увеличения размерности бинарного пространства поиска (пункт 5 паспорта специальности).

Теоретическая значимость работы заключается в получении новых знаний о способах решения сложных задач глобальной оптимизации со смешанными переменными. Разработанный метод динамической адаптации области поиска в рамках генетического алгоритма на основе кластерного анализа реше-

ний дополняет множество метаэвристических подходов оптимизации. Предложенный подход позволяет формализовать процесс локализации перспективных областей поиска, так как вместо случайного поиска для идентификации областей применяются методы кластерного анализа. Анализ количества, состава и характеристик кластеров позволяет лучше понимать структуру пространства поиска. Полученные результаты могут быть использованы при разработке новых алгоритмов оптимизации.

Практическая значимость работы обусловлена тем, что разработанный метод динамической коррекции области поиска позволяет создавать адаптивные алгоритмы оптимизации, не требующие от пользователя ручного управления схемой кодирования. Предложенный метод динамической коррекции в составе генетического алгоритма внедрен в рамках экспериментальных систем анализа данных по проектам «Классификация государственных контрактов на основе методов обработки естественного языка», «Интеллектуальный анализ юридических лиц для выявления недостоверных сведений в Едином государственном реестре юридических лиц», «Модели и алгоритмы для выявления налогоплательщиков, имеющих признаки незаконного использования схем аренды контрольно-кассовой техники», «Прогнозирование применения необходимых финансовых мер в рамках финансового мониторинга крупных налогоплательщиков». Применение предложенного подхода для проектирования моделей машинного обучения повысило эффективность решения прикладных задач по сравнению с ранее применяемыми методами проектирования.

Методы исследования. В работе использовались методы системного анализа, теории эволюционных алгоритмов, кластерного анализа, теории вероятностей и математической статистики, теории обработки информации, нейро-сетевого моделирования, а также подходы к созданию прикладных интеллектуальных информационных систем.

Основные положения, выносимые на защиту:

1. Предложенный алгоритм корректировки границ кластеров на основе относительной пригодности индивидов позволяет автоматически изменять структуру области поиска, что повышает точность бинарных генетических алгоритмов по сравнению с их известными версиями без корректировки границ при одинаковых вычислительных бюджетах в среднем на множестве эталонных те-

стовых задач, что показано на примере бинарных ГА, алгоритма роя частиц и дифференциальной эволюции.

2. Процедура динамической модификации схемы бинарного кодирования фенотипов (решений) обеспечивает объединение разрозненных после кластеризации областей вещественного пространства поиска в едином бинарном представлении и увеличивает точность бинарного кодирования в перспективных областях при сохранении размерности бинарного пространства поиска.

3. Разработанный метод динамической коррекции области поиска бинарного генетического алгоритма обеспечивает повышение точности решения задач глобальной оптимизации «черного ящика» со смешанными переменными по сравнению со стандартным бинарным генетическим алгоритмом без изменения размерности бинарного пространства поиска и при одинаковом вычислительном бюджете. Повышение точности достигается как для генетического алгоритма с фиксированными настройками, так и для его самоконфигурируемой версии.

Степень достоверности полученных результатов и выводов обеспечивается корректным применением методов сбора, обработки и статистического анализа данных экспериментального исследования предложенных подходов на репрезентативном множестве эталонных задач оптимизации и результатами решения практических задач.

Апробация работы. Процесс разработки и результаты, представленные в диссертации, докладывались и обсуждались на научных конференциях различного уровня, среди которых XXXI Международная открытая научная конференция «Modern Informatization Problems in the Technological and Telecommunication Systems Analysis and Synthesis (MIP-2026'AS)» (г. Йелм, штат Вашингтон, США, январь 2026 г.); XXX Байкальская Всероссийская с международным участием конференция «Информационные и математические технологии в науке и управлении» (г. Иркутск, 2025); XII Международная научная Четаевская конференция «Аналитическая механика, устойчивость и управление» (г. Казань, 2025 г.); Третья молодежная конференция с международным участием «Комплексная автоматизация проектирования и производства 2025» (КАПП(М)-2025) (г. Москва, 2025 г.); 2024 IEEE International Conference on Data Mining Workshops (ICDMW) (г. Абу-Даби, ОАЭ, 2024 г.); XXVIII Международная научно-практическая конференция «Решетневские чтения» (г. Красноярск,

2024 г.); The Thirteenth International Workshop on Mathematical Models and their Applications, Krasnoyarsk (г. Красноярск, 2024 г.).

Диссертационная работа обсуждалась на научных семинарах кафедры «Системного анализа и исследования операций» СибГУ им. М.Ф. Решетнева, кафедры автоматизированных и вычислительных систем Воронежского государственного технического университета, НОЦ «ФНС России» и НОЦ «Технологии искусственного интеллекта» при МГТУ им. Н.Э. Баумана»

Публикации. По теме данного исследования опубликовано 14 печатных работ, среди которых 5 статей в научных журналах, входящих в перечень ВАК, 4 статьи в журналах, входящих в Q1 Web of Science и Scopus, 1 публикация в материалах конференции, индексируемой Web of Science и Scopus, 4 публикации в других изданиях. Получены 2 свидетельства о государственной регистрации программ для ЭВМ.

Объем и структура работы. Диссертация состоит из введения, трёх глав, заключения и двух приложений. Полный объём диссертации составляет 140 страниц с 33 рисунками и 14 таблицами. Список литературы содержит 191 наименование.

1. Эволюционные метаэвристики для решения задач смешанной глобальной оптимизации «черного ящика»

1.1 Задачи смешанной глобальной оптимизации

Задачи смешанной глобальной оптимизации предполагают поиск глобального экстремума целевой функции с учетом комбинации непрерывных и дискретных переменных, часто при наличии ограничений [1; 2]. Такие задачи возникают в различных областях, от проектирования сложных инженерных систем до настройки гиперпараметров моделей машинного обучения. Они характеризуются сочетанием дискретных и непрерывных переменных, что приводит к увеличению сложности поиска решений. Даже при фиксированных значениях непрерывных параметров дискретные переменные порождают пространство решений, мощность множества решений которого растет экспоненциально с увеличением числа параметров («проклятие размерности») [3]. Например, в задачах структурно-параметрического синтеза моделей машинного обучения (machine learning, ML) дискретные переменные могут определять выбор архитектуры нейросети (число слоев, типы функций активации), а непрерывные переменные – веса модели. Такие задачи классифицируются как NP-трудные [4].

Многоэкстремальность целевой функции представляет ещё одну сложность в задачах смешанной глобальной оптимизации [5]. В областях, где целевая функция невыпукла и имеет множество локальных экстремумов (рисунок 1.1), классические методы оптимизации, такие как градиентный спуск, склонны к застреванию в субоптимальных решениях [6]. В частности, в задачах обучения глубоких нейросетей или градиентного бустинга ландшафт функции потерь часто содержит плато [7] и локальные экстремумы [8], что затрудняет поиск глобального оптимума.

Для преодоления этой проблемы применяются стохастические методы, включая различные метаэвристики (например, генетический алгоритм или алгоритм роя частиц), которые исследуют пространство решений глобально, сохраняя разнообразие решений и снижая риск преждевременной сходимости. Например, в оптимизации гиперпараметров XGBoost использование генетиче-

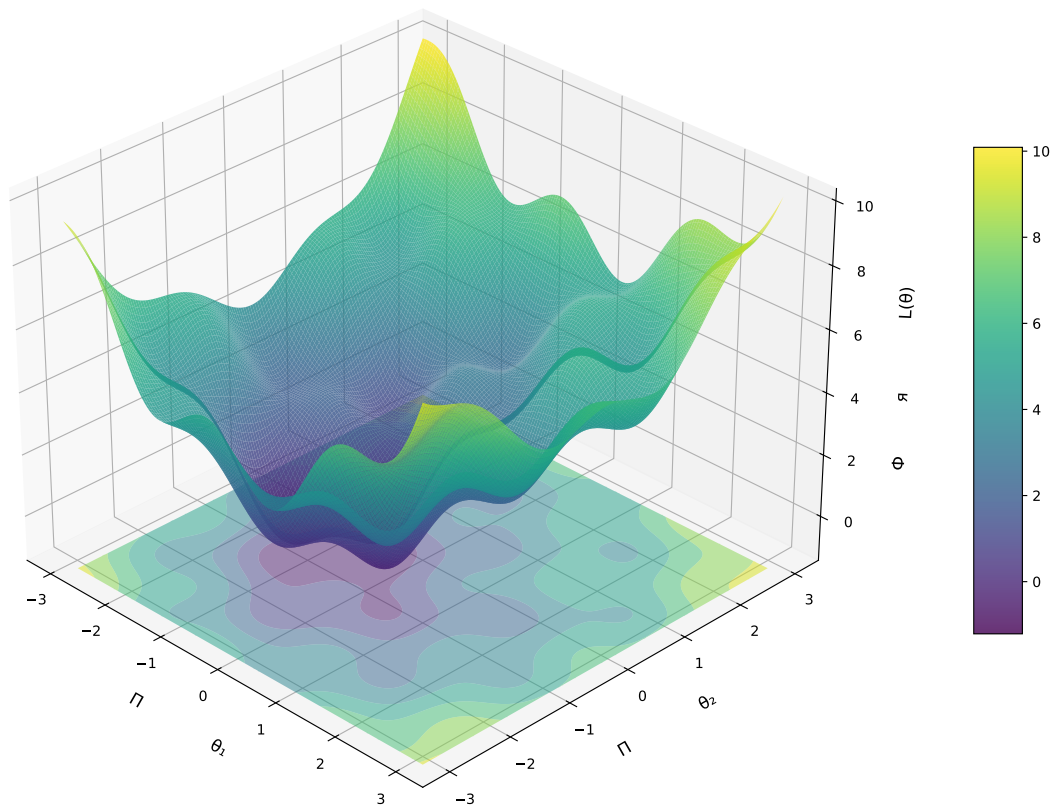


Рисунок 1.1 — Пример многоэкстремального ландшафта функции потерь.

ских алгоритмов позволяет находить комбинации параметров (глубина деревьев, скорость обучения), которые обеспечивают баланс между точностью и обобщающей способностью модели, минуя локальные оптимумы, характерные для градиентных подходов.

Наличие ограничений, определяющих допустимые области решений [9], также усложняет задачу. В задачах смешанной глобальной оптимизации ограничения могут быть как равенствами, так и неравенствами, отражающими физические законы, ресурсные ограничения (например, вычислительная мощность) или бизнес-правила (например, требования к интерпретируемости ML-моделей). В частности, при оптимизации архитектуры нейросети ограничения могут касаться количества параметров модели (для соблюдения требований к производительности) или времени вывода (для реального времени). Учет ограничений требует адаптации методов оптимизации, таких как введение штрафных функций, проекций на допустимое множество или использование методов внутренней точки. Однако в задачах со смешанными переменными эти подходы сталкиваются с проблемами дискретности: классические методы, основанные на оптимизации гладких функций, не работают напрямую. В ML-приложениях это проявляется, например, в задачах «справедливого обучения» (fair learning) [10],

где необходимо минимизировать ошибку модели с учетом ограничений на точность предсказаний.

Традиционные методы оптимизации, такие как последовательное квадратичное программирование (Sequential Quadratic Programming, SQP) [11], ориентированы на пространства поиска с непрерывными переменными и неэффективны при наличии дискретных переменных, которые критичны для задач структурно-параметрического синтеза (SPS) в машинном обучении. Например, выбор архитектуры модели (типы функций активации, количество слоев) или бинарные признаки требуют дискретного представления, тогда как методы SQP вынуждены использовать аппроксимации (округление или непрерывные аналоги), что приводит к потере точности. Даже комбинаторные подходы (метод ветвей и границ) сталкиваются с экспоненциальным ростом сложности в высокоразмерных задачах [12], таких как оптимизация гиперпараметров в нейросетях [13].

Кроме того, преобразование ограничений в штрафные функции усложняет ландшафт целевой функции, увеличивая риск застревания в локальных экстремумах. Вычисление матрицы Гессе для высокоразмерных пространств становится невозможным, что делает классические методы неприменимыми к задачам смешанной глобальной оптимизации большой размерности [14]. Такие ограничения традиционных методов подчеркивают необходимость перехода к альтернативным подходам, способным эффективно обрабатывать переменные разных типов и работать в условиях высокой размерности без использования градиентов [15]. Для решения задач глобальной оптимизации с алгоритмически заданными целевыми функциями и смешанными переменными в последние десятилетия хорошо себя зарекомендовали различные методы эволюционной оптимизации [16].

В работе рассматривается задача глобальной оптимизации

$$f(x) \rightarrow \min_{x \in S} (\max), \quad (1.1)$$

где $x = (x^r, x^z, x^o, x^b, x^c)$ — вектор переменных задачи, $x^r \in \mathbb{R}^{n_r}$ — вещественные переменные, $x^z \in \mathbb{Z}^{n_z}$ — целочисленные переменные, $x^o \in O^{n_o}$ — порядковые переменные, $x^b \in B_2^{n_b}$ — бинарные переменные, $x^c \in C^{n_c}$ — категориальные переменные, $S = \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \times O^{n_o} \times B_2^{n_b} \times C^{n_c}$ — пространство поиска.

Допустимая область может иметь сложную структуру, быть невыпуклой и даже состоять из нескольких несвязных областей. В общем случае как целевая функция, так и ограничения могут быть многоэкстремальными и задаваться моделью типа “черного ящика”. При этом каждое вычисление функции может требовать значительных вычислительных затрат.

Теоретически решение задачи глобальной оптимизации предполагает построение алгоритма, который способен за конечное число вычислений целевой функции определить значение, являющееся ε -приближением глобального минимума. Иными словами, требуется гарантировать, что найденное решение отличается от истинного глобального оптимума не более чем на заданную величину ε по значению целевой функции.

Однако на практике достижение подобной гарантии часто оказывается невозможным. В задачах высокой размерности или при наличии алгоритмически заданных целевых функций число необходимых вычислений может быть чрезвычайно велико, а вычисление значения функции само по себе может быть трудоемким [17]. Поэтому на практике применяется более прагматичная постановка задачи, в которой алгоритм должен за конечное число вычислений вернуть наилучшее найденное значение целевой функции [18]. При этом предполагается, что полученное решение либо принимается в качестве глобального оптимума, либо используется как приближение, после чего поиск может быть повторен с измененными параметрами алгоритма.

Дополнительные трудности возникают в задачах, где целевая функция определяется алгоритмически и не имеет аналитического выражения. В таких случаях трудно получить строгую оценку точности ε -приближения, поскольку неизвестны свойства функции, необходимые для построения теоретических границ ошибки. Поэтому в большинстве прикладных задач используется практическая постановка оптимизации, ориентированная на поиск наименьшего найденного значения функции при ограниченном числе вычислений.

1.2 Методы решения задач со смешанными переменными

Эволюционные алгоритмы (ЭА) [19] представляют собой класс стохастических методов оптимизации, вдохновленных принципами естественного отбора и генетики. Они находят применение в задачах смешанной глобальной параметрической оптимизации, где необходима одновременная обработка дискретных, непрерывных и категориальных переменных, когда целевая функция не является выпуклой, гладкой или зашумлена. Ключевые подклассы ЭА включают генетический алгоритм (Genetic Algorithm, GA) [20], дифференциальную эволюцию (Differential Evolution, DE) [21], алгоритм муравьиной колонии (Ant Colony Optimization, ACO) [22], эволюционная стратегия (Evolution Strategy, ES) [23] и генетическое программирование (Genetic Programming, GP) [24]. ЭА обладают высокой универсальностью при решении задач со смешанными переменными и демонстрируют высокую эффективность благодаря архитектурной гибкости, устойчивости к шуму и возможности естественной параллелизации вычислений.

К первой группе относятся бинарные алгоритмы [25], которые могут работать напрямую с битовыми строками. Среди них выделяются такие методы, как ГА, алгоритм на основе оценок распределений (Estimation of Distribution Algorithms, EDA) [26] и эволюционное программирование (Evolutionary Programming, EP) [27].

Вторую группу составляют алгоритмы [28], предназначенные для работы с вещественными переменными. К ним относятся дифференциальная эволюция (DE) [29], эволюционные стратегии (ES) [30] и их расширенная версия – CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [31].

Третья группа включает алгоритмы [32], которые применяются для задач, где решение представляет собой упорядоченную последовательность или комбинацию элементов. Примерами являются генетические алгоритмы для задач комбинаторной оптимизации (например, как задача о коммивояжёре (TSP) [33] и задача размещения [34]). Также к этой категории относят алгоритм муравьиной колонии (ACO) [35], хотя он отличается от традиционных эволюционных алгоритмов, но используется для решения схожих задач.

В бинарных эволюционных алгоритмах решение представляется в виде вектора бинарных переменных, называемого хромосомой [36]. Такой подход предполагает, что переменные различных типов, присутствующие в задаче смешанной оптимизации, должны быть преобразованы к единому бинарному представлению. В результате исходный вектор решения, содержащий бинарные, номинальные, порядковые и вещественные переменные, отображается в бинарное пространство поиска. Пусть исходный вектор решения имеет вид

$$x = (x^r, x^z, x^o, x^b, x^c),$$

где x^b – бинарные переменные, x^c – номинальные, x^o – порядковые (включая целочисленные), x^r – вещественные, x^z – целочисленные. После бинаризации формируется единая хромосома

$$g = g^r \parallel g^z \parallel g^o \parallel g^b \parallel g^c,$$

которая является конкатенацией двоичных кодов, полученных для переменных каждого типа [37].

Наиболее простым вариантом является решение задач с исходными бинарными переменными, которые не требуют операции кодирования и декодирования, то есть генотипическое и фенотипическое представления совпадают.

Для номинальных переменных применяется процедура нумерации категорий. Пусть переменная принимает значения из множества категорий $\{c_1, \dots, c_m\}$. Каждой категории ставится в соответствие целочисленный индекс, после чего используется стандартное двоичное кодирование этого индекса. Число бит определяется как

$$l = \lceil \log_2 m \rceil.$$

Основной недостаток такого кодирования связан с тем, что между номинальными значениями отсутствует естественное отношение порядка [38]. Поэтому сама нумерация категорий является произвольной и не отражает структуру исходного пространства решений. В результате близость бинарных строк не соответствует содержательной близости категорий, а бинарное пространство поиска не обладает регулярной структурой относительно исходной задачи.

Порядковые и целочисленные переменные кодируются аналогичным образом, однако в данном случае нумерация значений отражает их естественный

порядок. Значения переменной упорядочиваются, каждому значению сопоставляется индекс, который затем кодируется стандартным бинарным кодом. Такой подход сохраняет информацию о порядке значений, хотя и здесь возможны ситуации, когда близкие значения переменной соответствуют существенно различающимся бинарным строкам.

Наиболее серьёзные проблемы возникают при переводе вещественных переменных в бинарное представление, поскольку непрерывный интервал значений необходимо предварительно дискретизировать. [39]. Пусть вещественная переменная определяется на интервале $[a, b]$. Тогда вводится шаг дискретизации h , формирующий набор допустимых значений

$$x_k = a + ih, \quad i = 0 \dots K - 1,$$

где K – число узлов сетки, $x_0 = a$, $x_{K-1} = b$.

Каждому узлу сетки ставится в соответствие индекс, который далее кодируется двоичным числом. Число бит определяется количеством узлов сетки и растёт при уменьшении шага дискретизации. Такая схема позволяет использовать бинарные алгоритмы для решения задач оптимизации с непрерывными переменными, однако она вводит аппроксимацию: поиск осуществляется только среди узлов сетки. Истинное положение глобального оптимума может находиться между ними.

В практических задачах шаг сетки обычно уменьшается для повышения точности представления. Однако это приводит к увеличению числа бит, необходимых для кодирования переменной. Увеличение длины хромосомы, в свою очередь, приводит к росту размерности пространства поиска и усложняет работу алгоритма. При бинаризации вещественных переменных возникает противоречие между точностью представления и размером пространства поиска.

После кодирования всех переменных формируется единый бинарный вектор решения g длины L , где L определяется суммарным числом бит, используемых для кодирования всех переменных. Таким образом, задача смешанной оптимизации преобразуется в задачу бинарной оптимизации в пространстве размерности L . Это позволяет применять стандартные операторы бинарных эволюционных алгоритмов независимо от типов переменных.

В алгоритмах, которые исходно предназначены для работы в вещественном пространстве, часто используется альтернативный подход. Двоичная хро-

мосома представляется как вещественный вектор, компоненты которого принимают значения на интервале $[0,1]$. При вычислении целевой функции выполняется пороговое преобразование: если значение компоненты меньше 0.5, соответствующий бит принимается равным нулю, иначе – единице. После этого полученный бинарный вектор декодируется в переменные исходной задачи. Такой подход позволяет использовать алгоритмы непрерывной оптимизации для решения бинарных задач, однако фактически переносит поиск в вещественное пространство большой размерности [40].

Генетические алгоритмы представляют собой класс популяционных стохастических методов глобальной оптимизации, основанных на формализации механизмов биологической эволюции, таких как естественный отбор, наследственность, рекомбинация и мутация. Данный подход был систематизирован в работах Дж. Холланда [41] и изначально рассматривался как способ поиска оптимальных решений в сложных задачах оптимизации. В отличие от многих детерминированных методов оптимизации, ГА оперируют не с одиночным решением, а с популяцией кандидатов, что позволяет осуществлять параллельный поиск и снижать вероятность застревания в локальных экстремумах.

В основе ГА лежит представление решений в виде хромосом – формализованных генетических кодов, состоящих из отдельных генов, каждый из которых кодирует один или несколько параметров оптимизируемой задачи. Такое представление может принимать различные формы, включая бинарные строки, векторы вещественных чисел. Универсальность хромосомного кодирования позволяет в рамках одного алгоритма совместно оптимизировать дискретные, целочисленные и непрерывные переменные, что является принципиально важным для задач структурно-параметрического синтеза, в которых необходимо одновременно определять архитектуру модели и значения её параметров [42]. В контексте машинного обучения это позволяет единообразно описывать такие разнородные компоненты, как выбор типа функции активации, число слоёв нейронной сети, связей между нейронами и значения весовых коэффициентов.

Классический генетический алгоритм включает последовательность этапов, начиная с инициализации начальной популяции [43], как правило, случайным образом, с последующей оценкой приспособленности каждой особи на основе целевой функции. Далее применяется оператор отбора, реализующий принцип выживания наиболее приспособленных решений, после чего к отобранным

хромосомам (в области эволюционной оптимизации они называются родители) применяются операторы скрещивания и мутации, обеспечивающие рекомбинацию генетического материала и поддержание разнообразия в популяции. Сформированное таким образом новое поколение популяции решений (потомков) замещает предыдущее, после чего процесс эволюции продолжается до выполнения критерия останова, связанного с достижением требуемого качества решения или исчерпанием вычислительных ресурсов.

Особое место среди генетических алгоритмов занимает бинарный генетический алгоритм [44], в котором хромосомы представлены в виде бинарных строк, а каждый ген принимает значение 0 или 1. Такое кодирование является наиболее простым и исторически первым, что обусловило его широкое распространение и хорошую теоретическую изученность. Бинарные ГА отличаются простотой реализации, устойчивостью к шуму и универсальностью, поскольку бинарные представления могут использоваться для кодирования как дискретных переменных, так и вещественных с заданной точностью. Вместе с тем бинарное кодирование может приводить к увеличению длины хромосом и ограничению точности аппроксимации непрерывных переменных, что стимулировало развитие альтернативных схем кодирования, ориентированных на вещественные или гибридные представления.

Ключевым преимуществом генетических алгоритмов по сравнению с градиентными методами оптимизации является их независимость от аналитических свойств целевой функции. ГА не требуют вычисления производных и допускают работу с невыпуклыми, разрывными, зашумлёнными или частично определёнными функциями, что позволяет рассматривать оптимизируемую систему как «чёрный ящик». Это свойство делает генетические алгоритмы особенно востребованными в задачах настройки гиперпараметров сложных моделей, таких как XGBoost [45], а также при оптимизации архитектур нейронных сетей, где аналитическое выражение градиента либо отсутствует, либо не отражает глобальных свойств пространства решений. В совокупности данные особенности определяют генетические алгоритмы как эффективный и гибкий инструмент глобальной оптимизации в современных задачах машинного обучения и интеллектуального анализа данных.

Отличительной чертой ГА является баланс между исследованием (exploration) и использованием (exploitation) пространства решений за счет ме-

ханизмов кроссовера [46], мутации [47] и селекции [48]. Например, в методе NeuroEvolution of Augmenting Topologies (NEAT) [49] используется ранжирование решений по сложности, что позволяет постепенно усложнять архитектуру нейросети без преждевременной сходимости. Для учета ограничений ГА интегрируют методы штрафных функций, проекции допустимых решений и правила воспроизводства, исключающие недопустимые комбинации на этапе мутации. В работе [50] предложена классификация стратегий, включающих адаптивные штрафные коэффициенты и многокритериальную оптимизацию для балансировки между целевой функцией и ограничениями.

Высокая степень параллелизма ГА обусловлена независимостью оценки приспособленности особей популяции, что позволяет распределять вычисления на GPU-кластерах или облачных платформах [51]. Асинхронные островные модели дополнительно повышают эффективность за счет локальных селективных процессов и миграции решений между подпопуляциями, что критично для ресурсоемких задач автоматического машинного обучения (AutoML) [52].

Применение ГА в машинном обучении охватывает автоматический дизайн нейросетей (Neural Architecture Search, NAS) [53], где оптимизируются топология слоев, типы активации и соединений, выбор признаков (feature selection) [48] для минимизации размерности данных без потери информативности и ансамблевое обучение, где настраиваются веса базовых моделей для повышения обобщающей способности, и другие модели машинного обучения. Современные фреймворки, такие как AutoKeras [54], используют ГА для автоматической генерации пайплайнов предобработки и обучения, снижая порог входа в область AutoML.

Исследования генетических и эволюционных алгоритмов получили развитие в работах многих отечественных и зарубежных авторов. Одной из первых систематических работ в данной области является монография Дж. Холланда, в которой предложена концепция адаптивных систем и сформулированы основные принципы функционирования генетических алгоритмов, включая популяционное представление решений, операторы рекомбинации и мутации, а также механизм отбора [55].

Дальнейшее развитие генетических алгоритмов отражено в работах Д. Голдберга, где рассмотрены основные механизмы эволюционного поиска и показаны возможности применения бинарного кодирования для решения раз-

личных задач оптимизации [56]. В исследованиях К. Дэба предложены методы работы с ограничениями и подходы к решению задач многокритериальной оптимизации на основе генетических алгоритмов [57].

Параллельно развивались и другие направления эволюционных методов оптимизации. В частности, Дж. Кеннеди и Р. Эберхарт предложили алгоритм оптимизации роя частиц, основанный на моделировании коллективного поведения агентов [58]. Позднее была разработана бинарная версия этого метода, позволяющая применять его в дискретных пространствах поиска [59]. Аналогичные подходы рассматривались и для алгоритма дифференциальной эволюции, для которого были предложены модификации, ориентированные на задачи дискретной оптимизации [60].

Отдельное направление исследований связано с представлением решений и схемами кодирования в генетических алгоритмах. В работах Д. Уитни рассматриваются методы перекодирования решений и различные схемы представления генотипов [61]. В исследованиях Н. Шраудольфа и Р. Белью предложены подходы динамического кодирования, при которых структура генетического представления может изменяться в процессе эволюции [62].

В отечественной научной литературе вопросы разработки и применения генетических алгоритмов рассматриваются в работах В. М. Курейчика, В. В. Курейчика [63] и Л. А. Гладкова [64]. В работах А. П. Карпенко представлен обзор методов и направлений развития эволюционных вычислений [65]. В работах А. В. Еремеева рассматриваются вопросы о теоретической сходимости генетических алгоритмов [66]. Исследования научной школы Е. С. Семенкина посвящены самонастраивающимся и самоконфигурируемым эволюционным алгоритмам [67]. В работах Е. А. Сопова рассматриваются различные способы бинарного кодирования решений для задач оптимизации [68].

Развитие эволюционных методов для задач глобальной оптимизации связано не только с расширением набора операторов, но и с попытками повысить устойчивость поиска в условиях высокой размерности, многоэкстремальности и отсутствия аналитической информации о целевой функции. Сравнительный анализ популяционных методов глобальной оптимизации, включающий генетические алгоритмы, эволюционные стратегии, дифференциальную эволюцию, алгоритм роя частиц и другие метаэвристики, показывает, что эффективность таких методов существенно зависит от структуры решаемой задачи и выбран-

ной схемы управления популяцией [69]. При этом одинаковое число вычислений целевой функции не гарантирует сопоставимого качества результатов для разных алгоритмов, поскольку отдельные методы по-разному используют информацию о распределении решений в пространстве поиска. В задачах типа “черного ящика” свойства целевой функции оцениваются только по результатам её вычислений.

Одним из направлений повышения эффективности эволюционного поиска является использование кооперативных и многоагентных схем. В работе [70] рассматриваются кооперативные алгоритмы глобальной оптимизации на основе генетического алгоритма, эволюционной стратегии и роя частиц, где взаимодействие реализуется через островные и коэволюционные механизмы. Такие подходы позволяют повысить надежность и скорость поиска по сравнению с одиночными алгоритмами, поскольку несколько подпопуляций исследуют разные области пространства решений и обмениваются найденной информацией. Сходная идея развивается в многоцелевых кооперативных ГА с параллельной реализацией, где островная модель объединяет несколько различных эволюционных схем и снижает зависимость результата от выбора единственного базового алгоритма [71]. Вместе с тем данные методы в основном перераспределяют вычислительные ресурсы между подпопуляциями и алгоритмическими стратегиями, но не решают напрямую задачу адаптивного изменения самой области поиска. Дополнительно практическое применение данной идеи показано в задаче генерации естественно-языковых объяснений предупреждений статического анализа, где генетический алгоритм с динамической коррекцией области поиска использовался для настройки гиперпараметров модели, а переход к многоцелевой оптимизации позволил учитывать одновременно качество и достоверность формируемых объяснений [72].

В смежных исследованиях развитие популяционных методов глобальной оптимизации также связывается с повышением адаптивности алгоритмов и более осмысленным управлением процессом поиска. В частности, в работе [73] рассматриваются методы интеллектуализации популяционных алгоритмов, включая параметрическую, структурную и структурно-параметрическую адаптацию, что подчеркивает важность автоматической настройки механизмов поиска под свойства решаемой задачи. Близкая по смыслу идея представлена в [74], где предлагается способ декомпозиции области поиска для задач глобальной опти-

мизации, ориентированный на повышение диверсификации и эффективности параллельных популяционных алгоритмов. Дополнительно в работе [75] предложен комплексный подход к анализу популяционных алгоритмов с учетом точности, скорости и универсальности их применения, что важно для объективного сравнения модифицированных эвристических методов. Применение гибридных эволюционных схем для решения прикладных оптимизационных задач также показано в [76], где исследуется сочетание роя частиц и сеточного поиска, подтверждающее актуальность объединения глобальных и локальных механизмов исследования пространства решений.

В задачах с бинарным представлением статистическая информация, накапливаемая популяцией в процессе эволюции, может использоваться для анализа состояния поиска. В работе [77] показано, что генетический алгоритм с бинарным представлением фактически собирает информацию о пространстве поиска в виде распределения единичных значений по позициям хромосомы. На этой основе рассматривается метод прогнозирования сходимости для ГА и RBIL-подобных алгоритмов. Данный подход важен тем, что переводит поведение бинарного ГА из полностью неявной формы в статистически интерпретируемую: состояние популяции становится источником информации о перспективности отдельных направлений поиска. Однако такая статистика используется главным образом для оценки сходимости, а не для перестройки границ или структуры поискового пространства.

Близкое направление представлено гибридными бинарными GA-EDA алгоритмами для сложных задач “черного ящика” [78; 79]. В этих работах подчеркивается, что классический ГА реализует стратегию проб и ошибок, сохраняя опыт поиска неявно в генах популяции, тогда как алгоритмы оценки распределений используют явную вероятностную модель перспективных решений. Совмещение этих подходов позволяет повысить информативность поиска в бинарных пространствах и перейти от чисто операторной эволюции к использованию статистической модели популяции. Для данной диссертации это особенно существенно, поскольку динамическая коррекция области поиска также опирается на идею извлечения структурной информации из текущего распределения решений, однако использует её не только для генерации новых хромосом, но и для изменения допустимой области переменных.

Отдельную группу составляют работы, посвященные крупномасштабной бинарной глобальной оптимизации. В [80; 81] рассматриваются EDA-ориентированные подходы к декомпозиции задач большой размерности с бинарным представлением. Основная идея заключается в том, что при большом числе переменных прямой поиск в полном пространстве становится неэффективным, поэтому требуется выявление структуры задачи и разбиение переменных на относительно независимые или слабо связанные подгруппы. Это направление близко к рассматриваемой диссертационной задаче, поскольку и декомпозиция, и кластерная коррекция области поиска направлены на снижение сложности оптимизации. Однако в EDA-декомпозиции основной акцент делается на группировке переменных и построении вероятностных моделей, тогда как проблема адаптивного изменения границ вещественной части смешанного пространства остается недостаточно разработанным.

Существенное развитие получили самоконфигурируемые генетические алгоритмы, в которых параметры и операторы подбираются автоматически в процессе работы. В [82] предложен самоконфигурируемый ГА с модифицированным оператором равномерного скрещивания, вводящим селективное давление уже на стадии рекомбинации. Такой подход позволяет не задавать вручную фиксированные вероятности операторов, а изменять их использование в зависимости от успешности. В прикладных задачах с алгоритмически заданными функциями смешанных переменных самоконфигурируемый ГА также показал эффективность за счет отсутствия необходимости ручной настройки параметров [83]. Эти результаты подтверждают целесообразность автоматизации управления эволюционным процессом, однако в них самоконфигурация в основном относится к операторам и управляющим параметрам, тогда как границы пространства поиска предполагаются заданными заранее.

Идея адаптации параметров получила дальнейшее развитие в генетическом алгоритме с механизмом success-history based parameter adaptation [84]. В этом подходе значения параметров, приводившие к улучшению решений, сохраняются в истории и используются для генерации новых параметров на последующих итерациях. Первоначально подобные механизмы были характерны для дифференциальной эволюции, однако их перенос в ГА позволил повысить эффективность как для бинарных, так и для непрерывных задач. Работа важна тем, что показывает возможность использования успешного опыта поиска как

источника адаптации. Вместе с тем объектом адаптации остаются параметры алгоритма, а не структура кодируемого пространства. Поэтому даже при успешной настройке вероятностей и интенсивности операторов алгоритм продолжает работать внутри исходных фиксированных границ.

В дифференциальной эволюции проблема адаптации параметров также является одной из центральных. В работе [85] для анализа адаптации параметров предложена метрика *expected fitness improvement*, позволяющая визуализировать пространство значений управляющих параметров DE и оценивать связь между масштабным коэффициентом, расстояниями между индивидами и ожидаемым улучшением приспособленности. Тем самым параметры алгоритма рассматриваются не как фиксированные константы, а как динамические величины, зависящие от состояния популяции. В дальнейшем эта линия была развита в направлении автоматического проектирования схем адаптации параметров для дифференциальной эволюции с помощью генетического программирования [86]. Такие гиперэвристические методы позволяют автоматически синтезировать правила управления параметрами, но сохраняют общую особенность большинства современных подходов: они изменяют поведение алгоритма внутри заданного пространства, не выполняя явной коррекции самой области поиска.

Более общий подход к автоматическому синтезу компонентов эволюционных алгоритмов представлен в работах по гиперэвристикам и генетическому программированию. В [79] генетическое программирование используется для автоматического синтеза операторов селекции в генетических алгоритмах. В работе [87] рассматривается проектирование *black-box* оптимизаторов с использованием *PushGP*, где программы генерируют операции над векторами решений и фактически формируют новые популяционные алгоритмы численной оптимизации. Аналогичная идея автоматического улучшения динамических оптимизаторов применяется в [88], где генетическое программирование используется для генерации новых эвристик локального поиска. Эти исследования демонстрируют общий тренд к автоматизации проектирования оптимизационных алгоритмов. Однако автоматическая генерация операторов или правил обновления не устраняет полностью проблему выбора области поиска: даже новый оператор работает с тем представлением и теми границами, которые заданы внешне.

Для многоэкстремальных задач особое значение имеют методы, ориентированные на поддержание нескольких перспективных областей поиска. В работе [89] предложен самоконфигурируемый многостратегийный мультимодальный ГА, предназначенный для задач с бинарным представлением, где традиционные методы ниширования часто оказываются недостаточно эффективными. Дальнейшее развитие этой идеи связано с ансамблем мультимодальных генетических алгоритмов [90], где управление взаимодействием нескольких поисковых стратегий позволяет повысить устойчивость поиска в сложных ландшафтах. Эти работы особенно близки к задаче динамической коррекции области поиска, поскольку предполагают наличие нескольких областей притяжения и необходимость распределять внимание алгоритма между ними. Тем не менее мультимодальные схемы главным образом поддерживают разнообразие популяции и выделение нескольких решений, но не всегда обеспечивают явную перестройку дискретизации и границ переменных в смешанном пространстве.

Кластеризация и декомпозиция также используются в прикладных задачах интеллектуального анализа данных и машинного обучения. Так, в работе [91] предложено снижение размерности для текстовой классификации на основе иерархической агломеративной кластеризации терминов и оптимизации весов кластеров с использованием коэволюционного генетического алгоритма. Этот пример показывает, что кластерный анализ может выступать не только инструментом постобработки, но и частью оптимизационной процедуры, уменьшающей эффективную размерность задачи. Для диссертационной работы данный результат важен как подтверждение того, что кластерная структура данных или решений может использоваться для перенаправления поиска и сокращения избыточных вычислений. Однако в указанной работе кластеризация применяется к признаковому пространству задачи классификации, а не к текущей популяции оптимизационного алгоритма с целью динамической коррекции области поиска.

Дополнительное обоснование необходимости учитывать структуру поискового ландшафта дают работы, исследующие распределение локальных оптимумов. В [92; 93] анализируется задача распределения буферов в производственных линиях с ненадежными машинами с точки зрения фитнес-ландшафта и поведения генетических алгоритмов. Отмечается, что в ряде экземпляров можно выделить несколько кластеров локальных оптимумов, а структура типа

“big valley” проявляется лишь частично. Это важно для задач глобальной оптимизации в целом: если локальные оптимумы образуют несколько групп, то равномерный поиск по фиксированной области может приводить либо к преждевременной концентрации в одной группе, либо к чрезмерному расходованию вычислений на малоперспективные зоны. Следовательно, алгоритму требуется механизм, способный учитывать кластерную структуру найденных решений и перераспределять поисковую активность между перспективными областями.

Теоретические исследования генетических алгоритмов также подтверждают значимость параметров отбора, мутации и структуры популяции. В работах [94; 95] развита техника level-based analysis, позволяющая получать оценки времени работы для широкого класса генетических алгоритмов и поисковых процессов. В [96; 97] рассматриваются оценки времени достижения локальных и глобальных оптимумов для генетических алгоритмов с высоким селективным давлением, а в [98] анализируется обобщение мутации с тяжелыми хвостами для варианта $(1 + (\lambda, \lambda))$ ГА. Эти результаты важны для понимания того, что эффективность ГА определяется не только фактом популяционного поиска, но и балансом между селективным давлением, вероятностью крупных изменений и способностью алгоритма покидать локальные области. Однако теоретические оценки, как правило, строятся для абстрактных тестовых функций или фиксированных пространств, тогда как в прикладных смешанных задачах существенную роль играет качество исходной дискретизации и выбор границ переменных.

Таким образом, анализ релевантных работ показывает, что современные эволюционные алгоритмы развиваются по нескольким взаимосвязанным направлениям: самоконфигурирование операторов, адаптация параметров на основе истории успешных шагов, кооперативные и островные модели, вероятностное моделирование распределения решений, декомпозиция переменных, мультимодальный поиск и автоматический синтез эвристик. Все эти подходы повышают устойчивость и эффективность оптимизации, однако большинство из них сохраняет фиксированную постановку пространства поиска. Даже когда состояние популяции используется явно, оно чаще служит для выбора операторов, настройки параметров, оценки сходимости или построения вероятностной модели новых решений. Вопрос динамической коррекции самой области поиска, а именно в задачах со смешанными переменными и бинарным кодированием вещественных параметров, остается недостаточно разработанным.

В связи с этим перспективным направлением является интеграция кластерного анализа популяции с механизмом адаптивного изменения границ и дискретизации поискового пространства. В отличие от методов, которые изменяют только параметры эволюционного процесса, такой подход позволяет использовать накопленную статистическую информацию о распределении решений для выявления перспективных областей, исключения заведомо слабых регионов и более детального исследования зон с индивидами, обладающими высокой приспособленностью. Это обеспечивает переход от самонастройки алгоритма к самонастройке представления задачи, что важно для бинарных ГА в смешанной глобальной оптимизации, где эффективность поиска существенно зависит от того, насколько выбранная область и точность кодирования соответствуют реальной структуре целевой функции.

Современные генетические алгоритмы за последние десятилетия претерпели существенные изменения, превратившись из универсальной поисковой эвристики в обширное семейство специализированных и гибридных методов [99]. Одним из ключевых направлений развития стало объединение ГА с локальными методами оптимизации, что привело к появлению меметических алгоритмов [100]. В таких подходах глобальный стохастический поиск, присущий ГА, дополняется локальным уточнением решений, что позволяет эффективно преодолевать локальные минимумы. Аналогичная идея лежит в основе гибридизации ГА с табу-поиском [101] и градиентными методами [102], где использование локальной информации ускоряет сходимость и повышает точность найденных решений. Однако данные подходы, как правило, ориентированы на конкретные классы задач и требуют ручной настройки взаимодействия между компонентами.

Другим направлением стало развитие механизмов адаптивной настройки параметров эволюции. В работе [67] показано, что динамическое управление вероятностями скрещивания, мутации, а также стратегией селекции позволяет повысить устойчивость алгоритма при изменении ландшафта целевой функции. В отличие от классических ГА с фиксированными параметрами, адаптивные схемы способны автоматически подстраиваться под фазу поиска: усиливать исследование на ранних этапах и концентрироваться вблизи оптимума на заключительных этапах поиска. Тем не менее, даже в этих методах адаптация за-

трагивает в основном операторы эволюции, но не структуру самого поискового пространства.

Существенное развитие получили и суррогатные методы, направленные на снижение вычислительной сложности задач [103]. Аппроксимационные модели, регрессионные и нейросетевые суррогаты позволяют частично заменить дорогостоящие вычисления целевой функции, сохраняя при этом приемлемую точность. Также активно развиваются параллельные и распределённые модели ГА [104], включающие островные схемы, миграционные механизмы и асинхронные обновления. Эти подходы позволяют масштабировать поиск и использовать современные вычислительные архитектуры, однако они также не затрагивают вопрос адаптации границ пространства поиска.

Отдельную группу составляют методы, направленные на поддержание разнообразия популяции. Ниширование [105], фитнес-шеринг [106], механизмы crowding [107] и коэволюционные схемы [108] позволяют формировать в популяции несколько устойчивых подгрупп, каждая из которых исследует собственную область пространства решений. Это снижает вероятность преждевременной сходимости и повышает шанс обнаружения глобального оптимума. Аналогичным образом развивались методы работы с ограничениями, включая адаптивные штрафы [109], ремонтные операторы [110] и многопопуляционные стратегии [111]. Все эти подходы повышают качество поиска в сложных допустимых областях, однако они оперируют уже заданными границами пространства.

В специализированных областях также получили развитие квантово-вдохновлённые ГА [112], а также алгоритмы, ориентированные на статистические задачи, например, оценку параметров распределений [113]. Эти методы используются для конкретных приложений и плохо применимы к другим классами задач.

Алгоритм SelfCSHAGA представляет собой самоконфигурируемую версию генетического алгоритма, в которой объединяются два механизма адаптации: настройка численных параметров на основе истории успешных применений и динамический выбор генетических операторов. Такой подход позволяет алгоритму изменять как интенсивность мутации и скрещивания, так и используемые операторы эволюции в процессе поиска решения [114].

Основой алгоритма является модифицированная версия SHAGA, в которой параметры мутации и скрещивания адаптируются на основе истории успеш-

ных применений. После каждого поколения сохраняются значения параметров, при которых были получены улучшения решений, и на их основе формируются распределения для генерации параметров на последующих итерациях. Это позволяет регулировать интенсивность вариационных операторов в зависимости от стадии поиска и структуры исследуемой задачи [115].

Дополнительно в алгоритме используется механизм самоконфигурирования генетических операторов. Для формирования потомков применяется набор альтернативных операторов селекции и скрещивания. Вероятности их использования корректируются в процессе работы алгоритма в зависимости от успешности применения. Если оператор чаще приводит к улучшению решений, вероятность его выбора увеличивается, тогда как менее эффективные операторы постепенно используются реже [116].

Алгоритм поддерживает несколько вариантов операторов скрещивания, включая одноточечное, двухточечное и различные варианты равномерного скрещивания с селективным давлением, а также многородительские схемы. В таких схемах вероятность наследования гена зависит от значений функции пригодности родителей, что позволяет учитывать селективное давление непосредственно на этапе скрещивания [114]. Кроме того, использование нескольких родителей позволяет увеличить разнообразие генетического материала в популяции.

Работа алгоритма включает инициализацию популяции и начальных значений параметров, генерацию потомков с адаптивными параметрами, обновление истории успешных применений и корректировку вероятностей выбора операторов. В результате алгоритм одновременно адаптирует параметры и конфигурацию операторов, что позволяет применять его для задач оптимизации с вещественными и булевыми переменными.

На этом фоне интерес представляют немногочисленные работы, в которых предприняты попытки динамически изменять само поисковое пространство в процессе эволюции. В частности, в работе [117] область поиска сужается по мере уточнения решений, а в методе GAASS [118] границы корректируются на основе характеристик текущей популяции и нормализованного расстояния по фитнесу. Несмотря на положительные результаты, подобные подходы требуют детальной настройки под конкретную физическую модель, что существенно ограничивает их универсальность.

Таким образом, несмотря на большое разнообразие современных модификаций генетических алгоритмов, в подавляющем большинстве работ область поиска остаётся фиксированной на всём протяжении эволюционного процесса. На практике это приводит к двум принципиальным ограничениям. С одной стороны, чрезмерно широкая область поиска может приводить к неэффективному расходованию вычислительных ресурсов на заведомо неперспективные области. С другой стороны, слишком узкие границы могут исключать глобально оптимальные решения и приводить к систематическому смещению результатов. Фактически успех оптимизации в немалой степени зависит от априорного ручного выбора границ проектных переменных.

Указанные обстоятельства обуславливают необходимость разработки универсальной процедуры динамической коррекции области поиска, которая могла бы адаптироваться к структуре текущей популяции непосредственно в процессе эволюции, не привязываясь к конкретной предметной области. Такой подход позволяет одновременно ускорять сходимость, снижать риск преждевременной стагнации и повышать надёжность поиска глобального оптимума в задачах инженерной оптимизации с жёсткими ограничениями.

Дифференциальная эволюция представляет собой эффективный стохастический метод оптимизации для непрерывных пространств, в котором новые решения формируются на основе масштабированных разностей между случайными индивидами популяции [119]. Такой механизм обеспечивает направленный поиск и позволяет ДЭ успешно решать задачи высокой размерности. Для повышения эффективности в ДЭ активно используются методы противоположного обучения (Opposition-Based Learning, OBL), расширяющие исследуемое пространство за счёт одновременного анализа текущих и противоположных решений [120]. Одним из таких подходов является метод BetaCOBL, в котором противоположные решения формируются на основе бета-распределения, обеспечивая баланс между глобальным поиском и локальной эксплуатацией.

Однако в базовой версии BetaCOBL границы поискового пространства остаются фиксированными, что ограничивает способность алгоритма адаптироваться на поздних этапах сходимости. Для устранения этого недостатка в работе [121] предложена адаптивная модификация ABetaCOBL, в которой границы поиска корректируются в зависимости от распределения популяции, что позволяет концентрировать поиск в перспективных областях.

Другим направлением динамической адаптации пространства поиска является структура ARSA [122], предназначенная для предотвращения стагнации в алгоритмах ДЭ. В данном подходе при отсутствии улучшений активируется механизм регенерации, формирующий новые решения в автоматически корректируемой области поиска. Эффективность ARSA подтверждена на тестовых задачах IEEE CEC 2017 [123] и ряде прикладных задач.

Дополнительно в работах по ДЭ активно применяются кластерные механизмы, в частности метод кластерного выбора родителей CMS [124], который формирует пары для скрещивания внутри кластеров популяции. Эксперименты на тестовых функциях CEC 2005 [125] показали, что такой подход повышает устойчивость сходимости и качество решений. Тем не менее, даже в этих модификациях кластеризация используется в основном для отбора родителей и поддержания разнообразия, а не для систематической коррекции границ поискового пространства.

Развитие дифференциальной эволюции во многом связано с разработкой механизмов автоматической настройки управляющих параметров. В классической DE эффективность алгоритма зависит от коэффициента масштабирования F , вероятности кроссовера CR и размера популяции. При фиксированных значениях этих параметров алгоритм может демонстрировать различную эффективность на разных классах задач. По этой причине в последующих работах были предложены методы адаптации параметров на основе информации, получаемой в ходе поиска [126].

Одним из алгоритмов этого направления является JADE [127]. В нём используется мутационная схема *current-to-pbest*, внешний архив и адаптация параметров по успешным потомкам. Архив содержит вытесненные решения и используется при формировании разностных векторов. Значения параметров F и CR генерируются на основе статистики тех комбинаций, которые привели к улучшению особей.

Идея адаптации параметров на основе истории успеха была развита в алгоритме SHADE [128]. В алгоритме поддерживается память исторически успешных значений параметров. Для каждого индивида параметры выбираются на основе случайного элемента этой памяти, после чего генерируются новые значения F и CR . После завершения поколения память обновляется с учётом параметров, при которых произошло улучшение.

В алгоритме L-SHADE механизм success-history adaptation дополнен процедурой линейного уменьшения размера популяции [129]. Размер популяции постепенно уменьшается в ходе работы алгоритма от начального значения к меньшему конечному. При большем размере популяции на ранних этапах поиска увеличивается разнообразие решений, тогда как на поздних этапах уменьшение популяции приводит к более концентрированному поиску.

После появления L-SHADE были предложены алгоритмы, [130] в которых исследуются различные способы адаптации параметров и управления популяцией [131]. В этих работах рассматриваются варианты обновления архива, изменения селективного давления и модификации правил генерации параметров.

В работах В. В. Становова и соавторов рассматриваются модификации алгоритмов семейства SHADE. В алгоритмах NL-SHADE-RSP [132] и NL-SHADE-LBC [133] используются механизмы смещённой адаптации параметров и модифицированные правила изменения селективного давления. В этих алгоритмах также применяется уменьшение размера популяции и использование архива решений.

В алгоритме L-SRTDE адаптация параметров выполняется с учётом показателя success rate, который характеризует долю успешных улучшений в текущем поколении [134]. Этот показатель используется при генерации параметров мутации и кроссовера. Алгоритм также использует уменьшение размера популяции и внешний архив.

Современные варианты дифференциальной эволюции обычно включают несколько механизмов адаптации. К ним относятся настройка параметров F и CR на основе истории успешных шагов, изменение размера популяции, использование архивов решений [135] и модификации мутационных стратегий [136].

Метод роя частиц (Particle Swarm Optimization, PSO) относится к классу популяционных стохастических алгоритмов оптимизации и был предложен Кеннеди и Эберхартом [137]. В алгоритме используется популяция частиц, каждая из которых представляет возможное решение задачи. Частицы перемещаются в пространстве поиска, изменяя своё положение на основе собственного опыта и информации, получаемой от других частиц роя. Каждая частица запоминает наилучшее найденное ею положение (personal best), а также использует информацию о наилучшем положении в популяции или в локальной окрестности. На

каждой итерации обновляются скорость частицы и её положение, что обеспечивает совместное исследование пространства решений.

В дальнейшем было предложено большое количество модификаций PSO. Они связаны с различными схемами топологии роя, изменением параметров ускорения и инерции [138], а также с адаптацией параметров алгоритма [139]. Отдельным направлением стало развитие дискретных и бинарных вариантов алгоритма, в которых положение частицы интерпретируется как бинарный вектор решений. В бинарном PSO изменение положения осуществляется через вероятностное преобразование скорости, обычно с использованием сигмоидальной функции [59]. Такие варианты алгоритма применяются в задачах комбинаторной оптимизации и выбора признаков.

Одной из модификаций алгоритма является Fully Informed Particle Swarm Optimization (FIPSO) [140]. В классическом PSO частица использует информацию только о своём личном лучшем положении и лучшем положении в окрестности. В FIPSO информация о направлении движения формируется на основе всех соседних частиц в заданной топологии роя. Скорость частицы определяется с учётом вкладов от нескольких лучших положений соседей, а не одного глобального или локального лидера. Такой подход позволяет использовать больше информации о структуре популяции и снижает зависимость поиска от одного лидера.

В Fully Informed PSO вклад каждого соседа обычно взвешивается, а итоговое направление движения формируется как сумма влияний всех соседних частиц. Это приводит к более равномерному распространению информации по популяции и может улучшать исследование пространства решений. В ряде исследований отмечается, что такие схемы позволяют уменьшить вероятность преждевременной сходимости и повысить устойчивость алгоритма при решении сложных оптимизационных задач [141].

Таким образом, на основе проведенного обзора можно сделать вывод, что исследователи делают упор на повышении эффективности алгоритмов оптимизации, а методы работы с пространством поиска изучены недостаточно хорошо.

1.3 Модификации эволюционных алгоритмов с использованием кластерного анализа

Существует ряд исследований, посвященных анализу поведения ГА в задачах высокой размерности и при наличии ограничений [142; 143]. В этих работах предлагаются адаптивные механизмы изменения области поиска и кластерно-ориентированные эволюционные операторы [144; 145], которые изменяют траекторию эволюции популяции в процессе оптимизации. Показано, что статические границы области поиска могут существенно снижать эффективность ГА в тех случаях, когда допустимая или перспективная область составляет лишь малую часть исходного пространства. В частности, Amirjanov и Sadikoglu [146] демонстрируют, что линейная коррекция области поиска уменьшает вычислительные затраты и ускоряет эволюционный процесс, подтверждая целесообразность адаптации границ в ходе оптимизации. Аналогичные выводы получены в работе Banerjee [147], где рекурсивная корректировка диапазонов переменных направляет популяцию в перспективные области пространства решений. Сходные закономерности выявлены и в задачах эволюционной кластеризации: ГА, как самостоятельно, так и в комбинации с алгоритмом k -средних [148], способны автоматически выделять качественные разбиения многомерных данных при неизвестном числе кластеров.

Дополняющие подходы ориентированы на адаптивное расширение области поиска при необходимости. Так, для генетических алгоритмов, работающих с непрерывными переменными, предложена стратегия расширения границ как средство борьбы с преждевременной сходимостью. В этом подходе область поиска динамически увеличивается вокруг перспективных решений, что позволяет алгоритму выходить из состояния стагнации и сохранять глобальные поисковые свойства [149]. Подобные стратегии демонстрируют, что динамическое управление областью поиска — как в сторону сужения, так и в сторону расширения — играет ключевую роль в балансировке эксплуатации и исследования.

Отдельное направление связано с самонастройкой параметров и кластеризацией. В работе Valamirugan и соавторов [150] предложен адаптивный генетический алгоритм с гибридным представлением решений, самонастраивающимися параметрами [151] и возможностью одновременной работы с дискретными и

непрерывными переменными. Аналогично, Zhang и соавторы [152] рассматривают адаптацию операторов скрещивания и мутации на основе кластеризации популяции: вероятности применения операторов корректируются с учётом относительного качества выделенных групп решений. Тем самым структура популяции используется для ускорения сходимости при одновременном сохранении разнообразия. Эти результаты подтверждают, что кластерная информация является информативным индикатором структуры поискового ландшафта и может служить основой для селективной интенсификации.

Адаптация области поиска исследовалась также в задачах с ограничениями. В модифицированных вещественно-кодируемых ГА, использующих эвристики сокращения диапазонов, показано повышение эффективности на инженерных задачах за счёт последовательного уточнения допустимой области по мере приближения к границам ограничений [153]. Другие подходы задействуют пространственные структуры представления, такие как деревья бинарного разбиения пространства, позволяющие избегать повторного посещения областей с низкой полезностью и направлять алгоритм в ранее не обследованные подпространства. Например, Shang и соавторы [154] представляют метод эволюционного поиска архитектур нейронных сетей [53], в котором используются адаптивное управление параметрами, анализ вклада отдельных генов, мутация с учётом их ценности и возрастная селекция. Это позволяет уменьшить риск преждевременной сходимости и получить архитектуры, превосходящие как ручные, так и эволюционные базовые решения. Подобные методы отражают общий тренд к повышению осведомлённости алгоритмов о структуре пространства поиска.

Близким по идейной направленности, но технически отличным направлением являются механизмы неповторного посещения точек в ГА. В работе Su и соавторов [155] показано, что классические ГА в задачах большой размерности часто повторно посещают ранее исследованные области, что приводит к неэффективному расходованию вычислительных ресурсов и увеличивает риск преждевременной сходимости. Предложенный авторами механизм основан на использовании дерева бинарного разбиения пространства (BSP) [156], в котором сохраняются все оцененные решения, а новые особи проверяются на повторяемость. Дополнительно используется стратегия обучения на основе BSP, направляющая поиск в перспективные области. Эксперименты на тестовых функциях с вещественными и бинарными переменными показали ускорение сходимости и

рост разнообразия популяции. Тем самым продемонстрировано, что явная пространственная память позволяет эффективно концентрировать вычисления на неизученных областях.

Отдельное направление связано с анализом операторов скрещивания в генетических алгоритмах для задач непрерывной оптимизации с ограничениями. Thakur и соавторы [157] предложили модификацию ГА на основе схемы LX-PM, объединяющей лапласовский кроссовер [158] и степенную мутацию [159]. При этом лапласовский оператор был заменён на ограниченный экспоненциальный кроссовер (ВЕХ) [160], обеспечивающий генерацию потомков в пределах допустимых границ переменных и повышенную концентрацию выборки вблизи родительских решений при сохранении глобального поиска. Экспериментальные результаты на двадцати пяти задачах с ограничениями показали, что схема ВЕХ-PM превосходит по устойчивости и доле допустимых решений как базовый LX-PM, так и ряд сопоставимых методов. В совокупности эти исследования отражают смещение акцента в сторону эволюционных подходов, использующих структуру популяции и адаптивное управление областью поиска, ориентированное на распределение перспективных решений.

В работе [161] предложен кластерный эволюционный алгоритм SLMEA (super-large-scale multi-objective optimization problems, SLMOPs) для решения сверхмасштабных многокритериальных задач оптимизации. Основная идея метода состоит в том, чтобы сократить эффективную размерность пространства поиска за счёт кластеризации переменных на основе бинарных представлений решений из архива. Для каждой переменной оценивается её разреженность как среднее значение соответствующей бинарной компоненты по архиву решений: если переменная почти всегда принимает значение ноль или единицу, она выделяется в отдельную группу. Переменные, которые не являются строго нулевыми или единичными, объединяются в группы по сходству их бинарных профилей с эталонной переменной, выбранной так, чтобы её оценка её разреженности была близка к половине. Степень сходства между переменными определяется по тому, насколько часто они принимают одинаковые или разные значения в наборе архивных решений.

При генерации потомков используется сжатое представление: вещественные и бинарные векторы родителей переводятся из исходного пространства высокой размерности в пространство из K групп, где каждая группа описыва-

ется усреднёнными характеристиками соответствующих переменных. Для вещественных компонент это усреднённые значения, а для бинарных – эмпирические вероятности единицы внутри группы. Затем решения в полном пространстве восстанавливаются на основе этих сгруппированных представлений, то есть всем переменным внутри группы присваиваются значения, полученные на уровне группы. Таким образом, эволюция частично ведётся в сильно уменьшенном пространстве, а затем результаты проецируются обратно в исходное пространство.

Два ключевых параметра: число групп K и доля потомков, порождаемых в сокращённом пространстве (ρ), адаптируются динамически. Коррекция этих параметров осуществляется на основе сравнения долей недоминируемых решений, полученных в сокращённом и исходном пространствах на текущей и предыдущей итерациях. Если в сжатом представлении удаётся генерировать больше качественных (недоминируемых) решений, алгоритм увеличивает долю работы в уменьшенном пространстве и соответственно подстраивает число групп. Если же преимущества сжатого представления снижаются, больше вычислительных ресурсов возвращается в исходное пространство.

Эксперименты [161] показывают, что SLMEA превосходит современные методы (NSGA-II, CCGDE3, MOEA/PSL) на задачах с числом переменных порядка 10^6 , обеспечивая меньшие значения показателей качества, таких как инвертированная генерационная дистанция (IGD) и гиперобъём (HV). За счёт реализации матричных операций на GPU время работы SLMEA снижается до примерно 10% от времени аналогов. Метод особенно эффективен для задач с разреженными оптимальными решениями, например при отборе признаков (наборы задач FS1–FS3) и при обучении нейронных сетей (NN1–NN3), где SLMEA позволяет находить решения с меньшей сложностью моделей и меньшей ошибкой.

В работе [162] предложен эволюционный алгоритм CLUMOEА для многокритериальной оптимизации, основанный на комбинации кластеризации и косинусной меры скученности решений. Для разделения популяции на группы используется алгоритм k -средних, что позволяет явно учитывать внутреннюю структуру множества решений и повышать как сходимость к фронту Парето, так и разнообразие популяции. Для оценки плотности решений применяется косинусная мера, которая учитывает относительное положение особей в про-

странстве значений целевых функций и их взаимные угловые расстояния, а не только линейные расстояния по отдельным осям. Это даёт более устойчивую оценку скученности в задачах с «кривыми» или неправильно сформированными фронтами.

Общая схема CLUMOEА включает несколько основных шагов. Сначала текущая популяция разбивается на K кластеров методом k -средних. Затем внутри кластеров выделяются недоминируемые решения и оценивается баланс между сходимостью и разнообразием. При обновлении популяции в неё приоритетно включаются решения, которые одновременно имеют хорошую аппроксимацию фронта Парето и расположены в менее плотных областях, что предотвращает «скупивание» особей в ограниченной зоне. Для сравнительной оценки качества решений используется инвертированная генерационная дистанция (IGD), вычисляемая как среднее расстояние от точек эталонного фронта Парето до ближайших точек полученного множества решений.

Экспериментальные исследования на тестовых задачах MOP1–MOP7 со сложной геометрией фронта Парето (в том числе невыпуклой и разрывной) показывают, что CLUMOEА обеспечивает меньшие значения IGD по сравнению с NSGA-II и MOEA/D-DE. Например, для задачи MOP1 уже после 20 поколений IGD для CLUMOEА примерно в несколько раз ниже, чем у NSGA-II и MOEA/D-DE. Кроме того, CLUMOEА демонстрирует более равномерное покрытие фронта Парето, особенно в задачах с многосвязными областями оптимума. Алгоритм удачно сочетает эффективность кластеризации и косинусной меры скученности, что позволяет при меньших вычислительных затратах достигать более точных и разнообразных наборов решений.

В работе [163] предложен эволюционный алгоритм, в котором кластеризация используется для анализа структуры пространства решений и повышения эффективности поиска глобального минимума в задачах молекулярного моделирования. На первом этапе авторы снижают размерность исходного пространства за счёт исключения избыточных межатомных расстояний: вместо исходного 136-мерного описания структуры рассматривается более компактное 45-мерное представление. Далее строится матрица попарных расстояний между структурами, на основе которой выполняется иерархическая кластеризация. В результате удаётся выделить 11 устойчивых кластеров, соответствующих различным геометрическим конфигурациям молекул. Анализ 800 неза-

висимых запусков эволюционного алгоритма показывает, как распределяются родительские структуры по кластерам на разных этапах поиска. Для поддержания разнообразия и предотвращения доминирования одной структуры авторы вводят механизм штрафования «перегруженных» кластеров: вероятность выбрать индивид из кластера уменьшается по мере роста его размера, так что малочисленные кластеры получают относительно больший шанс участия в создании поколения. Дополнительно используется выделение выбросов (outliers) — структур, которые не вписываются ни в один из крупных кластеров и могут указывать на новые, малоисследованные области пространства поиска. В совокупности эти приёмы позволяют повысить успешность нахождения глобального минимума с примерно 28% до 41%, что подчёркивает полезность кластерного анализа для направленного управления эволюционным поиском.

Несмотря на существенный прогресс, интеграция перечисленных механизмов в самоконфигурирующиеся ГА остаётся ограниченной. Современные самонастраивающиеся фреймворки, как правило, адаптируют параметры операторов или точность кодирования, но редко включают явную фильтрацию области поиска или её динамическое сужение. Это формирует исследовательский зазор, особенно актуальный для задач со смешанными переменными, где необходимо одновременно обеспечивать согласованность кодировок и соблюдение границ переменных. Интеграция самонастройки, кластеризации и динамической коррекции области поиска позволяет адаптировать в ходе выполнения как параметры алгоритма, так и структуру поискового пространства. Такой подход снижает избыточные вычисления и обеспечивает более точную концентрацию поиска в перспективных областях, расширяя возможности применения ГА к крупномасштабной задаче оптимизации "чёрного ящика" без изменения базового представления решений.

1.4 Выводы по главе 1

В первой главе представлены теоретические основы и современные подходы к решению задач смешанной глобальной оптимизации, описаны сложности их реализации в условиях высокой размерности, многоэкстремальности и

наличия ограничений. Показана актуальность применения эволюционных алгоритмов, в частности ГА, для структурно-параметрического синтеза моделей машинного обучения. Выполнен сравнительный анализ методов эволюционной оптимизации, выявлены ключевые свойства бинарных ГА: обработка разношкальных переменных, устойчивость к шуму, параллелизм и применимость к задачам «черного ящика».

Однако анализ как классических бинарных эволюционных алгоритмов, так и существующих модификаций, выявил существенный недостаток – фиксированная область поиска ограничивает эффективность алгоритма при решении сложных задач. Использование равномерной статической сетки при дискретизации приводит к равномерному исследованию всех регионов пространства поиска и, как следствие, к избыточным вычислениям в неперспективных областях и недостаточной точности поиска в перспективных регионах. Существующие модификации алгоритмов частично решают такую проблему через динамическое изменение параметров, но сталкиваются с высокими вычислительными затратами на оценку состояния популяции или отсутствием системного подхода к фокусировке поиска на перспективных регионах.

Для преодоления указанных недостатков перспективным направлением является разработка метода адаптации области поиска, основанного на кластеризации решений для выявления групп решений с высокой приспособленностью, что позволит сосредоточить поиск вблизи потенциальных глобальных оптимумов, а также на адаптивном сужении/расширении границ области поиска в зависимости от плотности популяции и скорости сходимости.

2. Метод динамической коррекции поискового пространства

2.1 Обоснование и разработка метода

Во многих задачах глобальной оптимизации целевая функция в многомерном пространстве поиска характеризуется сложным многомодальным ландшафтом с узкими локальными областями, содержащими оптимальные или близкие по значению локально-оптимальные решения. При фиксированном вычислительном бюджете использование классического бинарного генетического алгоритма с постоянными границами поиска и жесткой дискретизацией пространства для вещественных переменных может приводить к тому, что значительная часть итераций расходуется на исследование регионов пространства поиска с низкой перспективностью. При этом либо требуется слишком мелкий шаг дискретизации (что резко увеличивает длину хромосомы и размер пространства поиска), либо алгоритм не способен обнаружить «узкие» области оптимума, если они лежат на малой доле исходного пространства.

Основная гипотеза, выдвигаемая в настоящем исследовании, состоит в следующем. Если в процессе работы генетического алгоритма динамически выделять кластеры решений в вещественном подпространстве для части хромосомы, кодирующей вещественные переменные, вычислять статистику значений функции пригодности для этих кластеров и адаптивно перераспределять область поиска в пользу статистически более перспективных регионов [164], одновременно корректируя шаг дискретизации и схему бинарного кодирования, то может повыситься вероятность попадания в области с лучшими значениями целевой функции и уменьшится эффект преждевременной сходимости. Предполагается, что такая адаптация позволит более эффективно использовать вычислительные ресурсы по сравнению с классическим генетическим алгоритмом с фиксированным пространством поиска.

Из этой гипотезы следует набор конкретных требований к разрабатываемому методу. Во-первых, должен существовать механизм локализации поиска – область поиска должна сужаться в направлении более перспективных кластеров решений из популяции и игнорировать те регионы, в которых накоплено

мало решений или средняя пригодность явно ниже общей по популяции. Во-вторых, такая локализация не должна быть чрезмерной: алгоритм обязан сохранять альтернативные области поиска до тех пор, пока их статистические характеристики не станут явно хуже. Это позволяет избежать преждевременной потери возможных решений. В-третьих, при сужении области поиска необходимо адаптивно уменьшать шаг дискретизации, иначе даже находясь в узкой области, точки сетки могут не попадать в область глобального оптимума. Наконец, все изменения могут накладываться как надстройка над генетическим алгоритмом: существующие операторы селекции, кроссовера и мутации сохраняются без модификаций.

Указанные требования определяют структуру всего алгоритма. Кластеризация решений позволяет разбивать популяцию на регионы пространства поиска и рассматривать их статистически: не только по отдельным особям, но и по локальным областям. Исключение неэффективных кластеров реализует механизм отбора «областей поиска»: если в регионе мало решений и их пригодность ниже средней, то такой регион не использует вычислительный бюджет и полностью удаляется. Несмотря на то, что поиск концентрируется на перспективных регионах, алгоритм в будущем может вернуться в исключенные ранее регионы за счет механизма расширения границ кластера. Проверка возможности коррекции пространства позволяет формально определить, в какой момент сужение границ действительно оправдано, когда значительная часть исходного пространства не участвует в поиске, значит популяция локализуется в малом подмножестве, и границы можно адаптировать. На этапе корректировки границ принимается решение, нужно ли расширять или сужать границу кластера. Это оценивается по эффективности индивидов, находящихся на текущих границах: если на границе расположен индивид с высокой относительной эффективностью, граница расширяется в его сторону; если на границе стоит явно неэффективный индивид, граница смещается к ближайшему эффективному. Таким образом, направление изменения границ основано на статистике пригодности, а не выбирается случайно.

После изменения границ, в случае сохранения старой схемы кодирования, можно уменьшить размерность бинарного пространства, потому что не используются узлы сетки исключенных областей. Также возникает возможность пересчёта шага дискретизации и адаптации схемы бинарного кодирования. В отли-

чие от стандартного бинарного кодирования ГА, где шаг постоянен, в предлагаемой модификации он должен изменяться вместе с масштабом пространства. Суженные области получают более мелкий шаг, что даёт возможность точно описывать узкие области без увеличения длины хромосомы. Расширенные области наоборот получают крупный шаг, что предотвращает излишнюю детализацию в малозначимых направлениях. Все это реализуется как надстройка над стандартным бинарным кодированием и не требует изменения генетических операторов.

Формальной гарантии достижения глобального оптимума данный метод, как и большинство эвристических методов, не предоставляет. Однако ключевые элементы алгоритма основаны на статистически обоснованном перераспределении поискового ресурса: узлы сетки исключаются из регионов с плохими показателями, а в перспективных регионах используется более плотная сетка. Это означает, что для задач с узкими областями оптимума и ограниченным бюджетом вычислений, ожидаемая эффективность алгоритма может быть выше, чем у ГА с фиксированными границами.

Для наглядного представления работы генетического алгоритма на рисунке 2.1 представлена блок-схема его функционирования.

На каждом поколении работы генетического алгоритма после оценки значений целевой функции выполняется процедура оценки возможности сужения или расширения и последующая динамическая коррекция поискового пространства (изменение границ поискового пространства происходит не на каждом поколении). Процедуру оценки возможности сужения или расширения можно выполнять не на каждой итерации, например, в вычислительно сложных задачах; периодичность запуска является гиперпараметром. Общая схема алгоритма показана на рисунке 2.2. Сначала осуществляется кластеризация решений (индивидов) в области поиска. Далее малые кластеры, содержащие недостаточное число индивидов, исключаются. Затем границы кластеров корректируются в зависимости от распределения эффективных и неэффективных решений. На следующем этапе обновляется структура поискового пространства с учетом сформированных областей интереса. В рамках этого процесса производится коррекция схемы бинарного кодирования индивидов для объединения разрозненных областей поиска. Завершающий этап включает удаление из популяции решений,



Рисунок 2.1 — Общий процесс работы генетического алгоритма с динамической коррекцией области поиска.

которые не попадают в новые границы поиска, после чего алгоритм переходит к формированию следующего поколения популяции решений (индивидов).

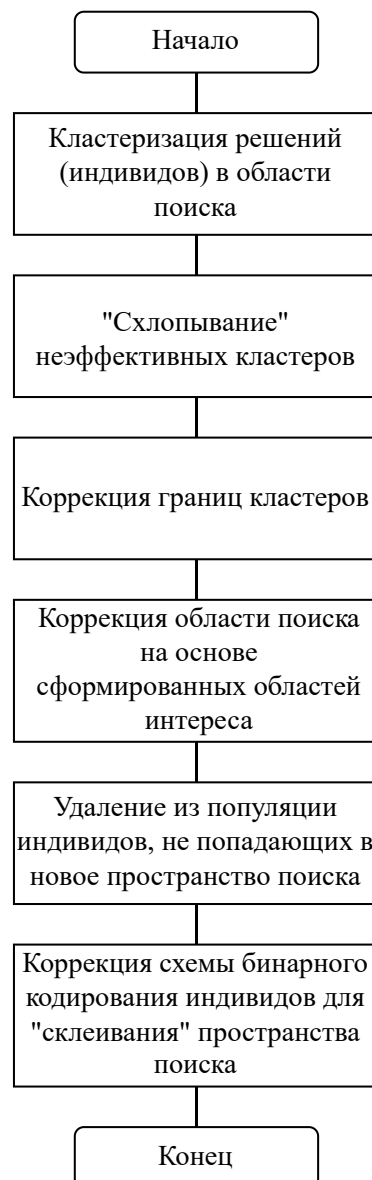


Рисунок 2.2 — Метод динамической коррекции поискового пространства на основе кластеризации решений.

Введем следующие обозначения:

G – общее число поколений ГА,

g – номер текущего поколения ГА,

M – общее количество индивидов в популяции,

m – номер рассматриваемого индивида,

K – общее число кластеров,

k – номер текущего кластера,

$D_{\vec{x}}^{g,k} = (D_{x_1}^{g,k}, \dots, D_{x_n}^{g,k})$ – область k -го кластера на поколении g в n -мерном пространстве поиска,

MS – минимальное количество точек в кластере (метод DBScan),

Ind_m^g – решение (индивид),

$Ind_m^{g,k}$ – m -ый индивид в поколении g в кластере k ,

$F_{Ind_m^g}$ – пригодность индивида Ind_m^g ,

M_k^g – количество индивидов в кластере k на поколении g ,

$IndSet^g$ – множество всех индивидов на поколении g ,

$IndSet_k^g$ – множество всех индивидов в кластере k на поколении g ,

$$IndSet^g = \bigcup_{k=1}^K IndSet_k^g, \quad (2.1)$$

$N_{IndSet_k^g}$ – число индивидов в кластере k на поколении g ,

$\bar{F}_{IndSet_k^g}$ – средняя пригодность индивидов в кластере k на поколении g .

Пространство поиска по переменной x_i на поколении g :

$$D_{x_i}^g = \bigcup_{k=1}^K D_{x_i}^{g,k}, \quad (2.2)$$

где $D_{x_i}^{g,k}$ – область поиска по переменной x_i на поколении g для кластера k . Если кластер убирается, то $D_{x_i}^{g,k} = \emptyset$.

В дальнейшем под x_i подразумеваются только вещественные переменные, для которых выполняется динамическая коррекция границ. Дискретные переменные не участвуют в кластеризации и коррекции границ напрямую, но передаются вместе с индивидом в целевую функцию.

Коррекция границ в общем случае осуществляется по формуле:

$$D_{x_i}^{g,k} = [x_i^{g,k,l} + \Delta_{x_i^{g,k,l}}, x_i^{g,k,h} + \Delta_{x_i^{g,k,h}}], \quad (2.3)$$

где $x_i^{g,k,l}$, $x_i^{g,k,h}$ – нижняя и верхняя граница по текущей переменной x_i для кластера k в поколении g , $\Delta_{x_i^{g,k,l}} \in \mathbb{R}$, $\Delta_{x_i^{g,k,h}} \in \mathbb{R}$ – это значения коррекции нижней и верхней границ соответственно.

Кластеризация решений. Кластеризация решений (индивидов) в области поиска выполняется одним из методов кластерного анализа [165]. Количество кластеров либо задается заранее (например, для K-Means, K-Medoids,

агломеративной кластеризации, спектральной кластеризации и т.д.), либо определяется автоматически (например, DBScan с минимальным числом точек в кластере и расстоянием или MeanShift с шириной полуядра и т.д.) [166].

На рисунке 2.3 показан пример кластеризации индивидов и выделения областей поиска в двумерной проекции вещественного пространства поиска (x_i, x_j) на поколении g .

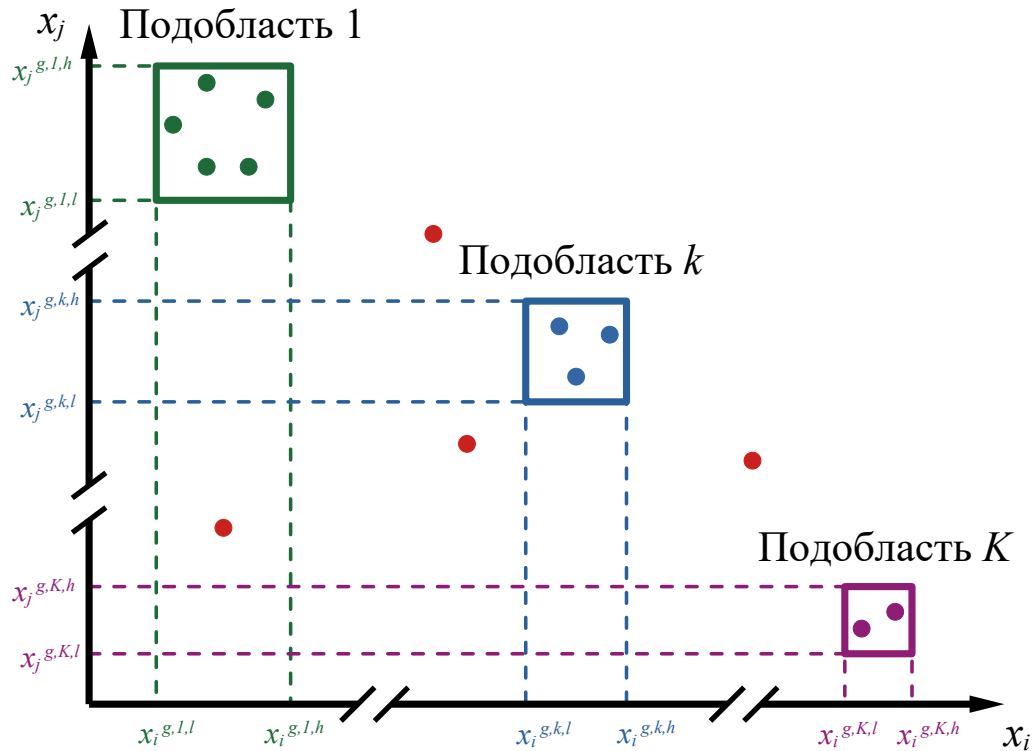


Рисунок 2.3 — Пример выделения областей поиска на основе K кластеров.

Каждый кластер $k = 1, \dots, K$ представлен совокупностью индивидов, близких по вещественным координатам, и аппроксимируется ограничивающим гиперпрямоугольником с нижними и верхними границами $x_i^{g,k,l}$, $x_i^{g,k,h}$ и $x_j^{g,k,l}$, $x_j^{g,k,h}$. На рисунке 2.3 сплошными линиями обозначены границы кластеров, определённые по текущему поколению, тогда как пунктирные линии отражают модифицированные границы области поиска, полученные в результате коррекции пространства относительно предыдущего поколения. Красными точками показаны отдельные индивиды, не входящие в кластеры, что иллюстрирует неоднородность распределения решений.

После кластеризации необходимо оценить, какие из обнаруженных областей пространства представляют интерес для дальнейшего исследования. Для

этого используется статистический анализ двух показателей: (1) размер кластера (число индивидов в кластере) и (2) средняя пригодность индивидов внутри кластера.

Малый размер кластера означает, что данный участок пространства либо редко достигается оператором генетического алгоритма, либо не является привлекательным с точки зрения поиска, поскольку популяция естественным образом не “накопилась” в этой области. В результате возникает статистически ненадёжная оценка качества области – выборка слишком мала, чтобы делать выводы о перспективности региона. Аналогично, если средняя пригодность кластера существенно ниже средней пригодности по всей популяции, то такой регион пространства проигрывает другим областям и тратить на него вычислительный ресурс нецелесообразно.

Оба условия: по размеру кластера и по пригодности, рассматриваются как гиперпараметры, задающие минимальный уровень статистической обоснованности признания области перспективной. В предлагаемой реализации используются сравнительные критерии:

$$N_{\text{IndSet}_k^g} < \bar{N}_{\text{IndSet}^g} \quad \text{и} \quad \bar{F}_{\text{IndSet}_k^g} < \bar{F}_{\text{Ind}^g}, \quad (2.4)$$

где $N_{\text{IndSet}_k^g}$ – число индивидов в k -м кластере на поколении g ; $\bar{N}_{\text{IndSet}^g}$ – среднее число индивидов по всем кластерам на поколении g ; $\bar{F}_{\text{IndSet}_k^g}$ – среднее значение функции пригодности индивидов в k -м кластере на поколении g ; \bar{F}_{Ind^g} – среднее значение функции пригодности по всей популяции на поколении g . Эти пороги не являются жёсткими правилами: при необходимости они могут быть модифицированы (например, заменены на квантиль распределения или долю от максимального значения), что позволяет адаптировать метод под конкретный класс задач.

Таким образом, удаляются только те кластеры, для которых одновременно выполняются два условия: (1) недостаточная статистическая репрезентативность и (2) низкое относительное качество решений. Это позволяет перераспределять вычислительный бюджет в пользу пространственных областей, демонстрирующих признаки локальных минимумов, сохраняя баланс между глобальным исследованием и локальным уточнением решений.

После проведения кластеризации и оценки границ кластеров в пространстве поиска предлагается проводить коррекцию пространства поиска в смысле

исключения индивидов в «неэффективных» кластерах [167]. При этом, эффективность оценивается в двух смыслах: количестве индивидов, попавших в кластер, а также отношении пригодности индивидов кластера к средней пригодности популяции. При этом, если кластер оказался «неэффективным», то область $D_{x_i}^{g,k}$ полностью исключается для всех $i = \overline{1, n}$.

Эффективность кластера оценивается по двум критериям: (1) количеству индивидов, попавших в кластер, и (2) отношению пригодности индивидов кластера к общей пригодности популяции. При признании кластера «неэффективным» из популяции удаляются все индивиды, принадлежащие этому кластеру.

Пошаговое представление процедуры:

Шаг 1. Рассчитать $N_{\text{IndSet}_k^g}$ для всех $k = \overline{1, K}$.

Шаг 2. Рассчитать среднее число индивидов по всем кластерам $\overline{N}_{\text{IndSet}^g} = \frac{1}{K} \sum_{k=1}^K N_{\text{IndSet}_k^g}$.

Шаг 3. Расчет средней пригодности по популяции $\overline{F}_{\text{Ind}^g} = \frac{1}{M} \sum_{m=1}^M F_{\text{Ind}_m^g}$.

Шаг 4. Установить $k = 1$.

Шаг 5. Если $N_{\text{IndSet}_k^g} < \overline{N}_{\text{IndSet}^g}$, то перейти к шагу 5, иначе – к шагу 7.

Шаг 6. Рассчитать среднюю пригодность индивидов в кластере $\overline{F}_{\text{IndSet}_k^g} = \frac{1}{M_k^g} \sum_{m=1}^M F_{\text{Ind}_m^{g,k}}$.

Шаг 7. Если $\overline{F}_{\text{IndSet}_k^g} < \overline{F}_{\text{Ind}^g}$, то $D_{x_i}^{g,k} = \emptyset$ для $x_i, (i = \overline{1, n})$, иначе перейти к шагу 8.

Шаг 8. Если $k < K$, то $k = k + 1$, перейти к шагу 5. Иначе – расчет окончен – $D_{x_i}^g = \bigcup_{k=1}^K D_{x_i}^{g,k}$.

Динамическая коррекция границ поискового пространства является потенциально агрессивной операцией, поскольку преждевременное сужение области поиска может привести к потере глобального минимума и деградации поисковых свойств генетического алгоритма. Поэтому применение процедуры коррекции должно быть ограничено формальным условием, отражающим степень фактической локализации популяции в пространстве решений. Иными словами, коррекция границ целесообразна лишь в том случае, когда большая часть поискового пространства оказывается невостребованной текущим распределением индивидов.

Для этого вводится критерий, основанный на сравнении размеров областей поискового пространства, занятых кластерами решений, и областей, не покрытых ни одним кластером. Такой подход позволяет количественно оценить баланс между глобальным исследованием и локализацией поиска без введения дополнительных параметров-порогов, зависящих от конкретной задачи.

Область пространства поиска, не принадлежащая ни одному из выделенных кластеров:

$$D_{\bar{x}}^{g-1, \bar{k}} = \left(D_{x_1}^{g-1} \setminus \bigcup_{k=1}^K D_{x_1}^{g,k}, \dots, D_{x_n}^{g-1} \setminus \bigcup_{k=1}^K D_{x_n}^{g,k} \right). \quad (2.5)$$

Если суммарный размер области поискового пространства на поколении $g - 1$, не занятой кластерами решений на поколении g , превышает суммарный размер областей, занятых кластерами, выполняется процедура коррекции поискового пространства:

$$\sum_{i=1}^n \text{card} \left(D_{x_i}^{g-1, \bar{k}} \right) \geq \sum_{i=1}^n \text{card} \left(\bigcup_{k=1}^K D_{x_i}^{g,k} \right). \quad (2.6)$$

Описанный критерий служит формальным условием целесообразности запуска процедуры коррекции поискового пространства. В многомерных задачах оптимизации нежелательно сужать пространство слишком рано, пока существует значимая доля “неисследованной” области, алгоритм должен сохранять способность к глобальному поиску. Соответственно, коррекция границ вводится только тогда, когда популяция естественным образом локализовалась в ограниченной части исходного пространства.

Условие (2.6) интерпретируется следующим образом: левая часть представляет собой суммарный «объём» областей, не покрытых кластерами, тогда как правая часть соответствует суммарному объёму областей, фактически занятых кластерами с решениями; если неиспользованная часть превышает покрытую, это означает, что большая часть исходного пространства не содержит решений, вследствие чего запускается процедура локализации.

Следует подчеркнуть, что в данном алгоритме не вычисляется точное значение многомерного объёма области пространства поиска. Используется проекционная аппроксимация: для каждой переменной x_i берутся интервалы $D_{x_i}^{g,k}$,

после чего вычисляется длина проекции:

$$\text{card} (D_{x_i}^{g,k}) = \left| x_i^{g,k,h} + \Delta_{x_i^{g,k,h}} - \left(x_i^{g,k,l} + \Delta_{x_i^{g,k,l}} \right) \right|.$$

Оценивается сумма одномерных отрезков, а не реальный гиперобъём. Это осознанное упрощение, позволяющее:

1. Избежать сложного вычисления гиперобъёма объединения кластеров (аналог *Hypervolume* в МКО, даже для 3D такая задача NP-трудна);
2. Сохранить вычислительную сложность алгоритма на уровне $O(Kn)$, где K — число кластеров, а n — размерность пространства (число переменных);
3. Использовать проверку как эвристический индикатор степени локализации популяции.

Таким образом, данное условие не претендует на точную метрику объёма пространства, а выполняет роль решающего механизма: если популяция покрывает лишь малую часть первоначального пространства (см. рисунок 2.4), то следует сузить границы и повысить разрешение в областях, где реально сосредоточены решения.

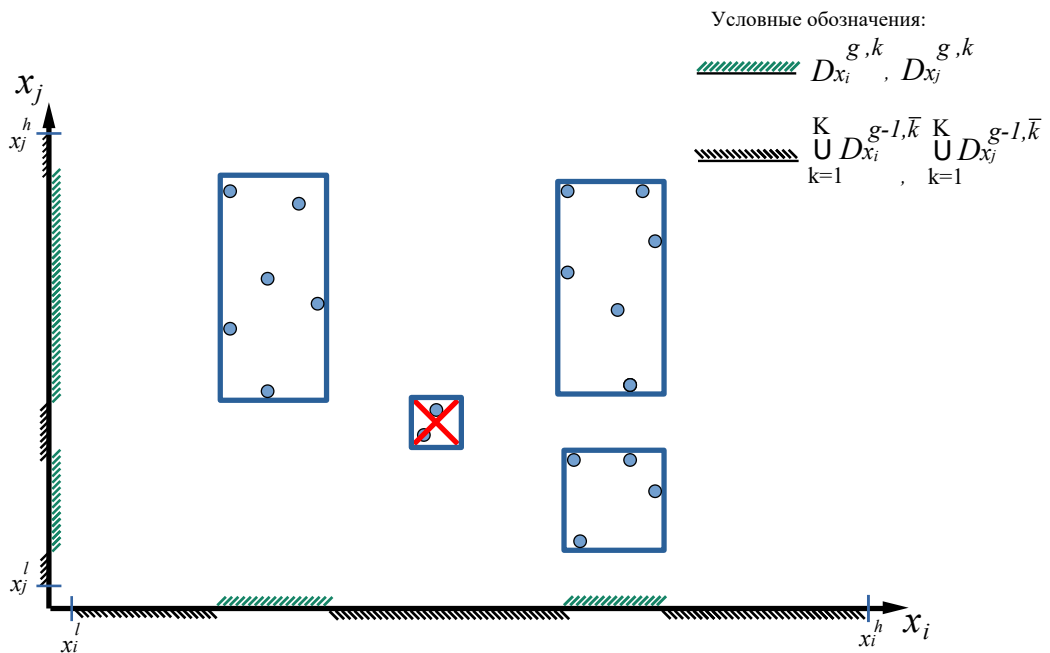


Рисунок 2.4 — Критерий возможности коррекции поискового пространства на примере двумерной области.

Корректировка границ кластеров. В результате выполнения предыдущих шагов алгоритма сформировано множество эффективных кластеров с решениями. При этом, в оценках границ областей пространства поиска не определены значения коррекции [168]. С целью расширения области поиска в сторону эффективных решений, а также редукции областей в малозначимых регионах предлагается процедура корректировки границ кластеров.

Введем следующие обозначения:

$F_{Ind_m^{g,k}}$ – пригодность m -го индивида в кластере k на поколении g ;

$Ph_m^{g,k} = Ph^g(Ind_m^{g,k}) = (x_1^{g,k,m}, \dots, x_n^{g,k,m})$ – фенотип m -го индивида в кластере k на поколении g ;

$Ph^g(\cdot)$ – схема перевода генотипа в фенотип на поколении g ;

$Ind_m^{g,k} = Bin^g(Ph_m^{g,k})$ – генотип m -го индивида в кластере k на поколении g ;

$Bin^g(\cdot)$ – схема перевода фенотипа в генотип на поколении g .

Рассмотрим произвольное поколение g работы ГА, кластер k и переменную пространства поиска x_i . Шаги нижеизложенной процедуры необходимо повторить для всех переменных по всем кластерам, для которых выполняется $D_{\vec{x}}^{g,k} \neq \emptyset$ ($k = \overline{1, K}$, $i = \overline{1, n}$).

Шаг 1. Рассчитать среднюю пригодность $RE_m^{g,k}$ всех индивидов во всех кластерах:

$$RE_m^{g,k} = RE(Ind_m^{g,k}) = \frac{F_{Ind_m^{g,k}}}{\sum_{p=1}^K \sum_{q=1}^M F_{Ind_q^{g,p}}} \quad (2.7)$$

Шаг 2. В каждом кластере найти по каждой переменной пространства поиска x_i ($i = \overline{1, n}$) индивида $IndL_{x_i}^{g,k}$ с наименьшим значением x_i и индивида $IndH_{x_i}^{g,k}$ с наибольшим значением x_i :

$$IndL_{x_i}^{g,k} = Bin^g \left(\min_{m=1, M_k^g} Ph_m^{g,k} \right) \quad (2.8)$$

$$\text{Ind}H_{x_i}^{g,k} = \text{Bin}^g \left(\max_{m=1, \overline{M_k^g}} \text{Ph}_m^{g,k} \right). \quad (2.9)$$

Шаг 3. Рассчитать значения коррекции верхней $\Delta_{x_i^{g,k,h}}$ и нижней $\Delta_{x_i^{g,k,l}}$ границ в соответствии с шагами 3.1-3.5.

Шаг 3.1. Рассчитать внутри-кластерную эффективность (Intra-Cluster Efficiency) – долю относительной эффективности m -го индивида в кластере k на поколении g :

$$\text{ICE}(\text{Ind}_m^{g,k}) = \frac{\text{RE}(\text{Ind}_m^{g,k})}{\sum_{q=1}^M \text{RE}(\text{Ind}_q^{g,k})}. \quad (2.10)$$

Шаг 3.2. Сформировать множество AI_k^g индивидов, для которых $\text{ICE}(\text{Ind}_m^{g,k}) \geq \frac{1}{M_k^g}$.

Шаг 3.3. Если $\text{ICE}(\text{Ind}L_{x_i}^{g,k}) \geq \frac{1}{M_k^g}$, то выполнить процедуру расширения нижней границы и рассчитать ее коррекцию:

$$\Delta_{x_i^{g,k,l}} = - \left| x_i^{g,k,h} - x_i^{g,k,l} \right| \text{ICE}(\text{Ind}L_{x_i}^{g,k}). \quad (2.11)$$

Перейти к шагу 3.5.

Шаг 3.4. Выполнить процедуру сужения нижней границы.

Заменить текущий $\text{Ind}L_{x_i}^{g,k}$ на индивид из множества AI_k^g в соответствии с формулой:

$$\text{Ind}L_{x_i}^{g,k} = \text{Bin}^g \left(\min_{\substack{x_i \\ \text{Ind}_m^{g,k} \in AI_k^g}} \text{Ph}_m^g(\text{Ind}_m^{g,k}) \right). \quad (2.12)$$

Перейти к шагу 3.3.

Шаг 3.5. Если $\text{ICE}(\text{Ind}H_{x_i}^{g,k}) \geq \frac{1}{M_k^g}$, то выполнить процедуру расширения верхней границы и рассчитать ее коррекцию:

$$\Delta_{x_i^{g,k,h}} = \left| x_i^{g,k,h} - x_i^{g,k,l} \right| \text{ICE}(\text{Ind}H_{x_i}^{g,k}). \quad (2.13)$$

Остановить алгоритм.

Шаг 3.6. Выполнить процедуру сужения верхней границы.

Заменить текущий $\text{Ind}H_{x_i}^{g,k}$ на индивид из множества AI_k^g в соответствии с формулой:

$$\text{Ind}H_{x_i}^{g,k} = \text{Bin}^g \left(\max_{\substack{x_i \\ \text{Ind}_m^{g,k} \in AI_k^g}} \text{Ph}^g (\text{Ind}_m^{g,k}) \right). \quad (2.14)$$

Перейти к шагу 3.5.

Описанная выше процедура позволяет не только производить редукцию границ областей поиска (пример показан на рисунке 2.5), но и расширять поисковое пространство в направлении расположения эффективных решений.

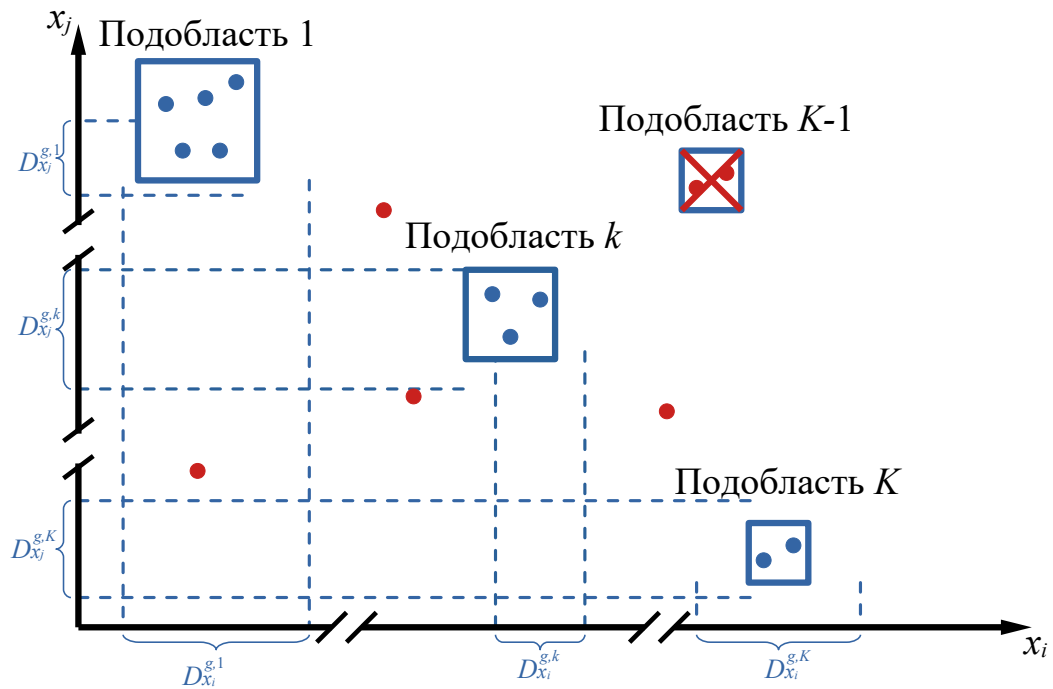


Рисунок 2.5 — Коррекция границ поискового пространства.

Корректировка границ требует количественной оценки вклада отдельных индивидов в общую эффективность кластера. Такая оценка должна быть совместима с классическими генетическими операторами (кроссовер, мутация), поэтому применяется стандартное бинарное кодирование фенотипа. Бинаризация не вводится как отдельный алгоритмический этап: она позволяет соединить два действия: адаптивное изменение границ и генетические операции. Тот же код, который участвует в селекции и мутирует, используется для перенастройки поиска.

Обоснование порога $\frac{1}{M_k^g}$. Выражение $\text{ICE}(\text{Ind}_m^{g,k}) \geq \frac{1}{M_k^g}$ означает, что вклад индивида в эффективность кластера не ниже усреднённого. Другими слова-

ми, индивид считается значимым, если он вносит не меньше, чем “средний” участник кластера, вклад в суммарную эффективность. Формально, величина $1/M_k^g$ трактуется как равномерное распределение эффективности между всеми индивидами. Если индивид превышает эту величину, то его можно считать статистически релевантным для формирования границы кластера.

Стоит отметить, что $\frac{1}{M_k^g}$ не является универсальным правилом. Это гиперпараметр, который может быть заменён, например, на:

$$\theta \cdot \frac{1}{M_k^g} \quad \text{или} \quad \text{ICE}(\text{Ind}_m^{g,k}) \geq \text{quantile}_p(\text{ICE}),$$

где $\theta > 0$ и $p \in [0,1]$ задаются пользователем, а $\text{quantile}_p(\text{ICE})$ – значение p -квантили распределения показателя ICE, то есть такое значение, что доля индивидов с меньшим или равным значением ICE составляет p .

Таким образом, порог определяет минимальный уровень статистической значимости при решении о расширении или сужении границ. В работе используется базовый вариант $\theta = 1$, что позволяет вводить критерий без дополнительных настроек и сохраняет вычислительную простоту процедуры.

Интерпретация порогового значения заключается в следующем: порог $1/M_k^g$ вводится не как точное оптимальное значение, а как минимально информативный критерий, позволяющий разделить «эффективные» и «неэффективные» индивиды внутри кластера. Он не предполагает прямого сравнения пригодности и численности особей; вместо этого используется нормировка вклада по распределению ICE, обеспечивающая относительное сравнение при низкой вычислительной сложности. Таким образом, порог служит механизмом регуляризации и гарантирует, что изменение границ кластера основано на статистически подтверждённой информации, а не на единичных аномальных точках.

Смена схемы кодирования. В данном разделе описывается изменение схемы бинарного кодирования вещественной части решения. Необходимость такой модификации связана с адаптивным изменением области поиска: при перераспределении границ вещественного подпространства требуется корректировать масштаб дискретизации, чтобы сохранять адекватное разрешение представления решений в новых границах. Переменные всех остальных типов не подлежат динамической коррекции масштаба поиска.

Шаг дискретизации в классическом ГА $\text{step}_{x_i} = \text{const}$, в предложенной модификации ГА $\text{step}_{x_i}^g$ — динамически меняется [169]. Начальный шаг $\text{step}_{x_i}^1$ и область поиска $D_{x_i}^1 = [x_i^l, x_i^h]$ задаются пользователем. Если $\text{Bin}^1(\cdot)$ — классическая схема бинарного кодирования, то $\text{Bin}^1 = \{\text{Bin}_{x_1}^1, \dots, \text{Bin}_{x_n}^1\}$ — оператор над компонентами фенотипа. Рассмотрим алгоритм подробнее:

1. Требуемое количество бит для кодирования определяется следующим выражением:

$$N\text{Bit}_{x_i} = \left\lceil \log_2 \left(\frac{x_i^h - x_i^l}{\text{step}_{x_i}^1} + 1 \right) \right\rceil. \quad (2.15)$$

2. Коррекция шага:

$$\text{step}_{x_i} C_{x_i}^1 = \frac{x_i^h - x_i^l}{2^{N\text{Bit}_{x_i}} - 1}. \quad (2.16)$$

Знаменатель $2^{N\text{Bit}_{x_i}} - 1$ обеспечивает точное попадание крайних бинарных кодов в границы интервала:

$$000 \dots 0 \rightarrow x_i^l, \quad 111 \dots 1 \rightarrow x_i^h.$$

Без «−1» значение x_i^h невозможно получить ни при каком бинарном коде.

3. Базовая схема кодирования фенотипа:

$$\text{Ind}_{m,x_i}^g = \left(\frac{\text{Ph}_{m,x_i}^g - x_i^l}{\text{step}_{x_i} C_{x_i}^1} \right)_{10 \rightarrow 2}. \quad (2.17)$$

Однако прямое применение формулы (2.17) затруднено, поскольку после изменения границ кластера значения Ph_{m,x_i}^g обычно не совпадают с узлами дискретизации, и деление на $\text{step}_{x_i} C_{x_i}^1$ даёт вещественное число. Чтобы сделать (2.17) реализуемой, требуется округление к ближайшему допустимому узлу:

$$t_{m,x_i}^g = \text{round} \left(\frac{\text{Ph}_{m,x_i}^g - x_i^l}{\text{step}_{x_i} C_{x_i}^1} \right), \quad \text{Ind}_{m,x_i}^g = (t_{m,x_i}^g)_{10 \rightarrow 2}. \quad (2.18)$$

Таким образом, перемещение в новую бинарную систему координат становится аппроксимационным, но всегда корректным. Это позволяет выполнять

динамическую смену масштаба поиска без потери совместимости с генетическими операторами (кроссовер и мутация остаются бинарными).

Для поколений, в которых не происходит коррекция границ поискового пространства, схема кодирования не претерпевает изменений, т.е. $\text{Bin}^g(\cdot) = \text{Bin}^{g-1}(\cdot)$.

Для поколений, в которых происходит коррекция границ поискового пространства, $\text{Bin}^g(\cdot)$ определяется как нижеследующая процедура:

-
1. Упорядочивание кластеров по возрастанию левых границ $x_i^{g,k,l} + \Delta_{x_i^{g,k,l}}$.
 2. Для каждого кластера k найти показатель удаленности границы минимального значения по x_i относительно начала $D_{x_i}^{g,l}$ пространства поиска $D_{x_i}^g$. Определим его как сдвигку $S_{x_i}^{g,k,l}$:

$$\begin{aligned}
S_{x_i}^{g,k,l} = & \left(x_i^{g,1,h} + \Delta_{x_i^{g,1,h}} \right) - \left(x_i^{g,1,l} + \Delta_{x_i^{g,1,l}} \right) + \\
& + \sum_{\substack{e=2, \overline{k-1} \\ x_i^{g,e,h} > x_i^{g,d,h}, \forall d=\overline{1, e-1} \\ x_i^{g,e,l} > x_i^{g,d,h}, \forall d=\overline{1, e-1}}} \left(\underbrace{\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \left(x_i^{g,e,l} + \Delta_{x_i^{g,e,i}} \right)}_{\text{чистые отрезки}} \right) + \\
& + \sum_{\substack{e=2, \overline{k-1} \\ x_i^{g,e,h} > x_i^{g,d,h}, \forall d=\overline{1, e-1} \\ x_i^{g,e,l} \leq x_i^{g,d,h}, \forall d=\overline{1, e-1}}} \left(\underbrace{\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \max_{d=\overline{1, e-1}} \left(x_i^{g,d,h} + \Delta_{x_i^{g,d,i}} \right)}_{\text{отрезки с перекрытиями}} \right).
\end{aligned} \tag{2.19}$$

Если $x_i^{g,e,l} > x_i^{g,e-1,h}$ - берем всю длину отрезка.

Если $x_i^{g,e,h} > x_i^{g,d,h}, \forall d = \overline{2..e-1}$ - тогда 0, т.к. вложенные отрезки не суммируются.

Если $x_i^{g,e,l} \leq x_i^{g,e-1,h}, \forall d = \overline{2..e-1}$ - тогда берем часть длины $\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \max_{d=\overline{1, e-1}} \left(x_i^{g,d,h} + \Delta_{x_i^{g,d,i}} \right)$.

Пример проекции значений отрезков на переменную x_i представлен на рисунке 2.6.

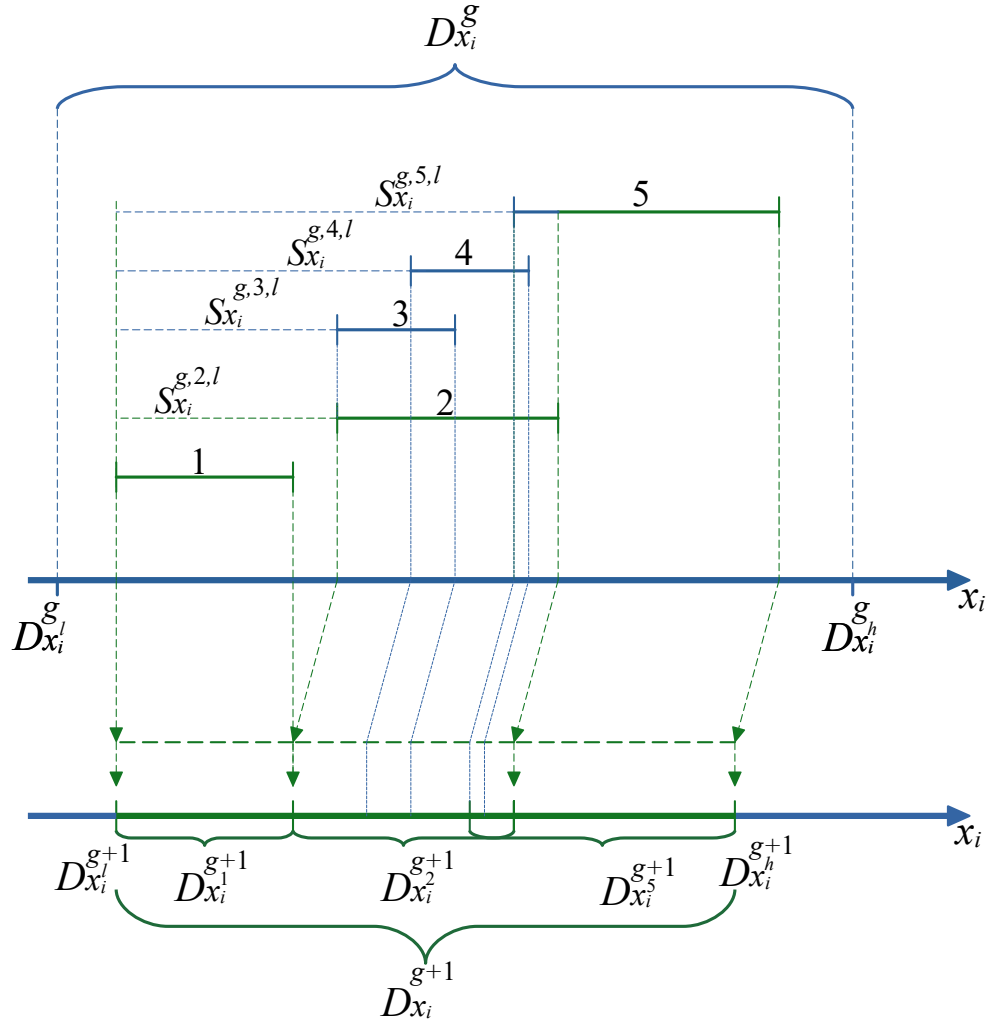


Рисунок 2.6 — Проекция отрезков на переменную x_i .

3. Коррекция шага по каждой переменной x_i :

$$\text{step}C_{x_i}^1 = \frac{\left(x_i^{g,k,h} + \Delta_{x_i^{g,k,h}}\right) - \left(x_i^{g,k,l} + \Delta_{x_i^{g,k,l}}\right)}{2^{N\text{Bit}_{x_i}}}. \quad (2.20)$$

4. Для всех решений, оставшихся после удаления индивидов, и не попадающих в новое пространство поиска, выполняем следующую процедуру:

$$\text{Ind}_{m,x_i}^{g+1,k} = \begin{cases} \frac{\text{Ph}_{m,x_i}^{g,k} - \left(x_i^{g,1,l} + \Delta_{x_i^{g,1,l}}\right)}{\text{step}C_{x_i}^1}, & k = 1 \\ S_{x_i}^{g,k,l} + \frac{\text{Ph}_{m,x_i}^{g,k} - \left(x_i^{g,k,l} + \Delta_{x_i^{g,1,l}}\right)}{\text{step}C_{x_i}^1}, & k = \overline{2..K} \end{cases}. \quad (2.21)$$

Таким образом, предлагаемая динамическая смена шага дискретизации и корректировка границ кластеров воздействует исключительно на вещественную подсистему переменных. Остальные переменные участвуют в эволюции через стандартные операторы скрещивания и мутации и влияют на пригодность $F_{Ind_m^g}$, тем самым опосредованно учитываются при отборе кластеров и коррекции поискового пространства.

Пример работы алгоритма. На рисунке 2.7 представлена последовательность визуализаций работы предложенного ГА для поиска перспективного поискового пространства функции Стыблински-Танга. Оптимизация проводилась в течение 50 поколений, и изображения показывают состояния на 0, 7, 14, 21, 28 и 43 поколениях. Исходно границы поискового пространства были заданы за пределами оптимальных значений, в квадрате, ограниченном прямыми от -1 до 1 по обоим переменным, с шагом начальным 0.1. В процессе работы алгоритм постепенно перемещал область поиска, в итоге сосредоточив внимание на одном из глобальных оптимумов функции.

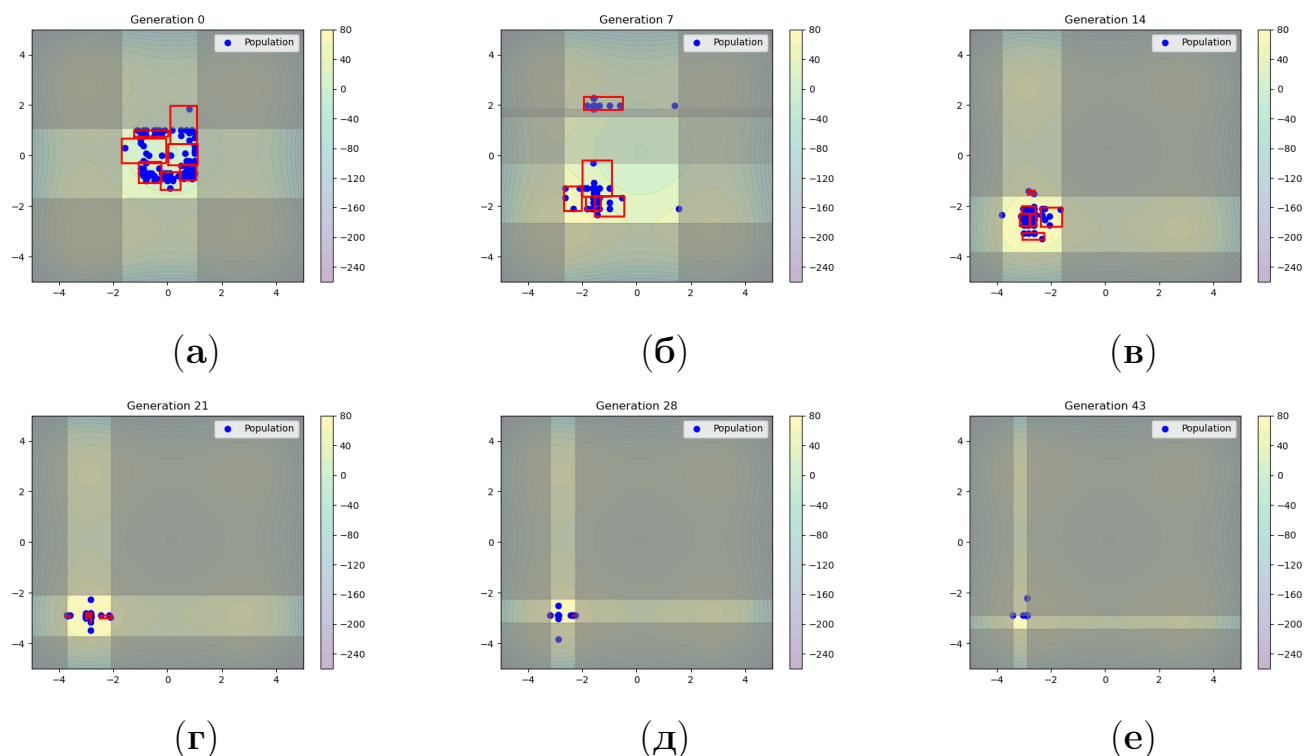


Рисунок 2.7 — Последовательность визуализаций работы ГА при поиске перспективного поискового пространства для функции Стыблински-Танга.

На рисунке 2.8 представлена последовательность визуализаций процесса поиска максимума функции Растригина в течение 50 поколений [170]. Оптимизация проводилась в течение 50 поколений, изображения соответствуют 0,

6, 12, 18, 24, 30, 36, 42 и 49 поколениям. Исходное область поиска охватывала диапазоны $x \in [-12, 5] \cup [8, 10]$ и $y \in [12, 10] \cup [10, 12]$ с начальным шагом 0.1. В ходе оптимизации границы области поиска постепенно сужались, в итоге сосредоточившись вблизи глобального максимума функции.

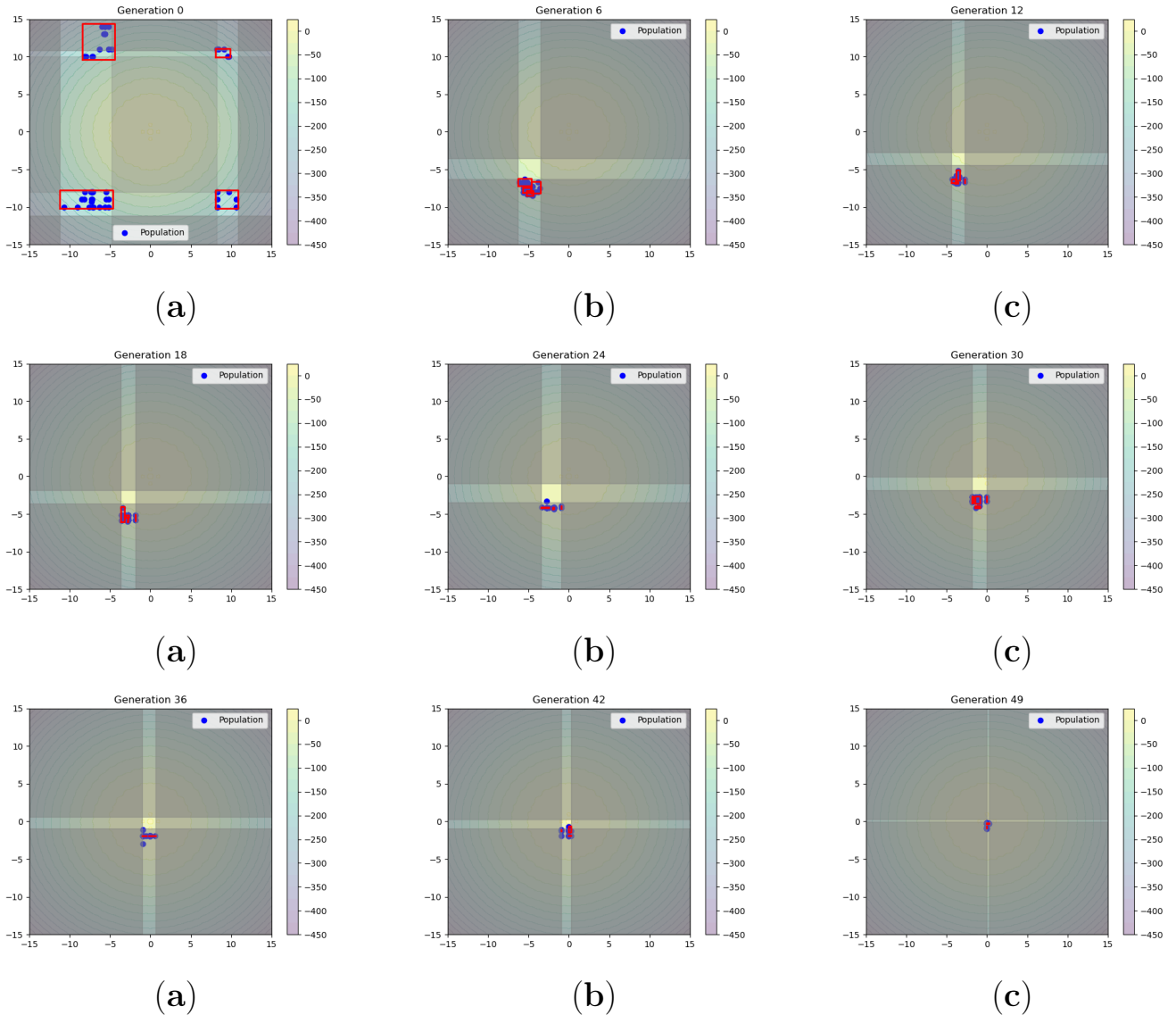


Рисунок 2.8 — Последовательность визуализаций работы адаптивного генетического алгоритма (ГА) при поиске перспективного поискового пространства для функции Растригина.

2.2 Анализ работоспособности и визуализация на базовых тестовых функциях

Все эксперименты выполнялись на системе с характеристиками: процессор Intel(R) Core(TM) J5-6600K CPU @ 3.50GHz, 8,00 ГБ оперативной памяти, 64-разрядная операционная система, процессор x64. Алгоритм запускался с начальным размером популяции 100 и максимальным числом поколений 100.

Для оценки эффективности метода были рассмотрены стандартные тестовые функции оптимизации, широко используемые в литературе благодаря разнообразию рельефов и сложности ландшафта целевой поверхности. В набор вошли как простые одномодальные функции (сфера, Розенброк), так и сложные многомодальные (Растригина, Экли, Эггхолдер, Хольдера и др.), а также функции с несколькими локальными минимумами и особыми свойствами рельефа (Била, Голдштейна–Прайса, Бута, Букина №6, Химмельблау, «трёхгорбый верблюд», Изюма, "крест на подносе" (Cross-in-tray), Матъяса, Леви N.13, Мак-Кормика, Шаффера N.2 и N.4, Стыбинского–Танга). Такой выбор обеспечивает всестороннюю проверку: от простых сценариев с единственным минимумом до высокоосциллирующих и разрывных функций, позволяя объективно оценить точность, скорость сходимости и устойчивость алгоритма.

Изменение шага и гиперобъема. Из рисунка 2.9 видно, что в ходе работы предложенного ГА дискретизационный шаг постепенно снижается по мере увеличения числа генераций.

Например, для функции Экли начальное значение составляет 0.0466, а к 29-й итерации оно опускается до 0.0014, что свидетельствует о более точном определении области поиска. Аналогичный тренд отмечается для функции Била: параметр изменяется с 0.0625 до примерно 0.0018 к 22-й итерации. Для функции Бута видно плавное уменьшение шага от 0.0519 до 0.0011 с незначительными колебаниями, что отражает стабилизацию вблизи локального минимума. В случае функции Букин №6 наблюдается резкое сокращение значения с 0.0620 до 0.00255, указывающее на быстрое сужение интервала поиска. Подобные закономерности фиксируются и для функции крест в ларце, где величина шага падает с 0.0528 до 0.002 к 13-й итерации, а затем остается на низком уровне. В совокупности эти данные подтверждают, что с увеличением числа итераций

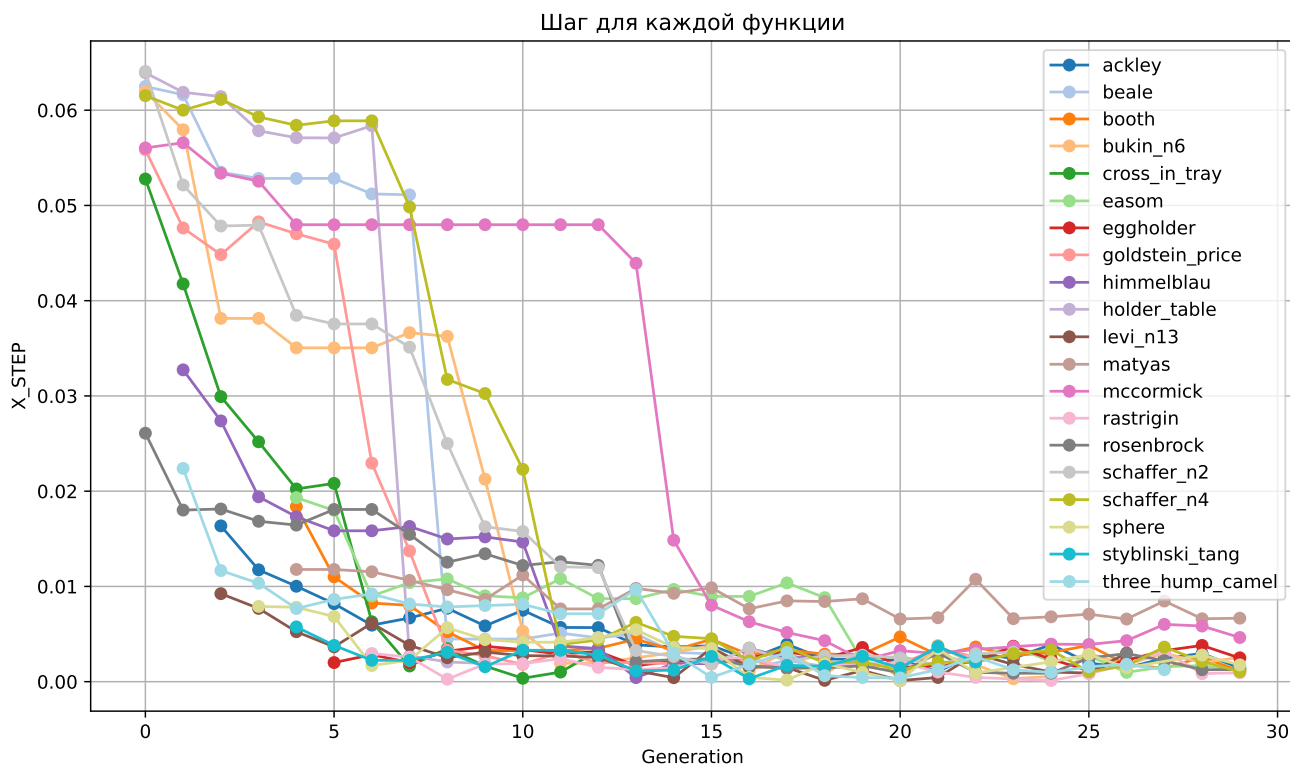


Рисунок 2.9 — Изменение дискретизационного шага алгоритма ГА+ДКОП в зависимости от числа итераций для различных тестовых функций при количестве кластеров 4.

алгоритм адаптивно уточняет дискретизацию, что способствует достижению высокой точности при локальной оптимизации. Такой механизм способствует эффективному сочетанию глобального исследования и детальной эксплуатации, что является важным для решения сложных оптимизационных задач. Гиперобъем в зависимости от числа итераций (рисунок 2.10) коррелирует с изменением шага дискретизации в процессе оптимизации.

2.3 Результаты исследования эффективности предложенного подхода на функциях CEC2017

Параметры алгоритмов и их реализации. В данном разделе проводится сравнительный анализ эффективности динамической коррекции области поиска (ДКОП) при применении к различным эволюционным алгоритмам. Рассматриваются бинарный генетический алгоритм (ГА), бинарный алгоритм роя

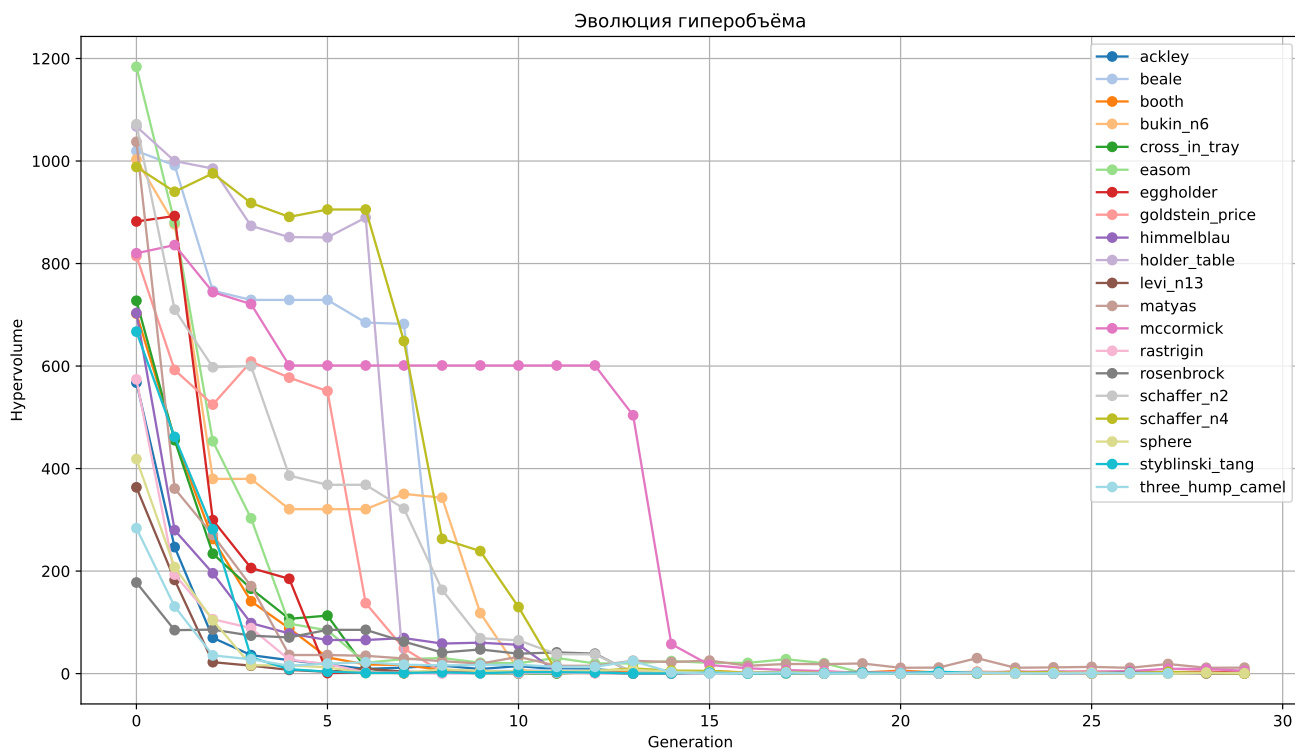


Рисунок 2.10 — Изменение перобъёма в зависимости от числа генераций для различных тестовых функций при количестве кластеров 4

частиц с полной информацией (FPSO) и бинарный алгоритм дифференциальной эволюции (DE), а также их модификации с использованием ДКОП.

В качестве тестового набора используется CEC-2017 [123], включающий 30 функций различной структуры, в том числе унимодальные, мультимодальные и композиционные. Выбор данного бенчмарка обусловлен его широким использованием в задачах глобальной оптимизации, наличием функций с различной сложностью ландшафта и возможностью сопоставления результатов с существующими работами. Использование данного набора позволяет рассматривать полученные результаты как репрезентативную оценку эффективности алгоритмов в задачах глобальной оптимизации.

Эксперименты проводились при размерностях пространства поиска $D \in \{10, 30, 50\}$. Для обеспечения сопоставимости результатов для всех алгоритмов использовался одинаковый бюджет вычислений, задаваемый через общее число вычислений целевой функции. Для каждой функции выполнялось по 50 независимых запусков.

В качестве метрик анализа использовались среднее значение целевой функции, стандартное отклонение, а также ранговые показатели алгоритмов

по каждой функции. Это позволяет оценить как точность, так и устойчивость алгоритмов.

Для обеспечения сопоставимости результатов параметры всех алгоритмов подбирались таким образом, чтобы сохранить одинаковый порядок вычислительной сложности и бюджет вычислений. В качестве основного ограничения использовалось общее число вычислений целевой функции NFE . Для генетического алгоритма и дифференциальной эволюции бюджет задавался как $NFE = N_{pop} \cdot G$, где N_{pop} — размер популяции, G — число поколений. Для алгоритма роя частиц использовалась стандартная для набора СЕС схема $NFE = 10000 \cdot D$. Границы области поиска во всех экспериментах фиксированы и задавались как $[-100, 100]$, а также использовалась единая схема многократных независимых запусков с различными начальными инициализациями.

Общие параметры включают размер популяции (или роя), число итераций (или эквивалентный бюджет NFE), а также границы области поиска. Специфические параметры зависят от используемого алгоритма. В генетическом алгоритме применяются параметры вероятности мутации и настройки селекции, а также параметры кластеризации в варианте с ДКОП. В алгоритме роя частиц используются коэффициент инерции, параметры ограничения скорости и параметр отображения в бинарное пространство. В дифференциальной эволюции применяются коэффициенты мутации и кроссовера, а также длина бинарного кодирования переменных. Параметры алгоритмов приведены в таблице 1.

Таблица 1 — Параметры экспериментальных исследований

Алгоритм	Размерн. D	Число поколе- ний	Размер популя- ции	Основные парамет- ры
ГА ГА+ДКОП	10	200	2000	p_c, p_m tour, elit, K
SelfCSHAGA SelfCSHAGA+ДКОП	10	200	2000	p_c, p_m Gray, step= 10^{-3} K, α_N, α_F
	30	200	4000	
	50	400	8000	
FPSO FPSO+ДКОП	10	200	2000	w, c $\omega = 0.9, v \in [-6, 6]$ $L = 10, DSP$
	30	200	4000	
	50	400	8000	
L-SRTDE L-SRTDE+ДКОП	10			F, CR
	30			
	50			

Реализация генетического алгоритма в эксперименте включает два варианта. На первом этапе для размерности $D = 10$ был рассмотрен классический бинарный генетический алгоритм как базовый метод и его модификация с ДКОП. Это позволило отдельно оценить влияние динамической коррекции области поиска на наиболее простой и понятной схеме эволюционного поиска. Однако уже при $D = 10$ обычный бинарный ГА показал слабые результаты по сравнению с более современными подходами, а дальнейшее увеличение размерности приводило к резкому росту значений целевой функции и ухудшению устойчивости. По этой причине проведение расчётов для классического ГА при $D = 30$ и $D = 50$ не представлялось целесообразным, поскольку получаемые значения оказывались существенно хуже и не позволяли получить содержательного сравнения. В связи с этим для дальнейших экспериментов был использован самонастраивающийся генетический алгоритм SelfCSHAGA [114], который удалось корректно применить для размерностей $D \in \{10,30,50\}$ как без ДКОП, так и с ДКОП.

Обычный бинарный ГА реализован по классической схеме с турнирной селекцией, одноточечным скрещиванием и побитовой мутацией. Вещественные переменные кодируются бинарными строками с фиксированным числом бит на переменную, определяемым из требуемого шага дискретизации, после чего декодируются в текущих границах области поиска. В реализации предусмотрена кластеризация популяции по фенотипу, фильтрация малых и неэффективных кластеров, проверка условия целесообразности сужения области поиска и последующее перекодирование особей после изменения границ. Дополнительно используется элитный архив лучших решений и механизм иммиграции, позволяющий восстанавливать численность популяции после фильтрации. В варианте без ДКОП алгоритм работает как с бинарным ГА, а в варианте с ДКОП к базовому циклу добавляется процедура кластерной локализации и коррекции границ.

Реализация SelfCSHAGA основана на самонастраивающемся генетическом алгоритме [171], в котором параметры эволюции адаптируются по ходу работы на основе статистики успешных решений. В отличие от обычного ГА, в данной реализации размер популяции сохраняется постоянным, а воздействие ДКОП осуществляется не через удаление особей, а через замену решений из кластеров, признанных малыми и неэффективными. Кластеризация выполня-

ется по фенотипическому представлению текущей популяции с использованием метода k -средних. Если кластер одновременно имеет размер ниже среднего и среднюю пригодность ниже медианного уровня, его особи с заданной вероятностью заменяются сначала решениями из элитного архива, а при их недостатке — случайно сгенерированными генотипами. Такая схема позволяет встроить ДКОП в цикл SelfCSHAGA без нарушения внутренней логики самонастройки операторов и без изменения длины популяции, что оказалось важным для устойчивой работы алгоритма на размерностях $D = 10, 30$ и 50 .

Реализация бинарного алгоритма роя частиц с полной информацией (FPSO) [141] основана на побитовом варианте бинарного PSO, в котором каждая координата кодируется бинарной строкой фиксированной длины $L = 10$. Переход между бинарным и вещественным представлениями осуществляется через равномерное квантование на отрезке $[-100, 100]$, что позволяет использовать функции набора SEC2017 при сохранении бинарной природы поиска. Скорости задаются для каждого бита отдельно и инициализируются случайными значениями из интервала $[-6, 6]$. Их обновление выполняется по схеме PSO с коэффициентом инерции $\omega = 0.9$ и одинаковыми когнитивным и социальным коэффициентами, величина которых зависит от размерности задачи. После обновления скоростей новые значения битов генерируются вероятностным образом с использованием сигмоидального преобразования, так что скорость интерпретируется как вероятность установки соответствующего бита в единицу.

В реализации с ДКОП коррекция области поиска выполняется в фенотипическом пространстве. Для повышения устойчивости вместо текущих положений частиц используются их лучшие найденные положения, которые декодируются в вещественное пространство и кластеризуются методом k -средних. Для каждого кластера строится гиперпрямоугольная область по покоординатным минимумам и максимумам, после чего исключаются малые кластеры и кластеры со средней пригодностью хуже средней по популяции. Полученные области дополнительно корректируются, чтобы избежать вырожденных интервалов и выхода за глобальные границы поиска. Далее проверяется принадлежность текущих частиц объединению найденных областей, и частицы, оказавшиеся вне допустимых регионов, пересэмплируются внутри них с последующим обратным бинарным кодированием и повторной оценкой. Таким образом, в FPSO+ДКОП

динамическая коррекция реализуется не через изменение размера роя, а через регулярное восстановление популяции внутри перспективных областей пространства поиска при сохранении механизма обновления скоростей и личных лучших положений.

Реализация бинарной дифференциальной эволюции опирается на идеи самонастраивающегося алгоритма L-SRTDE [134], исходно разработанного для оптимизации вещественных переменных. В исходной реализации используются вещественная популяция, адаптация коэффициента кроссовера по памяти успешных значений, зависимость масштаба мутации от текущей успешности, а также уменьшение эффективного размера популяции по мере исчерпания бюджета вычислений. В проведённом исследовании эти принципы были перенесены в бинарное пространство, для чего была построена отдельная бинарная реализация алгоритма.

В предложенной бинарной версии популяция хранится непосредственно в виде битовых строк, а каждая вещественная переменная кодируется фиксированным числом бит. Декодирование в пространство поиска осуществляется линейным отображением двоичного целого значения в интервал $[-100, 100]$, после чего вычисляется значение целевой функции. Оператор мутации реализован через бинарную разностную схему: для трёх случайно выбранных особей формируется вектор различий на основе операции XOR, после чего часть различающихся битов переносится в донорский вектор с вероятностью F_{bit} . Кроссовер выполняется по побитовой биномиальной схеме с параметром CR , причём, как и в классической DE, гарантируется наследование хотя бы одного бита от донора. Отбор осуществляется жадно: пробная особь замещает целевую только в том случае, если не уступает ей по значению целевой функции.

В модификации L-STRDE+ДКОП процедура динамической коррекции области поиска выполняется в декодированном вещественном пространстве. После очередного поколения текущая популяция кластеризуется методом k -средних, затем отбрасываются малые кластеры и кластеры со средней пригодностью хуже средней по популяции. Для оставшихся кластеров строятся гиперпрямоугольные области, после чего дополнительно проверяется критерий целесообразности коррекции: суммарная длина непокрытой части пространства должна превышать суммарную длину занятой части по координатным проекциям. Если условие выполняется, границы кластеров корректируются на основе

относительной эффективности решений внутри кластера, а особи, оказавшиеся вне полученных областей, заменяются новыми точками, сэмпированными внутри допустимых регионов, с последующим повторным бинарным кодированием. Таким образом, в варианте L-STRDE+ДКОП бинарная дифференциальная эволюция дополняется механизмом локализации поиска в перспективных подобластях без изменения общей схемы эволюционного обновления популяции.

Результаты ГА и SelfCSHAGA. В таблице A1 представлены средние значения целевых функций тестового набора CEC2017, полученные по результатам серии запусков алгоритмов ГА, ГА+ДКОП, SelfCSHAGA и SelfCSHAGA+ДКОП при различных размерностях пространства поиска. Результаты, представленные на рисунке 2.11, демонстрируют поведение ГА и его модификации при размерности $D = 10$.

Видно, что ДКОП ускоряет сходимость в начальной фазе эволюции и позволяет достигать меньших значений функции пригодности. На графиках видно, что использование ДКОП в составе генетического алгоритма приводит к более раннему снижению значения функции пригодности по сравнению с базовой версией алгоритма [172]. В большинстве функций кривая ГА+ДКОП достигает области стабильности на меньшем числе поколений, что отражает более быстрый переход к финальному этапу поиска. Также высота колебаний по усреднённым значениям в кластере запусков сочетается с меньшим интервалом вариации по сравнению с классическим ГА. Данный эффект можно наблюдать по характеру зелёных областей, которые в ряде случаев более узкие относительно оранжевых. Аналогичные наблюдения применимы и к алгоритму SelfCSHAGA при $D = 10$, что подтверждается рисунком 2.12.

При увеличении размерности до $D = 30$ (рис. 2.13) влияние динамической коррекции области поиска становится более выраженным. Усреднённые кривые по большинству функций выходят на область стабильных значений быстрее при использовании варианта SelfCSHAGA с ДКОП. Также в ряде задач (например, $f3$, $f5$ и $f24$) заметен более узкий коридор вариации значений, что указывает на повышенную устойчивость поведения алгоритма на множестве запусков.

При размерности $D = 50$, что показано на рисунке 2.14, обе версии алгоритма выходят на область низких значений функции пригодности по большинству задач, однако по ряду функций (например, $f3$, $f18$, $f24$) заметно постепенное снижение значения после основного этапа поиска, что реализуется только

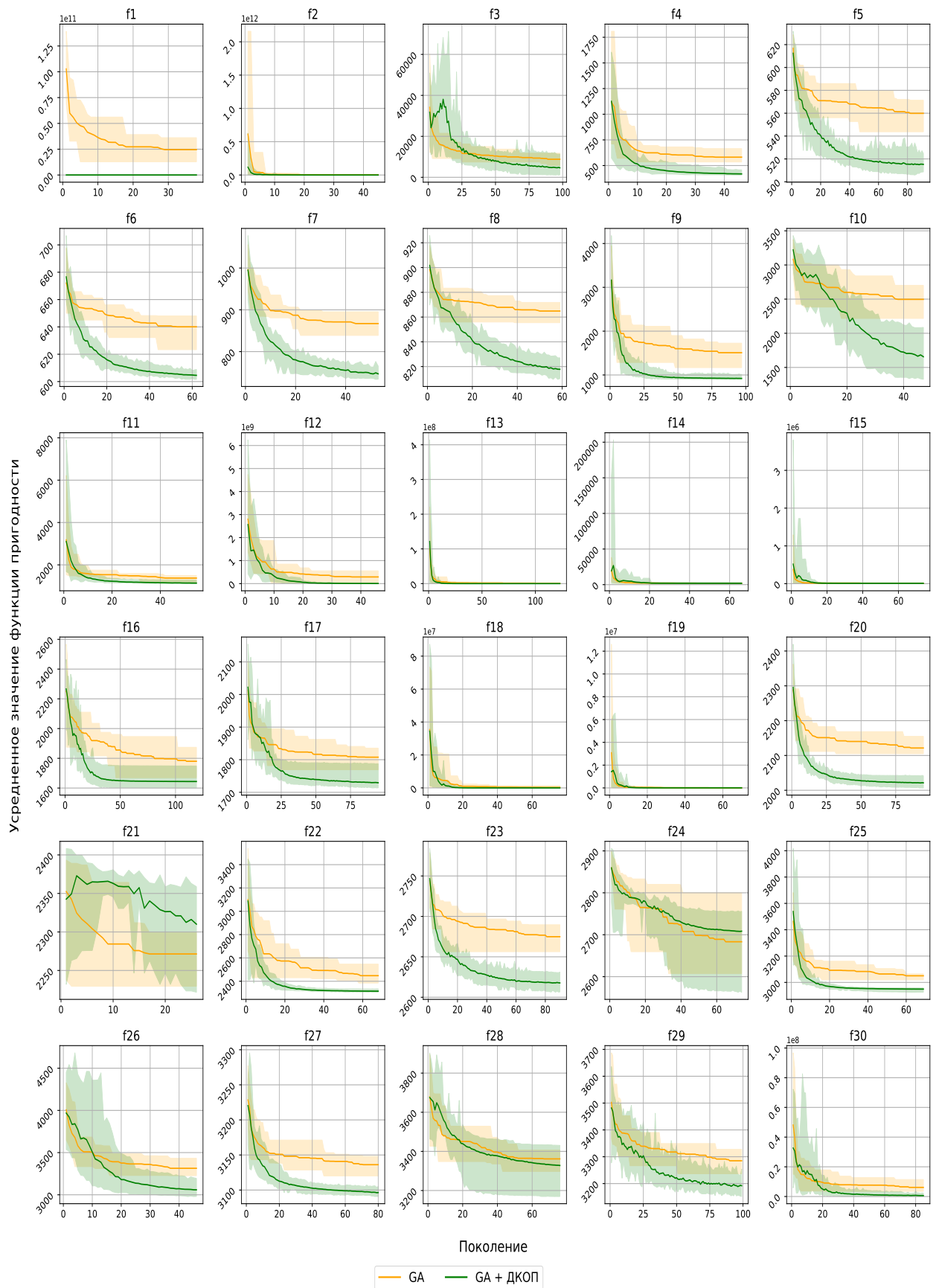


Рисунок 2.11 — Эволюция значения функции пригодности ГА и ГА+ДКОП при $D = 10$.

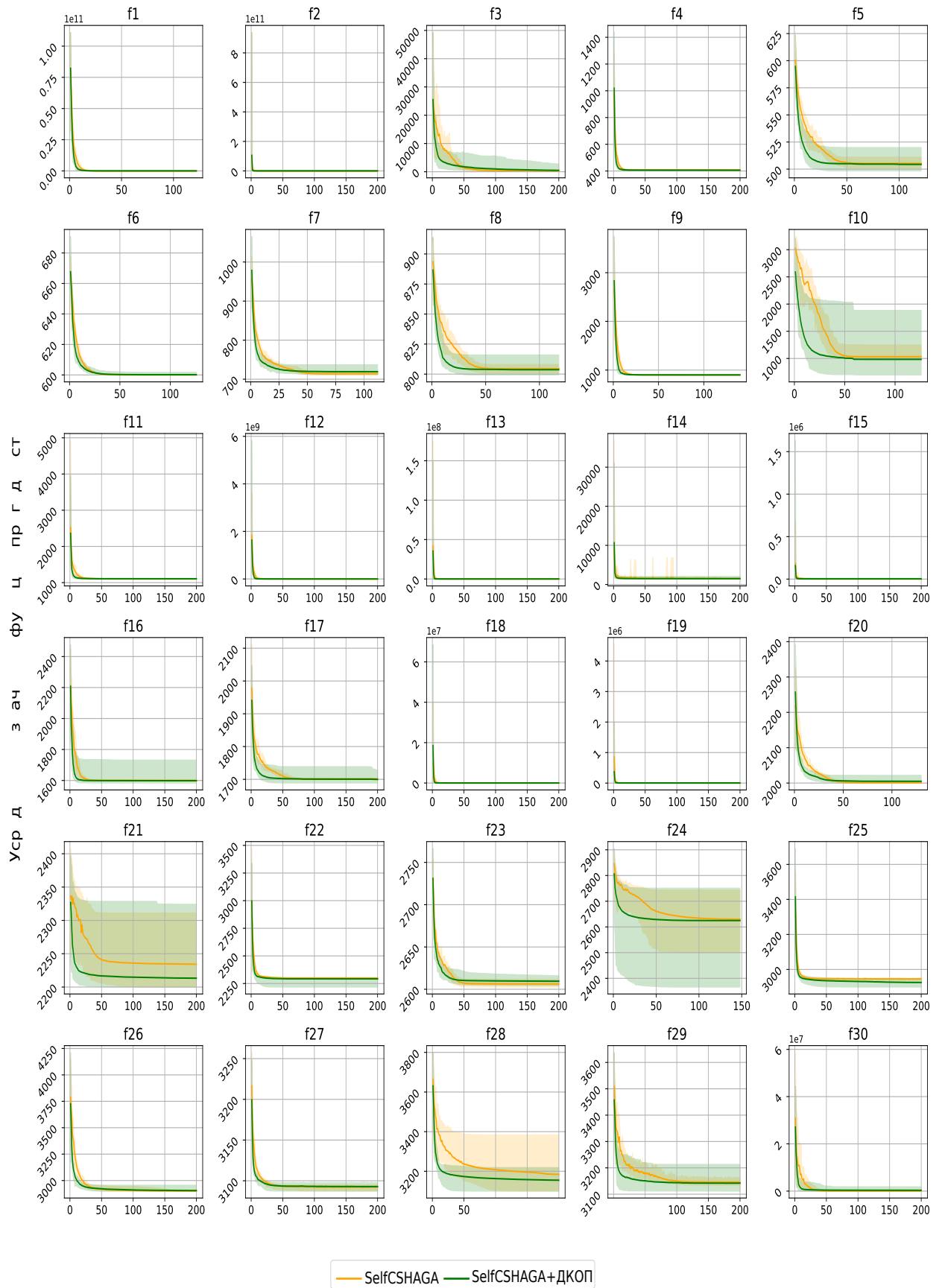


Рисунок 2.12 — Динамика SelfCSHAGA и SelfCSHAGA+ДКОП при $D = 10$.

при использовании механизма ДКОП. Также на некоторых функциях, таких

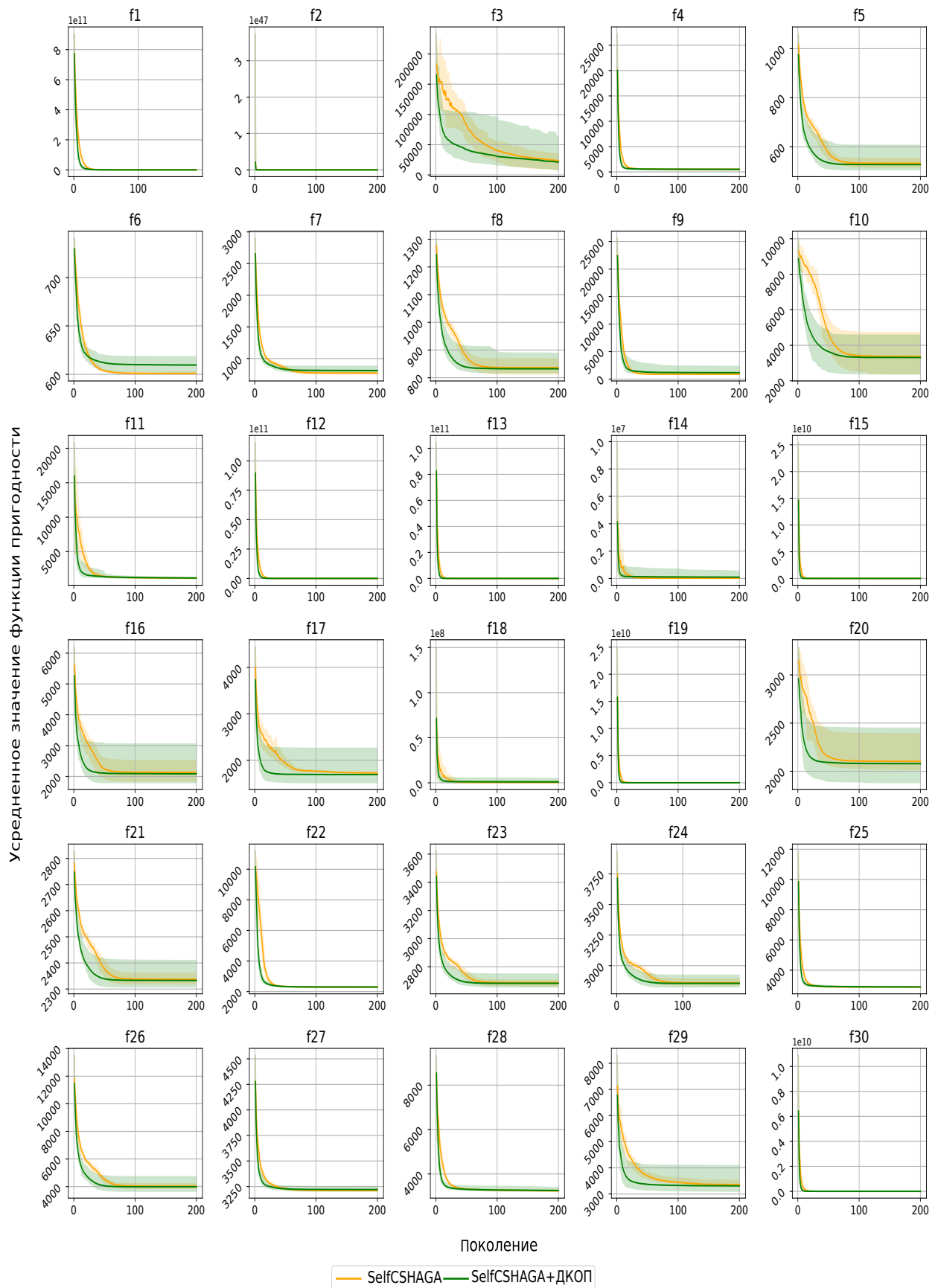


Рисунок 2.13 – Динамика SelfCSHAGA и SelfCSHAGA+ДКОП при $D = 30$.

как f_{21} – f_{25} , наблюдается растяжение по оси поколений, что может указывать

на снижение градиентности поискового ландшафта при увеличении размерности и необходимость адаптивного регулирования структуры популяции.

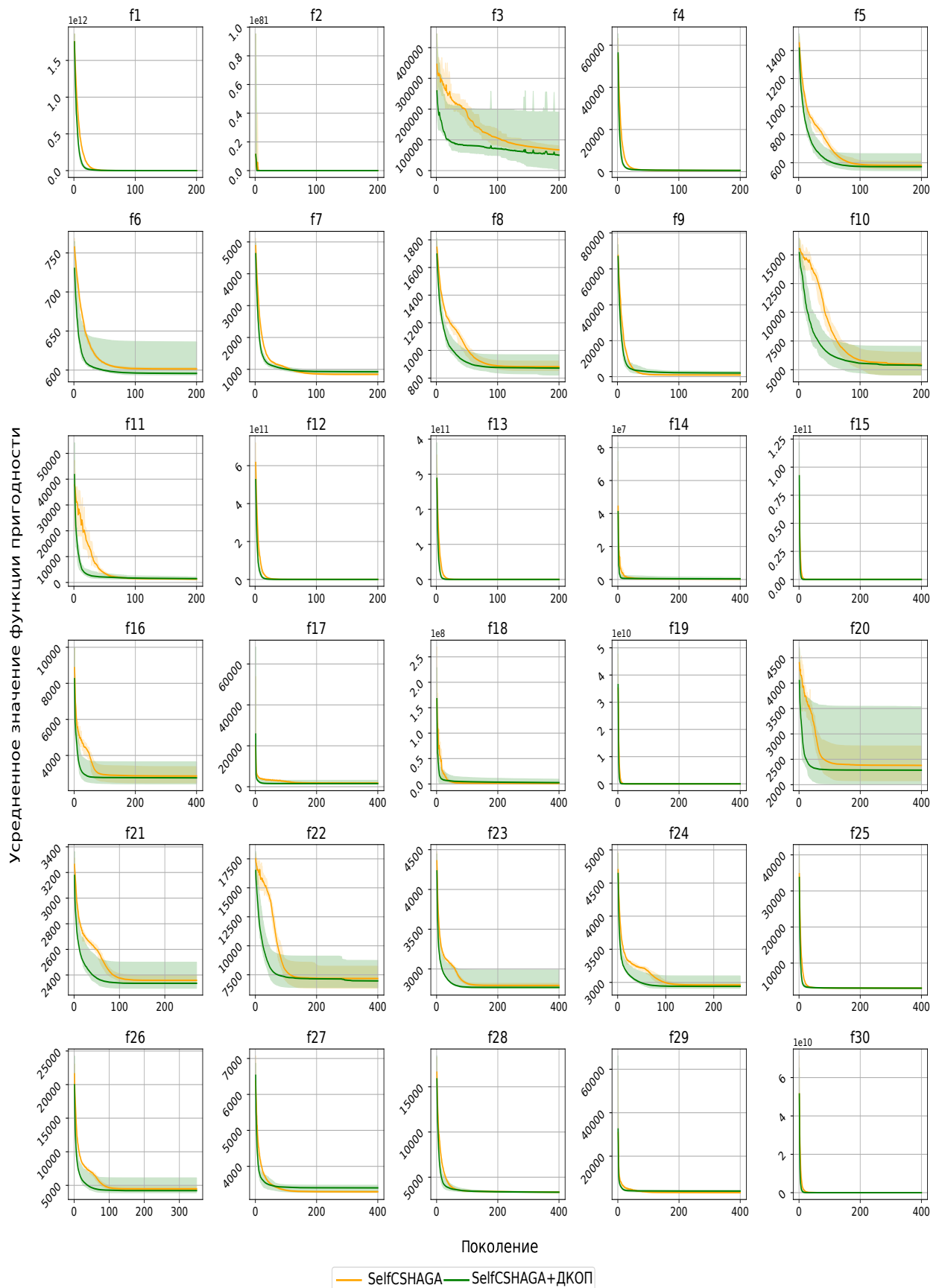


Рисунок 2.14 — Динамика SelfCSHAGA и SelfCSHAGA+ДКОП при $D = 50$.

Сравнение алгоритмов. После проведения исследования с FPSO и L-STRDE, аналогичного описанию в разделе выше, получились результаты, представленные в таблицах 2, 3 и 4. В таблице A2 представлены средние значения целевых функций тестового набора CEC2017, полученные для алгоритмов BPSO и BPSO+ДКОП при различных размерностях пространства поиска. В таблице A3 приведены средние значения функций тестового набора CEC2017, полученные бинарной версией алгоритма L-SRTDE (bL-SRTDE) и его модифицированной версией при различных размерностях пространства поиска.

Таблица 2 — Анализ эффективности бинарных ГА с ДКОП в сравнении с базовым алгоритмом без ДКОП.

Базовый алгоритм, размерность	Показатель	Унимод. (f1–f2)	Многоэстр. (f3–f9)	Гибридные (f10–f19)	Композиц. (f20–f30)	Итого
GA $D = 10$	Победы	2	7	8	10	27
	Ничьи	0	0	2	0	2
	Поражения	0	0	0	1	1
SelfCSHAGA $D = 10$	Победы	1	7	6	10	24
	Ничьи	1	0	3	0	4
	Поражения	0	0	1	1	2
SelfCSHAGA $D = 30$	Победы	1	7	9	9	26
	Ничьи	1	0	0	1	2
	Поражения	0	0	1	1	2
SelfCSHAGA $D = 50$	Победы	1	7	7	9	24
	Ничьи	1	0	2	0	3
	Поражения	0	0	1	2	3

Таблица 3 — Анализ эффективности бинарных PSO с ДКОП в сравнении с базовым алгоритмом без ДКОП.

Базовый алгоритм, размерность	Показатель	Унимод. (f1–f2)	Многоэстр. (f3–f9)	Гибридные (f10–f19)	Композиц. (f20–f30)	Итого
BPSO, $D = 10$	Победы	1	6	5	3	15
	Ничьи	1	1	5	8	15
	Поражения	0	0	0	0	0
BPSO, $D = 30$	Победы	2	5	6	10	23
	Ничьи	0	1	4	1	6
	Поражения	0	1	0	0	1
BPSO, $D = 50$	Победы	2	7	6	9	24
	Ничьи	0	0	3	2	5
	Поражения	0	0	1	0	1

Таблица 4 — Анализ эффективности бинарных L-SRTDE с ДКОП в сравнении с базовым алгоритмом без ДКОП.

Базовый алгоритм, размерность	Показатель	Унимод. (f1–f2)	Многоэкстр. (f3–f9)	Гибридные (f10–f19)	Композиц. (f20–f30)	Итого
L-SRTDE, $D = 10$	Победы	1	4	5	5	15
	Ничьи	1	3	5	6	15
	Поражения	0	0	0	0	0
L-SRTDE, $D = 30$	Победы	2	4	5	5	16
	Ничьи	0	3	5	6	14
	Поражения	0	0	0	0	0
L-SRTDE, $D = 50$	Победы	2	4	8	7	21
	Ничьи	0	2	1	3	6
	Поражения	0	0	0	1	1

Это агрегированные показатели для бинарных ГА, метода роя частиц и дифференциальной эволюции по четырём группам тестовых функций набора CEC 2017: унимодальным, многоэкстремальным, гибридным и композиционным. Для каждой группы указано число задач, в которых модификация с ДКОП продемонстрировала статистически значимое по критерию Манна-Уитни-Уилкоксона ($p < 0.05$) улучшение точности по сравнению с базовой версией алгоритма (победы), отсутствие значимых различий (ничьи) и большую эффективность базового алгоритма (ничьи).

Как показали численные эксперименты, на наборе задач IEEE CEC 2017 алгоритмы с ДКОП превосходят свои базовые алгоритмы без коррекции практически на всех задачах, что делает выбор алгоритма с ДКОП более предпочтительным при решении задач с алгоритмически заданными функциями.

Для сравнения эффективности разных метаэвристик с ДКОП результаты их решения задач были проранжированы по средним значениям целевой функции, полученным по серии независимых запусков (меньший ранг соответствует лучшему результату). Далее ранги усреднялись по всем функциям набора для каждой размерности задачи. Результаты сравнения приведены в таблице 5.

Как видно из таблицы, наилучшие значения среднего ранга на всех рассмотренных размерностях демонстрирует подход SelfCSHAGA+ДКОП. Несмотря на то, что алгоритм базовый L-SRTDE является ведущим для решения задач оптимизации с вещественными переменными, его бинарная версия существенно уступает бинарному ГА. Наиболее сильное отставание бинарных версий метода

Таблица 5 — Сводная таблица рангов алгоритмов BPSO+ДКОП, L-SRTDE (bDE)+ДКОП, SelfCSHAGA (ScGA)+ДКОП на тестовом наборе CEC2017.

Функ.	$D = 10$			$D = 30$			$D = 50$		
	BPSO	bDE	ScGA	BPSO	bDE	ScGA	BPSO	bDE	ScGA
f_1	3	2	1	2	3	1	3	2	1
f_2	3	2	1	3	2	1	2	3	1
f_3	3	2	1	1	3	2	1	3	2
f_4	2	3	1	3	2	1	3	2	1
f_5	3	1	2	3	2	1	3	2	1
f_6	3	1	2	3	1	2	3	1	2
f_7	3	1	2	3	2	1	2	3	1
f_8	3	1	2	3	2	1	2	3	1
f_9	3	2	1	1	2	3	3	2	1
f_{10}	3	1	2	2	3	1	2	3	1
f_{11}	3	2	1	3	2	1	3	2	1
f_{12}	3	2	1	3	2	1	3	2	1
f_{13}	3	2	1	3	2	1	3	2	1
f_{14}	2	3	1	3	2	1	3	2	1
f_{15}	2	3	1	2	3	1	2	3	1
f_{16}	3	2	1	3	2	1	3	2	1
f_{17}	3	2	1	3	2	1	3	1	2
f_{18}	3	1	2	3	2	1	3	2	1
f_{19}	3	1	2	1	3	2	1	3	2
f_{20}	3	2	1	3	1	2	3	2	1
f_{21}	3	2	1	3	2	1	3	2	1
f_{22}	1	2	3	1	2	3	1	2	3
f_{23}	3	2	1	3	2	1	3	2	1
f_{24}	1	3	2	3	1	2	3	2	1
f_{25}	3	2	1	3	1	2	3	2	1
f_{26}	2	1	3	2	1	3	2	1	3
f_{27}	3	2	1	3	2	1	3	1	2
f_{28}	1	3	2	2	3	1	3	2	1
f_{29}	3	1	2	3	1	2	3	2	1
f_{30}	3	1	2	3	2	1	3	2	1
Ср. ранк	2.567	1.800	1.633	2.567	2.067	1.367	2.533	2.300	1.167

роя частиц и дифференциальной эволюции наблюдается с ростом размерности, что объясняется способом бинаризации, применяемым в этих алгоритмах.

Учитывая полученные результаты, дальнейшие эксперименты проводятся с использованием самонастраиваемого ГА.

2.4 Результаты решения тестовых прикладных задач со смешанными переменными

Задача проектирования пружины сжатия/растяжения. Цель задачи оптимизации пружины сжатия/растяжения (Tension/Compression Spring Design, TCSD) — минимизация веса пружины при соблюдении ограничений на касательные напряжения, прогиб и частотные характеристики. Это классическая задача инженерной оптимизации, широко применяемая как бенчмарк при тестировании алгоритмов оптимизации. Она была включена в набор тестовых задач конкурса CEC 2006 по оптимизации с ограничениями и с тех пор остаётся популярной для сравнения различных методов [173].

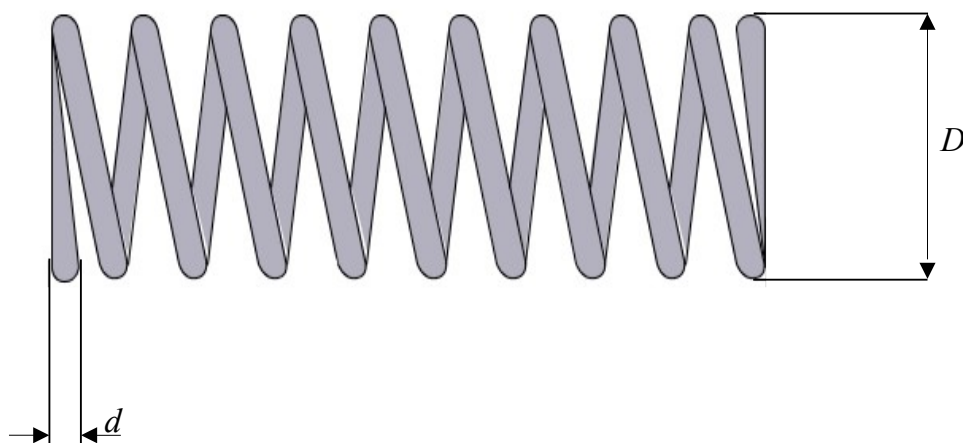


Рисунок 2.15 — Иллюстрация к задаче проектирования пружины сжатия/растяжения.

Рассматриваются три переменные $\mathbf{X} = [x_1, x_2, x_3] = [d, D, N]$. Обозначения переменных приведены в таблице 6.

Целевая функция задачи оптимизации имеет вид:

$$\min f(\mathbf{X}) = (x_3 + 2) \cdot x_2 \cdot x_1^2.$$

Таблица 6 — Переменные задачи проектирования пружины: описание, тип и границы

№	Переменная	Описание	Тип	Диапазон значений
x_1	d	диаметр проволоки	вещественная	$0.05 \leq x_1 \leq 2.00$
x_2	D	средний диаметр витка	вещественная	$0.25 \leq x_2 \leq 1.30$
x_3	N	число активных витков	целочисленная	$N \in \{2, 3, \dots, 15\}$

Ограничения задачи оптимизации имеют вид:

$$\begin{aligned}
 g_1(\mathbf{X}) &= 1 - \frac{x_3 \cdot x_2^3}{71875 \cdot x_1^4} \leq 0, \\
 g_2(\mathbf{X}) &= \frac{4x_2^2 - x_1x_2}{12566(x_1x_2^3 - x_1^4)} + \frac{1.508x_1}{12566(x_1x_2^3 - x_1^4)} - 1 \leq 0, \\
 g_3(\mathbf{X}) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 g_4(\mathbf{X}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0.
 \end{aligned}$$

Задача проектирования редуктора скорости (Speed Reducer Design, SRD) представляет собой сложную многосвязную задачу оптимизации, используемую для тестирования эффективности алгоритмов. Редуктор является ключевым элементом в трансмиссионных системах, применяемых в автомобилестроении, авиации и промышленной технике. Оптимизация конструкции позволяет повысить эффективность передачи, снизить уровень шума и уменьшить потери энергии. Задача проектирования редуктора скорости использовалась в конкурсах CEC 2006 и CEC 2010 в качестве эталонной задачи для тестирования алгоритмов оптимизации с ограничениями [174].

Цель — минимизировать массу редуктора при соблюдении ограничений по прочности шестерён, долговечности валов и геометрическим требованиям. Схема конструкции представле на на рисунке 2.16.

Целевая функция задачи оптимизации имеет вид:

$$\begin{aligned}
 \min f(\mathbf{X}) &= 0,7854 x_1 x_2^2 (3,3333 x_3^2 + 14,9334 x_3 - 43,0934) \\
 &\quad - 1,508 x_1 (x_6^2 + x_7^2) + 7,4777 (x_6^3 + x_7^3) + 0,7854 (x_4 x_6^2 + x_5 x_7^2).
 \end{aligned}$$

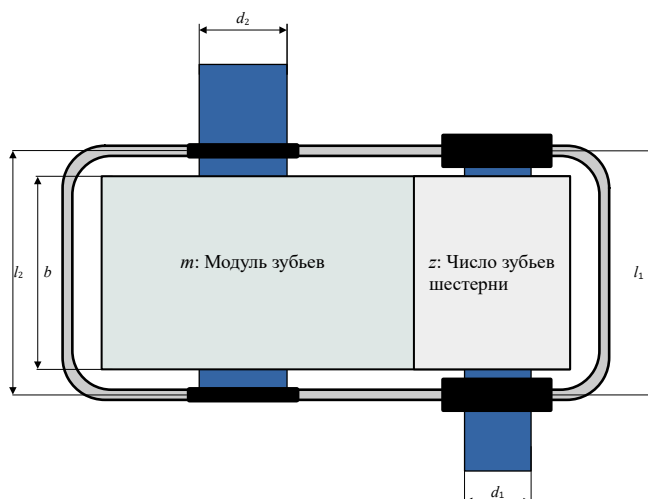


Рисунок 2.16 — Схема к задаче проектирования редуктора скорости

Переменные:

$$\mathbf{X} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2].$$

Ограничения задачи оптимизации имеют вид:

$$g_1(\mathbf{X}) = 27x_1x_2^2x_3 - 1 \leq 0,$$

$$g_2(\mathbf{X}) = 397,5x_1x_2^2x_3^2 - 1 \leq 0,$$

$$g_3(\mathbf{X}) = \frac{1,9x_4^3}{x_2x_6^4x_3} - 1 \leq 0,$$

$$g_4(\mathbf{X}) = \frac{1,93x_5^3}{x_2x_7^4x_3} - 1 \leq 0,$$

$$g_5(\mathbf{X}) = \left[\left(\frac{745(x_4/x_2x_3)}{\quad} \right)^2 + 16,9 \times 10^6 \right] / (110x_6^3) - 1 \leq 0,$$

$$g_6(\mathbf{X}) = \left[\left(\frac{745(x_5/x_2x_3)}{\quad} \right)^2 + 157,5 \times 10^6 \right] / (85x_7^3) - 1 \leq 0,$$

$$g_7(\mathbf{X}) = \frac{x_2x_3}{40} - 1 \leq 0,$$

$$g_8(\mathbf{X}) = 5x_2x_1 - 1 \leq 0,$$

$$g_9(\mathbf{X}) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(\mathbf{X}) = 1,5x_6 + 1,9x_4 - 1 \leq 0,$$

$$g_{11}(\mathbf{X}) = 1,1x_7 + 1,9x_5 - 1 \leq 0.$$

Обозначения переменных приведены в таблице 7.

Таблица 7 — Переменные задачи и их типы с границами

№	Переменная	Описание	Тип	Диапазон значений
x_1	b	ширина зубчатого венца	вещественная	$2,6 \leq x_1 \leq 3,6$
x_2	m	модуль зубьев	вещественная	$0,7 \leq x_2 \leq 0,8$
x_3	z	число зубьев шестерни	целочисленная	$x_3 \in \{17, 18, \dots, 28\}$
x_4	l_1	длина первого вала между подшипниками	вещественная	$7,3 \leq x_4 \leq 8,3$
x_5	l_2	длина второго вала между подшипниками	вещественная	$7,3 \leq x_5 \leq 8,3$
x_6	d_1	диаметр первого вала	вещественная	$2,9 \leq x_6 \leq 3,9$
x_7	d_2	диаметр второго вала	вещественная	$5,0 \leq x_7 \leq 5,5$

Задача проектирования давления на сосуд (Pressure Vessel Design Problem, PVDP) является классической инженерной задачей оптимизации, используемой в конкурсе CEC 2020 [175]. Цель — минимизировать общую стоимость изготовления цилиндрического сосуда, учитывая затраты на материал, сварку и проверку. При этом оптимизация проводится по четырём переменным: толщине корпуса и днища (дискретные переменные), внутреннему радиусу и длине корпуса (вещественные переменные). Схема задачи показана на рисунке 2.17. Задача включает нелинейные ограничения, связанные с прочностными и объёмными требованиями к сосуду.

Целевая функция задачи оптимизации имеет вид:

$$\begin{aligned} \min f(\mathbf{X}) = & 0.6224 T_s R L + 1.7781 T_h R^2 \\ & + 3.1661 T_s^2 L + 19.84 T_s^2 R. \end{aligned} \quad (2.22)$$

Обозначения переменных приведены в таблице 8.

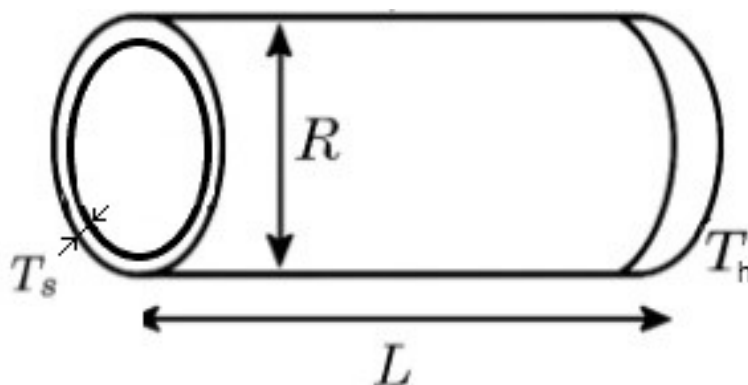


Рисунок 2.17 — Схема к задаче проектирования давления на сосуд.

Таблица 8 — Переменные задачи давления на сосуд и их типы.

№	Переменная	Описание	Тип	Границы
x_1	T_s	Толщина корпуса (shell thickness)	дискретная	$T_s \in \{1.0, 1.5, 2.0, 2.5\}$ мм
x_2	T_h	Толщина днища (head thickness)	дискретная	$T_h \in \{1.0, 1.5, 2.0, 2.5\}$ мм
x_3	R	Внутренний радиус сосуда	вещественная	$10.0 \leq R \leq 200.0$ мм
x_4	L	Длина корпуса (не включая днище)	вещественная	$10.0 \leq L \leq 200.0$ мм

Ограничения:

$$g_1(\mathbf{x}) : -T_s + 0.0193 R \leq 0,$$

$$g_2(\mathbf{x}) : -T_h + 0.00954 R \leq 0,$$

$$g_3(\mathbf{x}) : -\pi R^2 L - \frac{4}{3}\pi R^3 + 1 \times 10^4 \leq 0,$$

$$g_4(\mathbf{x}) : L - 240 \leq 0.$$

Результаты решения задач. Все вычислительные эксперименты проводились на основе 50 независимых запусков для каждой задачи оптимизации, что обеспечило статистическую достоверность полученных результатов. Для алгоритмов ГА и ГА+ДКОП использовались одинаковые базовые параметры: размер популяции $pop_size = 100$, число поколений $n_generations = 100$, вероятности кроссовера и мутации $p_c = 0.8$ и $p_m = 0.1$ соответственно, размер турнира $tournament_size = 3$.

Для модифицированного алгоритма ГА+ДКОП дополнительно применялись параметры адаптивной коррекции пространства поиска: число кластеров

$n_clusters = 5$, порог по размеру кластера $\alpha_N = 1.0$, порог по пригодности $\alpha_F = 1.0$, параметр ICE-коррекции $\theta_{ice} = 1.0$, желаемый шаг дискретизации $desired_step = 0.001$, минимально допустимый размер популяции $min_pop_size = 20$, а также размер элитного архива $elite_archive_size = 10$.

Алгоритмы SelfCSHAGA и SelfCSHAGA+ДКОП также запускались при размере популяции $pop_size = 100$ и числе поколений $n_generations = 100$, что обеспечивало корректное сравнение динамики сходимости и качества получаемых решений между всеми рассматриваемыми методами.

На рисунке 2.18 представлено количественное сравнение распределений лучших значений целевых функций, полученных SelfCSHAGA и SelfCSHAGA+ДКОП, по сериям независимых запусков для трёх инженерных задач.

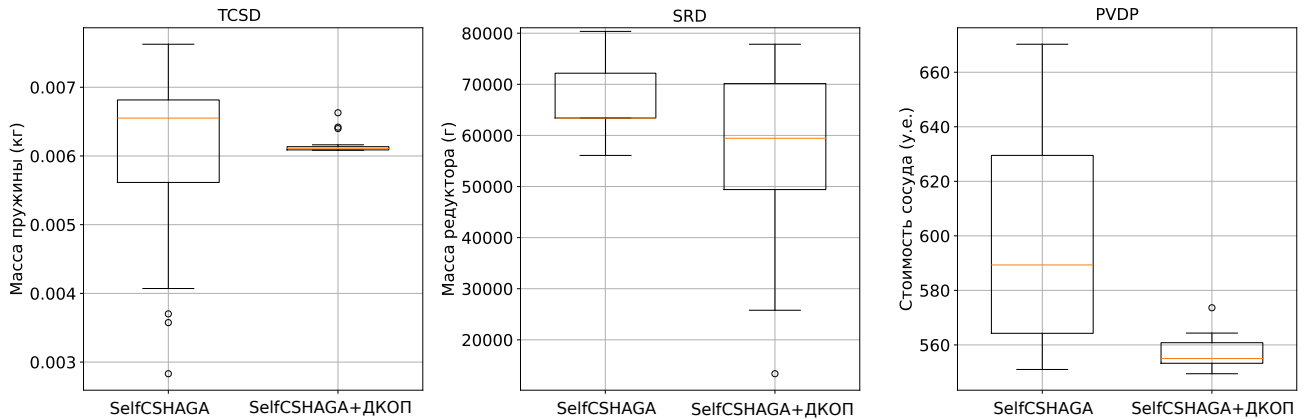


Рисунок 2.18 — Сравнение распределений лучших значений целевой функции для алгоритмов SelfCSHAGA и SelfCSHAGA+ДКОП при решении инженерных задач TCSD, SRD и PVDP.

Для задачи TCSD медианное значение целевой функции при использовании SelfCSHAGA+ДКОП оказалось ниже, чем у классического SelfCSHAGA, приблизительно на 5–7%, при одновременном снижении межквартильного размаха. В задаче PVDP применение SelfCSHAGA+ДКОП также приводит к снижению медианы целевой функции примерно на 8–12%, при этом резко сокращается число выбросов, присутствующих при использовании классического SelfCSHAGA.

На рисунке 2.19 представлено количественное сравнение распределений лучших значений целевых функций, полученных алгоритмами SelfCSHAGA и

SelfCSHAGA+ДКОП, по сериям независимых запусков для трёх инженерных задач.

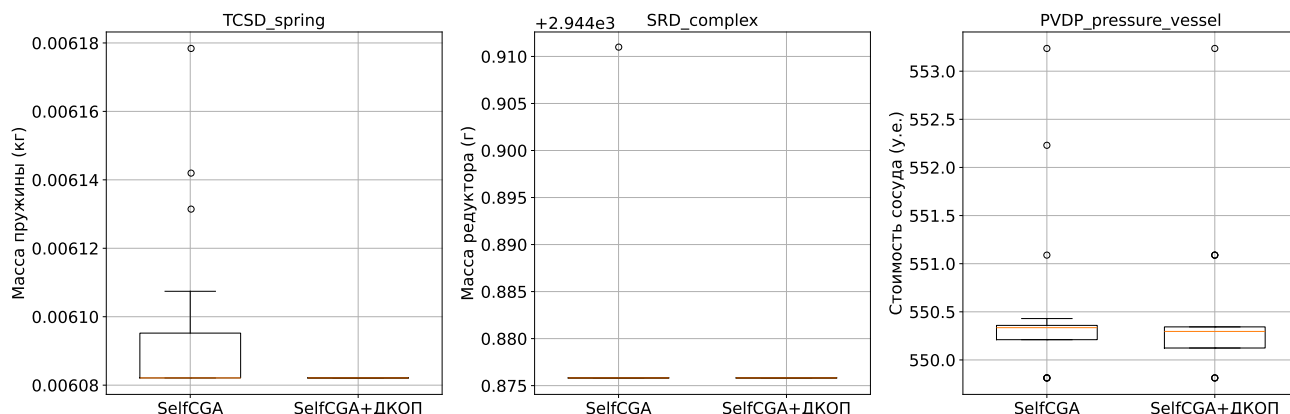


Рисунок 2.19 — Сравнение распределений лучших значений целевой функции для алгоритмов SelfCSHAGA и SelfCSHAGA+ДКОП при решении инженерных задач TCSD, SRD и PVDP.

Для задачи TCSD медианное значение целевой функции при использовании SelfCSHAGA+ДКОП оказывается таким же, как и у базового SelfCSHAGA, при этом наблюдается отсутствие межквартильного размаха. Аналогичная ситуация и для задачи SRD, при этом базовый SelfCSHAGA имеет один выброс. Для задачи PVDP применение SelfCSHAGA+ДКОП обеспечивает снижение медианного значения целевой функции на 6–9%, а также сокращение числа выбросов.

На рисунке 2.20 представлены зависимости лучших значений целевых функций поколения для трёх инженерных задач (TCSD, SRD и PVDP), полученные в результате сравнения базовых и модифицированных алгоритмов. Кривые характеризуют усреднённую по серии независимых запусков динамику сходимости.

Для всех трёх задач наблюдается ускорение сходимости при использовании алгоритмов с адаптивными механизмами. В задачах TCSD и PVDP применение ГА+ДКОП приводит к более быстрому выходу на стационарное значение целевой функции уже на ранних поколениях. Аналогичная тенденция наблюдается при сравнении SelfCSHAGA и SelfCSHAGA+ДКОП. Для всех задач внедрение кластерной обработки популяции обеспечивает ускорение начального этапа сходимости и снижение уровня финальных значений целевой функции.⁴

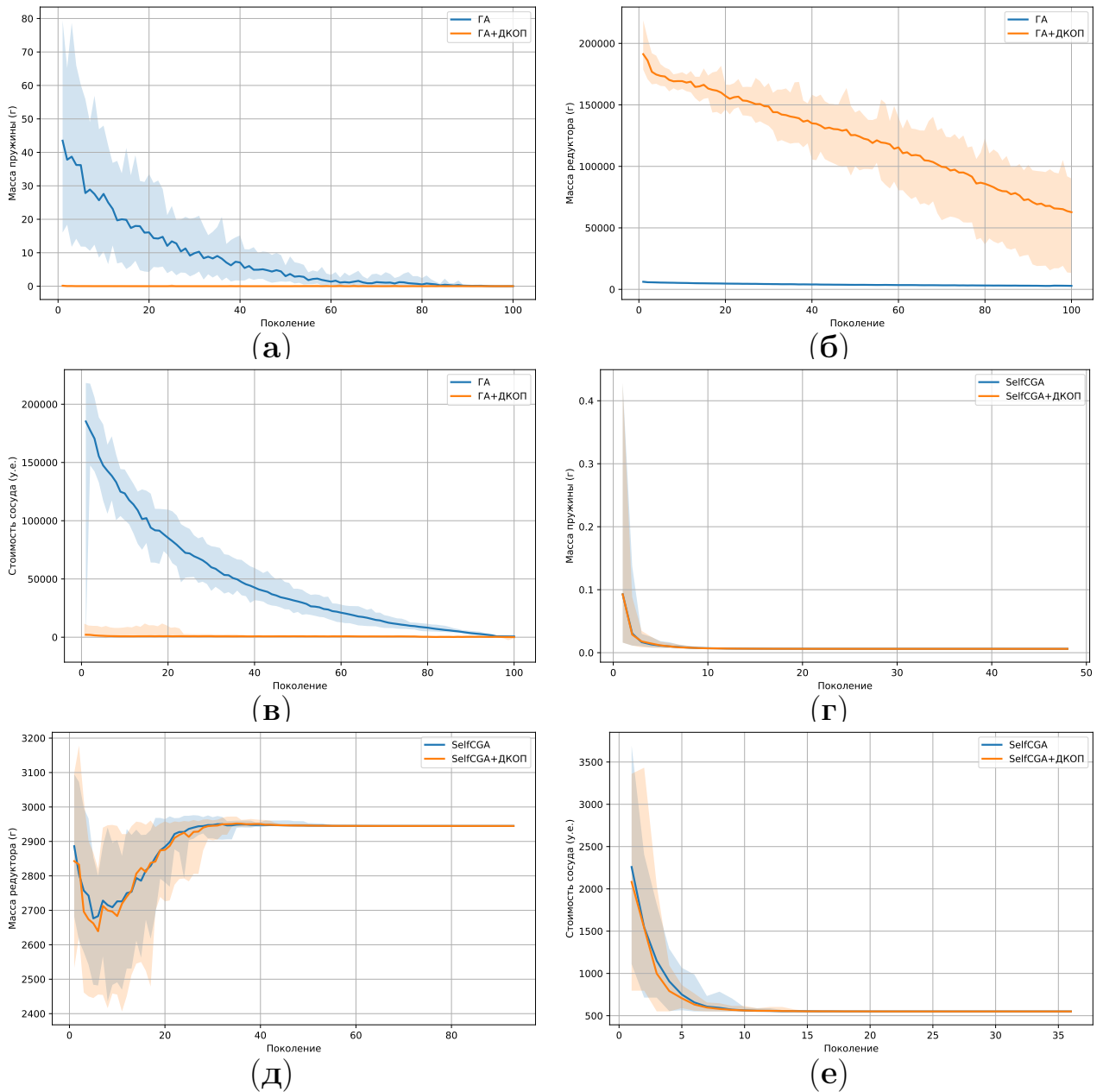


Рисунок 2.20 — Сравнение сходимости алгоритмов при решении инженерных задач: зависимости лучших значений целевых функций от поколения. (а)–(в) — сравнение ГА и ГА+ДКОП для задач TCSD, SRD и PVDP соответственно; (г)–(е) — сравнение алгоритмов SelfCSHAGA и SelfCSHAGA+ДКОП для тех же задач.

2.5 Выводы по главе 2

В главе предложен метод динамической коррекции поискового пространства для бинарного генетического алгоритма, ориентированный на задачи глобальной оптимизации со сложным многомодальным ландшафтом и ограничен-

ным вычислительным бюджетом. В основе метода лежит идея идентификации перспективных областей пространства поиска для изменения способа бинаризации вещественных переменных с целью удалить узлы сетки дискретизации из неперспективных областей и повысить точность поиска, уплотнив сетку в перспективных.

Разработан алгоритм, в котором кластеризация решений используется для выделения локальных областей, характеризующихся сходными значениями переменных. На основе статистики пригодности внутри кластеров реализована процедура исключения неэффективных областей и адаптивной корректировки границ поискового пространства. При этом направление изменения границ определяется не случайно, а на основе показателей относительной и внутрикластерной эффективности индивидов.

Введён формальный критерий целесообразности коррекции пространства поиска, основанный на сравнении занятых и свободных областей. Это позволяет избежать преждевременного сужения пространства и сохранить баланс между исследованием и использованием.

Предложена схема динамического изменения шага дискретизации и соответствующей адаптации бинарного кодирования. Это обеспечивает согласованность представления решений с текущим масштабом пространства поиска без изменения базовых генетических операторов.

Показано, что предложенный метод может быть реализован как надстройка над классическими и самоконфигурируемыми эволюционными алгоритмами, не требующая изменения их структуры. Все модификации сосредоточены на уровне представления решений и управления областью поиска. Визуализация работы алгоритма на тестовых функциях продемонстрировала его способность локализовать поиск вблизи перспективных областей и последовательно уточнять границы пространства. Проведённые вычислительные эксперименты на наборе стандартных тестовых и инженерных задач показали повышение точности решений и ускорение сходимости по сравнению с алгоритмом без динамической коррекции области поиска, а также снижение вариативности результатов в серии запусков.

3. Апробация метода динамической коррекции области поиска на практических задачах оптимизации

3.1 Оптимизация гиперпараметров моделей машинного обучения в системе классификации текстовых материалов госконтрактов

Постановка задачи. Рассматривается прикладная задача, направленная на снижение числа ошибок при присвоении кодов ОКВЭД (Общероссийский классификатор видов экономической деятельности) государственным контрактам за счёт использования методов классификации, позволяющих выявлять случаи несоответствия между текстовым содержанием контракта и присвоенным кодом экономической деятельности.

С точки зрения методов оптимизации, данную задачу можно рассматривать как поиск такой конфигурации гиперпараметров модели машинного обучения, при которой достигается наилучшее значение выбранной метрики качества классификации на валидационной или тестовой выборке. Иными словами, оптимизируемыми переменными выступают гиперпараметры классификатора, а целевая функция определяется значением метрики качества модели. В работе данная постановка используется для исследования возможностей бинарного генетического алгоритма с динамической коррекцией области поиска. В этом случае задача оптимизации формулируется в пространстве со смешанными переменными, поскольку гиперпараметры моделей машинного обучения включают как дискретные, так и непрерывные параметры. При этом аналитическое выражение целевой функции отсутствует, поскольку её значение может быть получено только в результате обучения и последующей оценки модели.

Описание набора данных и экспериментальной схемы. В рамках исследования [176] использовался набор данных, содержащий около 20 млн записей о государственной закупочной деятельности. Каждая запись включает текстовое описание контракта, стоимость, сроки исполнения и код экономической деятельности по классификатору ОКВЭД. Использование кодов ОКВЭД позволяет структурировать данные по видам экономической деятельности и анализировать распределение государственных закупок по различным секторам экономики.

Обработка текстовых данных выполнялась с использованием платформы Apache Spark, обеспечивающей распределённую обработку больших массивов данных. Предварительная обработка включала нормализацию текстов, удаление служебных символов и токенизацию. Для снижения морфологического разнообразия применялись методы стемминга и лемматизации.

После предварительной обработки тексты преобразовывались в числовые признаки с использованием моделей *bag-of-words* и TF-IDF. Векторизация выполнялась с помощью алгоритмов `CountVectorizer` и `TfidfVectorizer`. В обоих случаях применялись одинаковые ограничения на словарь признаков. Эти ограничения уменьшали размерность признакового пространства, исключали слишком редкие слова и удаляли слова, встречающиеся в большинстве документов. Такой подход снижал влияние случайного шума, опечаток и общеупотребимых слов, которые слабо помогают различать классы. Используемые параметры векторизации приведены в таблице 9.

Таблица 9 — Параметры векторизации текстовых описаний

Параметр	Назначение параметра	Count	TF-IDF
<code>max_features</code>	Ограничивает размер словаря словами, наиболее часто встречающимися в корпусе после предварительной обработки; снижает размерность матрицы признаков.	1500	1500
<code>min_df</code>	Исключает слова, встречающиеся менее чем в пяти документах; удаляет редкие слова, опечатки и единичные обозначения.	5	5
<code>max_df</code>	Исключает слова, встречающиеся более чем в 70% документов; удаляет слишком общие слова с низкой различающей способностью.	0.7	0.7

Полученные векторные представления текстовых описаний контрактов использовались в задаче многоклассовой классификации. В качестве целевой переменной выступал код экономической деятельности по классификатору ОКВЭД, который рассматривался как метка класса.

Для решения задачи применялись модели машинного обучения, обучаемые на векторизованных текстовых признаках. Подбор гиперпараметров моделей выполнялся с использованием SelfCSHAGA с ДКОП.

В данной постановке каждый индивид SelfCSHAGA соответствует набору гиперпараметров модели машинного обучения, а значение функции пригодности определяется показателями качества классификации на валидационной выборке. Таким образом, SelfCSHAGA используется для поиска конфигураций

гиперпараметров, обеспечивающих наилучшие значения метрик качества моделей. Общая схема экспериментального исследования представлена на рисунке 3.1.

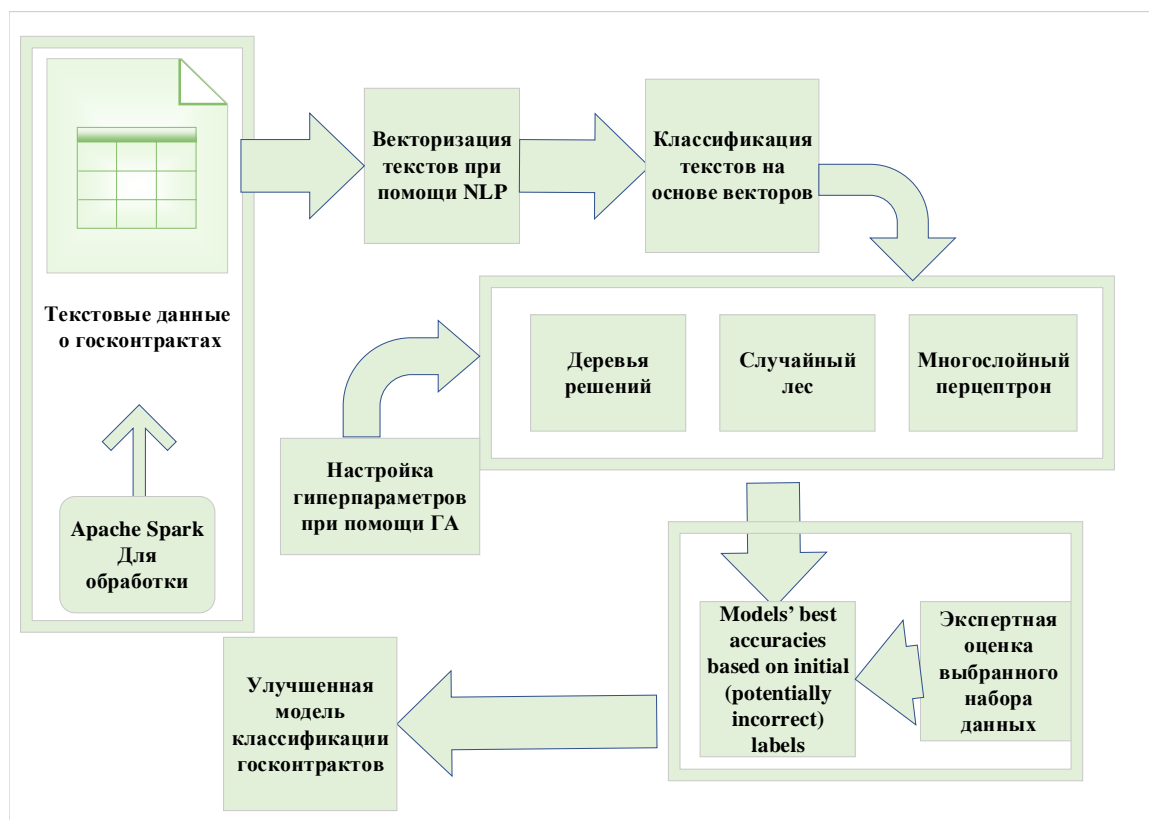


Рисунок 3.1 — Экспериментальная схема обработки данных и обучения моделей.

Оптимизация параметров классификатора с использованием SelfCSHAGA. Основным критерием эффективности моделей являлась точность классификации. Сравнение методов векторизации показало близкие значения точности, при этом TF-IDF обеспечивал более высокую скорость преобразования текстов. По этой причине для дальнейших экспериментов использовалось представление TF-IDF.

Эксперименты проводились с алгоритмами Naive Bayes, Decision Tree, Random Forest и Multilayer Perceptron для классификации экономической деятельности по кодам ОКВЭД. Подбор гиперпараметров выполнялся с использованием SelfCSHAGA. В качестве фитнес-функции использовалась точность классификации на валидационной выборке. Вероятность применения оператора скрещивания к паре родительских индивидов составляла 0.5, а вероятность применения оператора мутации – 0.1. В качестве оператора скрещивания использовалось одноточечное скрещивание, при котором две родительские хромо-

сомы обмениваются фрагментами после случайно выбранной позиции. Мутация выполнялась как случайная замена значения гена на другое допустимое значение из пространства гиперпараметров соответствующей модели.

SelfCSHAGA в данном эксперименте использовался на внешнем уровне и выполнял подбор гиперпараметров классификаторов. Каждый индивид популяции задавал одну конфигурацию модели. Для вычисления значения фитнес-функции соответствующая модель строилась на обучающей выборке, после чего качество классификации оценивалось на валидационной выборке.

Классификатор Naive Bayes показал точность 66% при малом времени обучения. Модель Decision Tree после подбора гиперпараметров генетическим алгоритмом достигла точности 62.7%. Наилучший результат среди рассмотренных моделей был получен для Random Forest. Для тематической подвыборки государственных контрактов, относящихся к строительству и монтажным работам по соответствующим кодам ОКВЭД, конфигурация с 48 деревьями, максимальной глубиной 93, минимальным числом объектов для разбиения 94 обеспечила точность 71%.

Модель MLP достигла точности 71% при значении F1-score 0.73. Используемая конфигурация включала один скрытый слой из 100 нейронов, функцию активации ReLU, размер батча 200 и 200 эпох обучения. Веса нейросети обучались с использованием оптимизатора Adam. Параметр $\alpha = 0.1$ задавал коэффициент L_2 -регуляризации весов, а параметры $\beta_1 = 0.9$ и $\beta_2 = 0.999$ определяли коэффициенты экспоненциального сглаживания оценок первого и второго моментов градиента в Adam.

Рисунок 3.2 иллюстрирует динамику изменения гиперпараметров и точности моделей (Decision Tree, Random Forest и MLP) в процессе оптимизации с использованием SelfCSHAGA.

Результаты экспериментов показали, что модель MLP обеспечивает наилучшие показатели, достигая точности 0.71 и F1-score 0.73. Деревья решений, несмотря на несколько более низкие результаты, обладает преимуществом интерпретируемости, что полезно для анализа ошибок классификации и понимания логики принятия решений. В дальнейшем, для обеспечения максимальной точности системы, в качестве основного инструмента классификации экономической деятельности использовался Multilayer Perceptron.

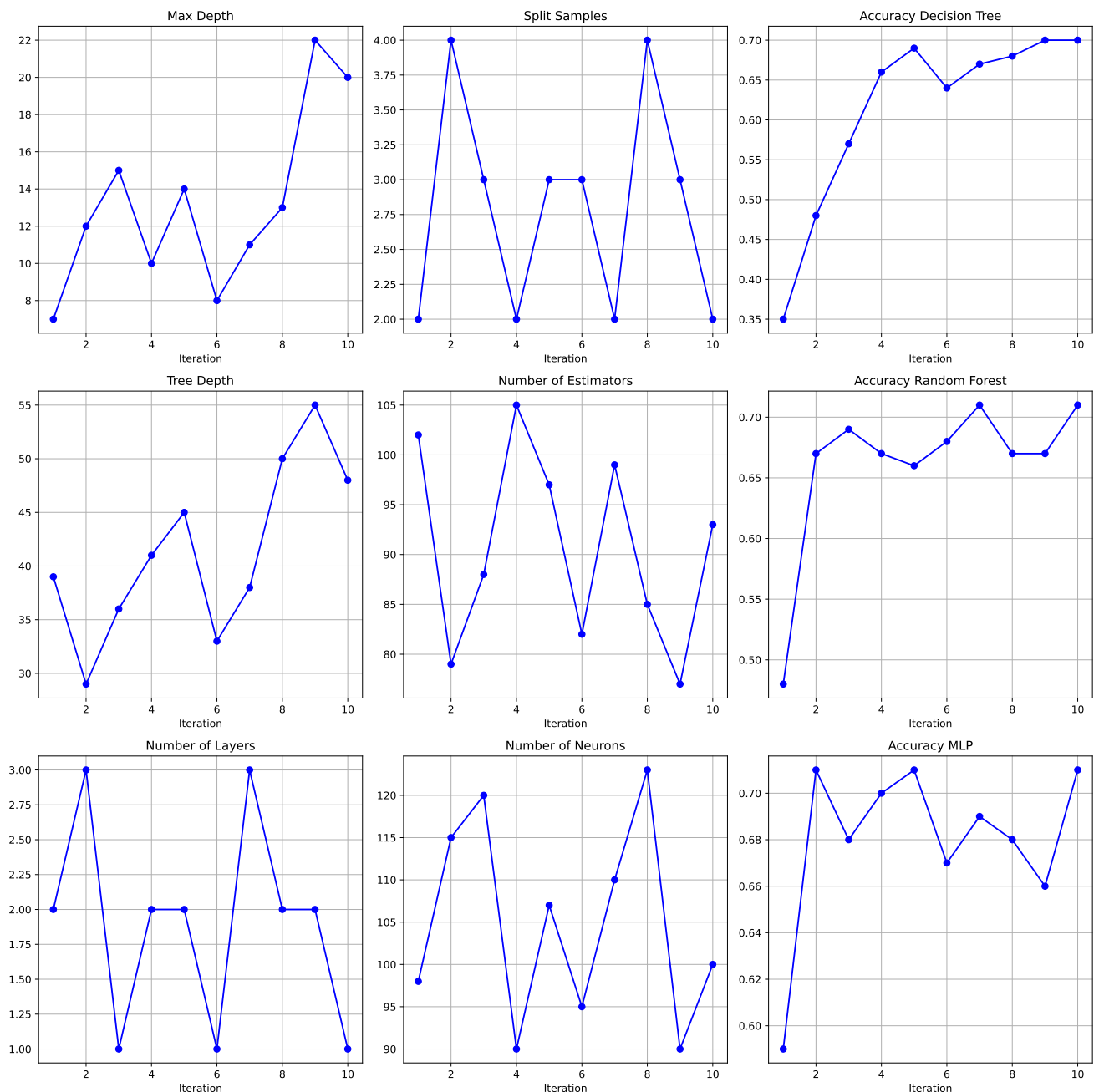


Рисунок 3.2 — Эволюция гиперпараметров и точности для классификаторов Decision Tree (верхний ряд), Random Forest (средний ряд) и MLP (нижний ряд) в ходе оптимизации с использованием SelfCSHAGA.

При разработке модели классификации выявлено большое число неправильно размеченных объектов деятельности [177]. Основные причины неправильного присвоения кодов:

- случайная ошибка при выборе кода,
- намеренный выбор неверного кода,
- выбор кода на основе недостаточной информации.

Таблица 10 приводит примеры описаний экономической деятельности с исходными и предсказанными кодами ОКВЭД, а также возможные причины расхождений.

Таблица 10 — Примеры неверно размеченных описаний экономической деятельности

Описание деятельности	Исх. ОКВЭД	Описание исх. кода ОКВЭД	Пред. ОКВЭД	Описание пред. кода ОКВЭД	Тип расхо-ия
Выполнение ремонтных работ на фасаде	41.2	Строительно-монтажные работы	43.3	Отделочные работы в зданиях и сооружениях	Намеренный выбор неверного кода
Пожарная база в деревне	42.1	Строительство дорог	26.3	Средства связи	Отсутствие описания
Выполнение работ по изготовлению и установке противопожарных дверей	43.2	Электротехнич. и прочие монтажные работы	43.3	Отделочные работы в зданиях и сооружениях	Ошибка выбора кода
Выполнение работ по устройству наклонного лифта для маломобильных групп населения	43.2	Электротехнич. и прочие монтажные работы	43.9	Прочие специализированные строительные работы	Ошибка выбора кода
Аварийное обслуживание в 2019 году	43.2	Электротехнич. и прочие монтажные работы	33.1	Ремонт металлоизделий, машин и оборудования	Отсутствие описания
Аренда специальной техники	43.9	Прочие специализированные строительные работы	68.2	Аренда объектов недвижимости	Намеренный выбор неверного кода

Например, описание контракта «Пожарная база в деревне» не содержит информации о характере деятельности (ремонт или закупка), а описание «Выполнение работ по изготовлению и установке противопожарных дверей» является ошибочным выбором кода, поскольку предсказанный код оказался более корректным. Это подчёркивает, что при обработке больших наборов данных

машинное обучение может обеспечить более высокую точность, чем ручная разметка.

Первая группа ошибочно размеченных объектов невелика и не требует отдельной обработки. Для второй группы целесообразно выявлять закономерности, связанные со стоимостью и сроками исполнения контрактов, поскольку некорректный выбор кода может быть обусловлен личной выгодой автора. В третьей группе, где код определяется на основании наиболее подходящего текстового описания, значительное количество ошибочно размеченных данных может снизить точность обучения модели.

В связи с этим внутри каждой группы, соответствующей отдельному коду ОКВЭД, выделялись кластеры записей с признаками возможной ошибки или неоднородности. Для решения данной задачи использована библиотека `pysubgroups` [178] — инструмент для выявления лексических закономерностей в текстовых описаниях контрактов. Подгруппы, представляющие собой подмножества данных, выделенные по специфическим признакам, позволяют анализировать правила (например, слова и устойчивые фразы в описаниях контрактов) и оценивать их соответствие присвоенному коду [179]. Например, в рамках одной подгруппы могут объединяться записи, относящиеся к строительной деятельности и обладающие сходными характеристиками, включая сезонность выполнения работ и тип оказываемых услуг.

Разбиение видов экономической деятельности на подгруппы [180] позволяет учитывать внутреннюю неоднородность крупных категорий ОКВЭД и выявлять группы записей со сходными текстовыми и классификационными признаками (например, «Технические консультационные услуги», «Строительный надзор», «Архитектурно-инженерные услуги»). Кроме того, анализ подгрупп позволяет корректировать неправильно размеченные объекты, формируя систему правил или модифицируя модели для более точного распознавания фактических категорий [181].

Эксперты оценили, насколько описание каждой деятельности госконтрактов, связанных со строительными и монтажными работами, соответствует ожидаемому значению кода ОКВЭД. Оценки использовались для удаления выбросов из данных (например, деятельности с длительностью свыше 100 лет или стоимостью менее 1 рубля). При анализе данных были выявлены следующие особенности:

— В группах кода 42.1 обнаружены описания с фразами, указывающими на ремонт дорожного покрытия (слова «обслуживание» и «дорога И обслуживание»), что не соответствует данной категории.

— В группах кода 41.1 зафиксированы описания, связанные с контролем и надзором за строительством, что противоречит назначению данного кода.

— Группы кода 41.2 включают деятельности, связанные с закупками, а не со строительством; описания, содержащие ключевое слово «закупка», отнесены к разнородной категории.

— Деятельность по коду 42.2 разделена на ремонт и строительство: подгруппы по правилам «ремонт» или «ремонт И капитальный» выделены отдельно, а оставшиеся объекты относятся к строительству. Несколько описаний со словами «технологический И подключение» отнесены к коду 43.2.

— Подгруппы кода 43.1, содержащие слово «благоустройство», отнесены к коду 42.9, а описания с ключевыми словами «дерево», «дерево И территория», «аварийное И дерево» и «поставка» классифицированы как разнородные.

— Для кода 43.2 выделена отдельная группа по правилу «сигнализация», не соответствующая анализируемому коду, и она также была отнесена к разнородной категории.

Оптимизация параметров классификаторов с использованием SelfCSHAGA+ДКОП. Для настройки моделей машинного обучения использовался SelfCSHAGA с динамической коррекцией области поиска (SelfCSHAGA+ДКОП). Набор данных разделялся на обучающую и тестовую выборки, а целевой переменной y выступали коды экономической деятельности (ОКВЭД). Каждый индивид SelfCSHAGA представлял набор гиперпараметров модели, а значение функции пригодности определялось взвешенной метрикой качества классификации

$$f(\theta) = F1_{\text{weighted}}(y_{\text{true}}, y_{\text{pred}}).$$

Оптимизация выполнялась в пространстве гиперпараметров моделей машинного обучения.

Для модели дерева решений оптимизировался вектор параметров

$$\theta = \{d, s, l, c\},$$

где d — максимальная глубина дерева ($2 \leq d \leq 100$), s — минимальное число объектов для разбиения ($2 \leq s \leq 50$), l — минимальное число объектов в листе ($1 \leq l \leq 20$), c — параметр выбора критерия разбиения (gini или entropy).

Каждый кандидат из популяции задавал конфигурацию модели, которая обучалась на обучающей выборке и оценивалась по F1-score. SelfCSHAGA осуществлял поиск гиперпараметров, максимизирующих функцию пригодности.

Результаты оптимизации представлены на рисунке 3.3. Наблюдается устойчивый рост значения F1-score по поколениям. Средние значения метрики увеличиваются примерно с 0.69–0.71 в начальных поколениях до 0.76–0.762 в поздних поколениях, что свидетельствует об эффективной адаптации параметров модели.

Для ансамбля случайного леса оптимизировался набор

$$\theta = \{d, s, l, c, n\},$$

где n — число деревьев в ансамбле ($10 \leq n \leq 200$). Остальные параметры соответствуют параметрам базового дерева решений.

Результаты оптимизации показаны на рис. 3.4. Значение метрики F1 увеличивается по поколениям примерно с 0.71–0.72 до 0.79.

Для нейросетевой модели оптимизировалась архитектура многослойного перцептрона

$$\theta = \{L, N, \eta\},$$

где L — число скрытых слоёв ($1 \leq L \leq 5$), N — число нейронов в слое ($8 \leq N \leq 128$), η — скорость обучения ($0.001 \leq \eta \leq 0.1$).

Архитектура сети задавалась как

$$\text{hidden_layer_sizes} = (N, N, \dots, N),$$

где число повторений равно L . Функция активации — ReLU, оптимизатор — Adam.

Результаты оптимизации представлены на рис. 3.5. Использование SelfCSHAGA+ДКОП позволяет повысить значение метрики F1 до 0.81–0.816 для ряда конфигураций.

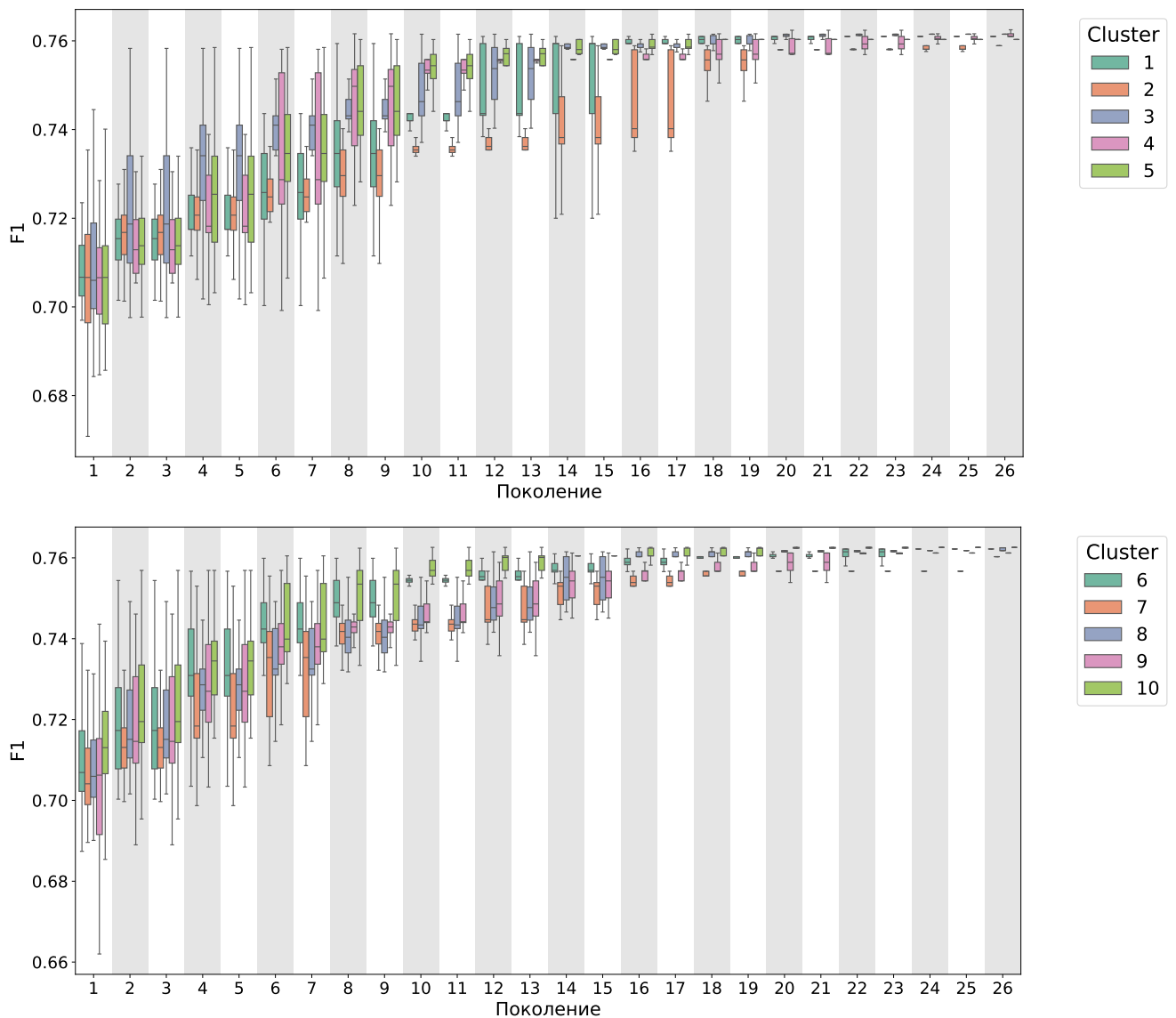


Рисунок 3.3 — Диаграмма размаха метрики F1 по поколениям для кластеров 1–5 (верхний рисунок) и 6–10 (нижний рисунок) при оптимизации параметров деревьев решений с использованием SelfCSHAGA+ДКОП.

В таблице 11 представлены результаты сравнения моделей машинного обучения, гиперпараметры которых оптимизировались SelfCSHAGA и SelfCSHAGA+ДКОП.

Таблица 11 — Сравнительные результаты методов классификации.

Метод классификации	SelfCSHAGA		SelfCSHAGA+ДКОП	
	Точность	F1-score	Точность	F1-score
Decision Tree	71%	0.72	75%	0.76
Random Forest	76%	0.77	79%	0.80
Multilayer Perceptron	79%	0.80	81%	0.82

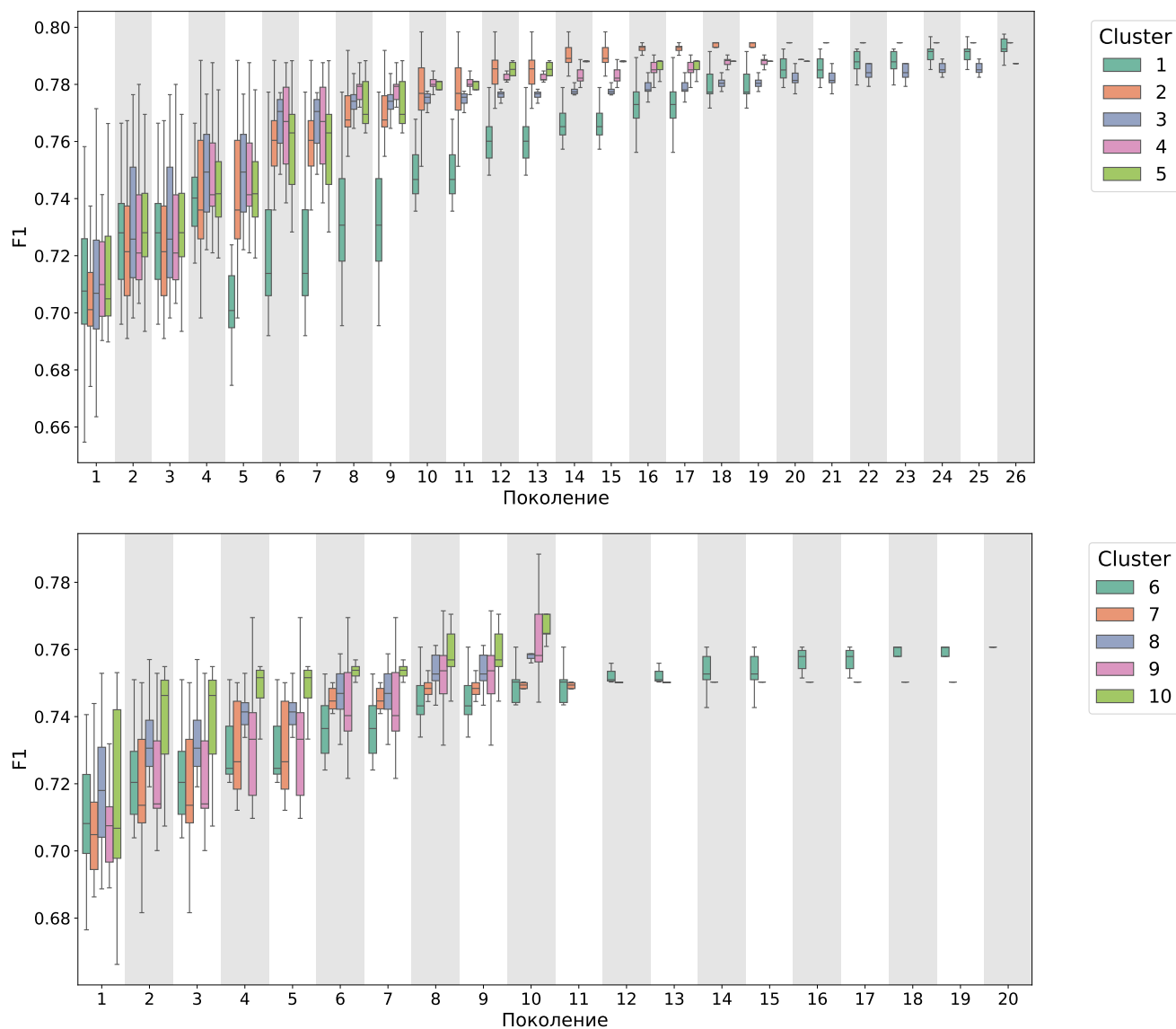


Рисунок 3.4 — Диаграмма размаха метрики F1 по поколениям для кластеров 1–5 (верхний рисунок) и 6–10 (нижний рисунок) при оптимизации параметров случайного леса с использованием SelfCSHAGA+ДКОП.

На первый взгляд, разница в точности кажется незначительной. Однако, следует учитывать, что рассматриваемая выборка государственных контрактов составляет порядка 2.5 млн записей. Наибольшее улучшение показала модель MLP, где оптимизация гиперпараметров SelfCSHAGA+ДКОП позволила увеличить точность классификации на 3%.

Согласно статистике экспертной оценки, соотношение верно классифицированных контрактов к ошибочно закодированным, но корректно выявленным моделью, составляет 33 429 к 3 744. Это даёт долю:

$$\frac{3744}{3744 + 33429} \approx 10\%. \quad (3.1)$$

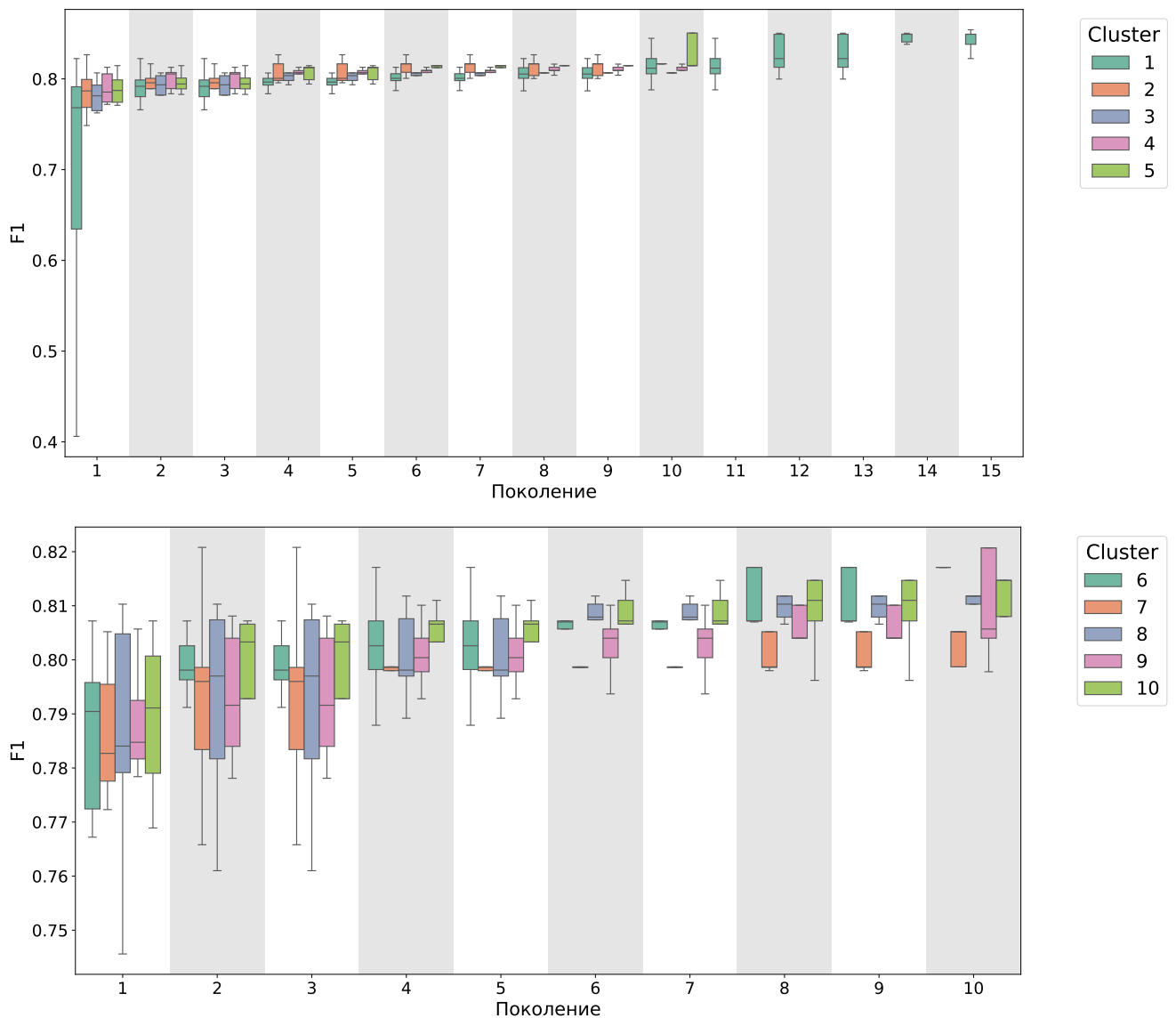


Рисунок 3.5 — Диаграмма размаха метрики F1 по поколениям для кластеров 1–5 (верхний рисунок) и 6–10 (нижний рисунок) при оптимизации параметров многослойного перцептрона с использованием SelfCSHAGA+ДКОП.

Применяя этот прирост производительности к общей выборке, можно ожидать, что дополнительное количество верно классифицированных контрактов составит порядка 100 000.

Среди этих контрактов, по оценкам экспертов, выявлено около 10 000 потенциально фиктивных государственных контрактов. Учитывая, что средняя стоимость одного контракта составляет порядка 100 тыс. рублей, применение модели позволит предотвратить нецелевое расходование до 1 млрд рублей бюджетных средств.

3.2 Задача оптимизации гиперпараметров модели бинарной классификации несбалансированных данных юридических лиц

Постановка задачи. Рассматривается прикладная задача, связанная с поддержкой принятия решений при проведении налогового контроля в отношении юридических лиц – камеральных проверок, состоящих в контроле за соблюдением законодательства о налогах и сборах на основе представленных налогоплательщиком деклараций, расчётов и документов, а также данных, которые есть у налогового органа [182; 183]. В условиях большого числа организаций и ограниченности ресурсов налоговых органов, возникает необходимость предварительного отбора объектов, для которых дополнительная проверка является наиболее целесообразной. Практический смысл задачи состоит в снижении числа необоснованных проверок при сохранении способности выявлять юридические лица с признаками повышенного риска. Такая постановка представляет интерес как с точки зрения повышения эффективности контрольных процедур, так и с точки зрения уменьшения трудозатрат на ручную обработку наблюдений, не содержащих существенных отклонений.

В прикладной постановке требуется по совокупности признаков, характеризующих юридическое лицо, определить, относится ли оно к классу организаций, требующих дополнительной налоговой проверки. Тем самым исходная предметная задача переводится в задачу бинарной классификации, в которой каждому объекту ставится в соответствие одна из двух меток: проверка не требуется или проверка требуется. Сложность данной задачи обусловлена несбалансированностью классов, перекрытием распределений признаков и повышенной ценой ошибок классификации, поскольку как необоснованное назначение проверки, так и пропуск рискованного объекта приводят к нежелательным практическим последствиям.

Данная задача может быть сформулирована как задача оптимизации, а именно как задача поиска такой конфигурации гиперпараметров модели машинного обучения, при которой достигается наилучшее значение выбранной метрики качества классификации на валидационной и/или тестовой выборке. В данной работе в качестве такой модели используется XGBoost, а вектор оптимизируемых переменных включает параметры, определяющие сложность ансамбля,

скорость обучения и долю используемых наблюдений и признаков. Значение целевой функции не задаётся аналитически и может быть получено только в результате обучения модели и её последующей оценки, что переводит задачу в класс вычислительно затратных задач оптимизации со смешанным дискретно-непрерывным пространством поиска.

Описание данных и методов. Работы [184; 185] рассматривают признаки, формируемые на основе данных косвенных налоговых проверок, не требующих непосредственной инспекции юридического лица (ЮЛ). На этапе предварительной обработки категориальные признаки были преобразованы методом One-Hot Encoding, а числовые признаки нормализованы. После этого были обучены модели машинного обучения для решения задачи бинарной классификации.

Набор данных содержит 97 095 записей, каждая из которых соответствует юридическому лицу, проверенному налоговыми органами во второй половине 2023 года. Для 75 851 организаций проверка оказалась ненужной, а для 21 243 потребовалась дополнительная проверка, что указывает на несбалансированность данных. Всего используется 56 признаков, включая 10 бинарных и 26 количественных. Целевая переменная отражает необходимость проведения налоговой проверки.

Для анализа структуры данных применена визуализация t-SNE [186]. Рисунок 3.6 показывает распределение классов целевой переменной. Наблюдается значительное перекрытие классов, что указывает на сложность задачи классификации.

Распределения бинарных признаков представлены на рисунке 3.7. Большинство признаков характеризуются выраженной несбалансированностью, что усложняет обучение моделей.

Для построения классификатора используется градиентный бустинг XGBoost. Оптимизация гиперпараметров выполняется генетическим алгоритмом. Каждый кандидат в популяции представляет вектор параметров

$$\theta = \{d, \eta, n, \gamma, s, c\},$$

где d — глубина деревьев, η — скорость обучения, n — число деревьев, γ — порог разбиения, s — доля обучающей выборки, c — доля признаков.

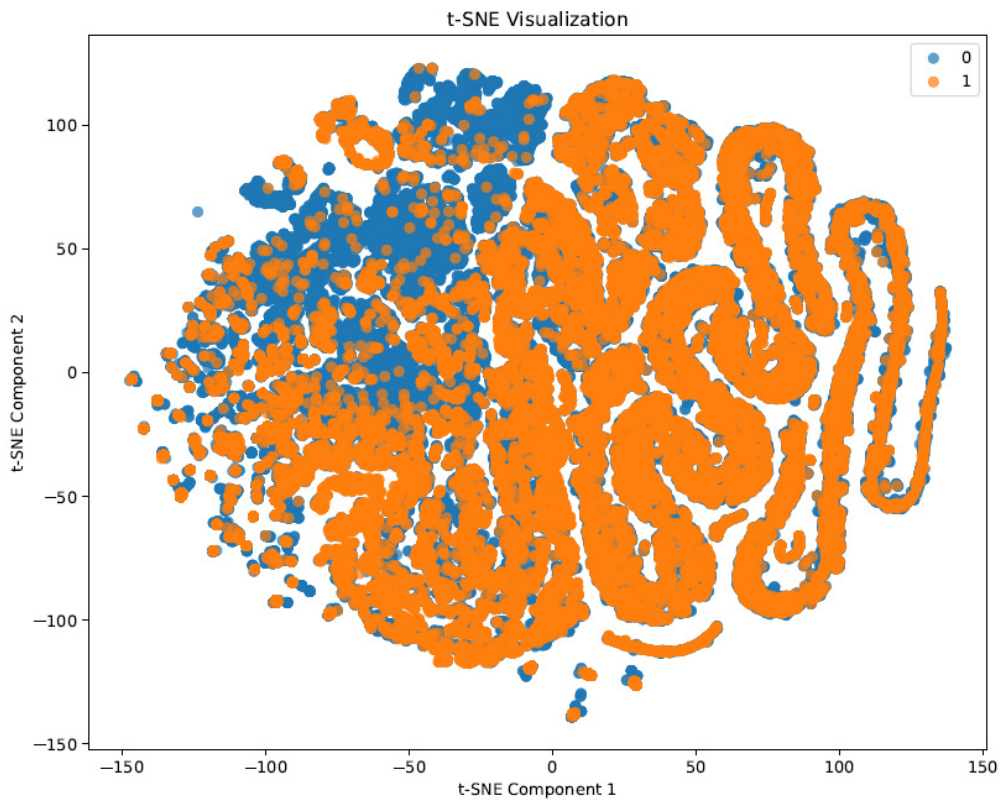


Рисунок 3.6 — Визуализация t-SNE для классов целевой переменной (0 — проверка не требуется, 1 — проверка требуется).

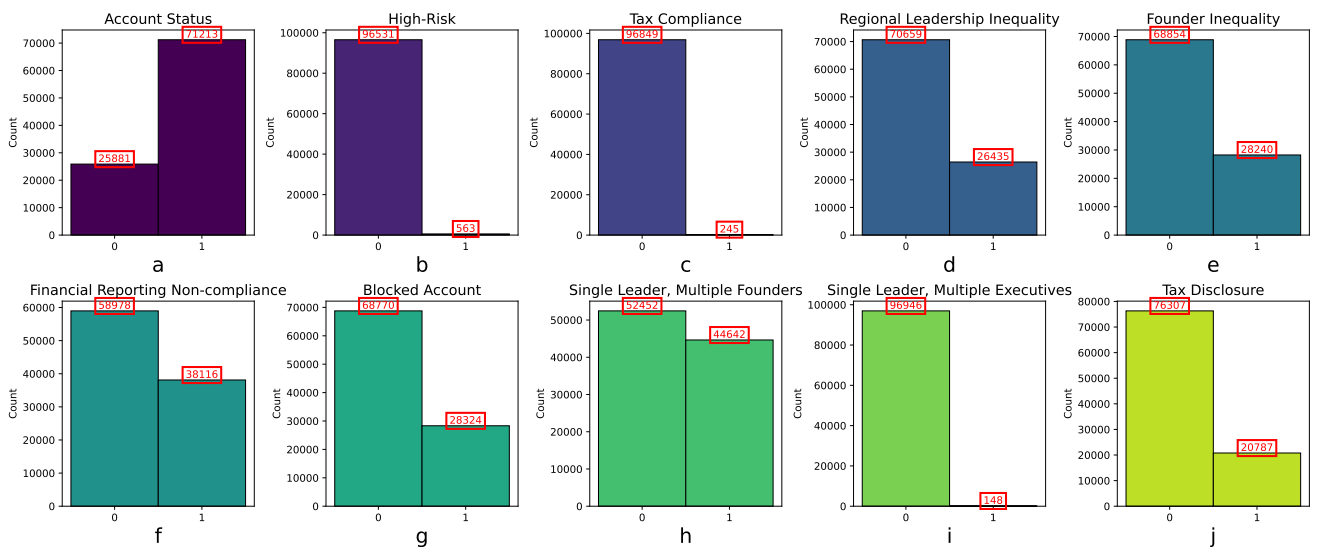


Рисунок 3.7 — Распределения бинарных признаков набора данных.

Качество модели оценивается по взвешенной F1-мере, а оптимизация проводится в течение нескольких поколений SelfCSHAGA. Такой подход позволяет находить эффективные конфигурации параметров без полного перебора пространства гиперпараметров.

В задачах бинарной классификации, включая прогнозирование необходимости налогового аудита, дополнительно используется настройка порогов уверенности и введение класса отклонений, что позволяет учитывать неопределённость предсказаний и снижать риск ошибочных решений.

Оптимизация гиперпараметров XGBoost SelfCSHAGA.

Это исследование рассматривает задачу бинарной классификации с целью максимизации F1-меры на тестовом наборе данных с использованием алгоритма XGBoost, оптимизированного с помощью SelfCSHAGA.

Пусть $X \in \mathbb{R}^{n \times m}$ представляет собой матрицу признаков, где n — количество образцов, а m — количество признаков. Вектор меток классов обозначается как $y \in \{0, 1\}$. Множество параметров модели XGBoost определяется как $\theta = \{n_{\text{estimators}}, \eta, d, s, c\}$, где:

- $n_{\text{estimators}}$ — количество деревьев в ансамбле,
- η — коэффициент обучения,
- d — максимальная глубина деревьев,
- s — коэффициент подвыборки обучающих примеров,
- c — коэффициент подвыборки признаков.

Следующие параметры выбираются для оптимизации гиперпараметров в рамках SelfCSHAGA:

- Количество деревьев ($n_{\text{estimators}}$):

$$n_{\text{estimators}} \sim \text{Uniform}(50, 500). \quad (3.2)$$

Этот параметр определяет общее количество деревьев, используемых в модели [187], с диапазоном от 50 до 500, чтобы сбалансировать гибкость и риск переобучения.

- Коэффициент обучения (η):

$$\eta \sim \text{Uniform}(0.01, 0.3). \quad (3.3)$$

Этот параметр управляет размером шага при обновлении весов, меньшие значения способствуют стабильному сходу [188]. Определенный диапазон поз-

воляет эффективно обучать модель без быстрой сходимости к локальным минимумам.

– Максимальная глубина деревьев (d):

$$d \sim \text{Integer}(3, 15). \quad (3.4)$$

Параметр максимальной глубины регулирует сложность модели [189], где более глубокие деревья могут захватывать более сложные зависимости, но могут привести к переобучению. Выбранный диапазон сбалансирован для обеспечения производительности с хорошей обобщающей способностью.

– Коэффициент подвыборки (s):

$$s \sim \text{Uniform}(0.5, 1). \quad (3.5)$$

Этот параметр определяет долю обучающих данных, используемых для построения каждого дерева [190], значения между 0.5 и 1 способствуют диверсификации обучения и улучшают обобщающие способности модели.

– Коэффициент подвыборки признаков (c):

$$c \sim \text{Uniform}(0.5, 1). \quad (3.6)$$

Этот параметр определяет долю признаков, используемых для каждого дерева, что помогает уменьшить корреляцию между деревьями в ансамбле [191]. Диапазон от 0.5 до 1 позволяет эффективно использовать признаки для обучения, сохраняя обобщение.

Цель оптимизации – максимизация F_1 -меры, определяемой как гармоническое среднее точности (Precision) и полноты (Recall):

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}. \quad (3.7)$$

Здесь $\text{Precision} = \frac{TP}{TP+FP}$ характеризует точность классификации положительного класса, а $\text{Recall} = \frac{TP}{TP+FN}$ — полноту обнаружения положительных объектов. Величины TP , FP и FN обозначают соответственно число истинно положительных, ложноположительных и ложноотрицательных классифика-

ций. Таким образом, значение F_1 определяется соотношением между TP , FP и FN .

SelfCSHAGA применяется для оптимизации параметров θ . Процесс включает следующие стадии:

1. Генерация начальной популяции P из N индивидов, каждый из которых представляет собой уникальный набор параметров:

$$P = \{\theta_1, \theta_2, \dots, \theta_N\}. \quad (3.8)$$

2. Каждый индивид θ_i проходит оценку пригодности для вычисления F_1 -меры:

$$f(\theta_i) = F_1(y, \hat{y}(X; \theta_i)), \quad (3.9)$$

где $\hat{y}(X; \theta_i)$ — предсказания, полученные с использованием модели XGBoost на наборе данных X с параметрами θ_i .

3. Отбор лучших индивидов из популяции на основе их оценок пригодности.

4. Создание потомков через кроссовер двух родительских индивидов:

$$\theta_{\text{child}} = \text{crossover}(\theta_{\text{parent1}}, \theta_{\text{parent2}}). \quad (3.10)$$

5. Внесение случайных изменений в параметры потомков:

$$\theta_{\text{mutated}} = \theta_{\text{child}} + \text{mutation}(\sigma), \quad (3.11)$$

где σ — стандартное отклонение, контролирующее размер мутации.

6. Обновление популяции заменой старой популяции на новую, повторяя процесс до достижения заранее заданного числа поколений или критериев остановки.

По завершении алгоритма, оптимальный набор параметров θ^* будет получен, что приведет к максимальной F_1 -мере на тестовом наборе данных:

$$\theta^* = \arg \max_{\theta} f(\theta), \quad (3.12)$$

где $f(\theta)$ — функция пригодности на основе F_1 -меры.

Настройка порога доверия и класс отклонений. Для несбалансированных наборов данных с наличием ложноположительных и ложноотрицательных результатов, предлагается расширить практическое применение классификатора за счёт введения класса отклонений. В этом случае задача рассматривается как задача классификации с возможностью отказа от принятия решения. Такой подход позволяет использовать оптимальные параметры модели XGBoost для выявления ситуаций, в которых уверенность предсказания недостаточна. Т.е цель – минимизировать число объектов, попадающих в класс отклонений и тем самым увеличивать область применимости модели.

Пороги, определяющие класс отклонений, подбираются эмпирически с целью оценки того, как распределение объектов между классами изменяется при использовании различных моделей. Такой подход позволяет выбирать значения порогов и анализировать условия, при которых число объектов, относимых к классу отклонений, уменьшается.

В стандартной задаче бинарной классификации модель оценивает вероятность $P(y = 1|x)$, где $y \in \{0,1\}$ обозначает метку класса (0 – аудит не требуется, 1 – аудит требуется), а x – вектор признаков. В базовом варианте порог принятия решения устанавливается равным 0.5:

$$\hat{y} = \begin{cases} 1, & \text{если } P(y = 1|x) \geq 0.5, \\ 0, & \text{если } P(y = 1|x) < 0.5. \end{cases} \quad (3.13)$$

Изменение порога решения позволяет регулировать баланс между точностью и полнотой классификации. При использовании порога $\tau > 0.5$ для предсказания положительного класса требуется более высокая уверенность модели, что уменьшает число ложноположительных результатов, но может привести к увеличению числа ложноотрицательных:

$$\hat{y} = \begin{cases} 1, & \text{если } P(y = 1|x) \geq \tau, \\ 0, & \text{если } P(y = 1|x) < \tau. \end{cases} \quad (3.14)$$

Настройка порога τ позволяет регулировать строгость принимаемых моделью решений. При более высоких значениях τ модель становится более консервативной, что снижает число необоснованных аудитов, тогда как при меньших

значениях возрастает чувствительность модели и уменьшается вероятность пропуска нарушений.

Для учета неопределённости предсказаний вводится класс отклонений, соответствующий ситуациям, когда уверенность модели недостаточна для принятия окончательного решения. В этом случае задаются два порога: τ_{low} и τ_{high} , причём $\tau_{\text{low}} < \tau_{\text{high}}$. Если значение предсказания модели попадает в интервал $(\tau_{\text{low}}, \tau_{\text{high}})$, решение отклоняется, что интерпретируется как недостаточная уверенность модели в классификации.

Класс отклонений выражается следующим образом:

$$\hat{y} = \begin{cases} 1 & \text{если } P(y = 1|x) \geq \tau_{\text{high}} \\ 0 & \text{если } P(y = 1|x) < \tau_{\text{low}} \\ \text{Отклонить} & \text{если } \tau_{\text{low}} \leq P(y = 1|x) < \tau_{\text{high}} \end{cases} . \quad (3.15)$$

Этот подход позволяет модели воздерживаться от принятия решения, когда уровень уверенности недостаточно высок, особенно в случаях, когда стоимость неправильных предсказаний велика, например, в случае ресурсоемких налоговых проверок.

В задаче предсказания налогового аудита класс отклонений используется для обработки случаев с недостаточной уверенностью модели. Например, при установке порогов $\tau_{\text{low}} = 0.3$ и $\tau_{\text{high}} = 0.7$ предсказание интерпретируется следующим образом: если $P(y = 1|x) \geq 0.7$, принимается решение о необходимости аудита; если $P(y = 1|x) < 0.3$, аудит считается не требующимся. Если же вероятность попадает в интервал $(0.3, 0.7)$, модель воздерживается от принятия окончательного решения, и соответствующий случай относится к классу отклонений для дополнительного анализа.

Для прогнозирования необходимости налогового аудита юридических лиц были обучены несколько моделей машинного обучения. Целью являлось снижение числа необоснованных аудитов при сохранении способности выявлять объекты, требующие проверки. Набор данных является несбалансированным: большинство наблюдений относится к классу «аудит не требуется». Поэтому оценка моделей проводилась с использованием метрик точности, полноты и F1-меры. Значения метрик приведены на рисунке 3.8.

Логистическая регрессия обеспечивает высокую точность для основного класса, однако хуже выявляет объекты меньшинства. Случайный лес дает более

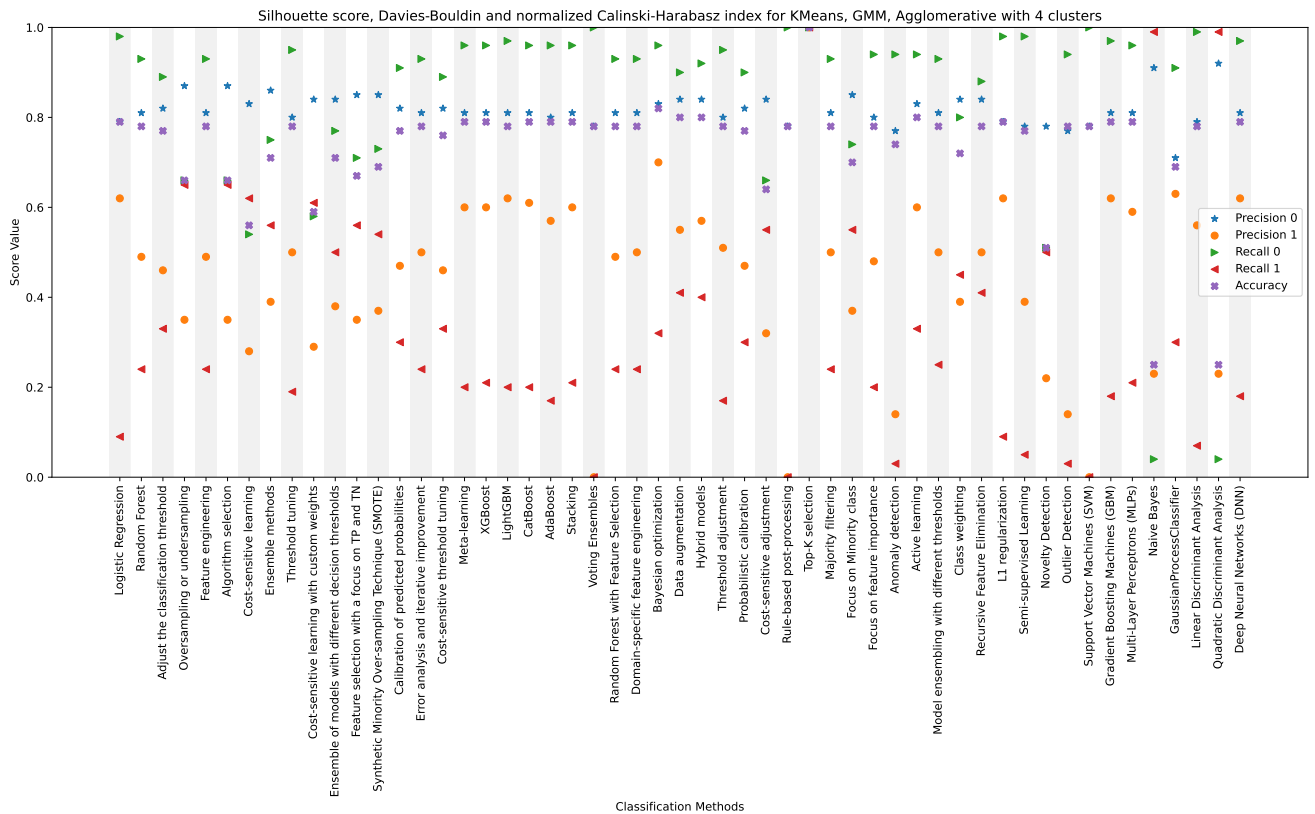


Рисунок 3.8 — Метрики качества классификации (0 – аудит не требуется, 1 – аудит требуется).

сбалансированные значения точности и полноты, но также сохраняет ограниченную способность выявления редких случаев. Изменение порога классификации и методы балансировки выборки увеличивают полноту для меньшинства ценой снижения общей точности.

Ансамблевые методы и алгоритмы градиентного бустинга (XGBoost, LightGBM, CatBoost) обеспечивают наилучший баланс между точностью и полнотой при работе с несбалансированными данными. Значения общей точности для этих моделей остаются на уровне около 0.79 при более устойчивом выявлении объектов, требующих аудита. Матрицы ошибок представлены на рисунке 3.9.

Сравнение матриц ошибок показывает, что модели бустинга (LightGBM, CatBoost, XGBoost) обеспечивают более благоприятное соотношение между выявлением необходимых аудитов и числом необоснованных проверок по сравнению с базовыми моделями.

Для модели XGBoost выполнена оптимизация гиперпараметров с использованием F1-меры в качестве целевой функции. Оптимизация проводилась по параметрам $n_estimators$, $learning_rate$, max_depth , $subsample$ и



Рисунок 3.9 — Матрицы ошибок для 25 классификаторов с точностью выше 0.5. Модели обучались с использованием разбиения данных на обучающую и тестовую выборки в пропорции 80/20.

colsample_bytree. Для анализа структуры пространства гиперпараметров использована визуализация t-SNE, снижающая размерность пространства до двух измерений (рисунок 3.10).

В пространстве гиперпараметров наблюдаются области конфигураций с близкими значениями F1-меры. По результатам анализа выбраны три конфигурации модели XGBoost с наилучшими значениями F1-меры, которые используются для дальнейших экспериментов.

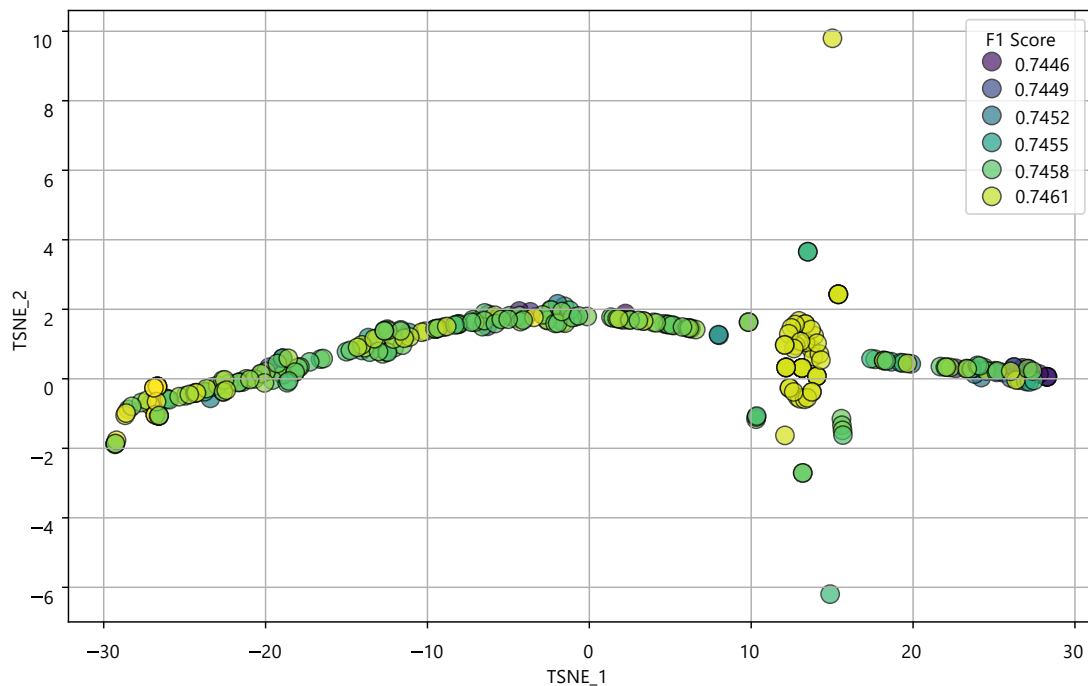


Рисунок 3.10 — Двумерная проекция пространства гиперпараметров XGBoost, полученная методом t-SNE.

На рисунке 3.11 приведено распределение наблюдений по основным бинарным признакам при использовании диапазона порога для класса отклонений 0.55–0.65. Рассматриваются признаки: сокрытие личного подоходного налога руководителем (PIT), участие в схемах уклонения от налогов (TES), индикатор высокого налогового риска и индикатор заблокированных счетов. Метки “(low)” соответствуют классу “аудит не требуется”, “(high)” — классу “аудит требуется”, а “(rej)” обозначает случаи, отнесённые к классу отклонений. Анализ выполнен на 10% подвыборке исходного набора данных.

Для признака несоответствия региона руководителя региону юридического лица модель Xgb1 относит большинство объектов к классу “аудит не требуется”. При этом 6 302 наблюдения классифицированы как отрицательный класс, 129 – отнесены к классу отклонений, а 505 – ошибочно классифицированы. Такая структура распределения указывает на ориентацию модели на снижение числа необоснованных аудитов.

Модель Xgb3, напротив, чаще относит наблюдения к классу “аудит требуется”. В частности, 422 объекта отнесены к положительному классу, при этом число наблюдений в классе отклонений составляет 101. Одновременно значи-

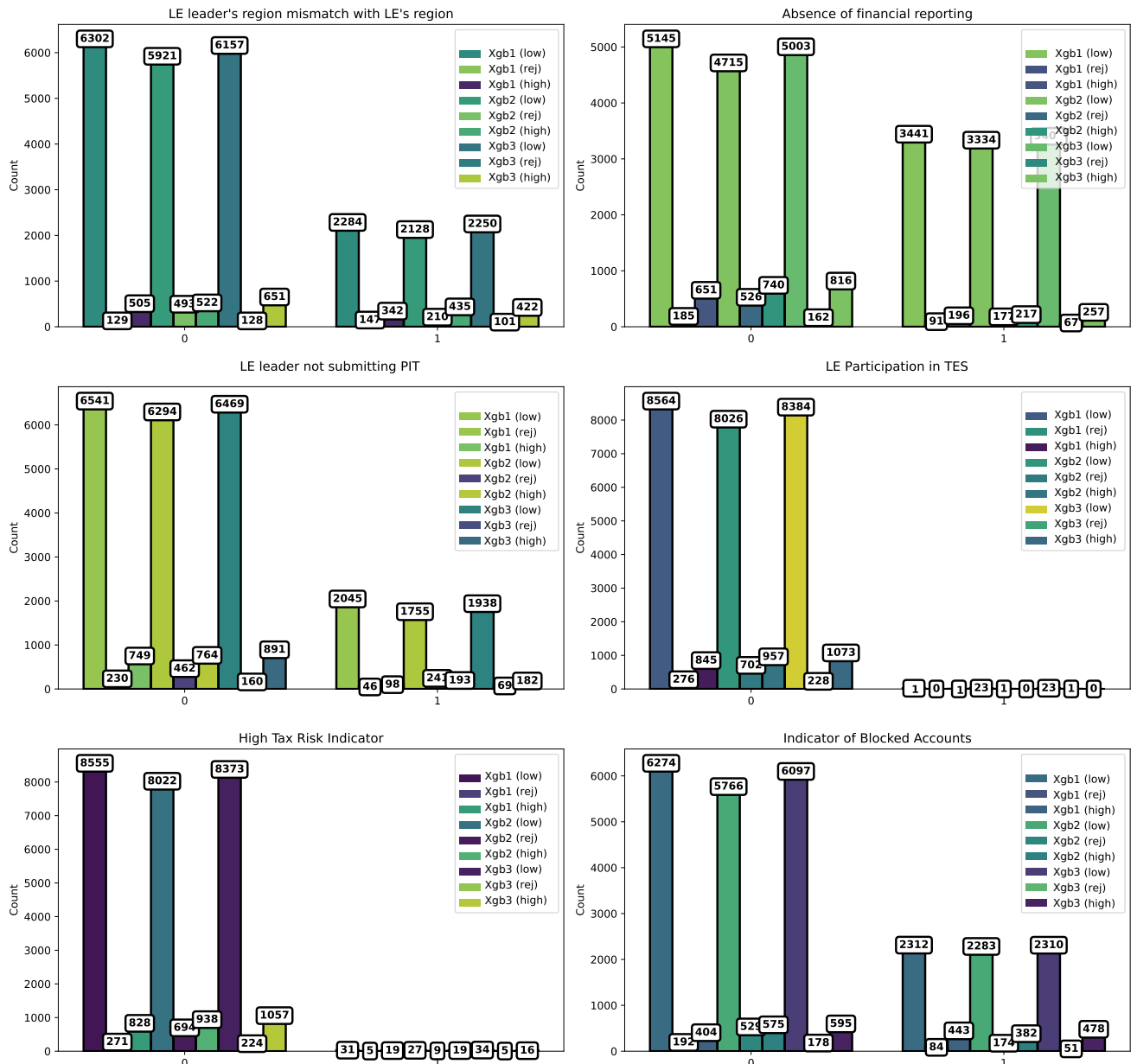


Рисунок 3.11 — Распределение наблюдений по бинарным признакам при диапазоне порога 0.55–0.65. Используются модели XGBoost (Xgb1, Xgb2, Xgb3). Метки (low) и (high) соответствуют решениям “аудит не требуется” и “аудит требуется”, а (rej) обозначает класс отклонений.

тельная часть наблюдений (2 250) классифицируется как отрицательный класс, что наблюдается и для других рассматриваемых моделей.

В целом модель Xgb1 характеризуется большей склонностью к отнесению наблюдений к отрицательному классу, тогда как модель Xgb3 чаще выделяет случаи, требующие налоговой проверки.

Анализ распределения бинарных признаков показывает, что для класса “аудит не требуется” преобладают благоприятные значения признаков: соответствие региона руководителю региону организации, наличие финансовой отчетности, подача налоговых деклараций, отсутствие признаков участия в схемах уклонения, низкий налоговый риск и отсутствие заблокированных счетов.

Для класса “аудит требуется” наблюдается противоположная структура признаков. Доля наблюдений, отнесённых к классу отклонений, меньше доли положительного класса, что указывает на ограниченное число случаев, в которых модель воздерживается от окончательного решения.

Оптимизация гиперпараметров XGBoost с использованием SelfCSHAGA+ДКОП.

Результаты оптимизации гиперпараметров модели XGBoost представлены на рис. 3.12. Диаграммы размаха отражают изменение значения F1-меры по поколениям SelfCSHAGA при различных вариантах кластеризации пространства гиперпараметров.

В начальных поколениях значение F1-меры находится в диапазоне 0.746–0.750. По мере работы SelfCSHAGA наблюдается постепенное улучшение метрики, и в последних поколениях она достигает значений порядка 0.756.

Для вариантов кластеризации с 1–4 кластерами получаются близкие значения F1-меры в финальных поколениях. При увеличении числа кластеров часть запусков завершается раньше, что связано с достижением сходимости SelfCSHAGA и отсутствием дальнейшего улучшения решения.

Полученные результаты показывают, что использование SelfCSHAGA+ДКОП позволяет эффективно исследовать пространство гиперпараметров модели XGBoost и находить конфигурации, обеспечивающие устойчивое улучшение качества классификации.

Оценка экономического эффекта использования SelfCSHAGA+ДКОП.

При оптимизации SelfCSHAGA (Рисунок 3.10) видно, что F1-метрика увеличивалась за 50 поколений с 0.7446 до 0.7461. В то же время, при оптимизации гиперпараметров XGBoost с использованием SelfCSHAGA+ДКОП и числом кластеров менее 5, рост начинается с 0.744, но уже за 20 поколений достигает 0.7560 (Рисунок 3.12). Увеличение метрики F1 на 0.0099 (0.01) привело к увеличению общей точности модели с 79% до 82%.

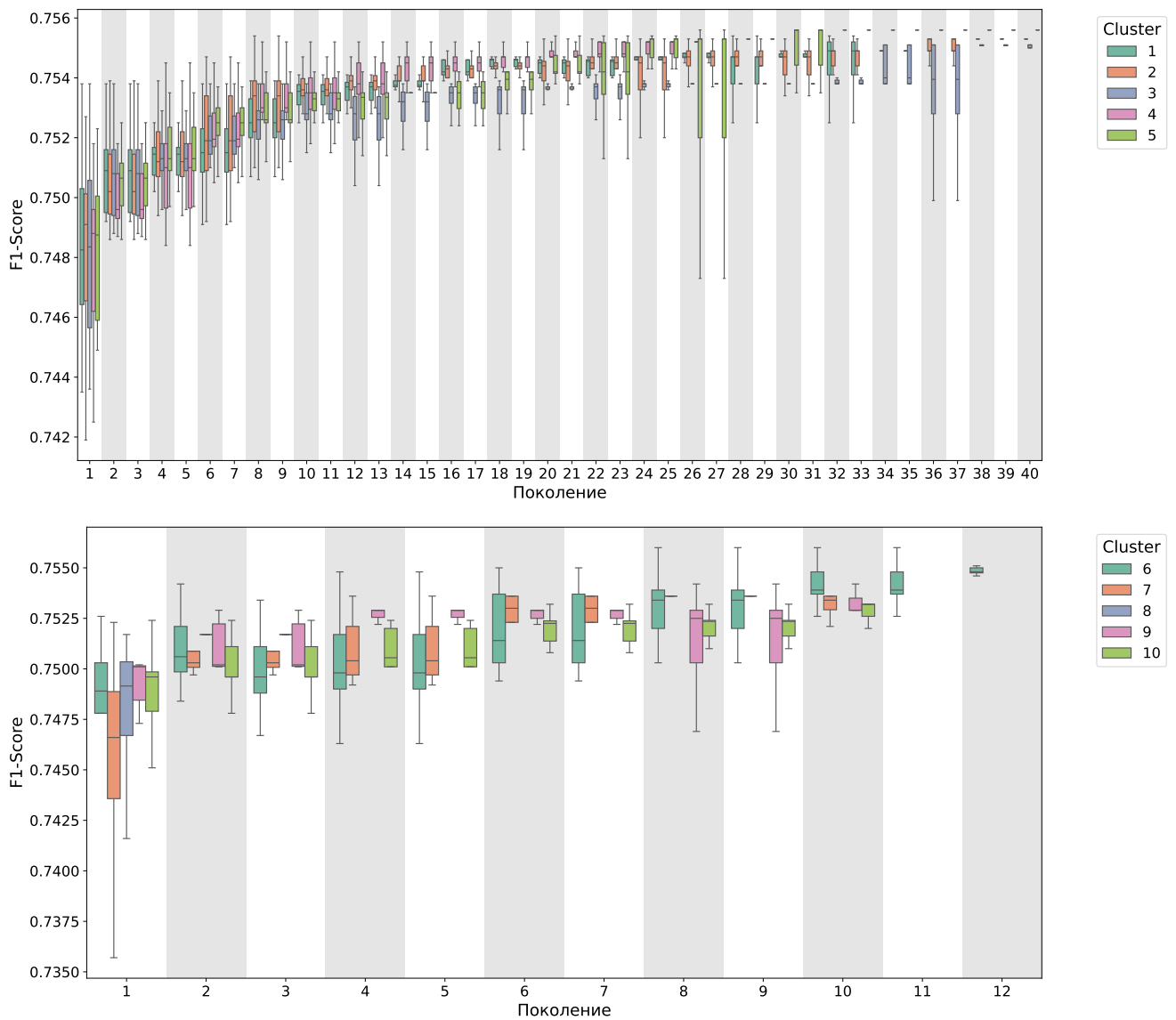


Рисунок 3.12 — Диаграммы размаха F1-меры по поколениям для кластеров 1–5 (верхний график) и 6–10 (нижний график).

Трудозатраты на ручную проверку зависят от сложности конкретного случая, состава документов, необходимости сверки с внешними источниками и внутренних процедур организации. Поэтому далее приводится не норматив рабочего времени, а сценарная оценка возможного масштаба эффекта. В базовом сценарии предполагается, что проверка одной записи занимает в среднем 30 минут. Тогда сокращение ручной проверки на 30 тыс. записей соответствует экономии около 15 тыс. человеко-часов. В более трудоемком сценарии, при средней длительности проверки 1 час на запись, потенциальная экономия может достигать 30 тыс. человеко-часов.

С учетом того, что в налоговой базе содержится порядка 1 миллиона записей о юридических лицах, то повышение точности модели на 3% потенциально

позволяет исключить до 30 000 записей из списка, требующих дополнительной проверки вручную. Допустим, на ручную верификацию одной компании уходит в среднем 30 минут, включая анализ документов, сверку данных и вынесение окончательного решения. В этом случае проверка 30 000 записей потребует 15000 человеко-часов. Однако на практике часть сложных случаев может требовать больше времени, например, до 1 часа на компанию. В таком сценарии общий объем трудозатрат возрастает до 30 000 человеко-часов.

Кроме того, повышение эффективности автоматизированных проверок может сократить число ложных срабатываний и снизить количество запросов на уточнение данных, что, в свою очередь, уменьшает нагрузку на бизнес и сокращает время рассмотрения заявок. Это ведет к ускорению процессов регистрации и налогового администрирования, снижению операционных издержек и потенциальному увеличению налоговых поступлений за счет более точной идентификации аномальных или рискованных компаний.

3.3 Выводы по главе 3

В главе рассмотрены практические задачи оптимизации моделей машинного обучения, а именно для системы многоклассовой классификации текстовых материалов госконтрактов и бинарной классификации несбалансированных данных юридических лиц. Показано, что качество модели определяется не только количеством обнаруженных аномалий, но и соотношением истинных и ложных срабатываний, напрямую влияющим на затраты и результативность проверочных мероприятий.

Обосновано использование генетического алгоритма с динамической коррекцией поискового пространства для настройки гиперпараметров моделей. Такой подход позволяет адаптивно перераспределять вычислительные ресурсы в пользу областей пространства гиперпараметров, обеспечивающих более благоприятный баланс между полнотой выявления аномалий и уровнем ложных срабатываний.

Предложен практический подход, в которой каждая конфигурация гиперпараметров интерпретируется как решение, а её качество оценивается с учётом качества предсказаний.

Показано, что даже умеренное увеличение точности модели и доли корректно выявленных случаев приводит к значимому росту экономической отдачи. При этом снижение уровня ложных срабатываний способствует более рациональному распределению ресурсов и уменьшению негативного воздействия на добросовестные организации.

Тем самым доказано, что SelfCSHAGA с динамической коррекцией поискового пространства является эффективным инструментом настройки гиперпараметров в задачах выявления аномалий. Его применение обеспечивает более рациональное использование вычислительных ресурсов и повышает применимость моделей машинного обучения.

Заключение

В рамках диссертационного исследования получены новые научные результаты, которые повышают точность решения сложных задач оптимизации со смешанными переменными бинарными генетическими алгоритмами за счет динамической коррекции пространства поиска. Цель диссертационного исследования достигнута путем решения поставленных задач, а именно:

1. Проведен анализ современного состояния методов эволюционной оптимизации, который показал, что несмотря на широкое применение бинарных ЭА в задачах глобальной оптимизации со смешанными переменными, их эффективность ограничивается фиксацией области поиска, что приводит к повышенному количеству вычислений целевой функции из-за нерационального распределения решений, а также к снижению надежности алгоритма и точности получаемых решений.

2. Разработан новый метод динамической коррекции области поиска, который позволяет перераспределять вычислительные ресурсы алгоритма в пользу перспективных областей пространства решений.

3. Разработан алгоритм корректировки границ кластеров, основанный на вычислении относительной пригодности индивидов, который позволяет адаптивно расширять или сужать область поиска по каждой координате в зависимости от качества решений на границах, исключая менее перспективные области поиска.

4. Разработана процедура динамической модификации схемы бинарного кодирования генотипов, которая позволяет за счет объединения разрозненных областей поиска в едином бинарном представлении гарантировать корректное применение стандартных генетических операторов и обеспечить повышение точности получаемых решений без увеличения размерности бинарного вектора.

5. Проведены экспериментальные исследования на наборах тестовых функций, которые включают стандартные задачи, задачи чемпионата мира по эволюционной оптимизации IEEE CEC 2017 и прикладные тестовые задачи с соревнований по оптимизации IEEE CEC. Результаты экспериментов показали, что применение предложенного подхода повышает эффективность базового алгоритма, который использует стандартное статическое бинарное кодирование.

Поскольку метод динамической коррекции не меняет структуру алгоритма, повышение эффективности достигается на всех исследованных алгоритмах, включая как стандартный ГА, так и ведущие на текущий момент подходы: бинарный алгоритм роя частиц с полной информацией, бинарный самонастраивающийся алгоритм дифференциальной эволюции на основе истории успеха L-SRTDE и самонастраивающийся генетический алгоритм SelfCSHAGA.

6. Проведена апробация на практических задачах структурно-параметрического синтеза моделей машинного обучения для классификации госконтрактов и выявления недостоверных сведений о юридических лицах, в которых получены новые практические результаты, превосходящие показатели стандартных алгоритмов.

Таким образом, в диссертационном исследовании поставлена и решена актуальная научная задача повышения эффективности решения сложных задач глобальной оптимизации алгоритмически заданных целевых функций со смешанными переменными, что имеет существенное значение для теории и практики системного анализа, управления и обработки информации.

Перспективы дальнейшего развития связаны с расширением метода на многокритериальные задачи оптимизации, а также его интеграцией в автоматизированные системы настройки моделей машинного обучения. Практически целесообразно применение предложенного подхода в задачах с высокой стоимостью вычисления целевой функции и сложной структурой пространства поиска. Метод может быть реализован как надстройка над существующими эволюционными алгоритмами, что обеспечивает простоту его внедрения в прикладные системы.

Список литературы

1. Global optimization of mixed-integer nonlinear programs with SCIP 8 / К. Bestuzheva [и др.] // Journal of Global Optimization. — 2023. — с. 1—24.
2. Mixed-integer and conditional trajectory planning for an autonomous mining truck in loading/dumping scenarios: A global optimization approach / В. Li [и др.] // IEEE Transactions on Intelligent Vehicles. — 2022. — т. 8, № 2. — с. 1512—1522.
3. *Crespo Márquez A.* The curse of dimensionality // Digital Maintenance Management: Guiding Digital Transformation in Maintenance. — Springer, 2022. — с. 67—86.
4. *Wu Y., Wang L.* A survey of tuning parameter selection for high-dimensional regression // Annual review of statistics and its application. — 2020. — т. 7, № 1. — с. 209—226.
5. Multimodal multi-objective evolutionary algorithm for multiple path planning / Х. Yao [и др.] // Computers & Industrial Engineering. — 2022. — т. 169. — с. 108145.
6. *Vlatakis-Gkaragkounis E.-V., Flokas L., Piliouras G.* Solving min-max optimization with hidden structure via gradient descent ascent // Advances in Neural Information Processing Systems. — 2021. — т. 34. — с. 2373—2386.
7. *Yoshida Y., Okada M.* Data-dependence of plateau phenomenon in learning with neural network—Statistical mechanical analysis // Advances in Neural Information Processing Systems. — 2019. — т. 32.
8. Evaluating the robustness of neural networks: An extreme value theory approach / Т.-W. Weng [и др.] // arXiv preprint arXiv:1801.10578. — 2018.
9. Evolutionary optimization of constrained problems / К. Watanabe [и др.] // Evolutionary Computations: New Algorithms and their Applications to Evolutionary Robots. — 2004. — с. 53—64.
10. *Mozannar H., Ohannessian M., Srebro N.* Fair learning with private demographic data // International Conference on Machine Learning. — PMLR. 2020. — с. 7066—7075.

11. *Hasan M. S., Chowdhury M. M.-U.-T., Kamalasan S.* Sequential quadratic programming (SQP) based optimal power flow methodologies for electric distribution system with high penetration of DERs // IEEE Transactions on Industry Applications. — 2024.
12. *Alev V. L., Jeronimo F. G., Tulsiani M.* Approximating constraint satisfaction problems on high-dimensional expanders // 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). — IEEE. 2019. — с. 180—201.
13. An effective algorithm for hyperparameter optimization of neural networks / G. I. Diaz [и др.] // IBM Journal of Research and Development. — 2017. — т. 61, № 4/5. — с. 9—1.
14. *Floudas C. A., Pardalos P. M.* State of the art in global optimization: computational methods and applications. — 2013.
15. High-dimensional asymptotics of feature learning: How one gradient step improves the representation / J. Ва [и др.] // Advances in Neural Information Processing Systems. — 2022. — т. 35. — с. 37932—37946.
16. Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network / D. Guo [и др.] // IEEE transactions on systems, man, and cybernetics: systems. — 2021. — т. 52, № 4. — с. 2084—2097.
17. *Johnstone I. M., Titterington D. M.* Statistical challenges of high-dimensional data // Philosophical transactions of the Royal Society A: Mathematical, physical and engineering sciences. — 2009. — т. 367, № 1906. — с. 4237—4253.
18. *Квасов Д. Е., Сергеев Я. Д.* Методы липшицевой глобальной оптимизации в задачах управления // Автоматика и телемеханика. — 2013. — № 9. — с. 3—19.
19. *Vikhar P. A.* Evolutionary algorithms: A critical review and its future prospects // 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC). — IEEE. 2016. — с. 261—265.
20. *Mirjalili S., Mirjalili S.* Genetic algorithm // Evolutionary algorithms and neural networks: Theory and applications. — 2019. — с. 43—55.

21. *Price K. V.* Differential evolution // Handbook of optimization: From classical to modern approach. — Springer, 2013. — с. 187—214.
22. *Dorigo M., Birattari M., Stutzle T.* Ant colony optimization // IEEE computational intelligence magazine. — 2007. — т. 1, № 4. — с. 28—39.
23. *Hansen N.* The CMA evolution strategy: a comparing review // Towards a new evolutionary computation: Advances in the estimation of distribution algorithms. — 2006. — с. 75—102.
24. *Espejo P. G., Ventura S., Herrera F.* A survey on the application of genetic programming to classification // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). — 2009. — т. 40, № 2. — с. 121—144.
25. A binary waterwheel plant optimization algorithm for feature selection / A. A. Alhussan [и др.] // IEEE Access. — 2023. — т. 11. — с. 94227—94251.
26. *Folly K.* An Improved Population-Based Incremental Learning Algorithm // International Journal of Swarm Intelligence Research. — 2013. — DOI: [10.4018/JSIR.2013010102](https://doi.org/10.4018/JSIR.2013010102).
27. *Iwamatsu M.* Generalized evolutionary programming with Lévy-type mutation // Computer Physics Communications. — 2002. — т. 147, № 1/2. — с. 729—732. — DOI: [10.1016/S0010-4655\(02\)00386-7](https://doi.org/10.1016/S0010-4655(02)00386-7).
28. *Nikolikj A., Munoz M. A., Eftimov T.* Benchmarking footprints of continuous black-box optimization algorithms: Explainable insights into algorithm success and failure // Swarm and Evolutionary Computation. — 2025. — т. 94. — с. 101895.
29. Differential evolution: A recent review based on state-of-the-art works / M. F. Ahmad [и др.] // Alexandria Engineering Journal. — 2022. — т. 61, № 5. — с. 3831—3872.
30. Two-stage double niched evolution strategy for multimodal multiobjective optimization / K. Zhang [и др.] // IEEE Transactions on Evolutionary Computation. — 2021. — т. 25, № 4. — с. 754—768.
31. Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection / J. Hu [и др.] // Knowledge-Based Systems. — 2021. — т. 213. — с. 106684.

32. Identification of clinically relevant T cell receptors for personalized T cell therapy using combinatorial algorithms / R. Pétremand [и др.] // *Nature biotechnology*. — 2025. — т. 43, № 3. — с. 323—328.
33. A comprehensive survey on the generalized traveling salesman problem / P. C. Pop [и др.] // *European Journal of Operational Research*. — 2024. — т. 314, № 3. — с. 819—835.
34. Чуб И. А., Новожилова М. В. Модификация точного метода решения задачи размещения прямоугольных объектов // *Автоматизированные системы управления и приборы автоматики*. — 2008. — № 145. — с. 57—63.
35. Микулук И. И., Благовещенская Е. А. Распараллеливание гибридного алгоритма муравьиной колонии с изменяющимися с помощью генетического алгоритма параметрами // *Проблемы информатики*. — 2023. — 2 (59). — с. 86—97.
36. Binary or Integer Chromosome: Which Is the Best Structure for Supervised Machine Learning Using Genetic Algorithms? / A. H. d. S. Alves [и др.] // *Applied Sciences*. — 2025. — т. 15, № 5. — с. 2608.
37. Golub M. An implementation of binary and floating point chromosome representation in genetic algorithm // *Proceedings of the 18th International Conference on Information Technology Interfaces*. — Zagreb: Sveučilišni računski centar Sveučilišta u Zagrebu (Srce). 1996. — с. 417—422.
38. Achiche S., Baron L., Balazinski M. Real/binary-like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: a comparative study // *Engineering Applications of Artificial Intelligence*. — 2004. — т. 17, № 4. — с. 313—325.
39. Binarized neural networks / I. Hubara [и др.] // *Advances in neural information processing systems*. — 2016. — т. 29.
40. Chu W., Gao X., Sorooshian S. A new evolutionary search strategy for global optimization of high-dimensional problems // *Information Sciences*. — 2011. — т. 181, № 22. — с. 4909—4927.
41. Holland J. H. Genetic algorithms // *Scientific american*. — 1992. — т. 267, № 1. — с. 66—73.

42. *Hamdia K. M., Zhuang X., Rabczuk T.* An efficient optimization approach for designing machine learning models based on genetic algorithm // *Neural Computing and Applications.* — 2021. — т. 33, № 6. — с. 1923—1933.
43. *Maaranen H., Miettinen K., Penttinen A.* On initial populations of a genetic algorithm for continuous optimization problems // *Journal of Global Optimization.* — 2007. — т. 37, № 3. — с. 405—436.
44. *Hu X.-B., Di Paolo E.* Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling // *IEEE Transactions on Intelligent Transportation Systems.* — 2008. — т. 9, № 2. — с. 301—310.
45. *Chen T., Guestrin C.* Xgboost: A scalable tree boosting system // *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.* — 2016. — с. 785—794.
46. Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification / Y. Xue [и др.] // *Knowledge-Based Systems.* — 2021. — т. 227. — с. 107218.
47. Screening of COVID-19 suspected subjects using multi-crossover genetic algorithm based dense convolutional neural network / D. Singh [и др.] // *IEEE Access.* — 2021. — т. 9. — с. 142566—142580.
48. *Ileberi E., Sun Y., Wang Z.* A machine learning based credit card fraud detection using the GA algorithm for feature selection // *Journal of Big Data.* — 2022. — т. 9, № 1. — с. 24.
49. NEAT: Nonlinearity aware training for accurate, energy-efficient, and robust implementation of neural networks on 1T-1R crossbars / A. Bhattacharjee [и др.] // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* — 2021. — т. 41, № 8. — с. 2625—2637.
50. *Deb K.* An efficient constraint handling method for genetic algorithms // *Computer methods in applied mechanics and engineering.* — 2000. — т. 186, № 2—4. — с. 311—338.
51. A review of parallel heterogeneous computing algorithms in power systems / D. Rodriguez [и др.] // *Algorithms.* — 2021. — т. 14, № 10. — с. 275.

52. *Jin H., Song Q., Hu X.* Auto-keras: An efficient neural architecture search system // Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. — 2019. — с. 1946—1956.
53. Systematic review on neural architecture search / S. Salmani Pour Avval [и др.] // Artificial Intelligence Review. — 2025. — т. 58, № 3. — с. 73.
54. Autokeras approach: A robust automated deep learning network for diagnosis disease cases in medical images / A. Alaiad [и др.] // Journal of Imaging. — 2023. — т. 9, № 3. — с. 64.
55. *Holland J. H.* Genetic algorithms and adaptation // Adaptive control of ill-defined systems. — Springer, 1984. — с. 317—333.
56. *Goldberg D. E., Korb B., Deb K.* Messy genetic algorithms: Motivation, analysis, and first results // Complex systems. — 1989. — т. 3, № 5.
57. *Deb K.* Introduction to evolutionary multiobjective optimization // Multiobjective optimization: Interactive and evolutionary approaches. — Springer, 2008. — с. 59—96.
58. Particle swarm optimization / R. Eberhart, J. Kennedy [и др.] // Proceedings of the IEEE international conference on neural networks. т. 4. — Perth, Australia. 1995. — с. 1942—1948.
59. *Kennedy J., Eberhart R. C.* A discrete binary version of the particle swarm algorithm // 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation. т. 5. — ieee. 1997. — с. 4104—4108.
60. *Gong T., Tuson A. L.* Differential evolution for binary encoding // Soft computing in industrial applications: Recent trends. — Springer, 2007. — с. 251—262.
61. *Whitley D.* A genetic algorithm tutorial // Statistics and computing. — 1994. — т. 4, № 2. — с. 65—85.
62. *Schraudolph N. N., Belew R. K.* Dynamic parameter encoding for genetic algorithms // Machine learning. — 1992. — т. 9, № 1. — с. 9—21.
63. *Kureichik V., Kureichik V.* A genetic algorithm for finding a salesman's route // Journal of Computer and Systems Sciences International. — 2006. — т. 45, № 1. — с. 89—95.

64. *Гладков Л., Курейчик В., Курейчик В.* Генетические алгоритмы. 2-е изд., исправл. и доп. // М.: Физматлит. — 2010. — с. 368.
65. *Карпенко А. П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. — 2016.
66. *Eremeev A.* On Complexity of Optimized Crossover for Binary Representations // Dagstuhl Seminar Proceedings. т. 6061. — 2006.
67. *Semenkin E., Semenkina M.* Self-configuring genetic algorithm with modified uniform crossover operator // International Conference in Swarm Intelligence. — Springer. 2012. — с. 414—421.
68. *Сонов Е. А.* Бинарный генетический алгоритм с декомпозицией задачи на основе EDA для глобальной оптимизации большой размерности // Siberian Aerospace Journal. — 2016. — т. 17, № 4. — с. 899—906.
69. *Sidorov M., Semenkin E., Minker W.* Unconstrained global optimization: A benchmark comparison of population-based algorithms // . т. 1. — 2015. — с. 230—237. — DOI: [10.5220/0005548002300237](https://doi.org/10.5220/0005548002300237).
70. *Sidorov M., Semenkin E., Minker W.* Multi-agent cooperative algorithms of global optimization // ICINCO 2014 - Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics. т. 1. — 2014. — с. 259—265. — DOI: [10.5220/0005049402590265](https://doi.org/10.5220/0005049402590265).
71. *Brester C., Semenkin E.* Cooperative multi-objective genetic algorithm with parallel implementation // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2015. — т. 9140. — с. 471—478. — DOI: [10.1007/978-3-319-20466-6_49](https://doi.org/10.1007/978-3-319-20466-6_49).
72. *Malashin I.* Generation of Natural-Language Explanations for Static-Analysis Warnings Using Single-and Multi-Objective Optimization // Computers. — 2025. — т. 14, № 12. — с. 534.
73. *Karpenko A., Agasiev T., Sakharov M.* Intellectualization Methods of Population Algorithms of Global Optimization // Studies in Systems, Decision and Control. — 2020. — т. 259. — с. 137—151. — DOI: [10.1007/978-3-030-32579-4_11](https://doi.org/10.1007/978-3-030-32579-4_11).

74. *Sakharov M., Karpenko A.* A new way of decomposing search domain in a global optimization problem // *Advances in Intelligent Systems and Computing.* — 2018. — т. 679. — с. 398—407. — DOI: [10.1007/978-3-319-68321-8_41](https://doi.org/10.1007/978-3-319-68321-8_41).
75. *Demidova L. A., Zhuravlev V. E.* Novel Four-stage Comprehensive Analysis Approach for Population-based Optimization Algorithms //. — 2023. — с. 263—268. — DOI: [10.1109/SUMMA60232.2023.10349534](https://doi.org/10.1109/SUMMA60232.2023.10349534).
76. *Demidova L., Klyueva I., Pylkin A.* The study of characteristics of the hybrid particle swarm algorithm in solution of the global optimization problem //. — 2016. — с. 322—325. — DOI: [10.1109/MECO.2016.7525772](https://doi.org/10.1109/MECO.2016.7525772).
77. *Sopov E. A., Sopov S. A.* The convergence prediction method for genetic and PBIL-like algorithms with binary representation // *2011 International Siberian Conference on Control and Communications, SIBCON 2011 - Proceedings.* т. 201. — 2011. — с. 203—206. — DOI: [10.1109/SIBCON.2011.6072632](https://doi.org/10.1109/SIBCON.2011.6072632).
78. *Sopov E.* Hybrid binary GA-EDA algorithms for complex "black-box" optimization problems // *IOP Conference Series: Materials Science and Engineering.* т. 173. — 2017. — DOI: [10.1088/1757-899X/173/1/012019](https://doi.org/10.1088/1757-899X/173/1/012019).
79. *Sopov E.* Genetic Programming Hyper-heuristic for the Automated Synthesis of Selection Operators in Genetic Algorithms // *International Joint Conference on Computational Intelligence.* т. 1. — 2017. — с. 231—238. — DOI: [10.5220/0006497002310238](https://doi.org/10.5220/0006497002310238).
80. *Sopov E.* EDA-based decomposition approach for binary LSGO problems // *IJCCI 2016 - Proceedings of the 8th International Joint Conference on Computational Intelligence.* т. 1. — 2016. — с. 132—139. — DOI: [10.5220/0006034301320139](https://doi.org/10.5220/0006034301320139).
81. *Sopov E.* Large-scale global optimization using a binary genetic algorithm with EDA-based decomposition // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* — 2016. — т. 9712 LNCS. — с. 619—626. — DOI: [10.1007/978-3-319-41000-5_62](https://doi.org/10.1007/978-3-319-41000-5_62).

82. *Semenkin E., Semenkina M.* Self-configuring genetic algorithm with modified uniform crossover operator // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2012. — т. 7331 LNCS, PART 1. — с. 414–421. — DOI: [10.1007/978-3-642-30976-2_50](https://doi.org/10.1007/978-3-642-30976-2_50).
83. *Semenkin E., Semenkina M.* Stochastic models and optimization algorithms for decision support in spacecraft control systems preliminary design // Lecture Notes in Electrical Engineering. — 2014. — т. 283. — с. 51–65. — DOI: [10.1007/978-3-319-03500-0_4](https://doi.org/10.1007/978-3-319-03500-0_4).
84. *Stanovov V., Akhmedova S., Semenkin E.* Genetic Algorithm with Success History based Parameter Adaptation // International Joint Conference on Computational Intelligence. т. 1. — 2019. — с. 180–187. — DOI: [10.5220/0008071201800187](https://doi.org/10.5220/0008071201800187).
85. *Stanovov V., Akhmedova S., Semenkin E.* Visualizing Parameter Adaptation in Differential Evolution with Expected Fitness Improvement // 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020. — 2020. — с. 321–328. — DOI: [10.1109/SSCI47803.2020.9308467](https://doi.org/10.1109/SSCI47803.2020.9308467).
86. *Stanovov V., Semenkin E.* Genetic Programming for Automatic Design of Parameter Adaptation in Dual-Population Differential Evolution // GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. — 2023. — с. 1736–1743. — DOI: [10.1145/3583133.3596310](https://doi.org/10.1145/3583133.3596310).
87. *Stanovov V.* Designing Black-Box Optimizers with PushGP // GECCO 2024 Companion - Proceedings of the 2024 Genetic and Evolutionary Computation Conference Companion. — 2024. — с. 535–538. — DOI: [10.1145/3638530.3654172](https://doi.org/10.1145/3638530.3654172).
88. *Stanovov V., Semenkin E.* Genetic Improvement of Dynamic Optimization Algorithms using PushGP // GECCO 2025 Companion - Proceedings of the 2025 Genetic and Evolutionary Computation Conference Companion. — 2025. — с. 1976–1983. — DOI: [10.1145/3712255.3734281](https://doi.org/10.1145/3712255.3734281).
89. *Sopov E.* A self-configuring multi-strategy multimodal genetic algorithm // Advances in Intelligent Systems and Computing. — 2016. — т. 419. — с. 15–26. — DOI: [10.1007/978-3-319-27400-3_2](https://doi.org/10.1007/978-3-319-27400-3_2).

90. *Sopov E.* Self-configuring ensemble of multimodal genetic algorithms // Studies in Computational Intelligence. — 2017. — т. 669. — с. 56—74. — DOI: [10.1007/978-3-319-48506-5_4](https://doi.org/10.1007/978-3-319-48506-5_4).
91. Dimension reduction with coevolutionary genetic algorithm for text classification / Т. Gasanova [и др.] // ICINCO 2014 - Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics. т. 1. — 2014. — с. 215—222. — DOI: [10.5220/0005020702150222](https://doi.org/10.5220/0005020702150222).
92. *Dolgui A., Ereemeev A., Sigaev V.* On Local Optima Distribution in Buffer Allocation Problem for Production Line with Unreliable Machines // IFAC-PapersOnLine. т. 55. — 2022. — с. 1092—1097. — DOI: [10.1016/j.ifacol.2022.09.535](https://doi.org/10.1016/j.ifacol.2022.09.535).
93. *Dolgui A., Ereemeev A., Sigaev V.* Fitness Landscapes of Buffer Allocation Problem in Production Lines and Genetic Algorithms Performance // GECCO 2025 Companion - Proceedings of the 2025 Genetic and Evolutionary Computation Conference Companion. — 2025. — с. 27—28. — DOI: [10.1145/3712255.3734243](https://doi.org/10.1145/3712255.3734243).
94. Level-based analysis of genetic algorithms and other search processes / D. Corus [и др.] // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2014. — т. 8672. — с. 912—921. — DOI: [10.1007/978-3-319-10762-2_90](https://doi.org/10.1007/978-3-319-10762-2_90).
95. Level-Based Analysis of Genetic Algorithms and Other Search Processes / D. Corus [и др.] // IEEE Transactions on Evolutionary Computation. — 2018. — т. 22, № 5. — с. 707—719. — DOI: [10.1109/TEVC.2017.2753538](https://doi.org/10.1109/TEVC.2017.2753538).
96. *Ereemeev A. V.* Runtime analysis of genetic algorithms with very high selection pressure // CEUR Workshop Proceedings. т. 1623. — 2016. — с. 428—439.
97. *Ereemeev A. V.* Hitting times of local and global optima in genetic algorithms with very high selection pressure // Yugoslav Journal of Operations Research. — 2017. — т. 27, № 3. — с. 323—339. — DOI: [10.2298/yjor160318016e](https://doi.org/10.2298/yjor160318016e).

98. *Eremeev A., Topchii V.* Generalization of the Heavy-Tailed Mutation in the (1+(,)) Genetic Algorithm // GECCO 2024 Companion - Proceedings of the 2024 Genetic and Evolutionary Computation Conference Companion. — 2024. — с. 93–94. — DOI: [10.1145/3638530.3664095](https://doi.org/10.1145/3638530.3664095).
99. *Alhijawi B., Awajan A.* Genetic algorithms: Theory, genetic operators, solutions, and applications // Evolutionary Intelligence. — 2024. — т. 17, № 3. — с. 1245–1256.
100. *Sadeghi Hesar A., Houshmand M.* A memetic quantum-inspired genetic algorithm based on tabu search // Evolutionary Intelligence. — 2024. — т. 17, № 3. — с. 1837–1853.
101. *Glover F., Kelly J. P., Laguna M.* Genetic algorithms and tabu search: hybrids for optimization // Computers & Operations Research. — 1995. — т. 22, № 1. — с. 111–134.
102. *D'Angelo G., Palmieri F.* GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems // Information Sciences. — 2021. — т. 547. — с. 136–162.
103. Design optimization of elastic metamaterials with multilayered honeycomb structure by Kriging surrogate model and genetic algorithm / L. Cao [и др.] // Structural and Multidisciplinary Optimization. — 2024. — т. 67, № 5. — с. 82.
104. *Li J., Gonsalves T.* Parallel hybrid island metaheuristic algorithm // IEEE Access. — 2022. — т. 10. — с. 42268–42286.
105. A niching framework based on fitness proportionate sharing for multi-objective genetic algorithm (moga-fps) / A.-R. Nuhu [и др.] // Proceedings of the Genetic and Evolutionary Computation Conference Companion. — 2021. — с. 191–192.
106. Best fit DNA-based cryptographic keys: the genetic algorithm approach / P. Mukherjee [и др.] // Sensors. — 2022. — т. 22, № 19. — с. 7332.
107. Multi-population genetic algorithm with crowding-based local search for fuzzy multi-objective supply chain configuration / X. Zhang [и др.] // Swarm and Evolutionary Computation. — 2024. — т. 91. — с. 101698.

108. *Wang C., Ma B., Sun J.* A co-evolutionary genetic algorithm with knowledge transfer for multi-objective capacitated vehicle routing problems // *Applied Soft Computing*. — 2023. — т. 148. — с. 110913.
109. *Xue Y., Cai X., Neri F.* A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification // *Applied Soft Computing*. — 2022. — т. 127. — с. 109420.
110. *Adekoya O., Aneiba A.* An adapted nondominated sorting genetic algorithm iii (nsga-iii) with repair-based operator for solving controller placement problem in software-defined wide area networks // *IEEE Open Journal of the Communications Society*. — 2022. — т. 3. — с. 888—901.
111. A novel multi-population evolutionary algorithm based on hybrid collaboration for constrained multi-objective optimization / Q. Wang [и др.] // *Swarm and Evolutionary Computation*. — 2024. — т. 87. — с. 101581.
112. Quantum-inspired genetic algorithm for designing planar multilayer photonic structure / Z. Xu [и др.] // *NPJ Computational Materials*. — 2024. — т. 10, № 1. — с. 257.
113. *Kılıç M. B., Şahin Y., Koca M. B.* Genetic algorithm approach with an adaptive search space based on EM algorithm in two-component mixture Weibull parameter estimation // *Computational Statistics*. — 2021. — т. 36, № 2. — с. 1219—1242.
114. *Шерстнев П. А., Семенкин Е. С.* SelfCSHAGA: самоконфигурируемый генетический алгоритм оптимизации с адаптацией на основе истории успеха // *Вестник Московского государственного технического университета им. НЭ Баумана. Серия «Приборостроение»*. — 2025. — 2 (151). — с. 8.
115. *Малашин И. П., Сопов Е. А.* Анализ эффективности самоконфигурируемого бинарного генетического алгоритма с модифицированным методом динамической коррекции области поиска // *Моделирование, оптимизация и информационные технологии*. — 2026. — т. 14, № 2. — DOI: [10.26102/2310-6018/2026.53.2.001](https://doi.org/10.26102/2310-6018/2026.53.2.001).

116. *Semenkina M. E., Semenkin E. S., Ryzhikov I. S.* Dynamics forecasting for electrical characteristics of spacecraft solar arrays by means of computational intelligence // *Siberian Aerospace Journal*. — 2014. — т. 15, № 3. — с. 139—145.
117. *Rasheed K., Hirsh H., Gelsey A.* A genetic algorithm for continuous design space search // *Artificial Intelligence in Engineering*. — 1997. — т. 11, № 3. — с. 295—305.
118. *Chan C.-H., Liu G.* Hysteresis identification and compensation using a genetic algorithm with adaptive search space // *Mechatronics*. — 2007. — т. 17, № 7. — с. 391—402.
119. Differential developmental expression of the two rat brain glutamate transporter proteins GLAST and GLT / K. Ullensvang [и др.] // *European Journal of Neuroscience*. — 1997. — т. 9, № 8. — с. 1646—1655.
120. *Mahdavi S., Rahnamayan S., Deb K.* Opposition based learning: A literature review // *Swarm and evolutionary computation*. — 2018. — т. 39. — с. 1—23.
121. *Choi T. J., Pachauri N.* Adaptive search space for stochastic opposition-based learning in differential evolution // *Knowledge-Based Systems*. — 2024. — т. 300. — с. 112172.
122. *Sun G., Li C., Deng L.* An adaptive regeneration framework based on search space adjustment for differential evolution // *Neural Computing and Applications*. — 2021. — т. 33. — с. 9503—9519.
123. *Wu G., Mallipeddi R., Suganthan P. N.* Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization // National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report. — 2017. — т. 9. — с. 2017.
124. A clustering based mate selection for evolutionary optimization / J. Zhang [и др.] // *Big Data & Information Analytics*. — 2017. — т. 2, № 1. — с. 77—85.
125. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization / P. N. Suganthan [и др.] // *KanGAL report*. — 2005. — т. 2005005, № 2005. — с. 2005.

126. *Neri F., Tirronen V.* Recent advances in differential evolution: a survey and experimental analysis // Artificial intelligence review. — 2010. — т. 33, № 1. — с. 61—106.
127. *Zhang J., Sanderson A. C.* JADE: adaptive differential evolution with optional external archive // IEEE Transactions on evolutionary computation. — 2009. — т. 13, № 5. — с. 945—958.
128. *Tanabe R., Fukunaga A.* Evaluating the performance of SHADE on CEC 2013 benchmark problems // 2013 IEEE Congress on evolutionary computation. — IEEE. 2013. — с. 1952—1959.
129. *Piotrowski A. P.* L-SHADE optimization algorithms with population-wide inertia // Information Sciences. — 2018. — т. 468. — с. 117—141.
130. *Brest J., Maučec M. S., Bošković B.* iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization // 2016 IEEE congress on evolutionary computation (CEC). — IEEE. 2016. — с. 1188—1195.
131. L-SHADE-E: Ensemble of two differential evolution algorithms originating from L-SHADE / X. Wang [и др.] // Information Sciences. — 2021. — т. 552. — с. 201—219.
132. *Stanovov V., Akhmedova S., Semenkin E.* NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization // 2021 IEEE Congress on Evolutionary Computation (CEC). — IEEE. 2021. — с. 809—816.
133. *Stanovov V., Akhmedova S., Semenkin E.* NL-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 Numerical Optimization // 2022 IEEE Congress on Evolutionary Computation (CEC). — IEEE. 2022. — с. 01—08.
134. *Stanovov V., Semenkin E.* Success rate-based adaptive differential evolution L-SRTDE for CEC 2024 competition // 2024 IEEE Congress on evolutionary computation (CEC). — IEEE. 2024. — с. 1—8.
135. *Stanovov V., Akhmedova S., Semenkin E.* Dual-population adaptive differential evolution algorithm l-ntade // Mathematics. — 2022. — т. 10, № 24. — с. 4666.

136. *Stanovov V., Kazakovtsev L., Semenkin E.* Hyper-heuristic approach for tuning parameter adaptation in differential evolution // *Axioms*. — 2024. — т. 13, № 1. — с. 59.
137. *Kennedy J., Eberhart R.* Particle swarm optimization // *Proceedings of ICNN'95-international conference on neural networks*. т. 4. — ieee. 1995. — с. 1942—1948.
138. *Clerc M., Kennedy J.* The particle swarm-explosion, stability, and convergence in a multidimensional complex space // *IEEE transactions on Evolutionary Computation*. — 2002. — т. 6, № 1. — с. 58—73.
139. *Computational intelligence: an introduction*. т. 2 / *A. P. Engelbrecht [и др.]*. — Wiley Online Library, 2007.
140. *Mendes R., Kennedy J., Neves J.* The fully informed particle swarm: simpler, maybe better // *IEEE transactions on evolutionary computation*. — 2004. — т. 8, № 3. — с. 204—210.
141. *Kessentini S.* Analysis and improvement of the binary particle swarm optimization // *Annals of Operations Research*. — 2025. — т. 351, № 1. — с. 101—131.
142. *Bohrer J. d. S., Dorn M.* Enhancing classification with hybrid feature selection: A multi-objective genetic algorithm for high-dimensional data // *Expert Systems with Applications*. — 2024. — т. 255. — с. 124518.
143. Dimensional decision covariance colony predation algorithm: global optimization and high-dimensional feature selection / *B. Xu [и др.]* // *Artificial Intelligence Review*. — 2023. — т. 56, № 10. — с. 11415—11471.
144. *Haldar K. L., Li H., Agrawal D. P.* A cluster-aware soft frequency reuse scheme for inter-cell interference mitigation in lte based femtocell networks // *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. — IEEE. 2013. — с. 1—6.
145. *Chen Q., Chen L.* DECRL: A Deep Evolutionary Clustering Jointed Temporal Knowledge Graph Representation Learning Approach // *Advances in Neural Information Processing Systems*. — 2024. — т. 37. — с. 55204—55227.
146. *Amirjanov A., Sadikoglu F.* Linear adjustment of a search space in genetic algorithm // *Procedia computer science*. — 2017. — т. 120. — с. 953—960.

147. *Banerjee A., Louis S. J.* A recursive clustering methodology using a genetic algorithm // 2007 IEEE Congress on Evolutionary Computation. — IEEE. 2007. — с. 2165—2172.
148. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data / A. M. Ikotun [и др.] // Information Sciences. — 2023. — т. 622. — с. 178—210.
149. *Tsutsui S., Goldberg D. E.* Search space boundary extension method in real-coded genetic algorithms // Information Sciences. — 2001. — т. 133, № 3/4. — с. 229—247.
150. *Baykasoğlu A., Ozsoydan F. B.* Dynamic optimization in binary search spaces via weighted superposition attraction algorithm // Expert Systems with Applications. — 2018. — т. 96. — с. 157—174.
151. *Rodzin S., Kureichik V., Rodzina L.* Co-evolutionary Self-adjusting Optimization Algorithm Based on Patterns of Individual and Collective Behavior of Agents // Computer Science On-line Conference. — Springer. 2022. — с. 254—266.
152. *Zhang J., Chung H. S.-H., Lo W.-L.* Clustering-based adaptive crossover and mutation probabilities for genetic algorithms // IEEE Transactions on evolutionary Computation. — 2007. — т. 11, № 3. — с. 326—335.
153. *Abdul-Rahman O. A., Munetomo M., Akama K.* An adaptive parameter binary-real coded genetic algorithm for constraint optimization problems: Performance analysis and estimation of optimal control parameters // Information Sciences. — 2013. — т. 233. — с. 54—86.
154. Evolutionary architecture search via adaptive parameter control and gene potential contribution / R. Shang [и др.] // Swarm and Evolutionary Computation. — 2023. — т. 82. — с. 101354.
155. A non-revisiting genetic algorithm based on a novel binary space partition tree / Y. Su [и др.] // Information Sciences. — 2020. — т. 512. — с. 661—674.
156. *Putra P. A., Tarigan J. T., Zamzami E. M.* Procedural 2d dungeon generation using binary space partition algorithm and l-systems // 2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA). — IEEE. 2023. — с. 365—369.

157. *Thakur M., Meghwani S. S., Jalota H.* A modified real coded genetic algorithm for constrained optimization // *Applied Mathematics and Computation*. — 2014. — т. 235. — с. 292—317.
158. *Nasab S. T. M., Abualigah L.* Improve Harris Hawkes optimizer algorithm via Laplace crossover // *Journal of Ambient Intelligence and Humanized Computing*. — 2024. — т. 15, № 4. — с. 2057—2072.
159. A comprehensive review of metaheuristic methods for the reconfiguration of electric power distribution systems and comparison with a novel approach based on efficient genetic algorithm / M. Mahdavi [и др.] // *IEEE Access*. — 2021. — т. 9. — с. 122872—122906.
160. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation / D.-C. Dang [и др.] // *Proceedings of the AAAI Conference on Artificial Intelligence*. т. 37. — 2023. — с. 12390—12398.
161. A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization / Y. Tian [и др.] // *IEEE/CAA Journal of Automatica Sinica*. — 2022. — т. 10, № 4. — с. 1048—1063.
162. A clustering based multiobjective evolutionary algorithm / H. Zhang [и др.] // *2014 IEEE Congress on evolutionary computation (CEC)*. — IEEE. 2014. — с. 723—730.
163. *Jørgensen M. S., Groves M. N., Hammer B.* Combining evolutionary algorithms with clustering toward rational global structure optimization at the atomic scale // *Journal of Chemical Theory and Computation*. — 2017. — т. 13, № 3. — с. 1486—1493.
164. Adaptive search space scaling in self-adaptive genetic algorithms for high-dimensional global black-box optimization problems / I. P. Malashin [и др.] // *Discover Artificial Intelligence*. — 2026. — DOI: [10.1007/s44163-026-01186-3](https://doi.org/10.1007/s44163-026-01186-3) ; URL: <https://doi.org/10.1007/s44163-026-01186-3> ; Published online: 7 April 2026.
165. *Малашин И. П., Тынченко В. С.* Применение методов кластерного анализа для динамической коррекции области поиска в генетическом алго-

- ритме // Сибирский аэрокосмический журнал. — 2025. — т. 26, № 3. — с. 318—333.
166. *Малашин И. П., Тынченко В. С.* Кластерно-ориентированный подход к оптимизации проектных решений в системах автоматизированного проектирования // Третья молодежная конференция с международным участием «Комплексная автоматизация проектирования и производства 2025» (КАПП(М)-2025) : сборник статей. — Москва, 2025. — с. 359—361.
167. Адаптивное исключение неэффективных кластеров в эволюционной оптимизации / И. П. Малашин [и др.] // XXX Байкальская Всероссийская с международным участием конференция «Информационные и математические технологии в науке и управлении» : сборник статей. — Иркутск, 2025. — с. 54—55.
168. *Малашин И. П.* Методика корректировки границ кластеров при динамическом формировании области поиска в генетическом алгоритме // Системы управления и информационные технологии. — 2025. — 3(101). — с. 96—100.
169. *Малашин И. П.* Процедура динамической модификации схемы бинарного кодирования индивидов в генетическом алгоритме // Моделирование, оптимизация и информационные технологии. — 2025. — т. 13, № 3. — DOI: [10.26102/2310-6018/2025.50.3.040](https://doi.org/10.26102/2310-6018/2025.50.3.040).
170. *Малашин И. П., Тынченко В. С.* Моделирование динамического изменения пространства поиска в эволюционных алгоритмах с использованием кластерного анализа // XII Международная научная Четаевская конференция «Аналитическая механика, устойчивость и управление» : сборник статей. — Казань, 2025.
171. *Малашин И. П., Сопов Е. А.* Модификация самонастраивающегося генетического алгоритма на основе динамической коррекции области поиска // Тридцать первая международная открытая научная конференция «Современные проблемы информатизации» (Modern Informatization Problems). — Yelm, WA, USA : Science Book Publishing House, 2026. — 1 ноября 2025 г. — 31 января 2026 г.

172. *Малашин И. П., Сопов Е. А.* Анализ причин немонотонной сходимости генетического алгоритма с динамической коррекцией области поиска // Системы управления и информационные технологии. — 2026. — 1(103). — с. 34—37.
173. *Mahdavi M., Jin Y., Suganthan P. N.* Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization // Technical Report. — 2007.
174. *Suganthan P. N., al. et.* Problem Definitions and Evaluation Criteria for the CEC 2010 Special Session and Competition on Constrained Real-Parameter Optimization // IEEE Congress on Evolutionary Computation (CEC), 2010. — IEEE, 2010. — с. 1—8.
175. Evaluating the performance of adaptive gainingsharing knowledge based algorithm on CEC 2020 benchmark problems / A. W. Mohamed [и др.] // 2020 IEEE congress on evolutionary computation (CEC). — IEEE. 2020. — с. 1—8.
176. Application of natural language processing and genetic algorithm to fine-tune hyperparameters of classifiers for economic activities analysis / I. Malashin [и др.] // Big Data and Cognitive Computing. — 2024. — т. 8, № 6. — с. 68.
177. Identifying and eliminating mislabeled training instances / C. E. Brodley, M. A. Friedl [и др.] // Proceedings of the National Conference on Artificial Intelligence. — 1996. — с. 799—805.
178. *Lemmerich F., Becker M.* pysubgroup: Easy-to-use subgroup discovery in python // Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part III 18. — Springer. 2019. — с. 658—662.
179. *Atzmueller M.* Subgroup discovery // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. — 2015. — т. 5, № 1. — с. 35—49.
180. *Kim M. P., Ghorbani A., Zou J.* Multiaccuracy: Black-box post-processing for fairness in classification // Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. — 2019. — с. 247—254.

181. A method of identifying and correcting miscoding, misclassification and misdiagnosis in diabetes: a pilot and validation study of routinely collected data / S. De Lusignan [и др.] // *Diabetic Medicine*. — 2010. — т. 27, № 2. — с. 203—209.
182. Статья 88. Камеральная налоговая проверка // *Налоговый кодекс Российской Федерации (часть первая)*: сайт. — Дата обращения: 16.05.2026. https://www.consultant.ru/document/cons_doc_LAW_19671/d063bb6c0a9ecaf015d02a2e02e66e8d0dd6ba21/.
183. *Федеральная налоговая служба*. Письмо от 09.06.2026 № ЕА-36-15/4924@ «О доведении Рекомендаций по проведению камеральных налоговых проверок». — 06.2026. — Дата публикации: 12.06.2026; подписано А. В. Егоричевым; дата обращения: 18.06.2026. https://www.nalog.gov.ru/rn77/about_fts/about_nalog/16630499/.
184. Confidence-Aware Rejection Class Minimization in Indirect Tax Audit Classification / I. P. Malashin [и др.] // *IEEE Access*. — 2024.
185. Anomaly Detection in Financial Data Using GA-Optimized MLP Models / I. Malashin [et al.] // *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*. — IEEE, 2024. — P. 278–285.
186. *Van der Maaten L., Hinton G.* Visualizing data using t-SNE. // *Journal of machine learning research*. — 2008. — т. 9, № 11.
187. *Sagi O., Rokach L.* Approximating XGBoost with an interpretable decision tree // *Information sciences*. — 2021. — т. 572. — с. 522—542.
188. *Stokanović S., Đukić D., Miljković N.* Robustness of XGBoost Algorithm to Missing Features for Binary Classification of Medical Data // *2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH)*. — IEEE, 2024. — с. 1—6.
189. An XGBoost-assisted evolutionary algorithm for expensive multiobjective optimization problems / F. Mao [и др.] // *Information Sciences*. — 2024. — т. 666. — с. 120449.
190. *Tang S.* The box office prediction model based on the optimized XGBoost algorithm in the context of film marketing and distribution // *PloS one*. — 2024. — т. 19, № 10. — e0309227.

191. *Lima Marinho T., Nascimento D. C. do, Pimentel B. A.* Optimization on selecting XGBoost hyperparameters using meta-learning // Expert Systems. — 2024. — т. 41, № 9. — e13611.

Приложение А

Результаты вычислительных экспериментов исследования метода динамической коррекции области поиска на эталонных тестовых функциях CEC2017

Таблица А1 – Сводная таблица средних значений функций CEC2017 для алгоритмов ГА, ГА+ДКОП, SelfCGA и SelfCGA+ДКОП

Функция	ГА	ГА+ДКОП	SelfCGA			SelfCGA+ДКОП		
	$D = 10$	$D = 10$	$D = 10$	$D = 30$	$D = 50$	$D = 10$	$D = 30$	$D = 50$
f_1	1.826e+10	1683.145	280.620	2176.234	1550.561	783.470	4343.632	3222.551
f_2	1.884e+08	5.131e+06	4163.989	6.134e+26	1.208e+56	4861.152	5.878e+20	3.392e+40
f_3	7560.042	2890.993	308.418	39214.669	109167.577	487.559	23648.406	67427.832
f_4	514.578	410.834	406.661	517.154	548.422	405.888	506.464	538.830
f_5	555.480	512.274	510.399	566.563	621.246	505.224	532.302	581.631
f_6	634.013	602.379	600.005	609.589	623.777	600.155	600.919	601.430
f_7	837.934	727.417	719.327	810.136	916.184	713.997	770.658	842.310
f_8	851.646	812.976	809.701	864.670	921.410	804.527	836.565	882.610
f_9	1419.679	913.254	900.000	1168.209	1905.505	900.000	900.849	915.219
f_{10}	2351.352	1365.254	1357.171	3371.840	5463.777	1032.656	3753.839	5825.344
f_{11}	1305.104	1119.292	1106.684	1176.167	1367.992	1102.608	1143.335	1273.912
f_{12}	1.040e+08	1.357e+06	20640.043	788404.315	5.339e+06	12435.558	2.116e+06	2.545e+06
f_{13}	269692.307	10969.347	6651.763	8262.688	3340.726	7932.134	12043.177	6178.622
f_{14}	1759.889	1648.591	1455.494	21471.729	244767.518	1501.059	92661.678	333914.988
f_{15}	7156.017	4419.921	2120.616	1837.052	5651.697	2719.733	9007.820	5157.807
f_{16}	1770.874	1643.033	1600.867	2435.274	3262.260	1617.685	2129.901	2858.332
f_{17}	1802.279	1719.875	1701.555	1965.345	2741.108	1712.841	1727.076	1777.941
f_{18}	110399.242	11382.387	10037.203	447981.914	1.221e+06	4003.753	976902.289	2.697e+06
f_{19}	5910.873	5828.056	2475.366	6622.204	16439.300	2679.868	10441.361	12817.336
f_{20}	2110.774	2015.703	2001.133	2266.818	2755.641	2004.948	2103.529	2380.280
f_{21}	2230.006	2225.167	2234.298	2366.513	2451.469	2213.298	2337.287	2361.546
f_{22}	2390.947	2308.657	2300.132	2301.816	7997.306	2292.416	2301.067	7176.709
f_{23}	2665.639	2613.909	2609.511	2715.399	2889.244	2606.193	2687.582	2792.559
f_{24}	2649.725	2700.771	2629.820	2888.819	3044.672	2663.769	2859.629	2961.384
f_{25}	3026.461	2946.483	2945.431	2887.679	3056.974	2925.363	2891.091	3048.225
f_{26}	3226.964	2992.465	2900.009	4337.898	5513.337	2905.149	4033.076	4477.553
f_{27}	3127.864	3089.502	3092.088	3220.600	3401.366	3092.850	3211.540	3293.450
f_{28}	3319.994	3280.772	3184.068	3241.129	3343.751	3154.870	3256.129	3347.528
f_{29}	3268.434	3171.788	3172.456	3357.309	3944.727	3146.014	3670.433	3221.751
f_{30}	4.080e+06	129103.435	39514.853	7718.049	1.103e+06	333959.646	13154.568	864739.711

Таблица А2 — Сводная таблица средних значений функций СЕС2017 для алгоритмов ВРСО и ВРСО+ДКОП

Функция	ВРСО			ВРСО+DSP		
	$D = 10$	$D = 30$	$D = 50$	$D = 10$	$D = 30$	$D = 50$
f_1	304876382.6	7.31×10^{10}	4.10×10^{11}	163919500	28243800	3.03×10^{10}
f_2	7596.255	1.19×10^{33}	4.28×10^{65}	866518.1	2.63×10^{29}	1.47×10^{54}
f_3	318.726	47636.29	164323.5	2278.549	38337.20	83487.89
f_4	408.119	2055.409	10578.33	406.468	533.865	1174.909
f_5	522.176	752.187	1076.573	533.063	685.236	898.911
f_6	603.655	655.827	693.671	611.286	612.372	649.295
f_7	717.727	867.686	1380.660	756.543	919.929	1147.650
f_8	809.252	1011.120	1359.591	825.364	983.580	1154.902
f_9	900.000	3623.705	22935.08	943.387	1019.182	3603.280
f_{10}	1682.203	7636.743	14326.22	1577.692	4271.632	8038.709
f_{11}	1105.059	1500.378	6457.959	1137.550	1363.474	2632.052
f_{12}	74946.61	7.89×10^8	5.68×10^{10}	4718976	8407965	5.34×10^7
f_{13}	8731.401	3.32×10^7	1.39×10^{10}	27145.24	372750.7	2.52×10^7
f_{14}	1445.831	36592.22	1.86×10^6	1550.421	42486.00	260580.5
f_{15}	1875.127	7958.541	7.33×10^6	2361.725	607416.5	80995.11
f_{16}	1637.457	3606.488	5993.220	1621.117	3249.453	4733.510
f_{17}	1713.570	2148.570	3720.183	1737.546	2134.067	3599.861
f_{18}	11928.67	1.05×10^6	1.50×10^7	11903.60	1.12×10^6	8.26×10^6
f_{19}	2924.661	2.77×10^6	55282.68	2827.364	1.78×10^7	150629.0
f_{20}	2013.241	2309.955	3253.177	2023.898	2364.360	3112.706
f_{21}	2205.343	2559.321	2907.653	2206.988	2471.719	2678.526
f_{22}	2282.279	3458.304	13997.70	2294.258	2315.139	9937.713
f_{23}	2652.229	3114.991	3861.234	2607.069	2849.803	3203.741
f_{24}	2508.308	3299.148	4152.711	2575.507	3030.928	3370.363
f_{25}	2918.843	3120.823	6715.147	2908.226	2915.248	3450.994
f_{26}	2905.013	7567.393	13050.85	2911.943	5536.997	7968.314
f_{27}	3121.450	3660.446	5416.080	3102.125	3325.367	3760.223
f_{28}	3140.268	3803.015	7654.692	3142.857	3274.952	3837.379
f_{29}	3207.201	4729.118	7108.179	3204.177	4012.186	5553.105
f_{30}	624605.2	2093266	5.03×10^8	429898.4	1.61×10^7	2.72×10^8

Таблица А3 — Сравнение средних значений функций СЕС2017 для двух вариантов алгоритма DE

Функция	bL-SRTDE			bL-SRTDE+ДКОП		
	$D = 10$	$D = 30$	$D = 50$	$D = 10$	$D = 30$	$D = 50$
$f1$	310238.000	3.22×10^9	3.31×10^{10}	225537.196	2.32×10^9	2.26×10^{10}
$f2$	132455.633	3.28×10^{27}	7.88×10^{58}	138582.440	2.18×10^{27}	5.77×10^{58}
$f3$	1791.205	144497.834	307391.948	1126.368	101656.540	219943.300
$f4$	616.245	701.857	960.688	406.917	531.074	763.779
$f5$	700.037	895.808	1261.347	507.216	651.645	878.952
$f6$	751.612	859.560	933.635	613.506	705.511	791.267
$f7$	1127.050	1207.387	1422.403	903.897	921.707	1203.239
$f8$	1230.325	1389.175	1801.281	922.991	1140.449	1516.196
$f9$	1128.012	1543.588	15392.198	1168.626	1562.214	16790.770
$f10$	2743.856	4459.586	7509.433	2052.573	3648.238	4809.596
$f11$	1399.166	1544.123	2313.330	1100.971	1319.425	1571.748
$f12$	14942.713	61792.532	413191.160	10630.568	57077.264	270932.475
$f13$	14467.518	54684.706	79593.916	11943.616	37907.168	66092.766
$f14$	1755.296	1644.051	2216.555	1481.318	1712.553	1768.562
$f15$	2629.378	2228.239	2770.111	1938.398	2139.420	2895.339
$f16$	2294.220	2335.730	2242.314	1607.465	2105.528	2118.845
$f17$	2029.467	1946.780	2018.909	1700.908	1832.083	2007.101
$f18$	5733.419	12721.452	17092.221	4243.765	11739.860	18274.227
$f19$	2262.363	3768.535	6187.976	2018.595	3945.537	4789.915
$f20$	2153.145	2268.528	2665.652	2000.034	2017.020	2268.663
$f21$	2423.654	2296.647	2671.836	2200.512	2220.750	2264.415
$f22$	2712.592	2598.918	4371.201	2300.307	2318.406	2998.130
$f23$	2858.879	2786.256	3309.495	2600.004	2679.804	2830.561
$f24$	2953.375	2995.996	3203.215	2766.280	2779.232	2919.223
$f25$	3220.993	3013.413	3345.631	2930.401	2893.239	2950.579
$f26$	4304.342	4619.211	5165.223	3366.545	3312.505	4484.255
$f27$	3358.115	3576.704	3910.909	3111.092	3283.509	3299.090
$f28$	3631.762	3665.336	3892.980	3396.882	3557.013	3381.237
$f29$	4491.894	5286.378	7173.305	3146.693	3553.186	4491.878
$f30$	13754.060	447995.844	4.02×10^7	11040.237	308765.991	4.19×10^7

Приложение В

Акты об использовании результатов

Акт

Об использовании результатов диссертационного исследования
Малашина Ивана Павловича

Настоящим актом подтверждается, что алгоритмическое и программное обеспечение, предложенное Малашиным Иваном Павловичем в рамках диссертационного исследования на соискание ученой степени кандидата технических наук, используется в информационной системе поддержки анализа данных налоговых проверок юридических лиц (в проекте «Перевод бизнеса» НОЦ «ФНС России и МГТУ им. Н.Э. Баумана»).


Предложенные автором алгоритмы оптимизации совместно с методами машинного обучения и искусственного интеллекта обеспечили более высокую точность выявления недостоверных данных по сравнению с ранее использованным подходом, что в итоге исключили более 3000 ручных проверок, трудозатраты которых оцениваются в 15 - 30 тыс. человеко-часов.

Старший научный сотрудник
НОЦ «ФНС России и МГТУ им. Н.Э. Баумана»
НОЦ «Технологии искусственного интеллекта»
Московского государственного технического
университета им. Н.Э. Баумана, кандидат
физико-математических наук



Виталий
Александрович
Красиков



• ВЕРНО •
НАЧАЛЬНИК ОТДЕЛА

ОТДЕЛ ОРГАНИЗАЦИИ РАБОТЫ
ЕДИНОГО ЦЕНТРА УСЛОН
М ГТУ ИМ. Н.Э. БАУМАНА

Акт

**об использовании результатов диссертационного исследования
Малашина Ивана Павловича**

Настоящим актом подтверждается, что предложенный Малашиным И.П. новый алгоритм оптимизации с использованием динамической коррекции области поиска был использован при создании алгоритмического и программного обеспечения интеллектуальной системы анализа госконтрактов. Применение подхода, предложенного Иваном Павловичем в рамках диссертационного исследования на соискание ученой степени кандидата технических наук, обеспечило повышение точности построения моделей машинного обучения относительно результатов, полученных стандартными алгоритмами, используемых в НОЦ «ФНС России и МГТУ им. Н.Э. Баумана» в проектах «Разработка системы автоматического распределения рисков, схем и задач с применением искусственного интеллекта» и «Создание алгоритма адаптации данных об импортируемых и экспортируемых товарах для мониторинга цен».

Использование результатов диссертационного исследования Малашина Ивана Павловича имеет большой потенциал для повышения эффективности построения и эксплуатации подобных информационных систем.

Старший научный сотрудник
НОЦ «ФНС России и МГТУ им. Н.Э. Баумана»
НОЦ «Технологии искусственного интеллекта»
Московского государственного технического *Красиков*
университета им. Н.Э. Баумана, кандидат
физико-математических наук

Виталий
Александрович
Красиков



« ВЕРНО »
НАЧАЛЬНИК ОТДЕЛА
Красиков КИРЯЖЕНА А.И.
ОТДЕЛ ПО ОРГАНИЗАЦИИ РАБОТЫ
ЕДИНОЙ ПРЕМИАЛЬНОЙ УЧСНА
М ГТУ ИМ. Н.Э. БАУМАНА