

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

На правах рукописи



Сопов Евгений Александрович

**ОБОБЩЕННЫЙ МЕТОД СИНТЕЗА ГИПЕРЭВРИСТИЧЕСКИХ
ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ОПТИМИЗАЦИИ
СЛОЖНЫХ СИСТЕМ**

05.13.01 – Системный анализ, управление и обработка информации
(космические и информационные технологии)

ДИССЕРТАЦИЯ

на соискание учёной степени доктора технических наук

Научный консультант

Семенкин Евгений Станиславович

доктор технических наук, профессор

Красноярск – 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ОПТИМИЗАЦИИ СЛОЖНЫХ СИСТЕМ	19
1.1 Прикладной системный анализ и сложные задачи оптимизации	19
1.2 Формализация задач оптимизации, возникающих в прикладном системном анализе.....	25
1.3 Эволюционные алгоритмы решения сложных задач оптимизации	29
1.4 Обзор современных подходов к проектированию и настройке эволюционных алгоритмов и формализация задачи синтеза эволюционных алгоритмов	34
1.5 Обзор современных подходов к оценке и сравнению эффективности эволюционных алгоритмов	41
Выводы по главе 1.....	48
ГЛАВА 2. ГИПЕРЭВРИСТИКА НА ОСНОВЕ МЕТОДА ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ СИНТЕЗА ОПЕРАТОРОВ СЕЛЕКЦИИ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ	51
2.1 Метод генетического программирования для формирования элементарных эвристик эволюционных алгоритмов.....	51
2.2 Конструктивная офлайн гиперэвристика синтеза оператора селекции на основе метода генетического программирования	64
2.3 Экспериментальное исследование предложенного метода.....	67
2.4 Решение практической задачи синтеза оператора селекции для идентификации партий электронных компонентов аэрокосмического применения	79
Выводы по главе 2.....	88
ГЛАВА 3. СЕЛЕКТИВНАЯ ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ НЕСТАЦИОНАРНОЙ ОПТИМИЗАЦИИ	90

3.1 Проблема нестационарности в задачах оптимизации сложных систем и обзор подходов к ее решению.....	91
3.2 Селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации	103
3.3 Экспериментальное исследование предложенного метода.....	108
3.4 Решение практической задачи нестационарной оптимизации для идентификации параметров боковой динамики движения малого беспилотного летательного аппарата в реальном времени	117
Выводы по главе 3.....	129
ГЛАВА 4. СЕЛЕКТИВНАЯ ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ИДЕНТИФИКАЦИИ МНОЖЕСТВА ЭКСТРЕМУМОВ.....	132
4.1 Проблема аппроксимации множества экстремумов в задачах оптимизации сложных систем и обзор подходов к ее решению	133
4.2 Селективная гиперэвристика синтеза ЭА идентификации множества экстремумов	142
4.3 Экспериментальное исследование предложенного метода.....	147
4.4 Решение практических задач аппроксимации множества экстремумов при поддержке принятия решений при формировании оптимальной кредитной политики коммерческого банка и при управлении инвестициями производственного предприятия ОПК	160
4.5 Решение практической задачи аппроксимации множества экстремумов при проектировании систем на нечеткой логике	166
Выводы по главе 4.....	176
ГЛАВА 5. СЕЛЕКТИВНАЯ ОНЛАЙН ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ДЛЯ ЗАДАЧ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ БОЛЬШОЙ РАЗМЕРНОСТИ	179
5.1 Проблема большой размерности в задачах глобальной оптимизации с алгоритмически заданными целевыми функциями и обзор подходов к ее решению	180

5.2	Селективная гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности.....	196
5.3	Экспериментальное исследование предложенного метода.....	199
5.4	Решение практической задачи повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах	210
5.5	Решение практической задачи проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов.....	218
	Выводы по главе 5.....	226
ГЛАВА 6. ОБОБЩЕННЫЙ МЕТОД ПРИМЕНЕНИЯ ГИПЕРЭВРИСТИК ПРИ РЕШЕНИИ СЛОЖНЫХ ЗАДАЧ ОПТИМИЗАЦИИ		229
6.1	Обобщенный метод применения гиперэвристик.....	229
6.2	Ограничение теоремы «бесплатных завтраков не бывает».....	235
	Выводы по главе 6.....	237
ЗАКЛЮЧЕНИЕ		238
СПИСОК ЛИТЕРАТУРЫ		242
Приложение А.	Результаты численных экспериментов исследования эффективности метода 1	288
Приложение Б.	Результаты численных экспериментов исследования эффективности метода 2.....	291
Приложение В.	Эталонные тестовые задачи мультимодальной оптимизации с бинарными переменными.....	299
Приложение Г.	Результаты численных экспериментов исследования эффективности метода 3.....	301
Приложение Д.	Результаты численных экспериментов исследования эффективности метода 4.....	314
Приложение Е.	Акты об использовании результатов	323

ВВЕДЕНИЕ

Актуальность темы исследования связана с постоянно возрастающей сложностью задач прикладного системного анализа. Развитие вычислительной техники и повышение эффективности методов анализа сложных систем в последние годы привели к тому, что при решении задач анализа, моделирования и управления сложными техническими системами, аналитикам требуется реагировать на большее число внешних и внутренних факторов, исследовать более сложные статические и динамические функции подсистем и элементов систем. Аналогично усложняются и задачи системного анализа при поддержке принятия решений в организационных системах. При этом практически на всех этапах прикладного системного анализа возникают постановки задач, которые сводятся к оптимизационным. К ним относятся, как исходно оптимизационные постановки задач (задачи математического программирования и исследования операций, задачи параметрического проектирования, задачи поиска эффективных альтернатив и другие), так и обеспечивающие задачи (структурная и параметрическая оптимизация моделей, синтез и настройка применяемых методов и алгоритмов, оптимальное управление и т.д.). Очевидно, что свойства сложных систем отражаются на свойствах возникающих задач оптимизации, которые можно отнести к классу сложных задач оптимизации.

Отсутствие или неполнота информации об исследуемой системе определяет общий признак сложных задач оптимизации — невозможность адекватно выполнить и применить на практике результаты математического анализа, либо сложность такого анализа оказывается сравнимой со сложностью самой задачи оптимизации. Целевые функции подобных задач рассматриваются как модели «черного ящика», а соответствующий класс задач – оптимизация «черного ящика» (*Black-box optimization*). В свою очередь, отсутствие информации о линейности, гладкости, выпуклости и прочих свойствах целевых функций исключает или существенно затрудняет применение традиционных методов математического программирования. Кроме того, во многих современных задачах оптимизации

оценка целевых функций в принципе осуществляется алгоритмически, например, в ходе компьютерных симуляций или экспериментов, а также в ходе экспертной оценки.

Измерения наблюдаемых факторов в сложных системах могут быть выражены в разных шкалах (числовые непрерывные и дискретные, шкалы порядка, наименований и перестановок), что в свою очередь затрудняет применение многих традиционных методов оптимизации. Среди других свойств, определяющих сложную систему, выделяют число факторов и связей. Увеличение количества переменных экспоненциально увеличивает пространство поиска («проклятье размерности»), а в случае немасштабируемых задач, меняет ландшафт целевой функции. Соответствующий класс называется задачами глобальной оптимизации большой размерности (*Large Scale Global Optimization*). Другое свойство сложных систем – нестационарность, связанная с внутренними и внешними изменениями. Изменения в среде могут произвольно изменять ландшафт целевой функции, меняя положения и свойства глобального и локальных оптимумов, создавая новые и уничтожая ранее существовавшие оптимумы. Изменения могут происходить во время решения задачи оптимизации, при этом момент времени и характер изменений заранее не известны и, в общем случае, могут быть случайными. Соответствующий класс задач называется глобальной оптимизацией в нестационарной среде (*optimization in the non-stationary environment*). В практических задачах анализа сложных систем для повышения устойчивости решений или для обеспечения разнообразия альтернатив для лица принимающего решения требуется найти не единственное оптимальное решение задачи, а некоторое подмножество глобальных и локальных решений задачи оптимизации. Данное требование определяет класс задач оптимизации, связанных с аппроксимацией множества оптимумов (*Multimodal Optimization*).

В настоящее время для решения выделенных классов задач большую популярность получили и успешно применяются на практике различные метаэвристические алгоритмы прямого («слепого») поиска. Наибольшую эффективность для многих классов сложных задач оптимизации демонстрируют

стохастические популяционные метаэвристики на основе моделирования процессов из биологии, физики, химии и других естественных процессов. Наиболее известной и применяемой на практике метаэвристикой являются эволюционные алгоритмы (ЭА), которые сегодня рассматриваются как самостоятельное направление в оптимизации – эволюционная оптимизация. Тем не менее, эффективность применения эволюционных алгоритмов для решения сложных задач оптимизации зависит от корректно выбранной структуры (типа) эволюционного алгоритма и настройки параметров алгоритма. Сегодня для решения задачи настройки параметров предложены и исследованы различные подходы самонастройки, самоадаптации и самоконфигурации. Как показывает практика, большой эффект можно получить за счет создания нестандартных комбинаций элементарных эвристик, т.е. проектирования новых эволюционных алгоритмов. В настоящее время это направление только развивается, а создание новых эволюционных метаэвристик осуществляется в ручном режиме на основе опыта и интуиции исследователя.

В области машинного обучения активно развивается направление автоматизации синтеза алгоритмов, цель которого представить задачу проектирования алгоритма как самостоятельную задачу машинного обучения. Аналогичный подход возник относительно недавно в области комбинаторной оптимизации и получил название гиперэвристики. Гиперэвристику можно рассматривать как метаэвристику верхнего уровня, решающую задачу построения (управления, поиска, выбора) метаэвристики нижнего уровня для решения конкретной задачи оптимизации. На сегодня гиперэвристики синтеза эволюционных алгоритмов для решения описанных выше задач оптимизации сложных систем практически не исследованы, поэтому тема выбранного исследования актуальна для развития теории и практики системного анализа.

Степень разработанности темы исследования. Систематизация гиперэвристических подходов и современная терминология предложена Бёрком Э. в 2009. В настоящее время наиболее цитируемые исследования в области гиперэвристик связаны с работами таких авторов, как Бёрк Э., Росс П., Пиллей Н.,

Эпитропакис М., Лиссовой А. Проблемой разработки и исследования гиперэвристик занимаются международная исследовательская группа COSEAL (*CO*nfiguration and *SE*lection of *AL*gorithms), группа «Исследование операций» Лондонского университета королевы Марии, проблеме посвящены ежегодные специальные сессии ведущих конференций по эволюционным вычислениям GECCO (*Genetic and Evolutionary Computation Conference by ACM*), CEC (*IEEE Congress on Evolutionary Computation*) и PPSN (*Parallel Problem Solving from Nature*). Среди российских исследователей, внесших вклад в развитие теории и практики ЭА и их применения при решении сложных задач оптимизации необходимо отметить Шалыто А.А., Буздalова М.В., Скобцова Ю.А., Демидову Л.А., Горнова А.Ю., Курейчика В.М., Карпенко А.П., Дивеева А.И., Пантелеева А.В., Гладкова Л.А. В их работах, помимо традиционных эволюционных алгоритмов, систематизированы и обсуждаются гибридные подходы, островные модели и методы самонастройки и самоконфигурации. В Красноярске теорией и практикой автоматизации проектирования эволюционных алгоритмов занимается научная школа Семенкина Е.С.

Идея настоящего диссертационного исследования состоит в обосновании и совершенствовании теории, систематизирующей идеи и принципы создания и комбинирования элементарных эвристик различных эволюционных алгоритмов или выбора среди известных комбинаций (известных эволюционных алгоритмов) как при проектировании эволюционных алгоритмов для решения частных задач или класса задач оптимизации, так и адаптивно в процессе решения конкретной задачи оптимизации.

Объектом диссертационного исследования является проблема автоматизированного (с минимальным участием или без участия человека) синтеза эволюционных алгоритмов решения сложных задач глобальной оптимизации с алгоритмически заданными целевыми функциями, включая задачи глобальной оптимизации большой размерности, задачи аппроксимации множества экстремумов и задачи оптимизации в нестационарной среде, а **предметом**

исследования – методы и алгоритмы решения данных задач на основе гиперэвристик.

Целью диссертационного исследования является расширение области применения и повышение обоснованности выбора методов эволюционной оптимизации сложных систем за счет адаптивного синтеза эволюционных алгоритмов с помощью гиперэвристик.

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

1. Провести системный анализ проблемы синтеза ЭА и выполнить систематизацию подходов на основе эволюционных мета- и гиперэвристик, формализовать цели синтеза новых алгоритмов для выбранных классов задач оптимизации и критерии оценки эффективности ЭА, полученных в результате синтеза.

2. Обосновать и разработать алгоритмы синтеза ЭА для исследуемых в диссертации классов задач оптимизации, провести численные исследования их эффективности и сравнительный анализ эффективности с известными подходами и другими специальными подходами.

3. Провести апробацию предложенных подходов путем решения прикладных задач оптимизации и сравнительный анализ с результатами, полученными традиционными подходами.

4. Обобщить результаты диссертационного исследования, обосновать и формализовать обобщенный метод синтеза ЭА для решения задач оптимизации сложных систем с помощью разработанных гиперэвристик.

Научная новизна работы состоит в следующем:

1. Впервые систематизированы знания о способах синтеза эволюционных алгоритмов решения сложных задач глобальной оптимизации с алгоритмически заданными функциями, предложена общая методика применения гиперэвристик синтеза ЭА для решения этих задач. Обобщенный метод позволяет адаптивно создавать новые эффективные ЭА в зависимости от целей и требований к решению задачи оптимизации и управлять применением предложенных гиперэвристик в зависимости от комбинации свойств исследуемой сложной системы.

2. Предложена и исследована новая конструктивная офлайн гиперэвристика для синтеза операторов селекции ЭА на основе алгоритма генетического программирования. Гиперэвристика отличается от известных подходов способом представления решений и вариантами оценки пригодности синтезируемых операторов.

3. Впервые предложена и исследована селективная онлайн гиперэвристика синтеза ЭА оптимизации в нестационарной среде, объединяющая идеи метода портфолио и идеи метода адаптации вероятностей на уровне популяции ЭА. Новая гиперэвристика позволяет идентифицировать изменения в среде и адаптировать ЭА путем формирования эффективной комбинации эвристик нестационарной оптимизации.

4. Впервые предложена и исследована селективная гиперэвристика синтеза ЭА идентификации множества экстремумов, включающая базовые эвристики, которые не используют в явном виде какую-либо информацию о свойствах целевой функции и пространства поиска. Новая гиперэвристика позволяет адаптивно формировать эффективную комбинацию эвристик поддержки разнообразия в популяции на основе анализа скопления решений в областях притяжения оптимумов.

5. Предложена и исследована новая эвристика для группировки переменных на основе алгоритма оценки распределений для применения бинарного генетического алгоритма в задаче глобальной оптимизации большой размерности. Новая эвристика на основе анализа распределения генов позволяет определять подмножество компонентов вектора решения, значения которых могут быть зафиксированы, сокращая таким образом пространство поиска.

6. Предложена и исследована новая эвристика, использующая распределения вероятностей для адаптивной группировки переменных в компоненты переменного размера на этапе декомпозиции в кооперативной коэволюции. Новая эвристика является обобщением метода DECC-G.

7. Впервые предложена и исследована селективная онлайн гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности на основе

островной модели ЭА. В новой гиперэвристике каждый остров реализует свою декомпозицию задачи, при этом гиперэвристика позволяет адаптивно управлять эвристиками декомпозиции путем изменения размеров островов.

Теоретическая значимость работы заключается в получении новых знаний о способах решения задач оптимизации сложных систем и эффективности их применения. Систематизация подходов к синтезу эволюционных алгоритмов и результаты исследования эффективности конкретных алгоритмов решения сложных задач оптимизации разных классов, дополняют знания об эвристических методах оптимизации. Метод синтеза эволюционных алгоритмов позволяет формировать новые комбинации элементарных эвристик для решения конкретных задач оптимизации, а исследование сформированных эффективных комбинаций эвристик может быть использован при проведении фундаментального анализа свойств класса задач, изначально представленных моделью типа «черный ящик».

Практическая значимость работы обусловлена практикой системного анализа, связанной с исследованием и управлением сложными системами, для которых характерно сочетание различных свойств, затрудняющих выбор и применением многих традиционных методов решения возникающих задач оптимизации. Использование предложенных в работе гиперэвристик позволяет без участия или с минимальным вовлечением предметного специалиста формировать новые элементарные эвристики и создавать комбинации эвристик (новые метаэвристики), которые являются эффективными для решения конкретных задач. В случае решения повторяющихся задач или множества задач одного класса, использование разработанных гиперэвристик обеспечит синтез эволюционного алгоритма эффективного в среднем на выбранном классе. Для задач оптимизации в нестационарной среде, применение гиперэвристик обеспечивает создание эволюционных алгоритмов, способных отслеживать изменения целевой функции и адаптивно формировать комбинацию элементарных эвристик эффективных в новых изменившихся условиях. Автоматизированный синтез алгоритмов оптимизации позволит предметным специалистам сфокусироваться на исходных задачах прикладного системного анализа.

В работе показано, что предложенные подходы позволяют повысить точность решения задач глобальной оптимизации большой размерности в среднем по сравнению с точностью, усредненной по базовым метаэвристикам. Это означает, что в случае решения задач с алгоритмически заданными целевыми функциями, когда обоснование выбора конкретного эволюционного алгоритма затруднено, использование предложенных гиперэвристик более предпочтительно, чем практически случайный выбор одного из базовых подходов. Аналогичный эффект наблюдается при решении задач аппроксимации множества экстремумов – предложенный подход позволяет получать лучшую аппроксимацию (большой процент идентифицированных оптимумов) в среднем, и при решении задач оптимизации в нестационарной среде – лучшее значение онлайн-ошибки (разницы между лучшим найденным решением и истинным глобальным оптимумом на каждой итерации алгоритма) в среднем.

Основные идеи и обобщенный метод синтеза эволюционных алгоритмов обеспечивают специалистов в области эволюционной оптимизации инструментом для создания новых гиперэвристик для решения прикладных задач оптимизации других классов и различных комбинаций исследованных в работе классов.

Практическая реализация результатов. С помощью предложенных в работе подходов синтеза эволюционных алгоритмов на основе гиперэвристик решены прикладные задачи оптимизации. Разработанные алгоритмы реализованы в виде программных систем для ЭВМ, которые используются в исследовательских и практических целях.

Предложенные в диссертационном исследовании методы проектирования эволюционных алгоритмов были использованы при решении следующих прикладных задач (акты использования и внедрения представлены в Приложении Е):

– проектирование алгоритмического и программного обеспечения для системы автоматизированного формирования и контроля специальных партий электрорадиоизделий космического применения, АО «Испытательный технический центр НПО-ПМ» (г. Железногорск);

- проектирование и внедрение системы поддержки инвестиционного планирования для задач внешнего и конкурсного управления, ООО «Инвент» (г. Красноярск);

- проектирование алгоритмического и программного обеспечения системы диспетчеризации распределенной энергетической системы» АО «Назаровская ГРЭС» (г. Назарово).

Научные и научно-технические результаты были получены и были использованы в рамках НИР при руководстве следующими проектами:

- ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы», Российско-германский проект «Разработка эффективного алгоритмического обеспечения для автоматизированного проектирования распределенных мультилингвистических систем поддержки электронного документооборота на облачных вычислениях» госконтракт N14.740.12.1341 (2011-2013);

- грант Президента РФ для молодых кандидатов наук «Самоконфигурируемая метаэвристика решения задач нестационарной оптимизации стохастическими поисковыми алгоритмами» МК-3285.2015.9 (2015-2016);

- государственное задание по проекту «Разработка и исследование самоконфигурируемых гиперэвристик решения сложных задач нестационарной мультимодальной оптимизации бионическими алгоритмами» №2.1676.2017/ПЧ (2017-2019).

Отдельные результаты работы были использованы при проведении работ в рамках НИР: ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы», Госбюджетная НИР по тематическому плану ЕЗН СибГАУ «Разработка теоретических основ эволюционного проектирования интеллектуальных информационных технологий анализа данных», 8.4368.2011 (2011-2013), ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы», Российско-германский

проект «Распределенные интеллектуальные информационные системы обработки и анализа мультилингвистической информации в диалоговых информационно-коммуникационных системах», госконтракт № 1.519.11.4002 (2011-2013), ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы», Российско-германский проект «Математическое и алгоритмическое обеспечение автоматизированного проектирования аппаратно-программных комплексов интеллектуальной обработки мультилингвистической информации в распределенных высокопроизводительных системах космического назначения», госконтракт № 16.740.11.0742 (2011-2013), ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы», проект «Разработка протокола безопасного обмена данными в распределенной информационно-вычислительной системе на основе технологии защиты с использованием движущейся цели», госконтракт № 14.574.21.0126 (2014-2016), ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы», проект «Распределенные самоконфигурируемые многоагентные технологии проектирования и управления интеллектуальными информационными сетями», госконтракт № 14.574.21.0037 (2014-2015), Задание на выполнение научно-исследовательской работы в рамках проектной части государственного задания в сфере научной деятельности, проект «Алгоритмическое обеспечение автоматизированного проектирования высоконадежных систем анализа, поддержки и управления огневými испытаниями жидкостных ракетных двигателей на базе самоконфигурируемых интеллектуальных информационных технологий», № 2.1889.2014/К (2014-2016), Грант РФФИ «Разработка и исследование методов автоматизированного проектирования технологий интеллектуального анализа данных на основе самоконфигурируемых эволюционных и бионических алгоритмов оптимизации» № 16-01-00767 а (2016-2018), Грант РФФИ «Алгоритмическое обеспечение интеллектуального анализа и автоматизированного проектирования предикторов показателей статистической отчётности на основе методов вычислительного интеллекта для Федеральной

службы государственной статистики по Красноярскому краю (Краснояркстат)», 16-41-240822 р_а (2016-2018), грант РФФИ на российско-австрийский проект «Методы машинного обучения для идентификации свойств задач глобальной оптимизации в нестационарной среде и автоматической адаптации эволюционных и бионических алгоритмов» №21-51-14003 АНФ_а (2021-2023) и другие.

Результаты диссертационного исследования в целом сформулированы и представлены к защите в рамках государственного задания по проекту «Развитие теории самоконфигурирующихся алгоритмов машинного обучения для моделирования и прогнозирования характеристик компонентов сложных систем» № FEFE-2020-0013 (2020-2022).

Методы исследования. Основные теоретические и практические результаты получены с применением методов прикладного системного анализа, исследования операций, теории оптимизации, теории вероятностей и математической статистики, теории алгоритмов и методов программирования.

Основные положения, выносимые на защиту:

1. Конструктивная офлайн гиперэвристика для синтеза операторов селекции ЭА на основе алгоритма генетического программирования позволяет автоматически создавать новые способы отбора индивидов в ЭА, которые для исследуемого множества задач глобальной оптимизации с алгоритмически заданными целевыми функциями обеспечивают большую среднюю эффективность ЭА по сравнению с использованием стандартных операторов селекции по критериям точности нахождения или скорости нахождения глобального оптимума с заданной точностью.

2. Селективная онлайн гиперэвристика синтеза ЭА оптимизации в нестационарной среде обеспечивает в среднем лучшее значение по критерию онлайн-ошибки независимо от комбинации типов изменений в среде по сравнению со значением, усредненным по результатам применения отдельных базовых эвристик.

3. Селективная гиперэвристика синтеза ЭА идентификации множества экстремумов позволяет получить в среднем больший процент

идентифицированных оптимумов по сравнению со значением, усредненным по результатам применения отдельных базовых эвристик для задач как с бинарными переменными, так и с вещественными.

4. Селективная гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности с алгоритмически заданными целевыми функциями превосходит по средней точности нахождения глобального оптимума оценку случайного выбора ЭА на основе одной из базовых эвристик и оценку случайного выбора одного из ведущих ЭА для задач большой размерности.

5. Обобщенный метод синтеза ЭА обеспечивает специалистов в области эволюционной оптимизации эффективным инструментом для создания новых гиперэвристик, расширяя область применения ЭА решения сложных задач оптимизации.

Степень достоверности полученных в работе результатов и выводов обеспечивается корректным применением методов статистической обработки и анализа данных экспериментального исследования алгоритмов оптимизации на репрезентативном множестве эталонных задач оптимизации, результатами решения прикладных задач, а также многократной апробацией на ведущих международных конференциях.

Апробация. Основные положения и результаты диссертационного исследования представлялись и прошли всестороннюю апробацию на всероссийских, международных и зарубежных научных и научно-практических конференциях и семинарах, включая:

- The Genetic and Evolutionary Computation Conference (GECCO, 2019, Прага, Чехия)
- IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2015, Кейптаун, ЮАР, IEEE SSCI 2020, Канберра, Австралия),
- International Conference on Swarm Intelligence (6th ICSI 2015, Пекин, Китай, 7th ICSI 2016, Бали, Индонезия, 9th ICSI 2018, Шанхай, Китай),
- 7th World Congress on Nature and Biologically Inspired Computing (NaBIC 2015, Питермарицбург, ЮАР),

– International Conference on Evolutionary Computation Theory and Applications (ECTA) в рамках International Joint Conference on Computational Intelligence (IJCCI) (7th ECTA/IJCCI 2015, Лиссабон, Португалия, 8th ECTA/IJCCI 2016, Порто, Португалия, 9th IJCCI 2017, Мадейра, Португалия, 10th IJCCI 2018, Севилья, Испания, 11th IJCCI 2019, Вена, Австрия),

– International Conference on Informatics in Control, Automation and Robotics (11th ICINCO 2014, Вена, Австрия, 12th ICINCO 2015, Кольмар, Франция, 13th ICINCO 2016, Лиссабон, Португалия, 15th ICINCO 2018, Порто, Португалия, 16th ICINCO 2019, Прага, Чехия),

– 4th International Congress on Advanced Applied Informatics (IIAI-AAI 2015, Окаяма, Япония),

– International Conference on Applied Mathematics and Computational Science (ICAMCS 2018, Будапешт, Венгрия),

– 2019 IEEE International Conference on Information Technologies, InfoTech-2019, София, Болгария,

– International Conference on Computer Science and Artificial Intelligence (ICCSAI 2014, Ухань, Китай), International Conference on Network Security and Communication Engineering (NSCE 2014, Гонконг, Китай),

– International Workshop on Mathematical Models and their Applications (IWMMA 2014-2020, Красноярск, Россия),

– 9th International Siberian Conference on Control and Communications (SIBCON 2011, Красноярск, Россия),

– Международная научно-практическая конференция «РЕШЕТНЕВСКИЕ ЧТЕНИЯ» (2009-2020, Красноярск, Россия),

– Пятая Международная конференция «Системный анализ и информационные технологии» (2013, Красноярск, Россия) и др.

Основные результаты исследований были отмечены Правительством и Законодательным собранием Красноярского края Государственной премией Красноярского края в области профессионального образования в 2016 году.

Диссертационная работа в целом обсуждалась на научных семинарах кафедры «Системного анализа и исследования операций» Сибирского университета науки и технологий (Красноярск) и факультета информационных технологий и программирования Университета ИТМО (Санкт-Петербург).

Публикации. По материалам диссертации опубликовано более 60 работ, в том числе 20 статей в российских рецензируемых периодических изданиях, рекомендуемых ВАК РФ для опубликования основных научных результатов диссертационных исследований, 39 работ в зарубежных изданиях, включенных в международную базу цитирования Scopus, 26 работ в зарубежных изданиях, включенных в международную базу цитирования Web of Science. В Роспатенте зарегистрированы 24 программы для ЭВМ.

Диссертация соответствует пункту 4 паспорта специальности 05.13.01 – разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации.

Структура и объем работы. Диссертация изложена на 325 страницах, состоит из введения, шести глав, заключения, списка литературы из 444 наименований и шести приложений.

ГЛАВА 1. ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ОПТИМИЗАЦИИ СЛОЖНЫХ СИСТЕМ

В Главе 1 проведен анализ свойств сложных систем и их отражение на свойствах задач оптимизации, возникающих на различных этапах прикладного системного анализа. Проведен анализ современных подходов к преодолению выявленных трудностей задач оптимизации, показаны сложности отнесения подобных задач к известным классам задач, изучаемых в математическом программировании и проблемы применения традиционных методов оптимизации. Рассмотрены традиционные подходы из области эволюционной оптимизации и современные подходы проектирования эволюционных алгоритмов, даны определения эвристического, метаэвристического и гиперэвристического подходов в оптимизации. Проведен обзор современных подходов и обоснован выбор способов оценки и сравнения эффективности эволюционных алгоритмов оптимизации.

1.1 Прикладной системный анализ и сложные задачи оптимизации

Системный анализ (СА) является областью научной и прикладной деятельности, направленной на анализ и проектирование, управление и поддержку принятия решений в сложных технических, организационных и социально-экономических системах [1-5]. В настоящее время термин «системный анализ» используется в разных аспектах, в самом широком смысле обозначая научный метод познания [6, 7], а в узком – конкретные методы и приемы, применяемые в предметной области, например, при проектировании систем управления и обработки информации [8-11]. Среди российских ученых, внесших вклад в развитие теории и практики СА стоит отметить Моисеева Н.Н. [12], Емельянова С.В. [13], Тарасенко Ф.П. [14], Антамошкина А.Н. [15]. Работы вышеперечисленных и многих других исследователей в большей степени связаны

с вопросами кибернетики и посвящены разработке и применению формальных методов исследования операций, математического моделирования и управления сложными системами.

В работах Оптнера С.Л. [16], Эмери Ф.Э. и Акоффа Р.Л. [17] вводятся такие термины как «проблема» и «целеустремленные системы», что определило развитие нового направления в области СА – прикладного системного анализа (ПСА). В отличие от традиционных подходов, использующих в качестве объекта исследования заранее определенную, «стандартную» систему (например, финансовая система, система управления и т.д.) или формальную организацию, в ПСА центральное место занимает прикладная проблема, сформулированная субъектом, а связанная с проблемой система идентифицируется или создается итеративно в процессе реализации ПСА и включает в себя те элементы, внутренние и внешние связи, которые необходимы для достижения целей субъекта [14].

ПСА является универсальным методом решения проблем произвольной природы и междисциплинарным подходом, опирающимся на общесистемное мышление и различные методы постепенного перехода от неструктурированных и слабоструктурированных моделей к частично или полностью формализованным моделям проблемосодержащей и проблеморазрешающей систем, включая методы математического и статистического анализа, исследования операций, методы машинного обучения и искусственного интеллекта и др.

Существует множество подходов к определению содержания и последовательности этапов ПСА, однако во всех случаях процедура ПСА приводит к идентификации или генерированию альтернативных вариантов решения проблемы, их оценке и выбору лучшей из альтернатив [18]. Формально данная задача может быть сведена к математической задаче глобальной оптимизации, где независимые переменные задачи определяют альтернативы, а целевая функция представляет собой критерий качества выбора альтернативы. Более того, задачи, которые сводятся к оптимизационным, возникают практически на всех этапах ПСА: при решении задач анализа, построения моделей, управления

(целенаправленного улучшающего воздействия на проблемную ситуацию) и др.

Такие задачи могут:

– иметь первоначально оптимизационные постановки:

- задачи математического программирования и исследования операций,
- задачи анализа данных,
- задачи параметрического проектирования,
- задачи поиска эффективных альтернатив и др.

– быть обеспечивающими задачами:

- структурная и параметрическая оптимизация моделей,
- конфигурирование и настройка применяемых методов и алгоритмов,
- оптимальное управление и др.

В настоящее время, в следствии развития методов анализа, проектирования и управления сложными системами и объектами, а также в следствии развития вычислительной техники и информационных систем, специалисты в области ПСА сталкиваются со следующими трудностями:

– требуется реагировать на большее число внешних и внутренних факторов исследуемой системы, анализировать большие объемы данных и информации. При этом, декомпозиция системы затруднена, т.к. синтетические свойства сложной системы (эмерджентные свойства), не могут быть объяснены или выражены через свойства ее отдельных частей;

– требуется исследовать более сложные статические свойства системы, ее подсистем и элементов, повышать точность моделей системы и методов управления и поддержки принятия решений;

– требуется исследовать более сложные динамические свойства системы, ее подсистем и элементов, а, следовательно, повышать точность нестационарных моделей функционирования системы;

– требуется проектирование не единственного, а множества эффективных вариантов решения проблемы, т.к. возможны различные сценарии развития

проблемной ситуации, необходимо учитывать мнения различных групп (лица принимающего решения (ЛПР) и стейкхолдеров). И др.

Проблемы и связанные с ними системы, обладающие вышеперечисленными и другими трудностями, называют сложными. В свою очередь, свойства сложных систем отражаются на свойствах возникающих в ПСА задачах оптимизации, которые можно отнести к классу сложных задач оптимизации.

Рассмотрим свойства сложных систем и соответствующее их отражение на задачах глобальной оптимизации, которые являются предметом данного диссертационного исследования:

1. *Свойство сложной системы: отсутствие или неполнота информации об исследуемой системе.* Данное свойство характерно для всех сложных систем. Растрин Л.А. называл данное свойство главной проблемой управления сложными системами [19]. В следствии недостатка информации о сложной системе, аналитик не способен построить адекватные модель композиции и модель состава систем, а также описать функции подсистем и элементов. Единственный способ работы с системой в таком случае – использование моделей типа «черный ящик».

С точки зрения возникающих задач оптимизации, отсутствие или неполнота информации приводит к тому, что целевая функция задается алгоритмически, вычисляется в ходе экспериментов или компьютерных симуляций, или как экспертная оценка. Поскольку модель «черного ящика» перечисляет все входы и выходы системы, и декомпозиция затруднена, в задачу оптимизации могут быть включены переменные, выраженные в разных шкалах, включая слабые (бинарные, наименования и др.). Из-за алгоритмического задания целевой функции у исследователя отсутствует информация о свойствах задачи оптимизации, таких как линейность, гладкость, дифференцируемость, выпуклость и прочих свойствах, которые можно использовать для выбора эффективного алгоритма оптимизации.

Известны примеры, когда целевая функция может быть задана строго аналитически, но математический анализ не позволяет определить глобальный оптимум аналитически или выявить свойства, необходимые для выбора

подходящего алгоритма оптимизации. В таком случае целевая функция также рассматривается как алгоритмически заданная.

Данный класс задач глобальной параметрической оптимизации называют задачей глобальной оптимизации с алгоритмически заданной целевой функцией или, как это встречается в англоязычных работах, оптимизацией «черного ящика», *Black-box Optimization* (BB) [20]. В дальнейшем, будем считать, что все классы задач оптимизации, исследуемые в данной диссертации, не зависимо от сочетания других свойств, являются задачами оптимизации с алгоритмически заданной целевой функцией.

2. *Свойство сложной системы: сложность обусловленная числом факторов и связей.* Сложность анализа и управления большими системами зачастую связана с ограниченностью ресурсов (временных, вычислительных, материальных и др.), необходимых для построения моделей, проектирования и реализации улучшающего вмешательства в проблемную ситуацию, даже если имеется достаточно информации для проведения ПСА. Ситуация существенно усложняется для случая «черного ящика».

Увеличение числа факторов и связей приводит к увеличению числа переменных (факторов), которые включаются в модель системы, а, следовательно, увеличивается размерность задач оптимизации. Увеличение количества переменных целевой функции экспоненциально увеличивает пространство поиска (так называемое «проклятие размерности» или эффект «комбинаторного взрыва» для задач с дискретными переменными) [21]. Для частных случаев, например, задачи линейного программирования, большая размерность не является проблемой. Как показывает практика, в общем случае задачи глобальной оптимизации, возникающие в ПСА, являются нелинейными, многоэкстремальными и не обладают свойствами, полезными для выбора эффективного алгоритма оптимизации, либо анализ таких свойств сравним по сложности с решением задачи оптимизации «черного ящика» [22, 23].

Некоторые задачи оптимизации могут быть масштабируемыми, т.е. сохраняют свои свойства с ростом размерности (пример, линейная и квадратичная

функции). В общем случае в плохо- и немасштабируемых задачах с ростом размерности задачи меняется ландшафт целевой функции, увеличивается число и меняется взаимное расположение глобального и локальных оптимумов [24].

Процедура СА подразумевает выполнение декомпозиции большой системы (выделение и анализа частей системы). Для сложных систем, в ходе декомпозиции теряются синтетические (эмерджентные) свойства системы. Как следствие, при решении задач оптимизации целевые функции являются несепарабельными.

Класс задач оптимизации больших систем называют задачей глобальной оптимизации с алгоритмически заданной целевой функцией большой размерности, в англоязычных работах – *Large Scale Global Optimization (LSGO)* [25].

3. *Свойство сложной системы: нестационарность, связанная с внутренними и внешними изменениями.* Нестационарность является следствием таких динамических свойств сложных систем как функциональность, стимулируемость и изменчивость системы и внешней среды во времени [14]. Изменения в среде оказывают влияние на ландшафт целевой функции, в частности, изменяются положения и свойства глобального и локальных оптимумов, появляются новые и исчезают ранее существовавшие оптимумы. При этом задачи оптимизации не могут быть разбиты на серию стационарных задач, т.к. в общем случае моменты времени и характер изменений заранее неизвестны и могут быть случайными.

Класс задач оптимизации, связанный с данным свойством сложных систем, называется задачи глобальной оптимизации с алгоритмически заданной целевой функцией в нестационарной среде, в англоязычных работах – *Optimization in the Non-stationary Environment* или *Dynamic Optimization Problems (DOP)* [26].

4. *Свойство, связанное с необходимостью обеспечить потребителей результатов ПСА (ЛПР и стейкхолдеров) множеством альтернативных решений.* Данное требование преследует несколько целей. Во-первых, поиск альтернативных решений исходной проблемы является главной целью ПСА. Во-вторых, наличие у ЛПР множества разнообразных альтернатив повышает устойчивость решений, т.к. для сложной системы мы имеем недостаточно адекватные модели из-за недостатка

или отсутствия информации. В-третьих, имея дело с моделью типа «черный ящик», дополнительная информация об альтернативных способах решения проблемы, позволяет ЛПР, в случае необходимости, точнее сформулировать цели ПСА, а аналитику – перейти от модели «черного ящика» к модели «серого ящика» [27, 28].

Традиционная постановка задачи оптимизации связана с поиском единственного оптимального решения задачи. Требование обеспечить ЛПР и стейкхолдеров различными альтернативами ставит новую задачу – найти все оптимумы или некоторое репрезентативное подмножество глобальных и локальных решений.

Класс задач оптимизации, связанный с данным требованием реализации ПСА называется задачей аппроксимации множества оптимумов в задаче глобальной оптимизации с алгоритмически заданной целевой функцией, в англоязычных работах – *Multimodal Optimization* (ММО) [29].

Стоит отметить, что выделенные выше классы задач оптимизации стали предметом исследования сравнительно недавно и большая часть работ в данном направлении ведется зарубежными исследователями, поэтому терминология, связанная с обозначением классов задач, их свойств и применяемых подходов среди отечественных исследователей еще не устоялась.

1.2 Формализация задач оптимизации, возникающих в прикладном системном анализе

Необходимость решать задачи оптимизации является следствием естественной проблемы рационального выбора, т.е. выбора лучшего варианта из многих. На практике альтернативные варианты могут быть определены заранее или формироваться в процессе поиска решения, а множество альтернатив быть конечным или бесконечным. В случае конечного множества альтернатив возможно решение задачи путем попарного сравнения всех альтернатив (полный перебор, *Brute-force Search*). Полный перебор неприменим, если множество альтернатив

велико или если ограничен бюджет оценок качества альтернатив, т.к. оценки могут требовать определенных вычислительных, временных или материальных затрат. В этом случае, для решения требуется математическая модель объекта оптимизации (критерий оптимальности, целевая функция), а решение ищется аналитически или путем усечённого перебора. Наличие информации об объекте оптимизации и идентифицированные свойства определяют класс задачи и методы, применимые для ее решения [21, 30-33].

Поиск наибольших или наименьших значений (вместе называемых экстремумами) некоторой функции является предметом общего исследования функции в математическом анализе [34]. Ограничением применимости методов математического анализа на практике является то, что для идентификации экстремумов целевая функция должна быть задана аналитически и быть дифференцируемой (достаточное условие требует вычисление вторых производных).

Разработкой теории и численных методов решения задач оптимизации занимается раздел математики – математическое программирование [32, 35-41]. В математическом программировании представлены классификация задач оптимизации по наличию определенных свойств целевых функций и ограничений, а также классификация методов решения. Несмотря на то, что математическое программирование предоставляет большой арсенал методов для решения многих слабо- и хорошо структурированных задач, на сегодня задачи, определённые в разделе 1.1, не имеют достаточного эффективного решения в рамках теории математического программирования.

Еще одной математической дисциплиной, связанной с решением задач оптимизации, является исследование операций [30, 31]. Исследование операций большей частью направлено на решение задач, которые могут быть сведены к одной из стандартных моделей исследования операций: задача о ранце, задача коммивояжёра, транспортная задача, задача динамического программирования, задача составления расписания и др. Несмотря на то, что исследование операций направлено на решение слабо- и хорошо структурированных задач, в исследовании

операций, в том числе, часто применяются приближенные численные и эвристические решения. Однако, как и для математического программирования, задачи, определённые в разделе 1.1, не имеют решения в рамках теории исследования операций и не могут быть сведены к одной из стандартных моделей.

В данном диссертационном исследовании рассматривается следующая общая постановка задачи глобальной безусловной параметрической оптимизации:

$$f(\bar{x}) \rightarrow \min_{\bar{x} \in S}(\max), f: S \rightarrow R^1, \quad (1.1)$$

где $\bar{x} = (x_1, x_2, \dots, x_n)$ – независимые переменные задачи оптимизации, f – целевая функция (критерий качества решения задачи оптимизации), $S \subset R \times F \times N \times P$ – пространство поиска, R – область изменения вещественных переменных, F – область изменения дискретных переменных, для которых задано отношение порядка, N – область изменения переменных наименований, P – область изменения переменных перестановок.

В случае, когда переменные задачи (1.1) выражены в разных шкалах, исходные переменные сводятся к наиболее слабой из используемых шкал. Один из наиболее часто используемых подходов, в том числе и в эволюционных вычислениях – метод бинаризации [42], позволяющий сводить произвольные задачи оптимизации к задачам псевдобулевой оптимизации [43].

Решением задачи (1.1), в предположении существования экстремума, является:

$$\bar{x}^* = \arg \min_{\bar{x} \in S} f(\bar{x}) \left(\arg \max_{\bar{x} \in S} f(\bar{x}) \right). \quad (1.2)$$

В данном исследовании на вид целевой функции $f(\bar{x})$ не накладываются никакие ограничения типа линейности, непрерывности, выпуклости и необходимость быть определенной во всех точках, запрашиваемых алгоритмом поиска. В общем случае целевая функция задана алгоритмически, информация о свойствах ее ландшафта отсутствует, целевая функция задачи (1.1) представляет собой модель «черного ящика».

Задачи оптимизации (1.1), в которых число переменных n велико, и это существенно затрудняет процесс решения задачи оптимизации, называют задачами оптимизации большой размерности. Во многих работах, посвященных проблеме оптимизации «черного ящика» показано, что для задач глобальной оптимизации с вещественными переменными и алгоритмически заданными целевыми функциями большинство известных подходов перестают быть эффективными при $n \geq 100$. В настоящее время мировое научное сообщество, рассматривает задачи глобальной оптимизации с алгоритмически заданными целевыми функциями большой размерности при $n \geq 1000$.

В задачах оптимизации в нестационарной среде целевая функция может меняться во времени, что приводит к изменению ландшафта функции, а также количества, положений, областей притяжения и значений глобального и локальных оптимумов. При этом задача (1.1) – (1.2) формулируется в следующей форме:

$$f(\bar{x}, t) \rightarrow \min_{\bar{x} \in S(t)} (\max), f: S(t) \rightarrow R^1, \quad (1.3)$$

$$X^* = \left\{ \bar{x}^*(t) = \arg \min_{\bar{x} \in S(t)} f(\bar{x}, t) \left(\arg \max_{\bar{x} \in S(t)} f(\bar{x}, t) \right), \forall t \in [0, t_{max}] \right\}, \quad (1.4)$$

где t – параметр времени, t_{max} – период решения задачи нестационарной оптимизации.

Замечание: в задачах оптимизации в нестационарной среде параметр времени не рассматривается в реальном масштабе. В модельных тестовых задачах период между двумя соседними изменениями в среде определяются числом вычислений целевой функции.

Задача аппроксимации множества экстремумов использует постановку задачи (1.1), но вместо решения (1.2) ищется множество решений (1.5):

$$X^* = \{x_i^{glob}, x_j^{local}, i = \overline{1, Global}, j = \overline{1, Local}\}, \quad (1.5)$$

где x_i^{glob} – решения, являющиеся глобальными экстремумами, x_j^{local} – решения, являющиеся локальными оптимумами, $Global$ и $Local$ известные число глобальных и локальных оптимумов, соответственно.

Замечание: решение (1.5) – идеальное требование для тестовых задач, используемых при разработке и исследовании новых методов, при решении практических задач, число оптимумов не известно и требуется дополнительная информация, чтобы отличить локальный оптимум от глобального.

В настоящее время при решении задач оптимизации «черного ящика» (1.1) – (1.5) хорошо зарекомендовали себя различные алгоритмы прямого поиска (нулевого порядка), которые последовательно улучшают решение задачи, начиная с некоторой начальной точки пространства поиска, на основе опыта, полученного на предыдущих итерациях. Алгоритмы прямого поиска используют лишь значение целевой функции в точке и не требуют дополнительной информации о типе и свойствах целевой функции и ее производных. Среди методов прямого поиска имеются хорошоформализованные детерминированные и стохастические подходы. Однако для сложных практических задач чаще применяются различные эвристические и метаэвристические алгоритмы [21, 35, 44, 45], которые, несмотря на часто слабое формализованное обоснование, на практике позволяют в конечное время получить точное или приемлемое приближенное решение.

1.3 Эволюционные алгоритмы решения сложных задач оптимизации

Эволюционные алгоритмы оптимизации представляют собой один из наиболее популярных и эффективных при решении многих сложных задач оптимизации класс алгоритмов прямого поиска.

Первые работы в области ЭА традиционно связывают с работой Дж. Г. Холланда «Адаптация в естественных и искусственных системах» (*Adaptation in Natural and Artificial Systems*, 1975) [46], в которой он обобщил различные подходы и идеи в области ЭА и формализовал «репродуктивные планы», известные сегодня как простые генетические алгоритмы (ГА). Тем не менее различные идеи моделирования естественных биологических процессов эволюции для решения задач оптимизации, адаптации и машинного обучения

встречались и ранее у различных авторов: эволюционные процессы для задач информатики (Фрэзер, 1957 и Фридберг, 1958) [47-48], эволюционное программирование (Фогель, 1966) [49], генетические алгоритмы (Холланд, 1967) [50], эволюционные стратегии (Рехенберг, 1965 и Швифель, 1965) [51-52], случайный глобальный поиск с адаптацией (Растрингин Л.А., 1965) [53].

Поскольку ЭА моделируют некоторые аспекты биологических процессов и используют принципы естественного отбора, в теории и практике ЭА используется специальная терминология. Введем обозначения, используемые в данной работе.

Решение (кандидат-решение) в ЭА называется *индивидом*. Каждый индивид представлен вектором переменных задачи оптимизации $\bar{x} = (x_1, x_2, \dots, x_n)$, называемым *хромосомой*, в которой каждый элемент называется *геном*, и оценкой *пригодности* (*фитнес-функция*, *fitness*), отражающей качество решения поставленной задачи. Множество решений, обрабатываемых параллельно на одной итерации алгоритма называется *популяцией*.

Для задач оптимизации с вещественными и целочисленными переменными хромосома обычно содержит в генах значения этих переменных. В ГА исходное представление решения задачи (*фенотип*) кодируется в хромосому (*генотип*). Генотип может быть представлен любым алфавитом, но наиболее распространенным является бинарное кодирование. Бинарное кодирование, являясь одной из слабейших шкал, может быть использовано для задач со смешанными переменными. Более того, во многих случаях бинарное кодирование повышает эффективность решения задач с вещественными переменными, т.к. в бинарном пространстве поиска меняется система окрестностей. В работах Барбулеску, Уитли и Рана было показано, что в бинаризованном пространстве поиска с ростом числа бинарных переменных число локальных оптимумов исходной вещественной задачи обычно уменьшается, а в частных случаях исходная многоэкстремальная задача в бинарном пространстве становится одноэкстремальной [54, 55].

Хромосома индивида может содержать не только переменные задачи, но и другие параметры, необходимые для управления работой ЭА (например,

коэффициенты мутации в ЭС). Хромосома может быть спроектирована с учетом особенностей решаемой задачи (например, ГА с изменяющимся локальным поиском, в котором старшие биты отвечают за большие изменения исходной переменной [56, 57]).

Функция пригодности используется для реализации основных операторов поиска в ЭА. На основе пригодности происходит вероятностный выбор решений (*оператор селекции*), информация которых используется для построения новых точек, исследуемых алгоритмом. Отобранные индивиды называются *родителями*, их хромосомы случайно перемешиваются (*оператор рекомбинации, скрещивание, кроссинговер*), формируя хромосому *решения-потомка*, к которому применяются случайные изменения (*оператор мутации*).

Функция пригодности должна отражать качество решения исходной задачи оптимизации, чтобы концентрировать поиск в перспективных регионах пространства поиска. Для задачи (1.1) пригодность может быть представлена значением целевой функции. Функция пригодности может также быть построена с учетом особенностей класса задачи оптимизации (например, штраф за нарушение ограничений, учет многокритериальности) и содержать элементы управления работой ЭА (например, пересчет пригодности с учетом попадания индивидов в одну область (нишу) для поддержки разнообразия).

Общая схема ЭА, так называемый «эволюционный цикл» состоит из следующих шагов:

Алгоритм 1. Общая схема ЭА.

Вход: задача оптимизации, параметры ЭА.

Инициализация: формирование случайной популяции индивидов.

Повторять, пока не выполнится условие останова:

 Оценить пригодность популяции.

Повторять, пока не сформирована новая популяция:

 Селекция: Выбрать индивидов-родителей.

 Рекомбинация: Сформировать хромосому потомка.

 Мутация: Случайные изменения в хромосоме потомка.

 Замещение предыдущей популяции новой.

Выход: лучшее найденное решение задачи.

ЭА относят к классу метаэвристик (метаэвристических алгоритмов), которые используют общие принципы построения конкретного поискового алгоритма на основе одной или множества базовых эвристик [44]. Дадим определения понятиям «эвристика» и «метаэвристика», используемым в данной работе.

В советской энциклопедии дается следующее определение: «Эвристика (от греч. – отыскиваю, открываю), спец. методы решения задач (эвристич. методы), которые обычно противопоставляются формальным методам решения, опирающимся на точные математич. модели. Использование эвристич. методов (эвристик) сокращает время решения задачи по сравнению с методом полного ненаправленного перебора возможных альтернатив» [58]. Эвристики могут отражать некоторый опыт решения задачи, быть аналогией другого способа решения и т.д. В области математической оптимизации под *эвристикой* будем понимать способ (идею, алгоритм) решения частной задачи оптимизации, который обеспечивает более быстрое решение задачи, чем традиционный формализованный метод или позволяет найти приемлемое приближенное решение, в случаях, когда традиционный формализованный метод не может получить точное решение задачи [35, 44].

Для решения некоторого множества задач, могут быть сформированы более сложные подходы, оперирующие в разных комбинациях более простыми эвристикami. Такие подходы называют *метаэвристикami*, термин был введен Ф. Гловером в 1986 [59]. За счет управления базовыми эвристикami в процессе решения задачи, метаэвристики более универсальны и могут применяться для целого класса или классов задач оптимизации. Включение в состав новых эвристик позволяет легко адаптировать известные метаэвристики к новым задачам оптимизации.

ЭА являются метаэвристическим алгоритмом и используют множество различных эвристик, которые обеспечивают в процессе решения задач оптимизации глобальную сходимость, устойчивость решений при различных начальных условиях, ускорение процесса нахождения решения, поддержку разнообразия решений и другие преимущества [60-65]. Среди основных эвристик:

- параллельная обработка множества решений, называемых популяцией, на каждой итерации поиска;
- использование случайных механизмов для формирования новых решений в операциях рекомбинации и мутации;
- использование операторов, основанных на принципах биологической эволюции, принцип естественного отбора реализованный в операции селекции. И др.

В большинстве исследований и практических приложений ЭА для реализации основных операций используются известные базовые эвристики. Например, селекция пропорционально значению пригодности или рангов индивидов, турнирная селекция, элитизм, точечное скрещивание, равномерное скрещивание, равномерное скрещивание по всей популяции, точечная мутация, мутация инверсией и другие. Тем не менее, возможно создание новых базовых эвристик с учетом особенностей задачи и конкретной реализации ЭА.

Несмотря на то, что существует традиционная последовательность применения базовых эвристик ЭА (Алгоритм 1), изменение порядка, состава и содержания этапов цикла формирует новые метаэвристики. Например, стационарный ГА (*Steady-state GA*) не использует идею популяции (в одном поколении формируется одно новое решение) [66], в эволюционных стратегиях (ЭС) главным оператором формирования новых решений на основе прошлого опыта является мутация [52], а в ГА – рекомбинация [46], в алгоритмах оценки распределений (*Estimation of Distribution Algorithms*, EDA) популяция замещается оценкой закона распределения генов [67, 68].

Класс ЭА включает десятки подклассов и сотни конкретных алгоритмов. Классификация ЭА и отнесение известных и новых алгоритмов к ЭА является предметом дискуссий [69-72]. В настоящее время к ЭА относят множество подходов, основанных на моделировании эволюционных, биологических, физических, химических и прочих принципов, которые все вместе называют «вдохновленные природой и биологией» (*nature- and biology-inspired*) [63, 76].

1.4 Обзор современных подходов к проектированию и настройке эволюционных алгоритмов и формализация задачи синтеза эволюционных алгоритмов

ЭА являются универсальным подходом к решению сложных задач оптимизации, однако эффективность их применения на практике существенно зависит от выбора конкретной конфигурации операторов и настройки их параметров. Несмотря на использование в литературе термина «настройка», каждая конфигурация формирует новую метаэвристику (новый ЭА), которая отличается не только итоговой эффективностью решения задачи оптимизации, но и реализацией процесса получения решения [44, 45, 62, 63, 73-75, 77].

Традиционным способом выбора и настройки конфигурации является многократное решение поставленной задачи или множества задач, оценка результатов и выбор эффективного варианта реализации ЭА. Однако даже в рамках традиционного эволюционного цикла различные комбинации типов операторов селекции, рекомбинации и мутации формируют десятки различных ЭА. Более того, базовые эвристики содержат целочисленные и вещественные параметры, которые также требуют тонкой настройки. Поэтому настройка ЭА является самостоятельной задачей, которая может быть сведена к оптимизационной постановке.

Одна из первых работ по структурной оптимизации алгоритмов случайного поиска принадлежит Растригину Л.А. [78], который предложил следующую постановку. Представим алгоритм оптимизации F в виде двойки:

$$F = \langle S, C \rangle, \quad (1.6)$$

где S – структура алгоритма F , $C = (c_1, \dots, c_s)$ – параметры алгоритма.

Сложившуюся в процессе оптимизации ситуацию обозначим как L :

$$L = \langle Q(X), \Omega \rangle, \quad (1.7)$$

где $Q(X)$ – целевая функция с известными свойствами, $X \in \Omega$ – область определения. Даже если Q и Ω заданы алгоритмически, подразумевается возможность идентификации их свойств в процессе применения алгоритма F .

Тогда задача выбора оптимального алгоритма сводится к задачам оптимизации (1.8) или (1.9):

$$K(F, L) \rightarrow \max_{F \in E}, \quad (1.8)$$

$$K(F, L) \rightarrow \max_{F \in E_S} \max_{F \in E_C}, \quad (1.9)$$

где K – критерий оценки эффективности алгоритма F , $E = E_S \times E_C$ – множество допустимых алгоритмов. В случае постановки (1.9) производится декомпозиция задачи на два уровня: структурная и параметрическая оптимизация.

В области ЭА задачу проектирования и настройки алгоритма чаще рассматривают не как самостоятельную задачу оптимизации, а как часть процесса решения исходной задачи. В мировой практике закрепилась следующая терминология: управление параметрами и настройка параметров (*parameter control, parameter tuning*), само-адаптация (*self-adaptation*) и само-конфигурация (*self-configuration*) [83]. Существуют различные классификации подходов [63, 84, 85], однако наиболее популярной и цитируемой является классификация Эйбена и Михалевича [86, 87]:

– Настройка параметров эвристик ЭА, которые остаются неизменными в процессе решения задачи оптимизации:

- вручную на основе опыта и интуиции исследователя;
- планирование эксперимента и экспериментальное исследование ЭА;
- мета-эволюция – решение двухуровневой задачи оптимизации, аналогичной (1.9).

– Управление параметрами эвристик ЭА в процессе решения задачи оптимизации:

- детерминированное правило, не использующее обратную связь от процесса решения задачи;

- адаптивная схема изменения параметров на основе обратной связи;
- само-адаптация за счет включения параметров ЭА в хромосому.

В свою очередь, адаптация может применяться:

- на уровне внешней среды, когда происходит перенастройка способа оценки пригодности индивидов;
- на уровне популяции, когда параметры ЭА являются глобальными и применяются ко всем индивидам одинаково;
- на уровне индивида;
- на уровне отдельных компонент хромосомы или генов.

В современных исследованиях к управлению параметрами добавляется структурная оптимизация ЭА [79-82], связанная с выбором эвристик из предопределенного множества базовых эвристик, которая по аналогии с выше перечисленной классификацией может применяться однократно, используя внешний детерминированный или адаптивный алгоритм, а также за счет само-конфигурации [63]. В качестве альтернативных конфигураций может, в том числе, рассматриваться конечное множество различных настроек эволюционных эвристик (например, выбор сильной, средней или слабой мутации).

К недостаткам известных подходов для конфигурации и настройки ЭА можно отнести то, что в них осуществляется настройка одного ЭА, реализующего единственную стратегию поиска оптимального решения. Алгоритмы настройки, адаптации или конфигурации фактически являются эволюционными метаэвристиками, которых предложено много, а выбор одной из них для задач типа «черный ящик» не очевиден.

В теории оптимизации существует теорема «бесплатных завтраков не бывает», предложенная Волпертом Д. и Макреди В. в 1997 (No Free Lunch (NFL) Theorems for Optimization) [88], согласно которой эффективность любых двух алгоритмов оптимизации усредненная на множестве всех задач будет одинаковой, а, следовательно, для любой задачи найдется наиболее эффективный узкоспециализированный алгоритм, превосходящий универсальный подход. Тем не менее, очевидна гипотеза, что ограничения данной теоремы можно преодолеть,

если при решении сложных задач вместо единственного, даже после тонкой настройки, алгоритма, выбирать алгоритм из множества или формировать новый алгоритм под конкретную задачу.

В области машинного обучения и систем искусственного интеллекта развивается направление автоматизированного и автоматического проектирования алгоритмов, в рамках которого задача проектирования алгоритма формулируется как самостоятельная задача машинного обучения [89-91]. Аналогичное направление в области решения задач оптимизации получило название «*гиперэвристики*». Термин гиперэвристика был впервые предложен в 1997 в работе [92] для описания подхода, объединяющего несколько методов искусственного интеллекта в задачах автоматизированного доказательства теорем. Позже термин стал употребляться в контексте решения сложных комбинаторных задач [93, 94], как эвристика для выбора других эвристик.

В настоящее время, под гиперэвристикой подразумевается метод автоматического (без или с минимальным участием человека) выбора или создания эвристик и метаэвристик для решения задач оптимизации [95].

Необходимость отделения гиперэвристик от метаэвристик связана с различиями пространств поиска, в которых применяются подходы. Эвристические и метаэвристические алгоритмы при решении задач оптимизации осуществляют поиск в пространстве решений исходной задачи, а результатом их применения является лучшее найденное решение задачи. Гиперэвристика осуществляет поиск в пространстве эвристик и метаэвристик, результатом является алгоритм решения задачи оптимизации и лучшее найденное им решение. Гиперэвристика обычно не использует информацию предметной области решаемой задачи, в то время как эвристики построены на основе некоторых знаний и опыта предметной области. Последнее называют «барьером предметной области» [96], а модель гиперэвристики представляют как двухуровневую (рисунок 1.1).

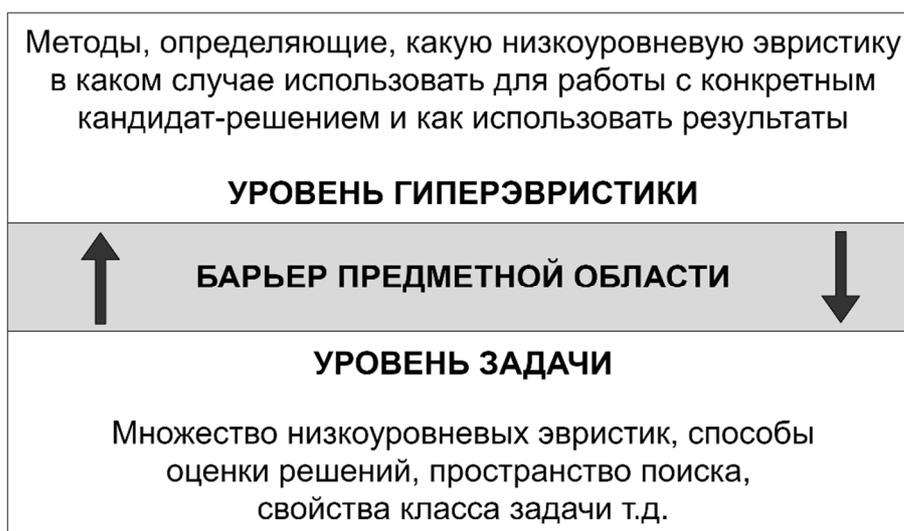


Рисунок 1.1 – Двухуровневая модель гиперэвристики

Наличие барьера предметной области позволяет использовать гиперэвристики для формирования алгоритмов оптимизации, решающих целый спектр (класс) задач с минимальным участием или без участия человека, т.к. знания и опыт решения конкретного экземпляра класса заложены на уровне задачи. Тем не менее, предложены различные способы реализации гиперэвристик, в том числе адаптированные с учетом особенностей класса задач оптимизации. Наиболее полная классификация гиперэвристик была предложена Бёрком Э.К. в 2010 и дополнена в 2018 [95]. Согласно классификации Бёрка все известные гиперэвристики можно классифицировать по двум признакам: природа пространства поиска и наличие обратной связи (рисунок 1.2).

Классификация по природе пространства поиска выделяет:

– *методы выбора гиперэвристик*, которые оперируют известными комбинациями базовых эвристик (фрагментами или целыми метаэвристиками) или формируют новые комбинации из некоторых predetermined конструкций базовых эвристик;

– *методы проектирования эвристик* используются для формирования полностью новых эвристик из допустимых операций, применяемых в предметной области для поиска решения задачи или осуществляют поиск комбинаций элементарных эвристик и их фрагментов.

Методы выбора и проектирования эвристик в свою очередь могут быть классифицированы по наличию обратной связи:

- *гиперэвристики без обучения* не используют информацию об эффективности применения базовых эвристик на прошлых этапах процесса поиска;
- *офлайн гиперэвристики*, по аналогии с методами машинного обучения, формируют алгоритм поиска путем решения множества примеров задач оптимизации с целью обобщить опыт на новые, ранее неисследованные примеры;
- *онлайн гиперэвристики* осуществляют выбор или проектирование эвристик в процессе решения конкретной задачи оптимизации.

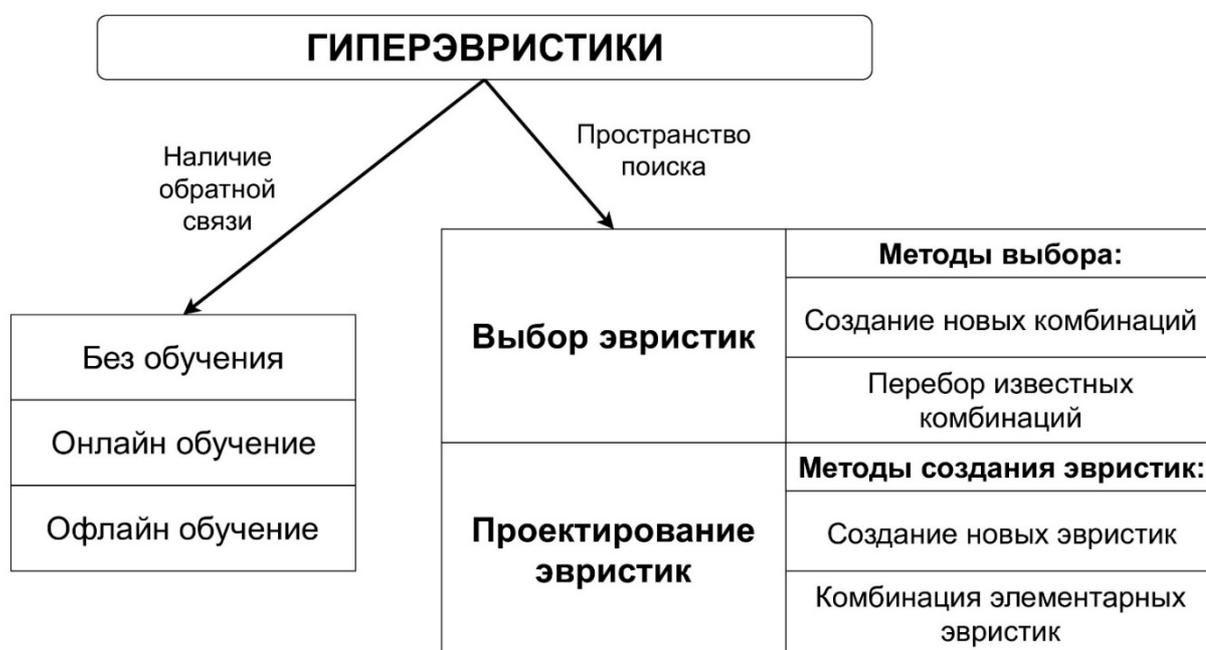


Рисунок 1.2 – Классификация гиперэвристик по Бёрку

На текущий момент проблема разработки, исследования и применения гиперэвристик разработана достаточно слабо. В работах [94, 96-100] представлены обзор современного состояния исследований, существующие гиперэвристики и примеры применения гиперэвристик при решении прикладных задач. В работе [101] профессором Мисир М. собрана база публикаций по теме гиперэвристик на основе международных баз научного цитирования Scopus, Web of Science, IEEE Xplore и DBPL. С 1999 по 2019 год было представлено всего 850 работ. Согласно

данных РИНЦ по состоянию на 2019 год, представлено всего 5 работ, в которых используется терминология и упоминаются подходы на основе гиперэвристик. Обзор работ показал, что подавляющее большинство работ посвящено решению задач комбинаторной оптимизации, таких как задача составления расписания, поиск кратчайшего пути и др. Гиперэвристики для решения задач оптимизации сложных систем (1.1) – (1.5), формализованных в разделе 1.2, практически не исследованы.

Формализуем задачу синтеза ЭА для решения сложных задач оптимизации. Для этого опишем структуру ЭА для решения задачи (задач) выбранного класса следующим образом:

$$EA = \{selection, \{p_{selection}\}, recombination, \{p_{recombination}\}, \\ mutation, \{p_{mutation}\}, \{heuristic\}, \{p_{heuristic}\}, \{control\}\}, \quad (1.10)$$

где $selection \in Selection$ и $\{p_{selection}\}$ – выбранная эвристика управления операцией селекции и множество ее параметров, $recombination \in Recombination$ и $\{p_{recombination}\}$ – выбранная эвристика управления операцией рекомбинации и множество ее параметров, $mutation \in Mutation$ и $\{p_{mutation}\}$ – выбранная эвристика управления операцией мутации и множество ее параметров, $\{heuristic\} \subset Heuristic$ и $\{p_{heuristic}\}$ – множество эвристик, учитывающих особенности класса задач и их параметры, $\{control\} \in Control$ – множество эвристик управления алгоритмом (способ инициализации и оценки пригодности индивидов, стратегия замещения при формировании новой популяции, критерии останова и др.).

Задача синтеза ЭА оптимизации может быть представлена как задача комбинаторной оптимизации более высокого уровня, в которой пространство поиска – это множество метаэвристик, описанных как (1.10), а целевая функция отражает оценку эффективности решения конкретной задачи или задач выбранного класса:

$$\psi(EA) \rightarrow \min_{EA \in HS}(\max), \psi: HS \rightarrow R^1, \quad (1.11)$$

где $HS \subset Selection \times Recombination \times Mutation \times Heuristic \times Control$ – пространство поиска метаэвристик.

Методы решения задачи оптимизации (1.11) будут называть гиперэвристиками. Решение задачи (1.11) одновременно с задачами (1.1), (1.3) или задачей (1.1), (1.5) требует разработки онлайн гиперэвристик, т.к. синтез метаэвристики должен осуществляться адаптивно под конкретную задачу оптимизации в процессе ее решения. В свою очередь наличие репрезентативного множества примеров некоторого класса задач оптимизации и необходимость синтеза алгоритма, обеспечивающего высокую эффективность в среднем, требует разработки офлайн гиперэвристик.

Очевидно, что для каждого класса задач оптимизации требуется формализовать свой критерий ψ и пространство поиска HS . Более того, практические задачи могут накладывать на ψ дополнительные ограничения, например, связанные с программно-аппаратной реализацией синтезированных алгоритмов. В более сложной постановке элементы множеств *Selection*, *Recombination*, *Mutation*, *Heuristic* и *Control* заранее не определены, поэтому синтез базовых эвристик является самостоятельной задачей поисковой оптимизации, для которой необходимо определить пространство поиска, включающее элементарные операции построения эвристик, а также критерий эффективности, который в общем случае совпадает с критерием ψ .

1.5 Обзор современных подходов к оценке и сравнению эффективности эволюционных алгоритмов

Многие эффективные метаэвристики, включая ЭА, имеют достаточно слабую теоретическую базу. На сегодня, несмотря на различные попытки формализации (гипотеза о строительных блоках и теорема схем, моделирование с

помощью Марковских процессов, моделирование с помощью оценки распределений и др.) [46, 63, 67, 102-106], на сегодня для ЭА доказана лишь сходимость по вероятности. Поэтому исследователи осуществляют оценку и сравнение эффективности ЭА путем проведения численных экспериментов. Экспериментальное исследование может осуществляться на различных искусственно созданных (тестовых) задачах и практических задачах. Использование искусственных тестовых задач имеет ряд преимуществ:

- большее разнообразие свойств и комбинаций свойств, сложных для алгоритма оптимизации (включая специальные классы сложных для ЭА (*EA-hard problems*) задач);

- большая гибкость в модификации и масштабировании задач;

- аналитическое или алгоритмическое представление обеспечивает воспроизводимость экспериментов для сравнения разных алгоритмов оптимизации, а также обычно требует меньших вычислительных затрат;

- широкое обсуждение и применимость в научной среде позволяет сопоставлять результаты исследования новых подходов с мировым уровнем.

Очевидно, что для адекватных выводов об эффективности ЭА оптимизации множество тестовых задач для конкретного класса задач должно быть достаточно представительным, для этого необходимо чтобы:

- задачи в тестовом множестве рассматривались как модели «черного ящика» и были непохожи друг на друга;

- тестовые задачи должны отражать особенности прикладных задач оптимизации;

- тестовые задачи должны обладать такими свойствами, которые обычно вызывают затруднения у алгоритмов оптимизации: большая размерность, нелинейность, несепарабельность, овражность в областях притяжения глобального и локальных оптимумов, области постоянных значений целевой функции и др.

- тестовые задачи должны вызывать затруднения у алгоритмов локального поиска;

– тестовые задачи должны обладать свойством «обманчивости» («десептивности»);

– тестовые задачи должны быть масштабируемыми.

Основная идея использования тестовых задач связана с тем, что алгоритм, успешно справляющийся со всеми трудностями тестовых задач, сможет эффективно решать прикладные задачи, которые редко обладают всеми свойствами задач оптимизации, искусственно воспроизведенными в тестовых задачах. На сегодня предложено большое количество различных тестовых задач, на основе которых предложены различные наборы [107-116], включая наборы, специально созданные для ЭА [117-120]. Рядом исследователей предложены генераторы задач [121-126]. К сожалению, не все известные тестовые задачи глобальной оптимизации являются сложными для ЭА (например, функция Растригина в исходной постановке), не для всех задач представлен анализ свойств и рекомендации по проведению экспериментов, нет оценок эффективности для сравнения с известными подходами.

В настоящее время в области ЭА устоялись наборы тестовых задач, которые называются эталонными или бенчмарками (*benchmark test problems*). Данные эталонные задачи:

– отражают различные особенности задач оптимизации для различных классов, что затрудняет разработку и выбор ЭА, эффективного на всем множестве задач;

– были предложены на ведущих научных конференциях в области ЭА, многократно обсуждались, корректировались, усложнялись;

– в открытом доступе представлены исходные коды для каждого набора задач и требования к проведению экспериментов и анализу результатов, что обеспечивает честное сравнение эффективности алгоритмов разными исследователями;

– являются стандартом (эталоном) в области ЭА, поэтому исследователи используют именно их, а, следовательно, существует возможность сопоставить результаты оценки новых ЭА с мировым уровнем.

В данной работе были выбраны наборы эталонных тестовых задач, которые разрабатывались, обсуждались и позже использовались при проведении соревнований по оптимизации в рамках ведущих конференций по эволюционной оптимизации: Конгресса по эволюционным вычислениям (*Congress on Evolutionary Computation*, CEC) при поддержке международного сообщества вычислительного интеллекта IEEE и Конференции по генетическим и эволюционным вычислениям (*The Genetic and Evolutionary Computation Conference*, GECCO) при поддержке международной ассоциации вычислительной техники (*Association for Computing Machinery*, ACM). Данные конференции имеют наивысший ранг А по метрикам ERA и Qualis [127].

В диссертации использованы следующие наборы эталонных тестовых задач:

– для задач глобальной оптимизации с алгоритмически заданной целевой функцией (1.1) – (1.2): *IEEE CEC Black-box optimization benchmark* (ВВОВ) [119];

– для задач глобальной оптимизации с алгоритмически заданной целевой функцией (1.1) – (1.2) большой размерности: *IEEE CEC Large-Scale Global Optimization Benchmark* (LSGO) [128, 129];

– для задач глобальной оптимизации с алгоритмически заданной целевой функцией в нестационарной среде (1.3) – (1.4): *The Moving Peaks Benchmark* (MPB) и *Benchmark Generator for CEC Competition on Dynamic Optimization* (DOP) [131];

– для задач аппроксимации множества оптимумов в задаче глобальной оптимизации с алгоритмически заданной целевой функцией (1.1) – (1.5): *Benchmark Functions for CEC Special Session and Competition on Niching Methods for Multimodal Function Optimization* (ММО) [132].

Следующим важным вопросом при оценке эффективности ЭА являются условия проведения и оценки результатов численных экспериментов. Поскольку ЭА используют случайные механизмы при создании начальной популяции и формировании новых решений, некорректно делать выводы об эффективности алгоритма исходя из результатов одного запуска алгоритма. В настоящее время общепризнанной практикой является многократное независимое решение задач

(прогоны алгоритма), накопление данных о процессе и результатах решения задачи с последующей статистической обработкой и анализом результатов.

В случае эталонных тестовых задач для сравнения с известными подходами условия и способ оценки результатов экспериментов заданы. Для других тестовых и практических задач, а также при использовании эталонных задач на этапе проектирования новых ЭА, в данной работе используются следующие общие условия экспериментов:

– для одного независимого запуска ЭА задается ограничение на максимальное число (бюджет) вычислений целевой функции. ЭА может расходовать бюджет вычислений любым образом, выбор размера популяции и числа итерация является одним из вариантов реализации ЭА. Вычисления целевой функции являются наиболее затратным шагом реализации оптимизационного алгоритма с точки зрения вычислительных, временных, а иногда и материальных затрат (экспериментальное исследование качества решений), поэтому именно по отношению к этому ресурсу различные ЭА или реализации (настройки) ЭА должны находиться в одинаковых условиях для объективного сравнения результатов. В данном исследовании время работы алгоритма не принимается во внимание, т.к. эта характеристика алгоритма зависит от конкретной программной реализации (язык и технология программирования, операционная система), производительности оборудования и других внешних факторов, не связанных со свойствами алгоритма решения оптимизационной задачи;

– каждый исследуемый ЭА решает задачу оптимизации заданное количество раз (прогоны алгоритма), каждый запуск является независимым. В ходе отдельного прогона собирается следующая информация: лучшее найденное решение и оценка количества вычислений (номер итерации), когда это решение было найдено. Для некоторых классов задач оптимизации могут задаваться дополнительные оценки;

– для оценки эффективности ЭА вычисляются оценки среднего значения, стандартного отклонения, лучшее, худшее и медиана лучших найденных решений в каждом прогоне. Для некоторых классов задач оптимизации могут задаваться дополнительные оценки;

– в рамках заданного бюджета вычислений лучшим в попарном сравнении является ЭА, обеспечивающий большую точность нахождения глобального оптимума. В случае одинаковой точности, лучшим можно признать алгоритм, использовавший меньшее число вычислений целевой функции. При попарном сравнении ЭА осуществляется проверка статистической значимости разницы в оценках, иначе эффективность ЭА признается сравнимой;

– в случае сравнения трех и более ЭА после попарного сравнения алгоритмы ранжируются (лучший алгоритм имеет меньший ранг).

Одним из недостатков попарного сравнения ЭА при решении нескольких тестовых задач является сильное смещение оценок из-за «неудачных» прогонов ЭА. Это приводит к известной проблеме множественного сравнения, сказывающейся на групповой вероятности ошибки первого рода (*Family-Wise Error Rate*) [133]. В работе [133] было показано, что из-за проблемы множественного сравнения матрица рангов, составленная по результатам попарного сравнения, часто не обладает свойством транзитивности. В настоящем исследовании, данная проблема решается за счет применения при сравнении разных ЭА идей метода глубокого статистического сравнения (*Deep Statistical Comparison, DSC*), предложенного в 2018 году Томе Эфтимов [133, 134]. Автор предложил термин «глубокое статистическое сравнение», т.к. для ранжирования алгоритмов используются не итоговые оценки средних или медиан, а полное распределение исходных оценок алгоритма.

Схема ранжирования, используемая в данном исследовании следующая. Введем следующие обозначения: *AlgNum* – число ЭА в сравнении, *PrNum* – число тестовых задач, на которых производится оценка алгоритмов, *RunNum* – число независимых запусков каждого алгоритма на каждой задаче. Результаты численных экспериментов представим в виде матриц V_i размера $RunNum \times AlgNum$, $i = 1, \dots, PrNum$. Элемент i -й матрицы $v_{i,j,l}$, $j = 1, \dots, RunNum$, $l = 1, \dots, AlgNum$ содержит оценку эффективности решения i -й задачи l -м алгоритмом в j -м прогоне.

Для каждой тестовой задачи выполняются $AlgNum \cdot (AlgNum - 1)/2$ попарных сравнений алгоритмов и проводится анализ статистической значимости различий в результатах. Для анализа статистической значимости различий проверяется гипотеза о равенстве законов распределений двух выборок значений. Для этого используется непараметрический статистический тест Уилкоксона-Манна-Уитни [137] (можно использовать и другие тесты, например, Колмогорова-Смирнова) на заданном уровне значимости α (в данной работе $\alpha = 0.05$). Результаты применения теста формируют матрицу M_i размера $AlgNum \times AlgNum$, в которой элементы определяются как (1.12):

$$m_{i,p,q} = \begin{cases} p_{value}, & \text{если } p \neq q \\ 1, & \text{если } p = q \end{cases} \quad (1.12)$$

где p и q – алгоритмы в сравнении, $p, q = 1, \dots, AlgNum$.

Далее применяется поправка Бонферрони и формируется матрица M'_i размера $AlgNum \times AlgNum$ (1.13):

$$m'_{i,p,q} = \begin{cases} 1, & \text{если } m_{i,p,q} \geq \alpha/C_{AlgNum}^2 \\ 0, & \text{если } m_{i,p,q} < \alpha/C_{AlgNum}^2 \end{cases}, \quad (1.13)$$

где α – уровень значимости, при котором проводились попарные сравнения и вычислялись p_{value} , $C_{AlgNum}^2 = AlgNum \cdot (AlgNum - 1)/2$ – количество рассматриваемых гипотез в множественном сравнении.

Далее проверяется транзитивность матриц M'_i и применяется метод трансформации матрицы рангов DSC, предложенный Томе Эфтимов в [133]. В заключении, к итоговой матрице рангов применяется тест Фридмана (или тест с поправкой Имана-Давенпорт).

Полный алгоритм ранжирования ЭА с использованием метода глубокого статистического сравнения (Алгоритм 2):

Алгоритм 2. Алгоритм ранжирования ЭА.

Вход: задача оптимизации, множество ЭА для сравнения, результаты численных экспериментов для каждого ЭА в каждом прогоне для серии независимых запусков алгоритма.

Вычислить матрицу M_i , используя (1.12).

Вычислить матрицу M'_i , используя (1.13).

Проверить транзитивность матрицы M'_i и применить соответствующий способ ранжирования DSC.

Выход: средние ранги сравниваемых ЭА для задачи.

В данной работе для ранжирования и итоговых тестов используется реализация подхода глубокого статистического сравнения на языке программирования R, предоставленная авторами, а также web-реализация, представленная авторами в открытом доступе [137].

Выводы по главе 1

В Главе 1 проведен анализ современных проблем прикладного системного анализа и необходимость развития методов анализа, проектирования и управления системами и объектами, которые обладают рядом свойств, относящих их к классу сложных систем. В свою очередь, свойства сложных систем отражаются на свойствах задач глобальной параметрической оптимизации, возникающих на различных этапах прикладного системного анализа. Проведен анализ современных подходов к преодолению выявленных трудностей задач оптимизации и формализованы постановки задач глобальной оптимизации разных классов, исследуемые в диссертации, а именно:

– общая задача глобальной безусловной параметрической оптимизации с алгоритмически заданной целевой функцией, которая может иметь как вещественные переменные, так и переменные, выраженные в разных шкалах, включая слабые;

– задача глобальной безусловной параметрической оптимизации с алгоритмически заданной целевой функцией большой размерности;

– задача глобальной оптимизации с алгоритмически заданной целевой функцией в нестационарной среде;

– задача идентификации множества оптимумов в задаче глобальной оптимизации с алгоритмически заданной целевой функцией.

Показаны сложности отнесения подобных задач к известным классам задач и сложность применения традиционных математических и численных методов оптимизации, изучаемых в математическом программировании. Свойство сложенных систем, связанное с отсутствием или неполнотой информации об исследуемой системе, является общим для всех исследуемых в диссертации классов задач оптимизации, поэтому целевые функции рассматриваются как алгоритмически заданные, что обусловило выбор подхода к решению поставленных задач – эволюционных алгоритмов. Рассмотрены традиционные подходы из области эволюционной оптимизации и даны определения эвристического и метаэвристического подходов.

Обзор литературных источников показывает, что несмотря на популярность и множество предложенных универсальных ЭА, эффективность решения каждой конкретной задачи глобальной оптимизации на практике существенно зависит от выбранной структуры конкретного ЭА и тонкой настройки параметров ЭА. Существующие подходы, связанные с настройкой параметров ЭА, само-адаптацией и само-конфигурацией осуществляют настройку или адаптацию в процессе решения задачи конкретного типа ЭА из множества известных. В то же время, возможно создание произвольных структур ЭА из множества базовых операций (эвристик), фрагментов алгоритма или целых алгоритмов. Такая задача синтеза ЭА сформулирована в Главе 1 как задача комбинаторной оптимизации, а методы ее решения названы гиперэвристиками. Даны определение гиперэвристического подхода в оптимизации и классификация гиперэвристик по отношению к исследуемым классам задач глобальной оптимизации.

Поскольку ЭА являются эвристическими и аналитическая оценка их эффективности затруднена, оценка новых ЭА и сравнение их эффективности с известными подходами проводится на наборах эталонных тестовых задач. В главе 1 сформулированы требования к тестовым задачам оптимизации и обоснован выбор эталонных задач разных классов. Приводится общая схема оценки эффективности разрабатываемых подходов. Для ранжирования алгоритмов сформулирован подход на основе метода глубокого статистического сравнения,

позволяющий компенсировать ошибку множественного сравнения при проверке статистических гипотез для оценки статистической значимости различий в результатах оценки эффективности сравниваемых ЭА.

Задачи оптимизации, исследуемые в данной диссертации являются сложными и универсальных эффективных подходов для них не предложено, поэтому перспективным подходом является синтез ЭА под конкретную решаемую задачу. Направление применения гиперэвристик для задач оптимизации является достаточно новым, а для формализованных в диссертации классов сложных задач оптимизации – практически не исследованным, требуется разработка новых гиперэвристик для синтеза ЭА, позволяющих эффективно решать задачи выделенных классов.

ГЛАВА 2. ГИПЕРЭВРИСТИКА НА ОСНОВЕ МЕТОДА ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ СИНТЕЗА ОПЕРАТОРОВ СЕЛЕКЦИИ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

Глава 2 посвящена проблеме разработки гиперэвристик на основе алгоритма генетического программирования для формирования элементарных эвристик ЭА. Оператор селекции является одним из наиболее важных в ЭА, т.к. именно он обеспечивает сходимость алгоритма в целом, а модификации оператора селекции часто применяются для учета свойств класса задач оптимизации. В данной работе, оператор селекции рассмотрен как алгоритм назначения вероятностей, определяющих выбор индивидов популяции для последующих операций с ними. Предложенная в главе 2 конструктивная офлайн гиперэвристика формирует алгоритм назначения вероятностей в виде символьного выражения (формулы). Численные эксперименты показывают, что предложенный подход позволяет формировать новые эвристики, которые демонстрируют высокую эффективность в среднем на всем множестве исследуемых задач и превосходят среднюю эффективность базовых операторов селекции. Апробация подхода при решении практической задачи проектирования эвристики для ЭА в задаче идентификации партий электронных компонентов аэрокосмического применения позволила улучшить решения, полученные ранее.

2.1 Метод генетического программирования для формирования элементарных эвристик эволюционных алгоритмов

Метод генетического программирования (ГП) был предложен профессором Коза как способ автоматизированного проектирования компьютерных программ путем моделирования принципов естественного отбора природной эволюции [138]. В контексте ГП компьютерная программа – это алгоритм решения поставленной задачи, представленный иерархической структурой вызовов предопределенных

процедур (функций, действий) на основе входных данных задачи. Множество функций и входных данных для проектирования программы задается пользователем, алгоритм ГП формирует композиции функций (суперпозиции функций) и осуществляет поиск наилучшей композиции в смысле заданного критерия качества решения поставленной задачи [139, 140]. По сути, ГП решает комбинаторную задачу поисковой оптимизации, требующей значительных вычислительных затрат, поэтому наибольшее развитие метод ГП получил лишь начиная с 2000-х годов в связи с развитием технологии программирования и увеличением производительности вычислительной техники [141]. Способ представления компьютерных программ предложенный в ГП является универсальным и может применяться для представления кода на языке программирования для последующей компиляции или интерпретации, для описания логики алгоритма действий или принятия решений (например, управление роботом), для поиска функциональных зависимостей (символьное моделирование, дифференцирование, интегрирование и решение уравнений), а также для описания любых других сложных структур, являющихся комбинацией элементарных функциональных и терминальных компонентов [142]. Поскольку для оценки качества программы в ГП, она должна быть выполнена, а результат проанализирован, в ГП необходим контроль синтаксиса решений. Одним из вариантов реализации ГП, в котором контроль синтаксиса заложен на уровне представления решения является метод эволюции грамматик (*grammatical evolution*) [143].

Одна из наиболее популярных областей применения ГП в последние годы – автоматизированный синтез алгоритмов машинного обучения и алгоритмов оптимизации [101, 144-150]. Как показывает обзор работ в данной области, подавляющее большинство работ посвящено проектированию эвристик и метаэвристик для решения сложных задач комбинаторной оптимизации, включая задачи составления расписания, поиска кратчайшего пути, транспортную задачу и другие. Подходов для решения задач глобальной параметрической оптимизации,

рассматриваемых в данном диссертационном исследовании, практически не предложено.

Как было показано ранее, различные ЭА являются метаэвристиками, имеющими общую схему – эволюционный цикл. Независимо от типа ЭА, эволюционный цикл сохраняет логику и последовательность применения элементарных эвристик: преимущественный отбор более приспособленных индивидов (селекция), формирование новых решений путем наследования генов родителей (рекомбинация) и путем случайных изменений (мутация). Принципиальные различия связаны с выбором, настройкой и управлением элементарными эвристиками ЭА. Следовательно, при формировании новых ЭА перспективным является синтез новых элементарных эвристик, применяемых в рамках эволюционного цикла конкретного ЭА.

В работах [151-153] были предложены подходы на основе ЭА и ГП для синтеза операторов рекомбинации и мутации. Наибольший эффект данные подходы дают для задач комбинаторной оптимизации, где эвристики определяют нетривиальные способы формирования новых решений для различных дискретных структур (например, графы). Для задач глобальной параметрической оптимизации высокой эффективности можно достичь путем настройки и управления стандартными эвристиками рекомбинации и мутации за счет самоконфигурации и самоадаптации ЭА [83, 86].

Теоретические исследования [102] и опыт практического применения ЭА показали, что оператор селекции является наиболее важным механизмом ЭА и ГА [154], т.к. именно за счет селекции реализуется эвристика, моделирующая естественный отбор – оператор селекции обеспечивает увеличение средней пригодности популяции за счет повышения вероятности более приспособленных индивидов передать свои гены потомкам или быть скопированными в следующее поколение. С точки зрения решения задач оптимизации, селекция концентрирует поиск в более перспективных регионах пространства поиска, обеспечивая сходимость ЭА в точку глобального или некоторого локального оптимума, в то время как рекомбинация осуществляет поиск внутри ранее локализованной

области (потомок может унаследовать только те гены, которые были у родителей), а мутация – поиск новых регионов за счет случайных изменений в ранее найденных решениях. Более того, операция селекции может быть спроектирована с учетом дополнительных требований к решению задачи: учет ограничений, многокритериальность и другие.

В работе [154] представлен анализ известных эвристик реализации оператора селекции. В работе были исследованы и сравнены операторы селекции с точки зрения их влияния на среднюю пригодность популяции, динамику изменения пригодности, потерю разнообразия в популяции, интенсивность селективного давления и др. К сожалению, исчерпывающий анализ удастся провести только для достаточно простых задач оптимизации (например, задача ONEMAX). Тем не менее, можно сделать вывод, что различные схемы реализации селекции обладают разными характеристиками, которые меняются от задачи к задаче, следовательно, нет гарантии, что выбор лучшей из стандартных эвристик селекции окажется оптимальным или достаточно эффективным для решения произвольной задачи оптимизации. Таким образом, синтез новых эвристик селекции под конкретную задачу или класс задач позволит повысить эффективность решения, в случаях, когда эффективность стандартных эвристик недостаточна.

Поскольку операция селекция определяет вероятность для индивида быть отобранным для применения к нему последующих операций, представим операцию селекции как отображение (функцию) s на интервал $[0, 1]$. По сути функция s является алгоритмом назначения вероятностей (плотность распределения). В зависимости от выбранной эвристики, областью определения функции s являются либо вещественные числа, отражающие значение функции пригодности в пропорциональной селекции, либо целые положительные числа, определяющие ранг индивида после сортировки по значению пригодности в ранговой, турнирной селекции и селекции с усечением. Таким образом, представим оператор селекции как (2.1) или (2.2) с учетом требования нормировки (2.3).

$$s(rank_i): rank_i \rightarrow [0, 1], rank_i \in \mathbb{Z}^+, i \in Population, \quad (2.1)$$

$$s(fitness_i): fitness_i \rightarrow [0, 1], fitness_i \in \mathbb{R}, i \in Population, \quad (2.2)$$

$$\sum_{i \in Population} s(rank_i) = 1 \quad (\sum_{i \in Population} s(fitness_i) = 1), \quad (2.3)$$

где i – индивид популяции $Population$, $rank_i$ – ранг индивида после сортировки по пригодности, $fitness_i$ – пригодность индивида.

Представим и проанализируем графики функции s для следующих стандартных схем селекции:

– селекция с усечением (порог 30%) – в отборе участвует только заданное порогом число лучших индивидов с равной вероятностью;

– пропорциональная селекция – вероятность отбора пропорциональна значению пригодности индивида;

– линейная ранговая селекция – индивиды сортируются по значению пригодности, лучший индивид имеет наивысший ранг, вероятность отбора определяются по формуле (2.4);

$$p_{linearRanking}(rank_i) = \frac{2 \cdot rank_i}{PopSize \cdot (PopSize + 1)}, \quad (2.4)$$

где $p_{linearRanking}$ – вероятность отбора индивида, $PopSize$ – размер популяции.

– экспоненциальное ранжирование – после сортировки, вероятность отбора определяются по формуле (2.5), параметр c определяет селективное давление (в данной работе рассмотрены значения $c = \{0.5, 0.8, 0.95\}$);

$$p_{exponentialRanking}(rank_i) = \frac{(1-c) \cdot c^{PopSize - rank_i}}{(1 - c^{PopSize})}, \quad (2.5)$$

где $p_{exponentialRanking}$ – вероятность отбора индивида, c – параметр, управляющий селективным давлением, $c \in (0, 1)$.

Турнирная селекция не рассмотрена, т.к. алгоритм назначения вероятностей отбора в турнирной селекции с увеличением размера популяции асимптотически сходится к алгоритму назначения вероятностей в линейной ранговой селекции [155].

Графики показаны на рисунках 2.1 – 2.3. Значения вероятностей s получены для популяции размером 50 индивидов, но выводы по анализу графиков справедливы для произвольного размера. Горизонтальная ось содержит номера (ранги) индивидов после сортировки, вертикальная – вероятность быть отобранным на этапе селекции.

Анализ графиков стандартных эвристик селекции показывает, что распределение вероятностей отбора индивидов после их ранжирования по пригодности является монотонной возрастающей функцией. В случае ранговой селекции (и, соответственно, турнирной) график является гладким. В случае пропорциональной селекции график зависит от распределения значений функции пригодности. Если распределение значений пригодности в популяции близко к равномерному, что встречается при использовании механизмов поддержки разнообразия в популяции, то распределение вероятностей будет похоже на распределение в линейной ранговой селекции. Этот же факт объясняет, почему линейная ранговая селекция менее склонна к преждевременной сходимости и обеспечивает большее разнообразие в популяции (наравне с турнирной), чем другие типы селекции. Если в популяции встречаются индивиды, пригодность которых значительно лучше пригодности других членов популяции, то график становится похож на график экспоненциальной ранговой селекции, в которой вероятность выбора решений с меньшей пригодностью стремится к нулю. Данная проблема называется проблемой «супериндивида», который получает высокую вероятность отбора, из-за чего появляются его клоны, и популяция теряет разнообразие. В итоге распределение становится похожим на распределение селекции с усечением, когда клоны «супериндивида» заполняют часть популяции и имеют равную высокую вероятность отбора, а другие члены популяции – близкую к нулю. С другой стороны, подобные эвристики могут быть полезными для повышения скорости локальной сходимости, их часто называют «жадными» эвристиками (*greedy heuristics*).

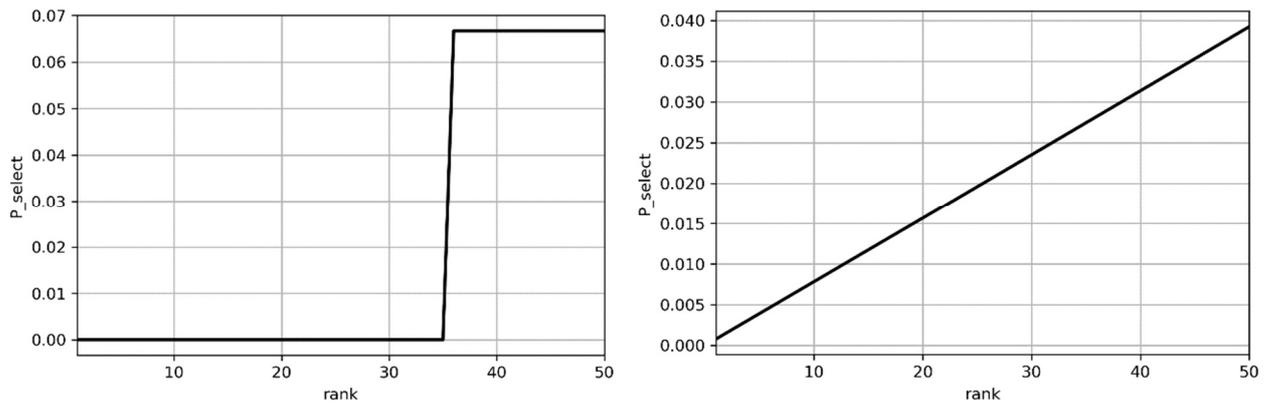


Рисунок 2.1 – График функции назначения вероятностей в селекции с усечением (слева) и линейной ранговой селекции (справа)

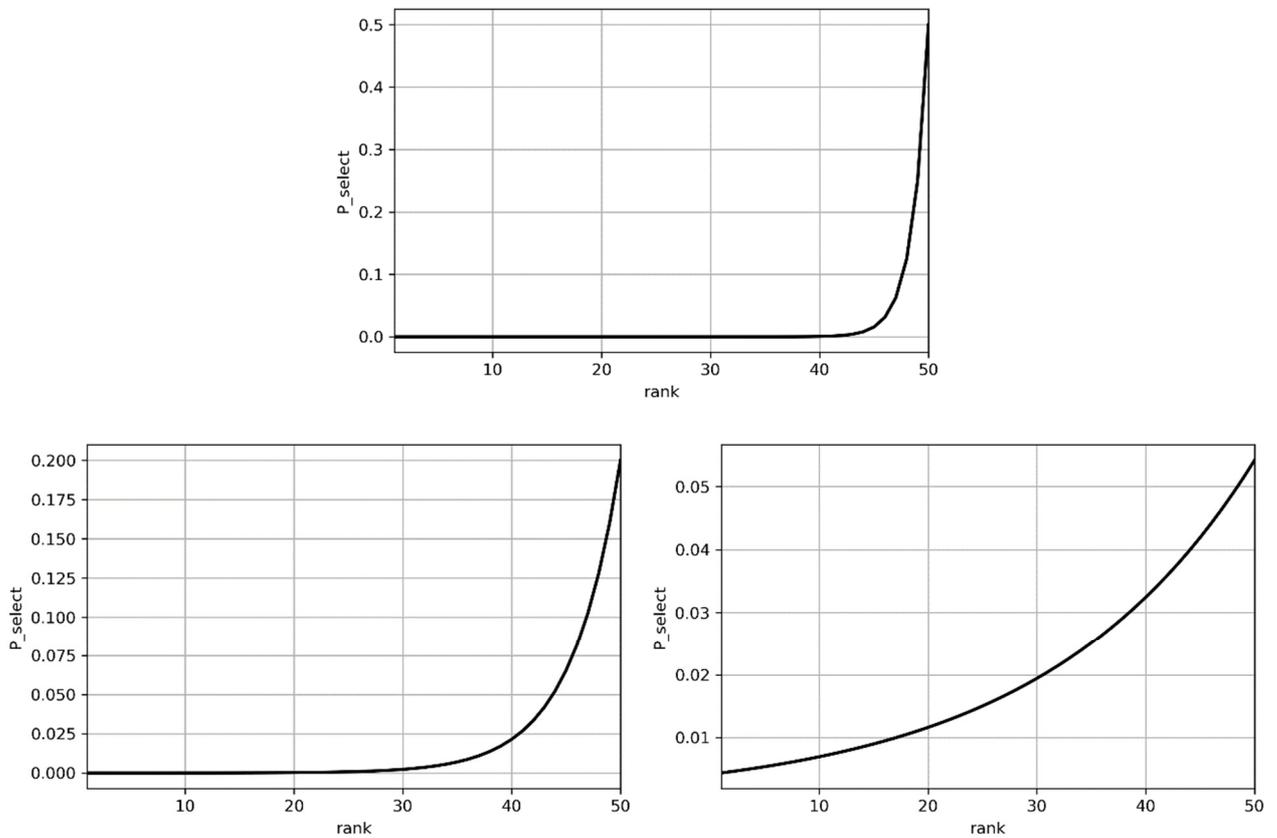


Рисунок 2.2 – График распределения вероятностей в экспоненциальной ранговой селекции с параметром $c = 0.5$ (вверху), $c = 0.8$ (внизу слева) и $c = 0.95$ (внизу справа)

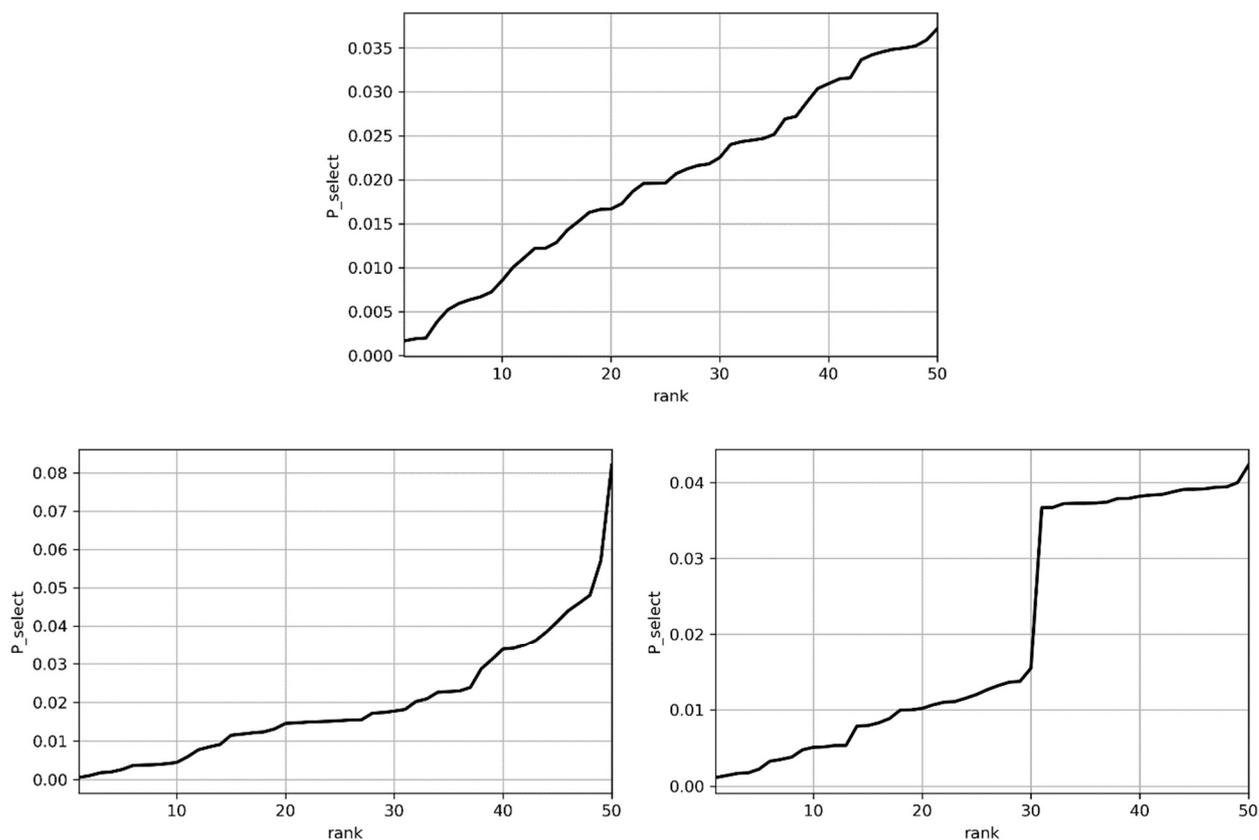


Рисунок 2.3 – График функции назначения вероятностей в пропорциональной селекции для популяции, в которой значения пригодности распределены равномерно (вверху), для популяции, в которой присутствуют индивиды с резко отличающимися значениями пригодности (внизу слева) и для той же популяции через 10 поколений (внизу справа)

Число стандартных эвристик селекции невелико, они основаны на моделировании биологических принципов (например, турнир) или на некоторых очевидных способах назначения вероятностей отбора для реализации принципов естественной эволюции (пропорционально рангам или пригодности). Очевидно, что для конкретной задачи или класса задач можно сформировать множество новых эффективных эвристик, которые будут представлены функцией назначения вероятностей отбора отличной от используемых в стандартных эвристиках. Для синтеза новых элементарных эвристик можно использовать метод ГП, а именно решить задачу символьной регрессии для формирования более эффективного распределения вероятностей отбора.

Существуют разные способы представления решений в ГП: на основе деревьев, линейный способ, способ на основе графов, жестко-типизированные представления и на основе формальных грамматик. В данном исследовании ограничимся представлением на основе деревьев, т.к. оно является наиболее популярным для задачи символьной регрессии и обладает рядом преимуществ:

- позволяет получить произвольные структуры решений, являющиеся суперпозицией элементарных функций;
- возможно использование предопределенных функций, объявляемых как терминальные элементы (автоматически определяемые функции, *automatically defined functions*, ADF);
- получаемые структуры могут иметь произвольный размер (глубина и ширина дерева);
- семантика представления обеспечивает однозначную интерпретацию деревьев решений;
- существуют эффективные способы регуляризации путем контроля роста деревьев (*bloat control*).

Дерево в ГП – это ориентированный граф, в котором каждая последующая вершина связана с одной и только одной предыдущей. Вершины дерева являются элементами одного из двух множеств:

- множество внутренних вершин дерева называется функциональным множеством. Элементы функционального множества обычно являются рабочими блоками программы: процедурами, функциями, условиями, циклами и т.д.
- множество внешних вершин дерева называется терминальным множеством. Элементы терминального множества – входные данные задачи: переменные и константы, результаты выполнения ADF, входные сигналы с сенсоров и т.д.

Дерево в ГП является генотипом подобно бинарной хромосоме в ГА, т.е. для оценки пригодности решения, необходим перевод генотипа в фенотип. В ГП для оценки фенотипа необходимо интерпретировать полученный код программы (алгоритм) и выполнить его для решения исходной задачи. Интерпретация

происходит иерархически, начиная с корня дерева. Пример представления на основе дерева показан на рисунке 2.4.

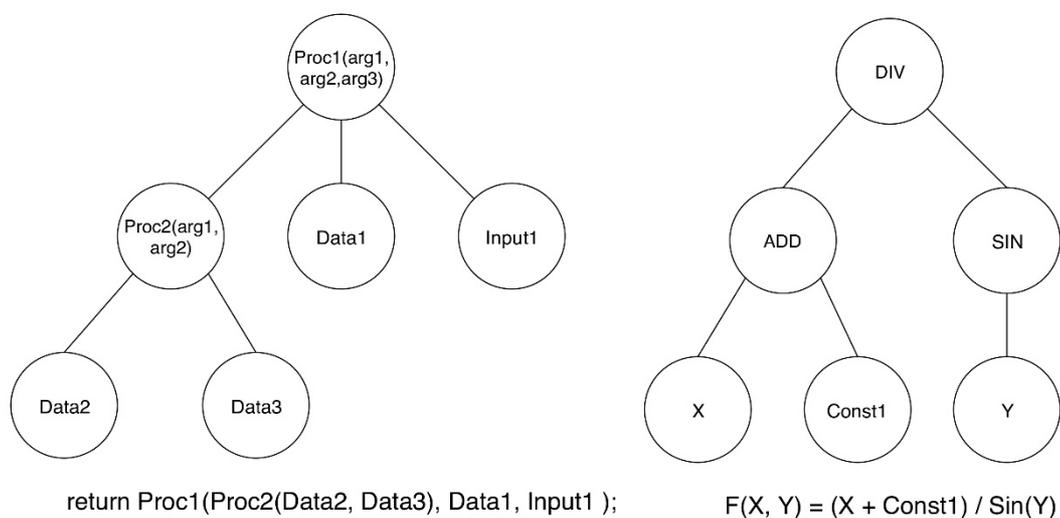


Рисунок 2.4 – Пример представления в ГП компьютерной программы (слева) и математической формулы (справа)

Общая схема алгоритма ГП (Алгоритм 3) повторяет эволюционный цикл (Алгоритм 1), основные отличия в способе представления решений и их оценке, схеме инициализации и способах реализации операторов рекомбинации и мутации.

Алгоритм 3. Общая схема ГП.

Вход: постановка задачи, способ представления семантически корректных решений, параметры ГП.

Инициализация: формирование случайной популяции индивидов с помощью метода выращивания деревьев.

Повторять, пока не выполнится условие останова:

Интерпретировать решения, выполнить программы и получить решения задачи.

Оценить пригодность популяции.

Повторять, пока не сформирована новая популяция:

Селекция: Выбрать индивидов-родителей.

Рекомбинация: Сформировать дерево потомка.

Мутация: Случайные изменения в дереве потомка.

Замещение предыдущей популяции новой.

Выход: лучшее найденное решение задачи.

Задача символьной регрессии (*symbolic regression*) заключается в отыскании в символьном виде (в виде математического выражения, аналитической формулы) зависимости между независимыми и зависимой переменными задачи, которая является наилучшей в смысле некоторого заданного критерия качества [138, 156, 157]. При этом в отличие от традиционного регрессионного анализа, где осуществляется подбор численных параметров в рамках параметрического семейства функций (например, линейная регрессия), в символьной регрессии одновременно ищется и функциональная зависимость, и ее численные параметры.

В традиционных подходах, таких как линейная регрессия, алгебраические полиномы, разложения в ряд, нейронные сети (персептрон), подбор вида (структуры) функциональной зависимости осуществляется исследователем на основе априорной информации о решаемой задаче или путем перебора, включая методы связанного Байесовского вывода [158]. В методе группового учета аргументов (МГУА, *Group Method of Data Handling*, GMDH) реализуется итеративный синтез регрессионной модели путем ее постепенного усложнения. В МГУА происходит отбор моделей на основе критерия качества аналогично селекции в ЭА, однако вид итоговой функции ограничен линейной комбинацией произведений независимых переменных [159].

В символьной регрессии решение формируется с помощью алгоритма ГП как произвольная суперпозиция элементов функционального и терминального множеств. Обычно функциональное множество включает в себя математические операции сложения, вычитания, деления и умножения, а также элементарные математические функции: \sin , \cos , \exp , возведение в степень и извлечение корня, модуль числа и другие. Терминальное множество включает независимые переменные задачи (аргументы функции) и константы, которые могут быть как произвольно заданными (например, случайно), так и специально предопределенными (например, число π , константы из физики или химии и др.).

Пусть заданы терминальное множество T_{set} (2.6) и функциональное множество F_{set} (2.7):

$$T_{set} = \{x_1, x_2, \dots, x_n, const\}, \quad (2.6)$$

$$F_{set} = \{f_1, f_2, \dots, f_k\}, f_i = f_i(x_1, \dots, x_n, const), f_i: R^n \rightarrow R, \forall i, \quad (2.7)$$

где x_1, x_2, \dots, x_n – независимые переменные задачи, $const$ – константы (численные параметры).

Обозначим A_i i -е решение в ГП, представляющее собой суперпозицию элементов функционального и терминального множеств, а Φ – множество всех возможных суперпозиций. Пусть задан критерий качества $L: A \rightarrow R$, тогда задачу символьной регрессии можно представить как задачу комбинаторной оптимизации (2.8):

$$L(A) \rightarrow \max_{A \in \Phi}. \quad (2.8)$$

Обычно задача символьной регрессии состоит в отыскании лучшего приближения функции по отношению к некоторой выборке данных (пар зависимой и независимых переменных), в этом случае в качестве критерия (2.8) используется среднеквадратическая ошибка. Во многих работах было показано, что если выборка порождается некоторой функцией и не содержит шумов, то метод ГП позволяет восстановить по точкам выборки точную формулу этой функции с точностью до значений коэффициентов [160]. В общем случае, критерий $L(A)$ может быть моделью, отражающей необходимые требования к решению поставленной задачи.

Сформулируем задачу синтеза оператора селекции для ЭА и соответствующие требования к критерию качества $L(A)$. Данную задачу можно отнести к задаче машинного обучения. Причем она обладает как признаками обучения с учителем, т.к. множество обучающих задач задано, конечно и определяется исследователем, так и признаками обучения с подкреплением, т.к. обучающие примеры не заданы парами вход-выход, а являются результатом взаимодействия со средой (моделью). Некоторые работы в этом направлении были предприняты в [161], где для синтеза оператора селекции использовался подход на основе регистровой машины (машина Минского). В [162] метод ГП был предложен для решения задачи символьной регрессии, формирующей распределение вероятностей. Похожий подход используется в [163].

Если задача синтеза оператора селекции решается для конкретной проблемы, то критерий качества $L(A)$ совпадает с критерием качеств решения исходной проблемы. В случае, если решается некоторое множество задач (класс задач), необходимо обеспечить высокую эффективность в среднем на множестве задач и, более того, высокую эффективность при решении новых задач класса, которые не были представлены на этапе синтеза, т.е. необходимо обеспечить баланс между обучением и обобщением. Для этого критерий качества $L(A)$ должен отражать оценку эффективности ЭА, для которого оператор селекции синтезируется, полученную на основе многократных запусков на задачах обучающего множества и статистической обработки результатов численных экспериментов. Для контроля переобучения необходимо использовать дополнительные оценки валидации.

Основная идея подхода к синтезу оператора селекции показана на рисунке 2.5. Метод ГП является мета-процедурой, осуществляющей формирование и поиск символического выражения – формулы, определяющей распределение вероятностей для выбранного ЭА. Решение в ГП может строиться как из базовых математических операции и функций, так и из функций назначения вероятностей известных схем селекции (ADF). Каждое решение в ГП при оценке пригодности запускает процедуру исследования эффективности ЭА, использующего синтезированный оператор селекции. Для этого дерево решения из ГП должно быть интерпретировано как формула, нормализовано (условие (2.3)), чтобы представлять плотность распределения вероятностей, и использовано в прогонах ЭА. В свою очередь ЭА решает каждую из задач обучающего множества используя синтезированный оператор. Оценки эффективности ЭА всех независимых прогонов усредняются по прогонам и по задачам обучающего множества. Итоговая средняя оценка эффективности ЭА определяет значение пригодности решения в ГП. Данная двухуровневая процедура ГП-ЭА повторяется до выполнения условия останова в ГП. Лучшее найденное решение в ГП является решением задачи синтеза оператора селекции.

Предложенный подход использует обучающее множество задач и требует значительных вычислительных затрат, поэтому реализуется в офлайн режиме.

Полученная в итоге эвристика создается (конструируется) из элементарных операций. Согласно классификации Бёрка [95], предложенный подход можно отнести к конструктивной офлайн гиперэвристике.

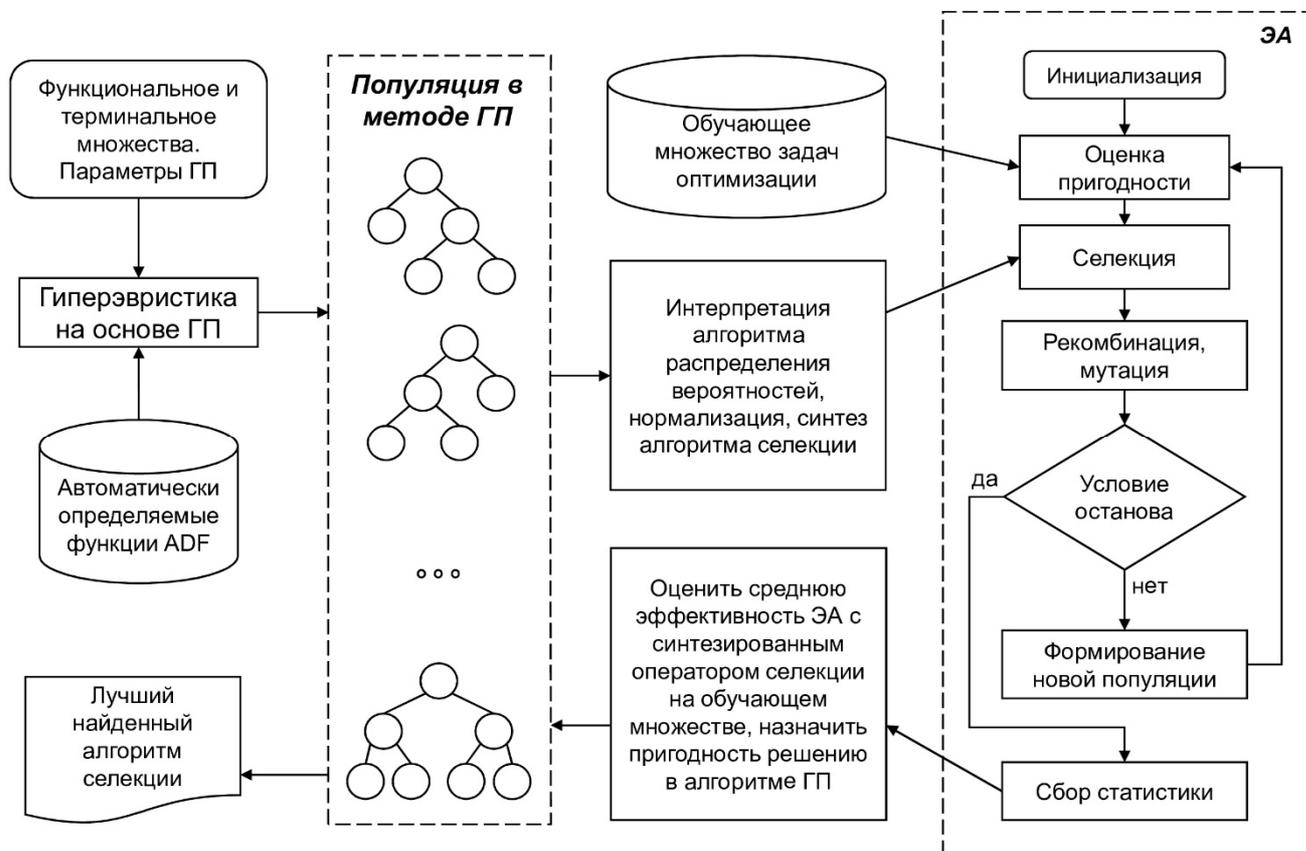


Рисунок 2.5 – Общая схема синтеза оператора селекции с помощью метода ГП

2.2 Конструктивная офлайн гиперэвристика синтеза оператора селекции на основе метода генетического программирования

Гиперэвристика – это метод решения задачи оптимизации (1.11). В задаче синтеза оператора селекции критерий ψ зависит лишь от эвристики управления операцией селекции, т.е. $\psi(selection, \{p_{selection}\})$ и определяется пригодностью индивидов в методе ГП.

В большинстве прикладных задач оптимизации основным критерием является точность нахождения глобального оптимума. Для синтеза оператора

селекции, обеспечивающего наибольшую точность в среднем на заданном множестве задач оптимизации, формализуем критерий эффективности (оценку пригодности) как:

$$fitnessGP(A) \rightarrow \max_{A \in \Phi}, \quad (2.9)$$

$$fitnessGP(A) = \frac{1}{F} \sum_{i=1}^F \left(\frac{1}{R} \sum_{j=1}^R PerformanceEA_{ij}(A) \right), \quad (2.10)$$

$$PerformanceEA_{ij} = \frac{\max(Dist_i) - Dist_{ij}}{\max(Dist_i)}, PerformanceEA_{ij} \in [0,1] \quad (2.11)$$

$$Dist_{ij} = \|\bar{x}_i^* - (\bar{x}_{bestFound})_{ij}\|, \quad (2.12)$$

где $fitnessGP$ – пригодность решения A в методе ГП, решение A представляет собой формулу (алгоритм) вычисления вероятностей отбора, F – число задач обучающего множества, R – число независимых запусков ЭА на одной задаче, $PerformanceEA_{ij}$ – оценка эффективности ЭА на i -й задаче в j -м запуске, $Dist_{ij}$ – расстояние от лучшего найденного решения $\bar{x}_{bestFound}$ до истинного значения глобального оптимума \bar{x}^* на i -й задаче в j -м запуске, $\max(Dist_i)$ – максимально возможное расстояние для i -й задачи.

Оператор селекции может быть спроектирован и с учетом других требований. В прикладных задачах оптимизации часто требуется повысить скорость нахождения решения, выраженную через число итераций или число вычислений целевой функции (мы не рассматриваем скорость в единицах астрономического времени, т.к. она определяется не только свойствами алгоритма, но и, в большей степени, программной и аппаратной реализацией). Поскольку требования точности и скорости нахождения решения практически всегда являются конфликтующими, данную задачу можно сформулировать как задачу условной оптимизации (2.13) при наличии ограничения (2.14).

$$ERT(A) = \frac{1}{F} \sum_{i=1}^F \left(\frac{1}{R} \sum_{j=1}^R \left[1 - \frac{\min\{RT_{ij}(A), ERT_r\}}{ERT_r} \right] \right), \quad (2.13)$$

$$g_{perf}(A) = \left[\frac{1}{F} \sum_{i=1}^F \left(\frac{1}{R} \sum_{j=1}^R SuccessRate_{ij}(A) \right) - \theta \right] \geq 0, \quad (2.14)$$

где ERT – оценка среднего числа вычислений целевых функции (*estimated runtime*) обучающего множества для решения A в ГП, ERT_r – оценка вычислений целевых функции обучающего множества, полученная для ЭА со стандартной линейной ранговой селекцией (оценка снизу), $RT_{ij}(A)$ – число вычислений целевой функции (*runtime*) для нахождения глобального оптимума с заданной точностью на i -й задаче в j -м запуске, θ – пороговое значение потери эффективности ЭА (определяется экспертно или по результатам решения задачи (2.9)-(2.12)), $g_{perf}(A)$ – ограничение задачи, $SuccessRate$ – процент успешных прогонов (когда глобальный оптимум найден с заданной точностью).

Тогда пригодность решения в методе ГП для задачи (2.9) определяется как (2.15) – (2.16). В данной постановке для работы с ограничением (2.14) используется метод динамических штрафов [164].

$$fitnessGP(A) = ERT(A) - \left(\frac{t}{2}\right)^2 \cdot \left(\hat{g}_{perf}(A)\right)^2, \quad (2.15)$$

$$\hat{g}_{perf}(A) = \min\{0, g_{perf}(A)\} \quad (2.16)$$

где $t = \overline{1, T}$ – номер итерации ЭА, T – максимальное число итераций ЭА в одном независимом запуске.

Гиперэвристика синтеза оператора селекции имеет следующий вид:

Метод 1. Конструктивная офлайн гиперэвристика на основе алгоритма ГП для синтеза оператора селекции ЭА.

Вход: Обучающее и тестовое множество задач оптимизации.

Инициализация: Определить основные параметры ГП и ЭА, задать функциональное и терминальное множества для ГП.

Повторять, пока не выполнится условие остановки ГП (внешний цикл ГП):

Сформировать очередную популяцию в ГП.

Повторять, пока не решены все задачи из обучающего множества (внутренний цикл ЭА):

Выбрать очередную задачу, применить ЭА с оператором селекции, синтезированным в ГП.

Собрать статистику о результатах независимых прогонов ЭА на текущей задаче, оценить эффективность ЭА.

Оценить среднюю эффективность ЭА на множестве задач и присвоить это значение в качестве пригодности решения в ГП, используя (2.9) – (2.12) или (2.13) – (2.16).

Выход: Лучшее найденное решение в ГП.

Терминальное множество ГП содержит только одну независимую переменную – ранг индивида после попарного сравнения и сортировки по пригодности, а также случайные константы. Функциональное множество содержит бинарные математические операции (+, −, ×, ÷) и математические функции (*sin*, *exp*). Также используются 2 автоматически определяемые функции (ADF, *automatically defined functions*), представляющие собой операторы линейного ранжирования (2.4) и экспоненциального ранжирования с коэффициентом $c = 0.8$ (2.5).

2.3 Экспериментальное исследование предложенного метода

Для исследования эффективности метода 1 было выбрано обучающее множество задач, которое содержит 24 эталонные тестовые задачи однокритериальной безусловной глобальной параметрической оптимизации алгоритмически заданных целевых функций (*Black-box optimization benchmark*, ВВОВ), используемые научным сообществом для оценки и сравнения новых ЭА, а также при проведении соревнований по оптимизации «черного ящика» на конференциях IEEE CEC и GECCO. Задачи ВВОВ сгруппированы по свойствам в 5 групп: сепарабельные, плохообусловленные, хорошообусловленные, многоэкстремальные с адекватной структурой расположения оптимумов и многоэкстремальные со слабой структурой расположения оптимумов. Большинство функций набора являются обманчивыми (десептивными) [119].

Поскольку задача символьной регрессии относится к задаче машинного обучения, для верификации обобщающей способности решения необходимо разбить множество задач на обучающее и тестовое множество. В данном

исследовании используется 5-кратная кросс-валидация, использующая стратифицированный отбор задач в обучающие и тестовые наборы.

В таблицах 2.1 и 2.2 приведены используемые в экспериментах параметры алгоритмов ГП и ЭА, включенных в метод 1. Данный набор параметров подобран экспериментально в ходе предварительных экспериментов. В качестве ЭА используется бинарный ГА, который, как обсуждалось ранее, имеет ряд преимуществ перед ЭА, работающими только с вещественными переменными.

Таблица 2.1 – Настройки ГП в методе 1

Параметр	Значение
Размер популяции	50
Максимальное число поколений	1000
Метод инициализации (метод выращивания деревьев)	Полный (<i>full</i>)
Максимальная глубина деревьев при инициализации	6
Функциональное множество с вероятностями выбора элементов при инициализации	$F_{set} = \{ +/0.5, -/0.2, */0.1, \div/0.1, \sin/0.05, \exp/0.05 \}$
Терминальное множество с вероятностями выбора элементов при инициализации	$T_{set} = \{ rank_i/0.5, const/0.2, ADF1/0.1, ADF2/0.1 \}$
Константы терминального множества	Равномерно случайно в интервале [0,1]
ADF1 (линейная ранговая селекция)	(2.4)
ADF2 (экспоненциальная ранговая селекция)	(2.5) с параметром $c = 0.8$
Рекомбинация / вероятность применения	Одно-точечная / 0.95
Мутация / вероятность применения	Одно-точечная / 0.01
Функция пригодности	(2.9)-(2.12) и (2.13)-(2.16)

Таблица 2.2 – Настройки ГА в методе 1

Параметр	Значение
Размер популяции	50
Максимальное число поколений	50
Точность кодирования вещественных переменных	1.0E-03
Длина хромосомы (бит)	$n_{bit} = 28$
Инициализация	Случайно равномерно в бинарном пространстве поиска
Тип селекции	Определяется решением в алгоритме ГП

Продолжение таблицы 2.2

Параметр	Значение
Рекомбинация / вероятность применения	Равномерное равновероятное с использованием двух родителей / 1.0
Мутация / вероятность применения	Инверсия битов / $\frac{1}{n_{bit}}$
Функция пригодности	Целевые функции из набора ВВОВ
Число независимых запусков для оценки эффективности решения ГП	40

В таблице 2.3 представлены сформированные случайным образом для кросс-валидации наборы задач из множества эталонных задач ВВОВ.

Таблица 2.3 – Наборы задач для кросс-валидации

Номер набора задач	Номера функций ВВОВ в обучающем множестве	Номера функций ВВОВ в тестовом множестве
1	1-2, 4-8, 10-11, 13-14, 16-19, 21-24	3, 9, 12, 15, 20
2	2-8, 10-13, 15, 17-23	1, 9, 14, 16, 24
3	1-4, 6, 8-11, 13-17, 19-21, 23-24	5, 7, 12, 18, 22
4	1-3, 5-8, 10-12, 14-18, 20-23	4, 9, 13, 19, 24
5	1, 3-7, 9, 11-17, 19-20, 22-24	2, 8, 10, 18, 21

Программная реализация алгоритмов для проведения численных исследований выполнена на языке Python 3.7 с использованием среды научного программирования Spyder. Метод 1 и ГА реализованы с использованием библиотеки NumPy, статистические тесты реализованы с использованием пакета Statistics в составе библиотеки SciPy (scipy.stats). ГП в составе метода 1 использует библиотеку DEAP [165] с поддержкой параллельных вычислений на основе пакета Multiprocessing на ЭВМ с процессором Intel Core i7-4790, использующим 8 параллельных потоков на частоте 3.6ГГц. Реализация эталонных тестовых задач ВВОВ на языке Python COCO (*COmparing COntinuous Optimizers*) взята из [166]. Для ранжирования используется реализация подхода глубокого статистического сравнения на языке программирования R, предоставленная авторами подхода.

В качестве оценки эффективности ГА с синтезированным методом 1 оператором селекции в работе используется процент успешных запусков ГА, в которых глобальный оптимум найден с заданной точностью, которая по условиям использования бенчмарка ВВОВ равна 10^{-8} . Полученные результаты сравнены с результатами решения эталонных задач ГА со стандартными эвристиками селекции, которые обсуждались выше: пропорциональная селекция, линейное ранжирование и экспоненциальное ранжирование с параметром $c = 0.8$.

В результате применения метода 1 для решения задачи (2.9) – (2.12) синтезирована следующая формула для вычисления вероятностей отбора:

$$p_{synth}(rank_i) = \frac{10 \sin(0.1rank_i - 5) + rank_i - 9.82453}{707.405}. \quad (2.17)$$

График функции назначения вероятностей для (2.17) показан на рисунке 2.6. Рисунок 2.7 демонстрирует изменение пригодности лучшего найденного решения и средней пригодности по популяции в алгоритме ГП.

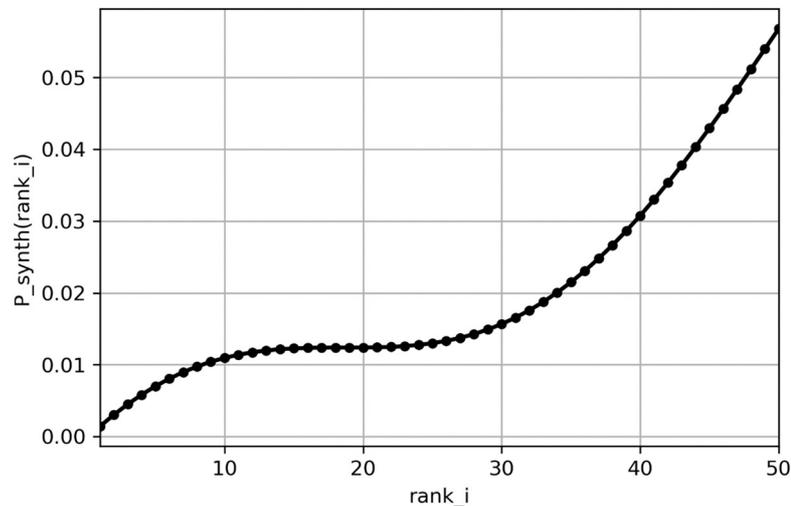


Рисунок 2.6 – График функции назначения вероятностей для синтезированного оператора селекции (2.17)

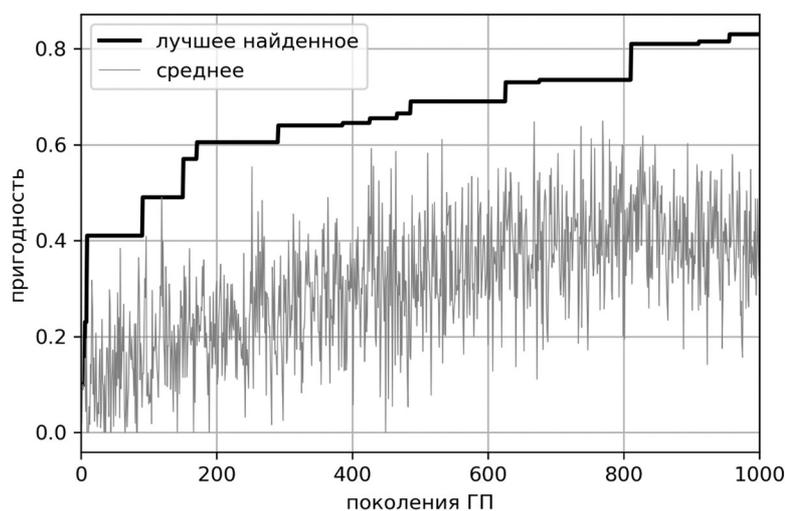


Рисунок 2.7 – График изменения пригодности ГП при решении задачи (2.9) – (2.12)

Таблица 2.4 содержит результаты оценки эффективности ГА (процент успешных запусков) с синтезированным оператором селекции на тестовых наборах. Детальные результаты по функциям набора представлены в Приложении А. На рисунке 2.8 представлена диаграмма размаха для оценок из таблицы 2.4. Как видно по диаграмме размаха, разброс относительно среднего симметричен, сильного смещения оценок на каком-либо из наборов нет. Можно сделать вывод о том, что синтезированный оператор селекции способен эффективно решать новые задачи класса.

Таблица 2.4 – Результаты кросс-валидации для задач ВВОВ с использованием селекции (2.17)

Тестовый набор	Среднее по набору
1	84.00
2	85.00
3	84.00
4	83.50
5	84.50
Среднее по наборам	84.20

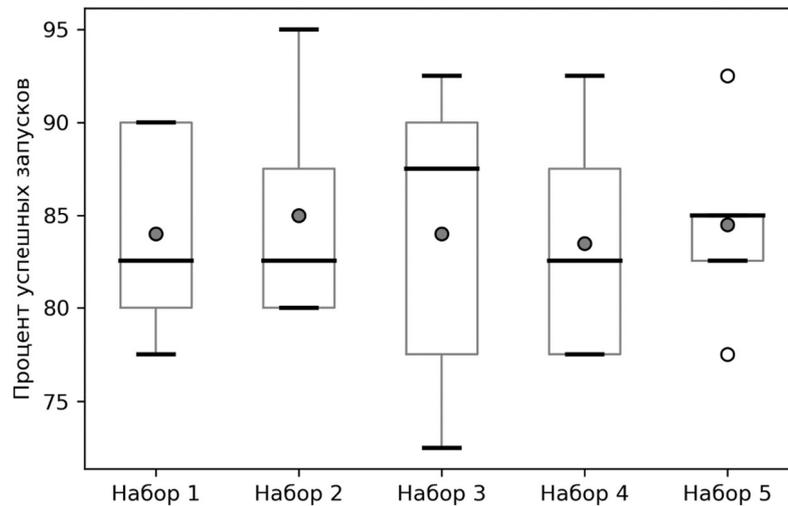


Рисунок 2.8 – Диаграмма размаха для оценок из таблицы 2.4

Сравним результаты оценки эффективности синтезированного оператора селекции со стандартными эвристиками. Очевидно, что на каждой отдельной задаче оптимизации лучшим окажется тот или иной оператор. Для задач с алгоритмически заданными целевыми функциями нельзя заранее сделать выбор наилучшей эвристики, а многократные эксперименты с разными типами селекции могут быть затруднены (например, из-за ограничений на число вычислений целевой функции). Поэтому целесообразно сравнить эффективность синтезированного оператора с оценкой эффективности случайного выбора одного из стандартных операторов, которая может быть получена как среднее по частным оценкам эффективности операторов. Если новый оператор превосходит оценку случайного выбора, то его использование можно считать более перспективным для произвольно выбранной задачи исследуемого класса, для которой неизвестен лучший в смысле эффективности решения задачи оператор селекции.

В таблице 2.5 представлены результаты численных экспериментов в виде средних оценок эффективности ГА со стандартными эвристиками отбора и с оператором селекции (2.17) усредненные по всем задачам бенчмарка ВВОВ. Все настройки ГА за исключением оператора селекции такие же как в таблице 2.2. Среднее по столбцам 2-4 отражает оценку эффективности случайного выбора одного из стандартных операторов. На рисунке 2.9 показана диаграмма размаха

оценок на всех задачах бенчмарка. Результаты экспериментов проранжированы с использованием метода глубокого статистического сравнения, ранги для типов селекции показаны на рисунке 2.10. Детальные результаты экспериментов в задачах синтеза эвристик селекции приведены в Приложении А.

По диаграмме размаха видно, что среднее и медиана для синтезированного оператора селекции лучше результатов стандартных операторов и лучше среднего по операторам. При этом разброс результатов существенно меньше, результаты стабильнее (повторяемые), в то время как стандартные операторы могут на некоторых задачах как превосходить синтезированный, так и иметь низкую эффективность (выбросы на диаграмме). Самые нестабильные результаты дает пропорциональная селекция, которая эффективна в основном на наиболее простых задачах, где обеспечивает быструю сходимость.

Из таблицы 1 приложения А видно, что на 14 из 24 задачах синтезированный оператор имеет больший процент успешных запусков, на 10 задачах лучшим оказывается один из стандартных. При этом синтезированный алгоритм превосходит среднее по стандартным операторам (эффективность случайного выбора) на 21 задаче, на двух задачах они равны, и только на одной хуже.

В таблице 2.6 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов с применением теста Манна-Уитни-Уилкоксона. В таблице показано, сколько раз на эталонных функциях ВВОВ синтезированная эвристика была лучше, сравнима или хуже стандартных типов селекции. Как видно из таблицы, различия в результатах статистически значимы за исключением сравнения на одной задаче с пропорциональной селекцией, на 3 задачах для линейной ранговой и на 5 задачах из 24 для экспоненциального ранжирования.

Таблица 2.5 – Результаты сравнения типов селекции

Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование	Среднее по столбцам 2-4	Оператор синтезированный методом 1
64.4	75.8	77.6	72.6	83.9

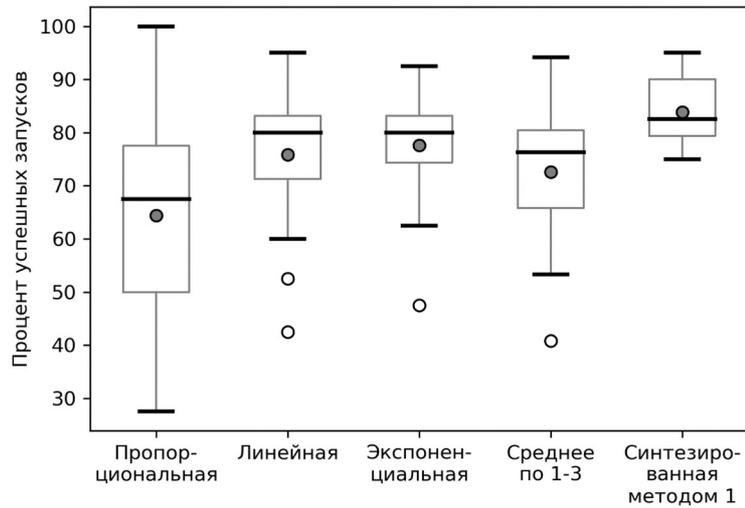


Рисунок 2.9 – Диаграмма размаха оценок операторов селекции

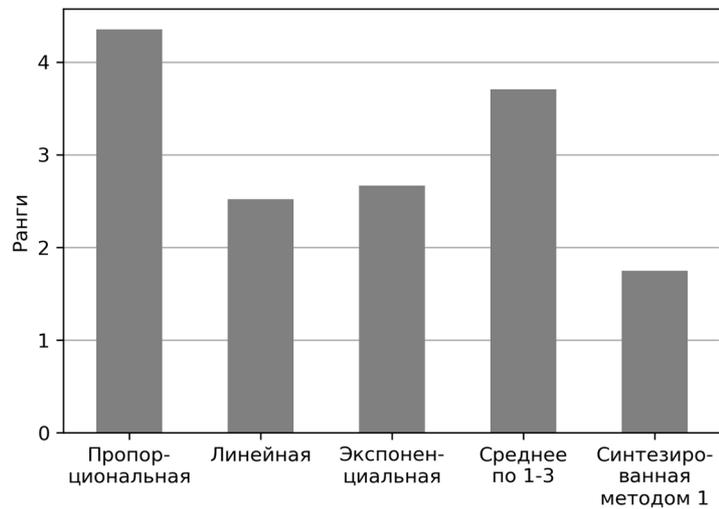


Рисунок 2.10 – Результаты ранжирования операторов селекции на ВВОВ

Таблица 2.6 – Результаты теста Манна-Уитни-Уилкоксона

Оператор синтезированный методом 1	Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование
Лучше	20	15	14
Нет статистически значимой разницы	1	3	5
Хуже	3	6	5

В итоге, можно сделать вывод, что оператор селекции (2.17), синтезированный при решении задачи (2.9) – (2.12) предпочтительнее случайного выбора одного из стандартных в задачах глобальной оптимизации с алгоритмически заданными целевыми функциями. Более того, на эталонных задачах ВВОВ синтезированная эвристика оказывается лучше стандартных на большинстве задач.

Применим метод 1 для решения задачи (2.9), (2.15) – (2.16). Настройки ГА и ГП аналогичны представленным в таблицах 2.1-2.2 за исключением функции пригодности в ГП. Численные эксперименты проведены для пороговых значений в ограничении (2.14), равных $\theta = \{60, 70, 80\}$.

Синтезирована следующая формула для вычисления вероятностей отбора при пороговом значении $\theta = 60$:

$$p_{synth}(rank_i) = \frac{0.25rank_i^2 + 50 \sin(0.1rank_i - 5)}{10349.4}. \quad (2.18)$$

График функции назначения вероятностей для (2.18) показан на рисунке 2.11. Рисунок 2.12 демонстрирует изменение пригодности лучшего найденного решения и средней пригодности по популяции в алгоритме ГП (отрицательные значения вызваны увеличением штрафа на каждой итерации).

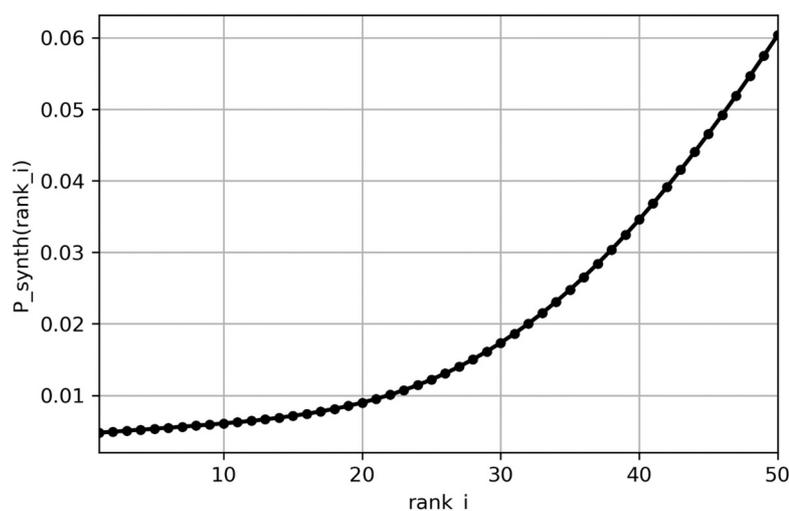


Рисунок 2.11 – График функции назначения вероятностей для синтезированного оператора селекции (2.17)

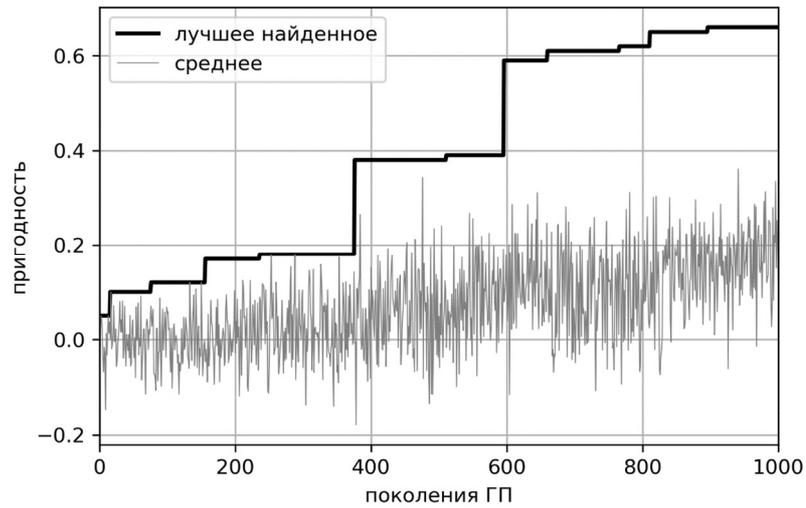


Рисунок 2.12 – График изменения пригодности ГП при решении задачи (2.9), (2.15) – (2.16)

Для пороговых значений $\theta = 70$ и $\theta = 80$ получены следующие функции назначения вероятностей: (2.19) и (2.20). Их графики показаны на рисунках 2.13 и 2.14, соответственно.

$$p_{synth}(rank_i) = \frac{0.1 \cdot e^{0.2 \cdot rank_i} - 0.5 \cdot (rank_i - 40)^2 + 760.38}{39707.08}. \quad (2.19)$$

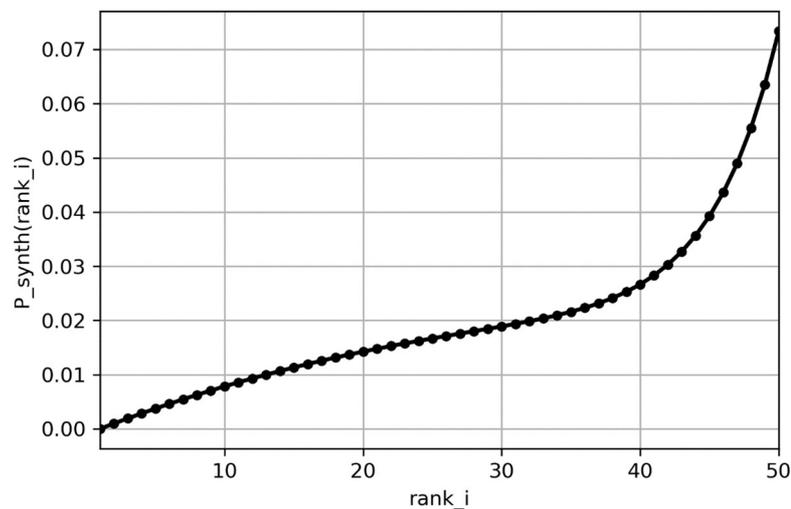


Рисунок 2.13 – График функции назначения вероятностей для синтезированного оператора селекции (2.19)

$$p_{synth}(rank_i) = \frac{rank_i - (rank_i - 2) \cdot \sin(0.05 \cdot rank_i)}{1422.26}. \quad (2.20)$$

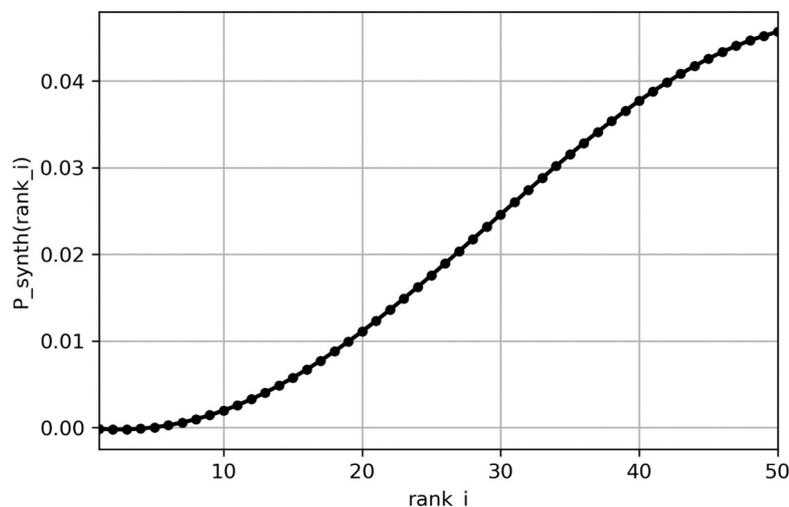


Рисунок 2.14 – График функции назначения вероятностей для синтезированного оператора селекции (2.20)

В таблицах 2.7 и 2.8 представлены результаты сравнения среднего числа итераций, необходимых для нахождения глобального оптимума, усреднённые по всем задачам бенчмарка ВВОВ и результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов, соответственно. Диаграмма размаха показана на рисунке 2.15. Детальные результаты экспериментов приведены в Приложении А.

Как видно из таблиц, с помощью метода 1 удастся синтезировать эвристику, которая позволяет находить глобальный оптимум в среднем за меньшее число итераций чем стандартные операторы селекции, однако платой за это является потеря надёжности (процента успешных прогонов ГА). При более строгом ограничении $\theta=80$ синтезированный оператор проигрывает по критерию числа итераций, при этом сохраняет высокую надёжность. На практике из-за того, что эти два критерия являются конфликтующими, исследователю необходимо прийти к некоторому компромиссу.

Экспериментальное исследование предложенной конструктивной офлайн гиперэвристики синтеза оператора селекции на основе метода генетического

программирования (метод 1) показало, что данный подход позволяет формировать новые базовые эвристики ЭА, которые демонстрируют высокую эффективность в среднем на множестве исследуемых задач и превосходят среднюю эффективность базовых операторов селекции, следовательно, использование синтезированных эвристик предпочтительнее случайного выбора одного из стандартных операторов для задач типа «черный ящик».

Таблица 2.7 - Среднее число итераций, необходимых для нахождения глобального оптимума на бенчмарке ВВОВ

Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование	Среднее по столбцам 2-4	Синтезированный оператор $\theta=60$	Синтезированный оператор $\theta=70$	Синтезированный оператор $\theta=80$
33.7	30.8	31.4	32.3	24.4	28.9	33.6

Таблица 2.8 – Результаты теста Манна-Уитни-Уилкоксона

Оператор синтезированный методом 1		Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование
$\theta=60$	Лучше	19	17	17
	Нет статистически значимой разницы	0	4	1
	Хуже	5	3	6
$\theta=70$	Лучше	14	15	14
	Нет статистически значимой разницы	5	5	3
	Хуже	5	4	7
$\theta=80$	Лучше	13	8	7
	Нет статистически значимой разницы	2	4	6
	Хуже	9	12	11

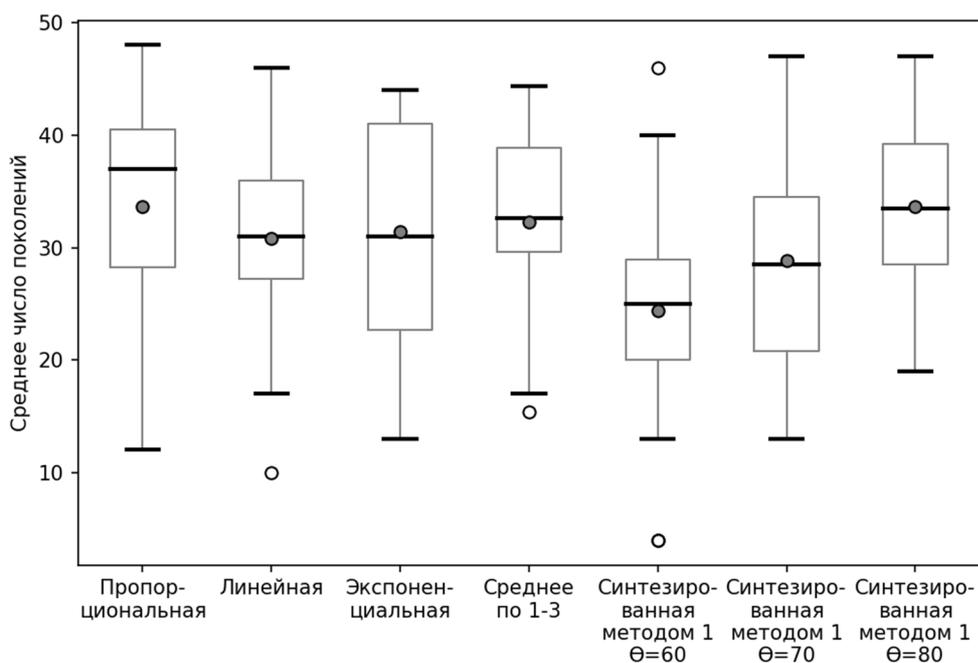


Рисунок 2.15 – Диаграмма размаха для среднего числа поколений

2.4 Решение практической задачи синтеза оператора селекции для идентификации партий электронных компонентов аэрокосмического применения

Для апробации метода 1 была решена практическая задача проектирования эвристики отбора для ЭА в задаче идентификации партий электронных компонентов аэрокосмического применения.

Современные бортовое оборудование и системы управления, используемые в аэрокосмической отрасли, предъявляют высокие требования к точности и надежности используемых электронных компонент. На сегодня в России подобные электронные компоненты не выделяют при производстве в отдельный класс, а потому отсутствуют специализированное производство и контроль качества [167]. Для контроля качества проводится отбраковка на стадии производства и приемочные испытания [168]. Одна из проблем таких испытаний заключается в том, что выборочные тесты изделий могут характеризовать электронные компоненты однородной группы. В то же время анализируемые изделия могут

быть от разных поставщиков (производителей) или произведены в разные периоды времени, и зачастую являются неоднородными из-за применения разного сырья и изменений в производственном процессе. Таким образом необходимо выявить различные группы однородных изделий и проводить испытания для каждой из групп.

Задача идентификации партий электронных компонентов аэрокосмического применения является задачей обучения без учителя и может быть сформулирована как задача кластеризации [169, 170]. В работах [168, 171] рассматриваются различные постановки данной задачи и способы ее решения. Для задач кластеризации с непрерывными переменными часто применяют методы типа k -средних, целью которых является размещение k точек, определяющих центры кластеров, которые являются решением задачи минимизации суммы квадратов расстояний точек кластеров от соответствующих центров (*Minimum Sum of Squares Clustering*, MSSC). В задаче идентификации партий электронных компонентов аэрокосмического применения различия в значениях измеряемых параметров малы и часто ограничены точностью используемого измерительного оборудования, а потому по результатам тестов вместо вещественных в базу данных попадают дискретные значения (шаг дискретизации определяется точностью измерительного оборудования). Для задач дискретной кластеризации применяются методы типа k -медиан или k -медоид, которые имеют схожие с k -средним преимущества и недостатки. В частности, все k -центроидные методы чувствительны к начальному размещению центров кластеров. Случайное размещение центров может привести к некорректному решению даже простых задач кластеризации, поэтому в последние годы было предложено много методов инициализации положений центров. Один из наиболее эффективных и популярных подходов – использование ЭА.

Задача k -центроидов (k -средних для квадрата эвклидова расстояния, k -медиан для других метрик, k -медоидов, если центры выбираются из множества объектов) является обобщением задачи размещения Ферма-Вэбера [172, 173]:

$$\sum_{i=1}^S w_i d(C_j, X_i) \rightarrow \min_{C_j, j=1, k}, \quad (2.21)$$

где X – множество объектов для группировки (точки), C – множество k центров кластеров, w – весовые коэффициенты важности объектов (при отсутствии информации $\forall i, w_i = 1$) и $d(*)$ – мера расстояний.

Задача (2.21) является NP-трудной, для ее решения было предложено множество эвристик, включая эвристики для ЭА. В [174] был предложен универсальный подход на основе ЭА.

На практике данные, собранные для анализа конкретных типов электронных изделий, являются уникальными и для работы с ними необходимы специальные алгоритмы кластеризации, учитывающие особенности данных. Поэтому вместо универсальных подходов, в работах [171, 175] были предложены специальные жадные эвристики, которые обеспечивают лучшую сходимость для данной прикладной задачи.

Применим метод 1 для синтеза операций селекций для повышения эффективности ЭА, решающего задачу идентификации партий электронных компонентов 140УД25АС1В и 1526ИЕ10. Наборы данных для анализа были предоставлены ОАО «Испытательный технический центр – НПО ПМ» (г. Железногорск) и ранее использовались при построении новых ЭА в работах [176, 177].

Набор данных для прецизионного усилителя 140УД25АС1В содержит информацию о 56 изделиях из 3 партий. Каждый экземпляр описан 42 измеренными параметрами. Набор данных для КМОП микросхемы 1526ИЕ10 включает информацию о 3987 изделиях из 7 партий. Каждый экземпляр описан 202 измеренными параметрами. Поскольку для данных экземпляров принадлежность к партиям известна, есть возможность после решения задачи кластеризации оценить точность классификации.

Для размещения кластеров будем использовать следующий алгоритм чередующихся распределений-размещений (*Alternating Location-Allocation*, ALA) [168]. Пусть $X = \{X_1, \dots, X_S\}$ обучающая выборка, где $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}, i = \overline{1, S}$ – объекты, которые необходимо кластеризовать.

$C^t = \{c_1^t, \dots, c_k^t\}$ – множество центров кластеров на t -м шаге работы алгоритма кластеризации, $c_j^t = \{c_{j1}^t, \dots, c_{jn}^t\}, j = \overline{1, k}$ и k – число кластеров.

Алгоритм 4. ALA-алгоритм для задачи кластеризации.

Вход: $X = \{X_1, \dots, X_S\}, C^0 = \{c_1^0, \dots, c_k^0\}$.

Повторять, пока $\exists (cluster_j^{t+1} \neq cluster_j^t), j = \overline{1, k}$:

Для $\forall X_i, i = \overline{1, S}$ найти ближайший центр $c_i^{closest} = \arg \min_{j=\overline{1, k}} \|X_i - c_j^t\|$.

Вычислить новые центры C^{t+1} для всех кластеров $cluster_j^{t+1} = \{i \in \{1, N\} | c_i^{closest} = j\}, j = \overline{1, k}$, используя формулу (2.22).

$$c_{jl}^{t+1} = \frac{\sum_{i \in cluster_j^{t+1}} w_i x_{il}}{\sum_{i \in cluster_j^{t+1}} w_i}, l = \overline{1, n}, j = \overline{1, k} \quad (2.22)$$

где $w_i = 1, \forall i$, если не заданы коэффициенты важности для объектов кластеризации.

Выход: центры кластеров $C^* = C^{t+1}, C^* = \{c_1^*, \dots, c_k^*\}$.

Для определения начальных центров $C^0 = \{c_1^0, \dots, c_k^0\}$ в алгоритме 4 будем использовать бинарный ГА (по схеме Алгоритма 1), т.к. область изменения значений параметров для задачи идентификации партий электронных компонентов определена, а сами значения носят дискретный характер. Хромосомы индивидов в ГА кодируют начальные центры кластеров C^0 , пригодность определяется по следующей формуле:

$$fitnessGA(C^0) = \sum_{i=1}^S \min_{C \in C^*} \|X_i - C\|, \quad (2.23)$$

где $fitnessGA$ – пригодность индивида в ГА, C^* – результат применения Алгоритма 4 из начальных центров C^0 , $\|*\|$ – евклидово расстояние.

В методе 1 настройки алгоритма ГП аналогичны приведенным в таблице 2.1, за исключением функции пригодности, в которой усреднение проводится только по числу независимых запусков и она минимизируется (учитывается при сортировке и ранжировании индивидов):

$$fitnessGP(A) \rightarrow \min_{A \in \Phi}, \quad (2.24)$$

$$fitnessGP(A) = \frac{1}{R} \sum_{r=1}^R fitnessGA(C_{bestFound,r}^A), \quad (2.25)$$

где $C_{bestFound,r}^A$ – лучшее найденное решение задачи кластеризации в прогоне r при использовании синтезированной эвристики A .

Настройки ГА совпадают с приведенными в таблице 2.2 за исключением размера популяции, который равен 100 (значение подобрано экспериментально).

В результате применения метода 1 получены следующие выражения для функции назначения вероятностей в операторе селекции: выражение (2.26) для 140УД25АС1В и выражение (2.27) для 1526ИЕ10. График функции (2.26) и графики изменения пригодности в алгоритме ГП для данных 140УД25АС1В показаны на рисунках 2.16 и 2.17. Для задачи 1526ИЕ10 – на рисунках 2.18 и 2.19.

$$p_{synth}(rank_i) = 1.59 \cdot 10^{-4} \cdot rank_i + \frac{(rank_i - 30.5)^6}{6.28 \cdot 10^{12}}, \quad (2.26)$$

$$p_{synth}(rank_i) = \frac{\sin(0.085 \cdot (rank_i - 10))}{252.53} + \frac{(rank_i - 15)^2}{2.1 \cdot 10^5}. \quad (2.27)$$

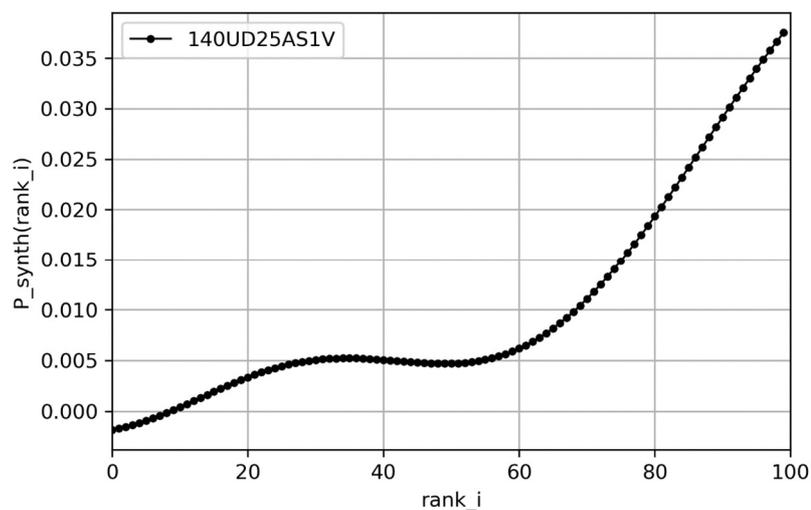


Рисунок 2.16 – График функции назначения вероятностей лучшего найденного решения для задачи 140УД25АС1В

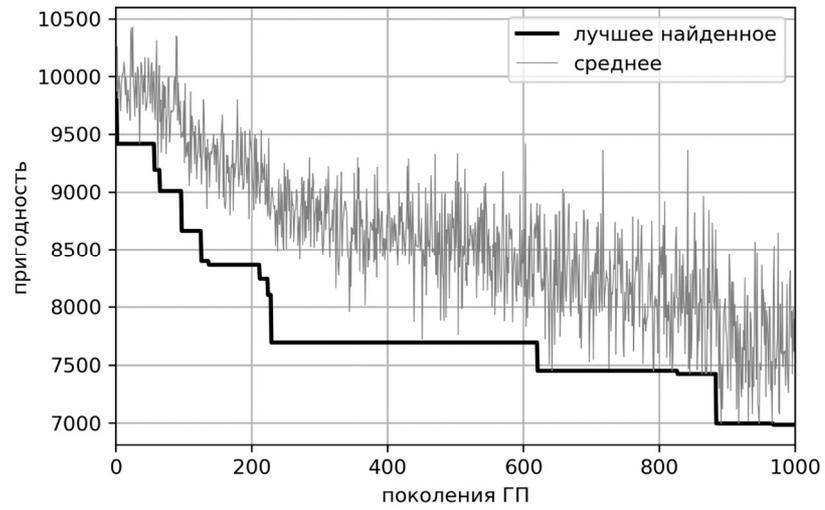


Рисунок 2.17 – График изменения пригодности ГП при решении задачи 140УД25АС1В

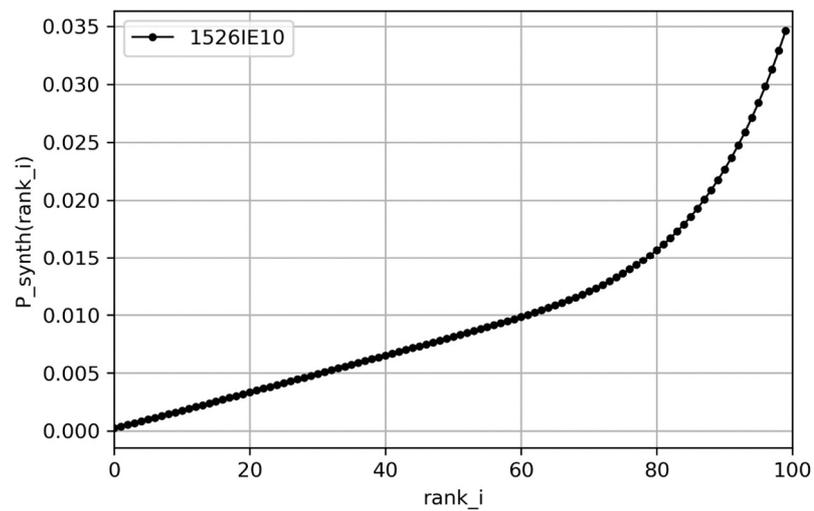


Рисунок 2.18 – График функции назначения вероятностей лучшего найденного решения для задачи 1526ИЕ10

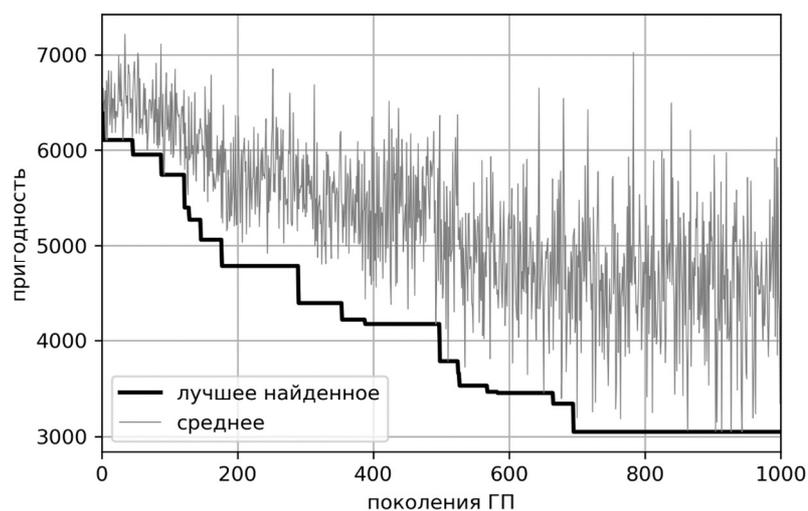


Рисунок 2.19 – График изменения пригодности ГП при решении задачи 1526IE10

Результаты решения задачи кластеризации были сравнены с решением полученным стандартным ГА с линейной ранговой селекцией (настройки аналогичны ГА с синтезированным оператором) и с результатом применения алгоритма 4 со случайной равномерной инициализацией центров кластеров (статистика собрана по 40 прогонам, как и в случае с ГА). Результаты численных экспериментов представлены в таблицах 2.9 и 2.10, на рисунках 2.20 и 2.21 показаны диаграммы размаха.

Таблица 2.9 – Результаты численных экспериментов для кластеризации 140УД25АС1В

Оценка	ГА с синтезированной селекцией	Стандартный ГА	ALA-алгоритм со случайной инициализацией
Лучшее	6758.66	7968.78	13534.32
Медиана	8088.69	9467.09	21587.31
Худшее	9852.17	13947.65	14891.72
Среднее	8064.35	9624.56	15048.25
Стандартное отклонение	588.57	1171.29	1318.86
Ошибка классификации	7.14%	16.30%	32.10%

Таблица 2.10 – Результаты численных экспериментов для кластеризации 1526IE10

Оценка	ГА с синтезированной селекцией	Стандартный ГА	ALA-алгоритм со случайной инициализацией
Лучшее	3050.7	3872.28	7349.74
Медиана	4872.59	5826.83	11980.54
Худшее	7211.01	9051.11	10522.45
Среднее	4937.55	5935.89	10488.22
Стандартное отклонение	920.53	1221.75	766.26
Ошибка классификации	11.21%	22.73%	38.74%

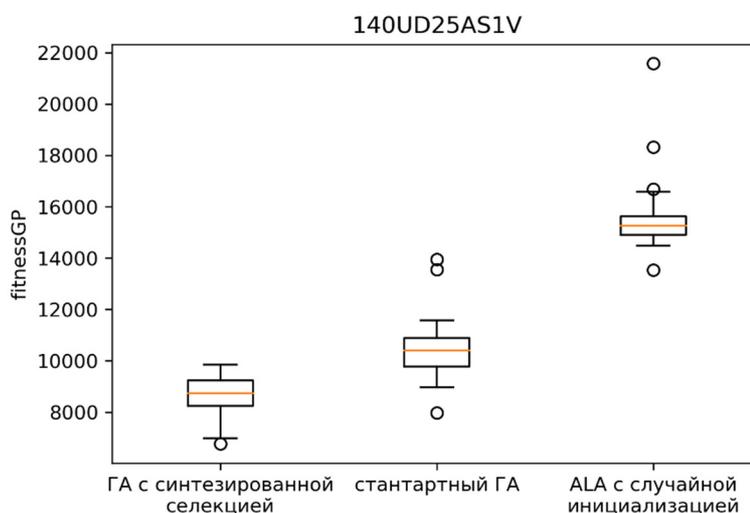


Рисунок 2.20 – Диаграмма размаха результатов численных экспериментов для кластеризации 140УД25АС1В

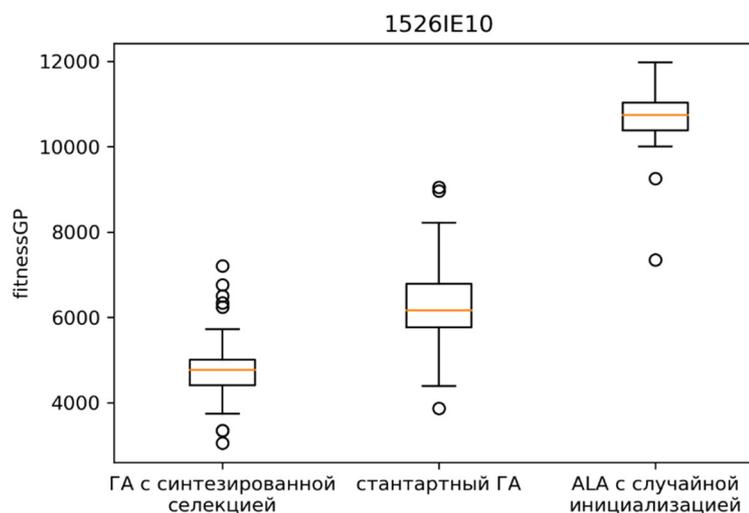


Рисунок 2.21 – Диаграмма размаха результатов численных экспериментов для кластеризации 1526IE10

Как видно из результатов, оба ГА превосходят способ со случайной инициализацией, синтезированный оператор превосходит ГА со стандартной эвристикой селекции как по лучшему найденному решению, так и по средней пригодности. Тест Манна-Уитни-Уилкоксона подтвердил статистическую значимость в результатах численных экспериментов.

Полученные результаты были сравнены с результатами ранее полученными при использовании других алгоритмов. Для микросхемы 140УД25АС1В лучшее найденное решение в [176] имеет значение по критерию (2.23) равное 7291.67 против 6758.66, полученного с помощью синтезированного оператора селекции, т.е. новое решение превосходит ранее полученное на 7.3%. Для микросхемы 1526IE10 полученное в [177] решение имеет пригодность 3440.1, против 3050.70, полученного с помощью синтезированного оператора, т.е. новое решение превосходит ранее полученное на 11.3%.

Таким образом, результаты апробации метода 1 при решении прикладной задачи, подтвердили, что с помощью конструктивной офлайн гиперэвристики на основе метода генетического программирования можно одинаково эффективно синтезировать операторы селекции как для класса задач на основе репрезентативного множества эталонных тестовых задач, так и для прикладных

задач глобальной оптимизации с алгоритмически заданными целевыми функциями.

Выводы по главе 2

В Главе 2 метод генетического программирования рассмотрен как гиперэвристика, позволяющая формировать новые элементарные эвристики для ЭА. Было показано, что оператор селекции можно представить, как функцию назначения вероятностей отбора индивидов для последующих действий с ними. Стандартные эвристики отбора предлагают всего несколько вариантов реализации данной функции назначения вероятностей, которые сводятся к монотонным линейно или экспоненциально возрастающим функциям (за исключением пороговой функции). С другой стороны, в качестве функции назначения вероятностей можно использовать любой другой вид зависимости, если новый оператор селекции обеспечит большую эффективность решения поставленной задачи оптимизации.

В данном исследовании предложена конструктивная офлайн гиперэвристика синтеза оператора селекции на основе метода ГП (метод 1), в которой задача синтеза оператора селекции в ЭА сформулирована как задача символьной регрессии. Поиск функции назначения вероятностей в методе 1 может осуществляться как для множества задач оптимизации (класса задач), так и для частной задачи. В первом случае необходимо обеспечить обобщающие свойства решения за счет использования кросс-валидации на обучающем множестве задач оптимизации.

Экспериментальное исследование предложенного метода на множестве эталонных задач глобальной однокритериальной оптимизации с алгоритмически заданными функциями показало, что метод 1 позволяет синтезировать новые эвристики отбора в ЭА. Полученные функции назначения вероятностей, демонстрируют высокую эффективность в среднем на множестве эталонных

тестовых задач ВВОВ по критерию вероятности нахождения глобального оптимума с заданной точностью. Более того, они превосходят оценку эффективности случайного выбора одного из базовых операторов селекции, а, следовательно, более предпочтительны в случае задач с алгоритмически заданными целевыми функциями, для которых заранее не известно, какой из базовых операторов продемонстрирует большую эффективность. Задачу синтеза эвристики отбора можно сформулировать с учетом других требований к свойствам ЭА. В данном исследовании было показано, что метод 1 способен синтезировать эвристики, позволяющие найти глобальный оптимум за меньшее число поколений ЭА.

Решение прикладной задачи проектирования эвристик отбора для ЭА в задаче идентификации партий электронных компонентов аэрокосмического применения показало, что метод 1 позволяет синтезировать оператор селекции, который обеспечивает большую эффективность решения задач идентификации партий микросхем 140УД25АС1В и 1526ИЕ10 по критериям суммы расстояний от объектов до центров кластеров и ошибки классификации по сравнению со стандартным ГА с линейной ранговой селекцией и методом k -медиан со случайной равномерной инициализацией кластеров. Применение синтезированной эвристики позволяет улучшить решения, полученные ранее другими алгоритмами.

Не все операторы ЭА могут быть представлены в виде дерева метода ГП как суперпозиция элементарных операций, т.к. являются более сложными алгоритмами, вовлекающими циклы, условия, разыгрывания вероятностей и другие операции и действия, связанные с решаемой задачей или особенностями ЭА. Поэтому помимо конструирования эвристик из элементарных операций, эффективная метаэвристика ЭА может быть сформирована путем выбора и комбинации готовых эвристик и метаэвристик. В частности, селективные эвристики могут быть использованы для классов задач, обладающих свойствами нестационарности.

ГЛАВА 3. СЕЛЕКТИВНАЯ ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ НЕСТАЦИОНАРНОЙ ОПТИМИЗАЦИИ

Глава 3 посвящена проблеме разработки и исследования гиперэвристик для решения задач нестационарной оптимизации. Нестационарность является одним из основных свойств сложных систем и является следствием динамической природы системы, изменений внешней среды, изменений целей системного анализа и управления, недостатком информации о системе и т.д. Можно выделить несколько основных типов изменений целевой функции, для каждого типа предложены эффективные эвристики. В практических задачах оптимизации тип (или комбинация нескольких типов) и характер изменений, а также моменты времени, когда происходят изменения заранее неизвестны. Поэтому требуется синтезировать алгоритм оптимизации непосредственно в процессе решения задачи. Предложенная в главе 3 селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации осуществляет контроль изменений в задаче оптимизации и выбор и комбинацию базовых эвристик из predetermined множества. Численные исследования показывают, что гиперэвристика всегда превосходит по эффективности среднюю эффективность ЭА, использующих базовые эвристики, а с увеличением сложности задачи превосходит и ЭА с лучшей базовой эвристикой. При апробации подхода на практической задаче идентификации параметров боковой динамики движения малого беспилотного летательного аппарата с помощью нейросетевой модели, удалось получить меньшее среднее значение ошибки идентификации параметров по сравнению с алгоритмом обратного распространения ошибки и ЭА, использующими одну базовую эвристику нестационарной оптимизации.

3.1 Проблема нестационарности в задачах оптимизации сложных систем и обзор подходов к ее решению

Как отмечалось ранее, нестационарность, связанная с внутренними и внешними изменениями является одним из основных свойств сложной системы. Изменения в среде оказывают влияние на ландшафт целевой функции. В следствие изменений целевой функции могут изменяться значения и положения глобального и локальных оптимумов. Подобное явление называется дрейфом экстремума.

В последние годы подобные задачи оптимизации часто называют динамическими (англ., DOP, *dynamic optimization problem*), нестационарными (англ., *non-stationary*) или задачами в условиях меняющейся (нестационарной, динамической) среды (англ., *optimization in changing (non-stationary, dynamic) environment*) [26, 178, 179]. В данном исследовании будем использовать термин «нестационарная оптимизация», чтобы избежать путаницы с терминами теории динамического программирования.

В классической теории оптимизации существует раздел математического программирования – стохастическое программирование, который позволяет учитывать неопределённость в оптимизационных моделях [180, 181]. Однако методы стохастического программирования не всегда применимы в сложных практических задачах, как минимум потому, что в сложных задачах оптимизации целевая функция (функции) и критерии не заданы аналитически. Более того, стохастическое программирование подразумевает хотя бы теоретическую возможность оценки законов распределения для неопределенных параметров оптимизационной модели (случай стохастической, вероятностной неопределенности). На практике же характер нестационарности заранее не известен и часто не является стохастическим. Поэтому для решения сложных задач нестационарной оптимизации предпочтительными, а зачастую и единственно применимыми, являются методы поисковой оптимизации.

Стационарные (или статические) задачи оптимизации (1.1)-(1.2) можно рассматривать как частный случай нестационарных задач (1.3)-(1.4) для

фиксированного момента времени. Основная задача алгоритмов стационарной поисковой оптимизации заключается в нахождении глобального оптимума, поэтому критериями эффективности алгоритма обычно являются точность и скорость (в терминах вычислительных затрат) нахождения решения.

В нестационарных задачах значение и положение оптимума может меняться с течением времени, следовательно, алгоритм должен иметь возможность отслеживать изменения и адаптироваться к новым условиям. Критериями эффективности алгоритма в данном случае могут выступать точность и скорость адаптации к изменениям. Традиционные подходы прямого поиска, включая ЭА, не обладают необходимыми свойствами, обеспечивающими адаптацию к изменениям среды, и имеют тенденцию сходиться к найденному оптимуму, теряя накопленную на прошлых этапах поиска информацию о пространстве поиска.

Первые известные работы, в которых упоминается о применении ЭА для задач нестационарной оптимизации: Фогель в 1966 [49] и Голдберг в 1987 [182]. Однако активный интерес к данной теме появился лишь в 2000-е, на эти годы приходится основной пик работ и публикаций. Среди исследователей, сделавших наибольший вклад стоит отметить Бранке, Викера, Морисона, Янга, а также классиков в области эволюционных алгоритмов Де Йонга и Деба [178, 183-187]. Проблеме нестационарной оптимизации посвящены специальные сессии ведущих международных конференций: с 1999 года EvoDOP на GECCO, с 2004 года EvoSTOC на EvoStar и ECiDUE на IEEE CEC. Начиная с 2009 в рамках конференции IEEE CEC проводились специальные конкурсы по решению задач нестационарной оптимизации. В настоящее время существует специальный информационный ресурс, www.dynamic-optimization.org, посвященный проблеме нестационарной оптимизации, который создан и поддерживается исследовательской группой департамента информационных технологий и искусственного интеллекта университета Гранады, Испания.

В работе [26] известный специалист в области нестационарной оптимизации Шенгшиян Янг назвал следующие вопросы, на которые должен ответить исследователь при разработке алгоритма оптимизации:

– зависит ли следующее состояние задачи оптимизации от текущего и предыдущих. Это свойство задач называется «зависимость от времени» (*time-linkage*);

– можно ли идентифицировать изменения в задаче оптимизации. «Наблюдаемость» (*visibility*);

– можно ли предсказать изменения. «Предсказуемость» (*predictability*);

– являются ли изменения циклическими. «Цикличность» (*cyclicity*);

– какие компоненты задачи меняются: целевая функция, допустимая область, ограничения или другие параметры.

При этом Шенгшиян Янг отмечает, что для большинства практических задач нестационарной оптимизации не проявляется зависимость от времени, меняются не только положение глобального оптимума, но топология целевой функции, изменения непредсказуемые, изменения часто бывают циклическими.

Анализ практических задач нестационарной оптимизации позволяет выявить основные типы изменений и их свойства. Обычно исследователи выделяют следующие типы [185]: координатные изменения (*coordinate transformations*), масштабирование (*rescalings*) и изменение зон притяжения оптимумов (*stretching*).

Координатные изменения приводят к изменению положений оптимумов, при этом значения в точках и взаимное положение оптимумов сохраняются. Пример координатных изменений показан на рисунке 3.1. Возможны следующие варианты изменений координат:

– *дрейф* – смещение всего графика целевой функции в некотором направлении. Дрейф может иметь постоянную, переменную (в том числе случайную) и хаотичную скорость изменений, которая определяется расстоянием в поисковом пространстве, на которое перемещается график функции;

– *вращение координат* – график функции вращается относительно некоторой точки, что приводит к цикличности изменений. Скорость вращения может быть постоянной, переменной и хаотичной;

– *хаотичное изменение координат*.

Масштабирование – это изменение значений целевой функции при сохранении общей структуры оптимумов. Значения могут как увеличиваться (растяжение), так и уменьшаться (сжатие). Скорость изменений может быть постоянной, переменной и хаотичной. Пример координатных изменений показан на рисунке 3.2.

Изменения зон притяжения оптимумов не меняют положения оптимумов и значения в точках оптимумов, но меняют топологию целевой функции. Пример координатных изменений показан на рисунке 3.3.

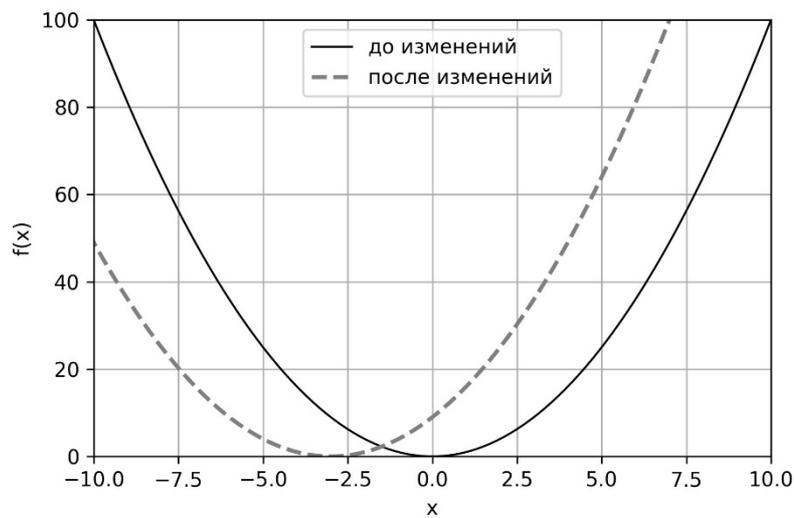


Рисунок 3.1 – Пример координатных изменений целевой функции

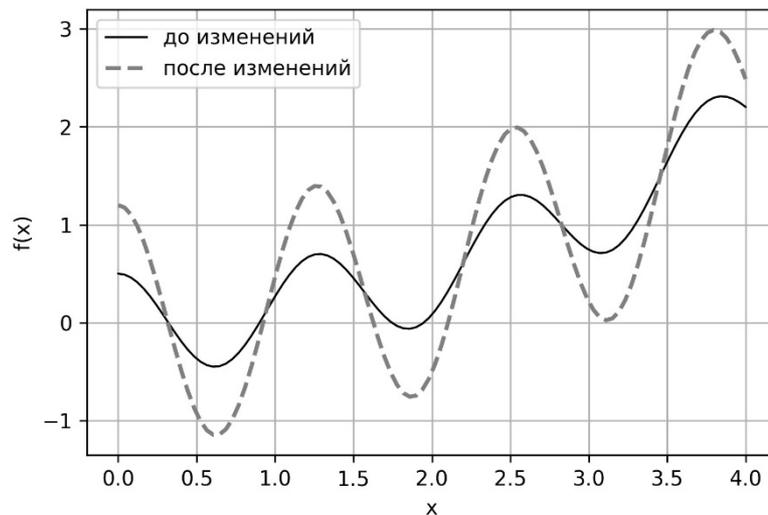


Рисунок 3.2 – Пример масштабирования целевой функции

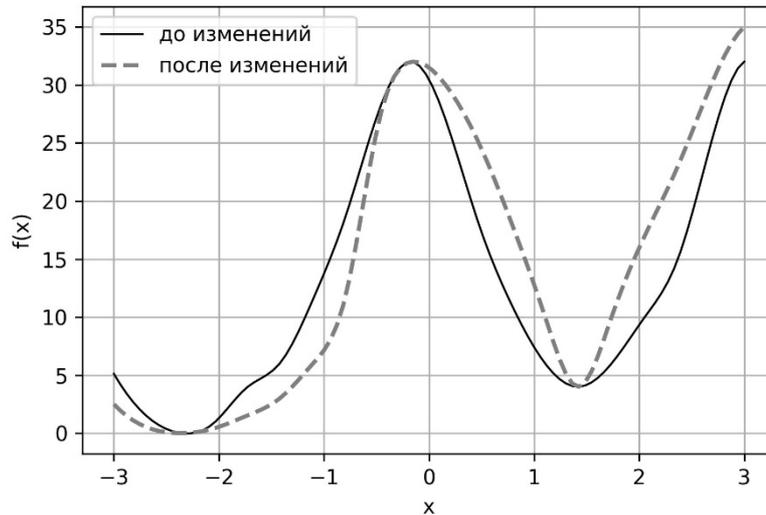


Рисунок 3.3 – Пример изменения зон притяжения оптимумов

Стоит отметить, что задача нестационарной оптимизации в постановке (1.3) – (1.4) практически не рассматривается в теории математического программирования. Во-первых, алгоритмическое задание целевой функции затрудняет аналитическое решение или выбор эффективного подхода, во-вторых, методы математического программирования редко имеют механизмы адаптации к изменениям задачи в процессе ее решения. В то же время вполне очевидной кажется идея, что отслеживание положения оптимума будет осуществляться более эффективно, если в поиске параллельно участвуют несколько решений. Поэтому подавляющее большинство исследователей при решении нестационарных задач оптимизации используют популяционные алгоритмы, среди которых можно выделить:

- эволюционные (включая генетические) алгоритмы [178, 186, 188-197];
- эволюционные стратегии и эволюционное программирование (*evolutionary strategies, evolutionary programming*) [198-203];
- алгоритмы оценки распределений (*estimation-of-distribution algorithms, EDA*) [207-209];

- стайные алгоритмы (*swarm intelligence*, PSO) [204-206];
- иммунные алгоритмы (*immune-based algorithms*) [210, 211];
- алгоритмы муравьиных колоний (*ant colony optimization*) [212, 213];
- самоорганизующиеся разведчики (*self-organizing scouts*) [214, 215] и др.

Из обзора информационных источников можно сделать вывод, что наибольшее количество исследований и наилучшие результаты на сегодня достигнуты в классе ЭА, другие подходы зачастую имеют узкое применение для частных прикладных задач [185, 216].

Традиционные поисковые алгоритмы не обладают необходимыми свойствами, обеспечивающими адаптацию к изменениям среды, поэтому были предложены различные эвристики для решения задач нестационарной оптимизации. Некоторые из эвристик разрабатывались независимо от алгоритмов оптимизации и могут использоваться с любым из них, другие являются модификацией базовых эвристик ЭА. Известны следующие эвристики нестационарной оптимизации:

- перезапуск процедуры поиска,
- локальный поиск для адаптации к изменениям,
- механизмы памяти,
- механизмы поддержания разнообразия,
- мультипопуляционный подход,
- адаптация и самоадаптация,
- алгоритмы с пересекающимися поколениями,
- моделирование динамики и прогноз изменений.

Перезапуск процедуры поиска. Данный подход является наиболее простым. В случае, если обнаружены изменения в задаче оптимизации, происходит перезапуск процедуры поиска – результаты прошлых итераций забываются, популяция инициализируется заново. Существуют модификации, когда при инициализации в популяцию добавляются несколько индивидов с прошлого запуска для обеспечения сходимости [217].

Преимущество подхода – нечувствительность к типам и масштабам изменений, т.к. оптимизационная процедура запускается каждый раз заново. Недостатки – весьма затратный способ, т.к. нет локальной адаптации к небольшим изменениям и цикличности, а в случае частых изменений алгоритм может не успевать сходиться в точку глобального оптимума [218].

Локальный поиск для адаптации к изменениям. В случае относительно небольших изменений целесообразно использовать операторы локального поиска в некоторой области относительно текущих индивидов. ЭА не обладают механизмами локального поиска, поэтому были предложены различные модификации базовых операторов. В частности, в [219] используется механизм гипермутации, где в случае изменений задачи оптимизации вероятность мутации определяется как произведение обычной интенсивности мутации и параметра гипермутации. Другой пример – изменяющийся локальный поиск (*variable local search*, VLS) для ГА, предложенный в [218]. Используется кодирование индивидов, при котором младшие биты определяют меньшее изменение фенотипа. Оператор мутации применяется к некоторому количеству первых (младших) битов. Число битов определяется в процессе поиска. В «мультинациональном» ГА (*multinational GA*), предложенном в [197, 220], интенсивность мутации определяется близостью решений к некоторой субпопуляции (называемой «нацией»), чем ближе индивид к своим представителям, тем ниже вероятность мутации. Более традиционный подход – использование внешней по отношению к ЭА процедуры локального поиска.

Преимущество подхода – в случае небольших координатных изменений с сохранением структуры целевой функции, удается быстро локализовать новые положения оптимумов без существенной потери информации, накопленной на прошлых шагах. Недостатки – низкая эффективность при сильных, скачкообразных изменениях и изменениях структуры оптимумов.

Механизмы памяти. Данный подход позволяет использовать прошлый опыт, в случаях, когда в результате изменений в задаче оптимизации ландшафт целевой функции совпадает или похож на ранее встречавшийся ландшафт. Запоминание

эффективных в прошлом решений может направить эволюционный поиск в перспективные регионы поискового пространства. Существует две концепции организации механизма памяти в ЭА: явная и неявная память.

Неявная память базируется на аналогии с биологическим явлением – парностью или диплоидностью хромосом. В диплоидном ЭА хромосома состоит из 2 наборов генов, каждый из которых однозначно кодирует решение задачи. Гены могут быть доминантными или рецессивными. Для оценки качества решения используется доминантный набор. В случае изменений в задаче оптимизации, определяется доминантный уровень каждого набора. При необходимости доминантный и рецессивный наборы меняются местами. Существуют модификации с триплоидным и более (мультиплоидным) набором хромосом. Таким образом в алгоритм вводится избыточный генетический материал, содержащий потенциально полезный прошлый опыт.

Диплоидный механизм впервые использовал Голдберг [182], дальнейшие работы были посвящены способам определения и представления доминантных наборов (например, на основе таблиц генов или масок, определяющих соответствие между генотипом и фенотипом) [221-223].

В случае использования явной памяти, создается специальное хранилище прошлых решений (архив), которые могут быть повторно оценены и при необходимости возвращены в популяцию. Существуют разные способы организации хранилищ. Наиболее часто используют внешнее хранилище, содержащее либо прошлые решения (прямое запоминание, *direct memory*), либо какую-то другую полезную информацию о прошлом опыте (ассоциативная память) [187, 224, 225]. Другой вариант – использование параллельных популяций, которые хранят информацию о различных ситуациях [226]. Работы в этом направлении различаются способами формирования, обновления и использования явной памяти.

Преимущества подхода – механизмы памяти хороши в случае циклических изменений, а также для контроля сходимости путем поддержания генетического разнообразия. Недостатки – эффективны только в случае, когда среда возвращается в первоначальное состояние. Если число уникальных состояний велико, требуется

хранить большой объем информации. На практике, число повторяющихся состояний априори не известно.

Механизмы поддержания разнообразия. Практически все подходы в области нестационарной эволюционной оптимизации так или иначе ориентированы на поддержание генетического разнообразия популяции. Тем не менее, можно выделить, по крайней мере, три группы специальных методов.

Первая группа – это методы прямого повышения разнообразия. Наиболее простой подход связан с добавлением на каждой итерации в популяцию случайного индивида (в литературе встречается как метод случайных иммигрантов, частичная гипермутация, гипермутация с фиксированным значением и др.) [227, 228]. Известные методы в основном различаются способом формирования случайного решения.

Вторая группа – методы с разделяющейся пригодностью, основанные на использовании понятия ниш (*niche*). Общая идея всех реализаций алгоритма – уменьшение значения пригодности решения, если в некоторой области данного решения (нише с заданным радиусом) присутствуют другие решения [197, 209].

Третья группа – методы ограничения спаривания (*restricted mating*). Популяция некоторым образом разбивается на подпопуляции, в рамках одной подпопуляции рекомбинация запрещается. Более простая реализация может ограничивать рекомбинацию с учетом расстояния – более близкие (похожие друг на друга) индивиды не участвуют в рекомбинации [230, 231].

Преимущества подхода – высокая эффективность при сильных изменениях задачи оптимизации. Могут быть эффективны в задачах с редкими изменениями, т.к. у алгоритма будет достаточно времени сойтись в новый оптимум. Недостатки – медленная сходимость, низкая эффективности при слабых изменениях.

Мультипопуляционный подход. Исходная популяция разбивается на конкурирующие подпопуляции, каждая подпопуляция может отвечать за отдельный регион пространства поиска или может иметь собственную цель. Подпопуляции могут взаимодействовать для корректировки поиска. Данный

подход можно рассматривать как комбинацию поддержки разнообразия, механизма памяти и адаптации.

Часто встречается алгоритм с двумя популяциями, где одна отвечает за поиск глобального оптимума, другая – отслеживает изменения [232]. Конкретные реализации подхода зависят от того, какие задачи ставятся перед популяцией и как происходит разбиение популяций [197, 233-236]. Наиболее часто для формирования подпопуляций применяются методы кластеризации – некоторое решение принимается за центр кластера, решения, образующие кластер, формируют подпопуляцию [232, 237, 238].

Преимущества подхода – обеспечивается достаточное генетическое разнообразие, в популяциях сохраняется прошлый опыт, которым можно повторно воспользоваться. Разные популяции могут отслеживать разные оптимумы. Недостатки – большое количество популяций замедляет поиск. Задачи выбора числа популяций, их размеров и областей поиска сложны сами по себе.

Методы самоадаптации. Различные идеи самоадаптации успешно реализованы в ЭА для стационарных задач. Эти же подходы могут быть применены и в нестационарной оптимизации, т.к. адаптация связана с обучением на прошлом опыте. Термин «самоадаптация» подразумевает кодирование управляющих параметров алгоритма в хромосоме вместе с представлением решения [83-86, 239], таким образом осуществляется одновременно поиск решения исходной задачи и настройка алгоритма под текущую ситуацию.

В частности, в [228] предложен ГА с самонастраиваемой мутацией (*GA with genetic mutation rate*), в котором параметры операции мутации закодированы в хромосоме и определяются с учетом пригодности индивида. Алгоритм способен увеличивать интенсивность мутации в случае изменений в задаче. Аналогичный подход реализован в многонациональном ГА (*multinational genetic algorithm*) [197], в котором в хромосому включаются 5 параметров алгоритма: вероятность мутации, вероятность рекомбинации, параметр селекции, коэффициент дисперсии и расстояния для мутации).

Преимущества подхода – нет необходимости настраивать параметры алгоритма, высокая эффективность на относительно простых задачах. Недостатки – предложенные подходы плохо справляются со сложными задачами нестационарной оптимизации.

Алгоритмы с пересекающимися поколениями. Стандартный генетический алгоритм на каждой итерации (поколении) формирует новую популяцию, теряя предыдущих индивидов, но сохраняя опыт прошлых поколений неявно в генах индивидов. Ранее упоминались подходы, в которых для сохранения опыта применялись специальные внешние механизмы памяти, различные популяции (субпопуляции) или методы контроля разнообразия решений. Другим подходом является отказ от итерационной модели (с меняющимися поколениями), такой алгоритм называется ЭА со стационарными состояниями. Наиболее известный пример – стационарный ГА (*steady-state GA*, SSGA). В SSGA, в отличие от итерационного ГА, на каждой итерации только 2 родителя участвуют в отборе и рекомбинации. Потомки после применения оператора мутации замещают отобранную пару родителей (либо лучший потомок замещает худшего родителя) [240]. Впервые стационарный ГА для задач нестационарной оптимизации был применен в [2341]. Дальнейшие работы были посвящены различным стратегиям замещения [242-244]. В частности, хорошие результаты дает бинарный турнир между старейшим и случайно выбранным решениями. Другой вариант – замена худшего из похожих. Также используются различные адаптивные подходы.

Преимущества подхода – эффективность алгоритма не зависит от типа изменений. Недостатки – эффективность зависит от выбранной стратегии замещения.

Моделирование динамики и прогноз изменений. Во многих случаях будущие изменения задачи оптимизации могут быть предсказаны на основе информации о прошлых изменениях и текущей ситуации. Разумно использовать эту информацию и корректировать поиск согласно прогноза. Повторяющиеся и циклические изменения могут неявно прогнозироваться алгоритмами с механизмами памяти.

Однако для произвольных изменений требуется разработка внешних процедур, способных обучаться на исторических данных.

В [245] информация о прошлых положениях оптимума используется для построения авторегрессионных моделей. В [201] для прогноза используются фильтры Кальмана. Новое положение оптимума прогнозируется в случае обнаружения изменений. По результатам прогноза либо корректируется оператор мутации для концентрации поиска в области спрогнозированного решения, либо поощряются решения близкие к спрогнозированному решению, либо генерируются новые решения в области спрогнозированного решения.

В [246, 247] используется подход, прогнозирующий 2 фактора. Во-первых, используется регрессионная модель для оценки поколения, когда произойдут изменения. Во-вторых, используется Марковская модель для оценки состояния, которое появится.

Существуют и другие частные модели прогноза, учитывающие специфику решаемой проблемы [208, 248].

Преимущества подхода – в случае успешного прогноза, задача вырождается в серию стационарных задач оптимизации. Недостатки – недостаточный объем данных для обучения и прогноза и обучение на неверных данных (если алгоритм был недостаточно эффективен на прошлых этапах поиска), не все типы изменений можно предсказывать. Главный недостаток заключается в том, что данный подход лежит за пределами рассматриваемого направления – эволюционных алгоритмов. Выбор, настройка, обучение и применение методов анализа динамики и прогноза изменений – задачи машинного обучения, сравнимые по сложности с исходной задачей оптимизации.

Как видно из обзора, среди известных подходов не удастся выделить универсальный, который был бы эффективен на различных классах задач нестационарной оптимизации. Более того, даже если тип и характер изменений известен заранее, то и в этом случае требуется выбор конкретного алгоритма и его тонкая настройка. Как упоминалось ранее, реальные задачи нестационарной

оптимизации включают комбинацию типов изменений, которые плохо прогнозируются.

3.2 Селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации

Гиперэвристика для решения задачи (1.11) должна выбирать и комбинировать известные эвристики нестационарной оптимизации. В области машинного обучения существует известный подход, называемый портфолио алгоритмов (*algorithm portfolio*), который был первоначально предложен для выбора стратегий на финансовых рынках [249], а позже стал применяться для выбора алгоритмов решения вычислительно сложных задач [250], в [251] метод был применен для выбора алгоритмов прямого поиска при решении задач вещественной параметрической оптимизации с алгоритмически заданными функциями. Основная идея метода портфолио алгоритмов заключается в оценке эффективности алгоритмов в зависимости от входных данных решаемой задачи. Пользователь метода должен определить критерий эффективности и стратегию выбора. При этом выбор алгоритма может осуществляться один раз (офлайн версия подхода) или по расписанию в процессе решения задачи в зависимости от оценки текущей ситуации процесса решения (онлайн версия).

Критерий эффективности портфолио для задачи нестационарной оптимизации должен отражать эффективность решения задачи (1.3)-(1.4). Наиболее используемой сегодня мерой в данной области является офлайн ошибка (*offline error*) (3.1). Однако для ее вычисления необходимо знать истинное значение глобального оптимума для каждого состояния среды, поэтому она применима только для модельных задач. В данной работе будем использовать модифицированную офлайн ошибку (*modified offline error*) (3.2):

$$OE = \frac{1}{T} \sum_{t=1}^T |f(x^*, t) - f(x_{bestFound}, t)|, \quad (3.1)$$

$$mOE = \frac{1}{T} \sum_{t=1}^T f(x_{bestFound}, t), \quad (3.2)$$

где OE – значение офлайн ошибки, mOE – значение модифицированной офлайн ошибки, x^* – глобальный оптимум задачи на шаге t , $x_{bestFound}$ – лучшее найденное значение на шаге t , T – период, для которого вычисляются ошибки.

Стратегия выбора эвристики должна обеспечить эффективное решение задачи (1.11), поэтому для предотвращения жадного (локального) поведения гиперэвристики будем использовать вероятностный выбор. В свою очередь вероятности выбора конкретной эвристики должны адаптироваться в процессе появления изменений в среде, уменьшаясь для менее эффективных на текущем этапе эвристик в пользу более эффективных. Подобный подход в ЭА называется метод адаптации вероятностей на уровне популяции (*Population-Level Dynamic Probabilities*, PDP) [252].

В случае, если вероятности будут принимать ненулевые значения, на каждой итерации будет использоваться не единственный ЭА, выбранный из предопределенного множества, а будет синтезироваться новый ЭА в виде комбинации различных алгоритмов. Согласно классификации Бёрка [95], предложенный подход можно отнести к классу селективных онлайн гиперэвристик.

Определим множество базовых эвристик нестационарной оптимизации, которые будут выбираться и комбинироваться гиперэвристикой. В данной работе для решения задач (1.3)-(1.4) в качестве ЭА используется бинарный ГА, который во-первых, как обсуждалось ранее, имеет ряд преимуществ перед ЭА, работающими только с вещественными переменными, а во-вторых, для бинарного ГА предложено больше базовых эвристик нестационарной оптимизации.

Обозначим множество базовых эвристик как $H = \{h_i\}$ ($i = \overline{1, |H|}$). Множество H содержит следующие эвристики, применяемые в области нестационарной оптимизации:

- перезапуск процедуры оптимизации,

– локальная адаптация к изменениям, реализованный в виде изменяющегося локального поиска (*variable local search*, VLS) [218],

– механизм явной памяти [188],

– механизм поддержания разнообразия на основе метода ниш [197],

– самонастраиваемый ЭА с управляемой мутацией [228].

Как обсуждалось ранее, каждая из эвристик имеет преимущества на определенных типах изменений среды.

Гиперэвристика синтеза ЭА нестационарной оптимизации на основе метода портфолио алгоритмов и метода адаптации вероятностей на уровне популяции имеет следующий вид:

Метод 2. Селективная онлайн гиперэвристика синтеза ЭА оптимизации в нестационарной среде.

Вход: множество H базовых эволюционных эвристик и метаэвристик, метод обнаружения изменений в среде, критерий оценки эффективности ЭА (3.4).

Инициализация: Случайная популяция решений делится на равные подпопуляции, подпопуляциям назначаются используемые эвристики.

Повторять, пока решается задача нестационарной оптимизации (цикл глобальной адаптации):

 Пересчитать вектор параметров глобальной адаптации, используя (3.5) – (3.11).

Повторять, пока не обнаружены изменения в среде (цикл локальной адаптации):

 Перераспределить размеры подпопуляций на основе векторов глобальной и локальной адаптации.

Повторять заданное число итераций (период локальной адаптации):

 Решать задачу оптимизации применяя выбранные эвристики к соответствующим подпопуляциями.

 Если обнаружены изменения, то конец цикла локальной адаптации.

 Пересчитать вектор параметров локальной адаптации.

Выход: множество лучших найденных решений на каждой итерации.

В методе 2 вероятности выбора эвристик не задаются в явном виде, а отражаются на распределении количества вычислений целевой функции, которые осуществляются каждой из эвристик. Для этого на каждом поколении популяция ЭА размера $PopSize$ разбивается на подпопуляции размера $subPop_i, i = \overline{1, |H|}$, где $PopSize = \sum_{i=1}^{|H|} subPop_i$. Каждой подпопуляции назначается своя эвристика,

размер подпопуляции выбирается пропорционально эффективности данной эвристики. Размер подпопуляции определяется путем оценки векторов параметров глобальной и локальной адаптации.

Вектор параметров глобальной адаптации v^{glob} (3.8) используется для оценки вероятности появления изменений конкретного типа. Поскольку идентификация типа (или комбинации типов) изменений среды является задачей сравнимой по сложности с исходной задачей нестационарной оптимизации, вектор параметров глобальной адаптации определяет вероятности применения конкретной эвристики. Вероятности использования эвристик, продемонстрировавших большую эффективность в предыдущем локальном цикле, должны увеличиваться. Пересчет значений параметров глобальной адаптации основан на модели изменяющихся вероятностей на уровне популяции (PDP), иногда применяемой для самоконфигурации ЭА.

Вектор параметров локальной адаптации v^{local} (3.9) определяет ранжирование эвристик по эффективности в локальном цикле адаптации до очередного изменения среды. Пул перераспределяемых ресурсов формируется путем вычитания из каждой подпопуляции Δ_{subPop} случайных индивидов. Значение Δ_{subPop} является параметром метода, при этом должно выполняться условие (3.3), которое будет гарантировать что даже наименее эффективная эвристика участвует в поиске решения и, если продемонстрирует высокую эффективность при следующих изменениях среды, размер ее подпопуляции будет увеличен (вырастет вероятность ее применения).

$$\Delta_{subPop}: subPop_i - \Delta_{subPop} \geq subPop_{min}, i = \overline{1, |H|}, \quad (3.3)$$

где Δ_{subPop} – параметр для перераспределения размеров подпопуляций $subPop_i, i = \overline{1, |H|}$, $subPop_{min}$ – минимальное значение размера подпопуляции.

Оценка эффективности для каждой из эвристик в одном локальном цикле определяется с помощью модифицированной офлайн ошибки (3.4), которая минимизируется:

$$mOE^{glob}(h_i) = \frac{1}{T_c} \sum_{t=1}^{T_c} f(x_{bestFound}(h_i), t), \quad (3.4)$$

где mOE^{glob} – оценка эффективности для эвристики h_i , T_c – число поколений между двумя изменениями в среде, c – номер локального цикла ($c = 1, 2, \dots$), f – значение целевой функции для лучшего найденного эвристикой h_i индивида $x_{bestFound}(h_i)$ на момент времени t .

Для вычисления коэффициентов глобальной $v_i^{glob}(t, c)$ и локальной $v_i^{local}(t)$ адаптации эвристики ранжируются по значениям mOE^{glob} и по $f(x_{bestFound}(h_i), t)$ соответственно:

$$rank_i^{glob} \leq rank_j^{glob}, \text{ если } mOE^{glob}(h_i) \leq mOE^{glob}(h_j), \quad (3.5)$$

$$rank_i^{local} \leq rank_j^{local}, \text{ если } f(x_{bestFound}(h_i), t) \leq f(x_{bestFound}(h_j), t), \quad (3.6)$$

где $rank_i^{glob}, rank_i^{local} \in [1, |H|], i = \overline{1, |H|}$.

При инициализации параметр глобальной адаптации и распределение размеров подпопуляций заполняются равными значениями (3.7) и (3.10). На очередном локальном цикле адаптации, глобальный параметр пересчитывается как линейная комбинация предыдущего и нового значений, где новое значение вычисляется, используя распределение, пропорциональное рангам глобальной адаптации (3.8).

$$v_i^{glob}(0,0) = \frac{1}{|H|}, i = \overline{1, |H|}, \quad (3.7)$$

$$v_i^{glob}(t, c + 1) = (1 - \eta) \cdot v_i^{glob}(t, c) + \eta \cdot \frac{2 \cdot (|H| - rank_i^{glob} + 1)}{|H| \cdot (|H| + 1)}, \quad (3.8)$$

$$v_i^{local}(t) = rank_i^{local}, \quad (3.9)$$

где $\eta \in [0, 1]$ – параметр, определяющий скорость глобальной адаптации (по умолчанию $\eta = 0.5$).

При вычислений новых значений размеров подпопуляций из каждой подпопуляции вычитается Δ_{subPop} случайных индивидов, суммарный пул индивидов распределяется с учетом значения параметра локальной адаптации

(пропорционально рангам, вычисленным в локальном периоде) для поощрения эффективных эвристик в рамках текущего состояния среды и с учетом значения глобального параметра, поощряя эвристики, которые были эффективны в прошлых периодах, для прогноза новых изменений в среде (3.11).

$$subPop_i(0) = \frac{PopSize}{|H|}, \quad (3.10)$$

$$subPop_i(t+1) = subPop_i(t) - \Delta_{subPop} + \frac{\Delta_{subPop} \cdot |H|}{2} \cdot \left[\frac{2 \cdot (|H| - v_i^{local}(t) + 1)}{|H| \cdot (|H| + 1)} + v_i^{glob}(t) \right]. \quad (3.11)$$

После определения новых размеров подпопуляций происходит перераспределение индивидов путем случайных миграций. Традиционный подход «лучший замещает худшего» менее эффективен, т.к. приводит к преждевременной сходимости и потере разнообразия популяции.

Контроль изменений в среде в методе 2 осуществляется путем повторного вычисления пригодности текущего лучшего найденного решения (в литературе такое решение называется детектором). Если значение целевой функции в данной точке изменилось, регистрируется изменение в среде и начинается новый период локальной адаптации.

3.3 Экспериментальное исследование предложенного метода

Для исследования эффективности метода 2 были выбраны два набора эталонных тестовых задач нестационарной оптимизации. Первый набор включает две наиболее часто используемых для оценки ЭА модельных задачи (используется компьютерная симуляция): динамическая функция Растригина и «задача подвижных пиков» (*the moving peaks benchmark*, МРВ). Динамическая функция Растригина (3.12) построена на основе обычной функции Растригина путем добавления случайных равномерно распределенных смещений по каждой из координат [253]. Задача МРВ (3.13) – (3.16) предложенная Бранке [254]

представляет собой многомерную функцию с предопределенным числом оптимумов (пиков), характеристики которых (ширина, высота и положение) меняются в процессе решения задачи. Число пиков и характеристики задачи MPB называются сценариями, исследователи используют 3 сценария с 5, 10 и 50 пиками.

$$f_{Rastrigin}(x, t) = \sum_{i=1}^n ((x_i + \Delta(t))^2 - 10 \cdot \cos(2\pi x_i + 2\pi \Delta(t)) + 10), \quad (3.12)$$

где n – размерность задачи, $x_i \in [-5.12, 5.12]$, $i = \overline{1, n}$ – переменные задачи, $\Delta(t)$ – случайное число, распределенное по равномерному закону в интервале $[-0.8, 0.8]$, t – параметр времени.

$$f_{MPB}(x, t) = \max_{j=1, PeakNum} \frac{H_i(t)}{1 + W_i(t) \cdot \sum_{i=1}^n (x_i - X_{ij}(t))^2}, \quad (3.13)$$

$$H_i(t) = H_i(t - 1) + h_{sev} \cdot \sigma, \sigma \in N(0,1), \quad (3.14)$$

$$W_i(t) = W_i(t - 1) + w_{sev} \cdot \sigma, \sigma \in N(0,1), \quad (3.15)$$

$$\bar{X}_i(t) = \bar{X}_i(t - 1) + \bar{v}, \quad (3.16)$$

где n – размерность задачи, $PeakNum$ – число оптимумов, $H_i(t)$ – высота пика в момент времени t , $W_i(t)$ – ширина пика, $\bar{X}_i(t)$ – координаты пика, h_{sev} и w_{sev} – параметры задачи, v – вектор фиксированной длины для перемещения центра пика, направление которого в зависимости от сценария выбирается случайно или на основе прошлых перемещений.

Пример реализации задачи MPB (сценарий 1) для размерности 2 и двух разных состояний среды показан на рисунке 3.4.

Второй набор включает множество эталонных тестовых задач, предложенных на соревновании по нестационарной оптимизации в рамках конференции IEEE CEC 2009. Эталонные тестовые задачи содержат 6 разных задач нестационарной оптимизации, для каждой из которых моделируются 7 различных типов изменений в среде (седьмой тип – случайный выбор одного из первых шести). Описание задач, их параметров и условия использования бенчмарка приведены в [255], исходные коды для моделирования предложены в [256].

В таблице 3.1 приведены параметры метода 2 при решении первого набора задач. Значения параметров подобраны экспериментально в ходе предварительных экспериментов.

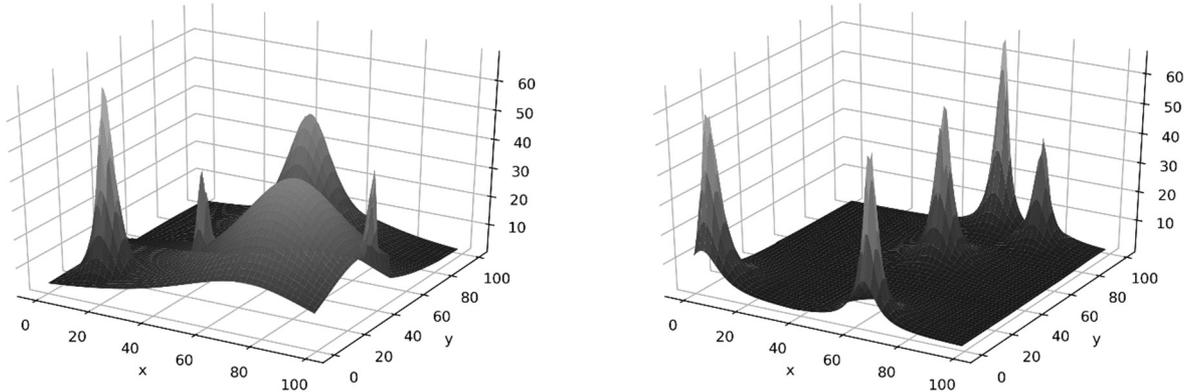


Рисунок 3.4 – Пример реализации задачи МРВ для двух разных состояний среды

Результаты исследования эффективности метода 2 сравнены с результатами исследования ЭА с одной из базовых эвристик. Как отмечалось ранее, каждая из эвристик имеет преимущества на одном из типов изменений в среде. Поскольку для задач с алгоритмически заданными целевыми функциями, мы не можем заранее определить их характеристики, выбор подходящей эвристики затруднен. Поэтому результаты применения метода 2 целесообразно сравнить с оценкой эффективности случайного выбора одной из базовых эвристик, вычисленной как среднее по оценкам эффективности отдельных алгоритмов.

Таблица 3.1 – Параметры метода 2 для модельных задач нестационарной оптимизации

Параметр	Значение
Размер популяции	500
Число подпопуляций	5
Минимальное значение размера подпопуляции	25

Продолжение таблицы 3.1

Параметр	Значение
Точность кодирования хромосомы для динамической функции Растригина	1.0E-2
Точность кодирования хромосомы для задачи МРВ	1.5E-3
Число вычислений целевой функции между двумя изменениями в среде	5000
Число изменений в среде	100
Число независимых запусков алгоритма	100
Размер архива для реализации механизма явной памяти	20
Размер ниши для реализации механизма поддержания разнообразия для динамической функции Растригина	0.05
Размер ниши для реализации механизма поддержания разнообразия для задачи МРВ	0.5

Оценка эффективности алгоритмов для задач первого набора осуществляется по значению абсолютной ошибки лучшего найденного решения на последней итерации каждого цикла между двумя изменениями в среде (3.17). Ошибка усредняется по количеству изменений в одном независимом запуске и по числу независимых запусков (3.18).

$$E_{c,run}^{last} = |f(x_{bestFound}, T_c) - f(x^*, T_c)|, \quad (3.17)$$

$$avgE^{last} = \frac{1}{numRun \cdot numChange} \sum_{run=1}^{numRun} \sum_{c=1}^{numChange} (E_{c,run}^{last}), \quad (3.18)$$

где $E_{c,run}^{last}$ – значение ошибки решения на последней итерации T_c цикла c для независимого запуска run , $numRun$ – число независимых запусков, $numChange$ – число изменений в среде в одном независимом запуске.

Средние значения ошибки для модельных задач представлены в таблицах 3.2 и 3.3. Результаты ранжирования методом глубокого статистического сравнения представлены на рисунках 3.5 и 3.6.

Таблица 3.2 – Результаты численных экспериментов на задаче нестационарной оптимизации МРВ (3.13) – (3.16)

Число пиков (сценарий)	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по столбцам 2-5	Метод 2
1 (2)	0.5	1.25	0.96	0.4	1.98	1.018	0.82
5 (1)	3.27	5.03	4.85	3.93	7.63	4.942	4.56
10 (2)	9.96	10.87	10.84	9.01	16.25	11.385	7.12
20 (2)	8.72	9.35	11.5	7.97	17.82	11.072	7.33
50 (3)	14.91	9.62	18.2	11.15	16.3	14.056	8.97

Таблица 3.3 – Результаты численных экспериментов на динамической функции Растригина (3.12)

Размерность	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по столбцам 2-5	Метод 2
2	0	0	0	0.7	1.3	0.4	0
5	3.56	4.02	6.74	6.87	8.15	5.868	2.83
10	14.53	15.01	21.23	21.8	34.36	21.386	11.56
20	16.27	19.35	34.09	17.59	35.62	24.584	11.91
50	30.62	36.07	43.70	29.82	41.11	36.264	28.37

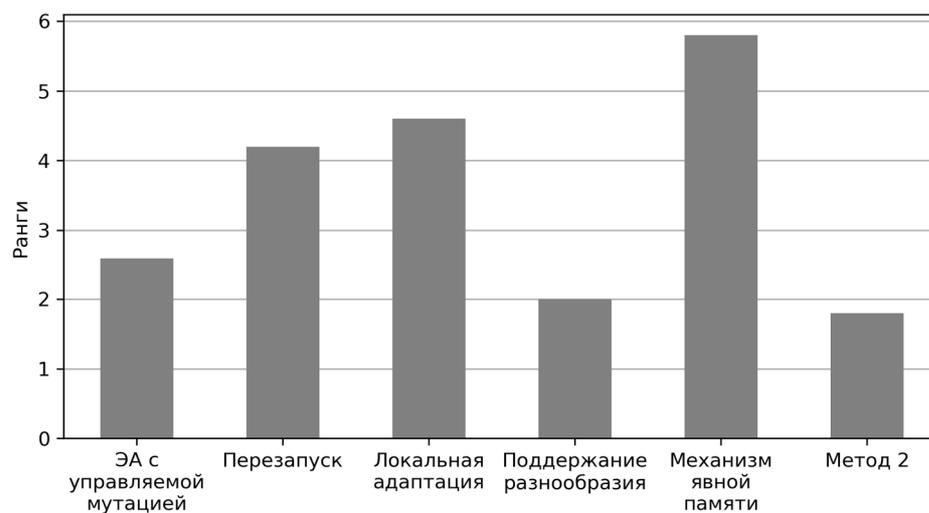


Рисунок 3.5 – Ранжирование алгоритмов на задаче МРВ

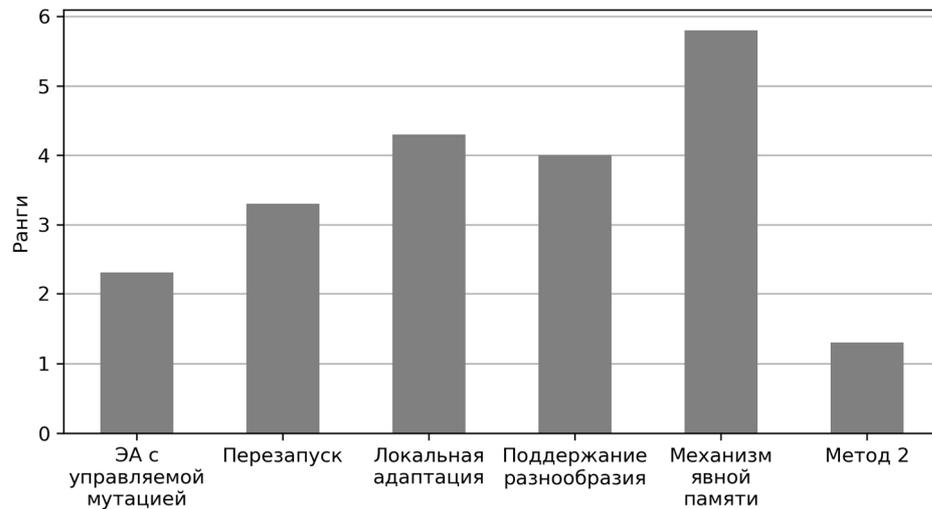


Рисунок 3.6 – Ранжирование алгоритмов на динамической функции Растригина

В таблице 3.4 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов с применением теста Манна-Уитни-Уилкоксона. В таблице показано, сколько раз на модельных задачах метод 2 был лучше, сравним или хуже ЭА с базовыми эвристиками нестационарной оптимизации.

Таблица 3.4 – Результаты теста Манна-Уитни-Уилкоксона на модельных задачах

Метод 2	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти
Лучше	6	7	7	7	10
Нет статистически значимой разницы	3	3	3	1	0
Хуже	1	0	0	2	0

Как видно из результатов численных экспериментов, селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации на модельных задачах по значению модифицированной офлайн ошибки превосходит оценку эффективности случайного выбора ЭА с одной из базовых эвристик. Более того, в подавляющем большинстве случаев метод 2 превосходит и сами ЭА с одной из базовых эвристик, что видно по результатам ранжирования и теста Манна-Уитни-Уилкоксона.

ЭА с механизмом явной памяти продемонстрировал наихудший результат, однако в составе гиперэвристики при поддержке других эвристик он демонстрирует высокую эффективность. На рисунке 3.7 представлен график распределения размеров подпопуляций в методе 2 в одном из прогонов на задаче МРВ. Вертикальные линии – моменты, когда происходят изменения в среде. Размер подпопуляции эвристики механизма явной памяти не уменьшается постоянно до минимального порогового значения. После накопления информации, в повторяющихся ситуациях, механизм явной памяти быстрее находит лучшее решение, и поэтому, после перераспределения индивидов, размер его подпопуляции увеличивается. При появлении новых состояний среды, метод 2 переключается на использование других подходов. Как видно из графика, несмотря на то, что в данном прогоне гиперэвристика чаще выбирала механизм поддержания разнообразия и перезапуск, при некоторых изменениях в среде, более эффективными оказывались и другие эвристики.

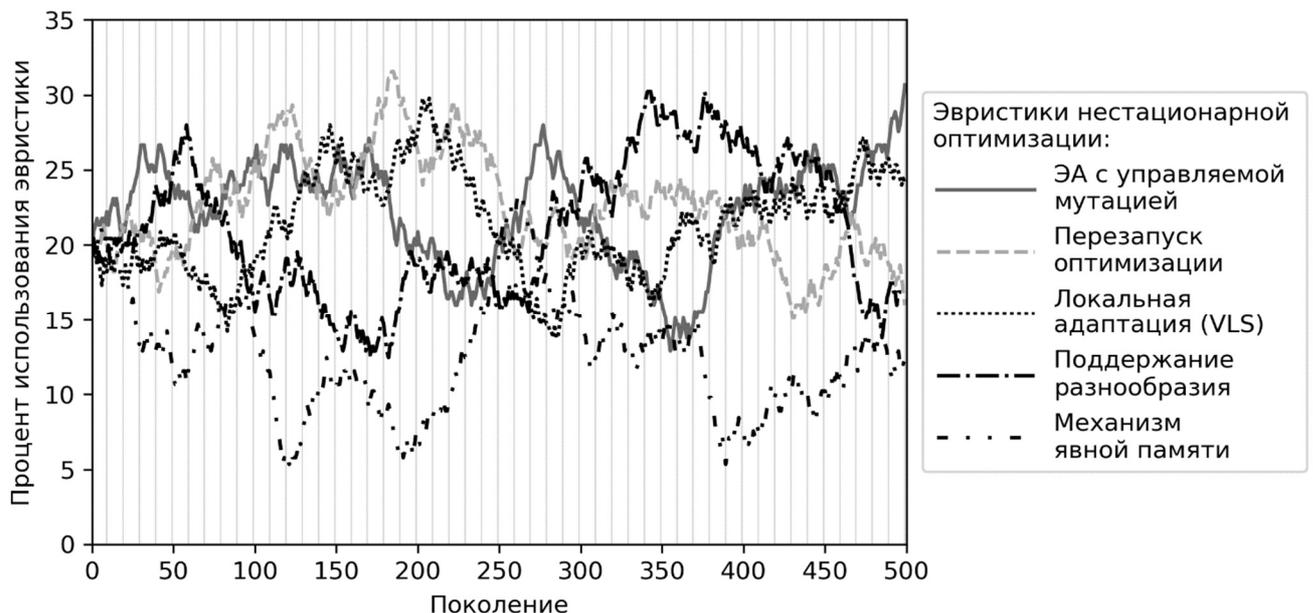


Рисунок 3.7 – Перераспределение подпопуляций в методе 2

Обобщенные результаты оценки гиперэвристики на эталонных задачах нестационарной оптимизации представлены в таблице 3.5. Критерий

эффективности, согласно условиям конкурса IEEE CEC – абсолютная ошибка на последней итерации цикла между двумя изменениями в среде. Параметры экспериментов установлены согласно требованиям конкурса.

Результаты сравнены с результатами известных алгоритмов, участвовавших в конкурсе IEEE CEC: алгоритм само-адаптивной дифференциальной эволюции (*self-adaptive differential evolution algorithm*, jDE) [257], алгоритм роя частиц с кластеризацией (*a clustering particle swarm optimizer*, CPSO) [258] и алгоритм эволюционного программирования с ансамблем архивов явной памяти (*evolutionary programming with ensemble of explicit memories*, EP with EEM)) [259]. Таблица 3.5 также содержит результаты применения традиционных ГА и алгоритмом роя частиц (PSO). Результаты в таблице усреднены по типам изменений в среде, чтобы продемонстрировать как меняется эффективность подходов для разных типов изменений. На рисунке 3.8 представлена диаграмма размаха для усреднённых результатов. Детальные результаты численных экспериментов приведены в таблицах Б.1-Б.6 и на рисунках Б.1-Б.6 Приложения Б. Поскольку для алгоритмов jDE, CPSO и EP with EEM авторы представили только итоговые усреднённые оценки, тест на статистическую значимость различий результатов численных экспериментов не проводился. Аналогично без исходных данных экспериментов нельзя провести и ранжирование по схеме глубокого статистического сравнения, поэтому ранги на рисунке 3.9 построены по итоговым данным традиционным способом.

Экспериментальное исследование предложенной селективной онлайн гиперэвристики синтеза ЭА нестационарной оптимизации показало, что данный подход позволяет синтезировать эффективный ЭА в процессе решения задачи, адаптируясь к типам изменений путем выбора подходящей базовой эвристики. Гиперэвристика всегда превосходит по эффективности среднюю эффективность ЭА, использующих базовые эвристики, а с увеличением сложности задачи может превосходить и ЭА с лучшей эвристикой. Разброс оценок эффективности по серии независимых экспериментов показывает, что метод 2 дает повторяемые результаты, сравнимые с наиболее эффективными из исследованных алгоритмов.

Аналогично и с разбросом оценок усредненных по типам изменений. Несмотря на то, что самые стабильные результаты дал алгоритм EP with EEM, есть алгоритмы, которые превосходят его на некоторых типах изменений. Стоит отметить, что EP with EEM специально проектировался для участия в конкурсе по нестационарной оптимизации и входит топ-5 алгоритмов по итогам конкурса.

Таблица 3.5 – Сравнение эффективности алгоритмов на эталонных тестовых задачах нестационарной оптимизации

Тип изменений в среде	Алгоритмы					Среднее по столбцам 2-4	Среднее по столбцам 2-6	Метод 2
	jDE	CPSO	EP with EEM	GA	PSO			
T1	3.376	25.007	32.619	51.680	132.247	20.334	48.985	28.188
T2	110.871	154.920	32.559	212.125	267.532	99.450	155.601	108.478
T3	114.743	141.251	30.503	178.257	230.507	95.499	139.052	96.026
T4	12.491	73.736	32.285	102.303	178.493	39.504	79.862	31.568
T5	104.955	172.239	24.095	245.253	249.860	100.430	159.281	124.885
T6	43.027	131.719	26.686	117.575	206.283	67.144	105.058	46.185
T7	32.223	118.132	28.642	120.087	176.050	59.666	95.027	34.264

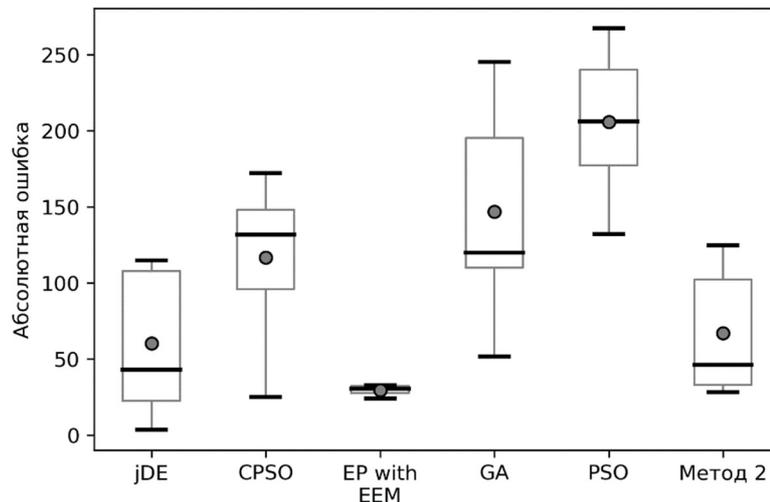


Рисунок 3.8 – Диаграмма размаха оценок эффективности алгоритмов на эталонных тестовых задачах нестационарной оптимизации

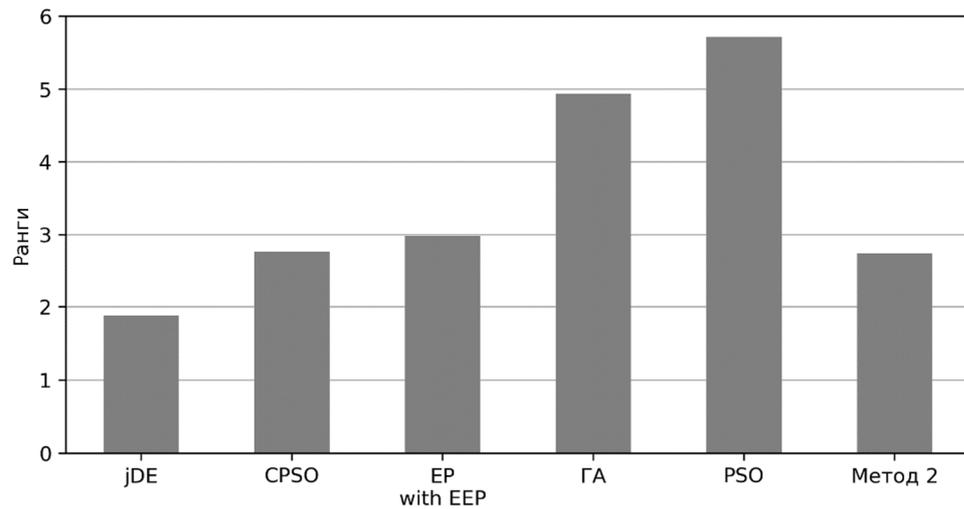


Рисунок 3.9 – Ранжирование алгоритмов на эталонных тестовых задачах нестационарной оптимизации

Известные узкоспециализированные ЭА нестационарной оптимизации могут превосходить предложенную гиперэвристику по средней эффективности, но оценки их эффективности могут существенно ухудшаться на некоторых типах изменений среды. Учитывая, что целевые функции заданы алгоритмически, а тип и характеристики изменений не известны заранее, выбор подходящего специализированного алгоритма затруднен. В то же время, предложенная гиперэвристика превосходит оценку эффективности случайного выбора алгоритма независимо от типа изменений в среде.

3.4 Решение практической задачи нестационарной оптимизации для идентификации параметров боковой динамики движения малого беспилотного летательного аппарата в реальном времени

Для апробации метода 2 была решена практическая задача нестационарной оптимизации для идентификации параметров боковой динамики движения малого беспилотного летательного аппарата (БПЛА) с фиксированным крылом в реальном времени. Современные БПЛА применяются практически во всех сферах

деятельности начиная от бытовых задач (фото-, видеосъемка, развлечения, спорт) и заканчивая специальными гражданскими (наблюдение, сельское хозяйство, доставка) и военными приложениями (разведка, ведение боя). Согласно Федеральным правилам использования воздушного пространства Российской Федерации [260], главным признаком БПЛА является отсутствие человека пилота, но при этом различают БПЛА управляемые автоматически, оператором по каналу связи и гибридные. Очевидно, что полностью автоматические БПЛА представляют наибольший интерес, т.к. имеют ряд преимуществ, в частности, уменьшение затрат на пилотирование, возможность пилотирования на протяжении более длительного времени, более быстрая реакция и возможность контролировать большее число внешних факторов одновременно. При разработке автономных БПЛА возникает задача проектирования системы управления (СУ) БПЛА, которая была бы достаточно робастной по отношению к изменению условий эксплуатации БПЛА (изменения направления и порывы ветра, изменение плотности воздушной среды и др.) и погрешностям аппаратуры (погрешности измерения аэродинамических параметров, погрешности исполнительных органов и др.). Поэтому при проектировании СУ БПЛА необходимо уметь идентифицировать параметры, на основе которых происходит управление БПЛА.

Были предложены разные подходы к идентификации параметров БПЛА [261, 262]. Поскольку БПЛА является сложной нелинейной динамической системой, одним из популярных инструментов для идентификации параметров являются искусственные нейронные сети (НС) [263-265]. Преимуществом НС также является возможность их простой аппаратной реализации. Обучение НС для идентификации параметров может осуществляться офлайн после сбора информации об эксплуатации БПЛА и онлайн непосредственно во время полета. Онлайн обучение НС позволяет адаптировать модель к изменениям условий эксплуатации во время полета, но обычно точность идентификации ниже, т.к. для обучения в реальном масштабе времени используется меньше обучающих данных [266]. Задача обучения НС является задачей оптимизации, решаемой градиентными методами [265]. Сложность онлайн обучения НС при идентификации параметров БПЛА

связана с тем, что решаемая задача оптимизации является нестационарной в следствии постоянных изменений условий эксплуатации БПЛА [267].

Любой БПЛА может быть описан как динамическая система, имеющая 6 степеней свободы, которые объединяют в 2 группы для описания продольной и боковой динамики [268]. Параметры для описания динамики связаны с его движением в плоскости и вне плоскости симметрии БПЛА. Поскольку возможна декомпозиция модели на две по 3 степени свободы, обычно их идентификация происходит независимо друг от друга. Продольная динамика чаще используется для решения задач движения по траектории и навигации, в то время как управление боковой динамикой необходимо для стабилизации БПЛА на траектории полета. В данном исследовании задача идентификации решается только для параметров боковой динамики.

Задача идентификации параметров боковой динамики движения в реальном времени решена для БПЛА, разработанного в институте инженерного дела и информационных технологий Университета Нового Южного Уэльса в Австралии (рисунок 3.10). БПЛА представляет собой компактный летательный аппарат с фиксированным крылом (высокоплан), бортовое оборудование представлено СУ на основе платформы PC/104, использующей данные трех-осевого инерциального измерительного блока, датчиков статического и динамического давления, потенциометра бокового скольжения и датчика угла атаки. Данные для идентификации параметров предоставлены профессором Мэтью Гарраттом (Школа инженерии и информационных технологий, Университета Нового Южного Уэльса, Канберра, Австралия).

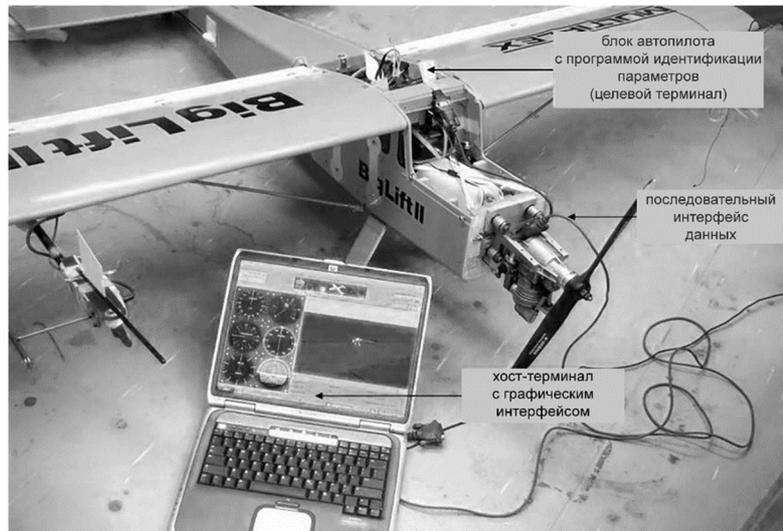


Рисунок 3.10 – Внешний вид БПЛА

Целевые параметры для решения задачи идентификации – углы тангажа, крена и рыскания, которые соответствуют трём углам Эйлера и определяют ориентацию БПЛА в нормальной системе координат (рисунок 3.11). Угол тангажа (θ) – угол между продольной осью БПЛА горизонтальной плоскостью, угол крена или угол наклона (γ) – угол поворота БПЛА вокруг продольной оси, угол рыскания (ψ) – угол поворота БПЛА в горизонтальной плоскости относительно вертикальной оси.

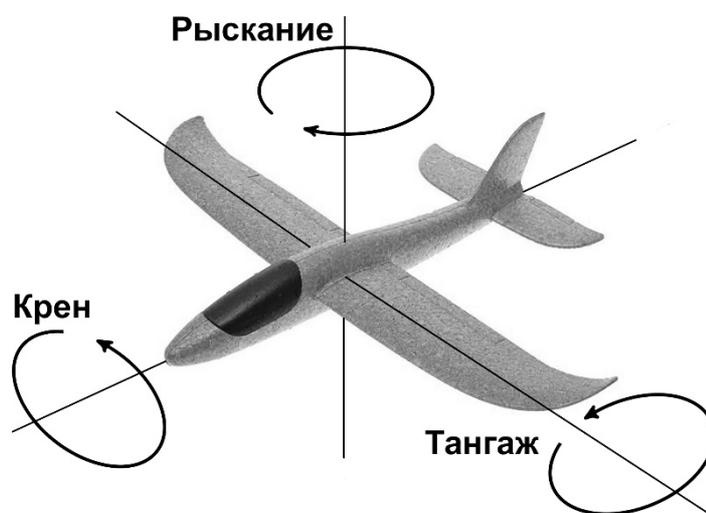


Рисунок 3.11 – Углы тангажа, крена и рыскания

Целевые параметры зависят от следующих значений управляющих параметров: положения элеронов ($\Delta\alpha$), руля высоты (Δe), руля направления (Δr) и рычага управления двигателем (Δth) БПЛА. Поскольку БПЛА является динамической системой, текущие значения целевых параметров также зависят от своих значений и значений управляемых параметров в прошлые моменты времени.

Идентификация параметров осуществляется с помощью рекуррентной нейронной сети – нелинейной авторегрессии с внешним входом (*nonlinear autoregressive with exogenous inputs model*, NARX) [265], которая зарекомендовала себя как эффективный инструмент моделирования и управления сложными динамическими системами [263, 269, 270].

Обозначим целевые параметры как $y(t) = (\theta(t), \gamma(t), \psi(t))$, а управляемые параметры как $u(t) = (\Delta\alpha(t), \Delta e(t), \Delta r(t), \Delta th(t))$. Тогда модель авторегрессии можно представить в виде зависимости (3.19), которую необходимо восстановить с помощью нейронной сети.

$$y(t) = f_{NARX} \left(u(t-1), u(t-2), \dots, u(t-T_u), y(t-1), \dots, y(t-T_y) \right), \quad (3.19)$$

Разработчики БПЛА спроектировали архитектуру НС, представленную на рисунке 3.12, и продемонстрировали, что задача обучения данной НС является нестационарной и традиционный алгоритм обратного распространения ошибки недостаточно эффективен для решения задачи в реальном времени [267]. НС имеет один скрытый слой с нейронами сигмоидного типа, значения T_u и T_y – являются параметрами модели.

Данные для обучения НС представлены 6 наборами, полученными для разных условий эксплуатации БПЛА, значения измеряемых параметров регистрировались с периодичностью 0,02 сек. Наборы содержат следующее количество записей: 17981, 11532, 6774, 20112, 8681 и 15756.

Пример данных первого набора показан на рисунках 3.13 и 3.14, пример данных в большем масштабе (интервал [2000,2500] длительностью 10 сек) показан на рисунках 3.15 и 3.16.

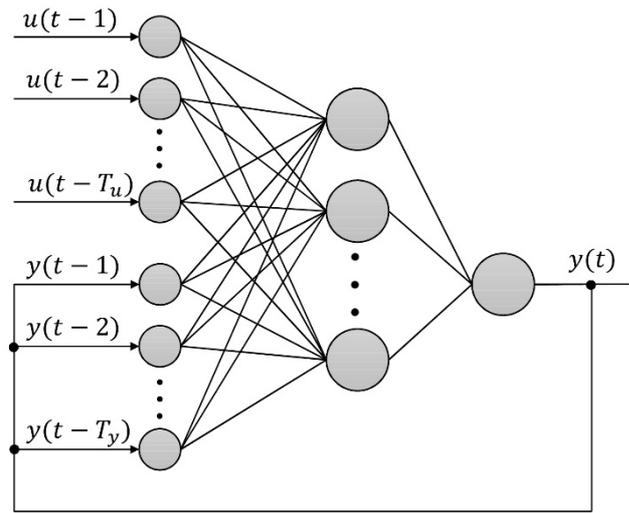


Рисунок 3.12 – Архитектура рекуррентной нейронной сети NARX

В связи с ограничениями бортового оборудования БПЛА, разработчики установили следующие требования к параметрам НС и процедуре обучения НС: число нейронов скрытого слоя до 10 (в [266] используется 4 нейрона), максимальное число эпох обучения НС равно 15, размер подвыборки (*mini-batch*) для обучения до 25 примеров. В данном исследовании будем использовать аналогичные требования к параметрам, чтобы сравнить результаты с ранее полученными разработчиками БПЛА.

Для применения метода 2 настройка гиперпараметров НС путем перебора по сетке дала следующий набор эффективных параметров: число нейронов скрытого слоя – 5, размер подвыборки – 25, $T_u = T_y = 7$. Настройки метода 2 представлены в таблице 3.6.

В качестве критерия эффективности решения задачи используется корень среднеквадратической ошибки (*root mean square error*, RMSE) для каждого целевого параметра. Результаты, полученные методом 2, сравнены с результатами полученными обычными методом обратного распространения ошибки, ЭА использующими одну из базовых эвристик нестационарной оптимизации и оценкой случайного выбора одной из базовых эвристик, а также результатами, полученными ранее разработчиками БПЛА [266].

Таблица 3.6 – Настройки метода 2 для задачи идентификации параметров БПЛА

Параметр	Значение
Размер популяции	100
Число подпопуляций	5
Минимальное значение размера подпопуляции	5
Размерность задачи оптимизации (весовые коэффициенты НС)	93
Точность кодирования хромосомы	1.0E-2
Число независимых запусков алгоритма	40
Размер архива для реализации механизма явной памяти	5
Размер ниши для реализации механизма поддержания разнообразия	0.025

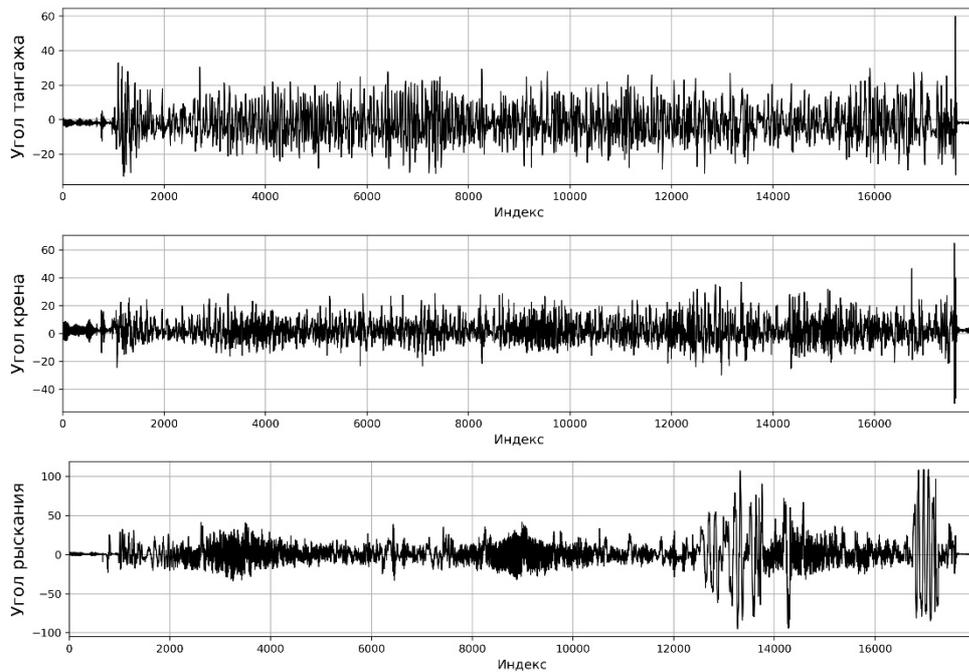


Рисунок 3.13 – График изменения целевых параметров на наборе данных 1

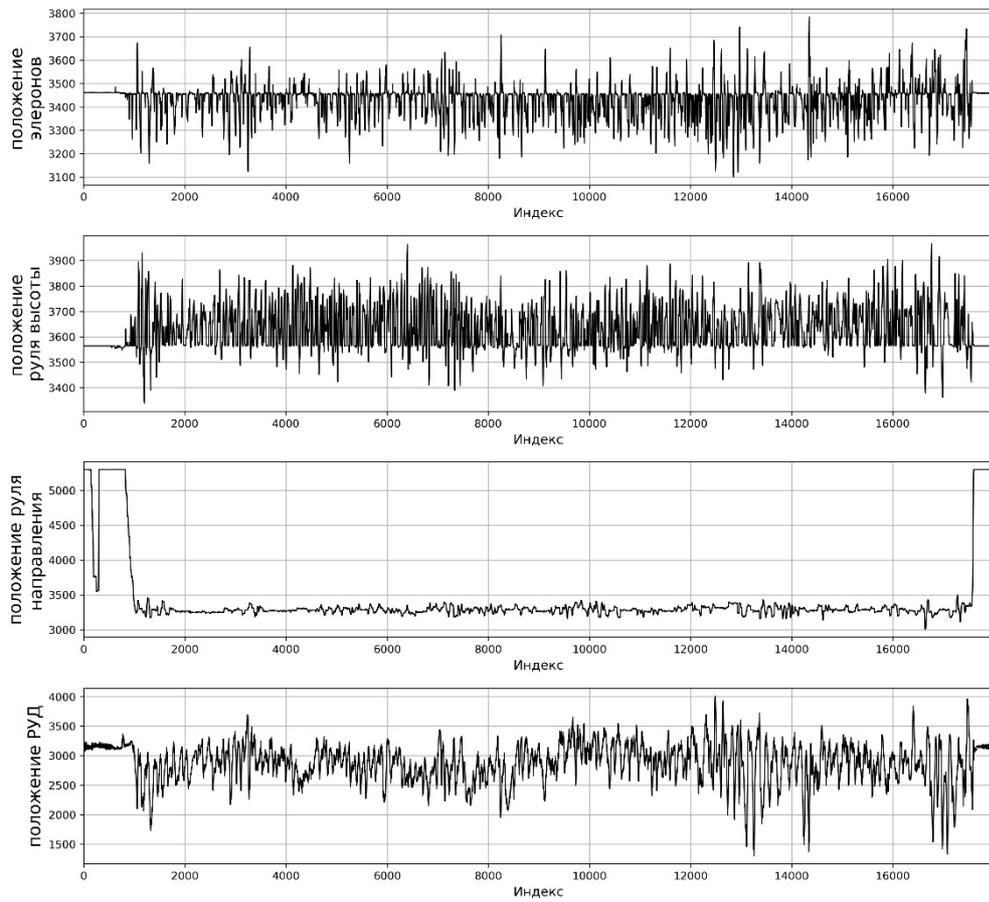


Рисунок 3.14 – График изменения управляемых параметров на наборе данных 1

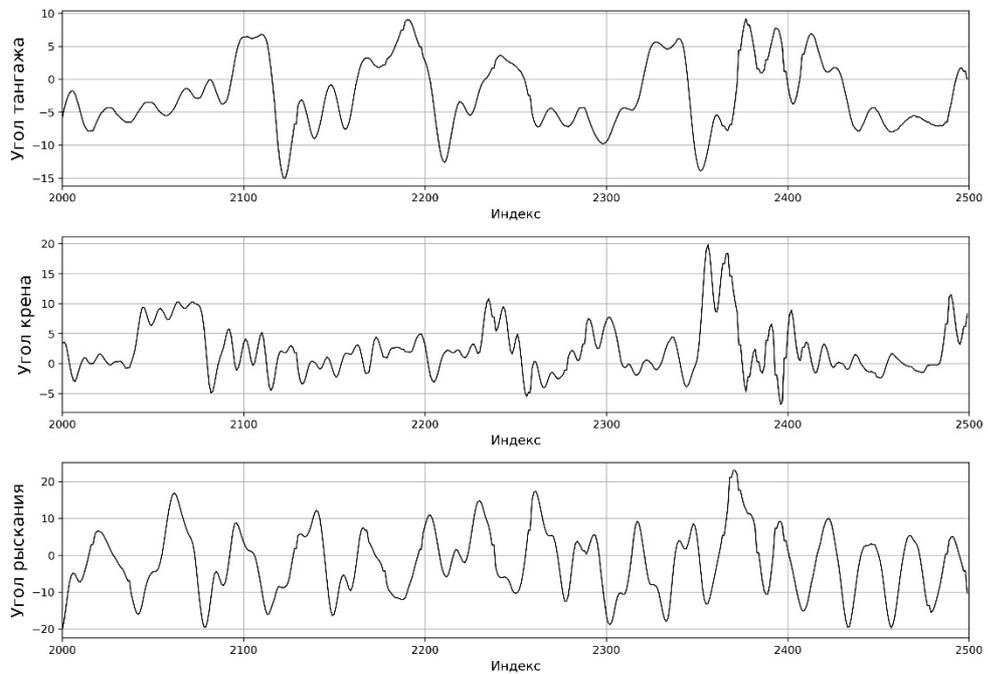


Рисунок 3.15 – Фрагмент графика изменения целевых параметров

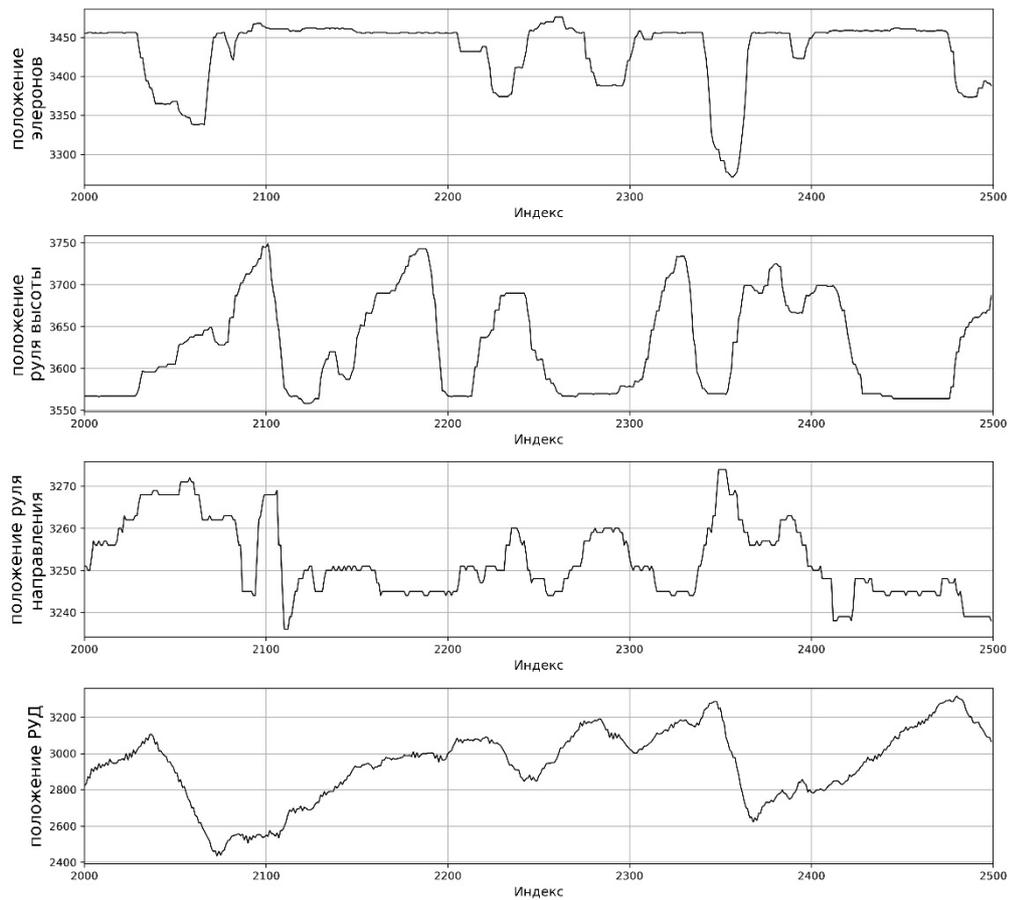


Рисунок 3.16 – Фрагмент графика изменения управляемых параметров

Программная реализация алгоритмов для проведения численных экспериментов выполнена на языке Python 3.7 с использованием пакета Keras для НС. Результаты решения задачи, усреднённые по всем наборам приведены в таблице 3.7, диаграмма размаха, построенная по независимым прогонам показана на рисунке 3.17. Детальные результаты для каждого набора приведены в таблицах Б.7-Б.12 Приложения Б. Пример работы НС на интервале в 500 значений (10 сек) показан на рисунке 3.18.

Таблица 3.7 – Результаты решения задачи идентификации параметров боковой динамики БПЛА

RMSE	Предыдущий результат	НС с алгоритмом обратного распространения ошибки	Лучшее при использовании одной базовой эвристики	Среднее по базовым эвристикам	Метод 2
Угол крена	0.00679	0.01021	0.004082	0.00840	0.00475
Угол тангажа	0.01667	0.05340	0.012319	0.01844	0.01076
Угол рыскания	0.00100	0.03164	0.000885	0.00224	0.00077
Среднее	0.00815	0.03175	0.005762	0.00969	0.00543

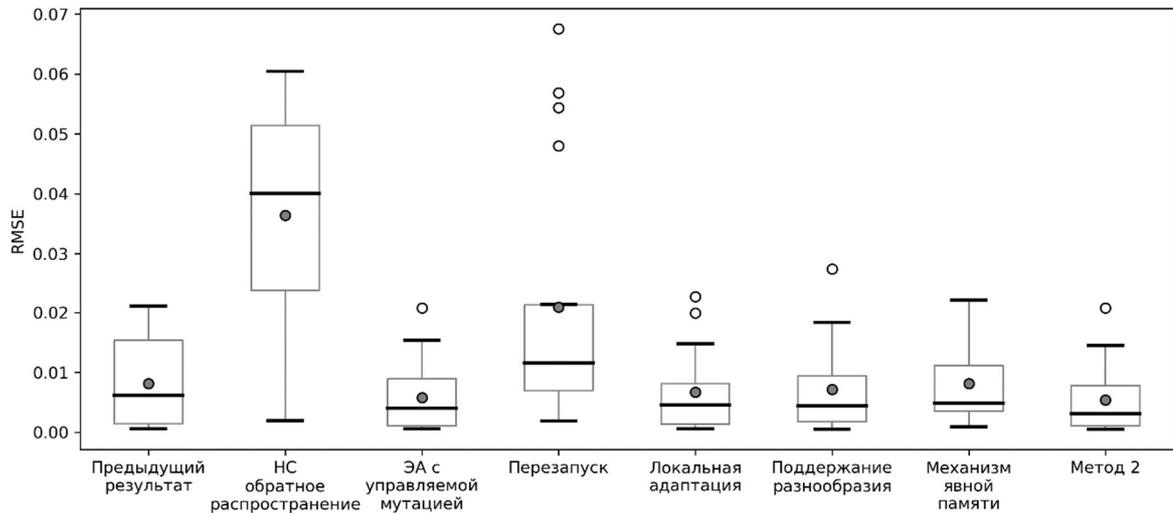


Рисунок 3.17 – Диаграмма размаха результатов решения задачи идентификации параметров боковой динамики БПЛА

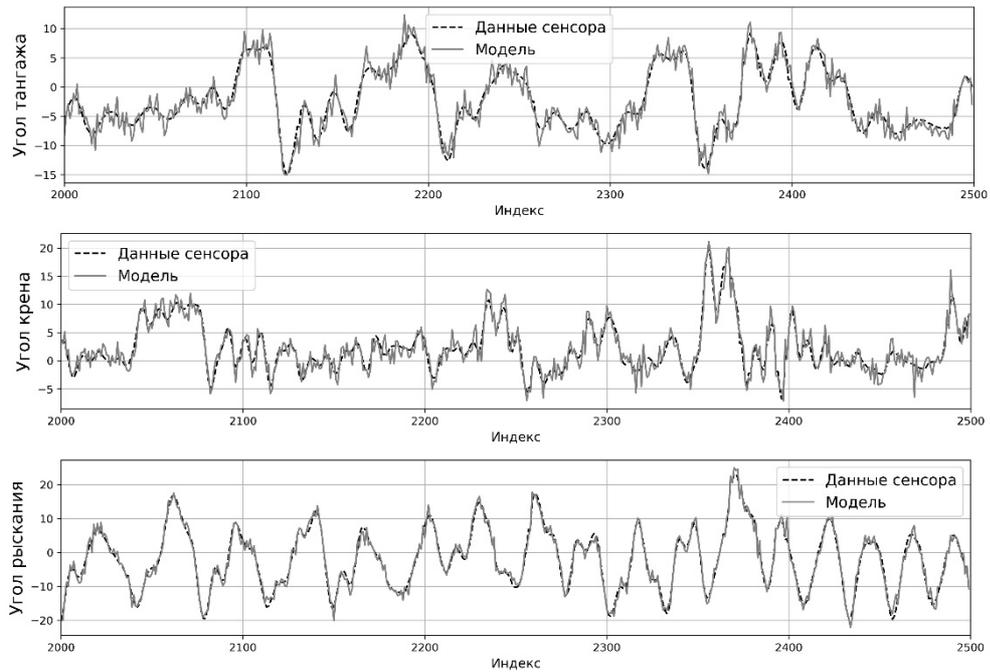


Рисунок 3.18 – Пример работы НС на интервале 10 сек на наборе данных 1

На рисунке 3.19 показаны результаты ранжирования исследуемых подходов методом глубокого статистического сравнения по всем данным, в таблице 3.8 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов (тест Манна-Уитни-Уилкоксона).

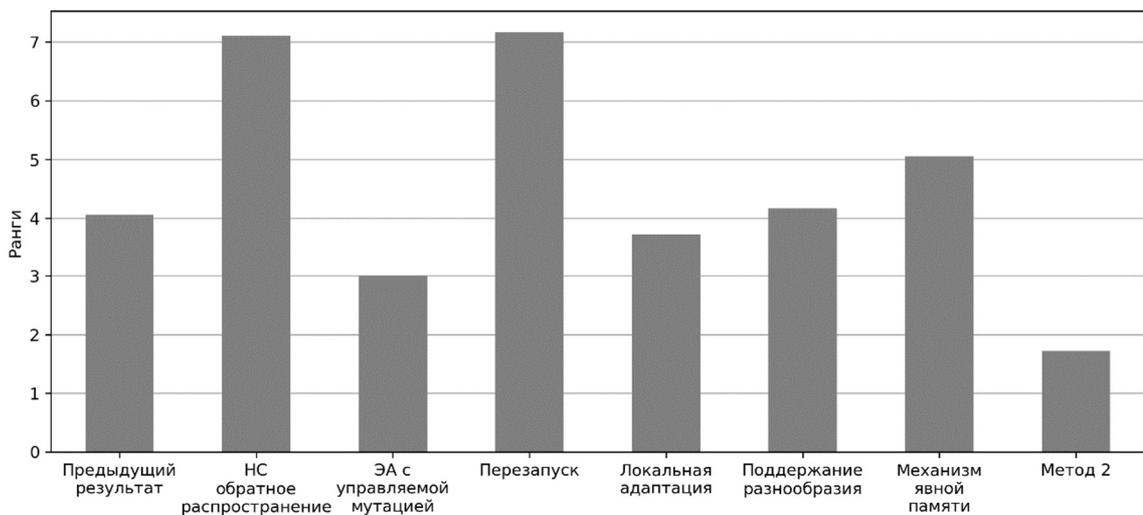


Рисунок 3.19 – Ранжирование подходов на задаче идентификации параметров боковой динамики БПЛА

Таблица 3.8 – Результаты теста Манна-Уитни-Уилкоксона

Метод 2	Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Итого
Угол крена								
Лучше	4	3	3	6	6	5	5	32
Нет статистически значимой разницы	1	0	2	0	0	1	1	5
Хуже	1	3	1	0	0	0	0	5
Угол тангажа								
Лучше	4	6	4	6	3	5	4	32
Нет статистически значимой разницы	1	0	1	0	3	1	2	8
Хуже	1	0	1	0	0	0	0	2
Угол рыскания								
Лучше	4	6	3	6	4	5	5	33
Нет статистически значимой разницы	0	0	3	0	2	1	1	7
Хуже	2	0	0	0	0	0	0	2

Как видно по результатам экспериментов, ЭА нестационарной оптимизации существенно превосходят традиционный метод обучения НС с помощью обратного распространения ошибки. Самый большой разброс результатов у эвристики перезапуска процедуры поиска, что может указывать на то, что изменения в среде не очень интенсивные и могут быть циклическими. Метод 2 превосходит оценку эффективности случайного выбора одной из базовых эвристик по всем целевым параметрам. Однако для угла крена лучший результат, усредненный по всем наборам данных, получен ЭА с управляемой мутацией. Для углов тангажа и рыскания метод 2 превосходит лучшие результаты, полученные одной базовой эвристикой.

Полученные с помощью метода 2 результаты превосходят по критерию корня из среднеквадратической ошибки результаты, ранее полученные разработчиками БПЛА. В дальнейшем, при практической реализации метода 2 на бортовом оборудовании БПЛА, при наличии более производительной СУ можно попытаться

уменьшить ошибку идентификации за счет увеличения числа поколений или размера популяции в методе 2.

Таким образом, результаты апробации метода 2 при решении прикладной задачи, подтвердили, что с помощью селективной онлайн гиперэвристики можно эффективно решать задачи нестационарной оптимизации с алгоритмически заданными целевыми функциями.

Выводы по главе 3

Нестационарность сложных систем приводит к проблеме дрейфа экстремума при решении задач оптимизации. Несмотря на то, что для задач нестационарной оптимизации предложены различные эвристики, для задач глобальной оптимизации с алгоритмически заданными функциями не удастся заранее определить тип или комбинацию типов изменений в среде и их характер, как следствие, выбор подходящей эвристики становится затруднительным. Более того, известные эвристики обычно демонстрируют высокую эффективность для конкретного типа изменений в среде, для других типов изменений их эффективность может существенно снижаться.

Для задач нестационарной оптимизации выбор подходящей эвристики должен осуществляться в процессе решения задачи на основе оценки состояния среды, что обеспечит адаптацию алгоритма поиска под конкретный тип изменений. Предложенная в главе 3 гиперэвристика синтеза ЭА нестационарной оптимизации на основе метода портфолио алгоритмов и метода адаптации вероятностей на уровне популяции (метод 2) осуществляет адаптацию к изменениям среды путем анализа эффективности используемых базовых эвристик, увеличивая или уменьшая вероятность их применения на уровне популяции. В методе 2 применяются два уровня адаптации: в локальном цикле – для оценки и выбора эвристики для текущей ситуации и в глобальном – для оценки вероятности появления конкретных типов изменений текущей решаемой задачи. Таким

образом, при возникновении изменений в среде, происходит автоматический выбор и комбинация наиболее подходящих для текущего состояния эвристик, т.к. формируется новый ЭА.

Анализ эффективности метода 2 при решении эталонных тестовых задач нестационарной оптимизации показал, что селективная онлайн гиперэвристика в большинстве случаев по значению модифицированной офлайн ошибки превосходит ЭА с лучшей базовой эвристикой и всегда превосходит оценку эффективности случайного выбора ЭА с одной из базовых эвристик, следовательно, выбор метода 2 является более предпочтительным для нестационарных задач глобальной оптимизации с алгоритмически заданными функциями. Метод 2 был сравнен с результатами известных специализированных алгоритмов, участвовавших в конкурсе по стационарной оптимизации. Для некоторых типов изменений в среде, выбор метода 2 оказывается более предпочтителен чем случайный выбор одного из специализированных алгоритмов. При этом эффективность метода 2 менее чувствительна к типам изменений. Анализ распределения вероятностей применения эвристик в процессе решения задач показал, что все базовые эвристики постоянно задействованы и даже наименее эффективные могут получить преимущество и внести вклад в общее решение в некоторые периоды решения задачи.

Эффективность метода 2 была также подтверждена результатами решения практической задачи нестационарной оптимизации для идентификации параметров боковой динамики движения малого БПЛА с фиксированным крылом в реальном времени. В среднем по трем идентифицируемым параметрам по критерию корня среднеквадратической ошибки результаты идентификации параметров с помощью метода 2 превосходят как результаты, полученные ранее разработчиками БПЛА, так и результаты, полученные ЭА с одной из базовых эвристик нестационарной оптимизации.

Поддержание разнообразия популяции ЭА является эффективной эвристикой для задач нестационарной оптимизации. Поддержка разнообразия также повышает шансы найти глобальный оптимум в многоэкстремальных задачах

оптимизации и поэтому часто используется для усиления традиционных ЭА. С точки зрения прикладного системного анализа поддержка разнообразия может помочь локализовать различные оптимумы задачи и, следовательно, обеспечит ЛПР и стейкхолдеров различными альтернативами. Применим подход на основе гиперэвристик для аппроксимации множества оптимумов в задаче глобальной оптимизации с алгоритмически заданной целевой функцией.

ГЛАВА 4. СЕЛЕКТИВНАЯ ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ИДЕНТИФИКАЦИИ МНОЖЕСТВА ЭКСТРЕМУМОВ

Глава 4 посвящена проблеме разработки и исследования гиперэвристик для решения задач аппроксимации множества экстремумов. При решении многоэкстремальных задач глобальной оптимизации многие подходы имеют тенденцию сходиться к лучшему найденному решению. С точки зрения прикладного системного анализа более перспективным является анализ множества альтернативных решений задачи, т.к. процедура прикладного системного анализа является итеративной и предпочтения ЛПР могут меняться в процессе изучения проблемосодержащей системы. Более того, в задачах с алгоритмически заданными целевыми функциями без дополнительной информации обычно сложно определить, является ли оптимум локальным или глобальным. Таким образом, решение задачи идентификации множества оптимумов может оказаться практически полезным как с точки зрения решения конкретной задачи оптимизации, так и с точки зрения анализа сложной системы, в которой данная задача возникла. Существуют разные эвристики управления ЭА, обеспечивающие поддержку разнообразия популяции, эффективность которых зависит от особенностей ландшафта целевой функции. Поэтому требуется выбирать и комбинировать эвристики аппроксимации множества экстремумов для конкретной задачи. Предложенная в главе 4 селективная гиперэвристика синтеза ЭА идентификации множества экстремумов осуществляет контроль распределении популяции в пространстве поиска и идентифицирует зоны притяжения оптимумов, найденные различными эвристикami. Гиперэвристика может быть реализована как офлайн, так и онлайн. Численные исследования на множестве тестовых и эталонных задач показывают, что предложенная гиперэвристика предпочтительнее случайного выбора ЭА с одной из базовых эвристик, а также превосходит некоторые из ведущих алгоритмов мультимодальной оптимизации и сравнима с эффективностью случайного выбора одного из них. Апробация подхода при

решении практических задач из области поддержки принятия решений и проектирования систем на нечеткой логике для решения задач классификации показала, что предложенная гиперэвристика обеспечивает эффективное решение задачи поиска глобального оптимума и обеспечивает ЛПР дополнительными локально-оптимальными решениями, анализ которых обеспечивает ЛПР дополнительной информацией о свойствах задачи и способах ее решения.

4.1 Проблема аппроксимации множества экстремумов в задачах оптимизации сложных систем и обзор подходов к ее решению

Многие прикладные задачи оптимизации имеют более одного глобального оптимума. Также в допустимой области пространства поиска может существовать множество удовлетворительных локально-оптимальных решений. Задачи, у которых в допустимой области имеется множество оптимумов, называются многоэкстремальными или мультимодальными (в противоположность унимодальным функциям) [32, 38]. Сложность решения многоэкстремальных задач связана с тем, что алгоритмы поисковой оптимизации, попадая в область притяжения одного из локальных оптимумов, сходятся в точку этого оптимума. Запуск процедуры поиска из разных стартовых позиций, в общем случае, не гарантирует, что алгоритм попадет в зону притяжения глобального оптимума или в зоны притяжения различных локальных оптимумов. В свою очередь, ЭА демонстрируют высокую эффективность и в многоэкстремальных задачах, т.к. используют стохастический популяционный поиск вместо последовательного улучшения единственного решения. Тем не менее, ЭА тоже имеют тенденцию сходиться к лучшему найденному решению, теряя разнообразие в популяции. Один из способов повышения эффективности ЭА в мультимодальной среде – это настройка ЭА на поиск нескольких различных экстремумов, что должно помогать алгоритму, в случае захвата одним из локальных оптимумов. Попытки повысить надежность ЭА за счет поиска многих оптимумов предпринимались со времен

работ Голдберга и Де Йонга [271, 272], определивших основы эволюционной мультимодальной оптимизации (*multimodal optimization*, ММО).

Как обсуждалось ранее в главе 1, в ПСА существует необходимость обеспечить потребителей результатов анализа (ЛПР и стейкхолдеров) множеством альтернативных решений. На практике наличие множества альтернатив повышает устойчивость решений. А получение дополнительной информации о структуре целевой функции, может помочь с анализом проблемы и поиском решений.

В последние годы задача поиска многих оптимумов становится более популярной. На сегодня предложено множество эффективных бионических и эволюционных подходов для ММО [273-276]. В 2013 году состоялась первая специальная сессия и соревнование по мультимодальной оптимизации в рамках конференции IEEE CEC.

Задача (1.1), (1.3), сформулированная в главе 1, направлена на идентификацию множества экстремумов и может преследовать следующие цели:

1. Поиск единственного глобально-оптимального решения в многоэкстремальном пространстве поиска путем идентификации нескольких областей притяжения и выбора лучшего из найденных оптимумов.
2. Поиск множества глобальных оптимумов (в предположении, что их несколько) или его репрезентативной аппроксимации.
3. Поиск множества всех (глобальных и локальных) экстремумов или его репрезентативной аппроксимации.

В данном исследовании решается третья задача, которая включает в себя первые две, и потому, является более сложной, но в то же время более интересной и полезной, как с теоретической (исследование ландшафта целевой функции, разработка новых методов), так и с практической (идентифицированные оптимумы являются различными альтернативами при решении исходной проблемы) точек зрения.

Сравнение традиционного и мультимодального подхода в глобальной оптимизации показано на примере поиска экстремумов функции Растригина на рисунке 4.1. Стандартный ЭА, начиная с популяции, случайно распределенной в

пространстве поиска, обычно на последних итерациях сходится к лучшему найденному решению, в идеале – в точку глобального оптимума. Мультиимодальный ЭА должен локализовать различные оптимумы и сойтись к ним, в идеале – в точки глобального и всех локальных экстремумов. На практике, исследователя могут интересовать не все локальные оптимумы, а лишь некоторое количество или те, которые отличаются от глобального не более чем на заданную величину.

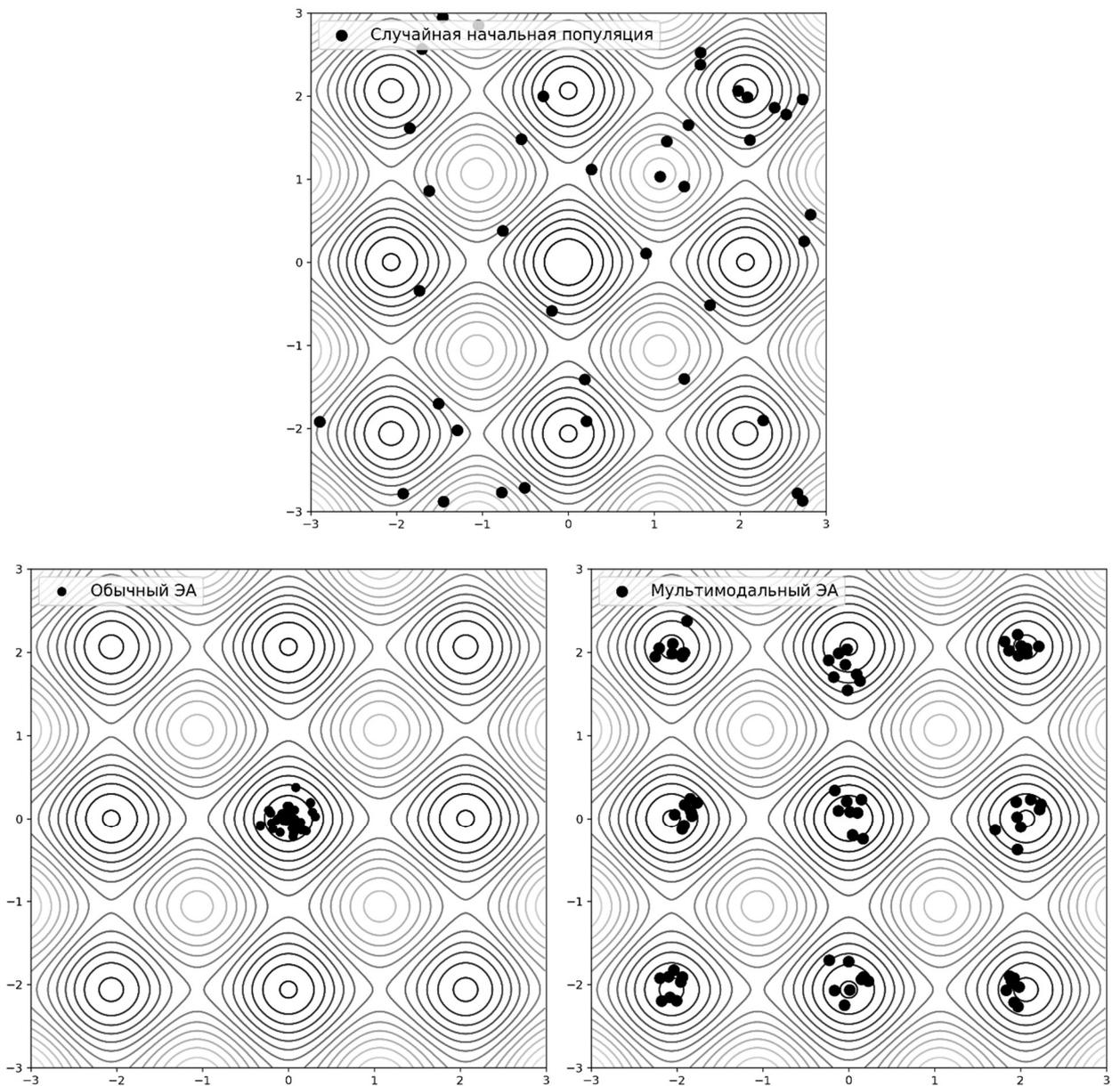


Рисунок 4.1 – Пример работы традиционного и мультиимодального ЭА

Очевидно, что задача (1.1), (1.3) будет решаться эффективнее при использовании популяционных методов, в которых разные члены популяции будут идентифицировать разные области притяжения оптимумов. Практически все ЭА аппроксимации множества оптимумов построены на идее поддержки разнообразия популяции, но отличаются способами исследования пространства поиска, обнаружения и идентификации областей притяжения оптимумов. В тоже время, большинство известных мультимодальных алгоритмов и наилучшие результаты в данной области получены для задач с вещественными переменными [274]. Основная причина – лучшее понимание и использование свойств ландшафта целевой функции в вещественном пространстве поиска, использование традиционных метрик для определения областей притяжения. Эвристики, предложенные для задач с вещественными переменными, к сожалению, становятся неэффективными для задач с дискретными и смешанными переменными. Как отмечалось ранее, разношкальные переменные – одна из особенностей задач оптимизации сложных систем.

Известные эвристики для аппроксимации множества экстремумов можно объединить в три группы [274, 276-278]:

- базовые эвристики, которые используются как модификация или замена базовых операций в ЭА;
- подходы, связанные с анализом топологии пространства поиска;
- различные био-инспирированные методы, не вошедшие в первую и вторую группы.

Подходы третьей группы включают модификации алгоритмов роевого интеллекта [278], алгоритма муравьиных колоний [279], иммунных алгоритмов [280, 281] и других. Подходы третьей группы обычно предназначены для узкоспециализированных проблем и не являются предметом данного исследования.

Вторая группа подходов использует гипотезы и информацию о свойствах ландшафта целевой функции в явном виде и включает такие подходы как алгоритмы на основе кластерного анализа решений [282, 283], многокритериальный подход [284], алгоритмы на основе анализа топологии

целевой функции [285, 286], меметические алгоритмы [287, 288], многонациональный ЭА [289] и другие. Как отмечалось ранее, подобные подходы разработаны для задач с вещественными переменными, и их преимущества теряются в пространствах поиска другого типа (в частности в бинаризованном, при использовании ГА). Некоторые подходы эффективны только для целевых функций, обладающих такими свойствами как гладкость, регулярная структура оптимумов и т.д.

Наибольший интерес представляют подходы первой группы, т.к. они не используют информацию о пространстве поиска в явном виде, а анализируют поведение популяции, следовательно, применимы для задач с алгоритмически заданными функциями. Предложенные эвристики могут быть использованы с любым типом ЭА и произвольными задачами аппроксимации множества экстремумов. Стоит отметить, что многие подходы второй группы в том или ином виде основаны на эвристиках первой группы. Рассмотрим базовые эвристики, используемые в данном исследовании, подробнее.

Базовым понятием для большинства подходов является понятие ниши (*niche*). Ниша определяет часть пространства поиска, в которой необходимо ограничить скопление индивидов. Предполагается, что скопление индивидов в нише образуется, когда ЭА сходится в зоне притяжения некоторого оптимума. Ограничение скопления индивидов в одной нише должно стимулировать ЭА локализовать зоны притяжения других оптимумов. Поэтому методы аппроксимации множества экстремумов часто называют методами формирования ниш (*niching methods*) [276, 290].

Разделение пригодности (fitness sharing). Эвристика основана на идее разделить популяцию на подгруппы на основе некоторой меры сходства, которая определяется расстоянием между индивидами в пространстве поиска. Пригодность индивидов уменьшается пропорционально числу похожих членов популяции, тем самым ослабляя селективное давление в нишах с плотным скоплением индивидов [291, 292]. Пригодность индивида при использовании метода разделения пригодности определяется следующим образом (4.1)-(4.3):

$$fitness'_i = \frac{fitness_i}{m_i}, \quad (4.1)$$

$$m_i = \sum_{j=1}^{PopSize} sharing(d_{ij}), \quad (4.2)$$

$$sharing(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^\alpha, & d_{ij} < \sigma_{share}, \\ 0, & \text{иначе} \end{cases} \quad (4.3)$$

где $fitness_i$ и $fitness'_i$ – значение пригодности i -го индивида до и после деления, $i = \overline{1, PopSize}$, m_i – счетчик ниши, $sharing$ – функция для деления пригодности, определяемая уровнем сходства индивидов i и j , d_{ij} – расстояние между индивидами i и j , σ_{share} – радиус ниши, α - параметр (обычно $\alpha = 1$).

Расстояние d_{ij} между индивидами может быть вычислено на уровне генотипа (мера Хемминга для бинарного ГА, евклидова метрика для вещественных переменных) или для фенотипа, на основе дополнительной информации о задаче и реализации решений. Радиус ниши обычно выбирается опытным путем на основе свойств задачи оптимизации или на основе числа оптимумов, которые необходимо идентифицировать (определяется аналитиком или ЛПР).

Обнуление пригодности (clearing). Эвристика основана на идее ограниченности ресурсов в каждой нише популяции. Лучший (сильнейший) индивид получает ресурс, а остальные индивиды ниши погибают. В методе обнуления пригодности значение пригодности лучшего в нише сохраняется, значения пригодности других индивидов в нише обнуляются (предполагается, что пригодность имеет неотрицательные значения и максимизируется) [293]. Процедура обнуления пригодности может быть выполнена, используя следующий алгоритм (Алгоритм 5):

Алгоритм 5. Процедура обнуления пригодности.

Вход: текущая популяция, радиус ниши ($\sigma_{clearing}$), емкость каждой ниши ($Kappa$).
Отсортировать популяцию по убыванию пригодности.

Повторять для i от 1 до $PopSize$:

Если ($fitness_i > 0$), **то**:

 Установить счетчик «победителей» ниши $nWin = 1$.

Повторять для j от $(i + 1)$ до $PopSize$:

Если ($fitness_j > 0$) и ($d_{ij} < \sigma_{clearing}$), **то**:

Если ($nWin < Kappa$), то:
 ($nWin = nWin + 1$),
 иначе
 ($fitness_j = 0$).

Выход: значения пригодностей индивидов в популяции.

Метод контроля скопления индивидов (crowding). Эвристика базируется на аналогии с естественными процессами борьбы индивидов за ресурсы, когда похожие особи вынуждены бороться за ресурс одной ниши, а менее похожие друг на друга особи занимают разные ниши и не участвуют в борьбе. В итоге популяция достигает некоторого разнообразия и находится в равновесии. В методе контроля скопления в ЭА потомки сравниваются со случайно выбранной группой индивидов CF (фактор скопления, *crowding factor*) и замещают наиболее похожих индивидов [271]. Различают детерминированный (*deterministic crowding*) и вероятностный (*probabilistic crowding*) контроль скопления.

В работе [294] было показано, что контроль с использованием фактора скопления, предложенный Де Йонгом в [271] неэффективен, была предложена модификация метода – детерминированный контроль скопления, который может быть представлен следующим алгоритмом (Алгоритм б):

Алгоритм б. Процедура детерминированного контроля скопления.

Вход: текущая популяция.

Повторять $PopSize/2$ раз:

Выбрать пару индивидов-родителей $parent_1$ и $parent_2$ случайно без замещения.

Применить оператор рекомбинации к родителям и получить потомков $offspring_1$ и $offspring_2$.

Применить оператор мутации к потомкам и получить $offspring_1^m$ и $offspring_2^m$.

Если $[d(parent_1, offspring_1^m) + d(parent_2, offspring_2^m) \leq d(parent_1, offspring_2^m) + d(parent_2, offspring_1^m)]$, то:

Если $fitness(offspring_1^m) \geq fitness(parent_1)$, то:
 $offspring_1^m$ замещает $parent_1$.

Если $fitness(offspring_2^m) \geq fitness(parent_2)$, то:
 $offspring_2^m$ замещает $parent_2$.

иначе

Если $fitness(offspring_2^m) \geq fitness(parent_1)$, то:
 $offspring_2^m$ замещает $parent_1$.

Если $fitness(offspring_1^m) \geq fitness(parent_2)$, то:
 $offspring_1^m$ замещает $parent_2$.

Выход: новая популяция.

Поскольку в детерминированном контроле скопления всегда отдается предпочтение индивиду с большей пригодностью, ЭА эффективно локализует глобальный оптимум, но может пропустить некоторые локальные. В [295] предложена модификация подхода – метод вероятностного контроля скопления, который для двух похожих индивидов x_1 и x_2 использует случайный выбор, вероятность которого пропорциональна пригодности индивида. Вероятность $individual_1$ заменить $individual_2$ определяется как (4.4):

$$p_{replace}(x_1) = \frac{fitness(x_1)}{fitness(x_1) + fitness(x_2)}. \quad (4.4)$$

Метод формирования ниш на основе кластеризации (*clustering-based niching*), предложенный в [296] избавляет от необходимости настройки параметра радиуса ниши σ_{share} . Более того, сформированные таким образом ниши могут иметь различный размер и форму. Поскольку число кластеров априори неизвестно, в данном подходе используется быстрая адаптивная версия классического алгоритма k -средних, предложенного МакКуином [297]. В адаптивном алгоритме кластеры могут объединяться в один, если расстояние между их центроидами меньше заданного порога d_{min} , либо анализируемая точка может сформировать новый кластер, если расстояние между ней и ближайшим центроидом больше порога d_{max} . Алгоритм формирования ниш на основе кластеризации имеет следующий вид (Алгоритм 7):

Алгоритм 7. Процедура формирования ниш на основе кластеризации.

Вход: текущая популяция, начальное число кластеров (ниш) k , пороговые значения d_{min} и d_{max} .

Отсортировать популяцию по убыванию пригодности.

Сформировать k начальных кластеров из первых k индивидов (k лучших), индивиды являются центроидами.

Вычислить расстояния между всеми парами текущих кластеров.

Повторять пока расстояние между ближайшими кластерами меньше d_{min} :

 Объединить два ближайших кластера в один.

 Вычислить новые положения центроидов.

Вычислить расстояния между всеми парами текущих центроидов.

Выполнить для оставшихся ($PopSize - k$) индивидов:

Отнести текущего индивида к кластеру с ближайшим центроидом, пересчитать положение этого центроида.

Если расстояние между новым и ближайшим центроидами меньше d_{min} , **то**:

Объединить кластеры и пересчитать положение нового центроида.

Повторять объединение кластеров, пока все центроиды не будут на расстоянии больше d_{min} .

иначе

Если расстояние между новым и ближайшим центроидами больше d_{max} , **то**:

Создать новый кластер, состоящий из одного текущего индивида.

Перераспределить индивидов по кластерам (нишам) с полученными центроидами.

Выход: число и состав кластеров (ниш).

Итоговая пригодность индивидов в методе формирования ниш на основе кластеризации определится как (4.5):

$$fitness'_i = \frac{fitness_i}{n_c \cdot \left(1 - \left(\frac{d_{i,c}}{2d_{max}}\right)^\alpha\right)}, \quad (4.5)$$

где n_c – число индивидов в нише c , содержащей индивида i , $d_{i,c}$ – расстояние между индивидом i и его центроидом c , d_{max} – максимально допустимое расстояние между индивидом и его центроидом, α - параметр.

Ограниченная турнирная селекция (restricted tournament selection, RTS) является объединением идей традиционной турнирной селекции и метода контроля скопления с использованием фактора скопления. В данном подходе случайно выбранная пара родителей производит потомков, которые сравниваются со случайно выбранной подгруппой индивидов из популяции размера w (размер окна, *window size*). Каждый из потомков участвует в турнире с ближайшим (наиболее похожим) индивидом из подгруппы, победитель остается в популяции [297].

В работе [299] проводилось сравнение различных подходов мультимодальной оптимизации, по результатам можно сделать вывод, что эффективность базовых эвристик меняется от задачи к задаче, и нельзя выделить единственную эвристику, эффективную для множества задач. Аналогичный вывод можно сделать по результатам конкурсов IEEE CEC по мультимодальной

оптимизации. Ведущие алгоритмы в данной области имеют большой разброс оценок эффективности на множестве эталонных тестовых задач [300]. Одним из решений данной проблемы является использование селективной гиперэвристики, которая бы выбирала и комбинировала базовые эвристики под конкретную задачу.

4.2 Селективная гиперэвристика синтеза ЭА идентификации множества экстремумов

Гиперэвристика для задач идентификации множества экстремумов (1.1), (1.5) аналогично методу 2 в главе 3 объединяет идеи метода портфолио алгоритмов и метода адаптации вероятностей на уровне популяции. При этом гиперэвристика может применяться как онлайн, например, при однократном решении новой задачи, так и офлайн для обобщения результатов на множество задач, например, для повторяющихся или однотипных задач некоторой предметной области.

Множество базовых эвристик, включенных в разрабатываемую селективную гиперэвристику включает: разделение пригодности, обнуление пригодности, детерминированный контроль скопления, вероятностный контроль скопления, формирование ниш на основе кластеризации и ограниченную турнирную селекцию.

Для управления выбором эвристик на очередном этапе решения задачи оптимизации необходимо определить способ оценки эффективности применения базовых эвристик. Для задачи идентификации множества экстремумов (1.1), (1.5) показатели эффективности должны учитывать сколько оптимумов было обнаружено и как распределены индивиды популяции в пространстве поиска. Достаточно полный обзор и сравнение метрик можно найти, например, в [276, 301]. Многие из метрик используют информацию об истинном положении оптимумов: процент идентифицированных оптимумов, точность идентификации оптимумов и другие. Очевидно, что подобные показатели могут быть использованы только для решения эталонных тестовых задач и хорошо изученных прикладных задач. Для

произвольных задач с алгоритмически заданными целевыми функциями требуется разработка универсальных подходов к оценке эффективности.

Поскольку в сложных задачах оптимизации истинное количество и положение оптимумов не известно, то выводы о качестве аппроксимации множества экстремумов делаются на основе анализа поведения ЭА. Как отмечалось ранее, в процессе своей работы ЭА сходятся в зону притяжения одного из оптимумов задачи, таким образом, по скоплению индивидов можно сделать вывод о локализации зоны притяжения одного из оптимумов. С другой стороны, поддержка разнообразия в популяции и запрет на скопление индивидов в одной области может стимулировать ЭА обнаружить зоны притяжения нескольких оптимумов. Поэтому показатели качества также должны отражать распределение популяции в пространстве поиска.

Для оценки эффективности применения базовых эвристик в селективной гиперэвристике для идентификации множества экстремумов в данном исследовании предложены следующие критерии:

– критерий *BasinRatio* (4.6)-(4.8) показывает долю областей притяжения оптимумов, обнаруженных каждой из эвристик;

– критерий *SDNN* (*sum of distances to nearest neighbour*) (4.9) – (4.11) вычисляет сумму расстояний до ближайшего соседа для всех членов популяции, критерий используется как штраф за скопление индивидов в одной области.

Показатель *BasinRatio* является аналогом метрики, оценивающей процент идентифицированных оптимумов, за исключением того, что вместо истинного положения оптимумов, используется информация о зонах притяжения оптимумов. Выводы об идентификации зоны притяжения можно сделать по скоплению индивидов в предположении, что ЭА сходятся в точку оптимума. Число различных областей притяжения определяется по результатам работы гиперэвристики в целом. Далее оценивается какая доля этих областей локализована каждой из гиперэвристик.

$$BasinRatio(h_i) = \frac{l(h_i)}{|B|}, \quad (4.6)$$

где $BasinRatio$ – доля областей притяжения оптимумов, идентифицированных при применении эвристики h_i ($i = \overline{1, |H|}$), $l(h_i)$ – число областей притяжения, обнаруженных i -й эвристикой, $|B|$ – число областей притяжения, обнаруженных всей популяцией (гиперэвристикой), B – множество медоидов, полученное в ходе решения задачи кластеризации индивидов, определяющих области притяжения идентифицированных оптимумов.

$$l(h_i) = \sum_{j=1}^{|B|} \min\{1, \sum_{\substack{x \in subPop_i \\ x \neq b_j}} basin(x, b_j)\}, \quad (4.7)$$

$$basin(x, b) = \begin{cases} 1, & \text{если } x \in bSet(b) \\ 0, & \text{если } x \notin bSet(b) \end{cases} \quad (4.8)$$

где $basin(x, b)$ – функция, определяющая нахождение индивида x из подпопуляции $subPop_i$, $i = \overline{1, |H|}$ в зоне притяжения $b_j \in B$, $bSet(b)$ – множество индивидов, находящихся в зоне притяжения b .

Для использования показателя $BasinRatio$ необходимо выбрать способ идентификации областей притяжения оптимумов. Для задач аппроксимации множества экстремумов с вещественными переменными области притяжения могут быть определены с помощью различных процедур кластеризации, например, алгоритмов Джарвиса-Патрика, ближайшего лучшего и других [302, 303].

В данной работе для задач аппроксимации множества экстремумов с бинарным представлением предложен следующий подход. Используется общая популяция (объединение подпопуляций всех эвристик). Для каждого индивида рассматривается предопределенное количество его ближайших по метрике Хемминга соседей. Если пригодность рассматриваемого индивида лучше, чем у его ближайших соседей, то он помечается как центр идентифицированной области притяжения. Псевдокод процедуры имеет следующий вид (Алгоритм 8):

Алгоритм 8. Определение областей притяжения оптимумов для задач с бинарными переменными.

Вход: популяция, оценки пригодности индивидов.

Положить $B = \emptyset$.

Повторять для $i = \overline{1, PopSize}$:

Определить множество S_{nn} ближайших к i соседей по метрике Хемминга.
Если $fitness_i > fitness_j, \forall j \in S_{nn}$, то:

$$B = B + \{x_i\}.$$

Выход: множество центров B областей притяжения оптимумов.

Критерий $SDNN$ может быть вычислен, используя следующие формулы:

$$SDNN(h_i) = \sum_{j=1}^{subPop_i} d_{nn}(x_j, subPop_i), \quad (4.9)$$

$$d_{nn}(x_j, subPop_i) = \min_{\substack{\forall k \in subPop_i \\ k \neq j}} d_{j,k}, \quad (4.10)$$

$$\overline{SDNN}(h_i) = \frac{SDNN(h_i)}{\max_{j=1, |H|} SDNN(h_j)}, \quad (4.11)$$

где $SDNN$ – сумма расстояний до ближайшего соседа, вычисленная для членов подпопуляции эвристики h_i , d_{nn} – расстояние d (конкретная метрика зависит от пространства поиска) до ближайшей соседней точки в подпопуляции, \overline{SDNN} – нормированное значение.

Итоговый показатель K_{MMO} для оценки эффективности эвристики h_i в селективной гиперэвристике решения задач идентификации множества экстремумов является сверткой критериев $BasinRatio$ и \overline{SDNN} (4.12):

$$K_{MMO}(h_i) = \alpha \cdot BasinRatio(h_i) + (1 - \alpha) \cdot \overline{SDNN}(h_i), \quad (4.12)$$

где $\alpha \in [0,1]$ – весовой коэффициент, определяющий вклад каждого из критериев в сумму, в случае отсутствия предпочтений по критериям $\alpha = 0.5$.

Гиперэвристика синтеза ЭА для решения задач идентификации множества экстремумов на основе метода портфолио алгоритмов и метода адаптации вероятностей на уровне популяции имеет следующий вид:

Метод 3. Селективная гиперэвристика синтеза ЭА идентификации множества экстремумов.

Вход: множество H базовых эволюционных эвристик и метаэвристик, критерий оценки эффективности ЭА.

Инициализация: Случайная популяция решений делится на равные подпопуляции, подпопуляциям назначаются используемые эвристики.

Повторять, пока не выполнено условие останова:

Повторять заданное число итераций (период адаптации):

Решать задачу оптимизации, применяя выбранные эвристики к соответствующим подпопуляциями.

Оценить эффективность применения каждой из эвристик по числу идентифицированных оптимумов *BasinRatio* и скоплению индивидов *SDNN*, используя (4.6) и (4.9).

Перераспределить размеры подпопуляций в пользу более эффективных эвристик.

Выполнить миграции индивидов – добавить в подпопуляции идентифицированные центры областей притяжения, заменив наиболее близких к ним индивидов.

Выход: множество точек, аппроксимирующих распределение оптимумов в пространстве поиска.

Перераспределение размеров популяций происходит по следующей схеме: определяется эвристика-победитель с наибольшим значением показателя $K_{ММО}$, размеры подпопуляций остальных эвристик уменьшаются на заданное число индивидов, размер подпопуляции эвристики-победителя увеличивается на сумму полученных от других эвристик индивидов. Чтобы эвристики, продемонстрировавшие низкую эффективность на прошлых этапах решения задачи не исключались из работы, устанавливается минимальное пороговое значение для размера подпопуляции.

На шаге миграции, обеспечивающем кооперацию эвристик, необходимо уравнивать стартовые точки подпопуляций на следующий период адаптации. Традиционная схема «лучший замещает худшего» приводит к сходимости в зону притяжения единственного лучшего найденного оптимума, поэтому она не пригодна для задач данного класса. Поскольку после каждого периода адаптации мы имеем аппроксимацию множества экстремумов (индивиды в множестве B , определяющие зоны притяжения оптимумов), в методе 3 в миграции участвуют индивиды множества B , которые замещают наиболее близких к ним индивидов в каждой подпопуляции.

4.3 Экспериментальное исследование предложенного метода

Для исследования эффективности метода 3 были выбраны два набора эталонных тестовых задач идентификации множества экстремумов: набор задач с бинарными переменными и набор задач с конкурса по мультимодальной оптимизации, предложенный в рамках конференции IEEE CEC 2013 и используемый на конкурсах IEEE CEC и GECCO по настоящее время.

Первый набор включает шесть задач с бинарными переменными, предложенные профессором Шугантоном на основе работ Голдберга, Деба и других. Задачи в наборе имеют несколько глобальных и множество локальных оптимумов и являются десептивными (обманчивыми) [304]. Данные функции интересны тем, что в отличие от функций с вещественными переменными не имеют интуитивно понятной топологии пространства поиска и свойств ландшафта целевой функции, поэтому исследователям сложно создать специальный алгоритм для подобных задач. Более того, исследование на тестовых задачах с бинарными переменными необходимо, т.к. переменные многих задач оптимизации сложных систем выражены в разных шкалах и могут быть бинаризованы для последующего использования бинарного ГА.

Второй набор включает 20 многоэкстремальных задач, основанных на 12 целевых функциях, решаемых с различным числом целевых переменных [132]. Часть функций является масштабируемыми, число их оптимумов увеличивается с ростом размерности. Часть функций является композицией других, данные функции обладают комбинацией свойств, затрудняющих решение задачи идентификации множества экстремумов (пример таких функций для размерности 2 показан на рисунке 4.2).

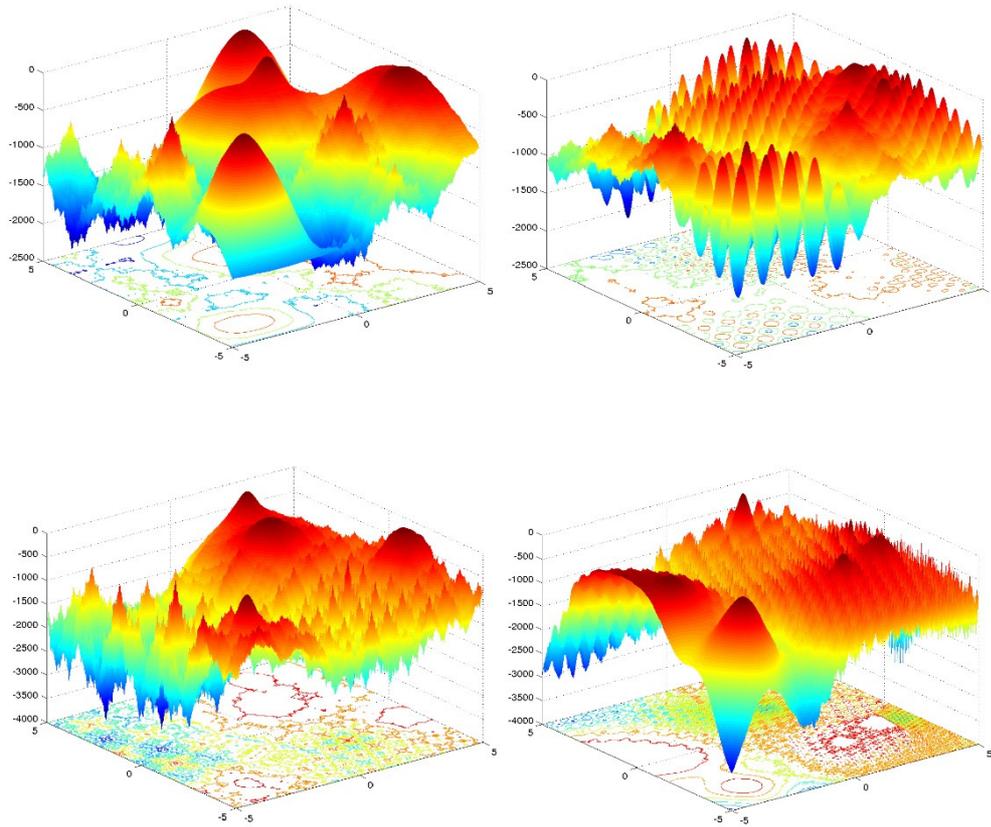


Рисунок 4.2 – Пример мультимодальных функций набора IEEE CEC [132]

Для оценки эффективности и сравнения алгоритмов при решении задач мультимодальной оптимизации используются следующие критерии:

– процент оптимумов, идентифицированных алгоритмом с заданной точностью PR (*peak ratio*) (4.13):

$$PR = \frac{|\{d_{nn}(q,x) \leq \varepsilon, q \in Q, x \in Pop\}|}{|Q|}, \quad (4.13)$$

где Q – множество известных оптимумов, которые необходимо идентифицировать, d_{nn} – расстояние от оптимума q до ближайшего индивида x из популяции Pop ;

– процент успешных запусков алгоритма, когда были найдены все оптимумы задачи SR (*success rate*).

В случае задач с бинарными переменными уровень точности теряет смысл, т.е. при решении прикладных задач, в общем случае, бинаризация не сохраняет структуру исходного пространства поиска. Поэтому критерий PR для задач с бинарными переменными имеет вид (4.14):

$$PR = \frac{|\{ \|q-x\|_H=0, q \in Q, x \in Pop \}|}{|Q|}, \quad (4.14)$$

где $\|q-x\|_H$ – расстояние Хемминга между оптимумом q и ближайшим индивидом x .

В дополнение к критерию PR для задач с бинарными переменными был добавлен критерий PD (*peak distance*), который показывает среднее расстояние Хемминга от известного оптимума до ближайшего индивида в популяции (4.15):

$$PD = \frac{1}{|Q|} \sum_{q \in Q} \|q-x\|_H. \quad (4.15)$$

Описание тестовых задач с бинарными переменными приведено в Приложении В. На текущий момент в открытой печати представлена единственная публикация с анализом эффективности методов мультимодальной оптимизации на бинарных задачах, в которой исследуется подход, представляющий собой ансамбль различных алгоритмов на основе метода ниш (*ensemble of niching algorithms, ENA*) [304].

Программная реализация алгоритмов для проведения численных исследований выполнена на языке Python 3.7 в среде научного программирования Spyder с использованием библиотек NumPy и SciPy. Исходные коды эталонных тестовых задач IEEE CEC и процедуры оценки критериев эффективности алгоритмов на языке Python представлены авторами бенчмарка.

Параметры метода 3 для исследования эффективности на множестве тестовых задач с бинарными переменными представлены в таблице 4.1. Значения параметров подобраны экспериментально в ходе предварительных экспериментов.

Эффективность метода 3 оценивалась по значениям критериев PR , SR и PR , усредненных по 50 независимым запускам. Для ENA авторы представляли только значения по критерию SR . Максимальное число вычислений целевой функции в одном независимом прогоне для метода 3 и ENA установлено одинаковое для честного сравнения алгоритмов.

Таблица 4.1 – Параметры метода 3 для исследования эффективности на множестве тестовых задач с бинарными переменными

Параметр	Значение
Максимальное число вычислений целевой функции	50000
Число независимых запусков алгоритма	50
Размер популяции (размер популяции в ENA)	200 (500)
Число подпопуляций	6
Минимальное значение размера подпопуляции	10
Период адаптации (число поколений)	10
Радиус ниши в методе обнуления пригодности*	3
Емкость ниши в методе обнуления пригодности*	2
Радиус ниши в методе разделения пригодности*	4
Параметр α в методе разделения пригодности*	2
Параметр d_{min} в методе формирования ниш на основе кластеризации*	2
Параметр d_{max} в методе формирования ниш на основе кластеризации*	5
Размер окна в методе ограниченной турнирной селекции*	5

* значения параметров базовых эвристик установлены опытным путем в ходе экспериментов.

Результаты решения тестовых задач с бинарными переменными, усреднённые по всему набору представлены в таблице 4.2. Диаграммы размаха оценок показаны на рисунках 4.3-4.5. Поскольку для алгоритма ENA авторы представляли только итоговые усреднённые оценки по критерию SR , ранжирование по данному критерию проводилось традиционным способом, для критериев PR и PD ранжирование осуществлялось с применением метода глубокого статистического сравнения. Результаты ранжирования показаны на рисунках 4.6-4.8. Детальные результаты по задачам с бинарными переменными представлены в Приложении Г.

Таблица 4.2 – Результаты численных экспериментов при решении задач аппроксимации множества экстремумов с бинарными переменными

Критерий	ENA	Обнуление пригодности	Разделение пригодности	Формирование ниш на основе кластеризации	Ограниченная турнирная селекция	Детерминированный контроль скопления	Вероятностный контроль скопления	Среднее по базовым эвристикам	Метод 3
PR	-	0.92	0.87	0.91	0.95	0.92	0.76	0.89	1.00
SR	1.00	0.92	0.89	0.90	0.93	0.89	0.76	0.88	1.00
PD	-	2.21	2.46	2.22	1.40	2.47	3.34	2.35	0.00

Как видно из таблицы 4.2, исходно бинарные тестовые задачи не представляют трудности для метода 3 и ансамбля алгоритмов ENA, оба подхода находят все известные глобальные оптимумы в каждом запуске. При этом эффективность базовых эвристик, применяемых по отдельности, может быть низкой. В таблице 4.3 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов с применением теста Манна-Уитни-Уилкоксона. В таблице показано, сколько раз на задачах с бинарными переменными метод 3 был лучше, сравним или хуже ЭА с базовыми эвристиками мультимодальной оптимизации. Метод 3 во всех случаях превосходит среднее по базовым эвристикам, следовательно, применение метода 3 предпочтительнее случайного выбора одной из базовых эвристик.

Набор эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC включает 20 многоэкстремальных задач, основанных на 12 целевых функциях, решаемых с различным числом целевых переменных. При этом оценки *PR* и *SR* вычисляются для 5 уровней точности: 1E-01, 1E-02, 1E-03, 1E-04 и 1E-05. Настройки алгоритмов аналогичны приведенным в таблице 4.1 за исключением параметров, заданных условиями конкурса IEEE CEC, а именно, число независимых прогонов – 50, максимальное число вычислений целевой функции в прогоне равно 5E+04 для функций 1-5 размерности 1 и 2, 2E+05 для функций 6-11 размерности 2 и 4E+05 для функций 6-12 размерности 3 и выше.

Таблица 4.3 – Результаты теста Манна-Уитни-Уилкоксона на задачах аппроксимации множества экстремумов с бинарными переменными

Метод 3	Обнуление пригодности	Разделение пригодности	Формирование ниш на основе кластеризации	Ограниченная турнирная селекция	Детерминированный контроль скопления	Вероятностный контроль скопления
Критерий <i>PR</i>						
Лучше	5	5	4	3	4	6
Нет статистически значимой разницы	1	1	2	3	2	0
Хуже	0	0	0	0	0	0
Критерий <i>SR</i>						
Лучше	4	4	5	3	6	6
Нет статистически значимой разницы	2	2	1	3	0	0
Хуже	0	0	0	0	0	0
Критерий <i>PD</i>						
Лучше	5	6	5	3	5	6
Нет статистически значимой разницы	1	0	1	3	1	0
Хуже	0	0	0	0	0	0

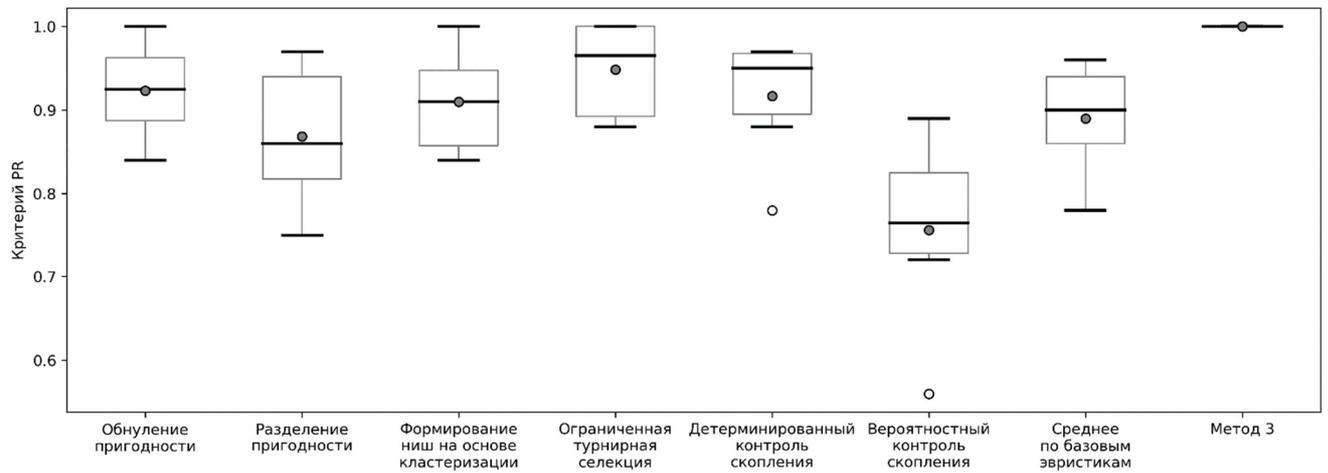


Рисунок 4.3 – Диаграмма размаха оценок по критерию *PR*

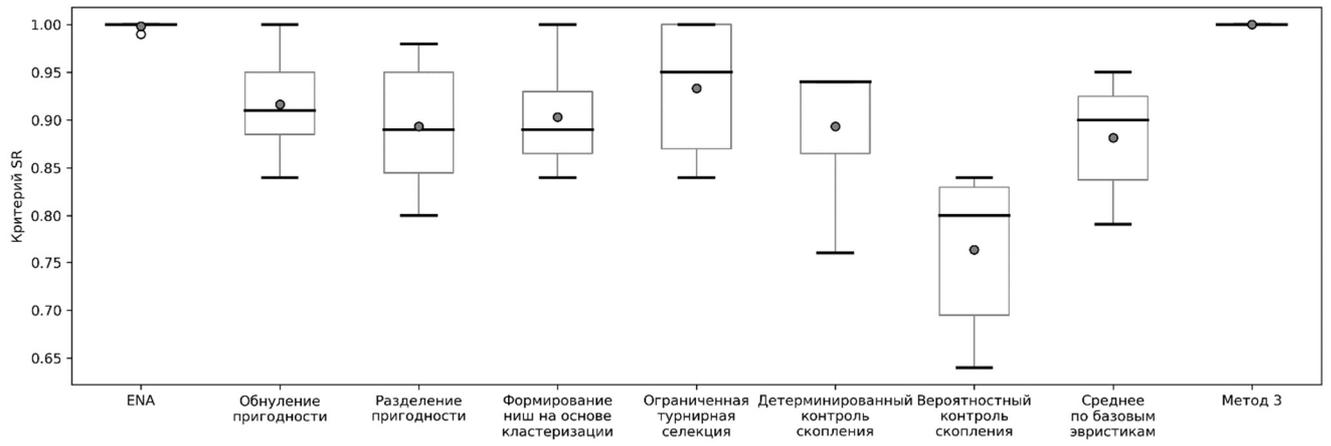


Рисунок 4.4 – Диаграмма размаха оценок по критерию *SR*

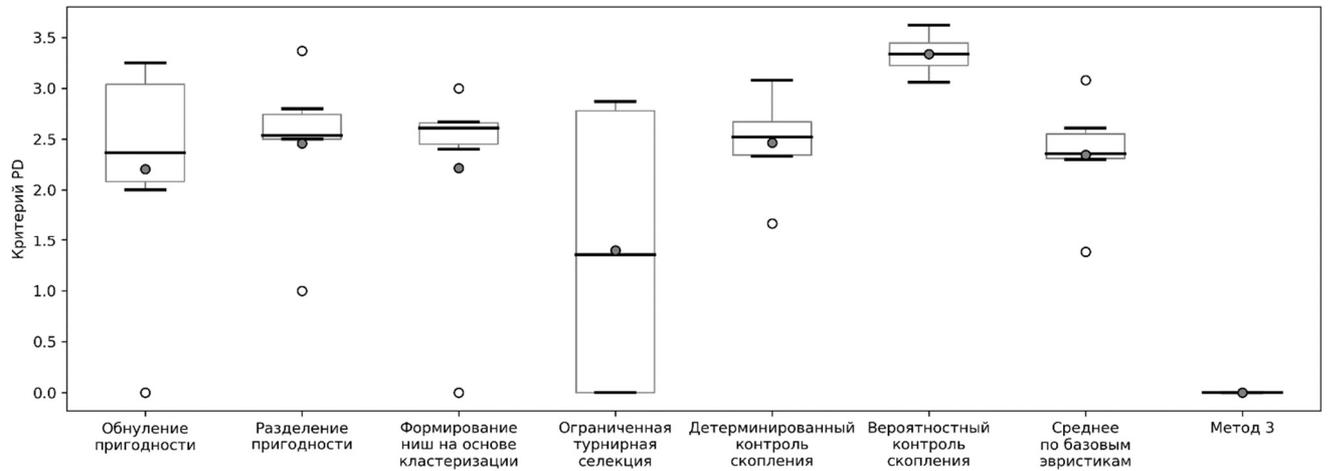


Рисунок 4.5 – Диаграмма размаха оценок по критерию *PD*

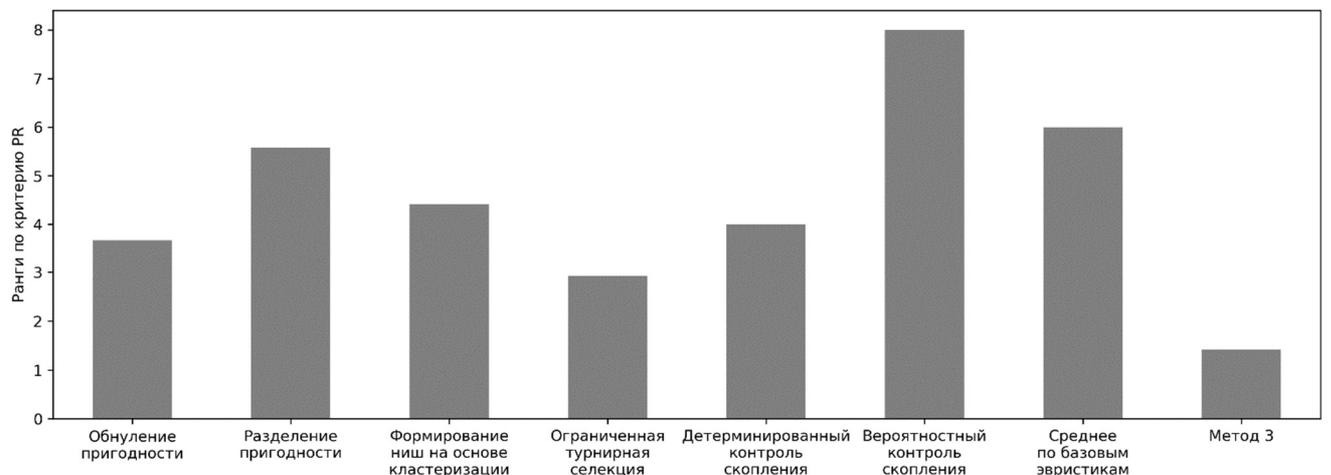
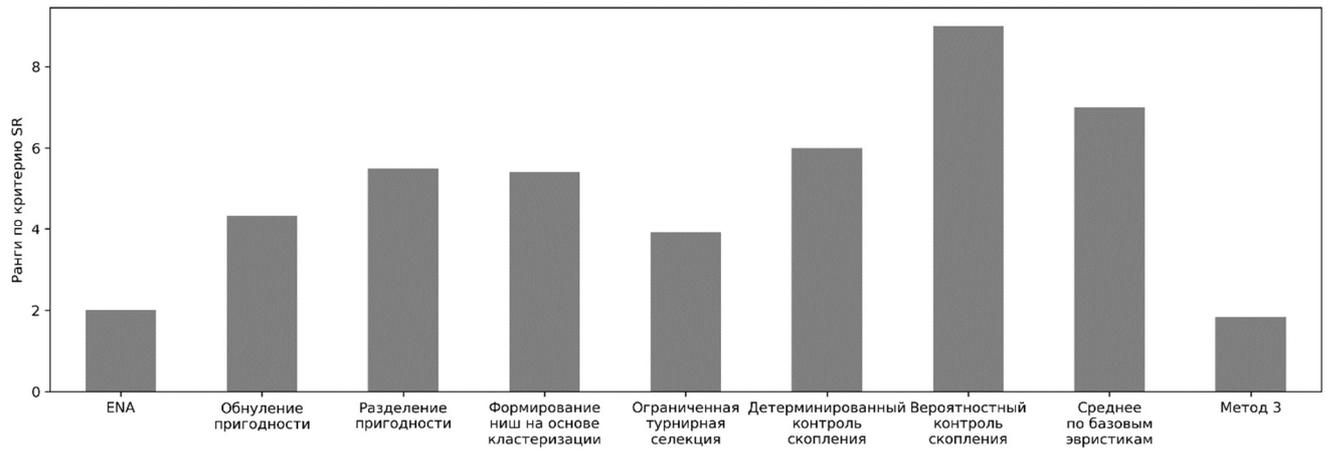
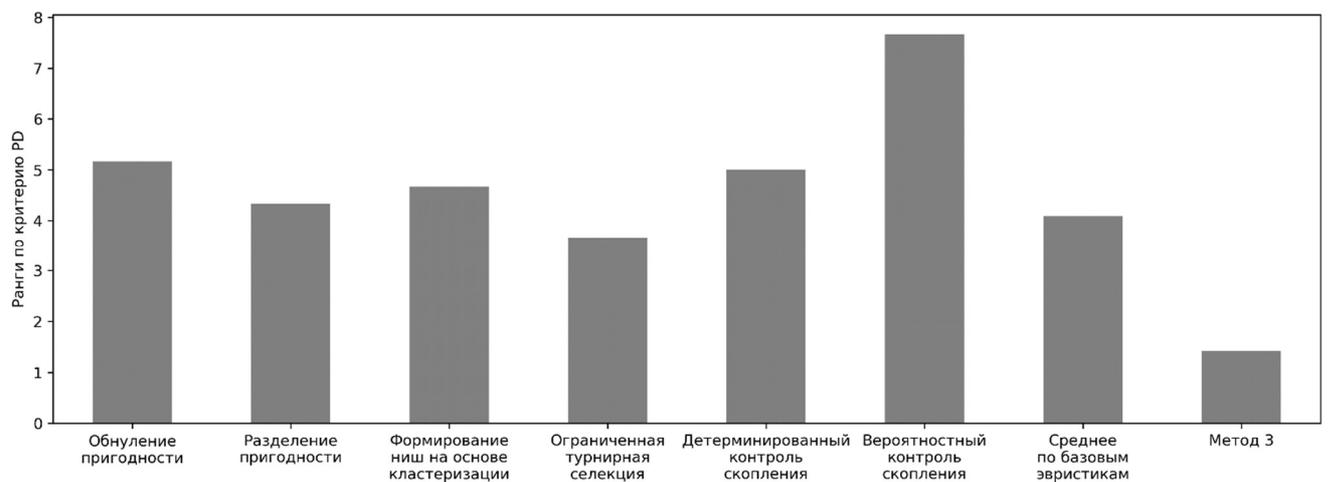


Рисунок 4.6 – Ранжирование по критерию *PR*

Рисунок 4.7 – Ранжирование по критерию *SR*Рисунок 4.8 – Ранжирование по критерию *PD*

Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC методом 3 и ЭА использующими одну базовую эвристику представлены в таблице 4.4. Результаты усреднены по набору задач, ранжирование проводилось с использованием метода глубокого статистического сравнения. Разброс оценок и ранги по каждому из критериев показаны на рисунках 4.9-4.12.

Таблица 4.4 – Результаты численных экспериментов при решении эталонных задач аппроксимации множества экстремумов IEEE CEC

Алгоритм	Обнуление пригодности		Разделение пригодности		Формирование ниш на основе кластеризации		Ограниченная турнирная селекция		Детерминированный контроль скопления		Вероятностный контроль скопления		Среднее по базовым эвристикам		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1E-01	0.63	0.44	0.53	0.43	0.54	0.38	0.58	0.50	0.59	0.43	0.51	0.42	0.56	0.43	0.80	0.60
1E-02	0.58	0.29	0.47	0.23	0.47	0.24	0.56	0.35	0.54	0.27	0.39	0.27	0.50	0.28	0.71	0.38
1E-03	0.54	0.26	0.50	0.26	0.48	0.21	0.55	0.29	0.59	0.28	0.48	0.25	0.52	0.26	0.70	0.36
1E-04	0.52	0.29	0.42	0.22	0.47	0.24	0.59	0.26	0.48	0.27	0.38	0.18	0.48	0.24	0.68	0.33
1E-05	0.45	0.22	0.34	0.16	0.41	0.19	0.43	0.24	0.42	0.21	0.34	0.17	0.40	0.20	0.57	0.28
Средняя	0.54	0.30	0.45	0.26	0.48	0.25	0.54	0.33	0.52	0.29	0.42	0.26	0.49	0.28	0.69	0.39
Ранг по PR	4.15		5.50		5.18		3.84		4.58		6.26		5.10		1.36	
Ранг по SR	4.44		5.14		5.26		3.81		4.66		5.50		4.97		2.20	

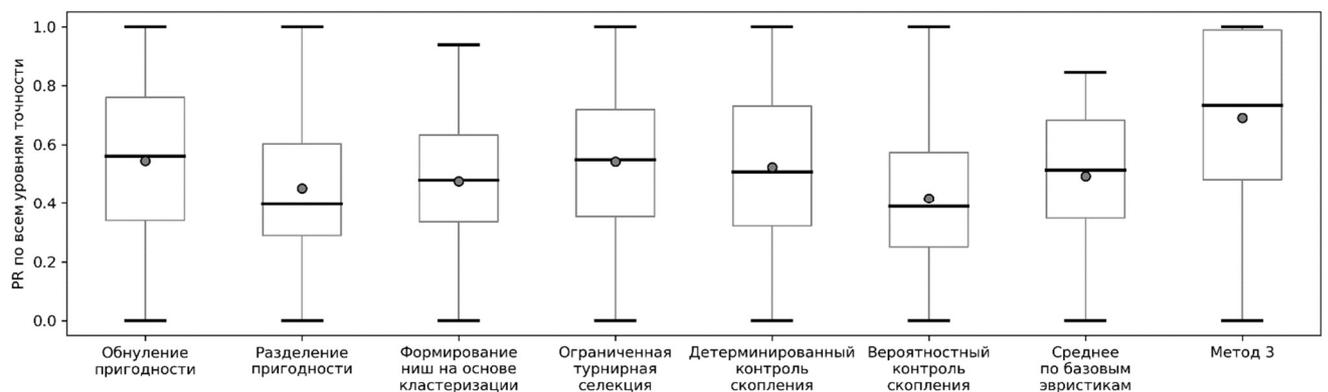
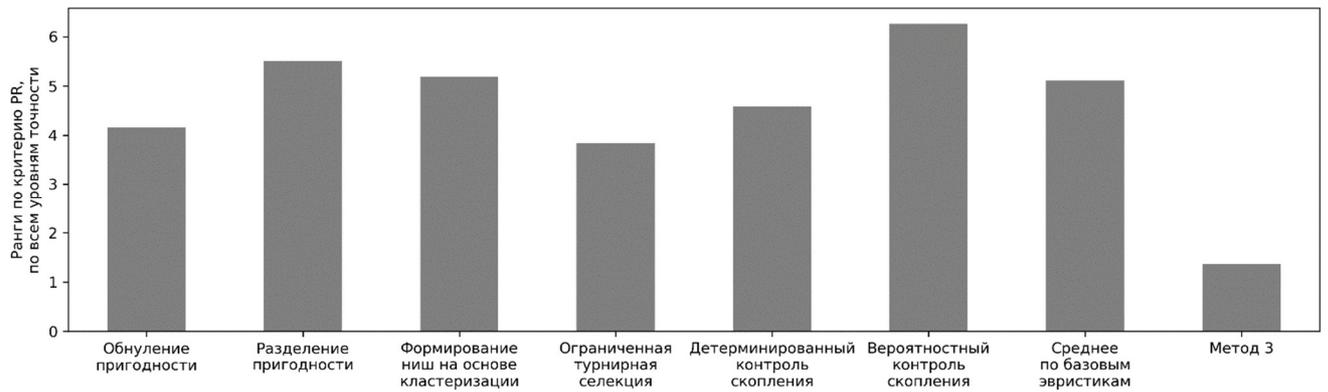
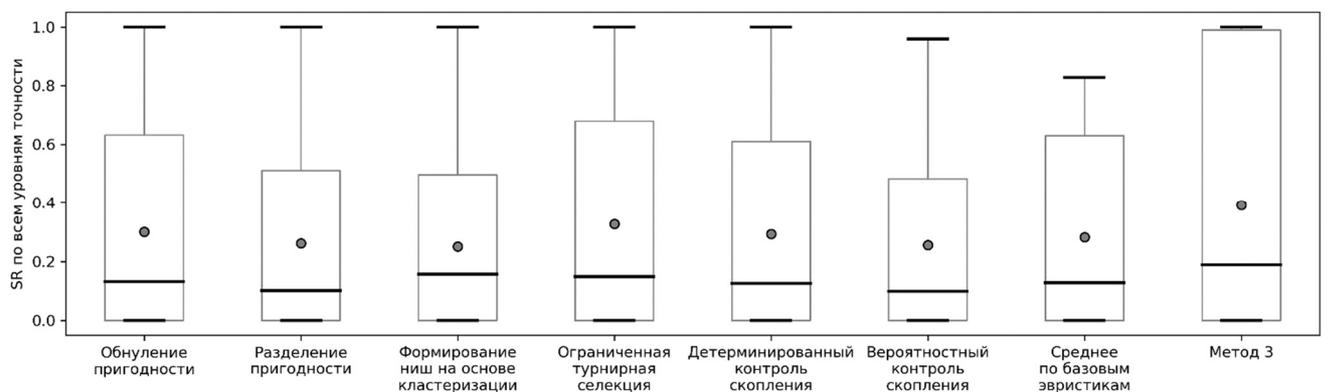
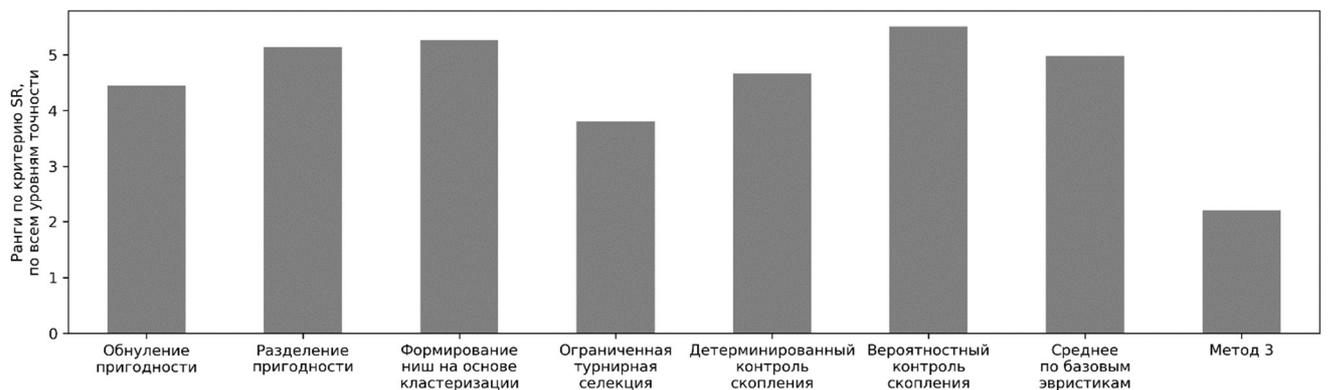


Рисунок 4.9 – Диаграмма размаха оценок по критерию *PR*

Рисунок 4.10 – Ранжирование по критерию PR Рисунок 4.11 – Диаграмма размаха оценок по критерию SR Рисунок 4.12 – Ранжирование по критерию SR

Как видно из таблицы 4.4, метод 3 превосходит по оценкам PR и SR ЭА с одной базовой эвристикой и среднее по базовым эвристикам (оценку случайного

выбора одной из них). В таблице 4.5 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов по всем уровням точности с применением теста Манна-Уитни-Уилкоксона.

Таблица 4.5 – Результаты теста Манна-Уитни-Уилкоксона на эталонных задачах аппроксимации множества экстремумов

Метод 3	Обнуление пригодности	Разделение пригодности	Формирование ниш на основе кластеризации	Ограниченная турнирная селекция	Детерминированный контроль скопления	Вероятностный контроль скопления
Критерий <i>PR</i>						
Лучше	77	84	82	75	76	90
Нет статистически значимой разницы	15	9	15	18	13	5
Хуже	8	7	3	7	11	5
Критерий <i>SR</i>						
Лучше	82	91	91	83	83	90
Нет статистически значимой разницы	9	5	5	7	9	6
Хуже	9	4	4	10	8	4

Результаты решения эталонных тестовых задач IEEE CEC были сравнены с ведущими алгоритмами мультимодальной оптимизации, которые побеждали и занимали призовые места на конкурсе: двухкритериальный подход с использованием метода ниш без параметров (*parameterless-niching-assisted bi-objective approach*, PNA-NSGA-II) [305], CMA-ES с использованием метода ниш на основе кластеризации по лучшим найденным индивидам (*niching the CMA-ES via nearest-better clustering*, NEA2) [306], дифференциальная эволюция на основе метода ниш с динамическим архивом (*dynamic archive niching differential evolution algorithm*, dADE\rand\1) [307], оптимизация по переменной решетке (*variable mesh optimization*, N-VMO) [308], дифференциальная эволюция (*differential evolution*, DE\rand\1) [309], дифференциальная эволюция на основе метода контроля скопления (*crowding-based differential evolution*, cDE\rand\1) [310] и

дерандомизированная самоадаптация в эволюционных стратегиях (*derandomized self-adaptation in evolution strategies*, CMA-ES) [311]. В таблице 4.6 представлены результаты сравнения, усредненные по набору эталонных задач, в приложении Г – детальные результаты по всем задачам и уровням точности. На рисунках 4.13-4.16 показаны диаграммы размаха для оценок *PR* и *SR* и результаты ранжирования алгоритмов. Поскольку, авторы методов представляют только итоговые результаты тестирования, ранжирование проводилось по традиционной схеме, а тест на статистическую значимость различий в оценках не выполнялся.

Таблица 4.6 – Результаты численных экспериментов при решении эталонных задач аппроксимации множества экстремумов IEEE CEC

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1E-01	0.75	0.52	0.81	0.44	0.86	0.63	0.88	0.81	0.60	0.30	0.71	0.38	0.73	0.40	0.80	0.60
1E-02	0.61	0.30	0.81	0.42	0.74	0.38	0.71	0.40	0.59	0.30	0.63	0.32	0.72	0.39	0.71	0.38
1E-03	0.60	0.30	0.80	0.42	0.73	0.37	0.69	0.32	0.58	0.30	0.58	0.28	0.72	0.38	0.70	0.36
1E-04	0.59	0.30	0.80	0.40	0.71	0.35	0.66	0.30	0.58	0.30	0.50	0.25	0.72	0.38	0.68	0.33
1E-05	0.53	0.27	0.75	0.37	0.65	0.31	0.57	0.28	0.55	0.30	0.45	0.21	0.68	0.36	0.57	0.28
Средняя	0.61	0.34	0.79	0.41	0.74	0.41	0.70	0.42	0.58	0.30	0.57	0.29	0.71	0.38	0.69	0.39
Ранг по PR	5.47		2.80		3.98		4.15		5.78		5.61		4.05		4.15	
Ранг по SR	4.95		3.96		4.21		4.33		5.11		5.07		4.39		3.94	

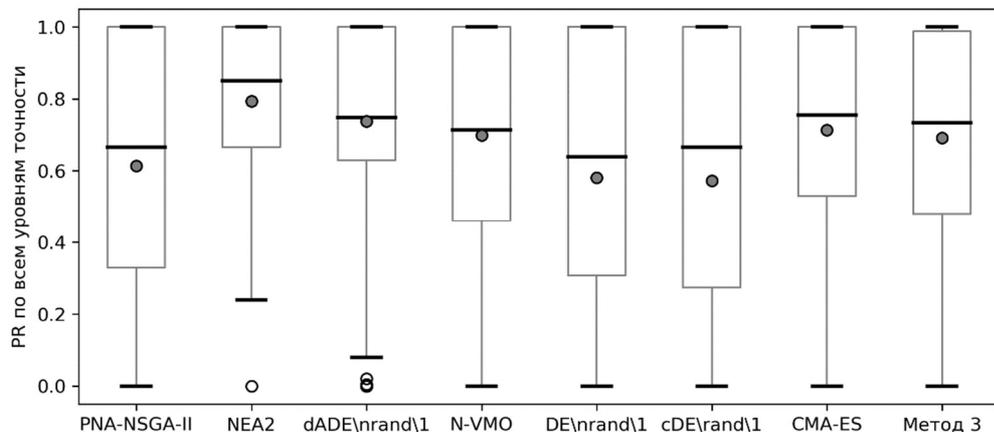


Рисунок 4.13 – Диаграмма размаха оценок по критерию *PR*

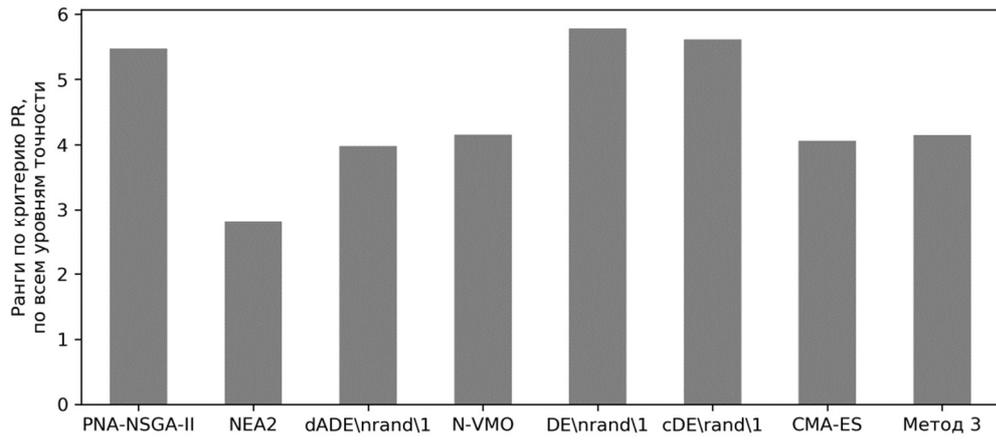


Рисунок 4.14 – Ранжирование по критерию *PR*

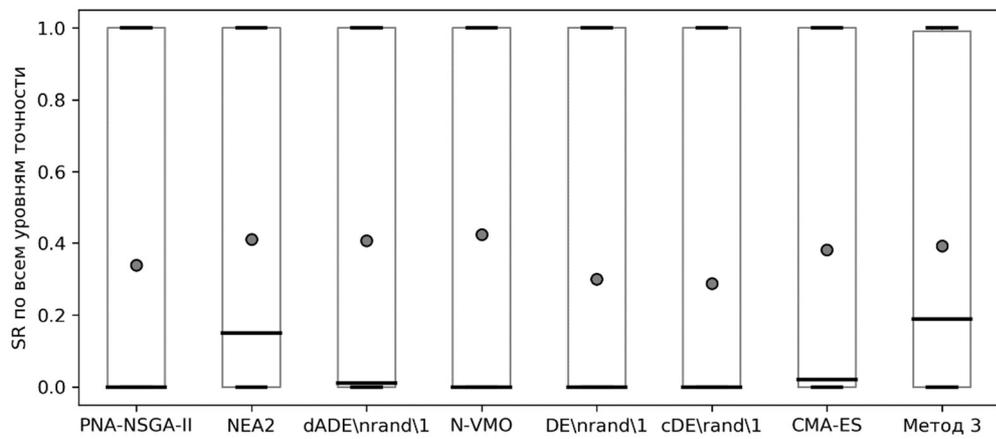


Рисунок 4.15 – Диаграмма размаха оценок по критерию *SR*

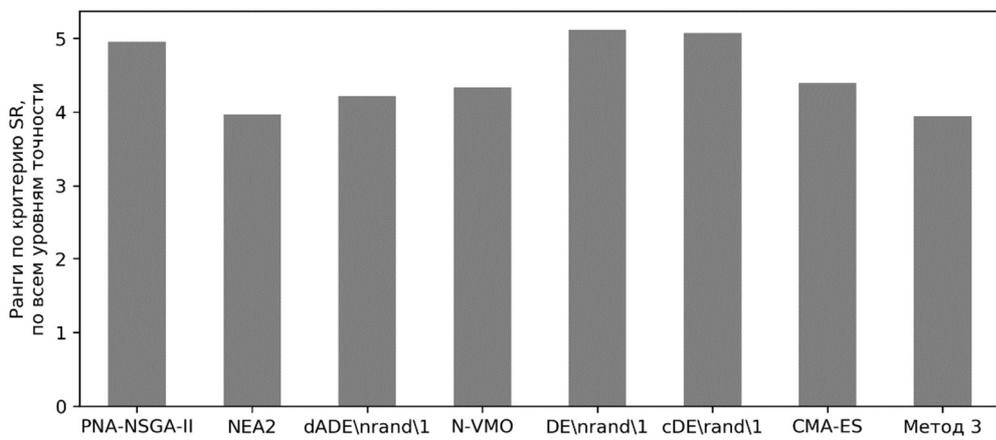


Рисунок 4.16 – Ранжирование по критерию *SR*

Как видно из результатов, метод 3 превосходит в среднем по оценке PR алгоритмы PNA-NSGA-II, $DE\backslash\text{rand}\backslash 1$ и $cDE\backslash\text{rand}\backslash 1$, сравним с N-VMO и уступает NEA2, $dADE\backslash\text{rand}\backslash 1$ и CMA-ES. По критерию SR : превосходит все подходы, кроме NEA2, с которым значения оценки сравнимы. Стоит отметить, что на конкурсе IEEE CEC 2013 алгоритмы NEA2, $dADE\backslash\text{rand}\backslash 1$, CMA-ES и N-VMO, заняли позиции с 1 по 4 соответственно. При этом метод 3 превосходит оценку случайного выбора одного из ведущих алгоритмов (среднее по оценкам всех алгоритмов) для всех уровней точности, кроме $1E-05$.

Известные алгоритмы мультимодальной оптимизации, обычно, являются глубокой модификацией ЭА, использующего одну или несколько базовых эвристик. Метод 3 оперирует множеством базовых эвристик, осуществляет выбор и их комбинирование в процессе решения конкретной задачи и конкретной ситуации в процессе поиска, а потому демонстрирует высокую эффективность при аппроксимации множества экстремумов.

4.4 Решение практических задач аппроксимации множества экстремумов при поддержке принятия решений при формировании оптимальной кредитной политики коммерческого банка и при управлении инвестициями производственного предприятия ОПК

Многие прикладные задачи поддержки принятия решения рассматриваются как задачи выбора подходящей альтернативы из множества допустимых альтернатив и могут быть сведены к оптимизационным постановкам, в которых целевые переменные описывают возможные альтернативы, а целевая функция отражает некоторый эффект выбора одной из них. Для многих сложных задач поддержки принятия решений аналитическое решение или полный перебор и попарное сравнение всех альтернативных вариантов затруднены или невозможны. В тоже время, традиционные методы оптимизации обеспечивают ЛПР единственным решением, к которому сходится алгоритм оптимизации, что

ограничивает анализ решений и устойчивость решения при реализации. Один из известных подходов – многокритериальная постановка задачи оптимизации, где целевые функции отражают различные требования к решению. Однако множество Парето, полученное таким образом, может содержать тривиальные решения с граничными значениями по критериям, которые не имеют практической ценности, но требуют дальнейшего анализа (например, множество Парето по критериям прибыль-риск содержит решение с нулевой прибылью и нулевым риском). В свою очередь в однокритериальной постановке различные глобальные и допустимые по значению локальные оптимумы отражают альтернативные способы решения задачи выбора. Таким образом, аппроксимация множества экстремумов может быть существенным подспорьем при решении прикладных задач поддержки принятия решения.

Для апробации метода 3 были решены практические задачи формирования оптимальной кредитной политики коммерческого банка и управления инвестициями производственного предприятия ОПК. Данные задачи хорошо изучены, и поэтому, есть возможность сопоставить результаты применения метода 3 с ранее полученными результатами. Рассмотрим постановки задач подробнее.

Задача формирования оптимальной кредитной политики коммерческого банка связана с выбором подмножества (кредитного портфеля) из множества кредитных заявок, максимизирующего прибыль, полученную в виде процентов за использование кредита, с учетом ограничений на свободные пассивы банка и суммарную рискованность портфеля. В [312] предложена следующая формализованная модель для данной задачи:

$$Profit(X) = \sum_{j=1}^N k_j \cdot (1 + d_j \cdot t_j) \cdot x_j \rightarrow \max_X, \quad (4.16)$$

$$Risk(X) = \frac{1}{\sum_{j=1}^N x_j} \sum_{j=1}^N P_j \cdot x_j \leq \rho, \quad (4.17)$$

$$\sum_{j=1}^N k_j \cdot x_j \leq B, \quad (4.18)$$

$$X = (x_1, \dots, x_N), x_i \in \{0,1\}, i = \overline{1, N}, \quad (4.19)$$

где N – число кредитных заявок в размере k на срок t под d процентов, P_j – оценка рискованности одобрения заявки, выраженная в виде вероятности невозврата средств и процентов по кредиту, x_j – булева переменная, которая отражает одобрена заявка (значение 1) или нет (значение 0), B – сумма свободных пассивов банка, ρ – ограничение на рискованность портфеля, установленное законодательством и политикой банка.

Исходные данные для задачи (4.16) – (4.19) представлены Красноярским филиалом банка Москвы и взяты из [312]. Данные содержат информацию о 50 кредитных заявках, таким образом пространство поиска включает $2^{50} \approx 1.12E + 15$ различных кредитных портфеля. Исходные данные приведены в таблице Г.7 приложения Г, ограничение $\rho = 0.03$.

Задача управления инвестициями производственного предприятия ОПК имеет схожую постановку и заключается в формировании подмножества (инвестиционного портфеля) из множества инвестиционных проектов, максимизирующего прибыль от реализации выбранных проектов с учетом ограничений на инвестиционный бюджет и рискованность проектов, а для предприятий ОПК дополнительно и на отраслевую норму прибыли. В крупных предприятиях структурные подразделения вынесены в самостоятельные центры финансовой ответственности (ЦФО), поэтому инвестиционный бюджет складывается из средств конкретного ЦФО и средств предприятия в целом. В [313] предложена следующая формализованная модель для данной задачи:

$$Profit(X) = \sum_{i=1}^C \sum_{j=1}^{N_i} \Pi_{ij} \cdot x_{ij} \rightarrow \max_X, \quad (4.20)$$

$$Risk(X) = \frac{1}{\sum_{i=1}^C \sum_{j=1}^{N_i} x_{ij}} \sum_{i=1}^C \sum_{j=1}^{N_i} R_{ij} \cdot x_{ij} \leq \rho, \quad (4.21)$$

$$\sum_{i=1}^C \sum_{j=1}^{N_i} b_{ij} \cdot x_{ij} \leq B_C + B_E, \quad (4.22)$$

$$\frac{1}{\sum_{i=1}^C \sum_{j=1}^{N_i} x_{ij}} \sum_{i=1}^C \sum_{j=1}^{N_i} \Pi_{ij} \cdot x_{ij} \leq r, \quad (4.23)$$

$$X = (x_{11}, \dots, x_{CN_C}), x_{ij} \in \{0,1\}, i = \overline{1, C}, j = \overline{1, N_i}, \quad (4.24)$$

где C – количество ЦФО, представляющих $N_i, i = \overline{1, C}$ инвестиционных проектов, P_{ij} – оценка объема прибыли, R_{ij} – оценка рискованности, b_{ij} – плановые затраты на реализацию j -го проекта в i -м ЦФО, $B_C = \sum_{i=1}^C B_i$ – сумма инвестиционных бюджетов всех ЦФО предприятия, B_E – общий инвестиционный фонд предприятия, ρ – ограничение на рискованность портфеля, установленное законодательством и политикой предприятия, r – допустимая средняя норма прибыли по отрасли, x_{ij} – булева переменная, которая отражает включение заявки ЦФО в инвестиционный портфель ($x_{ij} = 1$, если заявка включена, $x_{ij} = 0$, иначе).

Исходные данные для задачи (4.20)-(4.24) представлены предприятием «Химзавод» – филиала АО «Красмаш» и взяты из [313]. Данные содержат информацию о 25 проектах от 5 ЦФО, пространство поиска содержит $2^{25} \approx 3.3E + 07$ различных кредитных портфелей. Данные приведены таблице Г.8 приложения Г. Значения параметров в ограничениях следующие: $B_E = 40, r = 0.5, \rho = 1.98$.

Настройки метода 3 совпадают с представленными в таблице 4.1, за исключением следующих:

- для задачи формирования кредитного портфеля:
 - максимальное число вычислений целевой функции – $5.0E+05$,
 - размер популяции – 500,
 - период адаптации – 20,
- для задачи формирования инвестиционного портфеля:
 - максимальное число вычислений целевой функции – $1.0E+05$,
 - размер популяции – 200,
 - период адаптации – 10.

Для учета ограничений при вычислении целевой функции в методе 3 используется метод статического штрафа [164].

В таблице 4.7 представлены 6 идентифицированных оптимумов с наибольшим значением по критерию прибыли, включая 1 глобальный оптимум (совпадающий с найденным ранее истинным глобальным оптимум) и 5 локальных

оптимумов, принадлежащие лучшему найденному решению методом 3 для задачи формирования оптимальной кредитной политики коммерческого банка, содержащее. Обобщенная статистика по 50 прогонам метода 3 представлена в таблице 4.8.

Таблица 4.7 – Идентифицированные оптимумы в задаче формирования оптимальной кредитной политики коммерческого банка

Структура портфеля	Доходность портфеля	Остаток свободных пассивов	Суммарная рискованность
01111011111111110111101000111110111010000101010111	199734518.9	30000	0.0292
11011100111110110110011101110101111010100101011110	199691164.0	15000	0.0286
0101111010111110001110111011111011001111011001110	199668728.9	15000	0.0280
00110011010001010110111000110110101111110101101111	199593407.3	10000	0.0276
00010001000011110111100001000101110101011001001111	199585077.0	20000	0,0278
00110011110110011010110010100011011110110000110010	199577920.7	5000	0,0269

Таблица 4.8 – Обобщенная статистика по 50 прогонам метода 3 в задаче формирования оптимальной кредитной политики коммерческого банка

Показатель	Метод 3	Лучшее по одной базовой эвристике
Наибольшее число идентифицированных оптимумов	23	18
Медиана числа идентифицированных оптимумов	16	15
Наименьшее число идентифицированных оптимумов	8	8
Средний максимум прибыли	1.997E+08	1.994E+08
Средний минимум прибыли	1.759E+08	1.150E+08
Средний максимум риска	2.82	2.81
Средний минимум риска	2.27	2.27
Средний максимум остатка свободных пассивов	45218.9	48122.6
Средний минимум остатка свободных пассивов	28054.4	26692.7

Как видно из таблицы 4.7, решения, полученные методом 3, имеют близкие значения по критериям прибыли и рискованности, но отличаются структурой портфеля кредитов. Данная информация может быть полезна для кредитных аналитиков. Например, если сделать выводы по шести портфелям из таблицы,

кредитные заявки 4, 19 и 49 содержатся во всех найденных оптимумах, заявки 13, 16, 21, 27, 30, 34, 35, 44 и 48 содержатся в 5 из 6 портфелях, а заявки 1, 17 и 43 включены в 1 из 6 портфелей. Если брать во внимание только первых 3 портфеля, содержащих наибольшие значения по критерию прибыли, то заявки 2, 4, 5, 9, 11-13, 15, 19, 23, 27, 28, 30, 34, 35, 44, 48 и 49 присутствуют во всех трех портфелях, а заявки 17, 36 и 45 не включены ни в один из трех портфелей. Набор альтернативных оптимальных решений вместо единственного лучшего найденного может быть полезен ЛПР, т.к. повышает устойчивость решения задачи, например, если по какой-то причине будут отозваны заявки, содержащиеся в глобально-оптимальном решении, или их одобрение. В целом, метод 3 позволяет идентифицировать до 23 различных оптимумов задачи, которые могут быть подвергнуты анализу ЛПР. При этом в каждом из запусков было найдено глобально-оптимальное решение.

Лучшее найденное решение методом 3 для задачи управления инвестициями производственного предприятия ОПК, содержащее 1 глобальный оптимум (совпадающий с найденным ранее истинным глобальным оптимум) и 3 локальных оптимума, представлено в таблице 4.9. Обобщенная статистика по 50 прогонам представлена в таблице 4.10.

Таблица 4.9 – Идентифицированные оптимумы в задаче управления инвестициями производственного предприятия

Структура портфеля	Доходность портфеля	Суммарная рискованность
10111111111111111110110001	117.7	1.98
10111101111111111110110001	111.00	1.96
00111111011111111110110001	104.80	1.91
10111111011111111010110001	104.60	1.91
00111110101111111111110001	101.10	1.89
10111111011111111110100001	98.30	1.85

Таблица 4.10 – Обобщенная статистика по 50 прогонам метода 3 в задаче управления инвестициями производственного предприятия

Показатель	Метод 3	Лучшее по одной базовой эвристике
Наибольшее число идентифицированных оптимумов	37	25
Медиана числа идентифицированных оптимумов	31	19
Наименьшее число идентифицированных оптимумов	18	9
Средний максимум прибыли	117.7	117.7
Средний минимум прибыли	97.42	73.95
Средний максимум риска	1.92	1.89
Средний минимум риска	1.69	1.50

Как видно из таблицы 4.9, решения, полученные методом 3, имеют близкие значения по критериям прибыли и рискованности. Структуры портфелей также почти полностью совпадают. В случае сравнения по всем 6 портфелям, решения различаются только по проектам 1, 7, 9, 17, 19 и 21, проекты 2 и 22-24 не включены не в один их 6 портфелей. Выводы при анализе трех лучших решений аналогичны, за исключением проектов 17 и 19, включенных во все три портфеля и проекта 19, не включённого ни в один из трех. Данная информация может быть полезной для инвестиционного аналитика, т.к. по проектам, совпадающим во всех оптимумах, можно сделать выводы о том, какие проекты являются принципиально важными для реализации, а какие должны быть исключены. В свою очередь, варьирующиеся в портфелях проекты дают возможность маневра при реализации.

4.5 Решение практической задачи аппроксимации множества экстремумов при проектировании систем на нечеткой логике

Современные подходы в области машинного обучения часто используют ЭА для решения задач проектирования и настройки методов и моделей. При этом ЭА, будучи универсальным оптимизационным подходом, может применяться к произвольным методам машинного обучения при решении любых прикладных

задач, и как результат, в литературе все чаще встречается термин «алгоритмы машинного обучения на основе ЭА» (*genetics-based machine learning algorithms*). Одним из крупных направлений в области машинного обучения являются методы на основе нечеткой логики [315, 316], преимуществом которых является возможность решать сложные задачи анализа данных, моделирования и управления в условиях неполной и нечеткой исходной информации.

Системы классификации на основе нечеткой логики (*fuzzy rule-based classification systems*, FRBCS) [317], помимо высокой точности решения задачи классификации, обеспечивают конечного пользователя удобной для интерпретации и анализа моделью принятия решений. Такая модель представляет собой набор правил (базу правил), представляющих собой продукции типа ЕСЛИ <условие>, ТО <вывод>. При этом переменные задачи, входящие в условие и вывод описываются лингвистическими терминами, схожими с понятиями естественной речи (например, ЕСЛИ Скорость=Большая, ТО Тормозное_усилие=Большое), а вывод из посылки осуществляется путем реализации механизмов нечеткого вывода, использующего операции с нечеткими множествами.

Сложность построения систем на нечеткой логике связана с формированием базы правил. Для хорошо изученных и простых задач, правила могут быть получены от эксперта предметной области, для сложных задач – правила можно сформировать в процессе обучения на прецедентах (задача обучения с учителем). Формализуем задачу проектирования FRBCS путем формирования базы правил как задачу оптимизации.

Пусть задана обучающая выборка задачи классификации:

$$S_m = \{(x_1, y_1), \dots, (x_m, y_m)\}, \quad (4.25)$$

где $x_i = (x_{i1}, \dots, x_{ij}, \dots, x_{in})$ – описание i -го объекта классификации, выраженное через значения характеризующих его признаков $j = \overline{1, n}$, y_i – метка (или номер) класса i -го объекта, m – размер выборки.

Система классификации на основе нечеткой логики задается базой правил R , представляющей собой нечеткое отношение (4.26):

$$R = A_{x_1} \times A_{x_2} \times \dots \times A_{x_n} \rightarrow A_y, \quad (4.26)$$

где A_{x_j} – лингвистическая переменная, описывающая j -й признак, A_y – лингвистическая переменная, описывающая переменную вывода продукции.

В свою очередь лингвистические переменные задаются следующим кортежем:

$$A_{x_j} = (x_j, T(x_j), U, G), \quad (4.27)$$

где x_j – имя четкой переменной, $T(x_j) = \{T_{j1}, \dots, T_{jN_j}\}$ – множество лингвистических термов переменной, являющихся нечёткими числами на множестве U ($x_j \in U$), G – дополнительные правила для определения синтаксиса и семантики термов лингвистической переменной.

Таким образом каждое правило имеет вид:

$$R_l: \text{ЕСЛИ } x_1 = T_{1k_{1l}} \text{ и } \dots \text{ и } x_n = T_{nk_{nl}}, \text{ то } y = T_{yk_{yl}}, \quad (4.28)$$

где $T_{jk_{jl}} \in T(x_j)$ – k_{jl} -й терм, которым описывается значение переменной x_j в l -м правиле.

Пусть база правил содержит N_R правил, и пусть задан целевой критерий качества решения задачи классификации L_{class} , тогда задача формирования базы правил может быть представлена как задача оптимизации (4.29):

$$L_{class} \left(T_{1k_{11}}, \dots, T_{nk_{n1}}, T_{yk_{y1}}, \dots, T_{1k_{1N_R}}, \dots, T_{nk_{nN_R}}, T_{yk_{yN_R}} \right) \rightarrow \min_T(\max), \quad (4.29)$$

В общем случае база правил может быть полной, т.е. содержать все элементы декартового произведения множеств термов переменных условия продукции. В таком случае в задачу (4.29) в качестве целевых переменных входят лишь термы вывода продукции. Однако на практике, такие базы содержат большое число правил и преимущество их интерпретируемости теряется. Например, для задачи классификации с 5 переменными, каждая из которых описана 3 термами, число

различных правил в полной базе правил равно 243, в случае использования 5 термов – 3125 правил. Поэтому на практике число правил стараются ограничить или задав число N_R , или путем введения дополнительного критерия, минимизирующего размер базы правил (многокритериальная постановка) [318-320].

Как описывалось ранее, традиционные ЭА в процессе решения задачи оптимизации сходятся к лучшему найденному решению. Для задачи (4.29) лучшее найденное решение будет иметь хорошее значение по критерию качества L_{class} (например, точность классификации), но нет гарантии, что полученная база правил будет удобной для понимания и анализа ЛПП. В тоже время, решая задачу аппроксимации множества экстремумов, можно получить несколько оптимальных решений (глобальных или допустимых по значению локальных), которые будут представлять различные базы правил, отражающие разные способы принятия решения. Такие решения могут быть полезны как с точки зрения решения конкретной задачи классификации, так и для анализа проблемосодержащей системы в целом.

Для апробации метода 3 были решены прикладные задачи классификации на основе наборов данных репозитория UCI Калифорнийского университета в Ирвайне [321]. Краткая информация о задачах и характеристики набора данных приведены в таблице 4.11.

Таблица 4.11 – Описание наборов данных задач классификации

Название набора данных	Объем выборки	Число атрибутов	Число классов	Краткое описание
Banknote Authentication	1372	5	2	Данные содержат результаты вейвлет преобразований изображений настоящих и поддельных банкнот.
Breast Cancer Wisconsin	699	10	2	Данные содержат оцифрованную информацию о результатах анализа изображений опухоли при диагностике рака груди.

Продолжение таблицы 4.11

Название набора данных	Объем выборки	Число атрибутов	Число классов	Краткое описание
Heart Disease	303	75	2	Данные содержат информацию анамнеза и диагностики пациентов с различными заболеваниями сердца.
Ionosphere	351	34	2	Данные содержат информацию о концентрации свободных электронов в ионосфере, полученную в ходе анализа сигналов радиолокационной станции.
Iris	150	4	3	Данные содержат информацию о геометрии цветка и стебля различных сортов ирисов.
Liver Disorders	345	7	2	Данные содержат информацию об анализах крови пациентов с заболеваниями печени.
Seeds	210	7	3	Данные содержат информацию о геометрии зерен нескольких сортов пшеницы.
Australian Credit	690	14	2	Задача кредитного скоринга банков Австралии.
German Credit	1000	20	2	Задача кредитного скоринга банков Германии.
Vertebral Column	310	6	2	Данные содержат информацию о форме и ориентации таза и поясничного отдела пациентов с проблемами позвоночника.
Wine Quality	4898	12	10	Данные содержат информацию о качестве португальских вин, полученную по данным химического анализа.

Для решения задач с помощью системы классификации на основе нечеткой логики в данной работе используется метод, предложенный профессором Ишибучи, использующий весовые коэффициенты для правил в базе (фактор определенности, *certainty factor* (CF)), на основе которых вычисляются назначаемые метки классов для правил и осуществляется итоговый вывод решения по базе правил [322].

Семантика лингвистических переменных для формирования правил использует метод грануляции нечетких множеств, при котором термы размещаются равномерно на интервале значений исходных четких переменных. В данном подходе исходные данные нормируются (например, в интервал $[-1, +1]$), задаются универсальные имена термов: «малое значение», «нормальное значение»,

«большое значение» и т.д. Традиционно используется разбиение на 3, 5 и 7 термов, выделяя центральный терм имеющий смысл «нормальное значение», «нулевое значение» и т.д.

В данном исследовании при построении системы классификации на основе нечеткой логики используются 5 термов с треугольной функцией принадлежности: «очень малое значение», «малое значение», «нормальное значение», «большое значение» и «очень большое значение». Базы правил, использующие 3 терма, обычно дают решение с низкой точностью, но обладают большой обобщающей способностью и удобны для интерпретации. Базы правил, использующие 7 термов, наоборот, позволяет существенно повысить точность классификации, но сложны в интерпретации, а их формирование путем решения оптимизационной задачи является более сложным, т.к. растет пространство поиска. Разбиение с использованием 5 термов обеспечивает компромисс между точностью и интерпретируемостью. Пример разбиения для 5 термов показан на рисунке 4.17.

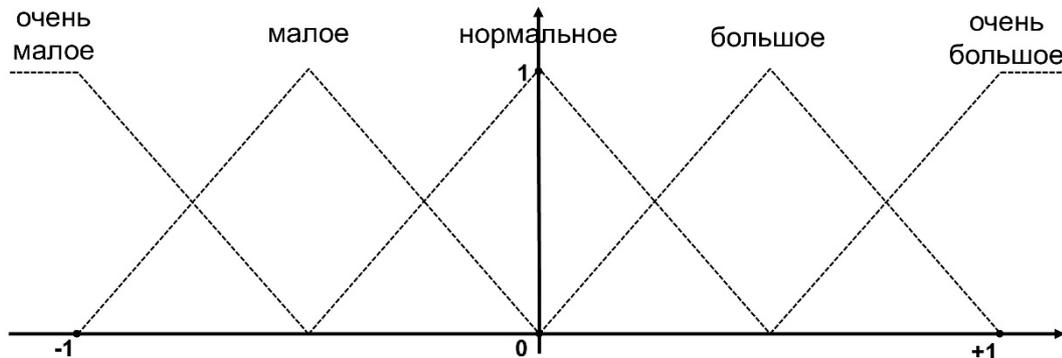


Рисунок 4.17 – Пример разбиения на 5 нечетких множеств

При форматировании базы правил к основным 5 термам был добавлен терм игнорирования (*don't care*, DC), который имеет значение функции принадлежности равное 1 на всей области значений переменной. Использование терма DC в правиле означает, что данная переменная игнорируется в правиле (правило будет применяться при любом значении переменной). Применение терма игнорирования повышает обобщающую способность классификатора и сокращает размеры правил

в базе. Если правило в базе содержит терм DC во всех позициях, то правило исключается из базы. Таким образом удается построить базы, содержащие меньшее число правил и правила меньшей длины.

Максимальное число правил в базе ограничено числом 12. Расстояние между двумя решениями в эвристиках метода 3 вычисляется как число различающихся термов в соответствующих позициях правил по всем правилам базы.

Функция пригодности в методе 3 базируется на значении точности классификации *Accuracy* (отношение числа верно классифицированных примеров к общему числу примеров) и содержит штраф на сложность базы *Complexity* (отношение числа отличных от DC термов к общему числу термов в базе правил). Функция пригодности имеет вид свертки (4.30):

$$fitness(R) = 0.9 \cdot Accuracy(R) - 0.1 \cdot Complexity(R). \quad (4.30)$$

Настройки метода 3 совпадают с представленными в таблице 4.1, за исключением следующих:

- максимальное число вычислений целевой функции – $8.0E+04$,
- размер популяции – 200,
- период адаптации – 20,
- число независимых запусков - 40.

Результаты численных экспериментов при решении задач классификации показаны в таблице 4.12. Поскольку в каждом независимом запуске может получиться уникальное множество идентифицированных оптимумов (различные базы правил), то были оценены следующие параметры: медиана числа идентифицированных оптимумов методом 3, максимум и медиана точности классификации, усредненные по всем базам правил идентифицированных оптимумов и точность лучшего найденного решения по всем прогонам. В таблице также содержится значение лучшего найденного решения, полученного обычным ЭА без мультимодальных эвристик.

Таблица 4.12 – Результаты решения задач классификации

Название набора данных	Метод 3				Обычный ЭА
	Медиана числа идентифицированных оптимумов	Максимум средней точности классификации	Медиана средней точности классификации	Лучшая найденная точность	Лучшая найденная точность
Banknote Authentication	19	0.941	0.923	0.950	0.945
Breast Cancer Wisconsin	21	0.935	0.911	0.962	0.939
Heart Disease	14	0.832	0.746	0.887	0.753
Ionosphere	7	0.712	0.689	0.803	0.722
Iris	5	0.905	0.867	0.962	0.935
Liver Disorders	5	0.592	0.506	0.607	0.581
Seeds	7	0.713	0.622	0.847	0.856
Australian Credit	22	0.754	0.597	0.881	0.860
German Credit	17	0.722	0.637	0.771	0.792
Vertebral Column	8	0.702	0.573	0.742	0.779
Wine Quality	15	0.856	0.733	0.910	0.880

На рисунках 4.18-4.20 представлены три лучших базы правил одного из прогонов метода 3 на задаче идентификации банкнот. Рисунки дают наглядное представление о том, как выглядят правила в базе: строки соответствуют правилам, столбцы – переменным, а выделенный темным терм обозначает терм, используемый в правиле в данной позиции, пустые правила (терм DC по всех позициях) исключены из базы, последний столбец содержит метку класса. Как видно на данных рисунках, базы имеют различную структуру, но есть возможность проанализировать, какие термы чаще встречаются в соответствующих переменных, а какие переменные оказываются неинформативными (часто содержат терм DC). Например, в приведенных рисунках наиболее редко терм игнорирования встречается с третьей переменной, а наиболее часто – с первой, четвертой и пятой. Данная информация может оказаться полезной для эксперта в предметной области.

DC		DC		DC	1
		DC	DC		1
DC	DC		DC	DC	2
DC		DC	DC		2
DC			DC	DC	1
	DC		DC	DC	1
	DC			DC	2
DC			DC	DC	2
DC					2
				DC	1
					2

Рисунок 4.18 – База правил, являющаяся локальным оптимум задачи (4.29),

точность классификации – 0.938

DC	DC	DC		DC	1
			DC		1
DC	DC		DC	DC	2
DC					2
DC		DC			2
DC	DC			DC	2
				DC	1
DC	DC		DC		2

Рисунок 4.19 – База правил, являющаяся локальным оптимум задачи (4.29),

точность классификации – 0.932

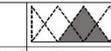
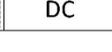
DC	DC		DC		2
				DC	2
DC			DC	DC	1
			DC	DC	1
					2
	DC		DC	DC	1
DC			DC	DC	1
DC			DC		2
DC	DC				2
				DC	1

Рисунок 4.20 – База правил, являющаяся локальным оптимум задачи (4.29),
точность классификации – 0.923

Как видно из результатов численных экспериментов, применение метода 3 в задаче проектирования систем на нечеткой логике позволяет получить множество оптимальных решений, представляющих собой базы правил с различной структурой базы и правил. Максимум точности классификации, усредненной по всем базам правил идентифицированных оптимумов, оказывается сравнимой или превосходит точность, полученную традиционным ЭА, не использующим мультимодальные эвристики, таким образом, можно сделать вывод, что метод 3 локализует глобальный оптимум и ближайшие по значению локальные оптимумы. При этом на 8 из 11 задачах классификации лучше найденное решение превосходит по точности классификации лучшее найденное традиционным ЭА, не использующим мультимодальные эвристики. Последнее объясняется тем, что традиционные ЭА, оказавшись в зоне притяжения некоторого оптимума могут потерять разнообразие и сойтись в точку данного оптимума, в то время как метод 3 будет контролировать скопление индивидов и поддерживать разнообразие в популяции, повышая тем самым шанс обнаружить зону притяжения другого оптимума, который может иметь лучшее значение.

Практическая польза от применения метода 3 при решении задач проектирования систем на нечеткой логике связана с предоставлением системному аналитику и ЛПР дополнительной информации о способах решения поставленных

задач классификации. Альтернативные решения могут быть подвергнуты дополнительному анализу, для лучшего понимания проблемной ситуации, или использованы для повышения надежности решения.

Выводы по главе 4

В Главе 4 рассмотрена проблема аппроксимации множества экстремумов в задачах оптимизации сложных систем. Анализ информационных источников показал, что лучшие результаты на сегодня достигнуты в классе стохастических популяционных алгоритмов, таких как ЭА. Традиционные эвристики мультимодальной оптимизации, построенные на биологических моделях и свойствах ландшафта целевых функций вещественного пространства поиска, имеют большой разброс оценок их эффективности на задачах разного типа, а также плохо переносятся на задачи со смешанными переменными. Предложенная в Главе 4 селективная гиперэвристика синтеза ЭА идентификации множества экстремумов (метод 3) оперирует только базовыми эвристиками мультимодальной оптимизации, которые не используют информацию о пространстве поиска в явном виде, а анализируют поведение популяции. В процессе решения задачи метод 3 анализирует число локализованных оптимумов и разнообразие популяции каждой из применяемых эвристик и осуществляет выбор и формирование новой комбинации эвристик путем перераспределения размеров популяций. В отличие от метода 1, который реализуется офлайн для обобщения результатов решения задач оптимизации, и метода 2, который реализуется онлайн для отслеживания изменений в среде, метод 3 может быть реализован обоими способами, в зависимости от потребностей исследователя.

Анализ эффективности метода 3 на эталонных тестовых задачах показал, что метод 3 всегда превосходит эффективность случайного выбора одной из базовых эвристик, и в большинстве случаев превосходит эффективность лучшей из базовых эвристик. Эталонные тестовые задачи с бинарными переменными не представляют

трудности для метода 3, алгоритм находит все известные глобальные оптимумы в каждом запуске, хотя эффективность базовых эвристик, применяемых по отдельности, может быть низкой. По результатам решения эталонных тестовых задач IEEE CEC, метод 3 превосходит по эффективности ряд ведущих алгоритмов мультимодальной оптимизации, а также превосходит оценку случайного выбора одного из ведущих алгоритмов на 4-х уровнях точности из 5-ти.

Для апробации метода 3 были решены практические задачи аппроксимации множества экстремумов при поддержке принятия решений при формировании оптимальной кредитной политики коммерческого банка и при управлении инвестициями производственного предприятия ОПК, а также задача аппроксимации множества экстремумов при проектировании систем на нечеткой логике. Применение метода 3 при решении практических задач обеспечивает системного аналитика и ЛПР дополнительной информацией об альтернативных решениях. Данная информация может быть использована для дальнейшего анализа проблемосодержащей системы, первоначально представленной моделью «черного ящика», и способов решения проблемы. Наличие множества альтернативных решений также повышает устойчивость реализации решения, в случае если одна из альтернатив не будет реализована. В общем случае, среди идентифицируемых оптимумов содержится и глобально-оптимальное решение. Более того, поддержка мультимодальности, повышает шанс найти глобальный оптимум в сложных задачах оптимизации в мультимодальной среде, когда традиционные ЭА сходятся в одном из локальных оптимумов.

Рассмотренные в предыдущих главах классы задач оптимизации и соответствующие подходы обычно имеют дело с сравнительно небольшими размерностями, для которых ЭА сохраняют эффективность и имеется достаточно вычислительных ресурсов. В то же время, для многих современных задач требуется реагировать на постоянно возрастающее число внешних и внутренних факторов и связей, что приводит к увеличению числа переменных задач оптимизации. Эффективность традиционных ЭА при решении задач глобальной оптимизации с алгоритмически заданными целевыми функциями падает с увеличением

размерности, следовательно, требуется разработка эффективных подходов для решения задач большой размерности. Сегодня предложены различные эвристики для работы с подобными задачами, однако выбор и корректное использование одной из них является самостоятельной задачей управления ЭА, для решения которой может быть применен подход на основе гиперэвристик.

ГЛАВА 5. СЕЛЕКТИВНАЯ ОНЛАЙН ГИПЕРЭВРИСТИКА СИНТЕЗА ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ДЛЯ ЗАДАЧ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ БОЛЬШОЙ РАЗМЕРНОСТИ

Глава 5 посвящена проблеме разработки и исследования гиперэвристик для решения задач глобальной оптимизации большой размерности с алгоритмически заданными целевыми функциями. Как обсуждалось ранее в Главе 1, одним из основных свойств сложной системы является сложность, обусловленная числом факторов и связей. Эффективность многих метаэвристик глобальной оптимизации, включая ЭА, падает с ростом размерности, поэтому для решения задач большой размерности требуется разработка специализированных алгоритмов оптимизации. Сегодня наилучшие результаты для данного класса задач получены с использованием идей декомпозиции задачи оптимизации и коэволюционного подхода. При этом существуют различные эвристики для разбиения задачи, демонстрирующие разную эффективность на задачах с разными свойствами. Предложенная в Главе 5 селективная онлайн гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности реализует островную модель ЭА, в которой каждый остров независимо реализует свою собственную декомпозицию задачи в соответствии с назначенной базовой эвристикой. Онлайн выбор и комбинирование эвристик осуществляется путем перераспределения размеров островов и кооперации независимых алгоритмов. Численные эксперименты на множестве эталонных тестовых задач, предложенных на специальной сессии и соревновании по глобальной оптимизации большой размерности IEEE CEC, показывают, что предложенная гиперэвристика превосходит оценку случайного выбора одной из базовых эвристик и оценку случайного выбора одного из ведущих алгоритмов. Для апробации предложенного подхода решена практическая задача повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах. Результаты решения задачи показали, что предложенная гиперэвристика дает более стабильный результат по сравнению с ранее используемыми подходами (лучшее значение медианы целевой функции по

серии независимых запусков), но уступает по лучшему найденному среди всех прогонов специализированному алгоритму GA-MPC, предложенному на конкурсе по глобальной оптимизации большой размерности.

5.1 Проблема большой размерности в задачах глобальной оптимизации с алгоритмически заданными целевыми функциями и обзор подходов к ее решению

Как показывает практика ПСА, современные требования к эффективности решения задач анализа (точности, надежности, устойчивости, повторяемости, интерпретируемости и т.д.) весьма высоки и требования постоянно ужесточаются, что связано как с практической необходимостью, так и с появлением более эффективных вычислительных методов исследования, моделирования и оптимизации сложных систем. Благодаря росту производительности вычислительной техники появилась возможность собирать и обрабатывать информацию о множестве факторов, которые включаются в модель сложной системы. Среди задач, которые обычно имеют большую размерность, можно выделить задачи планирования и составления расписаний, задачи управления распределенными «умными» сетями в энергетике, задачи управления трафиком и планирования маршрутов, задачи распознавания генов в биоинформатике и другие [323].

Как результат, размерность задач анализа и сопутствующих им задач оптимизации выросла и продолжает расти. Увеличение количества целевых переменных экспоненциально увеличивает пространство поиска («проклятие размерности», «комбинаторный взрыв») [21]. Класс задач глобальной оптимизации с алгоритмически заданной целевой функцией, в которых большая размерность является препятствием, затрудняющим эффективное решение задачи, называется *Large Scale Global Optimization* (LSGO) [25]. Стоит отметить, что в класс LSGO не попадают задачи, для которых большая размерность не представляет

дополнительной трудности, например, некоторые типы сепарабельных задач, линейное и выпуклое программирование и другие задачи с хорошо известными свойствами.

Одни из наиболее известных специалистов в области LSGO, Мохаммед Наби Омидвар и Сяодун Ли, отмечают, что термин «большая размерность» не является определённым – размерности задач, которые относят в LSGO к большим, меняются (растут) со временем (рисунок 5.1), а также могут быть разными в разных прикладных областях. Современный уровень развития метаэвристик, включая ЭА, рассматривает задачи LSGO от 1000 вещественных переменных [25] и порядка одного миллиона бинарных [324].

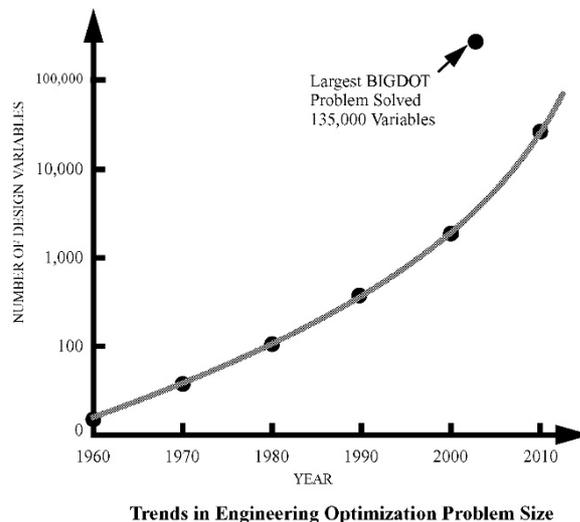


Рисунок 5.1 – Тренд роста размерностей задач оптимизации в инженерных приложениях, представленных в отчете NASA в 2002 году [325]

Как отмечалось ранее в Главе 1, несмотря на то, что процедура СА подразумевает выполнение декомпозиции большой системы, в ходе декомпозиции могут теряться синтетические (эмерджентные) свойства системы. Поэтому многие задачи класса LSGO являются несепарабельными, что затрудняет их решение.

Введем ряд обозначений, которые используются в области LSGO.

В задаче (1.1) переменная x_i называется *сепарабельной* и *невзаимодействующей* с другими переменными, если:

$$\arg \min_{\bar{x} \in S} (\max) f(\bar{x}) = \left(\arg \min_{x_i \in S} (\max) f(\bar{x}), \arg \min_{x_j \in S, j \neq i} (\max) f(\bar{x}) \right), \quad (5.1)$$

где $\bar{x} = (x_1, \dots, x_n)$ – вектор целевых переменных, $x_i, i \in [1, n]$ – i -я компонента вектора (i -я переменная).

Наличие взаимодействующих переменных может существенно усложнять решение задачи оптимизации, в следствии изменения ландшафта целевой функции: поворота осей, появления овражности, потери линейности и других изменений. На рисунке 5.2 показан пример взаимодействующих переменных квадратичной функции.

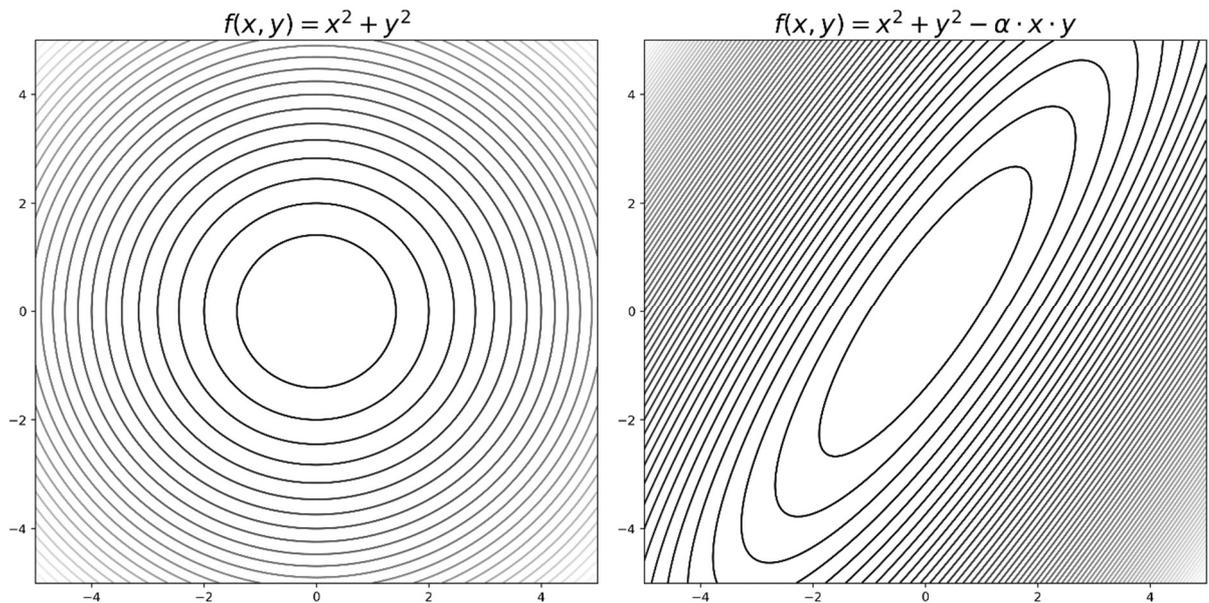


Рисунок 5.2 – Пример взаимодействующих переменных

С понятием взаимодействующих переменных связаны понятия сепарабельной, частично сепарабельной и полностью несепарабельной функций [25, 326, 327].

Функция $f(\bar{x})$ называется *частично сепарабельной* с t независимыми компонентами, если:

$$\arg \min_{\bar{x} \in S} (\max) f(\bar{x}) = \left(\arg \min_{\bar{x}_1 \in S} (\max) f(\bar{x}_1), \dots, \arg \min_{\bar{x}_m \in S} (\max) f(\bar{x}_m) \right), \quad (5.2)$$

$$\bar{x} = \bigcup_{j=2}^m \bar{x}_j, \bigcap_{j=2}^m \bar{x}_j = \emptyset, \quad (5.3)$$

где \bar{x}_j – непересекающиеся компоненты вектора переменных \bar{x} , $2 \leq m < n$.

Функция $f(\bar{x})$ называется *сепарабельной (полностью сепарабельной)*, если все компоненты в (5.2)-(5.3) содержат только одну переменную, т.е. $m = n$.

Функция $f(\bar{x})$ называется *частично аддитивно сепарабельной*, если:

$$f(\bar{x}) = \sum_{j=1}^m f_j(\bar{x}_j). \quad (5.4)$$

Функция $f(\bar{x})$ называется *полностью несепарабельной*, если любая пара переменных является взаимосвязанной.

Глобальный экстремум сепарабельных функций достигается в экстремумах ее компонентов, следовательно, задачи большой размерности могут быть сведены к совокупности независимых задач меньшей размерности, а в предельном случае – к совокупности одномерных задач оптимизации. К сожалению, практические задачи класса LSGO редко бывают полностью сепарабельными, а поскольку они представлены моделями «черного ящика», анализ свойств сепарабельности затруднен.

Несмотря на то, что ЭА являются универсальными подходами для решения различных задач оптимизации и демонстрируют высокую эффективность при решении сложных задач глобальной оптимизации с алгоритмически заданными целевыми функциями, достаточно эффективных подходов для класса LSGO пока не предложено. Исследования в области ЭА, связанные с решением задач большой размерности, можно разделить на две большие группы:

– подходы, основанные на модификации и повышении эффективности использования традиционных эвристик ЭА;

– подходы, основанные на декомпозиции задачи оптимизации.

Подходы первой группы включают модификации способов инициализации популяции [328, 329], модификацию операторов скрещивания и мутации [330],

использование аппроксимаций целевой функции [331], использование меметических алгоритмов [332, 333] и гибридизацию подходов [334, 335]. Несмотря на обилие работ, подходы первой группы обычно применимы лишь для некоторых типов ЭА (наиболее часто используется алгоритм дифференциальной эволюции) и часто проигрывают по эффективности подходам второй группы.

Подходы, основанные на декомпозиции задачи оптимизации, являются более универсальными, т.к. не подразумевают использование конкретной метаэвристики и пространства поиска (в отличие от подходов первой группы, ориентированных на задачи с вещественными переменными и использующих локальный поиск и поиск по направлениям). В настоящее время, большинство работ в области LSGO посвящены этой группе подходов, а некоторые из подходов на основе декомпозиции являются референтными (например, DECC-G [336]) при оценке эффективности новых алгоритмов и проведении соревнований по глобальной оптимизации с алгоритмически заданными целевыми функциями. Методы второй группы в основном различаются эвристикой, используемой для декомпозиции задачи оптимизации. В свою очередь, проблема выбора и комбинации эвристик декомпозиции может быть решена с помощью гиперэвристического подхода.

Метод декомпозиции задач большой размерности заключается разбиении исходной задачи на несколько задач меньшей размерности путем группировки переменных в непересекающиеся субкомпоненты, которые могут быть равного или произвольного размера. Оптимизация по каждой из субкомпонент производится независимо, но оценка качества решения должна производиться по полному вектору переменных исходной задачи (рисунки 5.3-5.4). Для этого в методе декомпозиции используется кооперативная коэволюция, предложенная Де Йонгом (алгоритм 9) [337].

Алгоритм 9. Алгоритм кооперативной коэволюции с декомпозицией задачи.

Вход: число субкомпонент, ЭА.

Инициализировать популяцию, сохранить лучшее найденное решение.

Шаг декомпозиции задачи. Сгруппировать переменные задачи в непересекающиеся субкомпоненты.

Повторять пока не выполнится условие останова:

Выбрать очередную субкомпоненту, оптимизировать ее заданное число поколений с помощью ЭА.

Шаг кооперативной коэволюции. При оценке пригодности текущей субкомпоненты, полный вектор решения формируется из оцениваемой субкомпоненты и лучших найденных значений по другим субкомпонентам (в первом цикле – из переменных лучшего найденного при инициализации).

Выход: итоговое решение, полученное объединением лучших найденных значений по субкомпонентам.

Выбор субкомпонент обычно происходит циклически, начиная с первой (алгоритм *round-robin*), а максимальное число вычислений целевой функции выбирается, так, чтобы алгоритм совершил несколько циклов оценки субкомпонент.

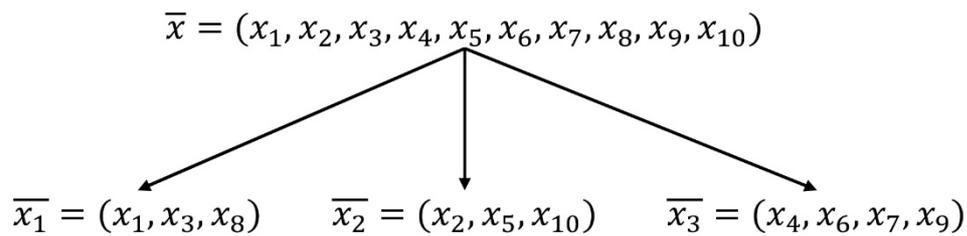


Рисунок 5.3 – Пример группировки переменных в 3 субкомпоненты

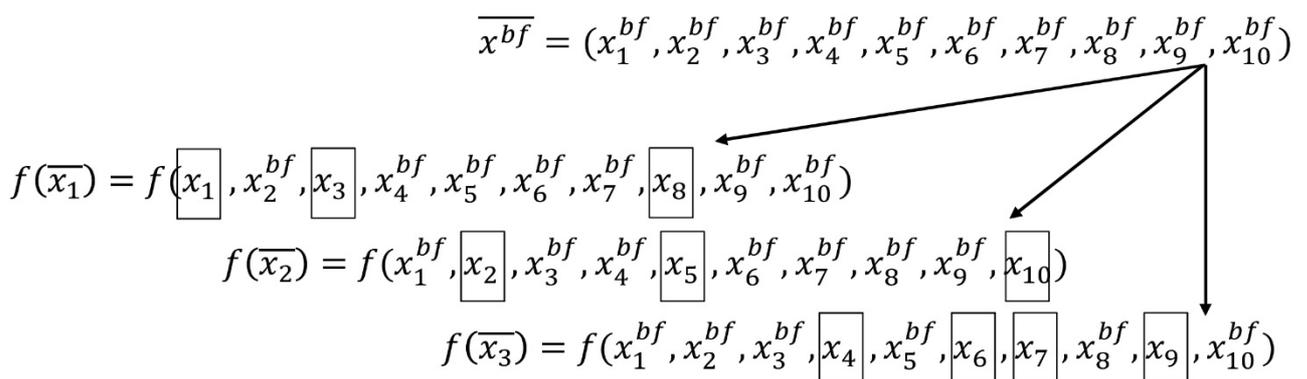


Рисунок 5.4 – Пример формирования полного вектора решения при оценке пригодности субкомпонент (переменные субкомпоненты обведены линией, *bf* – лучшее найденное решение)

Шаг декомпозиции задачи является наиболее важным. Существуют различные подходы, которые можно объединить в две группы, использующие статическую и динамическую группировку переменных.

В методе статической группировки правило формирования субкомпонент задается до начала решения задачи и не меняется в процессе решения. Одни из первых подходов, предложенные Поттером и Де Йонгом для ГА: CCGA-1 и CCGA-2 использовали субкомпоненты с одной переменной [337] и стратегию разделения задачи с n переменными на 2 субкомпоненты равного размера по $n/2$ переменных [338]. В последствии были предложены модификации для различных типов ЭА: FEPCC [339], CPSO [340], DE [341] и др. Предложенные подходы превосходят обычные ЭА на сепарабельных задачах большой размерности, но не имеют преимуществ на частично и полностью несепарабельных. В обзорной работе [323] отмечается, что большинство подходов, использующих статическую группировку, эффективны для размерностей до 100 переменных и уступают методам динамической группировки.

Подходы на основе динамической группировки позволяют менять структуру субкомпонент в процессе решения задач оптимизации, основываясь на оценке эффективности группировки на прошлых итерациях. При этом различают подходы, использующие случайную группировку (*random dynamic grouping*), и подходы, основанные на анализе взаимодействия переменных для формирования компонент (*learning-based dynamic grouping*).

При разработке гиперэвристики для задач глобальной оптимизации большой размерности, подходы данной группы являются наиболее перспективными. Одним из наиболее известных и эффективных подходов случайной динамической группировки является метод случайной группировки с адаптивными весами (*framework with random grouping and adaptive weighting*, DECC-G) [336], на основе которого создано множество модификаций и специализированных алгоритмов для решения прикладных задач [323]. Среди известных подходов на основе анализа взаимодействия переменных стоит отметить метод дифференциальной группировки (*differential grouping*, DECC-DG) [342] и метод дельта-группировки

(*delta grouping*, DECC-D) [343]. Данные подходы позволяют автоматически находить взаимодействующие переменные и объединять их в группы. Однако на полностью несепарабельных задачах методы анализа взаимодействия переменных могут уступать методам со случайной группировкой.

Рассмотрим основные идеи эвристик выбранных подходов: случайная группировка, дифференциальная группировка и дельта-группировка.

Идея метода случайной группировки с адаптивными весами заключается в разбиении n -мерной задачи оптимизации на m субкомпонент по s переменных ($n = m * s$). Каждая компонента оптимизируется отдельно, используя принцип кооперативной коэволюции (Алгоритм 9) заданное число итераций, называемых циклом. В каждом цикле группировка происходит случайно, т.е. каждая переменная вектора решения имеет одинаковый шанс попасть в любой из субкомпонентов. По окончании каждого цикла, выполняется еще один шаг оптимизации – оптимизация лучшего найденного, случайного и худшего найденного решения путем оптимизации весовых коэффициентов субкомпонентов [336].

Дополнительный шаг оптимизации с помощью весовых коэффициентов реализует следующую идею. Для любого решения $\bar{x} = (x_1, x_2, \dots, x_n)$ справедливо (5.5)-(5.7).

$$f(\bar{x}) \equiv f(w_c \cdot \bar{x}), \quad (5.5)$$

где $w_c = (1, 1, \dots, 1)$ – вектор констант длины n .

$$\min_w f(w \cdot \bar{x}) \leq f(w_c \cdot \bar{x}), \quad (5.6)$$

где $w = (w_1, w_2, \dots, w_n)$ – вектор весовых коэффициентов длины n .

$$\min_w f(w \cdot \bar{x}) \leq \min_w f(\tilde{w} \cdot \bar{x}) \leq f(w_c \cdot \bar{x}), \quad (5.7)$$

где $w = (\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_m)$ – вектор весовых коэффициентов длины m , а каждый из коэффициентов \tilde{w}_j применяется ко всем s переменным j -й субкомпоненты ($j = \overline{1, m}$).

Таким образом адаптация вектора позволяет выполнить дополнительный шаг оптимизации в пространстве размерности $m \ll n$.

Метод дифференциальной группировки основан на определении частично аддитивно сепарабельной функции (5.4). В работе [342] авторы доказали следующую теорему.

Пусть $f(\bar{x})$ частично аддитивно сепарабельная функция. Для $\forall a, b_1 \neq b_2, \delta \neq 0 \in R$, если выполняется условие

$$\Delta_{\delta, x_p} f(\bar{x})|_{x_p=a, x_q=b_1} \neq \Delta_{\delta, x_p} f(\bar{x})|_{x_p=a, x_q=b_2} \quad (5.8)$$

$$\Delta_{\delta, x_p} f(\bar{x}) = f(\dots, x_p + \delta, \dots) - f(\dots, x_p, \dots) \quad (5.9)$$

то переменные x_p и x_q являются несепарабельными.

На основе свойства (5.8) предложен следующий алгоритм группировки переменных в субкомпоненты (Алгоритм 10).

Алгоритм 10. Алгоритм дифференциальной группировки.

Вход: целевая функция $f(\bar{x})$, размерность задачи n .

Задать множество номеров переменных $dims = \{1, 2, \dots, n\}$.

Задать множество номеров сепарабельных переменных $seps = \emptyset$.

Задать множество идентифицированных несепарабельных групп переменных $allgroups = \emptyset$.

Повторять для $i \in dims$:

$group = \{i\}$

Задать произвольные значения $lb \neq ub, lb < 0 < ub$.

Повторять для $j \in dims \wedge i \neq j$:

Задать значения всех переменных векторов решения \bar{p}_1 и \bar{p}_2 значениями равными lb .

$\bar{p}_2(i) = ub$

$\Delta_1 = f(\bar{p}_1) - f(\bar{p}_2)$

$\bar{p}_1(j) = 0, \bar{p}_2(j) = 0$

$\Delta_2 = f(\bar{p}_1) - f(\bar{p}_2)$

Если $|\Delta_1 - \Delta_2| > \epsilon$, то

$group = group \cup \{j\}$

$dims = dims \setminus group$

Если $|group| = 1$, то

$seps = seps \cup group$

иначе

$allgroups = allgroups \cup \{group\}$

$allgroups = allgroups \cup \{seps\}$

Выход: множество субкомпонент.

Параметр ϵ в Алгоритме 10 определяет чувствительность нахождения взаимодействующих переменных. В работе [342] было показано, что для сепарабельных и частично сепарабельных задач, подход способен сформировать истинные группы несепарабельных и сепарабельных компонент.

Метод дельта-группировки основан на следующей эвристике. В случае сепарабельной целевой функции возможна оптимизация по отдельным переменным. В случае несепарабельной функции, и особенно если взаимодействующие переменные были сгруппированы в разные субкомпоненты, улучшения значений целевой функции при изменении по отдельным координатам или некорректно сгруппированным субкомпонентам будут небольшими [343]. В методе дельта-группировки вычисляются средние по всей популяции координатные изменения для каждой из переменных, произошедшие за одну итерацию ЭА – дельта-вектор (5.10).

$$\Delta = \{\overline{\delta}_1, \overline{\delta}_2, \dots, \overline{\delta}_n\}, \quad (5.10)$$

$$\overline{\delta}_i = \frac{\sum_{j=1}^{popSize} \delta_{i,j}}{popSize}, i = \overline{1, n}, \quad (5.11)$$

где n – размерность задачи, $popSize$ – размер популяции, $\delta_{i,j}$ – изменение i -й координаты j -го индивида.

Компоненты вектора Δ сортируются в порядке убывания значений $\overline{\delta}_i$ и группируются в заданное число субкомпонент равного размера. Алгоритм дельта-группировки имеет следующий вид:

Алгоритм 11. Алгоритм дельта-группировки.

Вход: Задача оптимизации, ЭА, число субкомпонент m .

Инициализация. Заполнить вектор Δ нулевыми значениями, сформировать субкомпоненты в естественном порядке переменных.

Повторять пока не выполнится условие останова:

Повторять для всех субкомпонент $i = \overline{1, m}$:

Выполнить одну итерацию оптимизации i -й субкомпоненты выбранным ЭА.

Вычислить значения вектора Δ по формуле (5.10).

Отсортировать значения вектора Δ по убыванию, сформировать соответствующие группы переменных.

Выход: лучшее найденное решение.

Как отмечалось ранее в Главе 1, в задачах оптимизации сложных систем независимые переменные могут быть выражены в различных шкалах (задачи со смешанными переменными). Для решения таких задач обычно используется бинарный ГА. Бинарный ГА может быть полезен и для задач с вещественными переменными, т.к. бинаризация формирует иную топологию ландшафта целевой функции, и, как отмечалось ранее, в бинаризованном пространстве поиска число локальных оптимумов обычно меньше. Для работы с бинарными переменными в данной работе предложена новая эвристика, основанная на анализе распределения вероятностей появления значений равных единице.

Популяционные стохастические алгоритмы оптимизации, использующие оценки распределения вероятностей для формирования решений относят к классу EDA (*estimation of distribution*) [67]. Для бинарных ГА распределение генов в популяции можно представить в виде следующего распределения вероятностей появления генов равных единице (5.12):

$$P(t) = (p_1(t), \dots, p_n(t)), p_i(t) = P(x_i = 1) = \frac{1}{popSize} \sum_{j=1}^{popSize} x_{ij}, \quad (5.12)$$

где x_i – i -я позиция в хромосоме, $P(x_i = 1)$ – вероятность появления единицы в i -й позиции, x_{ij} – значение i -го гена j -го индивида на t -й итерации.

Ранее автором исследования был предложен новый ГА – вероятностный ГА (ВГА), использующий оценку распределения (5.12) для сохранения опыта исследования прошлых решений и формирования новых решений [344]. В ВГА отсутствуют традиционные операторы рекомбинации и мутации, решения формируются на основе оценки распределения. Было показано, что ВГА превосходит по оценкам вероятности нахождения глобального оптимума и среднего числа итераций обычный ГА. В работах [345, 346] были предложены модификации ВГА для повышения его эффективности и решения задач оптимизации различных классов.

В работах по исследованию эффективности ВГА было выявлено свойство, связанное с динамикой изменения компонент вектора вероятностей (5.12) в случаях, когда алгоритм сходится к одному из оптимумов:

$$\begin{cases} p_i(t) \rightarrow 0, \text{ если } x_i^* = 0 \\ p_i(t) \rightarrow 1, \text{ если } x_i^* = 1 \end{cases} \quad (5.13)$$

где x_i^* – значение i -й переменной глобального оптимума, $i = \overline{1, n}$, $p_i(t)$ – соответствующая i -й компонента вектора вероятностей (5.12).

Данное свойство можно использовать для анализа и прогноза изменений компонент вектора вероятностей и прогноза точки, в которую сойдется алгоритм. Данное свойство можно использовать в задачах большой размерности для группировки переменных. Поскольку по сепарабельным переменным возможны более сильные изменения значений (данное свойство используется в дельта-группировке), то можно предположить, что по этим переменным будет более выраженная динамика изменений и соответствующих компонент вектора вероятностей. Переменные, для которых можно сделать прогноз значения фиксируются, и оптимизация продолжается по оставшейся части переменных, т.е. решается задача меньшей размерности.

Основная идея новой эвристики декомпозиции на основе алгоритма оценки распределения EDA-GA следующая. В течение предопределённого числа итерации (t_{adapt} , период адаптации) часть случайно выбранных переменных (α_{fixed} , процент числа переменных задачи) фиксируется, оптимизация осуществляется по оставшимся переменным. Значение зафиксированных переменных определяется на основе распределения (5.12) по правилу (5.14):

$$x_i^{fixed}(t) = \begin{cases} 0, \text{ если } p_i(t) < (0.5 - \delta) \\ random, \text{ если } (0.5 - \delta) \leq p_i(t) \leq (0.5 + \delta), \\ 1, \text{ если } p_i(t) > (0.5 + \delta) \end{cases} \quad (5.14)$$

где $\delta \in (0, 0.5)$ – пороговое значение для принятия решения, о том, что алгоритм сходится к значению 1 или 0.

Значения α_{fixed} , δ и t_{adapt} являются параметрами алгоритма. В работе [347] было показано, что на множестве эталонных тестовых задач глобальной оптимизации IEEE CEC большой размерности [128, 129] лучшие в среднем по точности нахождения глобального оптимума результаты получаются при $\alpha_{fixed} = 50\%$ и $\delta = 0.15$ для алгоритма с одной популяцией без использования кооперативной коэволюции, а при использовании коэволюции на основе островной модели ЭА – $\alpha_{fixed} = 75\%$ и $\delta = 0.15$. Размер периода адаптации зависит от ограничений на максимальное число вычислений целевой функции конкретной задачи.

Наличие порога δ позволяет повысить точность прогноза значений переменных, т.к. небольшие отклонения значений компонент вектора вероятностей могут быть вызваны действием оператора мутации. На рисунке 5.5 показан пример изменения одной из компонент вектора вероятностей для произвольного запуска ГА на функции Растригина, значение данной переменной в точке глобального оптимума равно нулю.

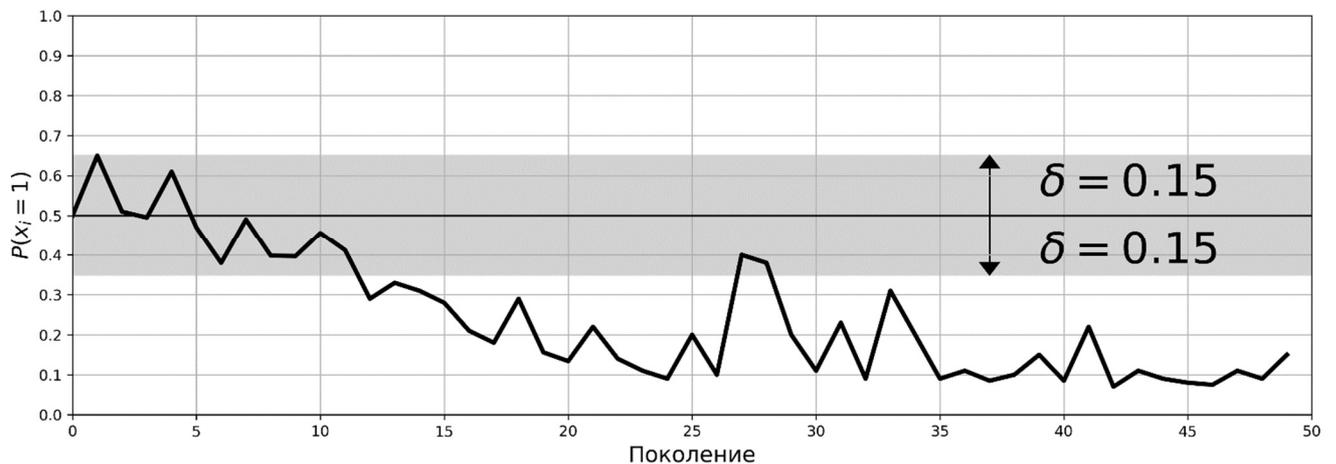


Рисунок 5.5 – Пример динамики изменения компоненты вектора вероятностей

Несмотря на то, что решение для прогноза значений в EDA-GA делается по каждой переменной отдельно, оценка распределения (5.12) формируется по всей популяции и отражает глобальное поведение ГА. Поэтому эвристика EDA-GA

может применяться не только для сепарабельных задач. В [347] было показано, что EDA-GA демонстрирует высокую эффективность при решении эталонных тестовых задач и превосходит некоторые из известных алгоритмов. Стоит отметить, что ведущие алгоритмы LSGO предназначены для решения только задач с вещественными переменными.

Известные методы группировки переменных формируют субкомпоненты задачи равного размера, число которых является параметром алгоритма. В данной работе предложена новая эвристика для случайной адаптивной группировки в компоненты переменного размера AVS-RG (*adaptive variable-size random grouping*), обобщающая метод случайной группировки DECC-G.

В методе декомпозиции с применением кооперативной коэволюции (Алгоритм 9) число поколений ЭА, когда выполняется оптимизация с использованием одного из варианта декомпозиции, обозначим как цикл ($cycle = 1, numCycles$), где $numCycles$ – максимальное число циклов. Обозначим s_j – размер j -й субкомпоненты \bar{x}_j , полученной путем группировки переменных вектора решения \bar{x} . Возможны следующие способы формирования субкомпонент:

1. Предопределенное число m субкомпонент равного размера: $s_1 = s_2 = \dots = s_m = s$, $m \cdot s = n$. Данный подход наиболее часто используется в известных эвристиках группировки переменных. Очевидно, что для произвольной задачи число групп и их состав может варьироваться, следовательно, эффективность данного способа ограничена.

2. Предопределенное число m субкомпонент произвольного размера: $\{s_1, s_2, \dots, s_m\}$, $\sum_{i=1}^m s_i = n$. Данный подход используется в методах статической группировки, которые, как отмечалось ранее, обычно проигрывают методам динамической группировки. Можно формировать группы произвольного размера в заданном диапазоне $\forall i, s_i \in [s_{min}, s_{max}]$, задав фиксированное число субкомпонент m . Однако данный подход все еще не будет достаточно гибким, когда требуется использовать число групп, отличное от m .

3. Универсальным вариантом является подход, в котором $\forall i, s_i \in [0, n]$ при условии $\sum_{i=1}^m s_i = n$. В случае, когда размер некоторых групп будет равен нулю, алгоритм может сформировать группы большего размера, сократив число субкомпонент, в случае групп с одной переменной мы получим классический вариант коэволюции Де Йонга, а при наличии одной группы, выючащей все переменные задачи, на очередном цикле оптимизации ЭА будет решать традиционную задачу оптимизации без декомпозиции.

Алгоритме AVS-RG использует третий подход, а для формирования групп применяется идея метода адаптации вероятностей на уровне популяции PDP. Для этого введем следующий вектор вероятностей (5.15)-(5.16):

$$\bar{P}(t) = (\bar{p}_1(t), \dots, \bar{p}_n(t)), \quad (5.15)$$

$$\bar{p}_i(t) = (p_i^1(t), \dots, p_i^k(t)), p_i^j(t) = P(x_i \in s_j) \wedge \sum_{j=1}^m p_i^j(t) = 1, \quad (5.16)$$

где n – размерность задачи оптимизации, m – максимальное число компонент ($m \leq n$), s_j – j -я компонента, $P(x_i \in s_j)$ – вероятность, назначения i -й переменной j -й компоненте.

На этапе инициализации при отсутствии информации о взаимозависимых переменных, используется случайное равномерное заполнение групп. Переменные попадают в субкомпоненты с равной вероятностью $\forall i, j: p_i^j(t=0) = \frac{1}{m}$.

В случае когда $\forall i, j, t: p_i^j(t) = \frac{1}{m}$, эвристика AVS-RG совпадает с методом случайной динамической группировки, предложенной в DECC-G. Однако, более перспективным вариантом является использование обратной связи об эффективности решения задачи с текущей группировкой переменных для адаптации вероятностей в распределении (5.15)-(5.16). Для этого будем использовать следующую идею. После заданного числа поколений, определяются субкомпоненты, для которых было найдено решение превосходящее предыдущее лучшее найденное, обозначим их как bEA , а оставшиеся как wEA . Значения вероятностей в распределении (5.15)-(5.16), соответствующих эффективным группам bEA , увеличиваются путем уменьшения вероятностей менее эффективных

групп wEA . Если новое лучшее решение не было найдено, компоненты ранжируются по средней эффективности в текущем периоде адаптации, вероятности перераспределяются пропорционально рангам. Заданный процент α субкомпонент с высшими рангами включается в множество bEA , остальные – в множество wEA . Адаптация вероятностей в AVS-RG происходит по следующей схеме (Алгоритм 12):

Алгоритм 12. Алгоритм адаптации вероятностей в AVS-RG.

Вход: цикл t , лучшее найденное решение на предыдущем цикле $bestFound(t - 1)$, ЭА для оптимизации каждой из компонент $(EA_j, j = \overline{1, m})$, параметры Δp и θ .

$sumP = 0$

Если $(\exists EA_j: bestFound_j(t) > bestFound(t - 1))$, **то**

$\forall j = \overline{1, m}$

Если $(bestFound_j(t) > bestFound(t - 1))$, **то**

$\forall i = \overline{1, n}$

Если $((i \notin s_j) \wedge ((p_i^j(t) - \Delta p) \geq \theta))$, **то**

$sumP += \Delta p$

$p_i^j(t + 1) -= \Delta p$

$\forall j = \overline{1, m}$

$\forall i = \overline{1, n}$

Если $(i \in s_j)$, **то**

$p_i^j(t + 1) += \frac{sumP}{|\{EA_j, j = \overline{1, m}: (bestFound_j(t) > bestFound(t - 1))\}|}$

иначе

Отсортировать субкомпоненты по средней пригодности, процент α лучших групп включить в bEA , остальные – в wEA

$\forall j = \overline{1, m}$

Если $(EA_j \in wEA)$, **то**

$\forall i = \overline{1, n}$

Если $((p_i^j(t) - \Delta p) \geq \theta)$, **то**

$sumP += \Delta p$

$p_i^j(t + 1) -= \Delta p$

$\forall j = \overline{1, m}$

Если $(EA_j \in bEA)$, **то**

$\forall i = \overline{1, n}$

$p_i^j(t + 1) += \frac{sumP}{|wEA|}$

Выход: $\bar{P}(t + 1)$.

Параметр Δp определяет значение, на которое будут изменяться вероятности в распределении после адаптации, параметр θ определяет пороговое значение, гарантирующее положительное значение для вероятностей.

В работе [348] было продемонстрировано, что на эталонных тестовых задачах IEEE CEC алгоритм AVS-RG сравним в среднем по точности нахождения глобального оптимума с некоторыми известными алгоритмами LSGO, при этом предложенная эвристика одинаково эффективно справляется как с сепарабельными, так и несепарабельными задачами.

Поскольку вышерассмотренные и предложенные эвристики реализуют различные схемы группировки, на разных задачах и разных этапах процесса оптимизации требуется выбор подходящей эвристики или комбинации эвристик.

5.2 Селективная гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности

В предыдущих главах были предложены гиперэвристики на основе идей метода портфолио алгоритмов и метода адаптации вероятностей на уровне популяции. Для всех классов, рассмотренных ранее, базовые эвристики в составе гиперэвристики решали одну и ту же задачу оптимизации. Для решения задач глобальной оптимизации большой размерности с помощью кооперативной коэволюции на основе декомпозиции, разные базовые эвристики группировки переменных могут формировать субкомпоненты различные по размеру и составу, таким образом каждый ЭА, реализующий свою эвристику LSGO, будет решать собственные подзадачи. В данном исследовании гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности использует островную модель ЭА [61, 63, 349, 350]. Каждый остров реализует собственную эвристику декомпозиции, а миграции лучших индивидов будут обеспечивать эффект кооперации ЭА. Управление выбором и комбинацией базовых эвристик LSGO происходит путем изменения размеров островов (размеров популяций ЭА,

реализующих базовую эвристику). Более успешные на последнем этапе оптимизации эвристики должны увеличивать размер острова, менее успешные – уменьшать. Как было показано ранее, на разных этапах процесса решения задачи оптимизации, ЭА сталкивается с различными особенностями ландшафта целевой функции, для которых требуется выбор более эффективной эвристики, и, следовательно, даже худшие на прошлых этапах эвристики должны участвовать в работе ЭА и оцениваться. Наличие минимального порогового значения для размера острова обеспечит возможность комбинировать все используемые эвристики и выбирать при необходимости любую из них.

Преимуществом островной модели для задач большой размерности является возможность эффективного распараллеливания вычислений на ЭВМ. Практические задачи большой размерности обычно требуют значительных вычислительных ресурсов, из-за этого проектирование новых подходов и их экспериментальное исследование весьма затруднено. Например, в [351] были получены следующие оценки времени исследования эффективности одного алгоритма на множестве эталонных тестовых задач IEEE CEC LSGO 2013: при использовании высокопроизводительного процессора AMD Ryzen 7 3800ГГц требуется 850 часов (более 35 дней) непрерывных вычислений. Проблема частично решается с помощью распараллеливания вычислений. Островная модель может быть распараллелена естественным образом, т.к. каждый остров может использовать собственный выделенный ресурс (процессор (ядро) для многопроцессорных (многоядерных) ЭВМ или узел в грид-системе). Большие вычислительные затраты также ограничивают и выбор типа гиперэвристики, для решения задач LSGO более предпочтительным является онлайн гиперэвристика.

Множество базовых эвристик, включенных в разрабатываемую селективную онлайн гиперэвристику включает: случайную группировку, дифференциальную группировку, дельта-группировку, декомпозицию на основе алгоритма оценки распределения EDA-GA и случайную адаптивную группировку в компоненты переменного размера AVS-RG.

Гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности на основе островной модели имеет следующий вид:

Метод 4. Селективная онлайн гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности.

Вход: множество H базовых эвристик и использующих их ЭА, критерий оценки эффективности ЭА, критерий останова.

Инициализация: Формируются острова со случайными популяциями решений, решение каждого острова формируется на основе одной из эвристик декомпозиции.

Повторять, пока не выполнено условие останова:

Повторять заданное число итераций (период адаптации) на каждом острове:

Решать задачу оптимизации, применяя выбранную эвристику декомпозиции задачи и коэволюционный подход.

Оценить эффективность решения задачи на каждом из островов.

Изменить размеры островов, увеличивая популяции более эффективных эвристик.

Выполнить миграции индивидов по схеме «лучший замещает худшего».

Выход: лучшее найденное решение.

В настоящее время мировое научное сообщество рассматривает задачи типа LSGO с одним критерием без ограничений, поэтому основным критерием эффективности решения является точность нахождения глобального оптимума в рамках ограничения на максимальное число вычислений целевой функции. Будем использовать данный критерий на шаге оценки эффективности эвристик и перераспределения размеров островов. Для этого ранжируем эвристики по значению пригодности лучшего найденного решения (лучшая эвристика имеет меньший ранг).

Обозначим $r_i, i = \overline{1, |H|}$ ранги эвристик, $popSize_i(t)$ – размер острова (размер популяции) i -й эвристики во время t -го периода адаптации, $\Delta popSize$ – число вычислений целевой функции (число индивидов), на которое уменьшается каждый остров, θ – минимальное пороговое значение для размера острова. Будем использовать следующий алгоритм для перераспределения размеров островов:

Алгоритм 13. Алгоритм перераспределения размеров островов в методе 4.

Вход: период адаптации t , размеры $popSize_i(t)$ и ранги r_i островов $i = \overline{1, |H|}$, параметры $\Delta popSize$, θ .

$pool = 0$

Повторять для всех $i = \overline{1, |H|}$:

Если $(popSize_i(t) - \Delta popSize) \geq \theta$, **то**

$pool = pool + \Delta popSize$

$popSize_i(t) = popSize_i(t) - \Delta popSize$

Перераспределить $pool$ по правилу (а) или (б):

(а) Жадный подход:

$popSize_{r_1}(t + 1) = popSize_{r_1}(t) + pool$

(б) Пропорционально рангам:

$\forall i: popSize_i(t + 1) = popSize_i(t) + pool \cdot \frac{2 \cdot (|H| - r_i + 1)}{(|H| + 1) \cdot |H|}$

Выход: размеры $popSize_i(t + 1)$ на следующий период адаптации.

Вариант (а) на шаге перераспределения в Алгоритме 13 является традиционным подходом (например, алгоритм Co-Operation of Biology Related Algorithms (COBRA) [350]), однако демонстрирует «жадное» поведение, когда лучшая на текущий момент эвристика получает значительно большее количество вычислений. Альтернативой такому подходу может быть распределение (б) пропорционально рангам (по аналогии с линейной ранговой селекцией).

На шаге миграции используется схема «лучший замещает худшего», что обеспечивает равные стартовые позиции для всех островов на следующий период адаптации – при реализации шага кооперативной коэволюции при оценках отдельных субкомпонент, все эвристики будут использовать лучшее найденное решение всей гиперэвристикой.

5.3 Экспериментальное исследование предложенного метода

Для исследования эффективности метода 4 был выбран набор эталонных тестовых задач глобальной оптимизации алгоритмически заданных функций большой размерности, предложенный в рамках конференции IEEE CEC 2013 и используемый на конкурсах IEEE CEC и GECCO по настоящее время [129].

Бенчмарк IEEE CEC LSGO 2013 содержит 15 эталонных тестовых задач, объединенных в следующие 4 группы:

- 3 полностью сепарабельные задачи,
- 4 частично аддитивно сепарабельные задачи с одной полностью сепарабельной компонентой,
- 4 частично аддитивно сепарабельные задачи с несепарабельными компонентами,
- 3 функции с пересекающимися компонентами (улучшение решения по одной из компонент ухудшает решение по другим),
- 1 полностью несепарабельная задача.

Набор задач 2013 года является развитием предыдущего бенчмарка 2010 года и спроектирован таким образом, чтобы лучше отражать свойства прикладных задач оптимизации большой размерности.

Экспериментальное исследование метода 4 проводилось согласно условий конкурса IEEE CEC LSGO с целью сравнить результаты экспериментов с результатами, полученными другими известными подходами, участвовавшими в конкурсе:

- решается задача минимизации,
- размерность всех задач равна 1000,
- для каждой задачи заданы границы изменений переменных,
- число независимых запусков ЭА – 25,
- критерием останова является максимальное число вычислений целевой функции равное $3.0E06$.
- критерием эффективности является лучшее найденное решение,
- сравнение алгоритмов осуществляется по оценкам медианы, среднего значения и стандартного отклонения для лучших найденных решений по серии независимых прогонов.

Программная реализация алгоритмов для проведения численных исследований выполнена на языке Python 3.7 в среде научного программирования Spyder с использованием библиотек NumPy и SciPy. Исходные коды эталонных

тестовых задач IEEE CEC и процедуры оценки критериев эффективности алгоритмов на языке Python представлены авторами бенчмарка.

В ходе первой серии экспериментов задачи набора IEEE CEC LSGO были решены ЭА, использующими отдельные базовые эвристики. Основная гипотеза аналогична гипотезам предыдущих глав работы – превосходит ли метод 4 оценку случайного выбора одной из базовых эвристик, что важно для задач с алгоритмически заданными целевыми функциями. Поскольку в ходе данных экспериментов получена полная статистика всех запусков алгоритмов, ранжирование подходов при сравнении проводилось с использованием метода глубокого статистического сравнения, а также проведена проверка гипотезы о наличии статистически значимой разницы в результатах экспериментов с применением теста Манна-Уитни-Уилкоксона.

Настройки алгоритмов приведены в таблице 5.1. Значения параметров подобраны экспериментально в ходе предварительных экспериментов.

Таблица 5.1 – Основные параметры метода 4 и базовых эвристик LSGO

Параметр	Значение
Число островов	5
Начальные размеры островов	200
Период адаптации	100
<i>DropSize</i>	10
θ	50
Тип ЭА для работы с базовыми эвристиками (кроме EDA-GA)	Само-адаптивная дифференциальная эволюция с локальным поиском SaNSDE [352]
Число субкомпонентов m в методе случайной группировки	50
Число субкомпонентов m в методе дельта-группировки	50
Точность кодирования в EDA-GA	0.01
Селекция в EDA-GA	Турнир размера 2
Мутация в EDA-GA	Точечная, вероятность обратно пропорциональна длине хромосомы
Рекомбинация в EDA-GA	Формирование на основе распределения вероятностей (5.12)

Продолжение таблицы 5.2

Параметр	Значение
Параметр α_{fixed} в EDA-GA	50%
Параметр δ в EDA-GA	0.15
Параметр t_{adapt} в EDA-GA	10
Параметр Δp в AVS-RG	0.05
Параметр θ в AVS-RG	0.1

Результаты ранжирования алгоритмов по медиане и разброс рангов на задачах бенчмарка показан на рисунках 5.6 и 5.7. Результаты ранжирования алгоритмов по среднему значению и соответствующий разброс рангов показан на рисунках 5.8 и 5.9. Детальная информация с результатами экспериментов представлена в таблице Д.1 приложения Д. В приложении Д также представлены графики сходимости, согласно требования конкурса (график медианы лучшего найденного, рисунки Д.1-Д.8), по которым можно судить о поведении алгоритмов на каждой из задач. В частности, из графиков видно, что метод 4 в большинстве случаев сходится к лучшему найденному решению или значению целевой функции, близкому к лучшему найденному, быстрее алгоритмов с одной базовой эвристикой, т.е. методу 4 требуется меньше вычислительных ресурсов для решения задачи.

Как видно из таблицы Д.1 и результатов сравнения, метод 4 превосходит по медиане и среднему лучшему найденному решению как оценку случайного выбора одной из базовых эвристик (среднее по базовым эвристикам), так и показатели лучшей в среднем из эвристик применяемой отдельно. В таблице 5.2 представлены результаты проверки гипотезы о наличии статистически значимой разницы в результатах экспериментов. Как видно из таблицы, метод 4 на большинстве задач бенчмарка или превосходит, или сравним с базовыми эвристиками. Наилучшие результаты из отдельных базовых эвристик показал алгоритм AVS-RG, поскольку он позволяет формировать группы произвольного размера, что полезно для поиска сепарабельных субкомпонент, в то время как эвристики с фиксированным числом субкомпонент либо не могут разбить сепарабельные субкомпоненты (требуется

большее число групп), либо наоборот относят пересекающиеся компоненты в разные группы (требуется укрупнение групп). Отдельные эвристики могут быть эффективными для задач одного типа, но существенно проигрывают на других. Поскольку метод 4 превосходит оценку случайного выбора одной из базовых эвристик, применение метода 4 при решении задач глобальной оптимизации алгоритмически заданных функций большой размерности более предпочтительно. Последнее особенно важно на практике, поскольку задачи большой размерности обычно вычислительно сложные (требуется много машинного времени), следовательно, исследование задачи и выбор подходящей эвристики путем многократного запуска ЭА с различными базовыми эвристиками могут быть затратными или недопустимыми в рамках ограниченных вычислительных ресурсов.

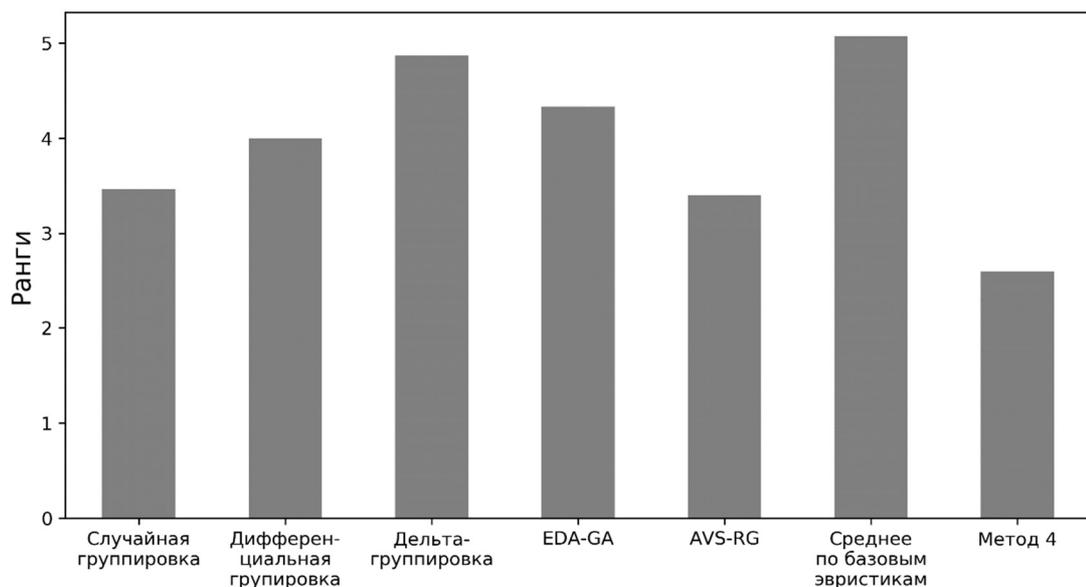


Рисунок 5.6 – Ранжирование метода 4 и базовых эвристик по значению медианы лучшего найденного решения

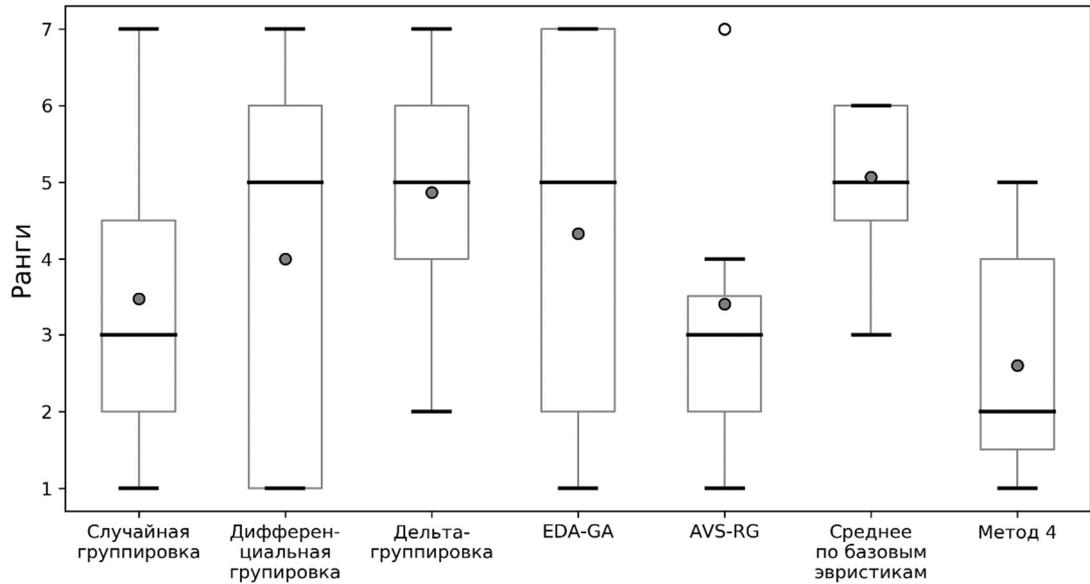


Рисунок 5.7 – Разброс рангов, полученных по медиане лучшего найденного решения

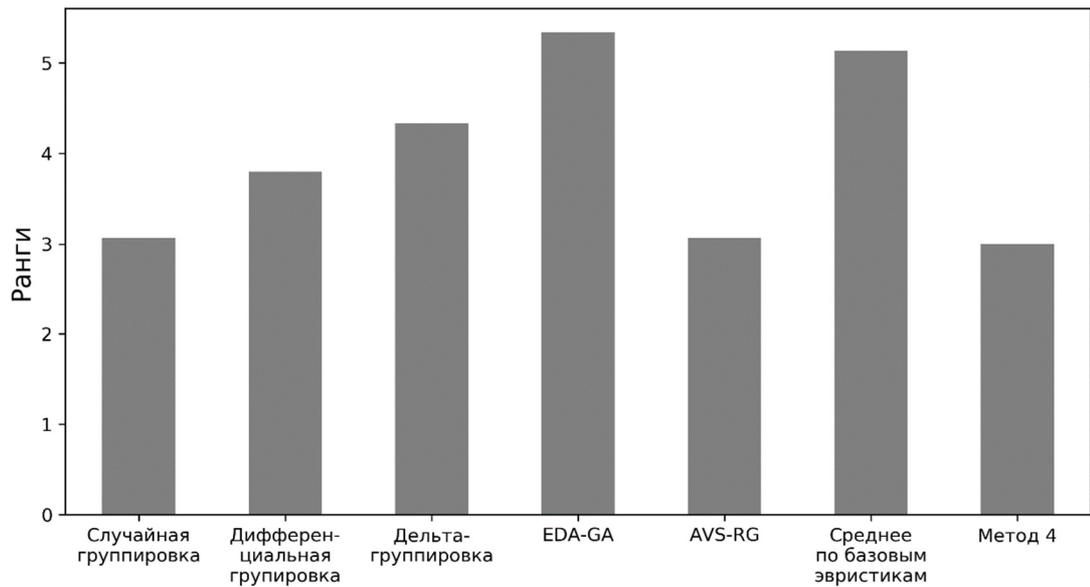


Рисунок 5.8 – Ранжирование метода 4 и базовых эвристик по значению среднего лучшего найденного решения

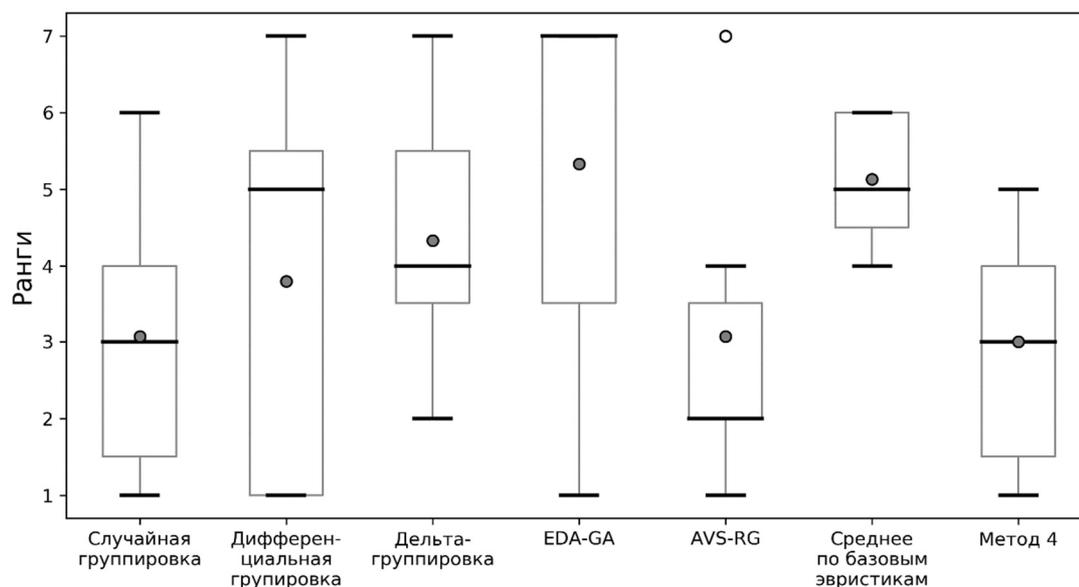


Рисунок 5.9 – Разброс рангов, полученных по среднему лучшему найденному решению

Таблица 5.2 – Результаты теста Манна-Уитни-Уилкоксона при сравнении метода 4 и базовых эвристик на задачах набора IEEE CEC LSGO 2013

Метод 4	Случайная группировка	Дифференциальная группировка	Дельта-группировка	EDA-GA	AVS-RG
Медина лучшего найденного					
Лучше	8	9	9	8	6
Нет статистически значимой разницы	4	1	4	6	6
Хуже	3	5	2	1	3
Среднее лучшего найденного					
Лучше	7	9	9	14	6
Нет статистически значимой разницы	3	1	2	0	2
Хуже	5	5	4	1	7

Результаты решения эталонных тестовых задач были сравнены результатами ведущих алгоритмов, которые побеждали и занимали призовые места на конкурсах IEEE CEC LSGO [129, 359,360]:

– кооперативная коэволюция алгоритма дифференциальной эволюции со случайной группировкой (*the random dynamic grouping DE-based CC method, DECC-G*) [129, 336], данный алгоритм является референтным, начиная с первого бенчмарка LSGO 2010;

– кооперативная коэволюция алгоритма эволюционных стратегий с масштабируемой матрицей ковариации (*scaling up covariance matrix adaptation evolution strategy using cooperative coevolution, CC-CMA-ES*) [353];

– гибридный алгоритм дифференциальной эволюции и поиска по переменной решетке (*variable mesh optimization and DE hybrid algorithm, VMO-DE*) [354];

– кооперативная коэволюция на основе сглаживающих и вспомогательных функций (*smoothing and auxiliary functions based cooperative coevolution, SACC*) [355];

– алгоритм отбора потомков, полученных разными метаэвристиками глобального и локального поиска (*multiple offspring sampling, MOS*) [356], данный алгоритм является лидером на конкурсе LSGO, начиная с 2013 года;

– итеративная гибридизация алгоритма дифференциальной эволюции и локального поиска (*iterative hybridization of DE with local search, IHDELS*) [357];

– гибридный алгоритм дифференциальной эволюции и модифицированного (эволюционного) алгоритма роя частиц (*differential evolutionary particle swarm optimization, DEEPSO*) [358].

Поскольку авторы алгоритмов предоставляют только итоговые оценки эффективности, при сравнении использовалось обычное ранжирование по функциям набора. В приложении Д представлены детальные данные сравнения алгоритмов. На рисунках 5.10-5.17 показаны результаты ранжирования, разброс рангов по задачам набора и значения рангов по задачам набора для лучших четырех алгоритмов по оценкам медианы, среднего значения, стандартного отклонения лучшего найденного по 25 прогонам и по оценке лучшего найденного за все прогоны.

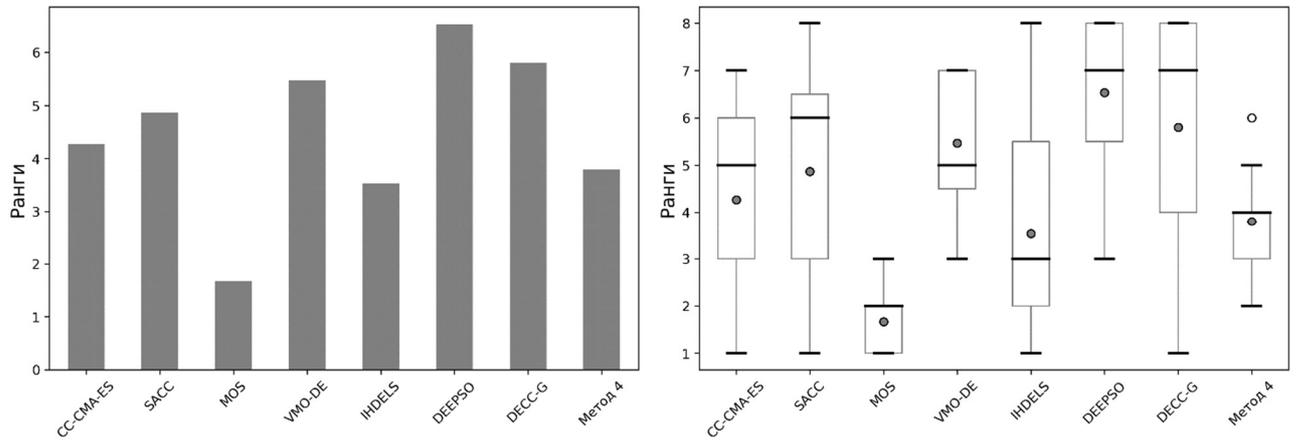


Рисунок 5.10 – Ранжирование алгоритмов по медиане (слева) и разброс рангов по задачам (справа)

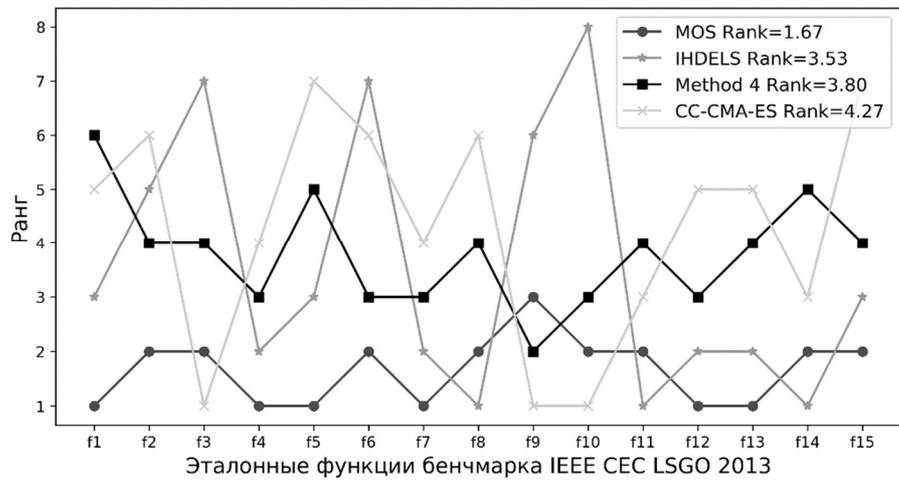


Рисунок 5.11 – Ранги по медиане для четырех лучших алгоритмов

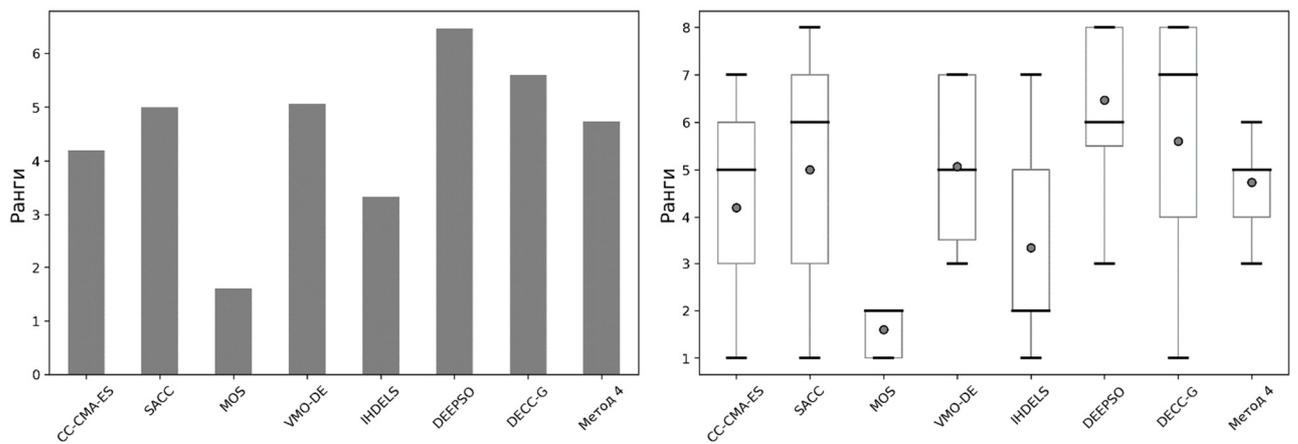


Рисунок 5.12 – Ранжирование алгоритмов по среднему (слева) и разброс рангов по задачам (справа)

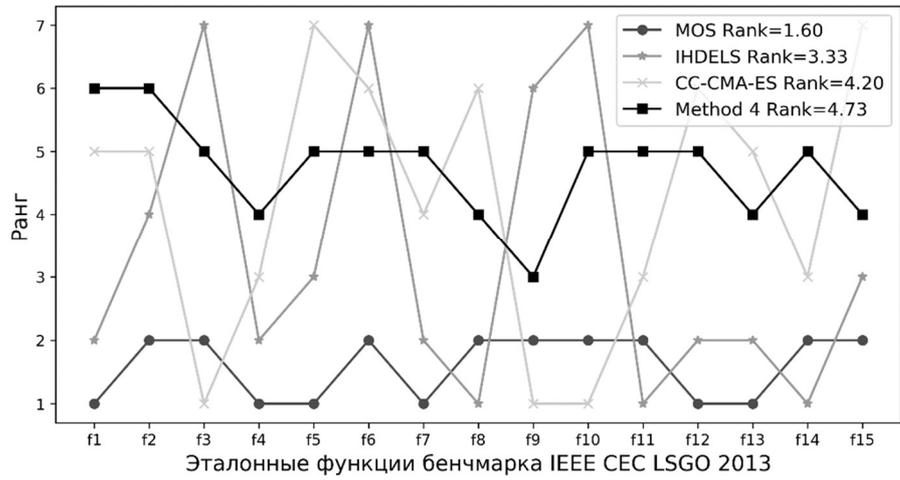


Рисунок 5.13 – Ранги по среднему для четырех лучших алгоритмов

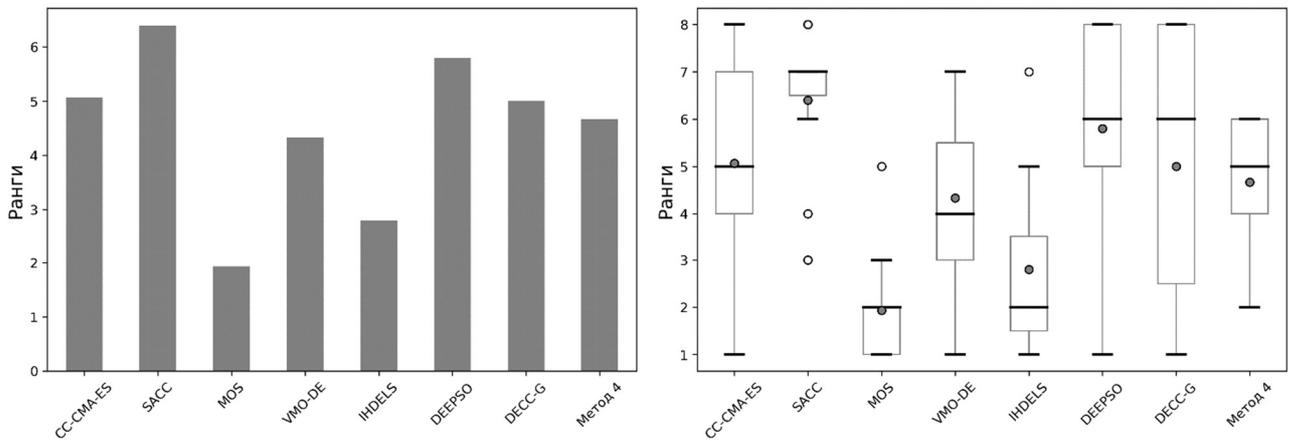


Рисунок 5.14 – Ранжирование алгоритмов по стандартному отклонению (слева) и разброс рангов по задачам (справа)

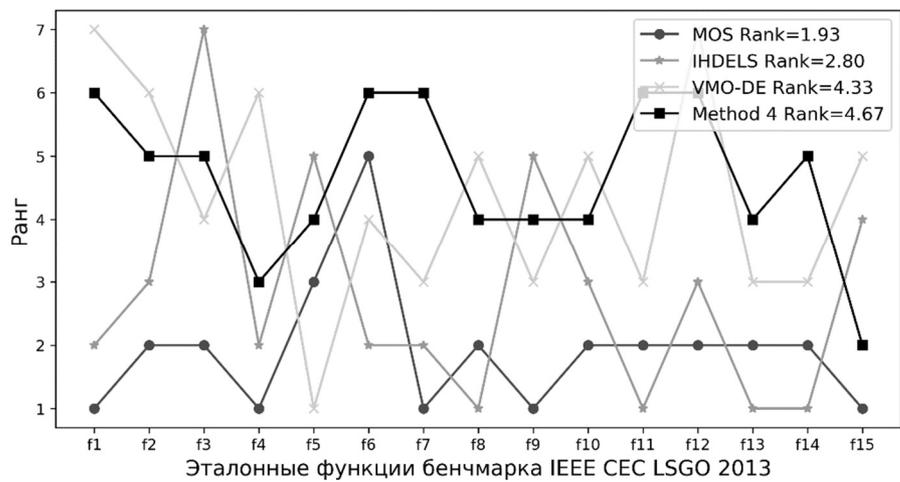


Рисунок 5.15 – Ранги по среднеквадратическому отклонению для четырех лучших алгоритмов

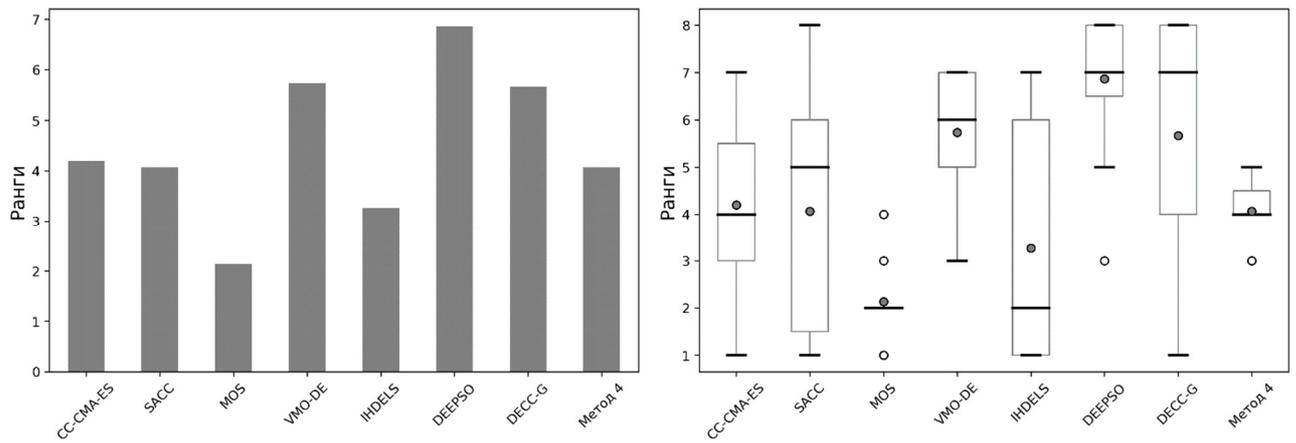


Рисунок 5.16 – Ранжирование алгоритмов по лучшему найденному (слева) и разброс рангов по задачам (справа)

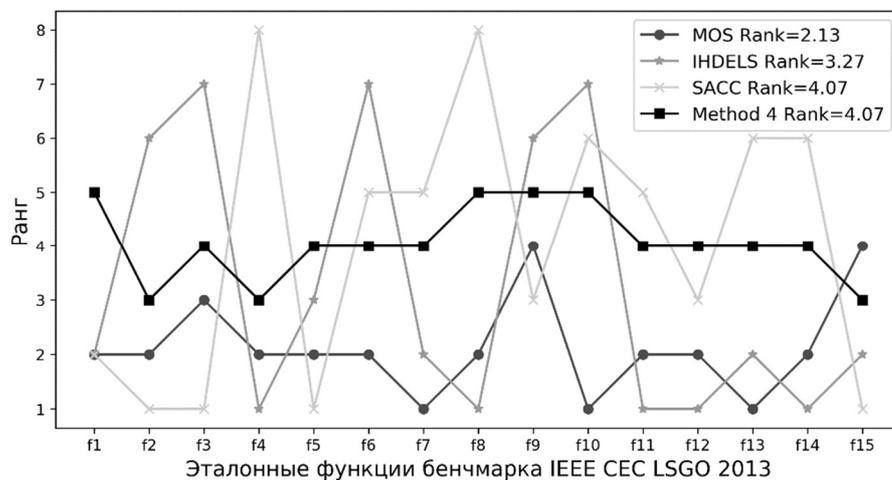


Рисунок 5.17 – Ранги по лучшему найденному для четырех лучших алгоритмов

Как видно из результатов сравнения, метод 4 занимает в среднем по набору задач третье место по оценке медианы лучшего найденного и четвертое место по остальным оценкам. При этом разброс рангов, в отличие от других подходов достаточно небольшой (лучше только алгоритм MOS), что говорит о меньшей чувствительности к типу решаемой задачи. Стоит отметить более стабильное поведение и на наиболее сложных задачах (f_{11} - f_{15}).

Метод 4 также превосходит оценку случайного выбора одного из алгоритмов, участвовавших в сравнении (таблица Д.2 приложения Д). По значению медианы метод 4 превзошел среднее по алгоритмам 11 раз, на одной задаче 1 раз был лучше алгоритма MOS и на 5 задачах был лучше алгоритма IHDELS. По оценке среднего

значения – превзошел среднее по алгоритмам 11 раз, ни разу не превзошел алгоритм MOS и на 4 задачах был лучше алгоритма IHDELS. По оценке среднеквадратического отклонения – превзошел среднее по алгоритмам 13 раз, ни разу не превзошел алгоритм MOS и на 4 задачах был лучше алгоритма IHDELS. По значению лучшего найденного за все прогоны метод 4 превзошел среднее по алгоритмам 15 раз на всех функциях, на одной задаче 1 раз был лучше алгоритма MOS и на 5 задачах был лучше алгоритма IHDELS. Стоит отметить, что алгоритмы в сравнении специально разрабатывались и модифицировались для участия в конкурсе, метод 4 является универсальным подходом, оперирующим только базовыми эвристиками. Несмотря на то, что по итоговому усредненному рангу некоторые алгоритмы превосходят метод 4, на некоторых типах задач их эффективность существенно падает, в то время как метод 4 продолжает демонстрировать стабильные результаты. Поэтому для новых и малоизученных задач глобальной оптимизации алгоритмически заданных целевых функций большой размерности применение метода 4 может быть более перспективным.

Проведем апробацию селективной онлайн гиперэвристики синтеза ЭА для задач глобальной оптимизации большой размерности на практической задаче и сопоставим результаты с ранее полученными специально спроектированными подходами.

5.4 Решение практической задачи повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах

Традиционный, централизованный способ производства электроэнергии подразумевает использование больших электростанций (ТЭЦ, атомные или ГЭС) и передачу энергии потребителям по линиям электропередачи большой мощности. В настоящее время активно внедряются и развиваются технологии распределенного производства электроэнергии. Распределенные энергетические системы включают множество, в общем случае независимых, производителей и потребителей

электроэнергии, связанных между собой. Производящие единицы (далее генераторы) являются гетерогенными, обладают разными мощностями и характеристиками эксплуатации. Как правило, распределенные сети легко масштабируются путем управления мощностью генерирующих единиц в зависимости от нагрузки сети и взаимного расположения потребителей и генераторов [361]. Для управления распределенными сетями, требуется сбор и анализ информации о генерируемой и потребляемой энергии, поэтому системы управления распределенными сетями часто называют «умными сетями электроснабжения» (*smart grid*).

Одной из задач управления распределенными сетями электроснабжения является задача повышения энергоэффективности диспетчеризации электроэнергии [362]. Задача заключается в минимизации затрат на выработку необходимой для потребителей мощности множеством генерирующих единиц с учетом технических характеристик и ограничений на рабочие режимы различных генераторов в системе. На производственных предприятиях число генерирующих единиц может достигать нескольких десятков, а в развитых «умных сетях электроснабжения» (например, в европейских странах, где активно используются возобновляемые источники энергии) – сотен и тысяч.

Задача диспетчеризации решается на некоторый период (обычно один час), для которого состав и структура распределённой энергетической системы считается известным и неизменным. Такая задача называется статической задачей повышения энергоэффективности диспетчеризации электроэнергии (*static economic load dispatch problem*, ELD). Иначе решается более сложная динамическая задача, в которой необходимо учитывать прошлые состояния распределённой энергетической системы.

Статическая задача повышения энергоэффективности диспетчеризации электроэнергии может быть сформулирована как задача однокритериальной условной оптимизации [363, 367]. Целевая функция и ограничения задаются алгоритмически, для них характерны: нелинейность, отсутствие свойств гладкости

и недифференцируемости. Размерность задачи определяется типами и числом генераторов энергии. Формализуем задачу и дадим оптимизационную постановку.

Модель оценки затрат на производство энергии на i -м генераторе имеет вид (5.17) или (5.18). Модель (5.18) учитывает эффект переключения клапанов для некоторых типов генераторов (например, на ТЭЦ пар идет через группы форсунок, для увеличения мощности последовательно включают в работу новые группы).

$$f_i(P_i) = a_i \cdot P_i^2 + b_i \cdot P_i + c_i, \quad (5.17)$$

$$f_i(P_i) = a_i \cdot P_i^2 + b_i \cdot P_i + c_i + \left| d_i \cdot \sin \left(e_i \cdot (P_i^{min} - P_i) \right) \right|, \quad (5.18)$$

где $P_i \in [P_i^{min}, P_i^{max}]$, $i = \overline{1, numUnits}$ – мощность энергии, производимой i -м генератором, $numUnits$ – количество генераторов, коэффициенты a_i, b_i, c_i и d_i, e_i – определяются характеристиками (типом) конкретного генератора.

Будем использовать модель (5.18), в которой коэффициенты d_i, e_i принимают нулевые значения, если не учитывается эффект переключения клапанов.

Цель решения задачи оптимизации энергоэффективности диспетчеризации электроэнергии – поиск значений P_i , обеспечивающих минимум совокупных затрат распределённой системы. При этом накладывается ряд ограничений, связанных с техническими требованиями и ограничениями отдельных генераторов и распределённой энергетической системы в целом.

Ограничение на генерируемую мощность:

$$\forall i: P_i^{min} \leq P_i \leq P_i^{max}, \quad (5.19)$$

Требование баланса мощности – равенства производимой энергии и суммы потребляемой энергии (нагрузка) и потерь в сети:

$$\sum_{i=1}^{numUnits} P_i = P_{demand} + P_{losses}, \quad (5.20)$$

$$P_{losses} = \sum_{i=1}^{numUnits} \sum_{j=1}^{numUnits} P_i \cdot B_{ij} \cdot P_j + \sum_{i=1}^{numUnits} B_{0i} \cdot P_i + B_{00}, \quad (5.21)$$

где B, B_0, B_{00} – матрицы коэффициентов потерь на передачу, определяемые для конкретной распределенной сети. С увеличением числа генераторов, потери на передачу обычно становятся несущественными.

Ограничение на скорость изменения мощности генератора. Изменение мощности генератора между двумя смежными периодами ограничено сверху и снизу характеристиками рампы:

$$\max(P_i^{min}, P_i^0 - DR_i) \leq P_i \leq \min(P_i^{max}, P_i^0 + UR_i), \quad (5.22)$$

где DR_i и UR_i – нижний и верхний предел скорости изменения мощности i -м генератором, P_i^0 – значение мощности при запуске генератора.

Ограничения на недопустимые режимы работы, связанные с условиями эксплуатации, ограничениями оборудования и другими требованиями. Для генератора могут устанавливаться несколько недопустимых режимов:

$$P_i \leq P_k^{lowPOZ} \wedge P_i \geq P_k^{upperPOZ}, k = \overline{1, numPOZ}, \quad (5.23)$$

где $numPOZ$ – число недопустимых режимов, P_k^{lowPOZ} и $P_k^{upperPOZ}$ – нижняя и верхняя границы i -го недопустимого режима.

Поскольку требования являются строгими, представим их в виде ограничений типа равенство:

$$h_1(P) = |P_{demand} + P_{losses} - \sum_{i=1}^{numUnits} P_i| = 0, \quad (5.24)$$

$$h_2(P) = \sum_{i=1}^{numUnits} \left((|P_i - P_i^{min}| - (P_i - P_i^{min})) + \right. \\ \left. + (|P_i^{max} - P_i| - (P_i^{max} - P_i)) \right) = 0, \quad (5.25)$$

$$h_3(P) = \sum_{i=1}^{numUnits} \left((|P_i - DR_i^{limit}| - (P_i - DR_i^{limit})) + \right. \\ \left. + (|UR_i^{limit} - P_i| - (UR_i^{limit} - P_i)) \right) = 0, \quad (5.26)$$

где $DR_i^{limit} = \max(P_i^{min}, P_i^0 - DR_i)$ и $UR_i^{limit} = \min(P_i^{max}, P_i^0 + UR_i)$.

$$h_4(P) = \sum_{i=1}^{numUnits} (\min(P_i - P_k^{lowPOZ}, P_k^{upperPOZ} - P_i), \\ \text{если } \exists k: k = \overline{1, numPOZ}: P_i \in (P_k^{lowPOZ}, P_k^{upperPOZ})) = 0, \quad (5.27)$$

Для учета ограничений воспользуемся методом штрафных функций [164]. В итоге, целевая функция задачи оптимизации энергоэффективности диспетчеризации электроэнергии имеет следующий вид:

$$F(P) = \sum_{i=1}^{numUnits} f_i(P_i) + \sum_{j=1}^4 \alpha_j \cdot h_j(P) \rightarrow \min_P, \quad (5.28)$$

где $\alpha_j, j = \overline{1,4}$ – весовые коэффициенты учета ограничений.

Для решения задачи (5.28) многие исследователи используют различные эвристики совместно с методами математического программирования и метаэвристики, включая ЭА [364, 365]. Эффективность традиционных подходов на данной задаче снижается уже на размерностях 10-20. В [366] задачи с 40 генерирующими единицами относят к задаче большой размерности.

В данной работе, для апробации метода 4 решена практическая задача оптимизации энергоэффективности диспетчеризации электроэнергии с 140 источниками энергии. Исходные данные были представлены на конкурсе по решению прикладных задач оптимизации эволюционными алгоритмами в рамках конференции IEEE CEC [367]. Параметры задачи следующие: $numUnits = 140$, $P_{demand} = 49342$ [МВт], $\alpha_1 = 1E + 07$, $\alpha_2 = 1E + 05$, $\alpha_3 = 1E + 07$, $\alpha_4 = 1E + 05$. Характеристики генераторов, представленные коэффициентами a_i, b_i, c_i, d_i, e_i , и значениями $P_i^0, DR_i, UR_i, P_k^{lowPOZ}$ и $P_k^{upperPOZ}$ представлены в таблицах Д.3 и Д.4 приложения Д.

Настройки метода 4 аналогичны приведенным в таблице 5.1 за исключением следующих (значения подобраны экспериментально):

Таблица 5.3 – Параметры метода 4 в задаче оптимизации энергоэффективности диспетчеризации электроэнергии

Параметр	Значение
Число независимых запусков	25
Начальные размеры островов	100
Период адаптации	30
$\Delta popSize$	10
θ	30
Число субкомпонентов m в методе случайной группировки	20
Число субкомпонентов m в методе дельта-группировки	20
Точность кодирования в EDA-GA	$\begin{cases} 0.1, \text{ если } (P_i^{max} - P_i^{min}) \leq 100 \\ 0.5, \text{ если } (P_i^{max} - P_i^{min}) > 100 \end{cases}$

Результаты решения задачи сравнены с результатами применения алгоритма генетический алгоритм с новым типом скрещивания, использующим 3 решения (3 родительских особи) (*genetic algorithm with a new multi-parent crossover*, GA-MPC) [369]. Алгоритм GA-MPC занял первое место на конкурсе по решению прикладных задач оптимизации эволюционными алгоритмами в рамках конференции IEEE CEC [368], продемонстрировав наибольшую эффективность в среднем по множеству задач конкурса.

Результаты численных экспериментов и сравнение с результатами применения алгоритма GA-MPC представлены в таблице 5.4.

Таблица 5.4 – Результаты численных экспериментов при решении задачи оптимизации энергоэффективности диспетчеризации электроэнергии (МО – математическое ожидание, СКО – среднеквадратическое отклонение)

	Случайная группировка	Дифференциальная группировка	Дельта-группировка	EDA-GA	AVS-RG	Среднее по базовым эвристикам	Метод 4	GA-MPC
Лучшее	1820532.9	1931897.8	1799057.2	1910007.0	2067152.1	1905729.4	1712032.6	1694798.2
Медиана	2160333.2	2031767.4	1850889.7	1977852.8	2229831.5	2050134.9	1753620.3	1777037.1
Худшее	2370001.5	2137338.3	1903015.0	2040059.9	2391157.1	2168314.4	1835069.0	1962504.6
МО	2281170.1	2026473.3	1850617.9	1983088.7	2231088.6	2074487.7	1773107.3	1781334.3
СКО	174547.7	49731.2	25905.8	34124.3	81388.1	73139.4	47293.8	68395.2

Как видно из таблицы, метод 4 уступает по лучшему найденному среди всех прогонов алгоритма алгоритму GA-MPC, но демонстрирует лучшее значение медианы и среднего значения целевой функции по 25 независимым запускам. Лучшее значение по медиане или среднему можно интерпретировать как более стабильный (повторяемый) результат, что полезно при многократном решении задачи при разных начальных условиях.

Результаты численных экспериментов с ЭА использующими одну базовую эвристику показали, что метод 4 превосходит их по всем оценкам и, как следствие, и оценку случайного выбора одной из них. Результаты ранжирования и разброс результатов решения задачи показаны на рисунках 5.18 и 5.19.

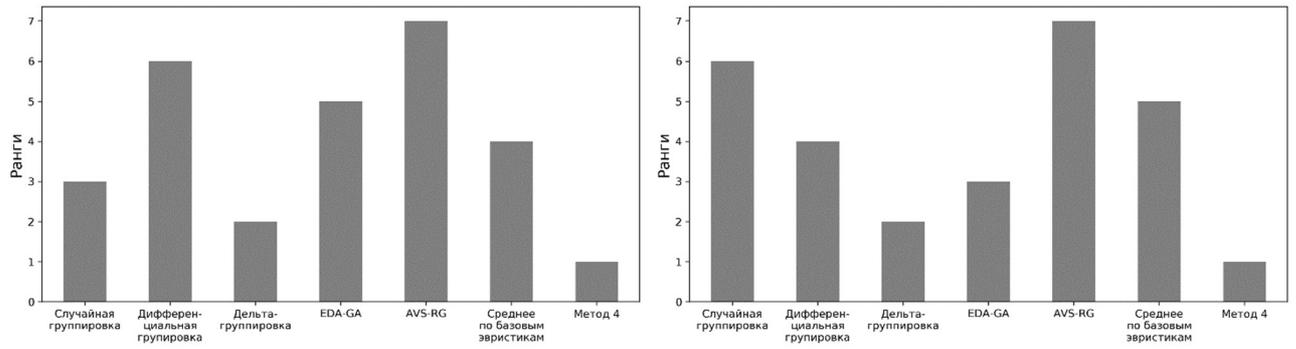


Рисунок 5.18 – Ранжирование по лучшему найденному (слева) и по медиане (справа)

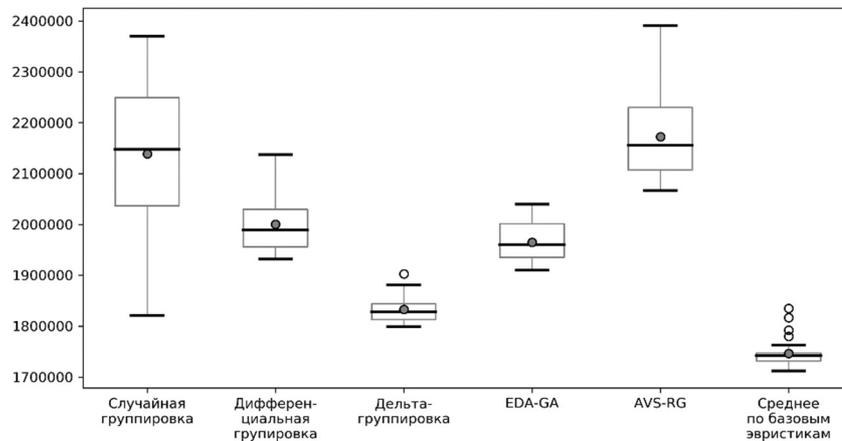


Рисунок 5.19 – Разброс значений лучшего найденного решения в серии независимых запусков

Как видно из диаграммы размаха, наименьший разброс значений показала эвристика дельта-группировки, а наибольший – случайная группировка. При этом метод 4 за счет онлайн выбора и комбинирования базовых эвристик позволяет получить более стабильные результаты. Таким образом, результаты апробации метода 4 при решении прикладной задачи, подтвердили эффективность селективной онлайн гиперэвристики.

5.5 Решение практической задачи проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов

При проектировании межпланетных миссий и миссий по исследованию объектов глубокого космоса критичной становится проблема эффективного использования топлива с целью максимизировать объем топлива (общую массу всего КА) в точке назначения. Для этих целей используется техника гравитационного маневра, связанная с использованием гравитационных полей небесных тел для целенаправленного изменения траектории и скорости (ускорение или замедление) полета КА [370]. Впервые гравитационный маневр был осуществлён в 1959 году советской автоматической межпланетной станцией Луна-3, и впоследствии стал популярным способом управления полетом как к планетам земной группы (Маринер-10, Мессенджер и др.), так и к планетам-гигантам, астероидам и объектам за пределами солнечной системы (Пионер-11, Галилео, Вояджер-1, Вояджер-2, Кассини и др.).

При проектировании траектории с использованием гравитационного маневра вначале формируется последовательность небесных тел, гравитация которых участвует в формировании траектории полета. Траектория состоит из участков между небесными телами и участков траектории на орбитах небесных тел, если совершается облет. Участки траектории, в свою очередь, разбиваются на сегменты. Обычно на участках между небесными телами осуществляется пассивный полет, а на любом из сегментов орбиты (в зоне гравитации небесного тела) КА может использовать кратковременное включение двигателя для придания КА импульса и корректировки траектории (рисунок 5.20).



Рисунок 5.20 – Пример траектории КА с использованием гравитационного маневра [371]

Математически задача проектирования траектории является задачей глобальной оптимизации с нелинейными ограничениями, цель которой – найти наилучшую траекторию для обеспечения максимальной массы КА в точке назначения. Система ограничений включает ограничения, связанные с временем полета на участках траектории, со свойствами небесных тел, с техническими ограничениями КА [370, 372-374].

Традиционно данная задача решалась экспертами в области баллистики или путем полного перебора для простых межпланетных миссий. Однако, для более сложных задач полный перебор затруднителен, а задача оптимизации является сложной, т.к. использует модель с алгоритмически заданной целевой функцией и ограничениями, в свою очередь, целевая функция имеет множество локальных оптимумов и определена не во всех точках пространства поиска. В настоящее время для решения данной задачи успешно применяются различные метаэвристики, включая эволюционные алгоритмы [372, 375].

В 2011 году на конкурсе по решению прикладных задач глобальной оптимизации были представлены задачи проектирования миссий КА Мессенджер и Кассини-2 [368]. Данные задачи являются сложными, даже несмотря на то, что размерность задач составила 26 и 22 переменные, соответственно. Тем не менее, задачи были успешно решены многими ЭА, включая самоадаптивную дифференциальную эволюцию SAMODE и генетический алгоритм с модифицированным оператором скрещивания GA-MPC [367].

В 2012 европейское космическое агентство разработало свободно распространяемое программное обеспечение Рукер для проведения научных исследований в области астродинамики [376]. Среди предложенного инструментария, Рукер имеет полный набор моделей для решения задач проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов. В настоящее время Рукер используется для проведения исследований европейским космическим агентством и НАСА (США), применялся при оптимизации предварительных сценариев миссии межпланетного спутника M-Argo и миссий исследования спутников Сатурна Титана и Энцелада, а также в проекте HERA, связанном с изучением астероидов для создания глобальной системы отражения астероидов и планируемой к запуску в 2024.

В данной работе для апробации метода 4 будут решены задачи проектирования межпланетных миссий более высокой размерности, которые можно отнести к классу задач большой размерности.

Задача проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов состоит в поиске вектора решения x , обеспечивающего максимум массы m_f КА в точке назначения (5.29):

$$m_f(x) \rightarrow \max_{x \in S} \quad (5.29)$$

Множество допустимых решений S определяется в ходе моделирования полета КА в соответствии с параметрами старта полета и использования импульсов

на траектории полета, полученными из решения x . Допустимость решения определяется в ходе симуляции по модели, которая содержит множество ограничений, контролируемых в зависимости от положения небесных тел на орбитах корректность построения траектории, корректность использования импульсов на сегментах траектории, корректность расчета входящих и исходящих углов и скоростей на орбитах в зоне действия гравитационного поля небесного тела.

Вектор решения включает следующие переменные:

- дата отправления t_0 ,
- скорость в точке отправления (относительно Земли) V_∞ ,
- стартовая масса КА m_0 ,
- на каждом участке траектории j и сегменте i направление и интенсивность импульса ΔV_{ij} ,
- для каждого гравитационного маневра, входящая и исходящая скорости относительно планеты,
- для каждого гравитационного маневра j дату маневра t_j ,
- для каждого гравитационного маневра j дату маневра m_j ,
- дата в точке назначения t_f ,
- масса в точке назначения m_f .

Все переменные скорости и импульсов выражаются в трехмерной системе декартовых координат.

Для апробации метода 4 решены две задачи:

- миссия на Юпитер, включающая гравитационный маневр на орбите Земли: Земля–Земля–Юпитер (Earth–Earth–Jupiter, EEJ);
- миссия на Меркурий, включающая серию маневров: Земля–Венера–Венера–Меркурий–Меркурий–Меркурий (Earth–Venus–Venus–Mercury–Mercury–Mercury, EVVMMM).

Исходные данные для модели в Рукер приведены в таблице 5.5. Данные взяты из [377] и представляют собой параметры реально планировавшихся миссий НАСА и европейского космического агентства.

Таблица 5.5 – Исходные данные задач проектирования траекторий КА

Параметр	EEJ	EVVMMM
Начальная масса КА, кг	20000	1300
Максимальная тяга, Н	2.26	0.34
Удельный импульс, сек	6000	3200
Диапазон дат старта	01.01.2020 – 01.01.2030	01.08.2020 – 27.04.2023
Скорость в точке отправления V_{∞} , км/с	≤ 2.0	≤ 1.925
Максимальная продолжительность полета, лет	10	12
Минимальный радиус облета, км	7000	1.1*(Радиус планеты)
Число сегментов на участках траектории	10	10
Размерность задачи оптимизации	75	222

Для данных задач ранее были получены решения алгоритмом мультистарта локального поиска (*Multistart*, MS), алгоритмом *Monotonic Basin Hopping* (BH) и методом имитации отжига (*Simulated Annealing*, SA). При решении задач помимо метода 4 был реализован алгоритм SAMODE, который можно рассматривать как конкурента гиперэвристике, т.к. алгоритм кроме адаптивного выбора значений параметров мутации и скрещивания, осуществляет структурную адаптацию, выбирая стратегию мутации.

В работе [377] авторы не указывают детали проведения экспериментов в виде количества независимых запусков алгоритмов и ограничений на число вычислений целевой функции. Для оценки алгоритмов SAMODE и метода 4 были выполнены 40 независимых запусков, с контролем стагнации в качестве критерия останова. Для учета ограничений используется метод адаптивных штрафов, предложенный в [378]. Настройки алгоритмов приведены в таблице 5.6 и 5.7, данные значения подобраны экспериментально в ходе предварительных экспериментов.

Таблица 5.6 – Настройки алгоритмов в задаче EJJ

Параметр	Значение
Критерий останова	100 итераций без изменения лучшего найденного
Метод 4	
Начальные размеры островов	100
Период адаптации	25
<i>DropSize</i>	10
θ	30
Число субкомпонентов m в методе случайной группировки	15
Число субкомпонентов m в методе дельта-группировки	15
SAMODE	
Размер популяции	250
Число вариантов схем мутации	18
Адаптация параметров F и CR	jDE

Таблица 5.7 – Настройки алгоритмов в задаче EVVMMM

Параметр	Значение
Критерий останова	100 итераций без изменения лучшего найденного
Метод 4	
Начальные размеры островов	200
Период адаптации	50
<i>DropSize</i>	10
θ	30
Число субкомпонентов m в методе случайной группировки	37
Число субкомпонентов m в методе дельта-группировки	37
SAMODE	
Размер популяции	400
Число вариантов схем мутации	18
Адаптация параметров F и CR	jDE

Результаты решения задачи представлены в таблицах 5.7 и 5.8.

Таблица 5.8 – Результаты решения задачи EEJ

	BH	SA	MS	SAMODE	Метод 4
Лучшее найденное	17102	17019	16961	17198	17231
Медиана	-	-	-	15574	16690
Среднее	16235	16302	15930	15797	16333
Стандартное отклонение	1345	1160	1320	1187	987

Таблица 5.9 – Результаты решения задачи EVVMMM

	BH	SA	MS	SAMODE	Метод 4
Лучшее найденное	1064	988	1052	1057	1089
Медиана	-	-	-	651	917
Среднее	724	522	530	649	832
Стандартное отклонение	277	253	261	278	246

Сравнение алгоритмов с одной базовой эвристикой и анализ эффективности случайного выбора одной из эвристик приведены в таблицах 5.10 и 5.11. На рисунках 5.21 и 5.22 показаны диаграммы разброса для базовых эвристик.

Таблица 5.10 – Сравнение базовых эвристик на задаче EEJ (МО – математическое ожидание, СКО – среднеквадратическое отклонение)

	Случайная группировка	Дифференциальная группировка	Дельта-группировка	EDA-GA	AVS-RG	Среднее по базовым эвристикам	Метод 4
Лучшее	17018	16787	16987	16899	17010	16940	17231
Медиана	16199	16304	15763	15561	15293	15824	16690
Худшее	12383	13954	13236	13961	13022	13311	13888
МО	15942	16019	15723	15507	15262	15691	16333
СКО	1138	874	1108	864	1122	1021	987

Таблица 5.11 – Сравнение базовых эвристик на задаче EVVMMM (МО – математическое ожидание, СКО – среднеквадратическое отклонение)

	Случайная группировка	Дифференциальная группировка	Дельта-группировка	EDA-GA	AVS-RG	Среднее по базовым эвристикам	Метод 4
Лучшее	897	915	1050	996	1002	972	1089
Медиана	686	751	780	630	846	739	917
Худшее	203	231	93	369	406	260	329
МО	652	699	672	660	777	692	832
СКО	206	219	365	196	184	234	246

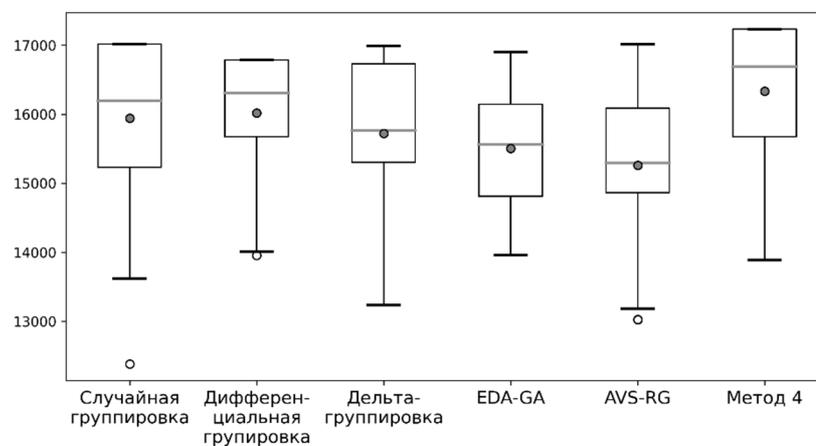


Рисунок 5.21 – Разброс значений лучшего найденного решения в серии независимых запусков для задачи EEJ

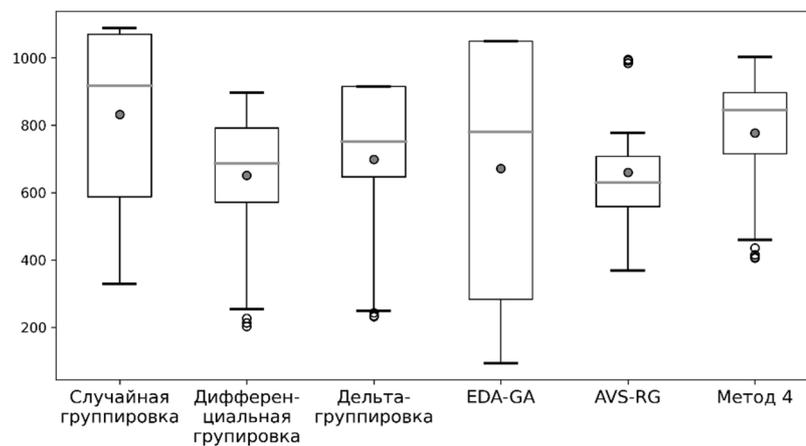


Рисунок 5.22 – Разброс значений лучшего найденного решения в серии независимых запусков для задачи EVVMMM

Как видно из результатов численных исследований, метод 4 и SAMODE превосходят ранее полученные результаты на обеих задачах. При этом метод 4 позволяет получить лучшее решение в серии запусков, а также лучшее значение медианы и более стабильные результаты (МО и СКО). Метод 4 так же позволил получить результаты лучше, чем любая из базовых эвристик, применяемая отдельно. Несмотря на то, что ни одна из базовых эвристик не показала результат по лучшему найденному и по медиане близкий к результату метода 4, гиперэвристика за счет комбинирования и управления работой эвристик обеспечивает более эффективное решение задачи.

Выводы по главе 5

В Главе 5 рассмотрена проблема решения задач глобальной оптимизации большой размерности с алгоритмически заданными целевыми функциями. Было показано, что рост размерности задач оптимизации – одна из особенностей современного ПСА. Одним из эффективных способов решения подобных задач является декомпозиция задачи путем группировки переменных. Однако, для задач с алгоритмически заданными функциями нет возможности заранее выделить сепарабельные субкомпоненты и, более того, заранее неизвестны число и размеры субкомпонент. ЭА позволяют формировать подходящие группы переменных в процессе решения задачи. Существующие методы группировки реализуют различные эвристики, которые демонстрируют разную эффективность на сепарабельных и несепарабельных задачах. Предложенная в Главе 5 селективная онлайн гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности осуществляет онлайн выбор и комбинирование эвристик, каждая из которых использует свою декомпозицию задачи. Помимо известных базовых эвристик в метод 4 были включены новые эвристики: декомпозиция на основе алгоритма оценки распределения EDA-GA и эвристика для случайной адаптивной группировки в компоненты переменного размера AVS-RG.

Экспериментальное исследование эффективности метода 4 на множестве эталонных тестовых задач показало, что предложенная гиперэвристика превосходит оценку случайного выбора одной из базовых эвристик по точности нахождения оптимума. При сравнении результатов применения метода 4 с ведущими алгоритмами глобальной оптимизации большой размерности, было показано, что, хотя метод 4 уступает лучшему из алгоритмов, он превосходит оценку случайного выбора одного из ведущих алгоритмов. При этом метод 4 менее чувствителен к типу решаемой задачи и дает более стабильные результаты на наиболее сложных задачах.

Для апробации предложенного подхода решены практические задачи. В задаче повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах целевая функция и ограничения задаются алгоритмически и являются нелинейными, негладкими и недифференцируемыми. Исследователи отмечают, что размерность задачи равная 40 уже представляет сложность для многих известных подходов, включая ЭА. Для апробации метода 4 решена задача с 140 переменными. Результаты численных экспериментов были сравнены с ранее полученными результатами. Несмотря на то, что метод 4 уступает по лучшему найденному значению по серии прогонов алгоритму GA-MPC, он демонстрирует лучшее значение медианы и среднего значения целевой функции, что говорит о более стабильном (повторяемом) результате решения практической задачи. В задаче проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов целевая функция и ограничения вычисляются в ходе компьютерной симуляции, целевая функция имеет множество локальных оптимумов и определена не во всех точках пространства поиска. При решении задач проектирования миссий Земля–Земля–Юпитер и Земля–Венера–Венера–Меркурий–Меркурий–Меркурий были решены задачи оптимизации размерности 75 и 222, соответственно. В результате численных экспериментов метод 4 превзошел все включенные в него базовые эвристики, алгоритм самоадаптивной

дифференциальной эволюции SAMODE, а также ранее полученные результаты другими известными подходами.

Рассмотренные в предыдущих главах особенности задач оптимизации сложных систем, возникающие на различных этапах прикладного системного анализа, позволили выделить классы задач оптимизации и соответствующие эвристики для их решения. Все рассмотренные классы являются задачами оптимизации с алгоритмически заданными целевыми функциями, поэтому выбор и корректное использование подходящей эвристики является самостоятельной задачей, сравнимой по сложности с исходной. Проектирование гиперэвристик путем конструирования или выбора базовых эвристик в процессе решения задачи позволяет находить эффективную комбинацию эвристик под каждую конкретную задачу оптимизации и под конкретную ситуацию в процессе решения задачи, что было подтверждено численными экспериментами при решении эталонных тестовых задач оптимизации и прикладных задач.

Прикладные задачи могут обладать одновременно свойствами нескольких классов сложных задач оптимизации, более того, системный аналитик и ЛПР могут предъявить различные требования к процедуре и результатам решения конкретной задачи оптимизации. Обобщим идею применения подхода на основе гиперэвристик и разработанные в данном диссертационном исследовании подходы и сформулируем обобщенный метод применения эволюционных гиперэвристик для решения сложных задач оптимизации.

ГЛАВА 6. ОБОБЩЕННЫЙ МЕТОД ПРИМЕНЕНИЯ ГИПЕРЭВРИСТИК ПРИ РЕШЕНИИ СЛОЖНЫХ ЗАДАЧ ОПТИМИЗАЦИИ

Глава 6 является обобщением результатов диссертационного исследования. В главе сформулирован метод решения сложных задач оптимизации на основе предложенных гиперэвристик в зависимости от целей и требований к решению задачи оптимизации. Обобщенный метод позволяет управлять применением предложенных в работе гиперэвристик в зависимости от комбинации свойств исследуемой сложной системы. Показано, что обобщенный метод применения гиперэвристик позволяет преодолеть ограничения известной теоремы «о бесплатных завтраках» в оптимизации.

6.1 Обобщенный метод применения гиперэвристик

В главе 1 отмечалось, что многие основные и вспомогательные этапы прикладного системного анализа или первоначально формулируются как задачи поиска оптимальных решений, или могут быть сведены к оптимизационным постановкам. В свою очередь свойства исследуемых сложных систем отражаются на задачах оптимизации, такие задачи можно назвать сложными задачами оптимизации. При исследовании сложных систем системный аналитик всегда работает в условиях неполноты или отсутствия информации о системе, в результате чего единственной доступной моделью становится модель «черного ящика».

Недостаток или отсутствие информации затрудняют не только выбор и применение методов решения задачи оптимизации, но и саму постановку задачи. В некоторых случаях исследователь не может априори выделить соответствующий класс задачи, и поэтому может выбрать недостаточно эффективные методы решения задачи, т.к. некоторые из свойств могут проявиться только в процессе решения (например, нестационарность в среде).

Методология прикладного системного анализа предполагает итерационный характер решения задач. Системный аналитик при получении дополнительной информации в ходе взаимодействия с исследуемой системой может уточнять используемые модели, менять или адаптировать способы применения алгоритмов анализа, оптимизации и управления, а также может уточнять или пересматривать исходные цели анализа.

Аналогичная итеративная процедура должна применяться и при решении задач оптимизации: исследователь определяет общие требования к решению задачи, исходя из целей анализа и управления сложной системой (например, однокритериальная задача глобальной оптимизации или многокритериальная, требуется одно решение или необходима аппроксимация множества экстремумов и т.д.), а уточнение класса задачи и выбор соответствующего метода или комбинации методов решения будет осуществляться по мере поступления новой или дополняющей информации. В случае использования исследуемого в данной диссертации метода на основе гиперэвристик, при идентификации новых свойств задачи необходимо дополнять множество базовых эвристик и процедуры управления ими. А если эффективное решение не может быть получено с имеющимися (известными) эвристиками, необходимо конструировать новые для исследуемой предметной области или задачи базовые эвристики.

Обобщим предложенные и исследованные в работе подходы и гиперэвристики.

Обобщенный метод применения гиперэвристик синтеза ЭА для решения сложных задач оптимизации имеет следующий вид (рисунок 6.1).

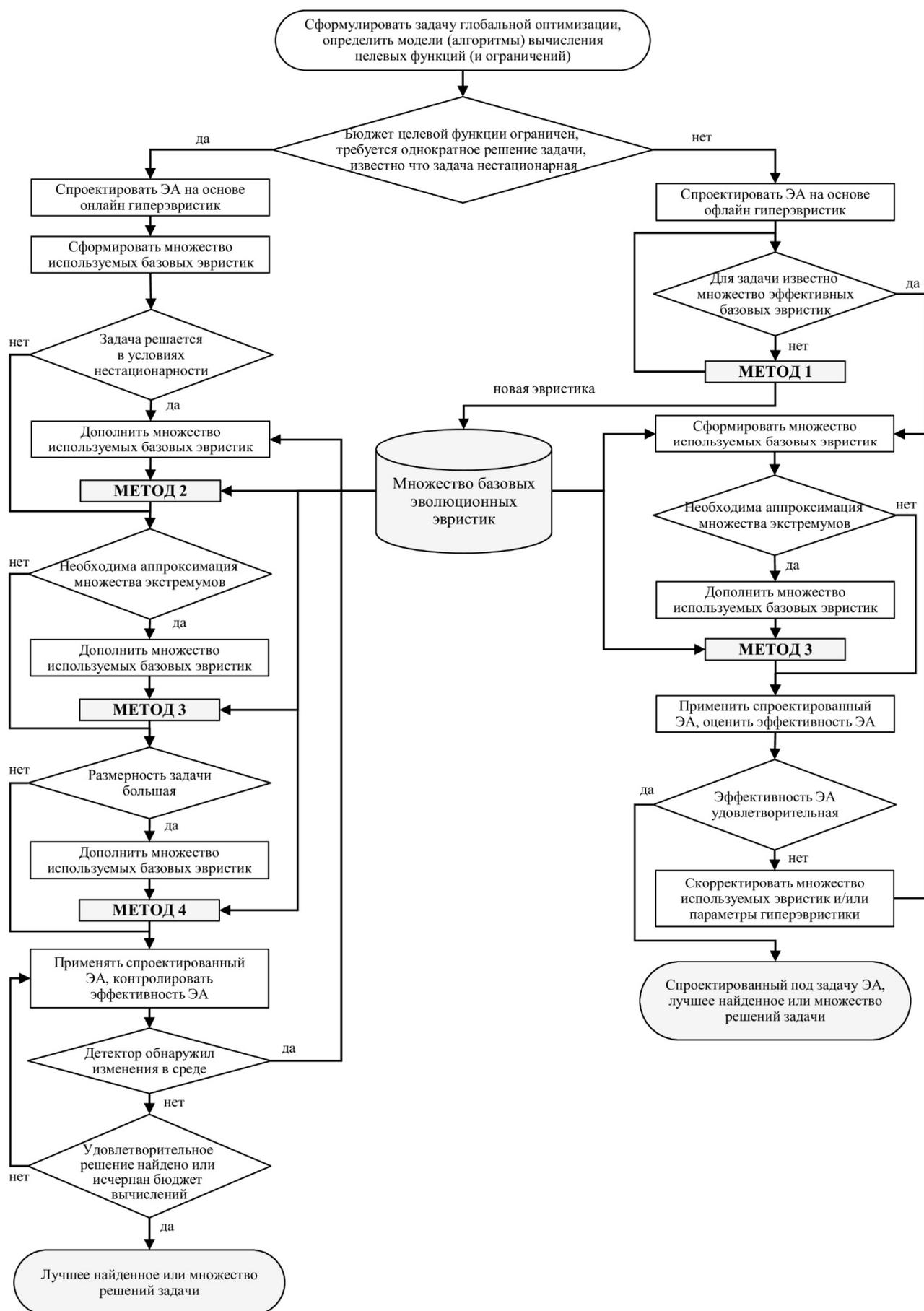


Рисунок 6.1 – Обобщенный метод применения гиперэвристик

На начальном этапе исследователь формулирует общую цель решения задачи оптимизации, дополнительные требования и ограничения, определяет способы оценки или сравнения решений (алгоритм, модель или правила проведения симуляции или экспериментальной оценки). Далее определяется тип применяемой гиперэвристики: онлайн или офлайн. Выбор типа гиперэвристики зависит от постановки задачи и ограничений.

В зависимости от ограничения на бюджет вычислений целевой функции, возможны следующие варианты:

– бюджет не ограничен, возможно многократное решение задачи. В таком случае проектируется ЭА на основе офлайн гиперэвристики. Данный вариант применим, когда оценка целевой функции осуществляется с помощью вычислительно несложных алгоритмов;

– бюджет ограничен. Бюджет может быть фиксирован (задано максимальное число вычислений целевой функции) или требуется минимизировать число вычислений целевой функции из-за их высокой стоимости (в терминах машинных операций, затрачиваемого времени или использования других ресурсов для оценок). В данном случае предпочтительно применение онлайн гиперэвристики.

В зависимости от целей решения исходной проблемы системного анализа:

– требуется однократное решение задачи оптимизации. В данном случае ЭА проектируется онлайн, пока задача решается. В области эволюционных вычислений такой подход называется «концепция (перспектива) проекта» – необходимо хотя бы один раз найти удовлетворительное решение задачи, после чего задача считается решенной и повторный поиск не требуется;

– требуется построение ЭА для решения множества задач класса. Для этого применяется офлайн гиперэвристика для синтеза ЭА, который будет эффективен в среднем для некоторого множества задач. В области эволюционных вычислений такой подход называется «концепция (перспектива) продукта» – необходимо обеспечить возможность находить удовлетворительные решения повторяющихся задач при каждом применении ЭА, возможно при различных начальных условиях (в рамках некоторого класса задач).

Если априори известно, что задача нестационарная, то применяется онлайн гиперэвристика для контроля и адаптации к изменениям.

Рассмотрим случай, когда применяется офлайн гиперэвристика.

На первом этапе исследователь должен сформировать множество базовых эвристик, которые будут использоваться при синтезе ЭА решения поставленной задачи оптимизации. Выбор множества эвристик может быть сделан на основе исходной постановки, типа синтезируемого ЭА и любой другой информации, которой обладает аналитик. Если известных базовых эвристик недостаточно (об этом или известно заранее, или выяснится в ходе экспериментов по неудовлетворительным результатам решения задачи) или задача решается впервые и известные эвристики не предложены, то применяется Метод 1 - конструктивная офлайн гиперэвристика синтеза оператора селекции на основе метода генетического программирования. Новые синтезированные базовые эвристики дополняют множество известных эвристик.

Задача глобальной оптимизации может быть сформулирована как традиционная задача поиска единственного глобально-оптимального решения или как задача аппроксимации множества экстремумов. В последнем случае к множеству базовых эвристик решения исходной задачи оптимизации добавляются эвристики мультимодальной оптимизации и для управления ими применяется Метод 3 – селективная гиперэвристика синтеза ЭА идентификации множества экстремумов.

Офлайн гиперэвристика применяется до тех пор, пока не будет получено удовлетворительное решение поставленной задачи. Результатом применения офлайн гиперэвристики являются лучшее найденное решение или множество идентифицированных оптимумов задачи оптимизации и новый ЭА, синтезированный для эффективного решения поставленной задачи или для эффективного в среднем решения множества исследуемых задач оптимизации.

Рассмотрим применение обобщённого подхода, когда применяется онлайн гиперэвристика.

На первом этапе исследователь формирует множество базовых эвристик, которые будут использоваться при синтезе ЭА решения поставленной задачи оптимизации.

Если имеется априорная информация о том, что задача решается в условиях нестационарной среды, в множество используемых эвристик добавляются эвристики нестационарной оптимизации и для управления ими применяется Метод 2 – селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации. Независимо от того, какое решение принято относительно включения Метода 2, в синтезируемом ЭА следует использовать детектор изменений в среде. Если изменения будут обнаружены в будущем в процессе решения задачи, то во множество используемых эвристик будут добавлены эвристики нестационарной оптимизации и будет активирован Метод 2.

Если исследователю требуется аппроксимация множества экстремумов, множество базовых эвристик дополняется эвристиками мультимодальной оптимизации и для управления ими применяется Метод 3 – селективная гиперэвристика синтеза ЭА идентификации множества экстремумов.

Если априори известно, что задача относится к классу задач большой размерности, то множество базовых эвристик дополняется эвристиками декомпозиции задачи и для управления ими применяется Метод 4 – селективная гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности. Как отмечалось ранее в главе 5, термин «большая размерность» зависит от природы задачи, и сложность, связанная с количеством переменных может проявляться при разных размерностях. В некоторых случаях, декомпозиция задачи может оказаться полезной и при относительно небольших размерностях, если не удастся эффективно решить задачу традиционным способом.

Спроектированный ЭА применяется для решения задачи, при этом осуществляется постоянный контроль эффективности и происходит выбор и комбинирование базовых эвристик. Критерием останова в случае онлайн гиперэвристик будут исчерпание бюджета вычислений целевой функции или найденное удовлетворительное решение поставленной задачи.

Результатом применения онлайн гиперэвристики является лучшее найденное решение или аппроксимация множества оптимумов. В отличие от офлайн гиперэвристики мы не получим в качестве результата синтезированный ЭА, т.к. он формировался и модифицировался только под конкретную задачу и конкретные ситуации в процессе решения задачи. Однако исследователь, в случае необходимости, может попытаться извлечь информацию о свойствах задачи, проведя дополнительное исследование данных об эффективности применения базовых эвристик в процессе решения задачи.

6.2 Ограничение теоремы «бесплатных завтраков не бывает»

В области численных методов оптимизации существует известная теорема «бесплатных завтраков не бывает», предложенная и доказанная Волпертом Д. и Макреди В. в 1997 (*No Free Lunch Theorems for Optimization, NFL*) [88].

Обозначим \vec{c} гистограмму (распределение) значений целевых функций некоторого алгоритма оптимизации a , полученных при решении задачи оптимизации f путем m вычислений значений целевой функции. Теорема NFL сформулирована следующим образом: для любой пары алгоритмов a_1 и a_2 выполняется равенство (6.1).

$$\sum_f P(\vec{c}|f, m, a_1) = \sum_f P(\vec{c}|f, m, a_2). \quad (6.1)$$

Согласно теоремы NFL, эффективность любых двух алгоритмов оптимизации усредненная на множестве всех возможных задач оптимизации будет одинаковой. Т.е. если алгоритм a_1 превосходит на некотором множестве задач алгоритм a_2 , то найдется множество задач такой же мощности, на котором алгоритм a_2 превзойдет a_1 .

ЭА являются универсальным подходом к решению произвольных задач оптимизации, т.к. не используют в явном виде информацию о свойствах целевой функции и пространстве поиска. Согласно теореме NFL, для любой задачи должен

найти подход, имеющий большую эффективность чем ЭА. Поэтому возникает вопрос о применимости ЭА.

Однако в 2005 Волперт и Макреди опубликовали другую работу (*Coevolutionary Free Lunches*) [379], в которой провели анализ NFL теоремы в случае, когда вместо одного алгоритма оптимизации используется подход, координирующий использование (переключение между) нескольких алгоритмов, в зависимости от того, какую эффективность они демонстрируют на каждой из задач (аналог коэволюционного подхода в ЭА). В данной работе авторы показали, что такой коэволюционный подход может превзойти по эффективности даже специально разработанный для конкретной проблемы алгоритм.

Обобщённый метод применения гиперэвристик при решении сложных задач оптимизации позволяет ослабить ограничения теоремы NFL, т.к. выбор (формирование) эффективного алгоритма происходит под конкретную задачу, т.е. гиперэвристика синтезирует узкоспециализированный алгоритм для каждого конкретного экземпляра целевой функции и класса задач.

Более того, выбор (формирование) эффективного алгоритма происходит и на разных этапах процесса оптимизации. Процесс решения задачи можно условно разделить на этапы:

- исследование пространства поиска – ЭА создает случайную популяцию и равномерно исследует пространство поиска,

- исследование областей притяжения разных оптимумов – ЭА начинает сходиться в разные зоны притяжения, но все еще продолжается исследование благодаря разнообразию в популяции,

- ЭА сходится к лучшему найденному решению с потерей разнообразия решений в популяции.

Новые особенности ландшафта целевой функции могут проявиться при локализации новых областей пространства поиска, например, область с множеством постоянства, овраг и т.д. В задачах нестационарной оптимизации, каждое изменение в среде формирует новый этап оптимизации.

Очевидно, что на каждом этапе решения задачи оптимизации эффективным будет конкретный специализированный алгоритм, а не единственный алгоритм, который является специализированным для задачи в целом. Применение онлайн гиперэвристики решает эту проблему, т.к. новый ЭА синтезируется постоянно на каждом шаге решения задачи оптимизации.

Выводы по главе 6

В Главе 6 подведен итог диссертационного исследования и сформулирован обобщенный метод применения гиперэвристик при решении сложных задач оптимизации, позволяющий в зависимости от потребностей исследователя и ограничений задачи в офлайн или онлайн режиме формировать множество базовых эвристик и управлять ими с помощью предложенных в данной работе гиперэвристик. Обоснована возможность преодоления ограничений теоремы «бесплатных завтраков не бывает», поскольку предложенный обобщенный метод автоматически формирует узкоспециализированный ЭА под каждую конкретную задачу оптимизации.

ЗАКЛЮЧЕНИЕ

В диссертации обоснованы теоретические положения и предложены новые гиперэвристики решения сложных задач глобальной оптимизации с алгоритмически заданными функциями, позволяющие автоматизировано (с минимальным участием или без участия человека) синтезировать ЭА для решения конкретной задачи или множества задач. В работе экспериментально доказано, что предложенные гиперэвристики превосходят среднюю эффективность ЭА, использующих отдельные базовые эвристики. Предложенные подходы сравнимы или превосходят среднюю эффективность некоторых известных алгоритмов, специально разработанных с учетом особенностей класса задач оптимизации. Данный результат важен для задач с алгоритмически заданными функциями, т.к. применение гиперэвристики предпочтительнее произвольного выбора одного из алгоритмов.

Цель диссертации достигнута путем решения поставленных задач, а именно:

1. Проведен анализ существующих подходов, методов и алгоритмов решения задачи автоматизации синтеза ЭА, систематизированы основные подходы к построению гиперэвристик для решения сложных задач оптимизации с алгоритмически заданными функциями, включая задачи глобальной оптимизации большой размерности, задачи аппроксимации множества экстремумов и задачи оптимизации в нестационарной среде.

2. Предложена и исследована новая конструктивная офлайн гиперэвристика на основе алгоритма ГП для синтеза операторов селекции ЭА (метод 1), позволяющая создавать новые базовые эвристики, обеспечивающие большую среднюю точность нахождения решения задачи глобальной оптимизации с алгоритмически заданными функциями на множестве задач по сравнению со стандартными операторами отбора индивидов в ЭА. Экспериментальные исследования доказали, что использование кросс-валидации на исследуемом множестве задач в ГП позволяет создать новую эвристику, которая

предпочтительнее случайного выбора одного из стандартных операторов селекции для задач типа «черный ящик».

Применение метода 1 в задаче идентификации партий электронных компонентов аэрокосмического применения позволило улучшить известные решения, полученные ранее с применением других алгоритмов.

3. Впервые предложена и исследована селективная онлайн гиперэвристика синтеза ЭА нестационарной оптимизации (метод 2), объединяющая идеи метода портфолио и идеи метода адаптации вероятностей на уровне популяции ЭА. Экспериментально доказано, что предложенная гиперэвристика превосходит оценку эффективности случайного выбора алгоритма независимо от типа нестационарной задачи по критерию офлайн-ошибки. Исследование динамики перераспределения вычислений при выборе эвристики в процессе решения задачи показало, что даже наименее эффективные в среднем эвристики вносят полезный вклад в общее решение задачи.

Применение метода 2 в задаче идентификации параметров боковой динамики движения малого беспилотного летательного аппарата с фиксированным крылом в реальном времени позволило уменьшить значение средней ошибки идентификации параметров по сравнению с решениями, полученными ранее разработчиками БПЛА.

4. Впервые предложена и исследована селективная гиперэвристика синтеза ЭА идентификации множества экстремумов (метод 3), включающая базовые эвристики, которые не используют в явном виде какую-либо информацию о свойствах целевой функции и пространства поиска. Метод 3 использует для выбора эвристик оценку доли областей притяжения обнаруженных оптимумов на основе анализа динамики и скопления индивидов. Для набора задач с бинарными переменными метод 3 обеспечил идентификацию всех известных оптимумов тестовых задач в каждом запуске алгоритма. Для задач с вещественными переменными метод 3 предпочтительнее случайного выбора отдельной эвристики, а также превосходит ряд ведущих алгоритмов с конкурса IEEE CEC 2013 по

проценту идентифицированных оптимумов и проценту успешных запусков алгоритма, в которых идентифицированы все оптимумы.

Применение метода 3 при решении практических задач поддержки принятия решений и задачи проектирования системы на нечеткой логике для решения задач классификации позволяет найти глобальный оптимум, а также обеспечивает пользователя дополнительными локально-оптимальными решениями, анализ которых позволяет уточнить модель «черного ящика».

5. Впервые предложена и исследована селективная гиперэвристика синтеза ЭА для задач глобальной оптимизации большой размерности (метод 4), на основе островной модели ЭА. Поскольку задачи глобальной оптимизации большой размерности являются вычислительно сложными, использование островной модели позволяет эффективно реализовать параллельные вычисления. Для решения задач глобальной оптимизации большой размерности предложены две новые базовые эвристики для группировки переменных при декомпозиции задач и оптимизации. Экспериментально доказано, что метод 4 предпочтительнее случайного выбора ЭА на основе одной из базовых эвристик, а также предпочтительнее случайного выбора некоторых из известных LSGO алгоритмов.

С помощью метода 4 были решены практические задачи. В задаче повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах метод 4 по серии независимых запусков уступил ранее реализованному подходу по лучшему найденному решению в серии прогонов, но показал лучшее значение медианы целевой функции, т.е. обеспечил лучшую повторяемость результатов. В задаче проектирования траектории КА с двигателями малой тяги с использованием гравитационных маневров и промежуточных импульсов метод 4 превзошел все включенные в него базовые эвристики, алгоритм самоадаптивной дифференциальной эволюции SAMODE, а также результаты, полученные ранее другими известными подходами.

6. Обобщенный метод применения гиперэвристик позволяет с учетом требований конкретной задачи глобальной оптимизации сформировать алгоритм ее решения, построенный на основе множества эвристик и критериев и способов

выбора эвристик методов 2-4, а при необходимости пополнить множество эвристик новыми, специально спроектированными автоматически для решаемой задачи.

Таким образом, в диссертации на основании выполненных исследований разработаны теоретические положения, совокупность которых можно квалифицировать как новое научное достижение в области системного анализа и обработки информации.

СПИСОК ЛИТЕРАТУРЫ

1. Оганджян, С.Б. Системный анализ / С.Б. Оганджян. – Текст : электронный // Большая российская энциклопедия. Электронная версия : [сайт]. – 2017. – URL: https://bigenc.ru/technology_and_technique/text/3666617 (дата обращения: 25.05.2021).
2. Перегудов, Ф.И. Введение в системный анализ / Ф.И. Перегудов, Ф.П. Тарасенко. – М. : Высшая школа, 1989. – 367 с.
3. Антонов, А.В. Системный анализ / А.В. Антонов. – М. : Высшая школа, 2004 – 456 с.
4. Спицнадель, В.Н. Основы системного анализа : учеб. Пособие / В.Н. Спицнадель. – СПб. : Бизнес-пресса, 2000. – 326 с.
5. Checkland, P. System Thinking, System Practice / P. Checkland. – John Wiley & Sons, 1999. – 424 p.
6. Блауберг, И.В. Становление и сущность системного подхода / И.В. Блауберг, Э.Г. Юдин. – М. : Наука, 1973. – 274 с.
7. Прангишвили, И.В. Системный подход и общесистемные закономерности / И.В. Прангишвили. – М. : СИНТЕГ, 2000. – 528 с.
8. Козлов, В.Н. Системный анализ и принятие решений / В.Н. Козлов. – СПб. : Изд-во Политехнического университета, 2009. – 223 с.
9. Анфилатов, В.С. Системный анализ в управлении : учеб. пособие / В.С. Анфилатов, А.А. Емельянов, А.А. Кукушкин ; под ред. А.А. Емельянова. – М. : Финансы и статистика, 2002. – 368 с.
10. Стаффорд, Б. Мозг фирмы : пер. с англ. / Б. Стаффорд. – М. : Едиториал УРСС, 2005. – 416 с.
11. Whitten, J.L. Systems analysis and design methods, 7th edition / J.L. Whitten, L.D. Bentley, G. Randolph, Sh. Dardan. – Irwin/McGraw-Hill, McGraw-Hill Companies Inc., 2007. – 745 p.
12. Моисеев, Н.Н. Математические задачи системного анализа / Н.Н. Моисеев. – М. : Наука, 1981. – 488 с.

13. Технология системного моделирования / Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов [и др.] ; под общ. ред. С.В. Емельянова [и др.]. – М. : Машиностроение ; Берлин: Техник, 1988. – 488 с.
14. Тарасенко, Ф.П. Прикладной системный анализ : учебное пособие / Ф.П. Тарасенко. – М. : КНОРУС, 2010. – 224 с.
15. Antamoshkin, A.N. System analysis, design and optimization an introduction / A.N. Antamoshkin, H.-P. Schwefel, A. Torn, G. Yin, A. Zilinskas. – Красноярск : Офсет, 1993. – 283 с.
16. Оптнер, С.Л. Системный анализ для решения проблем бизнеса и промышленности / С.Л. Оптнер. – 2-е изд. – М. : Концепт, 2003. – 206 с.
17. Акофф, Р. О целеустремленных системах : пер. с англ. / Р. Акофф, Ф. Эмери : под. ред. И. А. Ушакова. – Москва : Советское радио, 1974. – 272 с.
18. Волкова, В.Н. Системный анализ и принятие решений: Словарь-справочник / В.Н. Волкова : под ред. В.Н. Волковой, В.Н. Козлова. – М. : Высш. шк., 2004 – 616 с.
19. Растрингин, Л.А. Адаптация сложных систем / Л.А. Растрингин. – Рига : Зинатне, 1981. – 375 с.
20. Audet, Ch. Derivative-Free and Blackbox Optimization / Ch. Audet, W. Hare. – Springer Series in Operations Research and Financial Engineering, Springer, 2017. – 302 p.
21. Floudas, Ch.A. Encyclopedia of Optimization / Ch.A. Floudas, P.M. Pardalos. – Springer-Verlag US, 2009. – 4622 p.
22. Edelkamp, S. Heuristic Search: Theory and Applications / S. Edelkamp, S. Schroedl. – Morgan Kaufmann, Elsevier, 2012. – 712 p.
23. Sioshans, R. Optimization in Engineering Models and Algorithms / R. Sioshans, A.J. Conejo. – Springer Optimization and Its Applications, Springer, 2017. – 412 p.
24. Li, X. Decomposition and Cooperative Coevolution Techniques for Large Scale Global Optimization / X. Li. – Текст : электронный. // 2015 IEEE Congress on Evolutionary Computation. – 2015. – URL:

<http://www.cs.usm.maine.edu/~congdon/Conferences/CEC2015/Li.CEC2015.tutorial.pdf> (дата обращения: 25.05.2021).

25. Omidvar, M.N. Evolutionary Large-Scale Global Optimization. An Introduction / M.N. Omidvar, X. Li. – Текст : электронный. // The Genetic and Evolutionary Computation Conference, GECCO 2017. – 2017. – URL: <https://titan.csit.rmit.edu.au/~e46507/publications/lsgo-gecco17-omidvar-li.pdf> (дата обращения: 25.05.2021).

26. Yang, Sh. Evolutionary Computation for Dynamic Optimization Problems / Sh. Yang. – Текст : электронный // The Genetic and Evolutionary Computation Conference, GECCO 2013. – 2013. – 63 p. – URL: <http://ieee-tf-ecidue.cug.edu.cn/Yang-GECCO13-Tutorial-ECDOP.pdf> (дата обращения: 25.05.2021).

27. Ларичев, О.И. Наука и искусство принятия решения / О.И. Ларичев. – Москва : Наука, 1979. – 200 с.

28. Ларичев, О.И. Методологические проблемы практического применения системного анализа / О.И. Ларичев // Системные исследования. Методологические проблемы. Ежегодник. Под ред. Д. М. Гвишиани, В. Н. Садовского. – № 11. – 1979. – М. : Наука, 1980. – С. 210–219.

29. Preuss, M. Multimodal Optimization / M. Preuss // Tutorial. PPSN 2014 13th International Conference on Parallel Problem Solving from Nature 2014, – 20137. – 77 p. – URL: <http://ppsn2014.ijs.si/files/slides/ppsn2014-tutorial4-preuss.pdf> (дата обращения: 25.05.2021).

30. Таха, Х.А. Введение в исследование операций 7-е издание.: Пер. с англ. / Х.А. Таха. – Москва : Издательский дом «Вильямс», 2005. – 912 с.

31. Вентцель, Е.С. Исследование операций : задачи, принципы, методология / Е.С. Вентцель. – 2-е изд., стер. – М. : Наука, Гл. ред. физ.-мат. лит., 1988. – 208 с.

32. Аттетков, А.В. Методы оптимизации: Учеб. для вузов / А.В. Аттетков, С.В. Галкин, В.С. Зарубин ; Под ред. В.С. Зарубина, А.П. Крищенко. – 2-е изд., стереотип. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2003. – 440 с.

33. Мину, М. Математическое программирование. Теория и алгоритмы / М. Мину. – Издательство : Л. : Наука, 1990. – 488 с.

34. Математический анализ. Учебник для вузов. В 3х томах : доп. М-вом высш. и сред. спец. образов. СССР. Т. 1. Начальный курс / В.А. Ильин, В.А. Садовничий, Бл.Х. Сендов ; под ред. А. Н. Тихонова. – 2-е изд., перераб. – М. : МГУ, 1985. – 660 с.
35. Поляк, Б.Т. Введение в оптимизацию / Б.Т. Поляк. – М. : Наука. Главная редакция физико-математической литературы, 1983. – 384 с.
36. Карманов, В.Г. Математическое программирование : учебное пособие / В.Г. Карманов. – 5 изд., стереотип. – М. : ФИЗМАТЛИТ, 2004. – 264 с.
37. Сухарев, А.Г. Курс методов оптимизации : учебное пособие / А.Г. Сухарев, А.В. Тимохов, В.В. Федоров. – 2 изд. – М. : ФИЗМАТЛИТ, 2005. – 368 с.
38. Рубан, А.И. Методы оптимизации : учебное пособие / А.И. Рубан. – Изд. 2-ое. – Красноярск НИИ ИПУ, 2001. – 528 с.
39. Семенкин, Е.С. Оптимизация технических систем : учебное пособие / Е.С. Семенкин, О.Э. Семенкина, С.П. Коробейников. – Красноярск, СИБУП, 1996. – 282 с.
40. Griva, I. Linear and Nonlinear Optimization / I. Griva, S.G. Nash, A. Sofer ; Society for Industrial Mathematics. – 2nd Ed. – 2008. – 764 p.
41. Gill, P.E. Practical Optimization / P.E. Gill, W. Murray, M.H. Wright. – Book, Academic Press, 1981. – 419 p.
42. Растрингин, Л.А. Решение задач разношкальной оптимизации методом бинаризации / Л.А. Растрингин, Э.Э. Фрейманис. – Вопросы разработки ТАСУ. Кемерово, 1984. – вып. 3. – С. 30-38.
43. Антамошкин, А.Н. Оптимизация функционалов с булевыми переменными / А.Н. Антамошкин. – Томск : Изд-во Том. ун-та, 1987. – 104 с.
44. Gendreau, M. Handbook of Metaheuristics / M. Gendreau, J.-Y. Potvin // International Series in Operations Research & Management Science 272, Springer. – 2019. – 604 p.
45. Du, K.L. Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature / K.-L. Du, M.N.S. Swamy. – Birkhäuser Basel, 2016. – 434 p.

46. Holland, J.H. *Adaptation in Natural and Artificial Systems* / J.H. Holland. – A Bradford Book ; Reprint edition, 1992. – 232 p.
47. Fraser, A.S. *Simulation of genetic systems by automatic digital computers* / A.S. Fraser. – *Aust. J. Biol. Sci.* 10, 1957. – P. 484–99.
48. Friedberg, R.M. *A learning machine: part I* / R.M. Friedberg. – *IBM J.* 2, 1958. – P. 2-13.
49. Fogel, L.J. *Artificial Intelligence through Simulated Evolution : forty years of evolutionary programming* / L.J. Fogel, A.J. Owens, M.J. Walsh. – John Wiley & Sons, Hoboken, 1999. – 162 p.
50. Holland, J.H. *Nonlinear environments permitting efficient adaptation* / J.H. Holland // *Computer and Information Sciences II*. – New York : Academic, 1967. – P. 147-164.
51. Rechenberg, I. *Cybernetic Solution Path of an Experimental Problem* / Royal Aircraft Establishment Library Translation 1122, 1965.
52. Beyer, H.G. *Evolution strategies – A comprehensive introduction* / H.G. Beyer, H.P. Schwefel : *Natural Computing, Volume 1, Issue 1*, 2002. – P. 3–52.
53. Растрингин, Л.А. *Случайный поиск в задачах оптимизации многопараметрических систем* / Рига: Зинатне, 1965. – 212 с.
54. Barbulescu, L. *Dynamic Representations and Escaping Local Optima: Improving Genetic Algorithms and Local Search* / L. Barbulescu, J.P. Watson, D. Whitley : *Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, 2000. – P. 879-884.
55. Rana, S. *Bit Representations with a Twist* / S. Rana, D. Whitley : *International Conference on Genetic Algorithms*, 1997. – P. 188-195.
56. Vavak, F. *Performance of a Genetic Algorithm with Variable Local Search Range Relative to Frequency of the Environmental Changes* / F. Vavak, K.A. Jukes, T.C. Fogarty : *In Genetic Programming 1998: Proceedings of the Third Annual Conference*, 1998. – P. 22-25.

57. Tinos, R., A genetic algorithm with gene dependent mutation probability for non-stationary optimization problems / R. Tinos, A.C.P.L.F. de Carvalho : Proceedings of the 2004 Congress on Evolutionary Computation, Vol.2, 2004. – P. 1278-1285.

58. Философский энциклопедический словарь / М.: Советская энциклопедия // Гл. редакция: Л.Ф. Ильичёв, П.Н. Федосеев, С.М. Ковалёв, В.Г. Панов, 1983.

59. Glover, F. Future paths for integer programming and links to artificial intelligence / *Comput Oper Res* 13(5), 1986. – P. 533–549.

60. Baeck, T. Handbook of Evolutionary Computation / T. Baeck, D.V. Fogel, Z. Michalewicz : Oxford University Press, 1997. – 1130 p.

61. Курейчик, В.В. Теория эволюционных вычислений / В.В. Курейчик, В.М. Курейчик, С.И. Родзин. – М.: ФИЗМАТЛИТ, 2012. – 260 с.

62. Гладков, Л.А. Биоинспирированные методы в оптимизации / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик, П.В. Сороколетов. – М.: ФИЗМАТЛИТ, 2009. – 384 с.

63. Карпенко, А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой (2-е издание) / Москва : Издательство МГТУ им. Н. Э. Баумана, 2017. – 446с.

64. Shalyto, A. Heuristic and metaheuristic solutions of pickup and delivery problem for self-driving taxi routing / A. Shalyto, V. Shalamov, A. Filchenkov : *EVOLVING SYSTEMS*, vol: 10 (1), 2019. – P. 3-11.

65. Skobtsov, Y. Hyper-heuristical particle swarm method for MR images segmentation / Y. Skobtsov, S. El-Khatib, S. Rodzin, V. Zelentsov : *Advances in Intelligent Systems and Computing*, Volume 764, 2019. – P. 259-264.

66. Syswerda, G. Uniform crossover in genetic algorithms / *Proc. 3rd Int. Conf. on Genetic Algorithms* (Fairfax, VA, 1989) ed J D Schaffer (San Mateo, CA: Morgan Kaufmann), 1989. – P. 2–9.

67. Larrañaga, P. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation / P. Larrañaga, J.A. Lozano : Kluwer, Boston, MA, 2002.

68. Pelikan, M. A survey of optimization by building and using probabilistic models / M. Pelikan, D.E. Goldberg, F. Lobo : Computational Optimization and Applications, 21, 2002. – P. 5–20.

69. Joshi, P. A Novel Classification for Evolutionary Algorithms / P. Joshi, M. Joshi, M. Goswami : International Journal of Pure and Applied Mathematics, Volume 119, N. 17, 2018. – P. 1499-1503.

70. Calegari, P. A Taxonomy of Evolutionary Algorithms in Combinatorial Optimization / P. Calegari, G. Coray, A. Hertz, D. Kobler, P. Kuonen : Journal of Heuristics, 5, 1999. – P. 145–158.

71. Vikhar, P.A. Evolutionary Algorithms: A Critical Review and its Future Prospects / 2016 IEEE International Conference on Global Trends in Signal Processing, Information Computing and Communication, 2016. – P. 261-265.

72. Yang, X.S. Review of meta-heuristics and generalised evolutionary walk algorithm / International Journal of Bio-Inspired Computation, 3(2), 2011. – P. 77-84.

73. Буздалов, М.В. Правило "одной пятой" с возвратами для настройки размера популяции в генетическом алгоритме $(1 + (\lambda, \lambda))$ / А.О. Басин, М.В. Буздалов, А.А. Шалыто. – Моделирование и анализ информационных систем, Т. 27, № 4, 2020. – С. 488-508.

74. Buzdalov, M. Fixed-Target Runtime Analysis / M. Buzdalov, B. Doerr, C. Doerr, D. Vinokurov : GECCO'20: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, 2020. – P. 1295-1303.

75. Buzdalov, M.V. Method of artificial fitness levels for dynamics analysis of evolutionary algorithms / M.V. Buzdalov, D.V. Vinokurov : Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 5, 2020. – P. 701–707.

76. Yang, X.-Sh. Nature-Inspired Optimization Algorithms / Elsevier Insights, 2014. – 263 p.

77. De Jong, K.A. Evolutionary Computation: A Unified Approach / A Bradford book, 2006. – 259 p.

78. Растрин, Л.А. Структурная адаптация алгоритмов случайного поиска / Вопросы кибернетики, Выпуск 45, Случайный поиск в задачах оптимизации, 1978. – С. 5-13.

79. Skobtsov, Y., et al. Application of Object-Oriented Simulation in Evolutionary Algorithms / Advances in Intelligent Systems and Computing, vol. 466, 2016. – P. 453-462.

80. Тихомиров, А.В. Применение адаптивного генетического алгоритма для генерации клеточных автоматов / А.В. Тихомиров, А.А. Шалыто. – Научно-технический вестник информационных технологий, механики и оптики, № 1 (77), 2012. – С. 100-105.

81. Демидова, Л.А. Планирование работы съемочной системы с применением гибридного эволюционного алгоритма / Л.А. Демидова, Н.А. Петрова. – Вестник Рязанского государственного радиотехнического университета, № 67, 2019. – С. 82-91.

82. Demidova, L.A. A Self-tuning Multiobjective Genetic Algorithm with Application in the SVM Classification / L.A. Demidova, M.M. Egin, R.V. Tishkin : Procedia Computer Science, 2019. – P. 503-510.

83. Ochoa, G. Workshop on Self-tuning, Self-configuring and Self-generating Search Heuristics (Self* 2010) / G. Ochoa, M. Schoenauer, D. Whitley : Parallel Problem Solving from Nature – PPSN XI 11th International Conference, Kraków, Poland, September 11-15, 2010.

84. Kramer, O. Evolutionary self-adaptation: a survey of operators and strategy parameters. / Evolutionary Intelligence, 3(2), 2010. – P. 51–65.

85. Aleti, A. A Systematic Literature Review of Adaptive Parameter Control Methods for Evolutionary Algorithms / A. Aleti, I. Moser : ACM Computing Surveys, 49 (3), 2016. – P. 1–35.

86. Eiben, A.E. Parameter Control in Evolutionary Algorithms / A.E. Eiben, Z. Michalewicz, M. Schoenauer, J.E. Smith : IEEE Transactions on Evolutionary Computation, Volume: 3, Issue: 2, 1999. – P. 124–141.

87. Hinterding, R. Adaptation in evolutionary computation: a survey / R. Hinterding, Z. Michalewicz, A.E. Eiben : Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), 1997. – P. 65-69.
88. Wolpert, D.H. No free lunch theorems for optimization / D.H. Wolpert, W.G. Macready : IEEE Transactions on Evolutionary Computation, Volume: 1, Issue: 1, 1997. – P. 67-82.
89. Haraldsson, S. Automated Design of Algorithms and Genetic Improvement: Contrast and Commonalities / S. Haraldsson, J. Woodward : GECCO 2014, the 2014 Genetic and Evolutionary Computation Conference, 2014. – P. 1373-1380.
90. Hutter, F. Automatic Machine Learning: Methods, Systems, Challenges / F. Hutter, L. Kotthoff, J. Vanschoren : The Springer Series on Challenges in Machine Learning, 2019. – 219 p.
91. Azzini, A. Evolutionary ANNs: A state of the art survey / A. Azzini, A.G.B. Tettamanzi : Intelligenza Artificiale 5, 2011. – P. 19–35.
92. Denzinger, J. High performance ATP systems by combining several AI methods / J. Denzinger, M. Fuchs : in International Joint Conference on Artificial Intelligence IJCAI, 1997. – P. 102–107.
93. Pillay, N. HyperHeuristics: Theory and Applications / N. Pillay, R. Qu : Natural Computing Series, Springer, 2018. – 130 p.
94. Epitropakis, M.G. Hyper-heuristics / M.G. Epitropakis, E.K. Burke : Martí R., Panos P., Resende M. (eds) Handbook of Heuristics. Springer, 2018. – P. 1–57.
95. Burke, E.K. A Classification of Hyper-Heuristic Approaches: Revisited / E.K. Burke, M.R. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward : International Series in Operations Research & Management Science, 2018. – P. 453–477.
96. Burke, E.K., et al. Hyper-heuristics: A Survey of the State of the Art / Journal of the Operational Research Society, Volume 64, Issue 12, 2013. – P. 1695–1724.
97. Kerschke, P. Automated Algorithm Selection: Survey and Perspectives / P. Kerschke, H.H. Hoos, F. Neumann, H. Trautmann : Evolutionary Computation Volume 27, Issue 1, 2018. – P. 3-45.

98. Chakhlevitch, K. Hyperheuristics: Recent Developments / K. Chakhlevitch, P. Cowling : Adaptive and Multilevel Metaheuristics, SCI 136, Springer, 2008. – P. 3–29.
99. Ryser-Welch, P. A Review of Hyper-Heuristic Frameworks / P. Ryser-Welch, J. Miller : AISB 2014 - 50th Annual Convention of the AISB, 2014. – P. 1-7.
100. Pillay. N. A review of hyper-heuristics for educational timetabling / Annals of Operations Research, Volume 239, Issue 1, 2016. – P. 3–38.
101. Misir, M. Hyper-heuristic Bibliography / Текст : электронный – URL: <https://mustafamisir.github.io/hh.html> (дата обращения: 25.05.2021).
102. Mühlenbein, H. Mathematical analysis of evolutionary algorithms / H. Mühlenbein, T. Mahnig : In: Essays and Surveys in Metaheuristics, Operations Research, Computer Science Interface Series, 2002. – P. 525–556.
103. Eiben, A.E. Theory of evolutionary algorithms: a bird's eye view / A.E. Eiben, G. Rudolph : Theoretical Computer Science, Volume 229, Issues 1–2, 1999. – P. 3-9.
104. Baluja, S. Population-Based Incremental Learning: a Method for Integrating Genetic Search Based Function Optimization and Competitive Learning / Technical Report CMU_CS_95_163, Carnegie Mellon University, 1994. – 41 p.
105. Семенкин, Е.С. Исследование эффективности модифицированного вероятностного генетического алгоритма на десептивных функциях «ловушках» / Е.С. Семенкин, Е.А. Сопов. – Москва-Воронеж, Системы управления и информационные технологии, № 3 (45), 2011. – С. 90-95.
106. Sopov, E. The convergence prediction method for genetic and PBIL-like algorithms with binary representation / E. Sopov, S. Sopov : Proceeding of International Siberian Conference on Control and Communications (SIBCON), Tomsk: The Tomsk IEEE Chapter & Student Branch, 2011. – P. 203-206.
107. Floudas, C.A. Handbook of Test Problems in Local and Global Optimization / C.A. Floudas, P.M. Pardalos, C. Adjiman, W.R. Esposito, Z.H. Günius, S.T. Harding, J.L. Klepeis, C.A. Meyer, C.A. Schweiger : Nonconvex Optimization and its Applications, Volume 33, 1999. – 442 p.
108. Neculai, A. An Unconstrained Optimization Test Functions Collection / Advanced Modeling and Optimization, Volume 10, Number 1, 2008. – P. 147-161.

109. Momin, J. A Literature Survey of Benchmark Functions For Global Optimization Problems / J. Momin, X.S. Yang : International Journal of Mathematical Modelling and Numerical Optimisation, 4(2), 2013. – P. 150-194.

110. Garden, R.W. Analysis and classification of optimisation benchmark functions and benchmark suites / R.W. Garden, A.P. Engelbrecht : In proc. 2014 IEEE Congress on Evolutionary Computation (CEC), 2014. – P. 1641-1649.

111. Pinter, J.D. Global Optimization: Software, Test Problems, and Applications / the Handbook of Global Optimization, Volume 2, P.M. Pardalos and H.E. Romeijn, eds. Kluwer Academic Publishers, 2002. – P. 515-569.

112. Neumaier, A. Global Optimization Test Problems / Текст : электронный – URL: <https://www.mat.univie.ac.at/~neum/glopt/test.html> (дата обращения: 25.05.2021).

113. Beasley, J.E. OR-Library: test data sets for Operations Research (OR) problems / Текст : электронный – URL: <http://people.brunel.ac.uk/~mastjjb/jeb/info.html> (дата обращения: 25.05.2021).

114. Evolutionary Computation Benchmark Repository (EvoCoBR) / Текст : электронный – URL: <http://www.cs.bham.ac.uk/research/projects/ecb/> (дата обращения: 25.05.2021).

115. Bingham, D. Optimization Test Problems / Текст : электронный // Virtual Library of Simulation Experiments: Test Functions and Datasets – URL: <https://www.sfu.ca/~ssurjano/optimization.html> (дата обращения: 25.05.2021).

116. Floudas, C.A. A Collection of Test Problems for Constrained Global Optimization Algorithms / C.A. Floudas, P.M. Pardalos : Springer Science & Business Media, 1990. – 180 p.

117. Whitley, D. Building better test functions / D. Whitley, K. Mathias, S. Rana, J. Dzubera : in Proc. 6th Int. Conf. Genetic Algorithms, L. J. Eshelman, Ed., July 1995. – P. 239-246.

118. Yang, F. On constructing alternative benchmark suite for evolutionary algorithms / F. Yang, S. Lou, Y. Yin : Swarm and Evolutionary Computation 44, 2019. – P. 287-292.

119. Chen, Q. Problem Definition and Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization / Q. Chen, B. Liu, Q. Zhang, J.J. Liang, P.N. Suganthan, B.Y. Qu : Technical Report, Zhengzhou University and Technical Report, Nanyang Technological University, 2014. - URL: <https://coco.gforge.inria.fr> (дата обращения: 25.05.2021).

120. Mersmann, O. Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis / O. Mersmann, M. Preuss, H. Trautmann : Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Proceedings, Part I, 2010. – P. 73-82.

121. Michalewicz, Z. Test-case generator for nonlinear continuous parameter optimization techniques / Z. Michalewicz, K. Deb, M. Schmidt, T. Stidsen : IEEE Transactions on Evolutionary Computation, Volume: 4, Issue: 3, 2000. – P. 197-215.

122. Gaviano, M. Software for generation of classes of test functions with known local and global minima for global optimization / M. Gaviano, D.E. Kvasov, D. Lera, Ya.D. Sergeyev : ACM Transactions on Mathematical Software, 29(4), 2011. – P. 469-480.

123. Gergel, V. Flexible Generator of Constrained Global Optimization Test Problems / V. Gergel, K. Barkalova, I. Lebedev, M. Rachinskaya, A.A. Sysoyev : Proceedings LeGO – 14th International Global Optimization Workshop, AIP Conference Proceedings 2070, 2019. – 5 p.

124. Rönkkönen, J. A Generator for Multimodal Test Functions with Multiple Global Optima / J. Rönkkönen, X. LiVill, K.J. Lampinen : SEAL 2008: Simulated Evolution and Learning, LNCS 5361, 2008. – P. 239-248.

125. Ng, Ch.K. Test problem generator for unconstrained global optimization / Ch.K. Ng, D. Li : Computers & Operations Research, Volume 51, 2014. - P. 338-349.

126. Рубан, А.И. Методы оптимизации: Учебное пособие. Изд. 2-ое / Красноярск НИИ ИПУ, 2001. – 528 с.

127. Conference Ranks / URL: <http://www.conferenceranks.com> (дата обращения: 25.05.2021).

128. Ke, T. Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization / T. Ke, L. Xiaodong, Y. Zhenyu, W. Thomas : Tech. report, Univ. Sci. Technol. China, no. 1, 2010. – 21 p.

129. Li, X., Tang, K., Omidvar, M.N., Yang, Z., Qin, K. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization / X. Li, K. Tang, M.N. Omidvar, Z. Yang, K. Qin : Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013. – 23 p.

130. Moser, I. Dynamic Function Optimization: The Moving Peaks Benchmark / I. Moser, R. Chiong : Metaheuristics for Dynamic Optimization. Studies in Computational Intelligence, vol 433. Springer, 2013. – P. 35-59.

131. Li, C., et al. Benchmark Generator for CEC'2009 Competition on Dynamic Optimization / Technical report, University of Leicester, 2008. – 14 p.

132. Xiaodong, Li. Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function / Li. Xiaodong, A. Engelbrecht, M.G. Epitropakis : Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, 2013. – 10 p.

133. García, S. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power / S. García, A. Fernández, J. Luengo, F. Herrera : Information Sciences, 180, 2010. – P. 2044-2064.

134. Eftimov, T. The Impact of Statistics for Benchmarking in Evolutionary Computation Research / T. Eftimov, P. Korošec : Proceedings of the Genetic and Evolutionary Computation Conference, 2018. – P. 1329-1336

135. Eftimov, T. A Novel Approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics / T. Eftimov, P. Korošec, B. Koroušic : Information Sciences, 417, 2017. – P. 186–215.

136. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников / М.: ФИЗМАТЛИТ, 2006. – 816 с.

137. Deep Statistical Comparison / DSC API, Computer Systems Department, Jozef Stefan Institute, Ljubljana, Slovenia – URL: <http://ws.ijs.si/dsc/> (дата обращения: 25.05.2021).

138. Koza, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection / Cambridge, MA: The MIT Press, 1992. – 609 p.

139. Banzhaf, W. Genetic Programming - An Introduction / W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone : San Francisco, CA: Morgan Kaufman Publishers and Heidelberg, Germany, 1998. – 481 p.

140. Poli, R. A field guide to genetic programming / R. Poli, W. Langdon, N.F. McPhee. – Текст : электронный // Электронная версия : [сайт]. – 2008. 250 p. – URL: <http://www.gp-field-guide.org.uk> (дата обращения: 25.05.2021).

141. Langdon, W.B. Genetic Programming and Evolvable Machines: ten years of reviews / W.B. Langdon, S.M. Gustafson : Genetic Programming and Evolvable Machines, Volume 11, Issue 3–4, 2010. - P. 321-338.

142. Poli, R. Genetic Programming. An Introductory Tutorial and a Survey of Techniques and Applications / R. Poli, W. Langdon, N.F. McPhee, J.R. Koza : In Fulcher J, Jain LC (eds). Computational intelligence: a compendium. Springer, Berlin, 2008. – P. 927-1028.

143. Conor, R. Grammatical evolution: Evolving programs for an arbitrary language / R.Conor, M. Collins : European Conference on Genetic Programming, EuroGP, 1998. – P. 83-96.

144. Pillay, N. Automated Design of Machine Learning and Search Algorithms / N. Pillay, R. Qu, D. Srinivasan, B. Hammer, K. Sorensen : IEEE Computational Intelligence Magazine, 13(2), 2018. – P. 16-17.

145. Burke, E.K. Exploring Hyper-heuristic Methodologies with Genetic Programming / E.K. Burke, M.R. Hyde, G. Kendall, G. Ochoa, E. Ozcan, J.R. Woodward : Computational Intelligence, Intelligent Systems Reference Library, vol 1, Springer, 2009. – P. 177-201.

146. Oltean, M. Evolving evolutionary algorithms using linear genetic programming / Evolutionary Computation, 13(3), 2005. – P. 387–410.

147. Poli, R. Extending particle swarm optimisation via genetic programming / R. Poli, W.B. Langdon, O. Holland : In M. Keijzer, et al., editors, Proceedings of the 8th European Conference on Genetic Programming, volume 3447 of Lecture Notes in Computer Science, 2010. – P. 169-176.

148. Burke, E.K. Evolving bin packing heuristics with genetic programming / E.K. Burke, M.R. Hyde, G. Kendall : In T. P. Runarsson, et al., editors, Parallel Problem Solving from Nature - PPSN IX, volume 4193 of LNCS, 2006. – P. 860–869.

149. Iztok, F. et al. Grammatical evolution as a hyper-heuristic to evolve deterministic real-valued optimization algorithms / Genetic Programming and Evolvable Machines, Volume 19, Issue 4, 2018. – P. 473–504.

150. Pericles, M. Generation of Particle Swarm Optimization algorithms: An experimental study using Grammar-Guided Genetic Programming / M. Pericles, P. Ricardo : Applied Soft Computing, Volume 60, 2017. – P. 281-296.

151. Hong, L. Automated design of probability distributions as mutation operators for evolutionary programming using genetic programming / L. Hong, J. Woodward, J. Li, E. Ozcan : In: Genetic Programming: 16th European Conference, EuroGP 2013, 2013. – P. 85-96.

152. Woodward, J.R. The automatic generation of mutation operators for genetic algorithms / J.R. Woodward, J. Swan : In: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation (GECCO '12), 2012. – P. 67-74.

153. Goldman, B.W. Self-configuring crossover / B.W. Goldman, D.R. Tauritz : Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation - GECCO '11, 2011. – P. 575-582.

154. Blicke, T. A comparison of selection schemes used in evolutionary algorithms / T. Blicke, L. Thiele : In: Evol. Comput. 4(4), 1996. – P. 361-394.

155. Галушин, П.В. Разработка и исследование асимптотического вероятностного генетического алгоритма / П.В. Галушин, О.Э. Семёнкина. – Журн. СФУ. Сер. Матем. и физ., 5:1, 2012. – С. 46–56.

156. Kronberger, G. Symbolic Regression for Knowledge Discovery – Bloat, Overfitting, and Variable Interaction Networks / Dissertation zur Erlangung des

akademischen Grades Doktor im Doktoratsstudium der Technischen Wissenschaften, Johannes Kepler Universität Linz, 2010. – 195 p.

157. Дивеев, А.И. Классические методы символьной регрессии для поиска структур математических выражений (обзор) / А.И. Дивеев, Е.Ю. Ммалькоб – Вопросы теории безопасности и устойчивости систем, 20 (20), 2018. – С. 100-132.

158. Стрижов, В.В. Методы индуктивного порождения регрессионных моделей / Научное издание. Вычислительный центр им. А. А. Дородницына Российской академии наук, 2008. – 61 с.

159. Ивахненко, А.Г. Индуктивный метод самоорганизации моделей сложных систем / Киев: Наукова думка, 1981. – 296 с.

160. Сопов, Е.А. Эволюционные алгоритмы моделирования и оптимизации сложных систем / Диссертация на соискание ученой степени кандидата технических наук, СибГАУ, Красноярск, 2004. – 129 с.

161. Woodward, J.R. Automatically designing selection heuristics / J.R. Woodward, J. Swan : Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation - GECCO '11, 2011. – P. 583-590.

162. Sopov, E. Automated Synthesis of Selection Operators in Genetic Algorithms Using Genetic Programming / E. Sopov, E. Semenkin : International Journal on Information Technologies & Security, Vol. 9 Issue 4, 2017. – P. 13-24.

163. Richter, S.N. The automated design of probabilistic selection methods for evolutionary algorithms / S.N. Richter, D.R. Tauritz : In Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '18, 2018. – P. 1545-1552.

164. Michalewicz, Z. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods / Proceedings of the 4th Annual Conference on Evolutionary Programming, 1995. – P. 135-155.

165. Fortin, F.A. DEAP: Evolutionary Algorithms Made Easy / F.A. Fortin, F.M. De Rainville, M.A. Gardner, M. Parizeau, C. Gagné : Journal of Machine Learning Research, no 13, 2012. – P. 2171-2175.

166. COCO (COmparing Continuous Optimizers) software / URL: <http://coco.gforge.inria.fr/> (дата обращения: 25.05.2021).

167. Орлов, В.И. Качество электронной компонентной базы – залог длительной работоспособности космических аппаратов / Решетневские чтения - 2013, Т. 1., № 17, 2013. – С. 238-241.

168. Казаковцев, Л.А. Алгоритмическое обеспечение поддержки принятия решений по отбору изделий микроэлектроники для космического приборостроения / Л.А. Казаковцев, В.И. Орлов, В.В. Федосов. – СибГУ им. М. Ф. Решетнева. Красноярск, 2017. – 228 с.

169. Ким, Дж.-О. Факторный, дискриминантный и кластерный анализ / Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка. – М.: Финансы и статистика, 1989. – 215 с.

170. Xu, D. A Comprehensive Survey of Clustering Algorithms / D. Xu, Y. Tian : Annals of Data Science, Volume 2, Issue 2, 2015. – P. 165–193.

171. Казаковцев, Л.А. Метод жадных эвристик для систем автоматической группировки объектов / Диссертация на соискание ученой степени доктора технических наук, Красноярск, 2016. – 429 с.

172. Wesolowsky, G. The Weber problem: History and perspectives / Location science, No. 1, 1993. – P. 5-23.

173. Farahani, R. Facility location: Concepts, models, algorithms and case studies / Berlin Heidelberg: Springer-Verlag, 2009. – 549 p.

174. Bandyopadhyay, S. An evolutionary technique based on K-Means algorithm for optimal clustering / S. Bandyopadhyay, U. Maulik : Information Science, Vol. 146, 2002. – P. 221-237.

175. Kazakovtsev, L. Genetic algorithm with fast greedy heuristic for clustering and location problems / L. Kazakovtsev, A. Antamoshkin : Informatica (Slovenia), Volume 38, Issue 3, 2014. – P. 229-240.

176. Kazakovtsev, A. Fuzzy clustering of EEE components for space industry / A. Kazakovtsev, A. Stupina, V. Orlov, V. Stashkov : IOP Conference Series: Materials Science and Engineering, Vol. 155. 2016. – P. 012-026.

177. Kazakovtsev, L. Increase in Accuracy of the Solution of the Problem of Identification of Production Batches of Semiconductor Devices / L. Kazakovtsev, I. Rozhnov, V. Orlov : in proc. 4th International Conference on Actual Problems of Electronic Instrument Engineering, APEIE 2018, 2018. – P. 363-367.

178. Branke, J. Evolutionary Optimization in Dynamic Environments / Genetic Algorithms and Evolutionary Computation, vol. 3, Springer, 2002. – 208 p.

179. Goh, C.K. Evolutionary Multi-objective Optimization in Uncertain Environments / C.K. Goh, K.C. Tan : Studies in Computational Intelligence, vol. 186, Springer, 2009. – 271 p.

180. Ермольев, Ю. М. Методы стохастического программирования / Главная редакция физико-математической литературы изд-ва «Наука», 1976. – 240 с.

181. Birge, J.R. Introduction to Stochastic Programming / J.R. Birge, F.V. Louveaux : Springer Series in Operations Research and Financial Engineering, 2011. – 485 p.

182. Goldberg, D. Nonstationary function optimization using genetic algorithm with dominance and diploidy / D. Goldberg, R. Smith : In Proceedings of the second international conference on genetic algorithms and their application. Lawrence Erlbaum Associates Inc., 1987.

183. Weicker, K. Performance Measures for Dynamic Environments / In: Parallel Problem Solving from Nature - PPSN VII, Lecture Notes in Computer Science 2349. Springer-Verlag 2002. – P. 64-73.

184. Morrison R.W. A test problem generator for non-stationary environments / R.W. Morrison, K.A. De Jong : Proc. the 1999 Congr. on Evol. Comput. 1999. – P. 2047-2053.

185. Nguyen T.T. Evolutionary dynamic optimization: A survey of the state of the art / T.T. Nguyen, S. Yang, J. Branke : Swarm and Evolutionary Computation. № 6, 2012. – P. 1–24.

186. Branke J. Designing evolutionary algorithms for dynamic optimization problems / J. Branke, H. Schmeck : Theory and Application of Evolutionary Computation: Recent Trends. Springer-Verlag, 2002. – P. 239–262.

187. Deb, K. Two Approaches for Single and Multi-Objective Dynamic Optimization. / *Metaheuristics for Dynamic Optimization, Studies in Computational Intelligence*, vol 433. Springer, 2013. – P. 99-116.

188. Branke, J. Memory enhanced evolutionary algorithms for changing optimization problems / In *Proceedings of the IEEE Congress on evolutionary computation*, vol 3. IEEE Press, 1999. – P. 1875-1882.

189. Weicker, K. Evolutionary algorithms and dynamic optimization problems / *Der Andere Verlag*, 2003. – P. 3-4.

190. Yang, S. Non-stationary problem optimization using the primaldual genetic algorithm / In *Proceedings of the IEEE Congress on evolutionary computation*, vol 3. IEEE Press, 2003. – P. 2246-2253.

191. Yang, S. Associative memory scheme for genetic algorithms in dynamic environments / *Applications of evolutionary computing. Lecture notes in computer science*, vol 3907, Springer, Berlin, 2006. – P. 788–799.

192. Yang, S. Genetic algorithms with memory- and elitism-based immigrants in dynamic environments / *Evol Comput* 16(3), 2008.

193. Bui, L.T. Diversity as a selection pressure in dynamic environments / L.T. Bui, J. Branke, H.A. Abbass : In *Proceedings of the genetic and evolutionary computation conference*. ACM, New York, 2005.

194. Dam, H. Evolutionary online data mining: an investigation in a dynamic environment / H. Dam, C. Lokan, H. Abbass : *Studies in computational intelligence*, vol 51. Springer, New York, 2007.

195. Deb, K. An evolutionary multi-objective adaptive metamodeling procedure using artificial neural networks / K. Deb, P. Nain : *Studies in computational intelligence*, vol 51. Springer, New York, 2007.

196. Ling, Q. Deterministic robust optimal design based on standard crowding genetic algorithm / Q. Ling, G. Wu, Q. Wang : *Studies in computational intelligence*, vol 51. Springer, New York, 2007.

197. Ursem, R.K. Multinational GAs: multimodal optimization techniques in dynamic environments / In Proceedings of the genetic and evolutionary computation conference. Morgan Kaufmann, Massachusetts, 2000.

198. Arnold, D.V. Optimum tracking with evolution strategies / D.V. Arnold, H.G. Beyer : *Evol Comput* 14(3), 2006.

199. Boumaza, A. Learning environment dynamics from selfadaptation: a preliminary investigation / In Proceedings of the 2005 workshops of the genetic and evolutionary computation conference. ACM, New York, 2005.

200. Rohlfshagen, P., Lehre, P.K., Yao, X. Dynamic evolutionary optimisation: an analysis of frequency and magnitude of change / P. Rohlfshagen, P.K. Lehre, X. Yao : In Proceedings of the genetic and evolutionary computation conference, 2009.

201. Rossi, C. Tracking moving optima using Kalman-based predictions / C. Rossi, M. Abderrahim, J. Cesar Diaz : *Evol Comput* 16(1), 2008.

202. Li, C. A generalized approach to construct benchmark problems for dynamic optimization / C. Li, S. Yang : In *Simulated evolution and learning. Lecture notes in computer science*, vol 5361. Springer, Berlin, 2008.

203. Tinos, R. Evolutionary programming with q-Gaussian mutation for dynamic optimization problems / R. Tinos, S. Yang : In Proceedings of the IEEE Congress on evolutionary computation, 2008.

204. Blackwell, T. Particle swarm optimization in dynamic environments / *Studies in computational intelligence*, vol 51. Springer, New York, 2007.

205. Du, W. Multi-strategy ensemble particle swarm optimization for dynamic optimization / W. Du, B. Li : *Inf Sci* 178 (15), 2008.

206. Li, C. Fast multi-swarm optimization for dynamic optimization problems / C. Li, S. Yang : In *Fourth international conference on natural computation*, vol 7. IEEE Computer Society, 2008.

207. Ghosh, A. Univariate marginal distribution algorithms for non-stationary optimization problems / A. Ghosh, H. Muhlenbein : *Int J Knowl Intell Eng Syst* 8, 2004.

208. Bosman, P. Learning and anticipation in online dynamic optimization / *Studies in computational intelligence*, vol 51. Springer, New York, 2008.

209. Fernandes, C.M. UMDAs for dynamic optimization problems / C.M. Fernandes, C. Lima, A.C. Rosa : In Proceedings of the genetic and evolutionary computation conference. ACM, New York, 2008.

210. Olivetti de Franca, F. An artificial immune network for multimodal function optimization on dynamic environments / F. Olivetti de Franca, F., F.J. Von Zuben : In Proceedings of the genetic and evolutionary computation conference. ACM, New York, 2005.

211. Trojanowski, K. Immune-based algorithms for dynamic optimization / K. Trojanowski, S.T. Wierzchon : Inf Sci 179(10), 2009.

212. Montemanni, R. A new algorithm for a dynamic vehicle routing problem based on ant colony system / R. Montemanni, L. Gambardella : In Second international workshop on freight transportation and logistics, 2003.

213. Fernandes, C.M. Binary ant algorithm / C.M. Fernandes, A.C. Rosa, V. Ramos : In Proceedings of the genetic and evolutionary computation conference. ACM, New York, 2007.

214. Lung, R.I. A new collaborative evolutionary swarm optimization technique / R.I. Lung, D. Dumitrescu : In Proceedings of the genetic and evolutionary computation conference. ACM, New York, 2007.

215. Zeng, S. Orthogonal dynamic hill climbing algorithm: ODHC / S. Zeng, H. Shi, L. Kang : Studies in computational intelligence, vol 51. Springer, New York, 2007.

216. Cruz, C. Optimization in dynamic environments: a survey on problems, methods and measures / C. Cruz, J.R. González, D. Pelta : Soft Computing, 15 (7), Springer-Verlag, 2011.

217. Grefenstette, J. J. Genetic algorithms for changing environments / In Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels, 1992. – P. 137–144.

218. Vavak, F. Performance of a genetic algorithm with variable local search range relative to frequency of the environmental changes / F. Vavak, K.A. Jukes, T.C. Fogarty : In Proc. of the Third Int. Conf. on Genetic Programming, 1998. – P. 602–608.

219. Cobb, H.G. An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Non-stationary

Environments / Technical Report AIC-90-001, Naval Research Laboratory, Washington, USA, 1990.

220. Ursem, R.K. Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization / Ph.D. Dissertaion, University of Aarhus, Denmark, 2003.

221. Ng, K.P. A new diploid scheme and dominance change mechanism for non-stationary function optimization / K.P. Ng, K.C. Wong : Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, 1995.

222. Uyar, A.S. A new population based adaptive domination change mechanism for diploid genetic algorithms in dynamic environments / A.S. Uyar, A.E. Harmanci : Soft Computing - A Fusion of Foundations, Methodologies and Applications 9 (11), 2005.

223. Yang, S. On the design of diploid genetic algorithms for problem optimization in dynamic environments / IEEE Congress on Evolutionary Computation, CEC, 2006.

224. Yang, S. Population-based incremental learning with associative memory for dynamic environments / S. Yang, X. Yao : IEEE Transactions on Evolutionary Computation, 12 (5), 2008.

225. Simoes, A. Memory-based CHC algorithms for the dynamic traveling salesman problem / A. Simoes, E. Costa : Genetic and Evolutionary Computation Conference, GECCO, ACM, NewYork, USA, 2011

226. Daneshyari, M. Dynamic optimization using cultural based PSO / M. Daneshyari, G. Yen : IEEE Congress on Evolutionary Computation, CEC, 2011.

227. Cobb, H.G. Genetic algorithms for tracking changing environments / H.G. Cobb, J.J. Grefenstette : In Proc. of the Fifth Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1993.

228. Grefenstette, J.J. Evolvability in dynamic fitness landscapes: A genetic algorithm approach / In IEEE Congress on Evolutionary Computation, 1999.

229. Liles, W. The usefulness of tag bits in changing environments / W. Liles, K.A. De Jong : In IEEE Congress on Evolutionary Computation, 1999.

230. Sarma, J. The behavior of spatially distributed evolutionary algorithms in non-stationary environments / J. Sarma, K. De Jong : In Proc. Of the Genetic and Evolutionary Computation Conf. GECCO-99, 1999.
231. Kirley, M. An empirical investigation of optimisation in dynamic environments using the cellular genetic algorithm / M. Kirley, D.G. Green : In Proc. of the genetic and evolutionary computation conf. (GECCO-2000), 2000.
232. Oppacher, F. The shifting balance genetic algorithm: improving the GA in a dynamic environment / F. Oppacher, M. Wineberg : In Genetic and Evolutionary Computation Conference, GECCO, 1999.
233. Cheng, H. Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks / H. Cheng, S. Yang : In Applications of Evolutionary Computation, Lecture Notes in Computer Science, vol. 6024, Springer, Berlin, 2010.
234. Nguyen, T.T. Benchmarking and solving dynamic constrained problems / T.T. Nguyen, X. Yao : In IEEE Congress on Evolutionary Computation, CEC, IEEE Press, 2009.
235. Goh, C.-K. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization / C.-K. Goh, K.C. Tan : IEEE Transactions on Evolutionary Computation, 13 (1), 2009.
236. Tsutsui, S. Forking genetic algorithms: GAs with search space division schemes / S. Tsutsui, Y. Fujimoto, A. Ghosh : Evolutionary Computation, 5 (1), 1997.
237. Li, C. A clustering particle swarm optimizer for dynamic optimization / C. Li, S. Yang : In IEEE Congress on Evolutionary Computation, CEC, IEEE Press, 2009.
238. Woldesenbet, Y.G. Dynamic evolutionary algorithm with variable relocation / Y.G. Woldesenbet, G.G. Yen : IEEE Transactions on Evolutionary Computation, 13 (3), 2009.
239. Smith J.E. Self Adaptation in Evolutionary Algorithms / Ph.D. thesis, University of the West of England, Bristol, 1998
240. Whitley, D. GENITOR: A different Genetic Algorithm / D. Whitley, J. Kauth : in Proc. of the Rocky Mountain Conf. on Artificial Intelligence, Denver, 1988.

241. Vavak, F. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments / F. Vavak, T.C. Fogarty : In Proc. of the Society for the Study of Artificial Intelligence and Simulation of Behaviour Workshop on Evolutionary Computing, 1996.

242. Smith, J.E. Replacement strategies in steady state genetic algorithms: dynamic environments / J.E. Smith, F. Vavak : Journal of Computing and Information Technology, 7 (1), 1999.

243. Dozier, G. Steady-state evolutionary path planning, adaptive replacement, and hyper-diversity / In Parallel problem solving from nature – PPSN VI. Berlin: Springer, 2000.

244. Stroud, P.D. Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations / IEEE Transactions on Evolutionary Computation, 5 (1), 2001.

245. Hatzakis, I. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach / I. Hatzakis, D. Wallace : In Genetic and Evolutionary Computation Conference, GECCO, ACM Press, New York, NY, USA, 2006.

246. Simoes, A. Evolutionary algorithms for dynamic environments: prediction using linear regression and Markov chains / A. Simoes, E. Costa : In International Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science, vol.5199, Springer, 2008.

247. Simoes, A. Improving prediction in evolutionary algorithms for dynamic environments / A. Simoes, E. Costa : In Genetic and Evolutionary Computation Conference, GECCO, ACM, Montreal, Quebec, Canada, 2009.

248. Nguyen, T.T. Dynamic time-linkage problem revisited / T.T. Nguyen, X. Yao : In European Workshops on Applications of Evolutionary Computation, EvoWorkshops, Lecture Notes in Computer Science, vol. 5484, 2009.

249. Thomas, M. Cover Universal Portfolios / Mathematical Finance, volume 1, issue 1, 1991. – P. 1-29.

250. Gomes, C. Algorithm portfolios / C. Gomes, B. Selman : Artificial Intelligence, Volume 126, Issues 1–2, 2001. – P. 43-62.

251. Baudiš P. Online Black-Box Algorithm Portfolios for Continuous Optimization / P. Baudiš, P. Pošík : In: Bartz-Beielstein T., Branke J., Filipič B., Smith J. (eds) Parallel Problem Solving from Nature – PPSN XIII. PPSN 2014. Lecture Notes in Computer Science, vol 8672, 2014. – P. 40-49.

252. Niehaus, J. Adaption of Operator Probabilities in Genetic Programming / J. Niehaus, W. Banzhaf : In: Miller J., Tomassini M., Lanzi P.L., Ryan C., Tettamanzi A.G.B., Langdon W.B. (eds) Genetic Programming. EuroGP 2001. Lecture Notes in Computer Science, vol 2038, 2001. – P. 325-336.

253. Du, W. Multi-strategy ensemble particle swarm optimization for dynamic optimization / W. Du, B. Li : Information Sciences, 178 (15), 2008. – P. 3096–3109.

254. Branke, J. Memory enhanced evolutionary algorithms for changing optimization problems / Proceedings of the 1999 Congress on Evolutionary Computation CEC'99, 1999. – P. 1875-1882.

255. Li, C. Benchmark Generator for CEC2009 Competition on Dynamic Optimization / C. Li, S. Yang, T.T. Nguyen, E.L. Yu, X. Yao, X., et al. : Technical Report 2008, Department of Computer Science, University of Leicester, U.K, 2008.

256. The problems, Technical Report, Codes for Benchmark Generator for CEC'2009 Competition on Dynamic Optimization / Текст : электронный. – URL: https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC-09-Dynamic-Opt/CEC09-Dyn-Opt.htm (дата обращения: 25.05.2021).

257. Brest, J. Dynamic optimization using self-adaptive differential evolution / J. Brest, A. Zamuda, B. Boskovic, M.S. Maucec, V. Zumer : In: Proc. Of IEEE Congr. Evol. Comput., 2009.

258. Li, C. A clustering particle swarm optimizer for dynamic optimization / C. Li, S. Yang : In Proc. Of the Congr. on Evol. Comput., 2009.

259. Yu, E.L. Evolutionary programming with ensemble of external memories for dynamic optimization / E.L. Yu, P.N. Suganthan : In: Proc. Of IEEE Congr. Evol. Comput., 2009.

260. Постановление Правительства РФ от 11.03.2010 N 138 (ред. от 13.06.2018) «Об утверждении Федеральных правил использования воздушного пространства Российской Федерации».

261. Биард, Р.У. Малые беспилотные летательные аппараты: теория и практика / У.Б. Рэндал, Т.У. МакЛэйн. – Москва: ТЕХНОСФЕРА, 2015. – 312 с.

262. Handbook of Unmanned Aerial Vehicles / Editors K. Valavanis, George J. Vachtsevanos / Springer Science+Business Media Dordrecht, 2015 – 3022 p.

263. Chen, S. Neural networks for nonlinear dynamic system modeling and identification / S. Chen, S.A. Billings : Int. J. Contr., vol. 56, no. 2, 1992. – P. 319–346.

264. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – Пер.с польск. И.Д. Рудинского : Издательство «Горячая линия-Телеком», 2013. – 384 с.

265. Хайкин, С. Нейронные сети: полный курс, 2-е издание. : пер. с англ. / М. : Издательский дом «Вильямс», 2006. – 1104 с.

266. Puttige, V.R. Comparison of Real-time Online and Offline Neural Network Models for a UAV / V.R. Puttige, S.G. Anavatti : In 2007 International Joint Conference on Neural Networks, Orlando, FL, 2007. – P. 412-417.

267. Isaacs Development of a memetic algorithm for Dynamic Multi-Objective Optimization and its applications for online neural network modeling of UAVs / Isaacs, V. Puttige, T. Ray, W. Smith, S. Anavatti : In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008. – P. 548-554.

268. Динамика полета: Учебник для студентов высших учебных заведений / А.В. Ефремов, В.Ф. Захарченко, В.Н. Овчаренко и др.; под ред. Г.С. Бюшгенса. – М.: Машиностроение, 2011. – 776 с

269. Егорчев, М.В. Адаптивное нейросетевое моделирование динамических систем / М.В. Егорчев, Ю.В. Тюменцев. – Том 12, № 3-1 : Современные информационные технологии и ИТ-образование, 2016. – 195-201 с.

270. Omkar, S.N. Identification of Helicopter Dynamics based on Flight Data using Nature Inspired Techniques / S.N. Omkar, M. Dheevatsa, J. Senthilnath, M.V. Kumar : In *Int. J. Appl. Metaheuristic Comput.* 6 (3), 2015. – P. 38-52.

271. De Jong, K.A. An analysis of the behavior of a class of genetic adaptive systems / Ph.D. Dissertation, Univ. Michigan, Ann Arbor, MI, 1975.

272. Harik, G. Finding multi-modal solutions using restricted tournament selection / In: *Proceedings of the Sixth International Conference on Genetic Algorithms, ICGA-95*, 1997. – P. 24–31.

273. Wong, K.-C. Evolutionary multimodal optimization using the principle of locality / K.-C. Wong, C.-H. Wu, R.K.P. Mok, C. Peng, Z. Zhang : *Information Sciences*, 194, 2012. – P. 138–170.

274. Das, S. Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art / S. Das, S. Maity, B.Y. Qu, P.N. Suganthan : *Swarm and Evolutionary Computation*, 1 (2), 2011. – P. 71–88.

275. Li, X. Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications / X. Li, G. Epitropakis, K. Deb, A. Engelbrecht : *IEEE Transactions on Evolutionary Computation* Volume: 21 , Issue: 4, 2016. – P. 518 - 538.

276. Preuss, M. *Multimodal Optimization by Means of Evolutionary Algorithms* / Natural Computing Series, Springer International Publishing, 2015. – 189 p.

277. Deb, K. Finding Multiple Solutions for Multimodal Optimization Problems Using a Multi-Objective Evolutionary Approach / K. Deb, A. Saha : In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010*, 2010. – P. 447-454.

278. Liu, Y. A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization / Y. Liu, X. Ling, Zh. Shi, M. Lv, J. Fang, L. Zhang : *Journal of Software*, Vol. 6, No. 12, 2011. – P. 2449-2455.

279. Yang, Q. Adaptive Multimodal Continuous Ant Colony Optimization / In *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, 2017. – P. 191-205.

280. Wierzchoń, S.T. Multimodal Optimization with Artificial Immune Systems / In: *Kłopotek M.A., Michalewicz M., Wierzchoń S.T. (eds) Intelligent Information*

Systems 2001. *Advances in Intelligent and Soft Computing*, vol 10. Physica, Heidelberg
Intelligent Information Systems, 2001. – P. 167-178.

281. Xu, X. An Improved Immune Evolutionary Algorithm for Multimodal Function Optimization / X. Xu, J. Zhang : *Third International Conference on Natural Computation (ICNC 2007)*, Haikou, 2007. – P. 641-646.

282. Yin, X. A fast genetic algorithm with sharing scheme using cluster analysis methods in multi-modal function optimization / X. Yin, N. Gernay : *In Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, 1993. – P. 450–457.

283. Ballester, P.J. An Algorithm to Identify Clusters of Solutions in Multimodal Optimisation / P.J. Ballester, J.N. Carter : *In: Ribeiro C.C., Martins S.L. (eds) Experimental and Efficient Algorithms. WEA 2004. Lecture Notes in Computer Science*, vol 3059, 2004. – P. 42-56.

284. Deb, K. "Multimodal Optimization Using a Bi-Objective Evolutionary Algorithm / K. Deb, A. Saha : *In Evolutionary Computation*, vol. 20, no. 1, 2012. – P. 27-62.

285. Stoean, C. Multimodal Optimization by Means of a Topological Species Conservation Algorithm / C. Stoean, M. Preuss, R. Stoean, D. Dumitrescu : *Evolutionary Computation*, *IEEE Transactions on*. 14, 2011. – P. 842-864.

286. Li, L. History-Based Topological Speciation for Multimodal Optimization / L. Li, K. Tang : *In IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, 2015. – P. 136-150.

287. Zhang, G. A Memetic Algorithm for Global Optimization of Multimodal Nonseparable Problems / G. Zhang, Y. Li : *In IEEE Transactions on Cybernetics*, vol. 46, no. 6, 2016. – P. 1375-1387.

288. Wang, X. A multilevel sampling strategy based memetic differential evolution for multimodal optimization / X. Wang, M. Sheng, K. Ye, J. Lin, J. Mao, Sh.Chena, Ww Sheng : *Neurocomputing*, Volume 334, 21, 2019. – P. 79-88.

289. Ursem, R.K. Multinational GAs: multimodal optimization techniques in dynamic environments / In Proceedings of the Second Genetic and Evolutionary Computation Conference, GECCO, Morgan Kaufmann, 2000.

290. Mahfoud, S. Niching method for genetic algorithms / Doctoral Dissertation, Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Illinois Genetic Algorithms Laboratory, IlliGAL, Report No. 95001, 1995. – 251 p.

291. Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning / Addison-Wesley, New York, 1989. – 432 p.

292. Goldberg, D.E. Genetic algorithms with sharing for multimodal function optimization / D.E. Goldberg, J. Richardson : In: Proceedings of the second International Conference on Genetic Algorithms, 1987. – P. 41–49.

293. Pétrowski, A clearing procedure as a niching method for genetic algorithms / In Proceedings of Third IEEE International Conference on Evolutionary Computation, ICEC'96, IEEE Press, Piscataway, NJ, 1996. – P. 798–803.

294. Mahfoud, S. Crowding and preselection revisited / In Parallel Problem Solving from Nature, vol. 2, 1992. – P. 27–37.

295. Mengsheel, O. Probabilistic crowding: deterministic crowding with probabilistic replacement / O. Mengsheel, D. Goldberg : In Proceedings of the Genetic and Evolutionary Computation Conference, 1999, GECCO 99, 1999. – P. 409-416.

296. Yin, X. A fast genetic algorithm with sharing scheme using cluster analysis methods in multi-modal function optimization / X. Yin, N. Gerny : In Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, 1993. – P. 450–457.

297. MacQueen, J. Some methods for classification and analysis of multivariate observations / Proc. 5th Berkeley Symp. Math. Stat. Prob. 281, 1967.

298. Harik, G. Finding multi-modal solutions in problems of bounded difficulty / Technical Report, Illinois Genetic Algorithms Laboratory, Report No. 94002, 1994. – 51 p.

299. Singh, G. Comparison of multi-modal optimization algorithms based on evolutionary algorithms / G. Singh, K. Deb : In Proceedings of the Genetic and Evolutionary Computation Conference, Seattle, 2006. – P. 1305–1312.

300. Li, X. Results of the 2013 IEEE CEC Competition on Niching Methods for Multimodal Optimization / X. Li, A. Engelbrecht, M.G. Epitropakis : Report presented at 2013 IEEE Congress on Evolutionary Computation Competition on: Niching Methods for Multimodal Optimization. Текст : электронный – URL: <http://titan.csit.rmit.edu.au/~e46507/cec13-niching/competition/Niching-final-presentation-CEC2013.pdf> (дата обращения: 25.05.2021).

301. Preuss, M. Measuring multimodal optimization solution sets with a view to multiobjective techniques / M. Preuss, S. Wessing : EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV. AISC, vol. 227, Springer, Heidelberg, 2013. – P. 123–137.

302. Jarvis, R.A. Clustering Using a Similarity Measure Based on Shared Near Neighbors / R.A. Jarvis, E.A. Patrick : IEEE Trans. Comput. 22, 11, 1973. – P. 1025–1034.

303. Preuss, M. Niching foundations: basin identification on fixed-property generated landscapes / M. Preuss, C. Stoean, R. Stoean : Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011, 2011. – P. 837–844.

304. Yu, E.L. Ensemble of niching algorithms / E.L. Yu, P.N. Suganthan : Information Sciences, 180 (15), 2010. – P. 2815–2833.

305. Bandaru, S. A parameterless-niching-assisted bi-objective approach to multimodal optimization / S. Bandaru, K. Deb : In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13), 2013. – P. 95-102.

306. Preuss, M. Niching the CMA-ES via nearest-better clustering / In Proceedings of the 12th annual conference companion on Genetic and evolutionary computation (GECCO '10). ACM, New York, NY, USA, 2010. – P. 1711-1718.

307. Epitropakis, M.G. A dynamic archive niching differential evolution algorithm for multimodal optimization / M.G. Epitropakis, X. Li, E.K. Burke : In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13), 2013. – P. 79-86.

308. Molina, D. Variable mesh optimization for the 2013 CEC special session niching methods for multimodal optimization / D. Molina, A. Puris, R. Bello, F. Herrera : In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13), 2013. – P. 87-94.

309. Epitropakis, M.G. Finding multiple global optima exploiting differential evolution's niching capability / M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis : In 2011 IEEE Symposium on Differential Evolution (SDE), 2011. – P. 1-8.

310. Thomsen, R. Multimodal optimization using crowding-based differential evolution / In the IEEE Congress on Evolutionary Computation, 2004. CEC2004, vol.2, 2004. – P. 1382-1389.

311. Hansen, N. Completely Derandomized Self-Adaptation in Evolution Strategies / N. Hansen, A. Ostermeier : Evolutionary Computation, 9 (2), 2001. – P. 159-195.

312. Пуртиков, В.А. Оптимизация управления формированием кредитного портфеля банка / Дисс. канд. техн. наук. – Красноярск: САА, 2001. – 148 с.

313. Семенкин, Е.С. Модели и алгоритмы распределения общих ресурсов при управлении инновациями реструктурированного машиностроительного предприятия / Е.С. Семенкин, В.М. Клешков – Проблемы машиностроения и автоматизации, №3, 2006. – С. 24-30.

314. Семенкин, Е.С. Система поддержки принятия решений при управлении инновациями реструктурированного машиностроительного предприятия / Е.С. Семенкин, А.А. Шабалов, В.М. Клешков. – Вестник сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева, 5 (38), 2011. – С. 207-211.

315. Ruspini, E.H. Handbook of fuzzy computation / E.H. Ruspini, P.P. Bonissone, W. Pedrycz : Institute of Physics Pub, 1998. – 1098 p.

316. Круглов, В.В. Нечеткая логика и искусственные нейронные сети / В.В. Круглов, М.И. Дли, Р.Ю. Голунов. – М.: Физматлит, 2001. – 201 с.
317. Ishibuchi, H. Hybridization of fuzzy GBML approaches for pattern classification problems / In IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, Volume 35, Issue 2, 2005. – P. 359-365.
318. Booker L.B. Classifier Systems and Genetic Algorithms / L.B. Booker, D.E. Goldberg, J.H. Holland : AI 40, 1989. – P. 235-282.
319. Carse, B. Evolving fuzzy rule based controllers using genetic algorithms / B. Carse, T.C. Fogarty, A. Munro : Fuzzy Sets and Systems 80, 1996. – P. 273-293.
320. Herrera, F. Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases / F. Herrera, F. Hoffmann, L. Magdalena, O. Cordon, O. Cordon : World Scientific Publishing Company, 2011. – 489 p.
321. Dua, D. UCI Machine Learning Repository / D. Dua, C. Graff : Irvine, CA: University of California, School of Information and Computer Science. Текст : электронный – URL: <http://archive.ics.uci.edu/ml> (дата обращения: 25.05.2021).
322. Ishibuchi, H. Rule weight specification in fuzzy rule-based classification systems / H. Ishibuchi, T. Yamamoto : IEEE Transactions on Fuzzy Systems, 13 (4), 2005. – P. 428–435.
323. Mahdavi, S., Shiri, M. E., & Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: A survey. Information Sciences, 295, 407–428.
324. Sastry, K. Towards billion-bit optimization via a parallel estimation of distribution algorithm / K. Sastry, D.E. Goldberg, X. Llorca : In Genetic and Evolutionary Computation Conference. ACM, 2007. – P. 577–584.
325. Garret, V. Very Large Scale Optimization / V. Garret, C.T. James : NASA report, 2002. – 55 p.
326. Щедрин, Н.И. Математические методы программирования в экономике / Н.И. Щедрин, А.Н. Кархов. – Москва: Статистика, 1974. – 144 с.
327. Колмогоров, А.Н. Элементы теории функций и функционального анализа / А.Н. Колмогоров, С.В. Фомин. – 7-е изд. М.: ФИЗМАТЛИТ, 2009. – 572 с.

328. Kazimipour, B. Why advanced population initialization techniques perform poorly in high dimension? / B. Kazimipour, X. Li, A.K. Qin : In SEAL, 2014. – P. 479–490.
329. Segredo, E. On the comparison of initialisation strategies in differential evolution for large scale optimization / In Optimization Letters, 2017. – P. 1–14.
330. Mohamed, A.W. Differential Evolution with Novel Mutation and Adaptive Crossover Strategies for Solving Large Scale Global Optimization Problems / A.W. Mohamed, A.S. Almazyad : In Applied Computational Intelligence and Soft Computing 2017, 2017.
331. Yang, Zh. Target shape design optimization by evolving B-splines with cooperative coevolution / In Applied Soft Computing 48, 2016. – P. 672–682.
332. Molina, D. Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains / In Soft Computing 15 (11), 2011. – P. 2201–2220.
333. LaTorre, A. A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test / A. LaTorre, S. Muelas : In Soft Computing 15.11 (2011), pp. 2187–2199.
334. Vitorino, L.N. A hybrid swarm intelligence optimizer based on particles and artificial bees for high-dimensional search spaces / L.N. Vitorino, S.F. Ribeiro, C.J.A. Bastos-Filho : In IEEE Congress on Evolutionary Computation, IEEE, 2012. – P. 1–6.
335. Bolufe-Rohler, A. A minimum population search hybrid for large scale global optimization / A. Bolufe-Rohler, S. Fiol-Gonzalez, S. Chen : In IEEE Congress on Evolutionary Computation, IEEE, 2015. – P. 1958–1965.
336. Yang, Zh. Large scale evolutionary optimization using cooperative coevolution / Zh. Yang, K. Tang, X. Yao : Inform. Sci. 178 (15), 2008. – P. 2985–2999.
337. Potter, M.A. A cooperative coevolutionary approach to function optimization / M.A. Potter, K.A. De Jong : In Parallel Problem Solving from Nature 1040 PPSN III, Springer, 1994. – P. 249–257.
338. Potter, M.A. Cooperative coevolution: an architecture for evolving coadapted subcomponents / M.A. Potter, K.A. De Jong : Evol. Comput. 8 (1), 2000. – P. 1–29.

339. Liu, Y. Scaling up fast evolutionary programming with cooperative coevolution / Y. Liu, X. Yao, Q. Zhao, T. Higuchi : Proceedings of the 2001 Congress on Evolutionary Computation, 2001, vol. 2, IEEE, 2001. – P. 1101–1108.

340. Van den Bergh, F. A cooperative approach to particle swarm optimization / F. Van den Bergh, A.P. Engelbrecht : IEEE Trans. Evol. Comput. 8 (3), 2004. – P. 225–239.

341. Shi, Y. Cooperative co-evolutionary differential evolution for function optimization / Y. Shi, H. Teng, Z. Li : In Advances in Natural Computation, Springer, 2005. – P. 1080–1088.

342. Omidvar, M.N. Cooperative co-evolution with differential grouping for large scale optimization / M.N. Omidvar, X. Li, Y. Mei, X. Yao : IEEE Trans. 1024 Evol. Comput. 18 (3), 2014. – P. 378–393.

343. Omidvar, M.N. Cooperative Co-evolution with delta grouping for large scale non-separable function optimization / M.N. Omidvar, X. Li, X. Yao : IEEE Congress on Evolutionary Computation, 2010.

344. Сопов, Е.А. Эволюционные алгоритмы моделирования и оптимизации сложных систем / Диссертация кандидата технических наук. – Красноярск, 2004.

345. Ворожейкин, А.Ю. Адаптивные эволюционные алгоритмы решения сложных задач оптимизации / Диссертация кандидата технических наук. – Красноярск, 2008.

346. Галушин, П.В. Асимптотический вероятностный генетический алгоритм решения сложных задач глобальной оптимизации / Диссертация кандидата технических наук. – Красноярск, 2012.

347. 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) Volume 9712 LNCS, 2016. – P. 619-626.

348. Sopov, E. Adaptive variable-size random grouping for evolutionary large-scale global optimization / Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10941 LNCS, 2018. – P. 583-592.

349. Whitley, D. The Island Model Genetic Algorithm: On Separability, Population Size and Convergence / D. Whitley, S.B. Rana, R.B. Heckendorn : CIT 2015, 2015. – 17 p.

350. Akhmedova, Sh. Co-Operation of Biology Related Algorithms / Sh. Akhmedova, E. Semenkin : In 2013 IEEE Congress on Evolutionary Computation (CEC'2013, Mexico), 2013. – P. 2207-2214.

351. Sopov, E. An investigation of parameter tuning in the random adaptive grouping algorithm for LSGO problems / E. Sopov, A. Vakhnin : In IJCCI 2018 - Proceedings of the 10th International Joint Conference on Computational Intelligence, 2018. – P. 255-263.

352. Yang, Zh. Self-adaptive differential evolution with neighborhood search / Zh. Yang, K. Tang, X. Yao : In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, 2008. – P. 1110-1116.

353. Liu, J. Scaling Up Covariance Matrix Adaptation Evolution Strategy Using Cooperative Coevolution / J. Liu, K. Tang : Lecture Notes in Computer Science, 2013. – P. 350–357.

354. López, E.D. Vmode: A hybrid metaheuristic for the solution of large scale optimization problems / E.D. López, A. Puris, R.R. Bello : Investigacion Operacional 36(3), 2015. – P. 232-239.

355. Wei, F. Smoothing and auxiliary functions based cooperative coevolution for global optimization / F. Wei, Y. Wang, H. Yuanliang : In 2013 IEEE Congress on Evolutionary Computation, 2013. – P. 2736-2741.

356. Muelas, L.S. Multiple Offspring Sampling in Large Scale Global Optimization / L.S. Muelas, J. Peña : In 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, 2012. – P. 1-8.

357. Molina, D. Iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization. / D. Molina, F. Herrera : In 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, 2015. – P. 1974-1978.

358. Miranda, V. Differential Evolutionary Particle Swarm Optimization (DEEPSO): A Successful Hybrid / V. Miranda, R. Alves : In 2013 BRICS Congress on

Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, 2013. – P. 368-374.

359. Li, X. 2013 IEEE Congress on Evolutionary Computation Competition on Large Scale Global Optimization / X. Li, M.N. Omidvar, K.Q.K. Tang, Zh. Yang. Текст : электронный – URL: <https://titan.csit.rmit.edu.au/~e46507/cec13-lsgo/competition/lsgo-competition-sumary-2013.pdf> (дата обращения: 25.05.2021).

360. Li, X. 2015 IEEE Congress on Evolutionary Computation Competition on: Large Scale Global Optimization / X. Li, K. Tang, Zh. Yang, D. Molina. Текст : электронный – URL: <http://titan.csit.rmit.edu.au/~e46507/lsgo-competition-cec15/lsgo-competition-summary-2015.pdf> (дата обращения: 25.05.2021).

361. Куликов, А.Л. Проблемы и особенности распределённой электроэнергетики / А.Л. Куликов, В.Л. Осокин, Б.В. Папков. – Вестник НГИЭИ, №11 (90), 2018. – С. 123-136.

362. Хохлов, А. Распределенная энергетика в России: потенциал развития / А. Хохлов, Ю. Мельников, Ф. Веселов, Д. Холкин, К. Дацко. – Энергетический центр Московской школы управления СКОЛКОВО, 2018. Текст : электронный – URL: https://energy.skolkovo.ru/downloads/documents/SEneC/Research/SKOLKOVO_EneC_DER-3.0_2018.02.01.pdf (дата обращения: 25.05.2021).

363. Bisen, D. Solution of Large Scale Economic Load Dispatch Problem using Quadratic Programming and GAMS: A Comparative Analysis / D. Bisen, H.M. Dubey, M. Pandit, B.K. Panigrahi : Journal of Information and Computing Science Vol. 7, No. 3, 2012. – P. 200-211.

364. Nagendra, S. Economic Load Dispatch with Valve Point Loading Effect and Generator Ramp Rate Limits Constraint using MRPSO / S. Nagendra, K. Yogendra : International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, Issue 4, 2013. – P. 1472-1477.

365. Vlachogiannis, J. Economic Load Dispatch—A Comparative Study on Heuristic Optimization Techniques With an Improved Coordinated Aggregation-Based PSO / J. Vlachogiannis, K.Y. Lee : IEEE Transactions on Power Systems, vol. 24, no. 2, 2009. – P. 991-1001.

366. Alam, M.N. State-of-the-Art Economic Load Dispatch of Power Systems Using Particle Swarm Optimization / arXiv:1812.11610 [Preprint], 2018. Текст : электронный – URL: <https://arxiv.org/abs/1812.11610> (дата обращения: 25.05.2021).

367. Suganthan, P.N. Testing Evolutionary Algorithms on Real-World Numerical Optimization Problems / Technical report. Competition on Testing Evolutionary Algorithms on Real-world Numerical Optimization Problems, IEEE CEC 2011, USA, 2011. Текст : электронный – URL: https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC2011_ranking.pdf (дата обращения: 25.05.2021).

368. Das, S. Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems / S. Das, P. N. Suganthan : Technical Report, Jadavpur University, India and Nanyang Technological University, Singapore, 2010. Текст : электронный – URL: <https://pdfs.semanticscholar.org/d2f5/46248edd0c66d833c3e5f67e094e6922d262.pdf> (дата обращения: 25.05.2021).

369. Elsayed, S.M. GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems / S.M. Elsayed, R.A. Sarker, D.L. Essam : 2011 IEEE Congress of Evolutionary Computation (CEC), 2011. – P. 1034-1040.

370. Ширококов, М.Г. Проектирование межпланетных перелетов с несколькими гравитационными маневрами и промежуточными импульсами / М.Г. Ширококов, С.П. Трофимов. – М., РАН, 2017.

371. Овчинников, М.Ю. Метод виртуальных траекторий для проектирования межпланетных миссий с гравитационными маневрами / М.Ю. Овчинников, С.П. Трофимов, М.Г. Ширококову – Космические исследования, том: 51, номер: 6, 2013. – 484 с.

372. Addis, B. Global Optimization for the Design of Space Trajectories / B. Addis, A. Cassioli, M. Locatelli, F. Schoen : Computational Optimization and Applications, volume 48, 2011. – P. 635–652.

373. Hong, Ch. Constrained global optimization of low-thrust interplanetary trajectories / Ch. Hong, Y. David, D. Lorenzo, D. Izzo : Evolutionary Computation (IEEE CEC), 2010.

374. Morante, D. A Survey on Low-Thrust Trajectory Optimization Approaches / D. Morante, M.S. Rivo, M. Soler : Aerospace, 8 (88), 2021.

375. Shirazi, A. Evolutionary algorithms to optimize low-thrust trajectory design in spacecraft orbital precession mission / A. Shirazi, J. Ceberio, J.A. Lozano : 2017 IEEE Congress on Evolutionary Computation (CEC), 2017. – P. 1779-1786.

376. Pykep / Текст : электронный – URL: <https://esa.github.io/pykep/index.html> (дата обращения: 25.05.2021).

377. Yam, C.H. Low-thrust trajectory design as a constrained global optimization problem / C.H. Yam, D.D. Lorenzo, D. Izzo : Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 225 (11), 2011. – P. 243–1251.

378. Farmani, R. Self-adaptive fitness formulation for constrained optimization / R. Farmani, J.A. Wright : IEEE Transactions on Evolutionary Computation, 7 (5), 2003. – P. 445–455.

379. Wolpert, D.H. Coevolutionary Free Lunches / D.H. Wolpert, W.G. Macready : IEEE Transactions on Evolutionary Computation, 9 (6), 2005. – P. 721–735.

380. Сопов, Е.А. Новый метод группировки переменных для задач параметрической оптимизации большой размерности / Е.А. Сопов, А.В. Вахнин : Сибирский журнал науки и технологий. – 2018. – Т. 19. – № 3 – С. 386-395.

381. Сопов, Е.А. Обобщенный метод анализа мультимодальных данных на основе оптимизационного подхода / Е.А. Сопов, И.А. Иванов, К.Ю. Брестер / Сибирский журнал науки и технологий. – 2017. – Т. 18. – № 4. – С. 736–743.

382. Сопов, Е.А. Модель функционирования программной системы на основе gert-сети / Е.А. Сопов, Т.А. Панфилова, И.А. Панфилов, В.В. Золотарев, И.В. Ковалев : Сибирский журнал науки и технологий. – 2017. – Т. 18. – № 4. – С. 773–778.

383. Сопов, Е.А. Бинарный генетический алгоритм с декомпозицией на основе оценки распределения для задач глобальной оптимизации высокой

размерности. Вестник Сибирского государственного аэрокосмического университета. – 2016. – Т. 17. – № 4. – С. 899-906.

384. Сопов, Е.А. Гибридный алгоритм обучения конволюционной нейронной сети / Е.А. Сопов, И.А. Иванов : Вестник Сибирского государственного аэрокосмического университета. – 2016. – Т. 17. – № 4. – С. 871-877.

385. Sopot, E. Multiple Optima Identification Using Multi-strategy Multimodal Genetic Algorithm. Journal of Siberian Federal University. Mathematics & Physics. – 2016. – 9(2). – P. 246–257.

386. Сопов, Е.А. Повышение надежности процесса бурения путем оптимизации управляемых параметров режима бурения / Е.А. Сопов, А.А. Плющик, В.В. Бухтояров : Научно-технический вестник Поволжья. – 2016. – №3. – С. 40-43.

387. Сопов, Е.А. Самоконфигурируемый ансамбль генетических алгоритмов для решения задач мультимодальной оптимизации / Е.А. Сопов, И.А. Иванов, И.А. Панфилов : Вестник Сибирского государственного аэрокосмического университета. – 2015. – Т. 16. – № 4. – С. 833-841

388. Сопов, Е.А. Многокритериальный подход к проектированию ансамбля нейросетевых классификаторов с отбором информативных признаков для решения задачи распознавания эмоций / Е.А. Сопов, С.С. Аплеснин : Вестник Сибирского государственного аэрокосмического университета. – 2015. – Т. 16. – № 4. – С. 819-827.

389. Sopot, E.A. Self-configuring Multi-strategy Genetic Algorithm for Non-stationary Environments. Vestnik SibGAU. – 2015. – Vol. 16. – No. 1. – P. 124-130.

390. Сопов, Е.А. Исследование эффективности работы генетического алгоритма оптимизации с альтернативным представлением решений / Е.А. Сопов, И.А. Панфилов, Е.П. Базанова : Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2013. – Вып. 4(50). – С. 68-71.

391. Сопов, Е.А. Исследование эффективности самоконфигурируемого коэволюционного алгоритма решения сложных задач многокритериальной

оптимизации / Е.А. Сопов, И.А. Иванов : Москва-Воронеж, Системы управления и информационные технологии. – 2013. – № 1.1 (51). – С. 141-145.

392. Сопов, Е.А. Самоконфигурируемый генетический алгоритм решения задач поддержки многокритериального выбора / Е.А. Сопов, И.А. Иванов : Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2013. – Вып. 1(47). – С. 30-35.

393. Сопов, Е.А. Интеллектуальные информационные технологии выявления и диагностики проблем в задачах анализа сложных систем / Е.А. Сопов, С.А. Сопов : Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2011. – Вып. 4(37). – С. 84-89.

394. Сопов, Е.А. Вероятностный генетический алгоритм решения сложных задач многокритериальной оптимизации с адаптивной мутацией и прогнозом множества Парето / Е.А. Сопов, С.А. Сопов : Вестник Самарского государственного аэрокосмического университета имени академика С.П. Королева. – 2011. – 6 (30). – С. 273-282.

395. Sopov, E.A. A Modified probabilistic genetic algorithm for the solution of complex constrained optimization problems / Vorozheikin A.Yu., Gonchar T.N., Panfilov I.A., Sopov E.A., Sopov S.A. : Vestnik of Siberian State Aerospace University named after academician M. F. Reshetnev. – 2009. – 5(26). – P. 31-36.

396. Сопов, Е.А. Об одной модификации вероятностного генетического алгоритма для решения сложных задач условной оптимизации / Ворожейкин А.Ю., Гончар Т.Н., Панфилов И.А., Сопов Е.А., Сопов С.А. : Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2009. – Вып. 4(25). – С. 79-84.

397. Sopov, E. Investigation of improved cooperative coevolution for large-scale global optimization problems / E. Sopov, A. Vakhnin : Algorithms. – 2021. – 14 (5). – № 146. – 19 p.

398. Sopov, E. Large-scale clustering using decomposition-based evolutionary algorithms / E. Sopov, A. Vakhnin : In 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020. – 2020. – P. 345-352.

399. Sopov, E. Investigation of the iCC framework performance for solving constrained LSGO problems / E. Sopov, A. Vakhnin : Algorithms. – 2020. – 13 (5). – № 108. – 18 p.

400. Sopov, E. Improving DE-based cooperative coevolution for constrained large-scale global optimization problems using an increasing grouping strategy / E. Sopov, A. Vakhnin : Conference Series: Materials Science and Engineering. – 2020. – 734 (1). – № 012099. – 11 p.

401. Sopov, E. Improved random adaptive grouping approach for solving unconstrained LSGO problems / E. Sopov, A. Vakhnin, I. Panfilov : Journal of Physics: Conference Series. – 2020. – 1515 (3). – № 032076. – 8 p.

402. Sopov, E. Investigating the effects of population size and the number of subcomponents on the performance of shade algorithm with random adaptive grouping for LSGO problems / E. Sopov, A. Vakhnin : GECCO 2019 Companion - Proceedings of the 2019 Genetic and Evolutionary Computation Conference Companion. – 2019. – P. 269-270.

403. Sopov, E. Online Selective Evolutionary Hyperheuristic for Large Scale Economic Load Dispatch Problem / E. Sopov, S. Videnin : Proceedings of the 2019 IEEE International Conference on Information Technologies (InfoTech-2019). – 2019. – 8860884. – 4 p.

404. Sopov, E. An approach for initializing the random adaptive grouping algorithm for solving large-scale global optimization problems / E. Sopov, A. Vakhnin : Conference Series: Materials Science and Engineering. – 2019. – 537(4). – 042006. – 11 p.

405. Sopov, E.A. A problem decomposition approach for large-scale global optimization problems / E.A. Sopov, A.V. Vakhnin, I.A. Panfilov., A.S. Polyakova, D.V. Kustov : Conference Series: Materials Science and Engineering. – 2019. – 537(5). – 052031. – 9 p.

406. Sopov, E. A decomposition-based approach for constrained large-scale global optimization. IJCCI 2019 / E. Sopov, A. Vakhnin : Proceedings of the 11th International Joint Conference on Computational Intelligence. – 2019. – P. 147-154.

407. Sopov, E. Genetic programming based synthesis of clustering algorithm for identifying batches of electronic components / E. Sopov, I. Panfilov : ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics. – 2019. – vol. 1. – P. 202-209.

408. Sopov, E. An approach for initializing the random adaptive grouping algorithm for solving large-scale global optimization problems / E. Sopov, A. Vakhnin : International Workshop Advanced Technologies in Material Science, Mechanical and Automation Engineering - MIP: Engineering. – 2019. – 11 p.

409. Sopov, E. A problem decomposition approach for large-scale global optimization problems / E.A. Sopov, A.V. Vakhnin, I.A. Panfilov., A.S. Polyakova, D.V. Kustov : International Workshop Advanced Technologies in Material Science, Mechanical and Automation Engineering - MIP: Engineering. – 2019. – 11 p.

410. Sopov, E. Comprehensive approach for solving multimodal data analysis problems based on integration of evolutionary, neural and deep neural network algorithms / E. Sopov, I. Ivanov, I. Panfilov : Conference Series-Materials Science and Engineering. – 2018. – vol. 450.

411. Sopov, E. An approach for automating the design of convolutional neural networks / E. Sopov, D. Plotnikov, I. Panfilov : Conference Series-Materials Science and Engineering. – 2018. – vol. 450.

412. Sopov E. A Novel Method for Grouping Variables in Cooperative Coevolution for Large-scale Global Optimization Problems / E. Sopov, A. Vakhnin : Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO. – 2018. – P. 261-268.

413. Sopov, E. On Tuning Group Sizes in the Random Adaptive Grouping Algorithm for Large-scale Global Optimization Problems / E. Sopov, A. Vakhnin, E. Semenkin : Proceedings of 2018 International Conference on Applied Mathematics and Computational Science, ICAMCS.NET. – 2018. – № 8955749. – P. 134-145.

414. Sopov, E. Hyper-heuristics for Structure and Parameters Tuning in Evolutionary Algorithms / E. Sopov, A. Pupkov, I. Panfilov, V. Samarin, N. Telesheva,

R. Kuzmich : 7th International Conference on Changes in Social and Business Environment (CISABE' 2018). – 2018. – P.95-102.

415. Sopov, E. Genetic Programming Hyper-heuristic for the Automated Synthesis of Selection Operators in Genetic Algorithms. IJCCI 2017 - Proceedings of the 9th International Joint Conference on Computational Intelligence. – 2017. – P. 231-238.

416. Sopov, E. Hybrid binary GA-EDA algorithms for complex "black-box" optimization problems. Conference Series: Materials Science and Engineering. – 2017. – 173 (1). – 012019. – P. 1-9.

417. Sopov, E. Self-configuring ensemble of multimodal genetic algorithms. Studies in Computational Intelligence. – 2017. – vol.669. – P. 56-74.

418. Sopov, E. EDA-based decomposition approach for binary LSGO problems. IJCCI 2016 - Proceedings of the 8th International Joint Conference on Computational Intelligence. – 2016. – Volume 1. – P. 132-139.

419. Sopov, E. A meta-heuristic for improving the performance of an evolutionary optimization algorithm applied to the dynamic system identification problem / E. Sopov, I. Ryzhikov, E. Semenkin : IJCCI 2016 - Proceedings of the 8th International Joint Conference on Computational Intelligence. – 2016. – Volume 1. – P. 178-185.

420. Sopov, E. Ensemble of multimodal genetic algorithms for design and decision making support problems / E. Sopov, E. Semenkin, I. Panfilov : ICINCO 2016 - Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics. – 2016. – Volume 1. – P. 160-167.

421. Sopov, E. A Self-configuring Multi-strategy Multimodal Genetic Algorithm. Advances in Nature and Biologically Inspired Computing Volume 419 of the series Advances in Intelligent Systems and Computing. – 2016. – P. 15-26.

422. Sopov, E. Self-configuring ensemble of neural network classifiers for emotion recognition in the intelligent human-machine interaction / E. Sopov, I. Ivanov : Proc. of 2015 IEEE Symposium Series on Computational Intelligence (SSCI 2015). – 2016. – P. 1808-1815.

423. Sopov, E. Multi-strategy multimodal genetic algorithm for designing fuzzy rule based classifiers / E. Sopov, V. Stanovov, E. Semenkin : Proc. of 2015 IEEE Symposium Series on Computational Intelligence (SSCI 2015). – 2016. – P. 167-173.

424. Sopov, E. Multi-strategy Genetic Algorithm for Self-Configuring Solving of Complex Optimization Problems. Proc. of the 4th International Congress on Advanced Applied Informatics, (IIAI-AAI 2015). – 2016. – P. 556-561.

425. Sopov, E. Feature and Decision Level Audio-visual Data Fusion in Emotion Recognition Problem / E. Sopov, M. Sidorov, I. Ivanov, W. Minker : The 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2015). – 2015. – P. 246 – 251.

426. Sopov, E. On the Application of Co-Operative Swarm Optimization in the Solution of Crystal Structures from X-Ray Diffraction Data / E. Sopov, A. Zaloga, S. Burakov, E. Semenkin, M. Semenkina : Advances in Swarm and Computational Intelligence, Lecture Notes in Computer Science. – 2015. – Volume 9140. – P. 89-96.

427. Sopov, E. The optimal solution prediction for genetic and distribution building algorithms with binary representation / E. Sopov, O. Semenkina : Conf. Ser.: Mater. Sci. Eng. – 2015. – vol. 70. – 10 p.

428. Sopov, E. A Self-configuring Metaheuristic for Control of Multi-Strategy Evolutionary Search. Advances in Swarm and Computational Intelligence. Lecture Notes in Computer Science. – 2015. – Volume 9142. – P. 29-37.

429. Sopov, E. Multi-strategy genetic algorithm for multimodal optimization. In: proc. of the 7th International Joint Conference on Computational Intelligence (IJCCI). – 2015. – Vol. 1. – P. 55-63.

430. Sopov, E. Design Efficient Technologies for Context Image Analysis in Dialog HCI Using Self-Configuring Novelty Search Genetic Algorithm / E. Sopov, I. Ivanov : In: proc. of the 11th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2014. – 2014. – P. 832-839.

431. Sopov, E. On the investigation of Coevolutionary Genetic Algorithm for Self-configuring Solving of Non-stationary Optimization Problems. Proc. International

Conference on Computer Science and Artificial Intelligence (ICCSAI 2014). –2015. – P. 202-207.

432. Sopov, E. Intrusion detectors design with self-configuring multi-objective genetic algorithm / E. Sopov, I. Panfilov : Network Security and Communication Engineering - Proceedings of the 2014 International Conference on Network Security and Communication Engineering (NSCE 2014). – 2015. – P. 185-188.

433. Sopov, E. Self-tuning SVM with Feature Selection for Text Categorization Problem / E. Sopov, I. Panfilov : Proc. International Conference on Computer Science and Artificial Intelligence (ICCSAI 2014). – 2015. – P. 208-211.

434. Сопов, Е.А. Программная система решения сложных задач нестационарной мультимодальной оптимизации на основе эволюционных самоконфигурируемых гиперэвристик / Е.А. Сопов, А.В. Вахнин, И.А. Панфилов, Е.Д. Агафонов, А.В. Гуменникова, А.А. Ступина, Т.А. Панфилова, О.А. Сопова, Е.А. Бежитская, Д.В. Плотников : Свидетельство о государственной регистрации программы для ЭВМ, № 2020611378. – 2020.

435. Сопов, Е.А. Программная реализация на основе параллельных вычислений метода DECC-RAG решения больших задач оптимизации / Е.А. Сопов, А.В. Вахнин, И.А. Панфилов : Свидетельство о государственной регистрации программы для ЭВМ, №2019613931. – 2019.

436. Сопов, Е.А. Программная система исследования эффективности эволюционных алгоритмов на задачах оптимизации большой размерности / Е.А. Сопов, А.В. Вахнин, И.А. Панфилов : Свидетельство о государственной регистрации программы для ЭВМ, №2018666438. – 2018.

437. Сопов, Е.А. Программная система автоматизированной обработки спектроскопических данных с использованием методов кластерного анализа и эволюционных алгоритмов нестационарной мультимодальной оптимизации / Е.А. Сопов, Ю.С. Ломаев, И.А. Панфилов : Свидетельство о государственной регистрации программы для ЭВМ, №2017662420. – 2017.

438. Сопов, Е.А. Программная система автоматизированного проектирования интеллектуальных информационных сетей / Е.С. Семенкин, И.А.

Панфилов, Е.А. Сопов, В.В. Становов, Д.И. Хритonenко, К.Ю. Брестер, О.Е. Семенкина : Свидетельство о государственной регистрации программы для ЭВМ, №2015662501. – 2015.

439. Сопов, Е.А. Программная система для автоматизированного управления интеллектуальными информационными сетями / Е.С. Семенкин, И.А. Панфилов, Е.А. Сопов, В.В. Становов, Д.И. Хритonenко, К.Ю. Брестер, О.Е. Семенкина : Свидетельство о государственной регистрации программы для ЭВМ, №2015662581. – 2015.

440. Сопов, Е.А. Система автоматического генерирования коллективов нейросетевых моделей распределенными самоконфигурируемыми эволюционными алгоритмами / Д.И. Хритonenко, И.А. Панфилов, Е.А. Сопов : Свидетельство о государственной регистрации программы для ЭВМ, №2014612727. – 2014.

441. Сопов, Е.А. Программная система выбора эффективных структур аппаратно-программных комплексов высокопроизводительных распределенных систем / Е.С. Семенкин, И.А. Панфилов, Е.А. Сопов : Свидетельство о государственной регистрации программы для ЭВМ, №2013619087. – 2013.

442. Сопов, Е.А. Программная система формирования эффективных конфигураций параллельных вычислительных систем (МВК, GRID и CLOUD) методами интеллектуальных информационных технологий / Е.С. Семенкин, И.А. Панфилов, Е.А. Сопов : Свидетельство о государственной регистрации программы для ЭВМ, №2013617236. – 2013.

443. Сопов, Е.А. Программная система решения сложных задач многокритериальной оптимизации самоконфигурируемым коэволюционным алгоритмом SelfCOMOGA / И.А. Иванов, Е.А. Сопов : Свидетельство о государственной регистрации программы для ЭВМ, №2013613773. – 2013.

444. Сопов, Е.А. Программная система «л-IT-on 2.0» - автоматизированное генерирование интеллектуальных технологий анализа данных / Е.С. Семенкин, И.А. Панфилов, Е.А. Сопов, А.А. Шабалов, Р.Б. Сергиенко : Свидетельство о государственной регистрации программы для ЭВМ, №2013617189. – 2013.

ПРИЛОЖЕНИЕ А
РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ ИССЛЕДОВАНИЯ
ЭФФЕКТИВНОСТИ МЕТОДА 1

Таблица А.1 – Результаты кросс-валидации для задач ВВОВ с использованием селекции (2.17)

Тестовый набор	Задача	Процент успешных запусков	Среднее по набору
1	3	77.50	84.00
	9	90.00	
	12	90.00	
	15	82.50	
	20	80.00	
2	1	95.00	85.00
	9	87.50	
	14	80.00	
	16	82.50	
	24	80.00	
3	5	92.50	84.00
	7	87.50	
	12	90.00	
	18	72.50	
	22	77.50	
4	4	92.50	83.50
	9	87.50	
	13	77.50	
	19	82.50	
	24	77.50	
5	2	92.50	84.50
	8	85.00	
	10	85.00	
	18	77.50	
	21	82.50	
Среднее по наборам			84.20

Таблица А.1 – Результаты численных экспериментов в виде процента успешных запуск ГА, когда глобальный оптимум найден с заданной точностью

Задача	Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование	Среднее по столбцам 2-4	Оператор синтезированный методом 1
1	100.0	82.5	92.5	91.7	95.0
2	97.5	95.0	90.0	94.2	92.5
3	80.0	75.0	75.0	76.7	77.5
4	77.5	82.5	80.0	80.0	92.5
5	50.0	67.5	72.5	63.3	90.0
6	57.5	72.5	80.0	70.0	92.5
7	62.5	65.0	62.5	63.3	87.5
8	80.0	85.0	80.0	81.7	85.0
9	75.0	80.0	82.5	79.2	90.0
10	77.5	77.5	72.5	75.8	85.0
11	80.0	82.5	85.0	82.5	82.5
12	77.5	87.5	90.0	85.0	90.0
13	57.5	67.5	75.0	66.7	77.5
14	70.0	85.0	82.5	79.2	80.0
15	65.0	85.0	80.0	76.7	82.5
16	77.5	85.0	85.0	82.5	82.5
17	50.0	72.5	77.5	66.7	80.0
18	42.5	80.0	85.0	69.2	77.5
19	70.0	82.5	80.0	77.5	82.5
20	60.0	82.5	75.0	72.5	80.0
21	27.5	52.5	82.5	54.2	77.5
22	40.0	72.5	67.5	60.0	80.0
23	37.5	60.0	62.5	53.3	77.5
24	32.5	42.5	47.5	40.8	75.0
Среднее	64.4	75.8	77.6	72.6	83.9

Таблица А.2 – Результаты численных экспериментов в виде среднего числа поколений, которые требуются чтобы найти глобальный оптимум с заданной точностью

Задача	Пропорциональная селекция	Линейное ранжирование	Экспоненциальное ранжирование	Среднее по столбцам 2-4	Оператор синтезированный методом 1		
					$\theta=60$	$\theta=70$	$\theta=80$
1	15	18	13	15.33	20	13	25
2	15	25	22	20.67	26	15	24
3	18	25	18	20.33	13	24	30
4	15	24	21	20.00	24	19	27
5	12	17	22	17.00	17	20	30
6	39	39	23	33.67	20	21	43
7	39	28	43	36.67	32	24	29
8	40	33	42	38.33	27	18	43
9	29	31	31	30.33	21	29	19
10	26	35	34	31.67	20	25	33
11	37	31	33	33.67	18	27	33
12	37	28	26	30.33	27	30	27
13	33	33	25	30.33	29	20	34
14	31	30	22	27.67	26	31	22
15	48	33	43	41.33	29	28	47
16	36	30	29	31.67	26	33	40
17	38	28	28	31.33	4	30	35
18	37	10	40	37.00	21	42	30
19	42	31	31	34.67	24	29	34
20	43	46	44	44.33	37	47	39
21	46	42	41	43.00	35	39	39
22	40	41	41	40.67	46	40	42
23	46	42	40	42.67	4	43	45
24	46	41	42	43.00	40	47	37
Среднее	33.7	30.8	31.4	32.3	24.4	28.9	33.6

ПРИЛОЖЕНИЕ Б
РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ ИССЛЕДОВАНИЯ
ЭФФЕКТИВНОСТИ МЕТОДА 2

Таблица Б.1 – Сравнение эффективности алгоритмов на функции 1 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	0.028	0.035	5.71	5.609	5.669	2.21
T2	3.587	2.718	10.65	10.08	10.24	3.78
T3	2.999	4.131	10.87	13.13	11.73	4.27
T4	0.015	0.094	1.503	21.22	21.89	1.07
T5	2.177	1.869	8.295	7.899	6.731	4.57
T6	1.145	1.056	13.12	29.25	32.01	2.03
T7	3.5017	4.54	13.123	12.45	12.8	3.705

Таблица Б.2 – Сравнение эффективности алгоритмов на функции 2 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	0.96	1.247	6.214	33.05	45.79	1.64
T2	43	10.1	7.223	182.9	186.9	10.02
T3	50.1	10.27	4.988	128.5	135.8	7.87
T4	0.79	0.566	4.206	32.85	53.57	3.52
T5	67	25.14	3.505	191.7	186.5	8
T6	3.36	3.651	3.478	43.25	73.34	3.55
T7	13.2524	3.651	6.7124	40.59	61.13	7.11

Таблица Б.3 – Сравнение эффективности алгоритмов на функции 3 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	11.39	137.5	151.98	158.1	553.6	150.6
T2	558.49	855.1	140.47	638.7	900.8	572.8
T3	572.1	765.9	136.67	573.9	827.1	512.61
T4	65.74	430.6	164.96	419.5	709	164.5

T5	475.76	859.7	95.12	741.9	829.1	610.3
T6	243.27	753	107.54	491.7	803.5	236.8
T7	153.673	653.7	107.69	499.5	715.4	147.89

Таблица Б.4 – Сравнение эффективности алгоритмов на функции 4 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	1.48568	2.677	6.601	45.92	55.05	2.347
T2	49.5044	37.15	8.1906	272.9	289.7	38.977
T3	51.9448	36.67	7.1991	230.1	223.6	17.889
T4	1.50584	0.7926	5.0355	52.76	73.85	1.001
T5	69.4395	67.17	3.121	335.5	285	69.5
T6	2.35478	4.881	3.5162	57.38	98.15	3.723
T7	11.7425	7.792	8.3141	93.45	92.21	8.031

Таблица Б.5 – Сравнение эффективности алгоритмов на функции 5 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	0.159877	1.855	7.9042	27.99	62.22	5.64
T2	0.333918	2.879	10.091	29.57	58.85	5.322
T3	0.357925	3.403	7.2867	25.4	44.51	0.917
T4	0.108105	1.095	6.2507	33.96	91.95	5.763
T5	0.409275	7.986	8.2195	24.42	29.03	6.023
T6	0.229676	4.053	7.9011	31.77	116.9	6.112
T7	0.434294	6.527	10.779	23.19	73.06	8.076

Таблица Б.6 – Сравнение эффективности алгоритмов на функции 6 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Тип изменений	jDE	CPSO	EP with EEM	GA	PSO	Метод 2
T1	6.22948	6.725	17.303	39.41	71.15	6.691
T2	10.3083	21.57	18.732	138.6	158.7	19.97
T3	10.954	27.13	16.005	98.51	140.3	32.6
T4	6.78734	9.27	11.753	53.53	120.7	13.552
T5	14.9455	71.57	26.311	170.1	162.8	50.914
T6	7.8028	23.67	24.558	52.1	113.8	24.892
T7	10.736	32.58	25.231	51.34	101.7	30.773

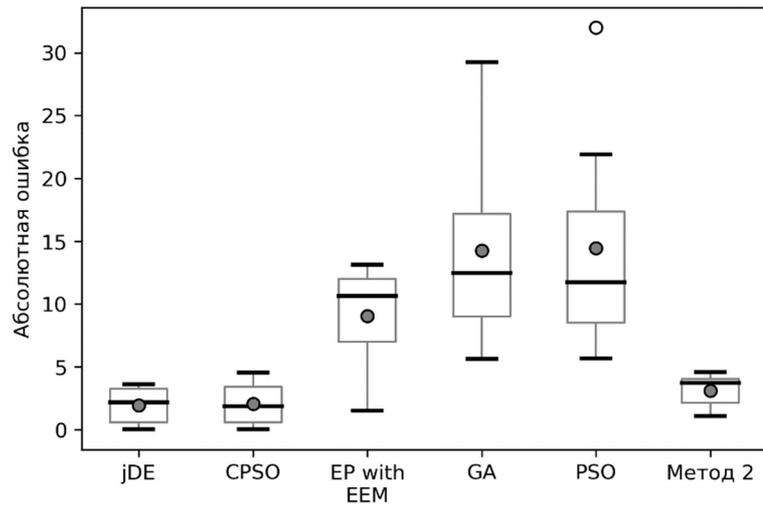


Рисунок Б.1 – Диаграмма размаха оценок эффективности на функции 1 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

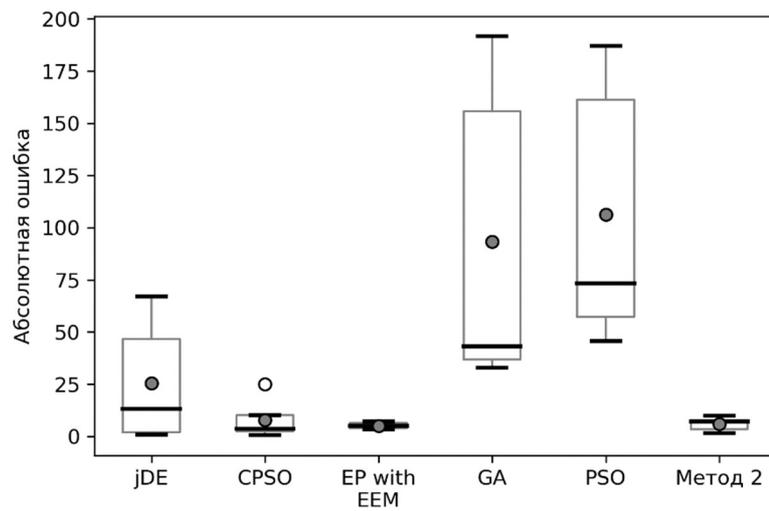


Рисунок Б.2 – Диаграмма размаха оценок эффективности на функции 2 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

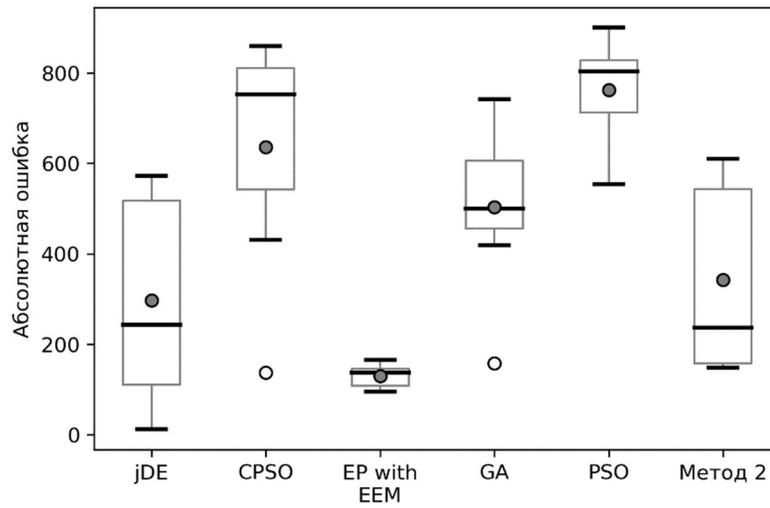


Рисунок Б.3 – Диаграмма размаха оценок эффективности на функции 3 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

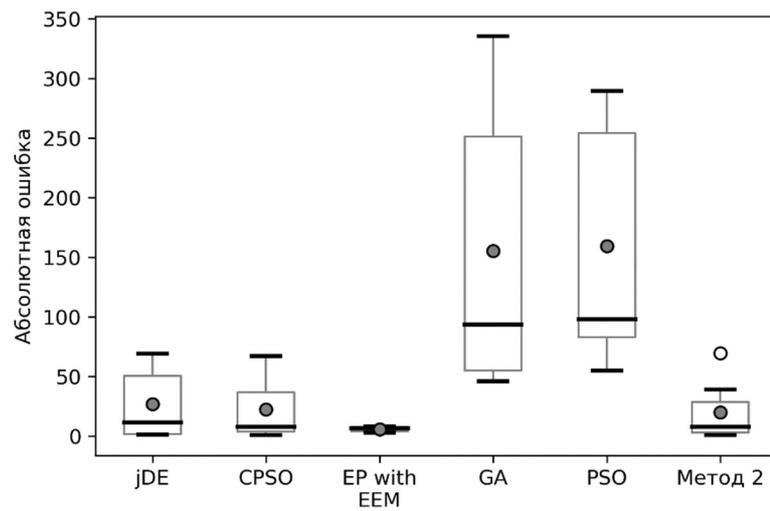


Рисунок Б.4 – Диаграмма размаха оценок эффективности на функции 4 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

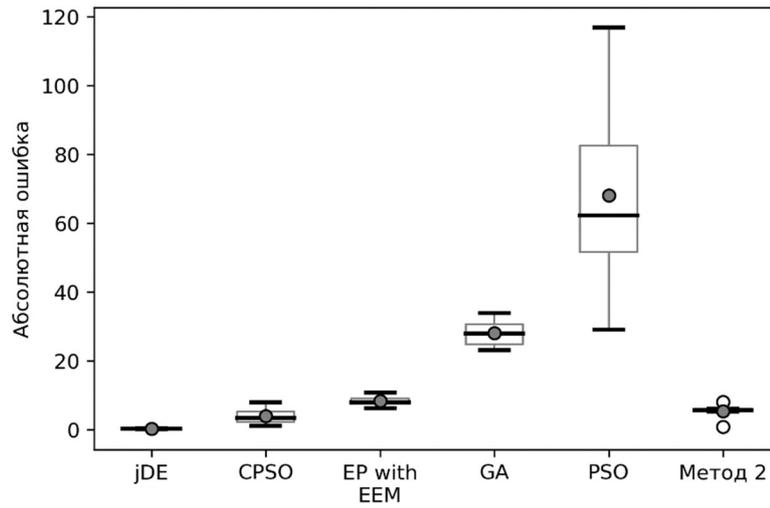


Рисунок Б.5 – Диаграмма размаха оценок эффективности на функции 5 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

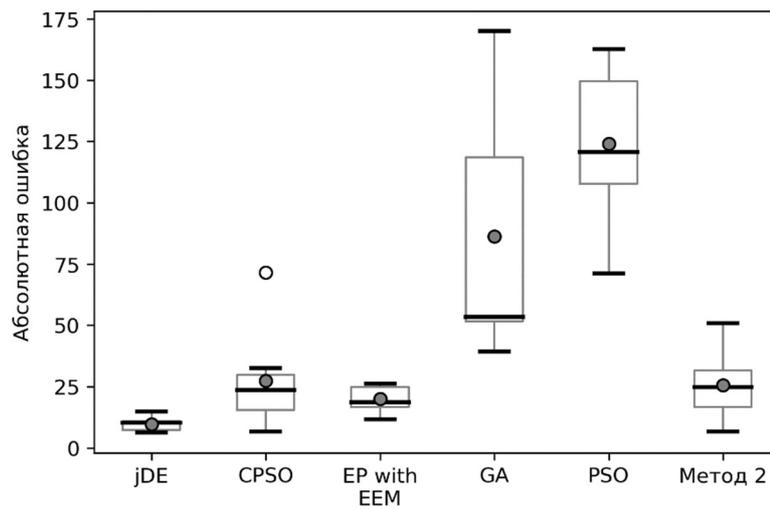


Рисунок Б.6 – Диаграмма размаха оценок эффективности на функции 6 набора эталонных тестовых задачах нестационарной оптимизации IEEE CEC

Таблица Б.7 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 1

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00646	0.02340	0.00334	0.00906	0.00335	0.00294	0.00389	0.00452	0.00267
Угол тангажа								
0.02110	0.05294	0.00947	0.05685	0.00817	0.00965	0.01166	0.01916	0.00821
Угол рыскания								
0.00059	0.03400	0.00164	0.01660	0.00156	0.00198	0.00552	0.00546	0.00152
Среднее								
0.00938	0.03678	0.00482	0.02750	0.00436	0.00485	0.00702	0.00971	0.00413

Таблица Б.8 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 2

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00595	0.00417	0.00717	0.04802	0.00797	0.00822	0.00839	0.01595	0.00659
Угол тангажа								
0.00260	0.05393	0.00615	0.01053	0.00765	0.00656	0.00649	0.00748	0.00607
Угол рыскания								
0.00072	0.00516	0.00058	0.00258	0.00060	0.00052	0.00059	0.00097	0.00057
Среднее								
0.00309	0.02109	0.00463	0.02038	0.00541	0.00510	0.00516	0.00813	0.00441

Таблица Б.9 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 3

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00488	0.00404	0.00149	0.02139	0.01480	0.01555	0.01690	0.01403	0.01028
Угол тангажа								
0.02010	0.05285	0.01150	0.02093	0.00971	0.01168	0.01811	0.01439	0.00897
Угол рыскания								
0.00067	0.05070	0.00096	0.00234	0.00130	0.00172	0.00421	0.00211	0.00093
Среднее								
0.00855	0.03586	0.00465	0.01489	0.00860	0.00965	0.01307	0.01017	0.00673

Таблица Б.10 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 4

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00812	0.00196	0.00285	0.00821	0.00314	0.00304	0.00395	0.00424	0.00267
Угол тангажа								
0.01980	0.05047	0.02077	0.06756	0.02264	0.01836	0.02159	0.03018	0.01451
Угол рыскания								
0.00084	0.03840	0.00079	0.00539	0.00105	0.00135	0.00237	0.00219	0.00058
Среднее								
0.00959	0.03028	0.00814	0.02705	0.00894	0.00758	0.00930	0.01221	0.00592

Таблица Б.11 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 5

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00684	0.01300	0.00467	0.01073	0.00423	0.00481	0.00397	0.00568	0.00289
Угол тангажа								
0.01770	0.04977	0.01063	0.02103	0.00795	0.00872	0.00961	0.01159	0.00604
Угол рыскания								
0.00124	0.03690	0.00071	0.00187	0.00114	0.00166	0.00243	0.00156	0.00050
Среднее								
0.00859	0.03322	0.00534	0.01121	0.00444	0.00506	0.00534	0.00628	0.00314

Таблица Б.12 – Результаты решения задачи идентификации параметров боковой динамики БПЛА на наборе данных 6

Предыдущий результат	НС с алгоритмом обратного распространения ошибки	ЭА с управляемой мутацией	Перезапуск процедуры оптимизации	Локальная адаптация к изменениям	Механизм поддержания разнообразия	Механизм явной памяти	Среднее по базовым эвристикам	Метод 2
Угол крена								
0.00850	0.01470	0.00496	0.01245	0.00494	0.00410	0.00342	0.00597	0.00339
Угол тангажа								
0.01870	0.06042	0.01538	0.05437	0.01996	0.02728	0.02209	0.02781	0.02077
Угол рыскания								
0.00191	0.02470	0.00062	0.00224	0.00092	0.00099	0.00091	0.00114	0.00050
Среднее								
0.00970	0.03327	0.00699	0.02302	0.00860	0.01079	0.00881	0.01164	0.00822

ПРИЛОЖЕНИЕ В
ЭТАЛОННЫЕ ТЕСТОВЫЕ ЗАДАЧИ МУЛЬТИМОДАЛЬНОЙ
ОПТИМИЗАЦИИ С БИНАРНЫМИ ПЕРЕМЕННЫМИ

Функция *binaryF1*

$$x = (x_{11}, x_{12}, \dots, x_{16}, x_{21}, x_{22}, \dots, x_{26}, x_{31}, \dots, x_{36}, x_{41}, \dots, x_{46}, x_{51}, \dots, x_{56}),$$

где $x_{ij} \in \{0,1\}$, бинарный вектор имеет 5 компонент по 6 бинарных переменных, длина бинарного вектора 30.

$$\chi_i = \sum_{j=1}^6 x_{ij}, \quad i = \overline{1,5},$$

$$u(\chi) = \begin{cases} 1, & \chi = 0 \vee \chi = 6 \\ 0, & \chi = 1 \vee \chi = 5 \\ 0.360384, & \chi = 2 \vee \chi = 4 \\ 0.640576, & \chi = 3 \end{cases}$$

$$\text{binaryF1}(x) = \sum_{i=1}^5 u(\chi_i),$$

Функция имеет 32 глобальных оптимума со значением 5 в точках, в которых:

$$\forall i: (x_{i1} = x_{i2} = \dots = x_{i6} = 1) \vee (x_{i1} = x_{i2} = \dots = x_{i6} = 0).$$

Функция *binaryF2*

$$\text{binaryF2}(x) = 5 \cdot \left(\frac{\text{binaryF1}(x)}{5} \right)^{15},$$

Число, значения и позиции глобальных оптимума совпадают с *binaryF1*.

Функция *binaryF3*

$$x = (x_{11}, x_{12}, \dots, x_{18}, x_{21}, x_{22}, \dots, x_{28}, x_{31}, \dots, x_{38}),$$

где $x_{ij} \in \{0,1\}$, бинарный вектор имеет 3 компоненты по 8 бинарных переменных, длина бинарного вектора 24.

Функция имеет 27 глобальных оптимума в точках, в которых компоненты содержат (0,0,0,0,0,0,0,0) или (1,0,0,0,1,1,0,0) или (0,1,0,0,1,0,1,0). Вклад каждой компоненты в значение функции равен 10. Значение в глобальных оптимумах равно 30.

Для всех остальных значений компонент вектора вклад в значение функции равен расстоянию Хемминга до ближайшей из компонент глобального оптимума.

Функция *binaryF4*

Вычисляется аналогично *binaryF3*, за исключением следующего: бинарный вектор имеет 5 компонент по 6 бинарных переменных, длина бинарного вектора 30. Компоненты глобального решения содержат (0,1,0,0,1,1) или (1,0,0,0,0,1) и вносят вклад в значение функции равный 6. Функция имеет 32 глобальных оптимума со значением 30.

Функция *binaryF5*

$$x = (x_{11}, x_{12}, \dots, x_{16}, x_{21}, x_{22}, \dots, x_{26}, x_{31}, \dots, x_{36}, x_{41}, \dots, x_{46}, x_{51}, \dots, x_{56}),$$

где $x_{ij} \in \{0,1\}$, бинарный вектор имеет 5 компонент по 6 бинарных переменных, длина бинарного вектора 30.

$$\chi_i = \sum_{j=1}^6 x_{ij}, \quad i = \overline{1,5},$$

$$u(\chi) = \begin{cases} 1 - 0.2 \cdot \chi, & \chi \leq 5 \\ 1, & \chi = 6 \end{cases},$$

$$\text{binaryF5}(x) = \sum_{i=1}^5 u(\chi_i),$$

Функция имеет 32 глобальных оптимума со значением 6.

Функция *binaryF6*

Бинарный вектор имеет 5 компонент по 8 бинарных переменных, длина бинарного вектора 40.

Функция имеет 32 глобальных оптимума в точках, в которых компоненты содержат (1,0,1,0,0,0,0,0) или (0,1,1,1,1,1,1,0). Вклад каждой компоненты в значение функции равен 1. Значение в глобальных оптимумах равно 5.

Для всех остальных значений компонент вектора вклад в значение функции вычисляется как:

$$u(\chi) = \begin{cases} 0.6, & \chi = 0 \vee 4 \vee 8 \\ 0.3, & \chi = 1 \vee 3 \vee 5 \vee 7 \\ 0, & \chi = 2 \vee 6 \end{cases}.$$

ПРИЛОЖЕНИЕ Г

РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ ИССЛЕДОВАНИЯ
ЭФФЕКТИВНОСТИ МЕТОДА 3

Таблица Г.1 – Результаты численных экспериментов при решении задач аппроксимации множества экстремумов с бинарными переменными

Критерий	ENA	Обнуление пригодности	Разделение пригодности	Формирование ниш на основе кластеризации	Ограниченная турнирная селекция	Детерминированный контроль скопления	Вероятностный контроль скопления	Среднее по базовым эвристикам	Метод 3
Задача <i>binaryF1</i>									
PR	-	0.94	0.84	0.91	1.00	0.97	0.78	0.91	1.00
SR	1.00	0.90	0.84	0.88	1.00	0.94	0.80	0.89	1.00
PD	-	2.40	3.37	2.40	0.00	2.33	3.30	2.30	0.00
Задача <i>binaryF2</i>									
PR	-	0.97	0.97	1.00	1.00	0.97	0.84	0.96	1.00
SR	1.00	0.96	0.98	1.00	1.00	0.94	0.84	0.95	1.00
PD	-	2.00	1.00	0.00	0.00	1.67	3.62	1.38	0.00
Задача <i>binaryF3</i>									
PR	-	1.00	0.96	0.96	0.93	0.96	0.89	0.95	1.00
SR	1.00	1.00	0.96	0.94	0.90	0.94	0.84	0.93	1.00
PD	-	0.00	2.50	2.67	2.80	2.67	3.37	2.34	0.00
Задача <i>binaryF4</i>									
PR	-	0.91	0.81	0.91	1.00	0.94	0.75	0.89	1.00
SR	1.00	0.92	0.92	0.90	1.00	0.94	0.80	0.91	1.00
PD	-	3.25	2.50	2.60	0.00	2.67	3.20	2.37	0.00
Задача <i>binaryF5</i>									
PR	-	0.88	0.88	0.84	0.88	0.88	0.72	0.85	1.00
SR	1.00	0.88	0.86	0.84	0.86	0.84	0.64	0.82	1.00
PD	-	2.33	2.57	2.62	2.71	2.37	3.06	2.61	0.00
Задача <i>binaryF6</i>									
PR	-	0.84	0.75	0.84	0.88	0.78	0.56	0.78	1.00
SR	0.99	0.84	0.80	0.86	0.84	0.76	0.66	0.79	1.00
PD	-	3.25	2.80	3.00	2.87	3.08	3.47	3.08	0.00

Таблица Г.2 – Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC, точность $1E-01$

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
cecF1 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF2 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF3 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF4 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF5 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF6 2D	0.56	0.00	0.96	0.48	1.00	1.00	1.00	1.00	0.44	0.00	1.00	1.00	0.78	0.02	0.76	0.43
cecF7 2D	1.00	1.00	0.95	0.16	1.00	1.00	1.00	1.00	0.34	0.00	0.70	0.00	0.53	0.00	0.81	0.46
cecF6 3D	0.35	0.00	0.24	0.00	0.99	0.46	0.41	0.00	0.11	0.00	0.85	0.00	0.11	0.00	0.66	0.17
cecF7 3D	0.48	0.00	0.62	0.00	0.83	0.00	1.00	1.00	0.10	0.00	0.27	0.00	0.28	0.00	0.62	0.41
cecF8 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF9 2D	0.88	0.42	0.98	0.88	0.87	0.54	1.00	1.00	0.67	0.00	0.67	0.00	0.99	0.94	0.91	1.00
cecF10 2D	0.75	0.00	0.85	0.18	0.75	0.00	0.84	0.12	0.62	0.00	0.37	0.00	0.79	0.06	0.78	0.09
cecF11 2D	0.70	0.00	0.98	0.86	0.74	0.16	1.00	0.98	0.67	0.00	0.67	0.00	0.98	0.88	0.89	0.71
cecF11 3D	0.93	0.70	0.83	0.16	0.94	0.78	1.00	1.00	0.67	0.00	0.68	0.00	0.76	0.02	0.84	0.59
cecF12 3D	0.67	0.06	0.74	0.02	1.00	1.00	1.00	1.00	0.53	0.00	0.73	0.02	0.68	0.02	0.83	0.45
cecF11 5D	1.00	1.00	0.67	0.00	0.89	0.60	1.00	1.00	0.67	0.00	0.68	0.02	0.67	0.00	0.88	0.51
cecF12 5D	0.92	0.64	0.70	0.00	0.96	0.84	1.00	1.00	0.34	0.00	0.56	0.10	0.59	0.00	0.80	0.63
cecF11 10D	0.64	0.64	0.67	0.00	0.66	0.02	1.00	0.98	0.45	0.00	0.56	0.16	0.34	0.00	0.68	0.41
cecF12 10D	0.02	0.00	0.67	0.00	0.50	0.24	0.27	0.14	0.22	0.00	0.00	0.00	0.60	0.00	0.40	0.11
cecF12 20D	0.00	0.00	0.36	0.00	0.08	0.02	0.00	0.00	0.14	0.00	0.36	0.28	0.45	0.00	0.18	0.11
Среднее	0.74	0.52	0.81	0.44	0.86	0.63	0.88	0.81	0.60	0.30	0.70	0.38	0.73	0.40	0.80	0.60
Ранг по PR	4.80		3.85		3.62		2.92		6.50		5.40		4.80		4.10	
Ранг по SR	4.62		4.95		3.57		2.80		6.10		5.12		5.20		3.57	

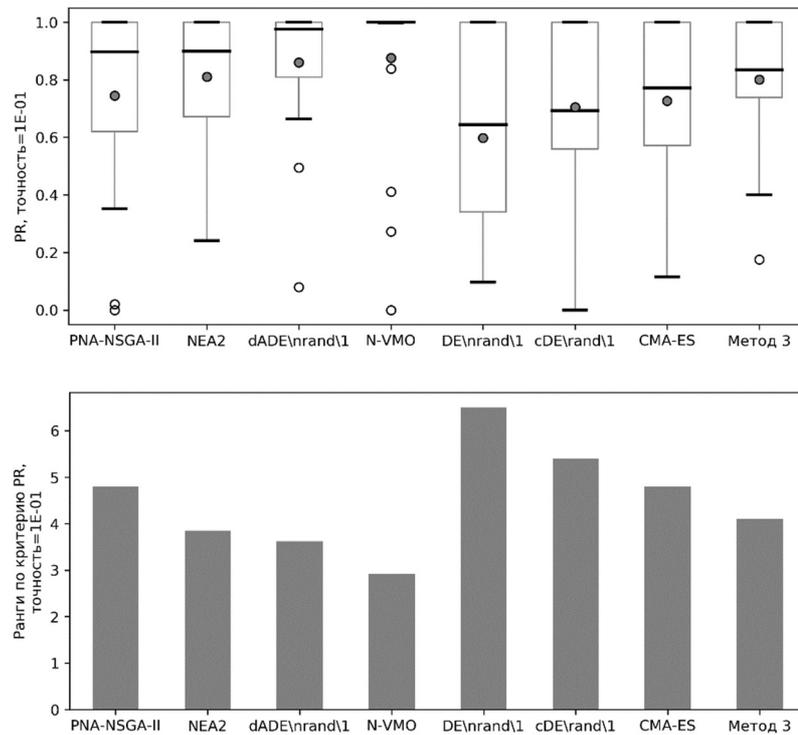


Рисунок Г.1 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок PR при решении эталонных тестовых задач IEEE CEC, точность $1E-01$

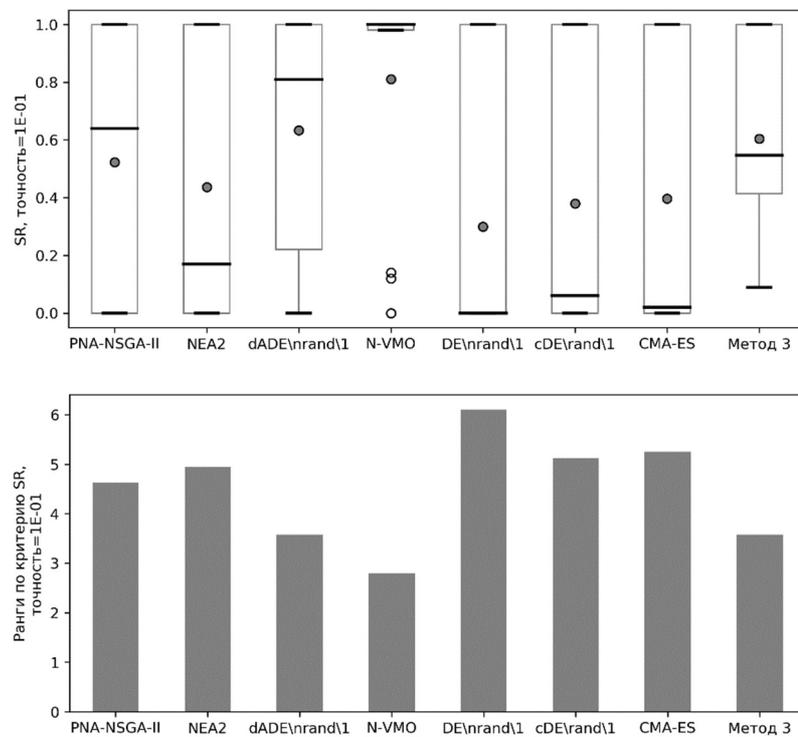


Рисунок Г.2 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок SR при решении эталонных тестовых задач IEEE CEC, точность $1E-01$

Таблица Г.3 – Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC, точность 1E-02

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
сесF1 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.48	1.00	1.00	0.78	0.85
сесF2 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF3 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF4 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF5 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF6 2D	0.54	0.00	0.96	0.48	1.00	1.00	1.00	0.92	0.44	0.00	1.00	1.00	0.78	0.02	0.84	0.48
сесF7 2D	0.74	0.00	0.93	0.08	0.96	0.18	1.00	1.00	0.34	0.00	0.70	0.00	0.53	0.00	0.87	0.39
сесF6 3D	0.33	0.00	0.24	0.00	0.99	0.50	0.30	0.00	0.11	0.00	0.84	0.00	0.11	0.00	0.45	0.13
сесF7 3D	0.33	0.00	0.59	0.00	0.59	0.00	0.69	0.00	0.10	0.00	0.27	0.00	0.28	0.00	0.25	0.00
сесF8 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF9 2D	0.68	0.00	0.97	0.80	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.98	0.86	0.73	0.30
сесF10 2D	0.72	0.00	0.85	0.18	0.75	0.00	0.74	0.00	0.61	0.00	0.09	0.00	0.79	0.06	0.85	0.07
сесF11 2D	0.67	0.00	0.97	0.82	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.97	0.80	0.77	0.38
сесF11 3D	0.67	0.00	0.82	0.10	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.75	0.02	0.72	0.03
сесF12 3D	0.50	0.00	0.72	0.00	0.64	0.00	0.72	0.00	0.52	0.00	0.70	0.00	0.66	0.00	0.73	0.00
сесF11 5D	0.52	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.63	0.00
сесF12 5D	0.35	0.00	0.70	0.00	0.48	0.00	0.48	0.00	0.32	0.00	0.41	0.00	0.59	0.00	0.48	0.00
сесF11 10D	0.12	0.00	0.67	0.00	0.63	0.00	0.47	0.00	0.35	0.00	0.28	0.00	0.34	0.00	0.51	0.00
сесF12 10D	0.02	0.00	0.67	0.00	0.12	0.00	0.13	0.00	0.18	0.00	0.00	0.00	0.60	0.00	0.32	0.00
сесF12 20D	0.00	0.00	0.36	0.00	0.01	0.00	0.00	0.00	0.13	0.00	0.01	0.00	0.45	0.00	0.19	0.00
Среднее	0.61	0.30	0.81	0.42	0.74	0.38	0.71	0.40	0.59	0.30	0.63	0.32	0.72	0.39	0.71	0.38
Ранг по PR	5.52		2.80		4.37		3.92		5.90		5.45		4.05		3.97	
Ранг по SR	5.05		3.80		4.35		4.55		5.05		4.97		4.22		4.00	

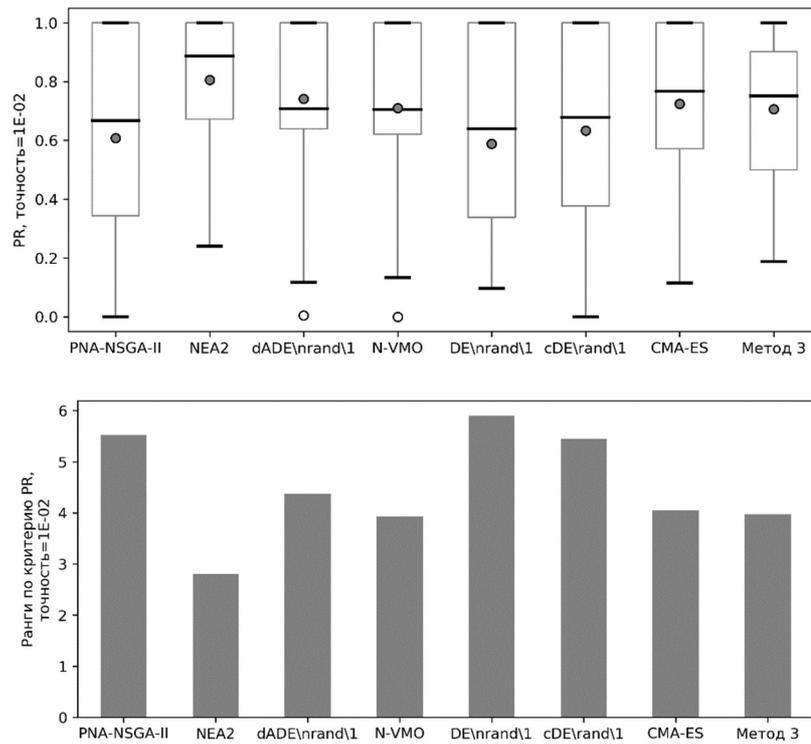


Рисунок Г.3 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок PR при решении эталонных тестовых задач IEEE CEC, точность $1E-02$

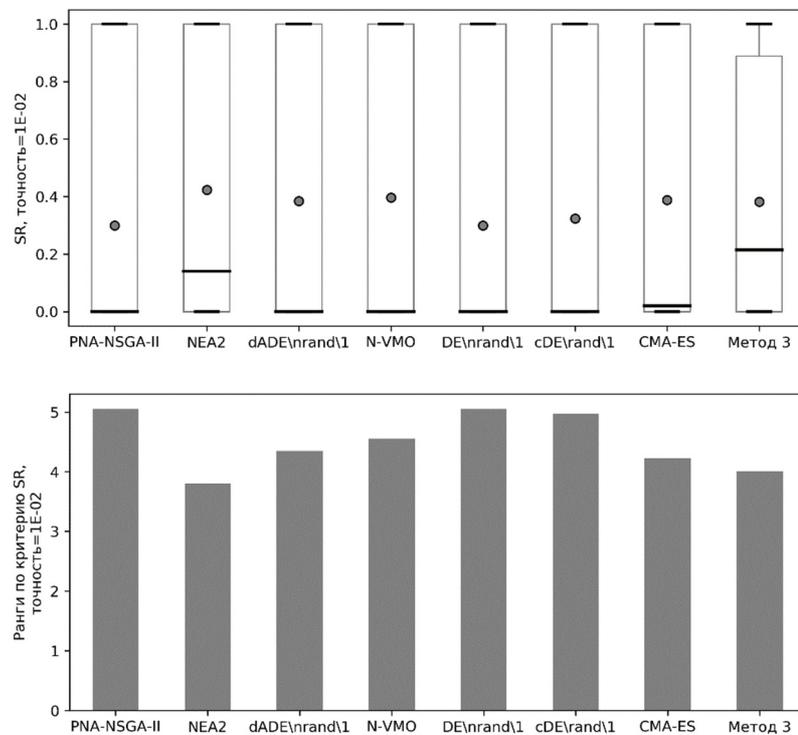


Рисунок Г.4 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок SR при решении эталонных тестовых задач IEEE CEC, точность $1E-02$

Таблица Г.4 – Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC, точность $1E-03$

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
сесF1 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.15	0.02	1.00	1.00	0.84	1.00
сесF2 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF3 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF4 2D	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF5 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
сесF6 2D	0.52	0.00	0.96	0.44	1.00	1.00	0.94	0.34	0.44	0.00	0.95	0.48	0.78	0.02	0.96	0.40
сесF7 2D	0.73	0.00	0.92	0.06	0.88	0.00	0.95	0.12	0.34	0.00	0.70	0.00	0.52	0.00	0.59	0.05
сесF6 3D	0.31	0.00	0.24	0.00	0.98	0.30	0.28	0.00	0.11	0.00	0.71	0.00	0.11	0.00	0.42	0.07
сесF7 3D	0.32	0.00	0.58	0.00	0.55	0.00	0.41	0.00	0.10	0.00	0.27	0.00	0.27	0.00	0.44	0.00
сесF8 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.99
сесF9 2D	0.67	0.00	0.97	0.80	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.97	0.82	0.85	0.27
сесF10 2D	0.67	0.00	0.84	0.18	0.74	0.00	0.73	0.00	0.61	0.00	0.01	0.00	0.78	0.06	0.91	0.02
сесF11 2D	0.67	0.00	0.96	0.76	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.96	0.76	0.81	0.32
сесF11 3D	0.67	0.00	0.81	0.08	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.74	0.02	0.70	0.03
сесF12 3D	0.49	0.00	0.72	0.00	0.62	0.00	0.68	0.00	0.52	0.00	0.64	0.00	0.66	0.00	0.58	0.00
сесF11 5D	0.52	0.00	0.67	0.00	0.67	0.00	0.66	0.00	0.66	0.00	0.67	0.00	0.67	0.00	0.62	0.00
сесF12 5D	0.34	0.00	0.70	0.00	0.42	0.00	0.45	0.00	0.31	0.00	0.29	0.00	0.59	0.00	0.57	0.00
сесF11 10D	0.11	0.00	0.67	0.00	0.63	0.00	0.46	0.00	0.31	0.00	0.20	0.00	0.34	0.00	0.35	0.00
сесF12 10D	0.04	0.00	0.67	0.00	0.08	0.00	0.13	0.00	0.15	0.00	0.00	0.00	0.60	0.00	0.21	0.00
сесF12 20D	0.00	0.00	0.36	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.45	0.00	0.21	0.00
Среднее	0.60	0.30	0.80	0.42	0.73	0.37	0.68	0.32	0.58	0.30	0.58	0.28	0.72	0.38	0.70	0.36
Ранг по PR	5.57		2.47		4.17		4.20		5.92		5.55		4.02		4.07	
Ранг по SR	5.17		3.62		4.42		4.60		4.97		4.90		4.30		4.00	

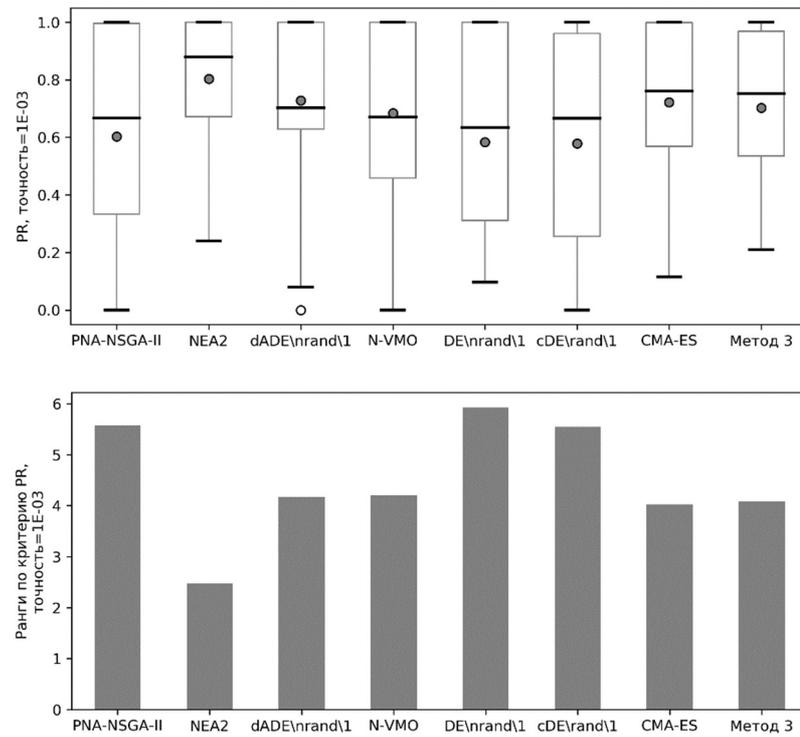


Рисунок Г.5 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок PR при решении эталонных тестовых задач IEEE CEC, точность $1E-03$

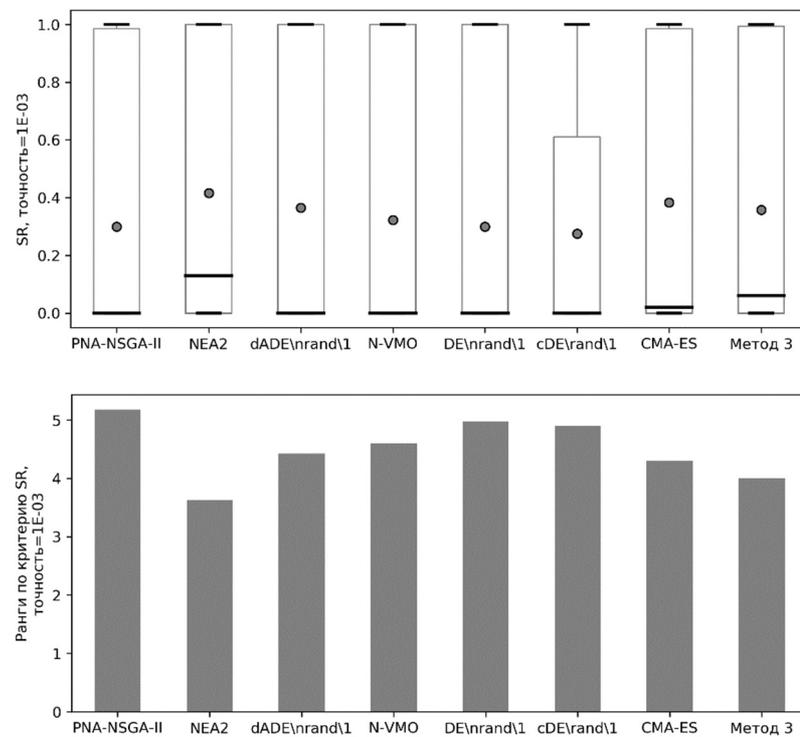


Рисунок Г.6 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок SR при решении эталонных тестовых задач IEEE CEC, точность $1E-03$

Таблица Г.5 – Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC, точность 1E-04

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
cecF1 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.11	0.00	1.00	1.00	1.00	0.71
cecF2 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF3 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF4 2D	0.99	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	0.99
cecF5 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF6 2D	0.47	0.00	0.95	0.38	0.99	0.78	0.68	0.00	0.44	0.00	0.10	0.00	0.78	0.02	0.69	0.27
cecF7 2D	0.71	0.00	0.91	0.04	0.81	0.00	0.90	0.00	0.34	0.00	0.70	0.00	0.52	0.00	0.80	0.01
cecF6 3D	0.28	0.00	0.24	0.00	0.96	0.16	0.20	0.00	0.11	0.00	0.29	0.00	0.11	0.00	0.41	0.02
cecF7 3D	0.30	0.00	0.58	0.00	0.44	0.00	0.28	0.00	0.10	0.00	0.27	0.00	0.27	0.00	0.46	0.00
cecF8 2D	1.00	1.00	0.99	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.90	1.00	0.97
cecF9 2D	0.68	0.00	0.96	0.76	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.96	0.78	0.73	0.34
cecF10 2D	0.64	0.00	0.84	0.16	0.74	0.00	0.71	0.00	0.61	0.00	0.00	0.00	0.76	0.06	0.34	0.05
cecF11 2D	0.66	0.00	0.96	0.74	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.95	0.72	0.79	0.20
cecF11 3D	0.66	0.00	0.81	0.06	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.74	0.02	0.69	0.01
cecF12 3D	0.47	0.00	0.72	0.00	0.60	0.00	0.64	0.00	0.51	0.00	0.53	0.00	0.66	0.00	0.58	0.00
cecF11 5D	0.42	0.00	0.67	0.00	0.67	0.00	0.66	0.00	0.66	0.00	0.67	0.00	0.67	0.00	0.65	0.00
cecF12 5D	0.30	0.00	0.70	0.00	0.40	0.00	0.44	0.00	0.30	0.00	0.16	0.00	0.59	0.00	0.48	0.00
cecF11 10D	0.11	0.00	0.67	0.00	0.61	0.00	0.46	0.00	0.29	0.00	0.18	0.00	0.34	0.00	0.51	0.00
cecF12 10D	0.02	0.00	0.67	0.00	0.02	0.00	0.13	0.00	0.14	0.00	0.00	0.00	0.60	0.00	0.30	0.00
cecF12 20D	0.00	0.00	0.36	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.45	0.00	0.16	0.00
Среднее	0.59	0.30	0.80	0.40	0.71	0.35	0.65	0.30	0.58	0.30	0.50	0.25	0.72	0.38	0.68	0.33
Ранг по PR	5.80		2.47		3.97		4.30		5.67		5.97		3.72		4.07	
Ранг по SR	5.05		3.65		4.30		4.80		4.80		5.22		4.02		4.15	

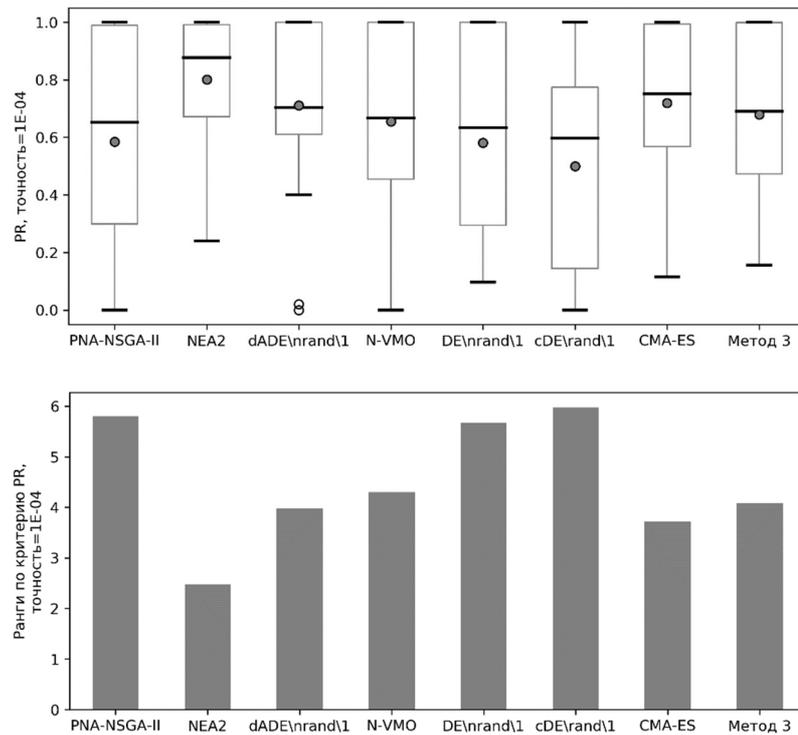


Рисунок Г.7 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок PR при решении эталонных тестовых задач IEEE CEC, точность $1E-04$

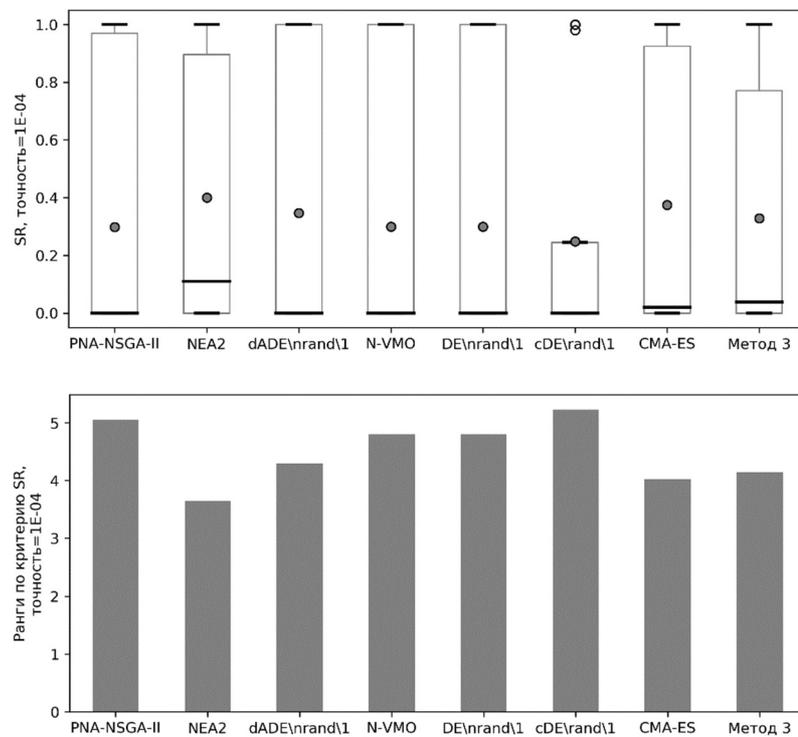


Рисунок Г.8 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок SR при решении эталонных тестовых задач IEEE CEC, точность $1E-04$

Таблица Г.6 – Результаты численных экспериментов при решении эталонных тестовых задач аппроксимации множества экстремумов IEEE CEC, точность $1E-05$

Алгоритм	PNA-NSGA-II		NEA2		dADE\rand\1		N-VMO		DE\rand\1		cDE\rand\1		CMA-ES		Метод 3	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
cecF1 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.10	0.00	1.00	1.00	0.53	0.49
cecF2 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF3 1D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF4 2D	0.81	0.42	0.99	0.96	1.00	1.00	1.00	1.00	1.00	1.00	0.60	0.16	0.99	0.96	0.98	1.00
cecF5 2D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cecF6 2D	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
cecF7 2D	0.68	0.00	0.91	0.04	0.71	0.00	0.80	0.00	0.34	0.00	0.70	0.00	0.52	0.00	0.53	0.01
cecF6 3D	0.25	0.00	0.24	0.00	0.95	0.10	0.03	0.00	0.11	0.00	0.05	0.00	0.11	0.00	0.17	0.02
cecF7 3D	0.28	0.00	0.58	0.00	0.35	0.00	0.19	0.00	0.10	0.00	0.27	0.00	0.27	0.00	0.34	0.00
cecF8 2D	1.00	1.00	0.98	0.76	1.00	1.00	0.97	0.68	1.00	1.00	1.00	1.00	0.98	0.76	0.98	0.97
cecF9 2D	0.66	0.00	0.96	0.76	0.67	0.00	0.67	0.00	0.67	0.00	0.67	0.00	0.95	0.74	0.76	0.06
cecF10 2D	0.57	0.00	0.83	0.14	0.74	0.00	0.59	0.00	0.60	0.00	0.00	0.00	0.76	0.06	0.41	0.01
cecF11 2D	0.62	0.00	0.95	0.70	0.67	0.00	0.66	0.00	0.67	0.00	0.67	0.00	0.95	0.70	0.74	0.11
cecF11 3D	0.61	0.00	0.80	0.06	0.67	0.00	0.63	0.00	0.66	0.00	0.67	0.00	0.74	0.02	0.65	0.01
cecF12 3D	0.44	0.00	0.71	0.00	0.62	0.00	0.38	0.00	0.50	0.00	0.39	0.00	0.65	0.00	0.52	0.00
cecF11 5D	0.32	0.00	0.67	0.00	0.67	0.00	0.65	0.00	0.66	0.00	0.67	0.00	0.66	0.00	0.66	0.00
cecF12 5D	0.25	0.00	0.70	0.00	0.36	0.00	0.33	0.00	0.29	0.00	0.05	0.00	0.58	0.00	0.39	0.00
cecF11 10D	0.09	0.00	0.66	0.00	0.60	0.00	0.33	0.00	0.24	0.00	0.17	0.00	0.34	0.00	0.41	0.00
cecF12 10D	0.01	0.00	0.67	0.00	0.00	0.00	0.09	0.00	0.12	0.00	0.00	0.00	0.60	0.00	0.21	0.00
cecF12 20D	0.00	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.43	0.00	0.10	0.00
Среднее	0.53	0.27	0.75	0.37	0.65	0.31	0.57	0.28	0.55	0.30	0.45	0.21	0.68	0.36	0.57	0.28
Ранг по PR	5.67		2.42		3.75		5.40		4.90		5.67		3.67		4.50	
Ранг по SR	4.87		3.80		4.42		4.92		4.65		5.15		4.17		4.00	

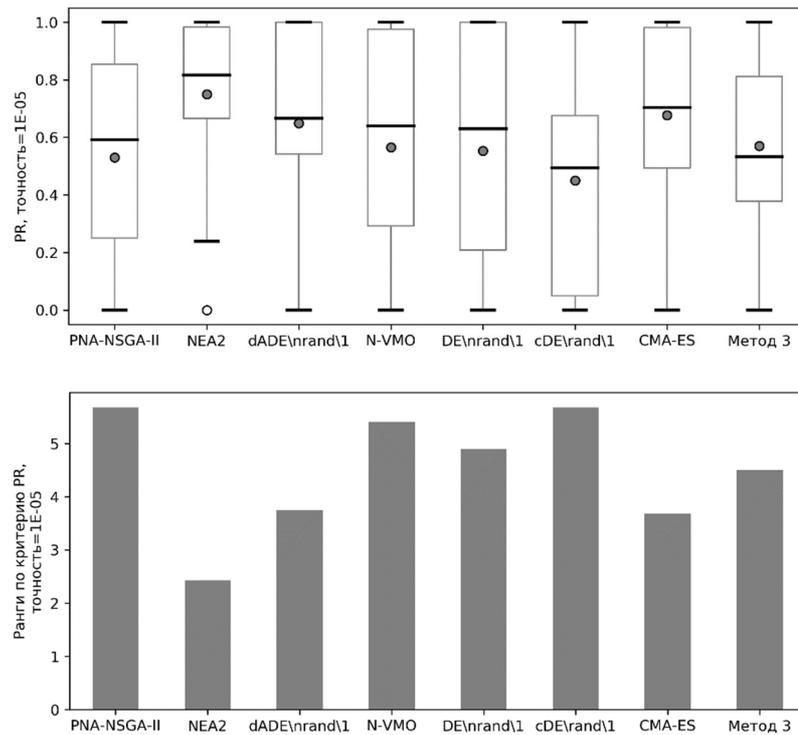


Рисунок Г.9 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок PR при решении эталонных тестовых задач IEEE CEC, точность $1E-05$

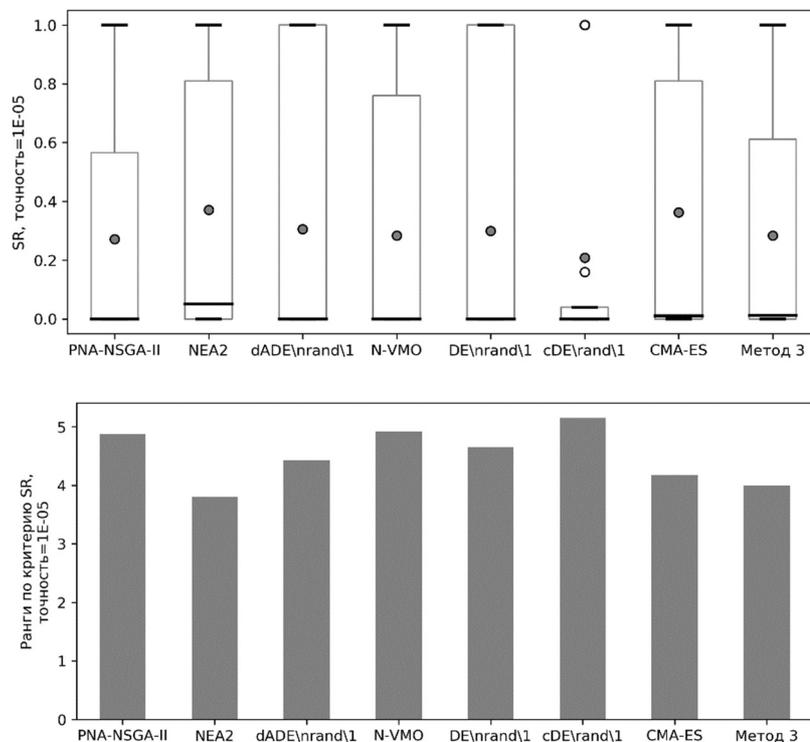


Рисунок Г.10 – Диаграмма размаха (вверху) и ранжирование (внизу) оценок SR при решении эталонных тестовых задач IEEE CEC, точность $1E-05$

Таблица Г.7 – Данные Красноярского филиала банка Москвы для задачи формирования оптимальной кредитной политики коммерческого банка

№	Размер заявки k	Процент за использование d	Срок кредитования t	Оценка рискованности P	№	Размер заявки k	Процент за использование d	Срок кредитования t	Оценка рискованности P
1	10000000	25	75	0.042	26	640000	24	91	0.042
2	5300000	28	80	0.039	27	440000	28	67	0.029
3	2400000	25	91	0.029	28	460000	28	91	0.018
4	50000000	23	84	0.033	29	6000000	27	62	0.021
5	1000000	28	64	0.026	30	7100000	26	78	0.036
6	500000	30	76	0.046	31	3260000	27	87	0.027
7	250000	37	91	0.044	32	2670000	25	75	0.014
8	100000	30	86	0.012	33	620000	27	82	0.038
9	330000	26	90	0.026	34	20000000	24	91	0.019
10	5600000	28	88	0.039	35	10000000	24	90	0.026
11	7300000	25	76	0.02	36	35000000	22	89	0.022
12	1220000	27	80	0.037	37	5100000	29	69	0.036
13	2900000	31	84	0.03	38	865000	30	74	0.021
14	950000	29	86	0.041	39	675000	27	63	0.017
15	4360000	25	88	0.021	40	4650000	29	69	0.026
16	3700000	26	90	0.035	41	135000	28	70	0.03
17	400000	26	79	0.029	42	400000	27	76	0.041
18	280000	28	84	0.03	43	1640000	26	87	0.017
19	5200000	30	91	0.039	44	1380000	29	88	0.021
20	1280000	27	90	0.04	45	1950000	26	71	0.013
21	8400000	25	86	0.035	46	1000000	27	85	0.014
22	670000	29	80	0.015	47	6900000	27	82	0.016
23	790000	28	84	0.024	48	9000000	27	86	0.024
24	950000	26	83	0.034	49	22000000	29	91	0.016
25	580000	27	90	0.038	50	350000	27	69	0.026
Сумма свободных пассивов $B=188500000$									

Таблица Г.8 – Данные «Химзавод» – филиала АО «Красмаш» для задачи управления инвестициями производственного предприятия ОПК

№	Проекты	Π_{ij}	R_{ij}	B_i	b_{ij}
ЦФО 1. ПТНП					
1	Производство форточных вентиляторов из УВС	3.5	2.2		2.5
2	Организация совместной деятельности по производству посуды из полимеров	5.0	3.0		3.0

3	Литье под давлением изделий из полимеров	3.2	1.8		4.9
4	Изготовление блоков-приборов и других комплектующих для завода холодильников "Бирюса"	6.2	1.9		8.0
5	Продукция ветеринарного и зоотехнического назначения	4.0	1.2		11.3
6	Производство настенных воздухоочистителей-ионизаторов из УВС	11.5	3.0		25.3
7	Производство из УВС нагревателей воздуха для автокарбюраторов	6.7	2.4		15.1
8	Производство тепловентиляторов из УВС	7.5	2.3		15.5
Всего по ЦФО 1:		47.6	17.8	70	85.6
ЦФО 2. ПЭП					
9	Трубы полиэтиленовые с радиационной обработкой	9.4	3.1		22.0
10	Производство профилей из соэкструдированного жесткого и мягкого ПВХ	7.6	1.5		9.8
11	Экструзия полимеров	4.5	1.4		10.8
12	Производство комплектующих для уплотнителей дверей холодильников	6.0	1.2		11.0
13	Утилизация ПЭТ бутылок	4.0	1.5		17.0
14	Производство напольных офисных покрытий	9.5	2.3		12.5
Всего по ЦФО 2:		41	11	72	83.1
ЦФО 3. Цех 43					
15	Организация совметной деятельности по газификации жидкого кислорода	4.3	1.5		10.2
16	Утилизация супертоксикантов	4.5	1.2		17.0
17	Производство сжиженного и газообразного аргона	3.7	2.2		6.8
18	Производство сжиженного и газообразного азота	4.6	2.2		5.3
19	Производство сжиженного и газообразного кислорода	3.0	2.2		3.2
Всего по ЦФО 3:		20.1	9.3	35	42.5
ЦФО 4. ПТПП					
20	Производство геополотна	6.0	2.1		13.6
21	Производство тканой полипропиленовой продукции	10.0	3.1		20.0
22	Производство пропиленовых мешков	6.9	3.3		9.1
Всего по ЦФО 4:		22.9	8.5	32	42.7
ЦФО 5. Цех 40					
23	Производство дифференциального редуктора	3.0	3.5		3.0
24	Производство электромагнитного инжектора	2.5	4.0		3.5
25	Комплектация ГБО ГИГ	1.0	1.5		0.5
Всего по ЦФО 5:		6.5	9	12	7
Итого по предприятию:		138.1	55.6	221	260.9

ПРИЛОЖЕНИЕ Д
РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ ИССЛЕДОВАНИЯ
ЭФФЕКТИВНОСТИ МЕТОДА 4

Таблица Д.1 – Сравнение метода 4 и ЭА, использующих одну базовую эвристику на наборе эталонных тестовых задач IEEE CEC LSGO 2013 (МО – математическое ожидание, СКО – среднеквадратическое отклонение)

Задача	Случайная группировка	Дифференциальная группировка	Дельта-группировка	EDA-GA	AVS-RG	Среднее по базовым эвристикам	Метод 4	
f1	Медиана	8.26E+02	1.02E+06	1.61E+06	4.59E-05	1.38E+05	5.54E+05	5.73E-06
	МО	1.64E+03	1.04E+06	1.54E+06	5.68E-04	1.69E+05	5.50E+05	3.25E-05
	СКО	1.94E+03	1.13E+05	5.19E+05	4.29E-04	2.83E+04	1.32E+05	3.75E-04
f2	Медиана	2.72E+03	4.72E+03	2.64E+03	1.82E+03	5.59E+03	3.50E+03	1.05E+03
	МО	2.69E+03	4.74E+03	2.66E+03	3.34E+03	5.60E+03	3.81E+03	2.54E+03
	СКО	1.02E+02	2.24E+02	1.70E+02	2.54E+02	1.08E+03	3.66E+02	2.38E+02
f3	Медиана	2.00E+01	2.05E+01	2.02E+01	2.94E-05	2.94E+00	1.27E+01	1.57E-04
	МО	2.00E+01	2.05E+01	2.02E+01	4.81E-01	6.87E-01	1.24E+01	4.78E-01
	СКО	7.86E-03	5.25E-02	1.90E-02	2.28E-01	8.50E-03	6.32E-02	4.80E-02
f4	Медиана	8.59E+08	1.45E+10	2.37E+09	6.60E+09	1.11E+09	5.09E+09	1.67E+09
	МО	9.48E+08	1.49E+10	2.45E+09	2.32E+10	1.54E+09	8.61E+09	6.82E+09
	СКО	3.78E+08	2.21E+09	9.95E+08	1.14E+10	3.78E+08	3.07E+09	4.06E+08
f5	Медиана	5.63E+06	1.60E+06	5.90E+06	7.59E+14	1.74E+06	1.52E+14	3.14E+13
	МО	5.66E+06	1.59E+06	5.82E+06	9.75E+14	2.32E+06	1.95E+14	5.48E+13
	СКО	1.28E+06	2.38E+05	1.10E+06	2.18E+06	2.38E+05	1.01E+06	1.65E+06
f6	Медиана	1.06E+06	9.96E+05	1.06E+06	6.25E+04	3.73E+05	7.10E+05	1.69E+05
	МО	1.06E+06	9.96E+05	1.06E+06	4.75E+05	3.73E+05	7.93E+05	3.50E+05
	СКО	1.29E+03	5.57E+02	1.49E+03	3.35E+05	1.69E+05	1.01E+05	9.17E+04
f7	Медиана	1.31E+06	1.36E+06	6.27E+06	7.65E+07	1.55E+06	1.74E+07	1.88E+06
	МО	1.33E+06	1.40E+06	6.90E+06	2.53E+08	1.33E+06	5.28E+07	2.06E+07
	СКО	1.35E+05	1.81E+05	3.55E+06	8.35E+07	2.74E+05	1.75E+07	1.75E+07
f8	Медиана	3.98E+13	5.68E+13	5.47E+13	4.49E+13	3.98E+13	4.72E+13	2.10E+13
	МО	4.25E+13	5.58E+13	5.35E+13	3.64E+14	4.25E+13	1.12E+14	4.34E+13
	СКО	1.84E+13	2.04E+13	1.65E+13	5.21E+14	1.65E+13	1.19E+14	2.66E+13
f9	Медиана	4.74E+08	1.61E+08	4.41E+08	4.15E+08	1.61E+08	3.30E+08	3.80E+08
	МО	4.96E+08	1.56E+08	4.54E+08	8.06E+08	2.21E+08	4.27E+08	4.24E+08
	СКО	1.17E+08	2.75E+07	9.83E+07	1.72E+08	2.76E+07	8.85E+07	1.21E+08
f10	Медиана	9.29E+07	9.05E+07	9.32E+07	6.18E+06	1.02E+07	5.86E+07	8.61E+06
	МО	9.30E+07	9.14E+07	9.33E+07	1.61E+07	1.61E+07	6.20E+07	2.49E+07
	СКО	5.71E+05	1.64E+06	3.65E+05	7.89E+06	3.65E+05	2.17E+06	9.49E+05

f11	Медиана	1.32E+08	2.90E+08	1.32E+08	2.60E+10	1.32E+08	5.34E+09	1.36E+08
	МО	1.63E+08	4.67E+08	1.63E+08	7.01E+10	1.63E+08	1.42E+10	2.25E+08
	СКО	9.02E+07	3.29E+08	9.02E+07	4.29E+10	9.04E+07	8.70E+09	4.80E+08
f12	Медиана	1.03E+03	1.24E+03	1.03E+03	7.72E+02	3.15E+03	1.44E+03	6.69E+02
	МО	1.03E+03	1.25E+03	1.05E+03	2.30E+03	3.16E+03	1.76E+03	1.23E+03
	СКО	1.76E+01	1.36E+02	2.43E+01	2.41E+03	9.30E+02	7.04E+02	4.86E+02
f13	Медиана	1.31E+08	3.40E+07	1.93E+09	8.02E+09	7.25E+07	2.04E+09	6.73E+07
	МО	1.49E+08	3.40E+07	1.96E+09	1.27E+10	8.44E+07	2.99E+09	5.52E+08
	СКО	7.64E+07	1.10E+07	1.02E+09	2.96E+09	8.34E+07	8.30E+08	2.67E+08
f14	Медиана	1.85E+08	6.22E+06	1.58E+09	1.42E+10	1.08E+08	3.22E+09	1.36E+09
	МО	1.84E+08	7.26E+06	2.79E+09	1.69E+11	3.54E+08	3.45E+10	3.46E+09
	СКО	3.22E+07	2.12E+06	3.47E+09	4.81E+10	4.46E+07	1.03E+10	7.84E+09
f15	Медиана	3.68E+06	2.21E+06	1.72E+07	2.40E+07	6.09E+07	2.16E+07	4.52E+06
	МО	6.69E+06	2.25E+06	1.69E+07	3.05E+07	6.40E+07	2.41E+07	4.75E+06
	СКО	7.49E+06	1.55E+05	1.80E+06	5.13E+06	7.03E+06	4.32E+06	3.29E+05

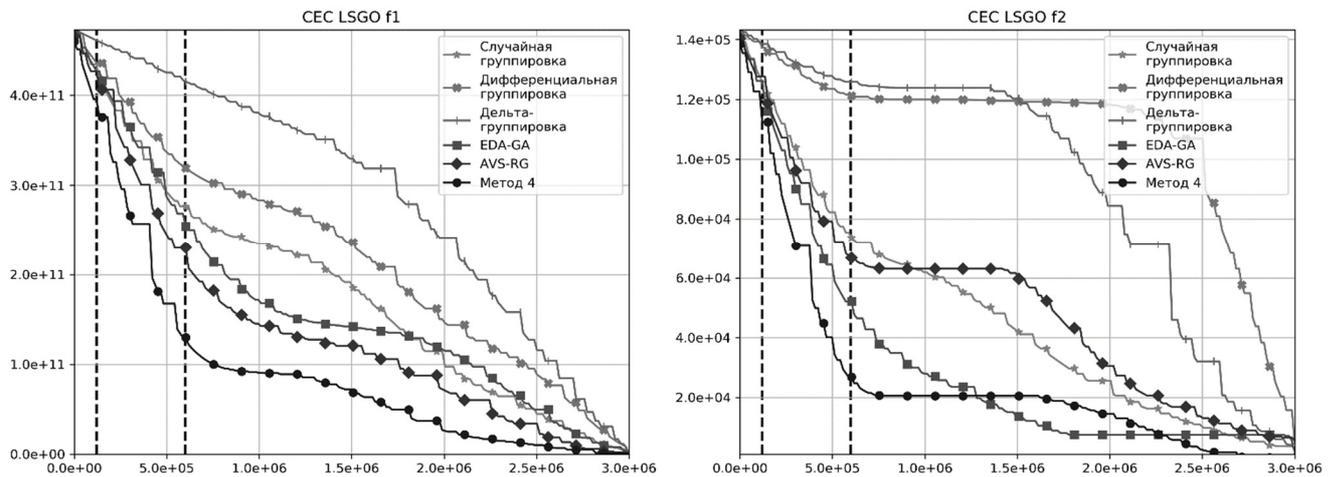


Рисунок Д.1 – График изменения медианы лучшего найденного на задачах LSGO

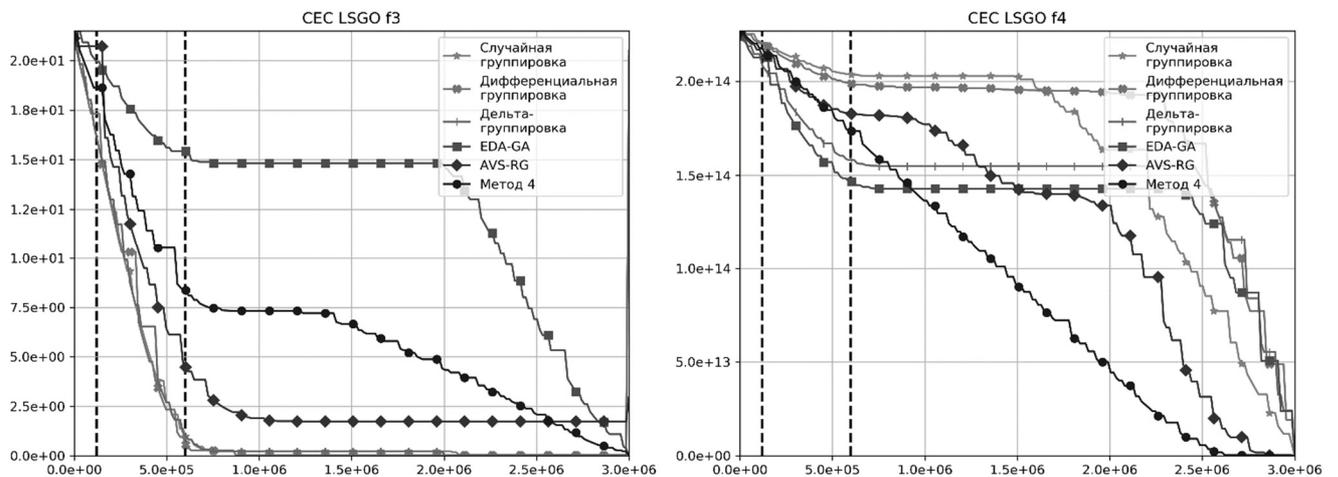


Рисунок Д.2 – График изменения медианы лучшего найденного на задачах LSGO

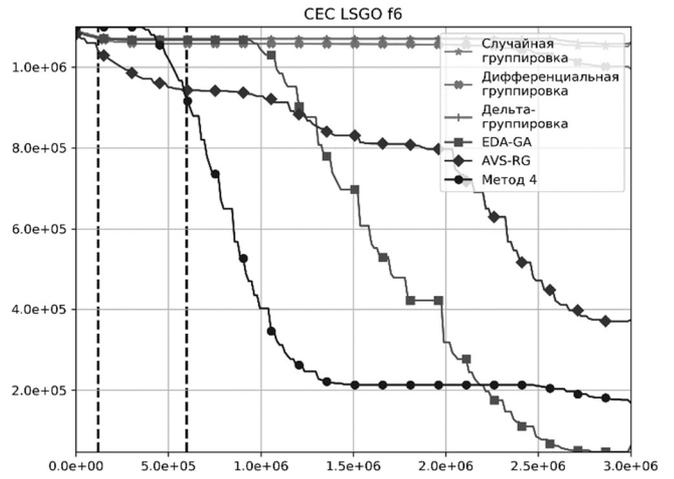
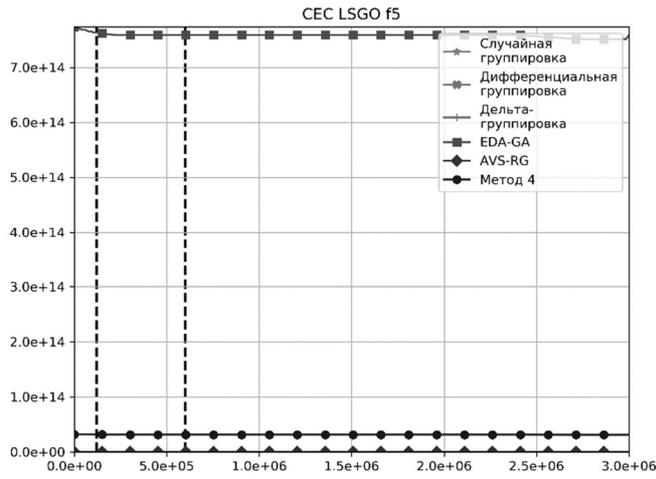


Рисунок Д.3 – График изменения медианы лучшего найденного на задачах LSGO

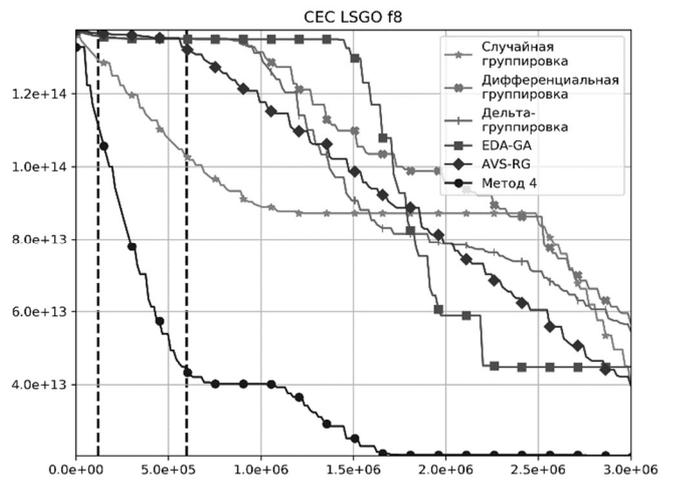
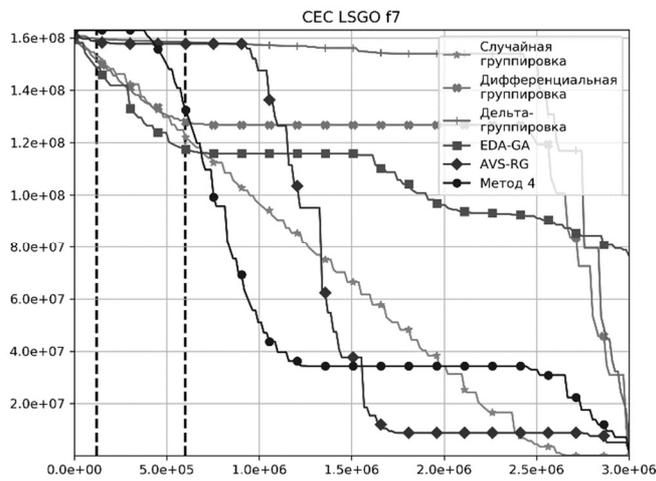


Рисунок Д.4 – График изменения медианы лучшего найденного на задачах LSGO

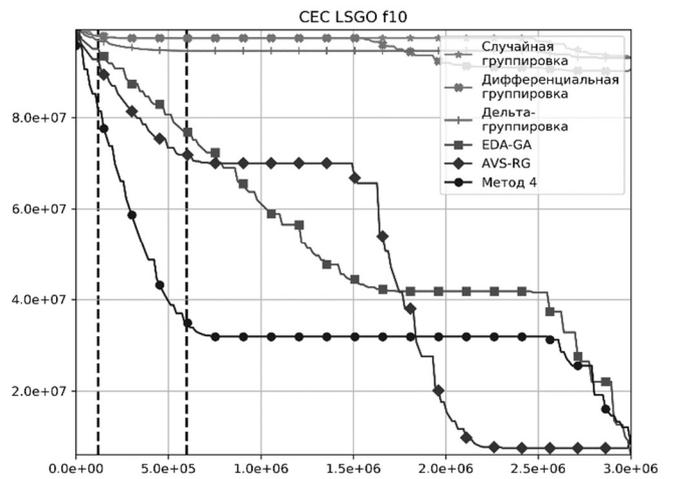
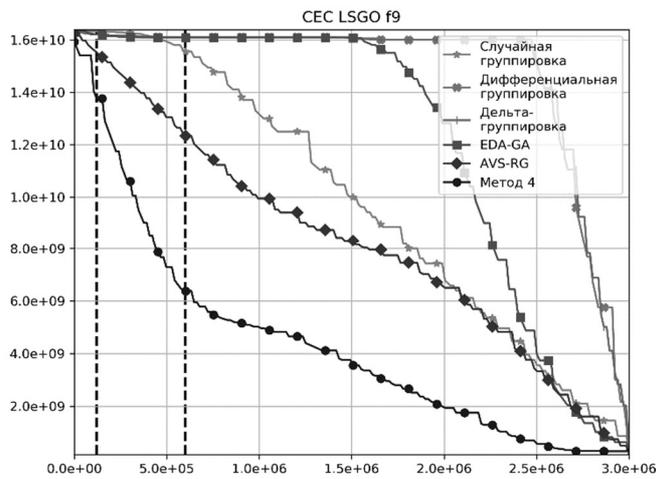


Рисунок Д.5 – График изменения медианы лучшего найденного на задачах LSGO

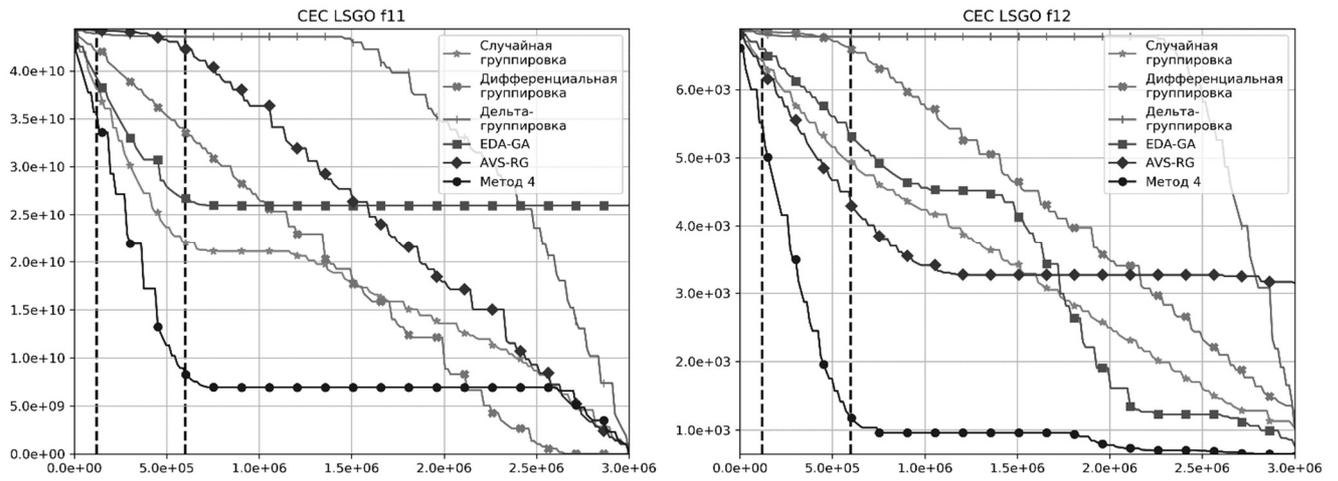


Рисунок Д.6 – График изменения медианы лучшего найденного на задачах LSGO

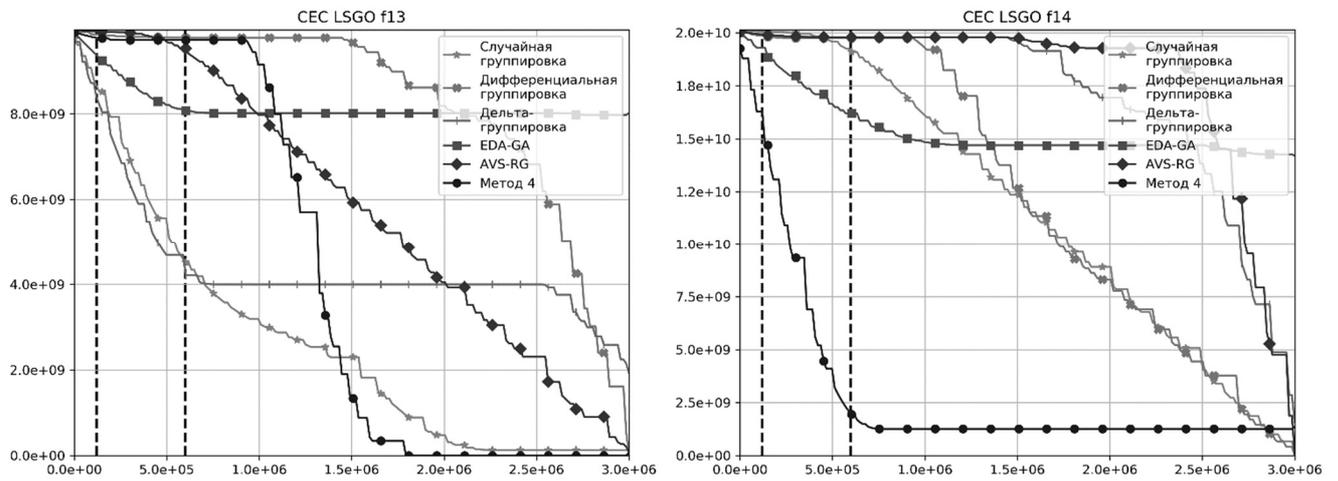


Рисунок Д.7 – График изменения медианы лучшего найденного на задачах LSGO

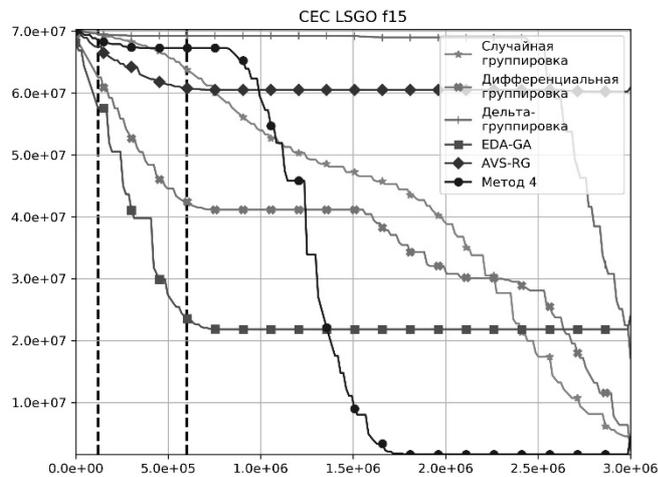


Рисунок Д.8 – График изменения медианы лучшего найденного на задачах LSGO

Таблица Д.2 – Сравнение метода 4 и алгоритмов, участвовавших в конкурсе IEEE CEC LSGO (МО – математическое ожидание, СКО – среднее квадратическое отклонение)

Задача		СС- CMA-ES	SACC	MOS	VMO-DE	INDELS	DEEPSO	DECC-G	Среднее по алго- ритмам	Метод 4
f1	Медиана	6.03E-09	0.00E+00	0.00E+00	8.51E-04	4.80E-29	2.53E+08	2.00E-13	3.61E+07	5.73E-06
	МО	5.77E-09	2.73E-24	0.00E+00	1.29E-03	4.34E-28	3.18E+08	2.03E-13	4.54E+07	3.25E-05
	СКО	1.00E-09	1.36E-23	0.00E+00	1.27E-03	1.23E-27	1.70E+08	1.78E-14	2.43E+07	3.75E-04
	Лучшее	3.46E-09	0.00E+00	0.00E+00	1.02E-04	0.00E+00	1.44E+08	1.75E-13	2.06E+07	2.11E-10
f2	Медиана	1.33E+03	5.71E+02	8.36E+02	5.51E+03	1.27E+03	1.68E+04	1.03E+03	3.91E+03	1.05E+03
	МО	1.33E+03	7.06E+02	8.32E+02	5.53E+03	1.32E+03	1.68E+04	1.03E+03	3.94E+03	2.54E+03
	СКО	1.11E+02	4.72E+02	4.48E+01	3.96E+02	6.98E+01	8.63E+02	2.26E+01	2.83E+02	2.38E+02
	Лучшее	1.09E+03	2.88E+02	7.40E+02	4.79E+03	1.27E+03	1.49E+04	9.90E+02	3.44E+03	9.30E+02
f3	Медиана	1.49E-13	1.21E+00	9.10E-13	3.41E-04	2.00E+01	2.08E+01	2.85E-10	6.00E+00	1.57E-04
	МО	1.51E-13	1.11E+00	9.17E-13	3.70E-04	2.01E+01	2.06E+01	2.87E-10	5.97E+00	4.78E-01
	СКО	6.73E-15	1.11E+00	5.12E-14	1.37E-04	1.36E-01	9.34E-02	1.38E-11	1.91E-01	4.80E-02
	Лучшее	1.39E-13	9.24E-14	8.20E-13	1.06E-04	2.00E+01	2.04E+01	2.63E-10	5.77E+00	1.48E-10
f4	Медиана	1.93E+09	3.66E+10	1.56E+08	8.48E+09	3.09E+08	7.52E+09	2.12E+10	1.09E+10	1.67E+09
	МО	2.19E+09	4.56E+10	1.74E+08	9.13E+09	3.04E+08	7.42E+09	2.60E+10	1.30E+10	6.82E+09
	СКО	1.31E+09	3.60E+10	7.87E+07	3.73E+09	1.07E+08	1.88E+09	1.47E+10	8.26E+09	4.06E+08
	Лучшее	4.51E+08	8.48E+09	1.10E+08	3.23E+09	1.03E+08	4.77E+09	7.58E+09	3.53E+09	3.28E+08
f5	Медиана	7.28E+14	6.95E+06	6.79E+06	7.28E+14	9.68E+06	1.84E+07	7.28E+14	3.12E+14	3.14E+13
	МО	7.28E+14	7.74E+06	6.94E+06	7.28E+14	9.59E+06	1.86E+07	7.28E+14	3.12E+14	5.48E+13
	СКО	4.65E+06	3.22E+06	8.85E+05	0.00E+00	2.03E+06	2.16E+06	1.51E+05	1.87E+06	1.65E+06
	Лучшее	7.28E+14	3.36E+06	5.25E+06	7.28E+14	5.88E+06	1.45E+07	7.28E+14	3.12E+14	1.44E+07
f6	Медиана	9.93E+05	2.07E+05	1.39E+05	1.99E+05	1.03E+06	1.04E+06	6.08E+04	5.24E+05	1.69E+05
	МО	5.83E+05	2.47E+05	1.48E+05	2.15E+05	1.03E+06	1.04E+06	4.85E+04	4.73E+05	3.50E+05
	СКО	4.79E+05	1.02E+05	6.43E+04	4.33E+04	1.95E+04	7.54E+03	3.98E+04	1.08E+05	9.17E+04
	Лучшее	5.61E+03	1.57E+05	1.95E+01	1.71E+05	1.00E+06	1.02E+06	6.96E-08	3.36E+05	1.39E+05
f7	Медиана	2.09E+06	1.58E+07	1.62E+04	3.44E+06	3.18E+04	3.20E+07	4.27E+08	6.86E+07	1.88E+06
	МО	7.44E+06	8.98E+07	1.62E+04	3.43E+06	3.46E+04	3.70E+07	6.07E+08	1.06E+08	2.06E+07
	СКО	1.21E+07	2.48E+08	9.10E+03	2.71E+05	1.33E+04	1.48E+07	4.09E+08	9.77E+07	1.75E+07
	Лучшее	6.66E+04	1.72E+06	3.49E+03	2.81E+06	1.33E+04	1.54E+07	1.96E+08	3.09E+07	1.86E+05
f8	Медиана	3.44E+14	9.86E+14	8.08E+12	3.26E+13	1.36E+12	1.48E+13	3.88E+14	2.54E+14	2.10E+13
	МО	3.88E+14	1.20E+15	8.00E+12	6.94E+13	1.36E+12	1.79E+13	4.26E+14	3.02E+14	4.34E+13
	СКО	2.87E+14	7.63E+14	3.07E+12	7.22E+13	6.85E+01	1.29E+13	1.53E+14	1.84E+14	2.66E+13
	Лучшее	8.48E+13	1.47E+14	3.26E+12	8.45E+12	5.36E+10	5.42E+12	1.43E+14	5.60E+13	8.94E+12
f9	Медиана	3.59E+08	5.77E+08	3.87E+08	7.51E+08	7.12E+08	1.29E+09	4.17E+08	6.42E+08	3.80E+08
	МО	3.71E+08	5.98E+08	3.83E+08	7.68E+08	6.74E+08	1.34E+09	4.27E+08	6.52E+08	4.24E+08
	СКО	1.83E+08	2.03E+08	6.29E+07	1.03E+08	1.30E+08	2.38E+08	9.89E+07	1.46E+08	1.21E+08
	Лучшее	1.29E+08	2.29E+08	2.63E+08	6.12E+08	3.27E+08	9.17E+08	2.20E+08	3.85E+08	3.03E+08
f10	Медиана	1.13E+06	2.11E+07	1.18E+06	9.91E+06	9.19E+07	9.18E+07	1.19E+07	3.27E+07	8.61E+06
	МО	7.55E+05	2.95E+07	9.02E+05	9.10E+06	9.16E+07	9.20E+07	1.10E+07	3.36E+07	2.49E+07
	СКО	5.02E+05	1.93E+07	5.07E+05	3.05E+06	9.17E+05	7.91E+07	4.00E+06	1.53E+07	9.49E+05

	Лучшее	1.28E+05	1.38E+07	5.92E+02	1.24E+06	9.05E+07	9.07E+07	9.29E+04	2.81E+07	5.29E+06
f11	Медиана	1.32E+08	5.30E+08	4.48E+07	1.58E+08	9.87E+06	7.59E+08	1.60E+11	2.31E+10	1.36E+08
	МО	1.59E+08	2.78E+09	5.22E+07	1.66E+08	1.07E+07	7.43E+08	2.46E+11	3.57E+10	2.25E+08
	СКО	1.47E+08	5.90E+09	2.05E+07	3.80E+07	4.12E+06	1.13E+08	2.03E+11	2.99E+10	4.80E+08
	Лучшее	3.29E+07	8.12E+07	2.06E+07	1.09E+08	5.59E+06	5.60E+08	4.68E+10	6.80E+09	5.29E+07
f12	Медиана	9.85E+02	8.74E+02	2.46E+02	2.34E+03	5.16E+02	3.43E+10	1.03E+03	4.90E+09	6.69E+02
	МО	1.27E+03	8.73E+02	2.47E+02	4.45E+03	3.77E+02	1.98E+10	1.04E+03	2.83E+09	1.23E+03
	СКО	4.26E+02	3.71E+02	2.54E+02	4.61E+03	3.30E+02	5.83E+09	5.76E+01	8.33E+08	4.86E+02
	Лучшее	9.77E+02	2.43E+02	2.22E-01	1.20E+03	2.63E-13	1.54E+10	9.80E+02	2.20E+09	9.46E+02
f13	Медиана	3.88E+08	1.51E+09	3.30E+06	2.43E+07	4.02E+06	1.47E+09	3.36E+10	5.29E+09	6.73E+07
	МО	6.69E+08	1.78E+09	3.40E+06	2.46E+07	3.80E+06	1.54E+09	3.42E+10	5.46E+09	5.52E+08
	СКО	1.14E+09	8.05E+08	1.06E+06	3.28E+06	9.72E+05	4.78E+08	6.41E+09	1.26E+09	2.67E+08
	Лучшее	2.18E+07	6.72E+08	1.52E+06	1.88E+07	1.90E+06	8.75E+08	2.09E+10	3.21E+09	2.16E+07
f14	Медиана	3.21E+07	7.34E+09	2.42E+07	9.35E+07	1.48E+07	1.65E+09	6.27E+11	9.09E+10	1.36E+09
	МО	7.10E+07	1.75E+10	2.56E+07	9.54E+07	1.58E+07	5.19E+09	6.08E+11	9.01E+10	3.46E+09
	СКО	1.25E+08	2.87E+10	7.94E+06	1.13E+07	5.11E+06	8.19E+09	2.06E+11	3.47E+10	7.84E+09
	Лучшее	1.90E+07	8.21E+07	1.54E+07	7.80E+07	9.41E+06	4.33E+08	1.91E+11	2.74E+10	2.86E+07
f15	Медиана	2.85E+07	1.88E+06	2.38E+06	1.11E+07	3.13E+06	9.57E+06	6.01E+07	1.67E+07	4.52E+06
	МО	3.03E+07	2.01E+06	2.35E+06	1.10E+07	2.81E+06	1.02E+07	6.05E+07	1.70E+07	4.75E+06
	СКО	6.08E+06	7.23E+05	1.94E+05	1.65E+06	1.01E+06	2.43E+06	6.45E+06	2.65E+06	3.29E+05
	Лучшее	2.31E+07	1.26E+06	2.03E+06	8.10E+06	1.41E+06	7.04E+06	4.63E+07	1.27E+07	1.47E+06

Таблица Д.3 – Исходные данные для задачи повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах

Unit	P_i^0	UR_i	DR_i	p_i^{min}	p_i^{max}	a	b	c	e	f
1	98.4	30	120	71	119	0.032888	61.242	1220.645	0	0
2	134	30	120	120	189	0.00828	41.095	1315.118	0	0
3	141.5	60	60	125	190	0.003849	46.31	874.288	0	0
4	183.3	60	60	125	190	0.003849	46.31	874.288	0	0
5	125	150	150	90	190	0.042468	54.242	1976.469	700	0.08
6	91.3	150	150	90	190	0.014992	61.215	1338.087	0	0
7	401.1	180	300	280	490	0.007039	11.791	1818.299	0	0
8	329.5	180	300	280	490	0.003079	15.055	1133.978	0	0
9	386.1	300	510	260	496	0.005063	13.226	1320.636	0	0
10	427.3	300	510	260	496	0.005063	13.226	1320.636	600	0.055
11	412.2	300	510	260	496	0.005063	13.226	1320.636	0	0
12	370.1	300	510	260	496	0.003552	14.498	1106.539	0	0
13	301.8	600	600	260	506	0.003901	14.651	1176.504	0	0
14	368	600	600	260	509	0.003901	14.651	1176.504	0	0
15	301.9	600	600	260	506	0.003901	14.651	1176.504	800	0.06
16	476.4	600	600	260	505	0.003901	14.651	1176.504	0	0
17	283.1	600	600	260	506	0.002393	15.669	1017.406	0	0
18	414.1	600	600	260	506	0.002393	15.669	1017.406	0	0
19	328	600	600	260	505	0.003684	14.656	1229.131	0	0
20	389.4	600	600	260	505	0.003684	14.656	1229.131	0	0
21	354.7	600	600	260	505	0.003684	14.656	1229.131	0	0
22	262	600	600	260	505	0.003684	14.656	1229.131	600	0.05
23	461.5	600	600	260	505	0.004004	14.378	1267.894	0	0

24	371.6	600	600	260	505	0.003684	14.656	1229.131	0	0
25	462.6	300	300	280	537	0.001619	16.261	975.926	0	0
26	379.2	300	300	280	537	0.005093	13.362	1532.093	0	0
27	530.8	360	360	280	549	0.000993	17.203	641.989	0	0
28	391.9	360	360	280	549	0.000993	17.203	641.989	0	0
29	480.1	180	180	260	501	0.002473	15.274	911.533	0	0
30	319	180	180	260	501	0.002547	15.212	910.533	0	0
31	329.5	600	600	260	506	0.003542	15.033	1074.81	0	0
32	333.8	600	600	260	506	0.003542	15.033	1074.81	0	0
33	390	600	600	260	506	0.003542	15.033	1074.81	600	0.043
34	432	600	600	260	506	0.003542	15.033	1074.81	0	0
35	402	660	660	260	500	0.003132	13.992	1278.46	0	0
36	428	900	900	260	500	0.001323	15.679	861.742	0	0
37	178.4	180	180	120	241	0.00295	16.542	408.834	0	0
38	194.1	180	180	120	241	0.00295	16.542	408.834	0	0
39	474	600	600	423	774	0.000991	16.518	1288.815	0	0
40	609.8	600	600	423	769	0.001581	15.815	1436.251	600	0.043
41	17.8	210	210	3	19	0.90236	75.464	699.988	0	0
42	6.9	366	366	3	28	0.110295	129.544	134.544	0	0
43	224.3	702	702	160	250	0.024493	56.613	3427.912	0	0
44	210	702	702	160	250	0.029156	54.451	3751.772	0	0
45	212	702	702	160	250	0.024667	54.736	3918.78	0	0
46	200.8	702	702	160	250	0.016517	58.034	3379.58	0	0
47	220	702	702	160	250	0.026584	55.981	3345.296	0	0
48	232.9	702	702	160	250	0.00754	61.52	3138.754	0	0
49	168	702	702	160	250	0.01643	58.635	3453.05	0	0
50	208.4	702	702	160	250	0.045934	44.647	5119.3	0	0
51	443.9	1350	1350	165	504	0.000044	71.584	1898.415	0	0
52	426	1350	1350	165	504	0.000044	71.584	1898.415	1100	0.043
53	434.1	1350	1350	165	504	0.000044	71.584	1898.415	0	0
54	402.5	1350	1350	165	504	0.000044	71.584	1898.415	0	0
55	357.4	1350	1350	180	471	0.002528	85.12	2473.39	0	0
56	423	720	720	180	561	0.000131	87.682	2781.705	0	0
57	220	720	720	103	341	0.010372	69.532	5515.508	0	0
58	369.4	2700	2700	198	617	0.007627	78.339	3478.3	0	0
59	273.5	1500	1500	100	312	0.012464	58.172	6240.909	0	0
60	336	1656	1656	153	471	0.039441	46.636	9960.11	0	0
61	432	2160	2160	163	500	0.007278	76.947	3671.997	0	0
62	220	900	900	95	302	0.000044	80.761	1837.383	0	0
63	410.6	1200	1200	160	511	0.000044	70.136	3108.395	0	0
64	422.7	1200	1200	160	511	0.000044	70.136	3108.395	0	0
65	351	1014	1014	196	490	0.018827	49.84	7095.484	0	0
66	296	1014	1014	196	490	0.010852	65.404	3392.732	0	0
67	411.1	1014	1014	196	490	0.018827	49.84	7095.484	0	0
68	263.2	1014	1014	196	490	0.018827	49.84	7095.484	0	0
69	370.3	1350	1350	130	432	0.03456	66.465	4288.32	0	0
70	418.7	1350	1350	130	432	0.08154	22.941	13813	1200	0.03
71	409.6	1350	1350	137	455	0.023534	64.314	4435.493	0	0
72	412	1350	1350	137	455	0.035475	45.017	9750.75	1000	0.05
73	423.2	780	780	195	541	0.000915	70.644	1042.366	0	0
74	428	1650	1650	175	536	0.000044	70.959	1159.895	0	0
75	436	1650	1650	175	540	0.000044	70.959	1159.895	0	0
76	428	1650	1650	175	538	0.001307	70.302	1303.99	0	0
77	425	1650	1650	175	540	0.000392	70.662	1156.193	0	0
78	497.2	1620	1620	330	574	0.000087	71.101	2118.968	0	0
79	510	1482	1482	160	531	0.000521	37.854	779.519	0	0
80	470	1482	1482	160	531	0.000498	37.768	829.888	0	0
81	464.1	1668	1668	200	542	0.001046	67.983	2333.69	0	0
82	118.1	120	120	56	132	0.13205	77.838	2028.954	0	0
83	141.3	180	180	115	245	0.096968	63.671	4412.017	0	0

84	132	120	180	115	245	0.054868	79.458	2982.219	1000	0.05
85	135	120	180	115	245	0.054868	79.458	2982.219	0	0
86	252	120	180	207	307	0.014382	93.966	3174.939	0	0
87	221	120	180	207	307	0.013161	94.723	3218.359	0	0
88	245.9	318	318	175	345	0.016033	66.919	3723.822	0	0
89	247.9	318	318	175	345	0.013653	68.185	3551.405	0	0
90	183.6	318	318	175	345	0.028148	60.821	4332.615	0	0
91	288	318	318	175	345	0.01347	68.551	3493.739	0	0
92	557.4	18	18	360	580	0.000064	2.842	226.799	0	0
93	529.5	18	18	415	645	0.000252	2.946	382.932	0	0
94	800.8	36	36	795	984	0.000022	3.096	156.987	0	0
95	801.5	36	36	795	978	0.000022	3.04	154.484	0	0
96	582.7	138	204	578	682	0.000203	1.709	332.834	0	0
97	680.7	144	216	615	720	0.000198	1.668	326.599	0	0
98	670.7	144	216	612	718	0.000215	1.789	345.306	0	0
99	651.7	144	216	612	720	0.000218	1.815	350.372	0	0
100	921	48	48	758	964	0.000193	2.726	370.377	0	0
101	916.8	48	48	755	958	0.000197	2.732	367.067	0	0
102	911.9	36	54	750	1007	0.000324	2.651	124.875	0	0
103	898	36	54	750	1006	0.000344	2.798	130.785	0	0
104	905	30	30	713	1013	0.00069	1.595	878.746	0	0
105	846.5	30	30	718	1020	0.00065	1.503	827.959	0	0
106	850.9	30	30	791	954	0.000233	2.425	432.007	0	0
107	843.7	30	30	786	952	0.000239	2.499	445.606	0	0
108	841.4	36	36	795	1006	0.000261	2.674	467.223	0	0
109	835.7	36	36	795	1013	0.000259	2.692	475.94	0	0
110	828.8	36	36	795	1021	0.000707	1.633	899.462	0	0
111	846	36	36	795	1015	0.000786	1.816	1000.367	0	0
112	179	120	120	94	203	0.014355	89.83	1269.132	0	0
113	120.8	120	120	94	203	0.014355	89.83	1269.132	0	0
114	121	120	120	94	203	0.014355	89.83	1269.132	0	0
115	317.4	480	480	244	379	0.030266	64.125	4965.124	0	0
116	318.4	480	480	244	379	0.030266	64.125	4965.124	0	0
117	335.8	480	480	244	379	0.030266	64.125	4965.124	0	0
118	151	240	240	95	190	0.024027	76.129	2243.185	0	0
119	129.5	240	240	95	189	0.00158	81.805	2290.381	600	0.07
120	130	120	120	116	194	0.022095	81.14	1681.533	0	0
121	218.9	180	180	175	321	0.07681	46.665	6743.302	1200	0.043
122	5.4	90	90	2	19	0.953443	78.412	394.398	0	0
123	45	90	90	4	59	0.000044	112.088	1243.165	0	0
124	20	300	300	15	83	0.072468	90.871	1454.74	0	0
125	16.3	162	162	9	53	0.000448	97.116	1011.051	0	0
126	20	114	114	12	37	0.599112	83.244	909.269	0	0
127	22.1	120	120	10	34	0.244706	95.665	689.378	0	0
128	125	1080	1080	112	373	0.000042	91.202	1443.792	0	0
129	10	60	60	4	20	0.085145	104.501	535.553	0	0
130	13	66	66	5	38	0.524718	83.015	617.734	0	0
131	7.5	12	6	5	19	0.176515	127.795	90.966	0	0
132	53.2	300	300	50	98	0.063414	77.929	974.447	0	0
133	6.4	6	6	5	10	2.740485	92.779	263.81	0	0
134	69.1	60	60	42	74	0.112438	80.95	1335.594	0	0
135	49.9	60	60	42	74	0.041529	89.073	1033.871	0	0
136	91	528	528	41	105	0.000911	161.288	1391.325	0	0
137	41	300	300	17	51	0.005245	161.829	4477.11	0	0
138	13.7	18	30	7	19	0.234787	84.972	57.794	0	0
139	7.4	18	30	7	19	0.234787	84.972	57.794	0	0
140	28.6	72	120	26	40	1.111878	16.087	1258.437	0	0

Таблица Д.4 – Исходные данные для задачи повышения энергоэффективности диспетчеризации электроэнергии в распределенных энергетических системах

Unit	P_1^{lowPOZ}	$P_1^{upperPOZ}$	P_2^{lowPOZ}	$P_2^{upperPOZ}$	P_3^{lowPOZ}	$P_3^{upperPOZ}$
8	250	280	305	335	420	450
32	220	250	320	350	390	420
74	230	255	365	395	430	455
136	50	75	85	95	0	0

ПРИЛОЖЕНИЕ Е

АКТЫ ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ



МЫ СОГРЕВАЕМ ГОРОДА
**СИБИРСКАЯ
ГЕНЕРИРУЮЩАЯ
КОМПАНИЯ**
НАЗАРОВСКАЯ ГРЭС

Акционерное общество «Назаровская ГРЭС»

Российская Федерация, 662204, Красноярский край, г. Назарово, в/я 42, АО «Назаровская ГРЭС», тел. (39155) 4-49-60, факс (39155) 5-11-84; e-mail: ngres_office@sibgenco.ru; ИНН 2460237901; КПП 246001001; р/с 40702810900340000709 Филиал "Газпромбанк" (Акционерное общество) "Восточно-Сибирский"; к/с 30101810100000000877; БИК 040407877

№ _____

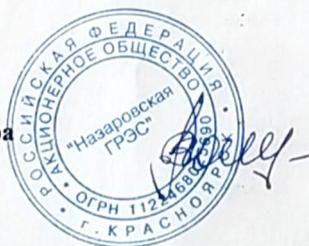
АКТ

о использовании результатов диссертационного исследования Сопова Е.А.

Настоящим актом подтверждается, что предложенная Соповым Евгением Александровичем селективная онлайн гиперэвристика синтеза эволюционных алгоритмов для задач глобальной оптимизации большой размерности, используется специалистами производственно-технического отдела АО «Назаровская ГРЭС» при решении задач проектирования распределенных децентрализованных энергосистем предприятия для оценки возможной энергоэффективности диспетчеризации электрической и тепловой энергии.

Применение методики, предложенной в рамках диссертационного исследования соискателя ученой степени доктора технических наук Сопова Евгения Александровича, позволяет снизить временные и трудозатраты на формирование и оценку проектов.

Заместитель главного инженера
по эксплуатации и наладке



В.М. Полухин

АКТ
о использовании результатов диссертационного исследования
Сопова Евгения Александровича

Настоящим актом подтверждается, что предложенный Соповым Е.А. метод синтеза базовых операций эволюционного алгоритма, использующий конструктивную офлайн гиперэвристику на основе алгоритма генетического программирования, был использован АО «Испытательный технический центр НПО-ПМ» при проектировании алгоритмического и программного обеспечения для системы автоматизированного формирования и контроля специальных партий электрорадиоизделий космического применения.

Применение новых подходов, предложенных в рамках диссертационного исследования соискателя ученой степени доктора технических наук Сопова Евгения Александровича, позволило автоматизировать процесс формирования эффективных алгоритмов оценки степени однородности партий под конкретный тип электрорадиоизделий. Предложенный метод синтеза алгоритма повышает обоснованность решения. В свою очередь, создание узкоспециализированного алгоритма под конкретную номенклатуру электрорадиоизделий позволяет повысить точность решения по сравнению с применением известных универсальных алгоритмов.



Директор АО ИТЦ - НПО ПМ", к.т.н.

В.И. Орлов

ООО «Инвент»
ИНН/ КПП: 2465251770/246201001
660037, Красноярск, ул. Мичурина, д.2д, оф.241
Тел. (391) 2935560



АКТ
о внедрении результатов диссертационного исследования
Сопова Евгения Александровича

Настоящим актом подтверждается, что разработанные Соповым Е.А. подходы для решения оптимизационных задач выбора эффективных вариантов управления инвестиционным портфелем и их программная реализация используется в деятельности Общества с ограниченной ответственностью «Инвент» при осуществлении конкурсного управления. Применение нового метода проектирования алгоритма оптимизации на основе эволюционных алгоритмов, предложенного в рамках диссертационного исследования соискателя ученой степени доктора технических наук Сопова Евгения Александровича, позволило автоматизировать процесс формирования оптимального портфеля инвестиций, а возможность получить множество эффективных способов распределения инвестиций существенно повышает обоснованность и устойчивость реализации решения, особенно для больших массивов исходных данных проектов.

Директор ООО «Инвент»
доцент, к.э.н. Карелин О.И.