

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

ЗАРУБА ДАРЬЯ ВИКТОРОВНА

**«Биоинспирированные методы и алгоритмы разбиения схем при
автоматизированном проектировании СБИС»**

Специальность: 2.3.7 – Компьютерное моделирование и автоматизация
проектирования

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
д.т.н., проф. В.В. Курейчик

Таганрог – 2023

ВВЕДЕНИЕ	4
1 АНАЛИЗ ПРОБЛЕМ И ПЕРСПЕКТИВ МЕТОДОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС	14
1.1 Анализ состояния проблемы разбиения схем при проектировании СБИС	14
1.2 Построение математической модели СБИС на основе гиперграфа	17
1.3 Постановка задачи разбиения схемы при проектировании СБИС	24
1.4 Классификация и анализ алгоритмов разбиения схем при проектировании СБИС	32
1.5 Краткие выводы	41
2 РАЗРАБОТКА МОДИФИЦИРОВАННЫХ БИОИНСПИРИРОВАННЫХ МЕТОДОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС	43
2.1 Общие положения теории биоинспирированного поиска	43
2.2 Построение комбинированной архитектуры биоинспирированного поиска	47
2.3 Разработка модифицированного метода оптимизации на основе моделирования перемещения колонии бактерий	50
2.4 Разработка модифицированного метода оптимизации на основе поведения роя светлячков	55
2.5 Разработка модифицированного метода генетического поиска	63
2.6. Построение модифицированных генетических операторов, ориентированных на решение задачи разбиения при проектировании СБИС	68
2.7 Краткие выводы	74
3. РАЗРАБОТКА МОДИФИЦИРОВАННЫХ АЛГОРИТМОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС	76
3.1 Разработка унифицированного механизма представления решений при разбиении схем	76
3.2 Разработка модифицированного бактериального алгоритма разбиения схем при проектировании СБИС	80
3.3 Разработка модифицированного светлячкового алгоритма разбиения схем при проектировании СБИС	87
3.4 Разработка модифицированного генетического алгоритма разбиения при проектировании СБИС	92
3.5 Разработка нового многоуровневого алгоритма разбиения схем при проектировании СБИС	95
3.6 Краткие выводы	100
4 ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ И АНАЛИЗ ЭФФЕКТИВНОСТИ РАЗРАБОТАННЫХ МЕТОДОВ И СРЕДСТВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС	102
4.1	102

4.2	105
4.3	107
4.4	108
4.5	115
4.6 Краткие выводы	129
Заключение	131
Список литературы	133
Приложение №1	146
Приложение №2	149

ВВЕДЕНИЕ

В современном обществе важнейшим следствием информационной революции стал особый статус и роль информации. Именно по этой причине современное общество называют «информационным» [1]. На сегодняшний день в области информатики и вычислительной техники возникают все новые проблемы, связанные с резкой интенсификацией информационных процессов. К одной из областей информатики относятся системы автоматизированного проектирования (САПР) [2-4]. На текущем этапе развития науки и техники САПР микроэлектроники представляют собой движущую силу прогресса в различных областях.

Быстрые темпы развития технологий производства сверхбольших интегральных схем (СБИС) стали одной из основных причин увеличения уровня эффективности и повышения качества характеристик при проектировании и изготовлении электронных вычислительных средств (ЭВС). Это оказало влияние на вес и габариты ЭВС, потребляемую и рассеиваемую мощности, показатели быстродействия и уровень надежности данного ЭВС [5].

Сферы применения современных ЭВС достаточно обширны, поэтому конструирование устройств данного класса на основе СБИС становится сложной и трудоемкой задачей. Такая широкая область применения ЭВС вызвала необходимость создания большого количества видов СБИС с различными параметрами, характеристиками, областями применения и эксплуатации. Таким образом, затруднительно учесть все требования и ограничения, предъявляемые к СБИС, при сохранении традиционных методов и средств разработки ЭВС [6].

Быстрые темпы развития микроэлектроники вызвал стремительный рост степени интеграции СБИС, что способствовало увеличению количества конструктивно законченных элементов (узлов) и блоков в рамках одного ЭВС. С учетом современных реалий закон Гордона Мура формулируется следующим образом: общее число транзисторов, располагаемых на поверхности кристалла СБИС, увеличивается в 2 раза каждые 18 месяцев [7]. Помимо этого, увеличение

количества транзисторов на кристалле ведет к увеличению средней площади кристалла, что также ведет к повышению трудоемкости их проектирования. Таким образом, проектирование СБИС является сложной и трудоемкой задачей.

Такое быстрое развитие производства СБИС было бы невозможно без применения ЭВМ на всех стадиях проектирования и изготовления документации, а также автоматизированного проектирования. Сокращение временных затрат при разработке СБИС достигается путем полной или частичной автоматизации выполнения проектных процедур, предусмотренных на каждом этапе проектирования, за счет использования САПР [2, 3].

Одной из наиболее значимых стадий при проектировании СБИС можно считать стадию конструкторского проектирования. Здесь реализуется интерпретация схемного представления элементов в виде их геометрического образа. Исходя из данных статистики, можно сделать вывод, что конструкторское проектирование занимает около 40% времени от принятия решения о разработке СБИС до выпуска изделия в серийное производство. Из этого следует, что сокращение временных затрат на проектирование СБИС является важным аспектом для ускорения научно-технического прогресса [8].

На этапе конструкторского проектирования выполняются такие проектные процедуры, как разбиение СБИС, планирование кристалла, размещение элементов на кристалле, трассировка соединений, упаковка и верификация. Реализация данных проектных процедур относится к задачам комбинаторно-логического типа, которые принадлежат к классу NP-полных задач, поиск точного решения которых возможно выполнить только полным перебором [9].

Поскольку современные СБИС могут содержать более 10^8 транзисторов на кристалле, то проектирование топологии всей схемы не представляется возможным, поэтому схему целесообразно разбивать на более мелкие части. В связи с повышением требований к миниатюризации и быстродействию СБИС и внедрением новых тенденций и технологий их изготовления, существующие методы, нацеленные на проектирование СБИС не могут в полной мере обеспечить эффективное и качественное решение данной задачи даже при условии высоких

темпов развития области информационных технологий. Поэтому для решения задач этапа конструкторского проектирования возникает необходимость разработки новых методов и средств на основе стохастических поисковых эвристик [10, 11].

В 1980-х гг. получили широкое распространение оптимизационные алгоритмы, инспирированные живой природой. При решении задач конструкторского проектирования к алгоритмам, представляющим особый интерес, можно отнести алгоритм роя частиц, алгоритмы, инспирированные поведением колонии муравьев при поиске пищи, роя пчел или светлячков, алгоритм бактериального поиска и др. Эффективность данных алгоритмов часто превосходит эффективность классических эволюционных и генетических алгоритмов [12].

Данный класс оптимизационных методов позволяет обрабатывать одновременно множество решений при учете нескольких критериев, что способствует их широкому применению в различных областях науки и техники, в том числе для создания современных САПР, решающих задачи конструкторского проектирования СБИС [2, 6].

На современном этапе развития САПР и биоинспирированных методов поиска наиболее весомый вклад в развитие данных областей внесли такие выдающиеся Российские и зарубежные ученые, как А.М. Бершадский, И.П. Норенков, Г.Г. Казеннов, Ю.Х. Вермишев, Л.С. Берштейн, В.М. Курейчик, П.И. Соснин, А.Л. Стемпковский, Д. Гольдберг, Д. Холланд, А. П. Карпенко, К.М. Пассино, К.Н. Кришнананд и др.

Одним из подходов к модернизации решений является многоуровневая оптимизация, основанная на алгоритмах, инспирированных живой природой, эффективность которых, как известно из литературных источников, превосходит классические алгоритмы. Поэтому с целью повышения качества проектирования и сокращении общей трудоемкости процесса и временных затрат **актуально и важно** создавать и разрабатывать новые методы компьютерного моделирования для разбиения схем на этапе конструкторского проектирования СБИС.

Цель диссертации состоит в повышение качества и эффективности автоматизированного разбиения схем с помощью методов и алгоритмов, инспирированных природными системами.

Достижение цели диссертации предусматривает решение ряда **задач**:

1. Разработать комбинированную архитектуру и построить многоуровневый алгоритм разбиения схем на основе поисковых эвристик, инспирированных природными системами.
2. Разработать унифицированный механизм кодирования и декодирования данных.
3. Разработать модифицированные бактериальный, светлячковый и генетические алгоритмы.
4. Построить модифицированные операторы генетического поиска.
5. Создать программно-алгоритмический комплекс для автоматизированного разбиения схем.

Объект исследования – СБИС.

Предмет исследования – модели, методы и средства поддержки автоматизированного разбиения схем.

Методы исследования. Для решения поставленных задач применяются теория графов, методы и алгоритмы математического и компьютерного моделирования, оптимизационные методы, инспирированные природными системами, и принципы и методы конструкторского проектирования. В рамках диссертационного исследования автор провел компьютерное моделирование предложенных методов и алгоритмов и провел вычислительный эксперимент на основе разработанного программного комплекса.

Научная новизна работы заключается в разработке методов и алгоритмов для эффективного решении задачи разбиения схем при проектировании СБИС. В работе:

1. **Построена комбинированная архитектура поиска** и на ее основе создан **многоуровневый алгоритм** разбиения схем, отличающийся применением методов биоинспирированной оптимизации (бактериального, светлячкового,

генетического). Данный позволяет получать наборы эффективных решений и избегать заикливания в локальных оптимумах (пп. 4, 6 паспорта специальности 2.3.7), 47–50, 95-100 стр. диссертации.

2. Разработаны модифицированные методы оптимизации, инспирированные природными системами (бактериальный, светлячковый и генетический), отличающиеся возможностью создания динамической области принятия решений для каждого агента. Данные методы позволяют выходить из локальных оптимумов. (п. 4 паспорта специальности 2.3.7), 50-68 стр. диссертации.

3. Разработаны модифицированные генетические операторы на основе закона распределения Больцмана и прямоугольных чисел, позволяющие сократить время поиска (пп. 4, 6 паспорта специальности 2.3.7), 68-74 стр. диссертации.

4. Предложен унифицированный механизм кодирования и декодирования решений при разбиении схем, отличающийся организацией единого подхода к представлению данных, что позволяет исключать потери процессорного времени при их передаче на всех стадиях работы многоуровневого алгоритма (п. 6 паспорта специальности 2.3.7), 76-80 стр. диссертации.

5. Разработаны модифицированные бактериальный, светлячковый и генетический алгоритмы разбиения схем, позволяющие выходить из локальных оптимумов и получать наборы эффективных решений за счет использования новых процедур и механизмов поиска (пп. 4, 6 паспорта специальности 2.3.7), 80-95 стр. диссертации.

Основные положения, выносимые на защиту. На защиту выносятся основные тезисы проведенного научного исследования:

1. Комбинированная архитектура биоинспирированного поиска решений.
2. Модифицированные методы бактериальной, светлячковой и генетической оптимизации.
3. Унифицированный механизм кодирования и декодирования решений.

4. Модифицированные бактериальный, светлячковый и генетический алгоритмы, ориентированные выполнение проектной процедуры разбиения схем.

5. Многоуровневый алгоритм разбиения схем.

Практическая ценность полученных результатов состоит в разработке программного комплекса и проведении компьютерного моделирования предложенных методов и алгоритмов разбиения при проектировании СБИС. Это позволило оценить качество полученных результатов и подтвердить преимущество разработанных методов, инспирированных природными системами.

Реализация и внедрение результатов работы.

Теоретические и практические результаты диссертационной работы внедрены в учебный процесс Института компьютерных технологий и информационной безопасности Южного федерального университета, в ООО «Научно-исследовательская лаборатория автоматизации проектирования RealLab», а также использовались в четырёх НИР, которые были направлены на решение оптимизационных задач конструкторского проектирования. Наиболее значимые из них приведены ниже:

1. Грант РФФИ №19-01-00059 «Разработка биоэвристик для создания интеллектуальной подсистемы принятия эффективных решений NP- трудных и NP-сложных комбинаторно-логических задач на графах».

2. Грант РФФИ № 16-01-00586 «Разработка инспирированных природой методов и принципов поиска оптимальных решений в задачах проектирования и управления».

3. Гранта РФФИ № 13-01-00371 «Разработка теории и принципов построения интеллектуальных систем проектирования для решения задач разбиения СБИС на основе биоинспирированных методов».

4. ГБ 213.01–11/2014-56ПЧ «Разработка теории и основных принципов эволюционных вычислений для поддержки принятия оптимальных решений при проектировании многоцелевых интеллектуальных систем».

Апробация. Основные научные и практические результаты работы докладывались и обсуждались на 15 конференциях различного уровня:

IS&IT(2015-2020гг.) (пос. Дивноморское,), SCM'2018, SCM'2019 (г. Санкт-Петербург, 2018, 2019гг.), RusAutoCon'2019, RusAutoCon'2020 (г. Сочи, 2019, 2020 г.), ИТСАиУ (2019 г.), IEEE SED-2019) (г. Прага, 2019, IV ФиПАКТиИБ'18, IEEE AICT'16 (г. Баку); IEEE EWDTS (2015, 2020 г.);.

Публикации. По тематике диссертационного исследования опубликовано 28 научных работ, из них 6 статей в журналах из перечня ВАК, 12 статей в Scopus и WoS, 1 учебно-методическое пособие и 2 программы для ЭВМ с официальной государственной регистрацией.

Личный вклад. Автором была разработана математическая модель схемы на основе гиперграфа и целевой функции, учитывающей два критерия оптимальности: количество разрезов цепей и задержки сигнала в схеме. Построена комбинированная поисковая архитектура, в основе которой лежит выполнение биоинспирированных эвристик. Разработан ряд модифицированных биоинспирированных алгоритмов для реализации многоуровневого поиска. Кроме того, был разработан программный комплекс для компьютерного моделирования проектной процедуры автоматизированного разбиения схем.

Структура и объем работы. Диссертация состоит из: введения, четырех разделов, заключения, изложенных на 132 страницах, содержит 72 рисунка, 13 таблиц, 122 наименования библиографии и приложения. Полный объем диссертационной работы составляет 156 страниц.

Во введении диссертационной работы поставлена цель работы, проведен анализ текущего состояния рассматриваемой проблемы, на основе которого была обоснована актуальность проведения исследований по данной тематике. Помимо этого, были приведены положения, выносимые на защиту, сформулирована научная новизна диссертационного исследования и практическая значимость полученных результатов, а также приведены сведения об апробации результатов проведенной работы на различных научных конференциях, их реализация и внедрение. Помимо этого, во введение описано краткое содержание всех разделов диссертационной работы.

В первом разделе диссертационного исследования проведен анализ современных проблем разработки и производства СБИС, на основе которого были выбраны математическая модель, позволяющая адекватно отобразить моделируемую схему. Представлена формализованная постановка задачи разбиения СБИС, а также выбран способ расчета оценки качества альтернативных решений. Построена математическая модель СБИС на основе гиперграфа с учетом двух критериев (количества разрезов цепей и временной задержки сигнала в схеме), объединение которых было произведено методом аддитивной свертки. В диссертационной работе приведены классификация и сравнительный анализ современных алгоритмов, позволяющих разбивать графы на части. Выявлены их достоинства и недостатки. На основе проведенного анализа была обоснована целесообразность применения методов биоинспирированной оптимизации для решения задачи, поставленной в диссертационной работе.

Во втором разделе диссертационного исследования рассмотрены общие положения теории биоинспирированного поиска и основные характеристики и свойства популяционных алгоритмов. Предложена новая комбинированная архитектура биоинспирированного поиска, представляющая собой двухуровневую систему. Здесь на первом уровне выполняется поиск роевыми методами оптимизации на основе моделирования перемещения колонии бактерий и поведения роя светлячков. Поиск на втором уровне реализуется за счет модифицированных генетических эвристик для улучшения качества альтернативных решений, полученных на предыдущем этапе поиска. В основе модификации метода генетического поиска лежит реализация модифицированных операторов, основанных на теории прямоугольных чисел и законе распределения Больцмана. Применение данных операторов позволяет сократить пространство поиска и уменьшить время работы алгоритма.

В третьем разделе диссертационной работы разработан новый многоуровневый алгоритм разбиения СБИС. Разработанный многоуровневый алгоритм обеспечивает последовательное и последовательно-параллельное выполнение модифицированных бактериального, светлячкового и генетического

алгоритмов в зависимости от значения параметра шкалы выбора, что позволяет эффективным образом распределять вычислительные ресурсы при разбиении СБИС. Для исключения потерь времени при передаче данных между этапами работы предложенного алгоритма создан унифицированный механизм представления данных. Данный механизм кодирования и декодирования альтернативных решений позволяет использовать единый формат данных при выполнении всех разработанных алгоритмов. Также разработаны модифицированные бактериальный алгоритм, основанный на процедурах хемотаксиса, ликвидации и рассеивания; модифицированный светлячковый алгоритм, реализующий обновленный механизм выбора пары агентов для скрещивания и модифицированный генетический алгоритм, включающий выполнение модифицированных генетических операторов. Данные алгоритмы обеспечивают получение наборов квазиоптимальных решений в приемлемые сроки.

В четвертом разделе проведена теоретическая оценка временных и пространственных характеристик разработанного многоуровневого алгоритма разбиения СБИС. Поставлена цель экспериментальных исследований. Приведено описание программного комплекса, реализующего компьютерные модели разработанных методов и алгоритмов разбиения СБИС. Для демонстрации эффективности разработанных методов и алгоритмов проведен ряд вычислительных экспериментов при различных значениях весовых коэффициентов целевой функции, которые позволяют оценить степень значимости частных критериев. В результате проведения вычислительных экспериментов получены зависимости времени и качества решений при сравнении работы многоуровневого алгоритма и алгоритмов разбиения MLPart и hMetis на тестовых примерах (бенчмарках). Результаты, полученные в ходе выполнения вычислительного эксперимента, продемонстрировали преимущества разработанного подхода к реализации проектной процедуры разбиения СБИС по сравнению с существующими аналогами.

В заключении сформулированы основные выводы и приведены результаты, полученные в ходе выполнения диссертационного исследования.

В приложение приведены копии актов внедрения и свидетельства об официальной регистрации программы для ЭВМ.

1 АНАЛИЗ ПРОБЛЕМ И ПЕРСПЕКТИВ МЕТОДОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС

1.1 Анализ состояния проблемы разбиения схем при проектировании СБИС

При проектировании ЭВС на этапе конструкторского проектирования наиболее трудоемкими считаются задачи разбиения СБИС, размещения элементов на кристалле и трассировка межсоединений. В современном производстве технологии изготовления ЭВС находятся на достаточно высоком уровне, но при этом требования к быстродействию и размерам проектируемых устройств постоянно повышаются. Таким образом, актуальность эффективного решения задачи разбиения СБИС, как исходной задачи на уровне конструкторского проектирования, возрастает с каждой годом [8].

С точки зрения блочно-иерархического подхода к проектированию технических объектов можно выделить следующие уровни [13, 14].

- Системный уровень. На данном уровне определяется назначение объекта и взаимосвязь его с другими объектами, учитывая при этом изменения, вносимые в окружающую среду.
- Структурный уровень. На данном уровне происходит структурное описание объекта.
- Функциональный уровень. На данном уровне описываются основные принципы работы всех подсистем объекта, а также обеспечивается работоспособность объекта в целом;
- Конструктивный или элементный уровень. На данном уровне отбираются и описываются элементы всей системы.

Как известно, при разработке САПР возможно применение различных подходов к декомпозиции всего процесса проектирования. Проектирование технического объекта можно разделить на несколько этапов, последовательность которых представлена на рисунке 1.1 [14]:



Рисунок 1.1 – Уровни проектирования СБИС

На стадии системного проектирования вся система рассматривается как единое целое, а также производится оценка и выбор входящих в систему компонентов, которые обусловлены архитектурными требованиями и удовлетворяют накладываемым ограничениям.

На уровне функционального проектирования разрабатываются функциональные и логические схемы устройств. Здесь выделяют регистровый и логический подуровни. На регистровом подуровне устройства проектируются из блоков (блоки типа регистров, дешифраторов, счетчиков и логических преобразователей). На логическом подуровне происходит проектирование устройства или составляющих его блоков из отдельных логических элементов, например вентилях и триггеров.

На схемотехническом уровне происходит проектирование принципиальных схем электронных объектов. На данном уровне элементами являются компоненты

схем, такие как резисторы, транзисторы, и др.

На компонентном уровне разрабатываются компоненты устройств, которые рассматриваются как системы, состоящие из элементов [14].

Качество производства объекта во многом зависит от результатов, полученных на этапе конструкторского проектирования, который играет немало важную роль в САПР СБИС. В последние годы технологии производства развиваются очень быстрыми темпами, поэтому автоматизация этапа конструкторского проектирования становится актуальной задачей. В условиях непрерывного развития технологий происходит не только разработка передовых эвристик поиска оптимальных решений, но и предъявляются новые требования к конструкции разрабатываемого объекта [15].

На этапе конструкторского проектирования СБИС решаются следующие основные задачи: разбиение схем, размещение элементов на монтажной поверхности, трассировка соединений, сжатие топологии и верификация. На текущем этапе развития технологий производства СБИС является актуальной разработка новых методов решения данных задач для повышения качества результатов, полученных на этапе конструкторского проектирования.

От качества выполнения проектной процедуры разбиения во многом зависит эффективность всего проектирования СБИС и качество результатов, получаемых на выходе [8].

На современном этапе развития технологии проектирования СБИС на стадии конструкторского проектирования происходит обработка больших массивов данных, поэтому решение задачи разбиения схем при проектировании СБИС играет важную роль. Т. к. задача разбиения является NP-полной, то разработка новых унифицированных методов и алгоритмов оптимизации для реализации проектной процедуры разбиения СБИС, позволяющих находить эффективные решения за приемлемое время, достаточно трудоемко. Создание новых методов и средств проектирования и, как следствие, увеличение степени интеграции СБИС является одной из основных причин для разработки новых современных технологий решения задач автоматизированного проектирования и, в частности,

задачи разбиения схем [14].

Задачу разбиения СБИС можно отнести к классу задач непрерывной глобальной оптимизации [15]. Задачи данного класса обладают рядом особенностей, которые приведены на рисунке 1.2.

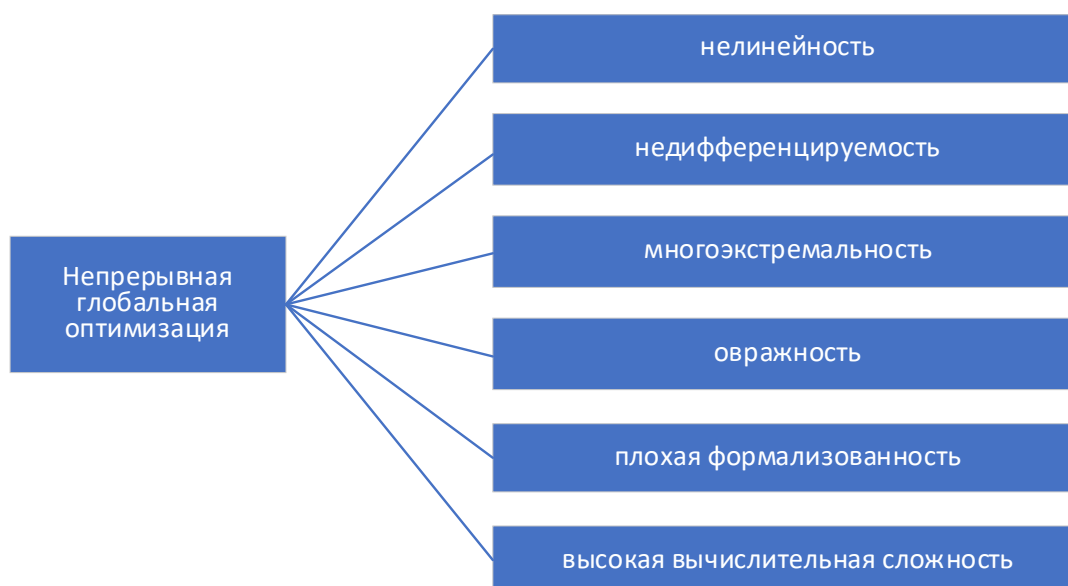


Рисунок 1.2 – Особенности задач непрерывной глобальной оптимизации

Таким образом, следует отметить, что для реализации проектной процедуры разбиения схем при проектировании СБИС было разработано большое количество методов, но не существует универсального подхода, отвечающего требованиям современного процесса проектирования.

1.2 Построение математической модели СБИС на основе гиперграфа

Как известно, в процессе проектирования СБИС одну из основных ролей играет метод представления данных, определяющий насколько соответствует данная математическая модель (ММ) реальному техническому объекту. Помимо этого, при разработке устройства необходимо учитывать объем машинной памяти, требуемый для представления модели, удобство проектирования, трудозатраты на оценку полученных результатов и т.д. При разработке ММ требуется определить

данные, которые должна отражать ММ, задать ограничения к модели и сформулировать правила перехода от объекта проектирования к ММ [17, 18].

В данной диссертационной работе объектами проектирования являются СБИС. К ММ предъявляются требования [18], представленный на рисунке 1.3.

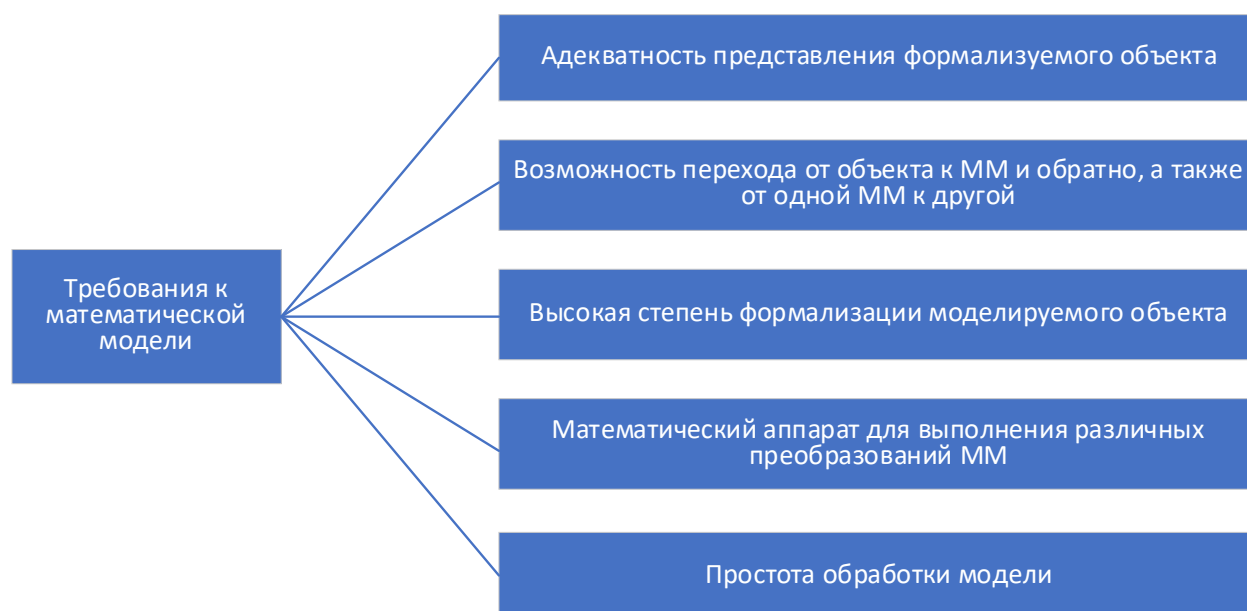


Рисунок 1.3 – Требования к ММ

Теория графов, важнейшая частей современной дискретной математики, является одним из наиболее эффективных способов для формализации решений различных задач оптимизации. Аппарат теории графов часто применяется при моделировании в различных областях науки и техники, таких, как теоретическая физика, химия, теория игр, комбинаторная оптимизация, биология, социология, экономика, лингвистика и т.д. [19].

В последнее время особенно возросла практическая роль графов, что связано с активным развитием следующих направлений [20]:

- автоматизированные системы планирования, проектирования и управления;
- представление данных, знаний и методов их поиска при решении задач искусственного интеллекта;
- разработка и создание интеллектуальных систем обработки данных и т.д.

Использование графовых моделей позволяет значительно уменьшить количество вычислительных процедур по сравнению с аналогичными способами представления данных и разрабатывать эффективные алгоритмы оптимизации при сохранении наглядности описания технических объектов.

При разработке устройства работа проектировщика упрощается, благодаря использованию легко-формализуемых моделей, где элементы моделируются с помощью вершин графа, а связи между ними – с помощью ребер. Такой способ представления технического объекта характеризуется высокой степенью наглядности и позволяет разработчику сконцентрироваться на наиболее существенных проблемах, что способствует нахождению качественных решений при проектировании. Описанный способ представления моделей СБИС называется графовым. Здесь технический объект задается в виде двух подмножеств (вершины и ребра), находящихся в некотором отношении между собой [21].

Для формализации технического объекта при решении задач этапа конструкторского проектирования целесообразно применять аппарат теории графов, который позволяет получать адекватную модель устройства с точки зрения полноты информации и корректности ее отображения.

Корректность отображения информации достигается путем соблюдения правил перехода от проектируемого объекта к ММ. Правилами перехода задается соответствие между компонентами объекта проектирования и элементами ММ. При проектировании СБИС основным вопросом является способ моделирования электрической цепи [22].

Использование в качестве ММ проектируемого объекта граф определенного вида позволяет:

- получать адекватные ММ с точки зрения полноты и корректности представления данных;
- сформулировать формализованную постановку задач конструкторского проектирования, позволяющую инженеру выбрать набор операций и способ перехода от модели начального описания объекта к результирующей модели [23].

Согласно общепринятой терминологии в теории графов, под графом $G(X, U, P)$ будем понимать совокупность непересекающихся множеств вершин X и ребер U , а также предикат $P(X, U, X)$, определенный на данных множествах [22].

Предикат $P(X, U, X)$ представляет собой конъюнкцию двух предикатов-инцидентности $\Gamma_1(X, U)$ и $\Gamma_2(U, X)$:

$$P(X, U, X) = \Gamma_1(X, U) \& \Gamma_2(U, X). \quad (1.1)$$

Предикат инцидентности $\Gamma_1(X, U)$ принимает значение «истинна» ($\Gamma_1(X, U) = \text{"и"}$) в том случае, если элементы множества X (вершины гиперграфа) находятся в отношении инцидентности элементам множества U (ребрам гиперграфа), а $\Gamma_2(U, X) = \text{"и"}$ в том случае, если элементы множества U (ребрам гиперграфа) инцидентны элементам множества X (вершинам гиперграфа) [22].

Пусть $X = \{x_1, x_2, x_3, x_4\}$, $U = \{u_1, u_2, u_3, u_4, u_5\}$ и предикаты инцидентности $\Gamma_1(X, U)$ и $\Gamma_2(U, X)$ истинны на парах

$$\begin{aligned} \Gamma_1(X, U): & (x_1, u_1), (x_1, u_2), (x_2, u_3), (x_3, u_1), (x_3, u_5), (x_4, u_2), (x_4, u_4), \\ \Gamma_2(U, X): & (u_1, x_2), (u_2, x_1), (u_2, x_3), (u_3, x_4), (u_4, x_3), (u_5, x_3). \end{aligned} \quad (1.2)$$

На рисунке 1.4 представлена геометрическая интерпретация предикатов инцидентности $\Gamma_1(X, U)$ и $\Gamma_2(U, X)$ и их конъюнкции $\Gamma_1(X, U) \& \Gamma_2(U, X)$.

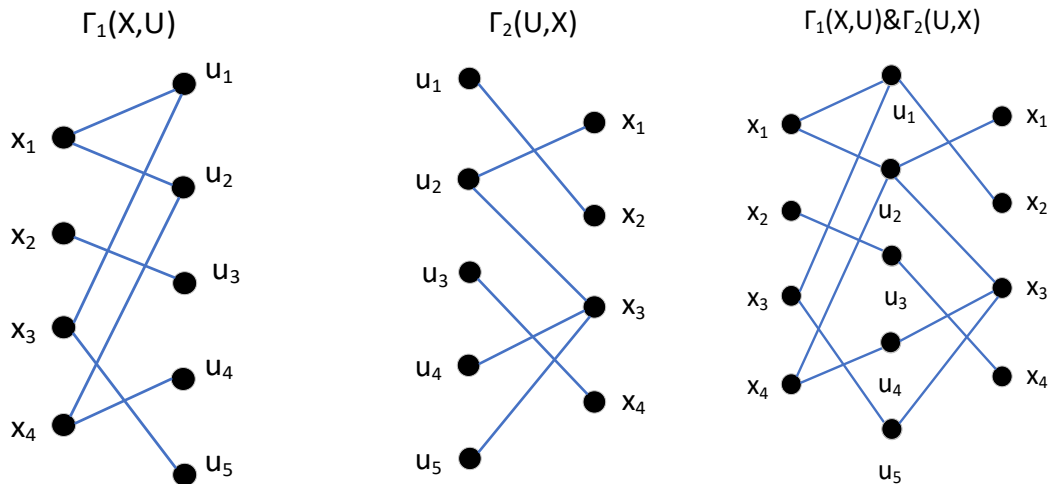


Рисунок 1.4 –Предикаты инцидентности и их конъюнкция

Таким образом, для всех графов для $X \neq \emptyset$ и $U \neq \emptyset$ верно, что:

$$\Gamma_1(x_i, u_j) = \text{"и"} \& \Gamma_2(u_j, x_k) = \text{"и"} \vee \Gamma_1(x_i, u_j) = \text{"и"} \& \Gamma_2(u_j, x_i) = \text{"и"}. \quad (1.3)$$

Для моделирования СБИС графовая модель является недостаточно

информативной, т. к. одно ребро графа соединяет не более двух вершин, в то время как СБИС содержит цепи, соединяющие более двух элементов. Поэтому в данной диссертационной работе автор предлагает использовать в качестве ММ гиперграф, т. к. такой способ представления позволяет наиболее полно и адекватно отобразить электрическую схему (СБИС).

Гиперграф задается следующим образом:

$$\begin{aligned} \forall x_i \in X, \forall u_i \in U (\Gamma_1(x_i, u_j) = \text{"и"} \& \Gamma_2(u_j, x_i) = \text{"и"}) \\ \exists u_i \in U (|\Gamma_2 u_j| > 2). \end{aligned} \quad (1.4)$$

Данное утверждение эквивалентно высказыванию «предикат инцидентности $\Gamma_2(U, X)$ обратен предикату $\Gamma_1(X, U)$ ». Гиперграф задается парой непересекающихся подмножеств X и U , на элементах которых предикаты $\Gamma_1(X, U)$ и $\Gamma_2(U, X)$ принимают значение «истинна», причем $\Gamma_2(U, X) = \Gamma_1^{-1}(X, U)$. Из этого следует, что гиперграф задается с помощью множества вершин X , множества ребер U и обобщенного предиката $\Gamma(X, U)$ [24].

На рисунке 1.5 приведен пример геометрического представления гиперграфа, где вершины задаются окружностями, ребра – контурами. При этом нахождение вершины внутри окружности (ребра) интерпретируется как, истинность предиката инцидентности $\Gamma(X, U)$ [22].

Эквивалентным способом задания гиперграфа является матрица инцидентности A размером $n \times m$, где $n = |X|$ и $m = |U|$. Таким образом задается взаимосвязь между элементами гиперграфа (ребрами и вершинами), которую можно определить следующим образом [24]:

$$a_{i,j} = \{1, \text{если } \Gamma(x_i, u_j) = \text{"истина"} \ 0, \text{если } \Gamma(x_i, u_j) = \text{"ложь"} \quad (1.5)$$

где $i = 1, n, j = 1, m$.

$$a_{i,j} = \{1, \text{если } \Gamma(x_i, u_j) = \text{"истина"} \ 0, \text{если } \Gamma(x_i, u_j) = \text{"ложь"} \quad (1.5)$$

где $i = 1, n, j = 1, m$.

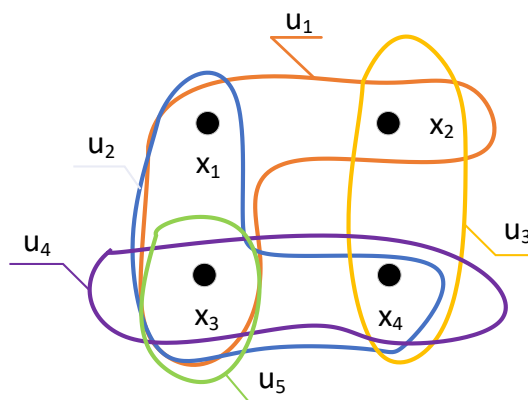


Рисунок 1.5 – Пример геометрического представления гиперграфа.

Матрица инцидентности A гиперграфа, приведенного на рисунке 1.5, примет вид

$$A = \begin{array}{c|ccccc} & u_1 & u_2 & u_3 & u_4 & u_5 \\ \hline x_1 & 1 & 1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 1 & 0 & 0 \\ x_3 & 1 & 0 & 0 & 0 & 1 \\ x_4 & 0 & 1 & 0 & 1 & 0 \end{array}$$

При конструкторском проектировании СБИС функциональные и принципиальные электрические схемы содержит набор связанных друг с другом базовых элементов. СБИС можно интерпретировать как множество элементов $X(x_1, x_n)$, связанных множеством цепей $U(u_1, u_n)$.

Гиперграф является адекватной ММ, если модули схемы возможно представить в виде вершин графа. Это допустимо в том случае, если контакты расположены произвольно или при решении задач, где порядковый номер вывода не имеет значения. К таким задачам относится задача разбиения схем при проектировании СБИС.

На рисунке 1.6 приведен пример фрагмента СБИС, содержащей 12 элементов и 12 цепей, соединяющих их. Необходимо разбить схему на 3 части с минимизацией количества межблочных соединений. На рисунке 1.7 приведен результат разбиения. В результате были получены 3 блока по 4 элемента и 5

межблочных соединений – цепи 2, 4, 5, 6 и 8.

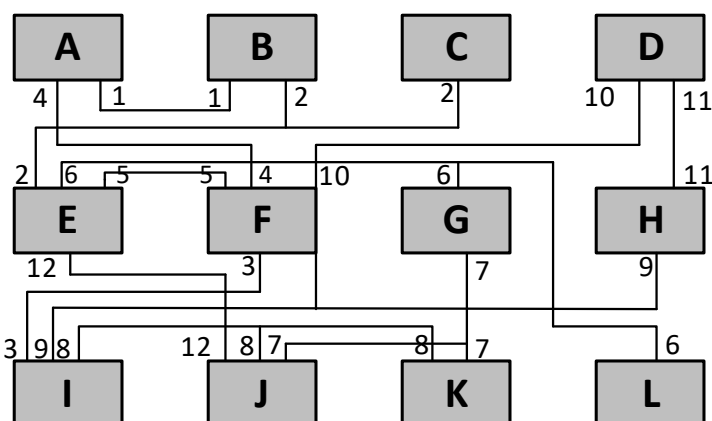


Рисунок 1.6 – Фрагмент СБИС

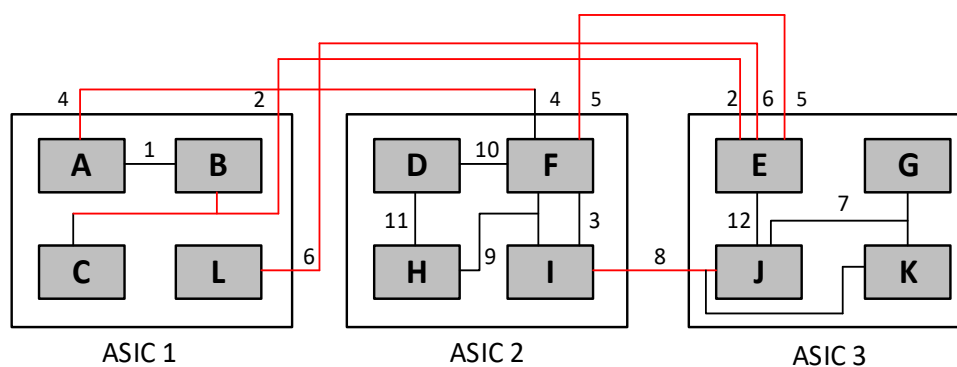


Рисунок 1.7 – Результат разбиения схемы на 3 блока

Автор применяет данную ММ, так как гиперграфовое представление позволяет точно оценить число электрических соединений. При наличии дополнительных данных о конструктивно-технологических характеристиках и особенностях элементов и их логических функциях представление схемы в виде гиперграфа весьма целесообразно, т. к. максимально удовлетворяется требование информационной полноты ММ. При задании элементов схемы в виде вершин графа эти данные (для всех моделей) могут задаваться в виде весовых характеристик каждой вершины.

Проанализировав вышеизложенные факты, можно утверждать, что гиперграфовая модель полностью удовлетворяет требованиям к моделируемому объекту, а также способствует последующей формализации и алгоритмизации при

решении задачи разбиения СБИС. Таким образом, автор предлагает использовать гиперграфовый способ представления данных для компьютерного моделирования разбиения схем при проектировании СБИС.

1.3 Постановка задачи разбиения схемы при проектировании СБИС

При разработке полнозаказных и полузаказных СБИС, а также многокристальных модулей наиболее затруднительными проектными процедурами этапа конструкторского проектирования являются разбиение, планирование, размещение и трассировка. Ввиду высокой степени интеграции элементов на кристалле и, следовательно, высокой размерности данных задач, разработка топологии схемы, как единого объекта, не представляется возможным даже с помощью современных технологий проектирования. В связи с этим целесообразно использование иерархического подхода, при котором схема разбивается на части для независимого выполнения проектных процедур [6, 8,9].

На сегодняшний день требования к миниатюризации и быстродействию СБИС постоянно повышаются, а возрастающая роль проектной процедуры разбиения становится все более очевидной. Разбиение схем при проектировании СБИС заключается в нахождение такого относительного «расположения» элементов схемы, при котором обеспечивалось бы выполнение предъявляемых к техническому объекту требований. Основными требованиями к СБИС являются [2]:

- максимальное быстродействие схемы, т. е. уменьшение временных задержек сигнала,
- помехоустойчивость и температурная стабильность,
- технологичность и др.

Данные требования тесно взаимосвязаны с параметрами задачи разбиения схемы, которые можно рассматривать как критерии. Количество внешних связей

между частями схемы представляет собой один из наиболее важных факторов, который определяет надежность СБИС.

Это связано с тем, что сокращение числа межмодульных соединений приводит к уменьшению индуктивных наводок, повышению помехоустойчивости, увеличению температурной стабильности, а также упрощению конструкции. Таким образом, при разбиении схемы при проектировании СБИС в качестве основного оптимизационного критерия принимается количество межблочных соединений, минимизация которого направлена на повышение характеристик надежности СБИС, сокращение влияния индуктивных наводок и временной задержки в схеме, а также повышение технологических характеристик разрабатываемого технического объекта [10].

Повышение степени интеграции СБИС возможно за счет уменьшения габаритов СБИС и увеличения числа уровней металлизации, что влияет на увеличение паразитной емкости и индуктивности и, соответственно, возрастанию задержки сигнала в схеме [25]. Для учета данных параметров используется топологическая норма, которая задает минимальное расстояние между элементами СБИС.

На сегодняшний день минимальные параметры первого уровня металлизации представлены в международной технологической дорожной карте International Technology Roadmap for Semiconductors (ITRS) [26], которые приведены в таблице 1.1.

С целью минимизации временных задержек прохождения сигнала в современных СБИС необходимо разбиение элементов схемы на части таким образом, чтобы количество межблочных соединений стремилось к минимальному значению.

На рисунках 1.8 и 1.9 приведено разбиение некоторой условной СБИС на три части согласно критерию общего количества межмодульных соединений [14].

Таблица 1.1 Параметры слоя металлизации по ITRS 2015.

Параметры первого уровня металлизации по ITRS-2015

Топологическая норма, нм	90	65	45	32	22	16	10	7
Число уровней	10	11	12	12	12	13	14	14
Задержка сигнала, пс	370	409	626	966	2364	3864	8731	10938
Первый уровень металлизации (M1)	Толщина (d), нм	170	150	125	95	65	40	30
	Ширина (w), нм	110	95	80	55	35	20	15
	Расстояние между проводниками (t_{ox}), нм	220	210	160	115	75	45	25
Диэлектрик	Толщина (h_{ox}), нм	320	205	130	85	55	35	20

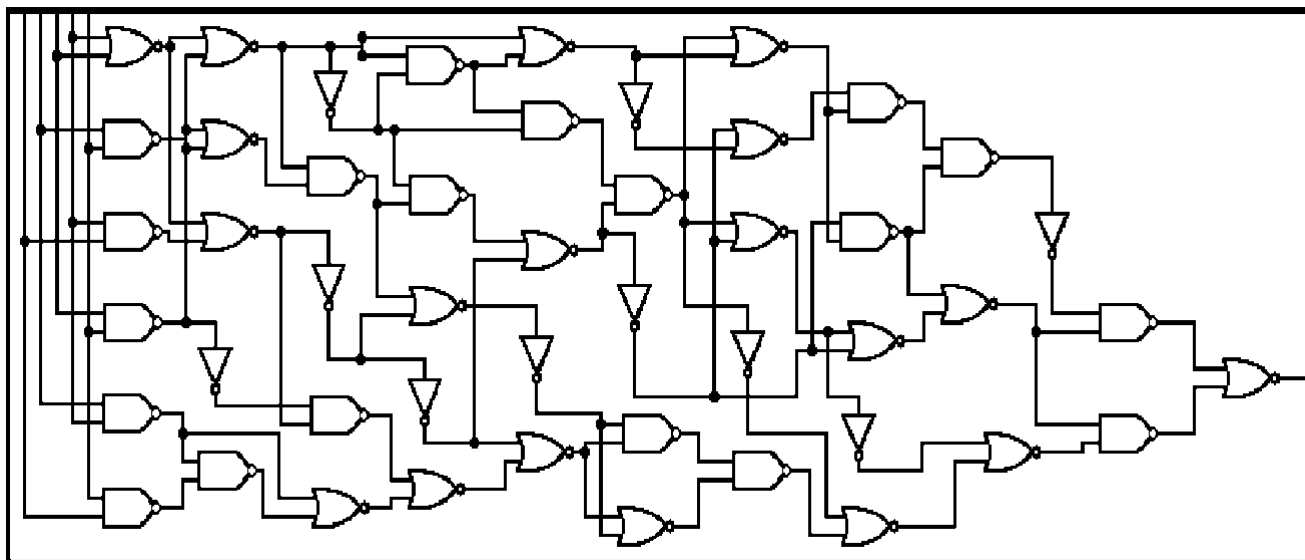


Рисунок 1.8 – Исходный участок СБИС (до разбиения)

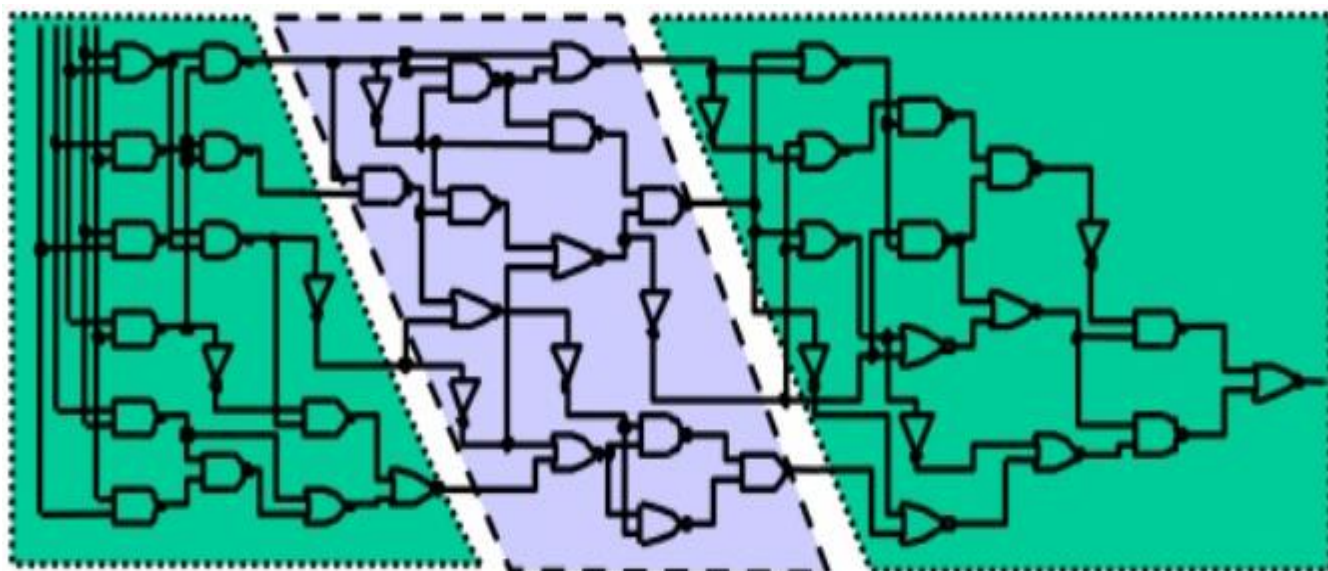


Рисунок 1.9 – Решение задачи разбиения участка СБИС

В данном примере емкость первого блока разбиения составляет $S_1 = 15$ элементов, второго - $S_2 = 16$ и третьего - $S_3 = 17$. Число разрезов цепей - $CUT = 8$.

В настоящее время размеры элементов (транзисторов) уменьшились до 0,09 микрон, что позволило увеличить производство широкой номенклатуры суперкристаллов, содержащих 10^9 транзисторов на кристалле 25x25 мм [27]. В связи с этим большая часть временной задержки приходится на область межсоединений.

Рисунок 1.10 отражает зависимость временных задержек сигнала в схеме от технологического процесса.

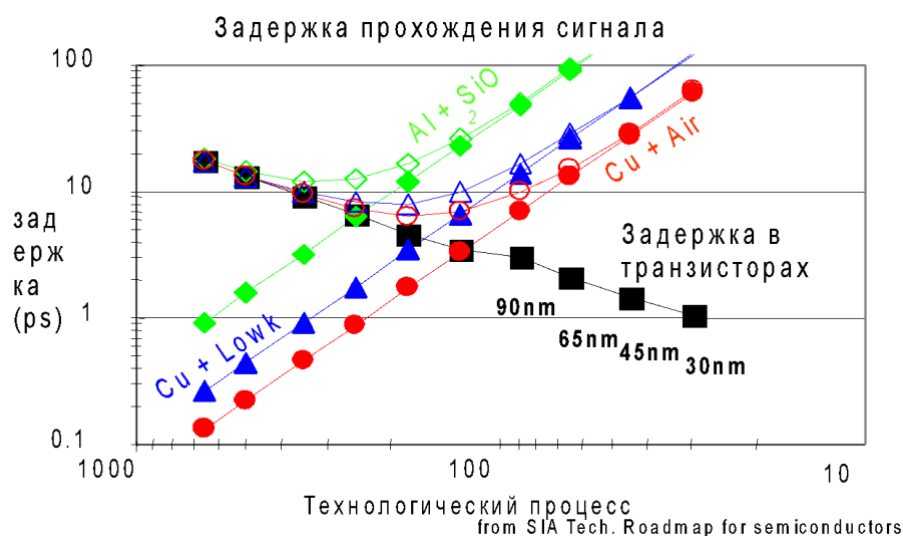


Рисунок 1.10 – График зависимости временных задержек сигнала в схеме от технологического процесса

При проектировании СБИС большой степени интеграции немаловажную роль играет помехоустойчивость и температурная стабильность, которые влияют на электромагнитнотепловую совместимость элементов СБИС. Данная характеристика, как правило, учитывается при размещении элементов на кристалле, т. к. введение дополнительных параметров при выполнении проектной процедуры разбиения может влиять на качество решения.

Поскольку значительный процент временной задержки сигнала при проектировании СБИС приходится на межсоединения, в данной диссертационной работе автор предлагает использовать комбинацию следующих критериев: количество разрезв цепей и временная задержка сигнала в схеме.

Проведя анализ работ, посвященных задаче разбиения элементов с учетом задержки временного сигнала при проектировании СБИС [27-32], автор выявил ряд проблем:

1. В ряде случаев, предлагается такой способ вычисления задержки сигнала в СБИС, который не способен адекватно описать процессы, протекающие в кристалле. При таком методе моделирования предлагается присвоить 1 задержке в вентильях, связям внутри одного узла – 0 и постоянный уровень задержки (const) – связям, соединяющим блоки разбиения [28, 29].

2. В связи с введением некоторых допущений и упрощений, например, при преобразовании схемы в вентили с двумя выходами, повышается суммарная погрешность, что негативно сказывается на точность окончательного решения [30, 31].

3. Для СБИС среднего и малого размеров временная сложность алгоритма разбиения оказывается неприемлемо высокой [32].

Для решения выявленных проблем автор предлагает брать в расчет не только количество разрезов цепей, но и временную задержку сигнала в схеме. В литературе описано множество методов и алгоритмов решения задачи разбиения схем при проектировании СБИС. С классической точки зрения, цель данной задачи заключается в минимизации общего числа разрезов [27]. Но применение классических методов и алгоритмов имеет ряд ограничений в связи с тем, что при проектировании СБИС необходимо учитывать различные критерии, что в свою очередь значительно увеличивает вычислительную сложность алгоритма. В процессе многокритериальной оптимизации основной проблемой является тот факт, что зачастую невозможно получить единственное оптимальное решение поставленной задачи, т. к. оптимум существует для каждого критерия в конкретном пространстве поиска. Следовательно, при нахождении оптимального значения по одному критерию, по другим критериям, как правило, приходится принимать неоптимальное решение. В связи с этим, понятие «хорошее решение» является противоречивым [33, 34].

Для построения математической модели схемы в данной диссертационной работе автор предлагает использовать гиперграфовую модель [22, 24]. Таким образом, задача разбиения схемы сводится к разбиению гиперграфа $H = (X, U, P)$ на блоки с учетом комбинации критериев количества разрезов цепей и временной задержки сигнала в схеме. Суммарный вес ребер, определяющий физические параметры проводника, задается множеством $\psi = \{\psi_j \mid j = 1, 2, \dots, m\}$. Целью является разбиение множества X на K непустых и непересекающихся подмножеств X_v :

$$X = \bigcup_v X_v, \quad X_{ij} \cap X_v = \emptyset, \quad i, j \neq v. \quad (1.6)$$

Ограничением является число элементов в каждом узле, которое задается вектором $N = \{n_v \mid v = 1, 2, \dots, k\}$. Каждый элемент здесь является максимальным значением количества вершин, которое может быть назначено в узел v .

Ограничения можно представить в виде неравенства:

$$|X_v| \leq n_v, \quad \text{где } v = 1, 2, \dots, k \quad (1.7)$$

Общее число ребер, которые пересекают разрез, определяется следующим образом:

$$C = \{u_j \mid (v) [u_j \cap X_v \neq \emptyset]\} \quad (1.8)$$

Задержка сигнала D является суммой следующих характеристик: временная задержка сигнала в критических связях, общее число разрезов каждого критического пути, а также вес ребер на критических путях.

$$D = \sum_i^{|U|} \psi_i u_i + \sum_j^K D_j k_j \quad (1.9)$$

где ψ_i – вес ребер, лежащих на критических путях; e_i – коэффициент, $e_i = [0;1]$, который зависит от того разрезается данное ребро или нет; $|U|$ – количество ребер; D_j – значение задержки в j -ой цепи; k_j – количество разрезов цепи j [27].

Временная задержка сигнала вычисляется на основании двух характеристик: задержка вентиля (не учитывается) и задержка сигнала в проводниках. Автор предлагает моделировать временную задержку проводников на основе модели Эльмора:

$$D(e) = R_e \left(\frac{C_e}{2} + C_i \right) \quad (1.10)$$

где R_e – сопротивление проводника e , C_e – емкость проводника e , C_i – емкость истоков каждой цепи [27].

Для определения R_e и C_e вычислим общую длину каждого ребра:

$$L_e = \sum_i ((\alpha * m_{ij}^\gamma - \beta) \frac{a_i * b_i}{a_i + b_i} + (a_i + b_i)) \quad (1.11)$$

где a_i и b_i – геометрические размеры узлов; α , β , γ – настроечные параметры, $\alpha = 1.1$, $\beta = 2.0$, $\gamma = 0.5$. Удельное сопротивление проводника считаем $r = 0.115$, удельная емкость проводника $c = 0.00015$ [27].

В данной диссертационной работе автор предлагает комплексный критерий F (целевую функцию), учитывающий следующие частные критерии: C – количество разрезов цепей в схеме и D – оценку временной задержки прохождения сигнала в схеме. При этом следует отметить, что нахождение оптимального количества разрезов C может не способствовать оптимизации временной задержки D и наоборот [34].

Очевидно, что рассматриваемые частные критерии C и D имеют различную размерность. Следовательно, для получения комплексного критерия оптимальности невозможно использовать их «натуральные» значения, а необходимо провести процедуру нормирования. Нормированная величина рассматриваемых критериев вычисляется с помощью деления каждого «натурального» критерия на нормирующую величину, которая измеряется в соответствующих частному критерию единицах [34].

Для нормализации частных критериев автор предлагает использовать

разность между максимальным и минимальным значениями частных критериев в области компромисса в качестве нормирующих делителей. Нормированные частные критерии C^N и D^N имеют вид:

$$C^N = \frac{C - C_{\min}}{C_{\max} - C_{\min}} \quad (1.12)$$

$$D^N = \frac{D - D_{\min}}{D_{\max} - D_{\min}} \quad (1.13)$$

где C – целевая функция, которая описывает количество разрезов цепей в схеме, D – целевая функция, описывающая временную задержку схемы [34].

Метод аддитивной свертки критериев заключается в объединении всех частных критериев для формирования обобщенной целевой функции (ЦФ), которая является взвешенной суммой всех частных критериев [33].

При решении задачи разбиения гиперграфа H на части ЦФ F примет вид:

$$F = \omega_1 * C^N + \omega_2 * D^N \quad (1.14)$$

где ω_1 и ω_2 – весовые коэффициенты, определяющие степень важности частных критериев ($\omega_1 + \omega_2 = 1$). Весовые коэффициенты введены для учета важности частных критериев при формировании комплексного критерия.

Данный метод дает возможность найти компромисс, при котором повышение качества одного частного решения компенсирует понижение качества другого частного критерия.

Цель задачи разбиения заключается в нахождении такого разбиения k гиперграфа H , при котором функция оценки текущего разбиения F , минимизируется с учетом заданных ограничений [30, 31, 35]:

$$(1.15)$$

$$F \rightarrow \min$$

ограничения: $|X_v| \leq n_v$, где $v = 1, 2, \dots, k$.

В качестве ограничения при решении задачи разбиения СБИС автор вводит площадь каждого блока разбиения. Площадь $S(h_j)$ элемента задается следующим образом

$$S(h_j) = p_j * t_j, \quad (1.16)$$

где p_j – длина j -го элемента, t_j – ширина j -го элемента.

Площадь $S(H_i)$ блока разбиения определяется как сумма площадей входящих в него элементов. Ограничение, описывающие площадь, занимаемую элементами, примет вид

$$Z = \sum_{j=1}^n S(H_j). \quad (1.17)$$

Отметим, что задача разбиения гиперграфа H на части является комбинаторно-логической, т. е. оптимальное решение может быть получено с помощью большого перебора различных вариантов разбиения [36, 37].

Таким образом, решение задачи разбиения СБИС с учётом комплексного критерия и поставленных ограничений, позволяет повысить качество разбиения и произвести подготовку для решения последующих задач на этапе конструкторского проектирования.

1.4 Классификация и анализ алгоритмов разбиения схем при проектировании СБИС

Задачу разбиения схем при проектировании СБИС можно отнести к задачам дискретной оптимизации. Существует большое количество методов и алгоритмов, нацеленных на решение задач данного класса, но наиболее эффективными принято считать методы, направленные на решение узкоспециализированного кластера задач (например, задачи линейного, квадратичного и выпуклого программирования и т. д.) [21].

Тем не менее, большую часть известных методов нецелесообразно использовать для решения поставленной задачи в связи с ее дискретным характером. Данная задача относится к классу комбинаторных, при решении которых общее число альтернативных решений эквивалентно числу парных перестановок из n элементов, т. е. $C_n = n!$. Учитывая ограничения на количество сформированных подмножеств (частей графа), общее количество решений задачи будет равно количеству сочетаний из n элементов по m , т. е.

$$C_n^m = \frac{n!}{m!(n-m)!} \quad (1.18)$$

где n – общее число элементов, m – количество частей разбиения [38].

Исходя из вышесказанного можно сделать вывод, что решать задачу разбиения схем при проектировании СБИС полным перебором неэффективно ввиду большого объема вычислений. Это стало причиной возникновения класса алгоритмов для решения задач данного типа, которые основаны на различных последовательных, математических и итерационных методах. При этом в последнее время стали широко применяться методы генетического и биоинспирированного поиска оптимальных решений. В основе данного класса методов лежит концепция перебора большого количества вариантов решения задачи, но каждый метод отличается принципом выполнения и характером вычислительной процедуры (последовательные, параллельно-последовательные, итерационные) [39].

На рисунке 1.11 приведена классификация алгоритмов разбиения элементов схем при проектировании СБИС. Основываясь на данной классификации, можно выделить алгоритмы разбиения конструктивных блоков и алгоритмы разбиения типовых блоков. При разбиении конструктивных блоков принято использовать математические, последовательные, параллельно-последовательные или итерационные алгоритмы. Математические методы условно делятся на методы целочисленного линейного программирования и метод ветвей и границ [40].

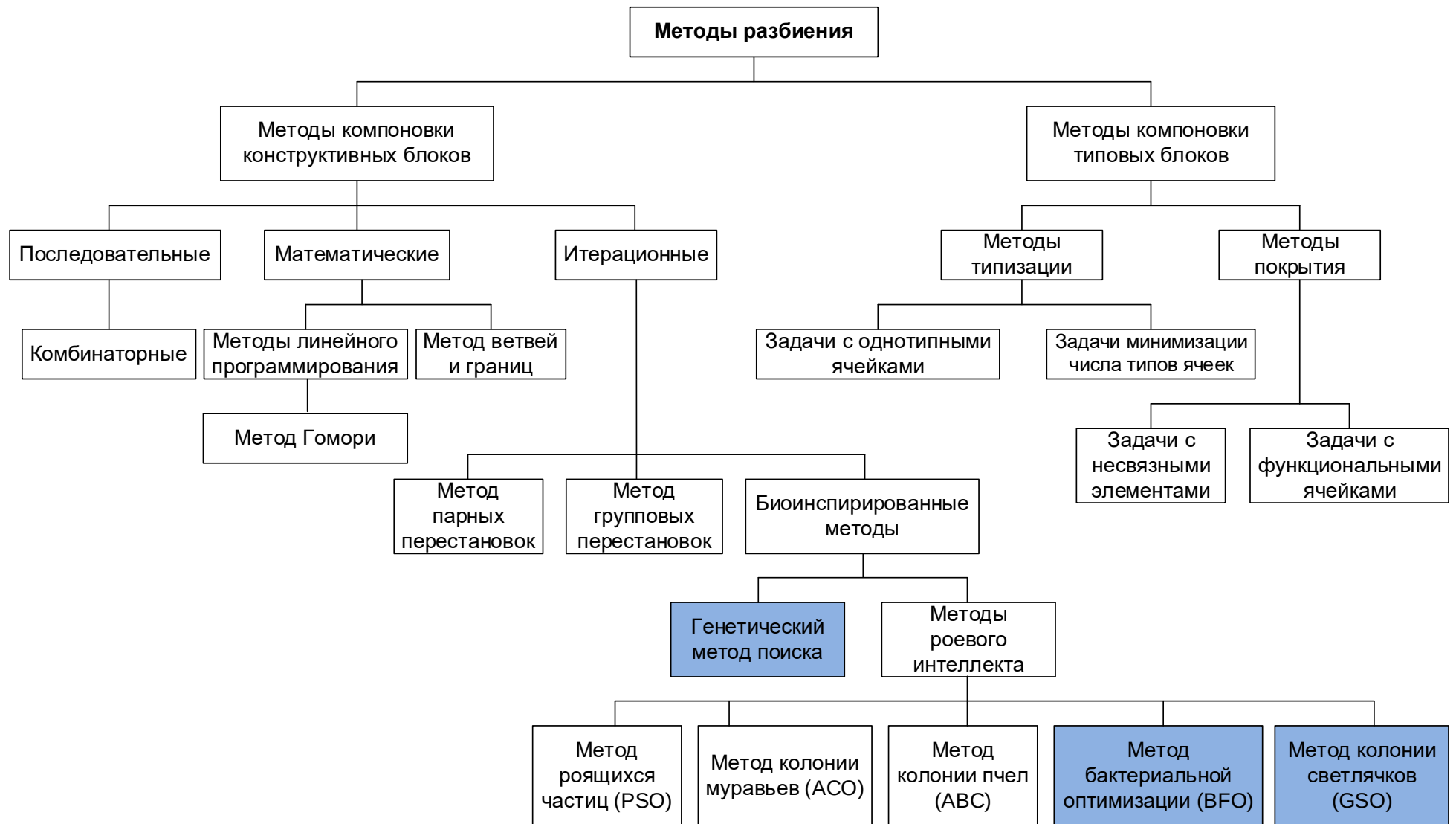


Рисунок 1.11 – Классификация методов разбиения

В первую группу алгоритмов входят так называемые методы отсечения, а именно алгоритм Гомори, который применяется для решения целочисленных задач линейного программирования. Алгоритм заключается в применении одного из симплекс-методов или методов внутренней точки без учёта целочисленности. В том случае, если полученное решение целочисленно, то задача считается решенной. В противном случае вводится дополнительное ограничение, а значение переменной приводится к целочисленному виду в соответствии с заданными правилами. Затем оно вводится в систему линейных ограничений, и задача решается заново с учетом исходных и дополнительных ограничений. Если целочисленное значение не получено, то необходимо повторить второй этап [41].

Часто в поисковых методах для получения квазиоптимального решения применяется иерархическое разбиение исходной задачи на множество подзадач, каждая из которых решается методом перебора на дереве решений. В большинстве случаев задачи оптимального разбиения сводятся к перебору альтернативных решений, близкому к полному, что для решения задачи разбиения неприемлемо. Чтобы избежать полного перебора, но обеспечить высокую точность полученных решений, применяют точные методы, классическим представителем которых является метод ветвей и границ.

В основе *метода ветвей и границ* лежит перебор альтернативных вариантов решений поставленной задачи при том условии, что количество вариантов решения является конечным множеством. Таким образом, в общем случае полный перебор заменяется частичным в связи с использованием априорных оценок вариантов, которые позволяют не учитывать так называемые «неперспективные» решения. Суть метода заключается в разбиении множества альтернативных решений на части с последующим их сокращением в зависимости от количества решений в каждой части. Далее определяется минимальное значение ЦФ рассматриваемого подмножества. Результатом является подмножество с оптимальным решением [42].

Достоинства: возможность получения решения, приближенного к оптимальному; использование иерархической стратегии для разбиения задачи на

несколько подзадач; удаление неэффективных решений из поискового пространства.

Недостатки: нет возможности разработки рекомендаций по выбору иерархического подхода к разбиению исходной задачи; высокая временная сложность алгоритма.

Последовательные алгоритмы, ориентированные на решение задачи разбиения, основаны на комбинаторных методах оптимизации. Комбинаторные оптимизационные методы – это методы, которые базируются на идеях комбинаторики – раздела математики, освещающего проблемы, связанные с расположением частей конечного множества объектов. Основная идея последовательных алгоритмов заключается в замещении первичной задачи деревом более «простых» подзадач или в разработке правил, исключающих изначально неоптимальные альтернативные решения. Комбинаторные методы можно отнести к классу эвристических алгоритмов, они отчасти уникальны для решения различных классов задач. Кроме того, чем специализированнее метод, тем более решения более высокого качества можно получить. При реализации последовательных алгоритмов выполняется последовательное разбиение всего множества решений, затем добавляется один элемент на каждом шаге алгоритма (в соответствии с его приоритетом) в следующую часть разбиения [43].

Достоинства: высокая степень быстроедействия; простота реализации.

Недостатки: отсутствует возможность нахождения глобального оптимума ЦФ (количества внешних связей); результаты высокого качества возможно

получить в том случае, если число вершин исходного графа больше числа вершин любого его разбиения.

В *параллельно-последовательных алгоритмах* на начальном этапе поиска отбирается определенное подмножество элементов, которое разбивается на части в соответствие с заданными критериями и ограничениями. Как правило, данный класс алгоритмов применяется при решении задачи разбиения схем в том случае, если предъявляются специальные требования такие, как минимизация количества однотипных блоков, и имеют достаточно сложную структуру. В большинстве случаев алгоритмы данного типа получают решения плохого качества и используются лишь для формирования начального разбиения. Временная сложность параллельно-последовательных алгоритмов составляет $O(n^2) - O(n^3)$ [43].

Итерационные алгоритмы работают с некоторым начальным решением, в качестве которого выступает в данном случае исходное разбиение графа. Начальное разбиение может быть сформировано либо случайным образом, либо на основе, например, последовательного алгоритма. Причем, чем удачнее было сгенерировано исходное решение, тем выше вероятность получения более качественного решения в результате работы итерационного алгоритма [44].

К классу итерационных алгоритмов можно отнести алгоритмы парных перестановок, групповых перестановок, а также обширный класс биоинспирированных алгоритмов, который будет более детально рассмотрен ниже.

Алгоритм *парных перестановок* является простейшим в своем классе. Исходными данными является начальное (случайное) разбиение. Суть алгоритма состоит в том, что из множества элементов выбирается пара альтернативных решений, которые затем меняются позициями. В том случае, если при этом значение ЦФ улучшается, т. е. происходит оптимизация критерия, то такая замена считается валидной. В противном случае выбираются два других элемента для их последующей перестановки. Алгоритм парных перестановок работает до тех пор, пока не будет достигнут критерий останова. Временная сложность не велика и равна $O(n^2)$. Однако решения, полученные с помощью этого алгоритма, не

обладают высоким качеством размещения, ввиду отсутствия целенаправленности поиска [43, 44].

Алгоритм *групповых перестановок* аналогичен методу парных перестановок. Вместо пары элементов выбираются подмножества максимально несвязных элементов. На каждой итерации выполняются тестовые перестановки. В том случае, если значение ЦФ не улучшается на протяжении заданного количества итераций, то процесс поиска останавливается на текущем решении. Временная сложность алгоритма равна $O(n^2) \dots O(n^3)$. Данный алгоритм является модификацией метода парных перестановок. При решении конкретных задач алгоритм групповых перестановок дает качественные результаты, но для большинства задач данный метод не является оптимальным и не лишен недостатков алгоритма парных перестановок: поиск также осуществляется «вслепую» [44].

Достоинства: по сравнению с последовательными алгоритмами, итерационные алгоритмы демонстрируют более высокую точность и гарантируют получение локального оптимума.

Недостатки: в большинстве случаев алгоритм находит локально-оптимальное решение, т. е. ни одна перестановка пары альтернативных решений не ведет к улучшению значения ЦФ. Итерационные алгоритмы отличаются не высоким быстродействием, по сравнению с конструктивными алгоритмами. В итерационных алгоритмах временная сложность лежит в пределах $O(n \log n) - O(n^2) - O(n^2 \log n)$.

Алгоритмы разбиения типовых блоков принято рассматривать как задачи покрытия и типизации. Под покрытием схемы понимается отображение схемы устройства с учетом известных параметров элементов схемы и взаимосвязей между ними. В данном случае выделяются 2 типа подзадач: с функциональными ячейками и несвязными элементами. Под типизацией понимается разбиение СБИС на части с учетом некоторого критерия оптимальности, в качестве которого может выступать номенклатура частей разбиения или однотипность ячеек, что позволяет существенно уменьшить временные затраты в процессе проектирования [43].

Биоинспирированные оптимизационные методы впервые были применены в середине 1950-х и вызвали широкий резонанс в научных кругах. В скором времени они стали широко применяться для решения большого круга задач науки и техники, а также для решения задач на этапе конструкторского проектирования СБИС [39]. Отличительной особенностью данного класса методов является возможность обработки большого количества альтернативных решений с учетом множества критериев и ограничений. Особенно большое распространение получили методы генетического поиска, в основе которых лежит механизм естественного отбора [46].

Генетический алгоритм (ГА) – эвристический алгоритм поиска оптимального решения на основе механизмов, происходящих в процессе естественной эволюции, таких, как селекция, скрещивание, мутация и отбор. Альтернативное решение кодируется в виде вектора, т. е. хромосомы, при этом каждый ген может задаваться битом, числом либо другим объектом. На каждой итерации ГА работает с закодированным множеством альтернативных решений, называемом популяцией [46].

Основная цель ГА заключается в получении альтернативного решения с наилучшими показателями критериев оптимизации, а значение ЦФ при этом стремится к экстремальному. Процесс поиска направляется с помощью генетических операторов (в большинстве случаев, кроссинговером и мутацией) для получения новых решений в популяции. Для них также определяется значение ЦФ, на основании чего происходит отбор наиболее перспективных решений в следующую генерацию. Процесс поиска производится итерационно до достижения критерия останова [46].

Достоинства: ГА показывают хорошие результаты при решении многокритериальных оптимизационных задач; работают с множеством альтернативных решений; устойчивы к попаданию алгоритма в локальную яму; относительно просты в реализации; эффективны в условиях неконтролируемо изменяющейся внешней среды.

Недостатки: нецелесообразно применять ГА для решения задач, в которых

необходимо найти точный глобальный оптимум, т. к. ГА в большинстве случаев находят квазиоптимальное значение, а также при затруднительном кодировании и декодировании альтернативных решений.

Метод колонии муравьев – эффективный эвристический метод оптимизации, направленный на нахождение приближенных решений графовых задач. Данный подход основывается на анализе поведения колонии муравьев во время поиска кратчайшего расстояния между источником питания и муравейником. Суть метода заключается в способности муравьев маркировать во время передвижения более удачные пути маршрута большим количеством фермента (феромона). С течением времени феромон испаряется, вследствие чего, на менее удачных маршрутах плотность феромона уменьшается [47, 48].

Метод колонии муравьев относится к классу методов дискретной оптимизации, но параметры ЦФ изменяются с течением времени непрерывно. Для перехода к дискретной оптимизации строится полный ориентированный граф, из каждой вершины которого выходят ребра со значениями параметров ЦФ. Каждая вершина графа ассоциируется с некоторым количеством муравьев. Цель каждого муравья заключается в прохождении такого количества вершин, сколько параметров содержит ЦФ. При этом муравей помечает пройденные ребра пропорционально качеству полученного решения [49].

Достоинства: при решении оптимизационных задач методом колонии муравьев учитывается информация обо всей колонии в целом, а не только о предыдущей генерации; данный метод в меньшей степени зависит от качества исходного решения, а также легко адаптируется к динамическим изменениям.

Недостатки: метод колонии муравьев гарантирует нахождение квазиоптимального решения, но время поиска определить весьма затруднительно; большую роль при реализации данного метода играют настроечные параметры алгоритма, определить которые возможно только экспериментальным путем.

Метод колонии пчел относится к классу стохастических биоинспирированных методов и основывается на моделировании поведения роя пчел в процессе поиска мест с наибольшим количеством нектара [48, 50–52].

Для этого в составе колонии выделяют два вида пчел: пчелы-разведчики и пчелы-фуражиры. Пчела-разведчик исследует пространство вокруг улья и затем передает информацию о найденном источнике нектара. Проанализировав полученную информацию, пчелы-фуражиры отправляются на наиболее перспективные участки, причем их количество прямо пропорционально качеству источника нектара.

Метод колонии пчел применяется для решения комбинаторных задач глобальной оптимизации. На начальном этапе выполняется «разведка» поискового пространства для выявления наиболее перспективных альтернативных решений с наилучшими значениями ЦФ (в большинстве случаев используется случайный перебор). Затем для уточнения полученных альтернативных решений производится исследование локальной области в пределах предварительно заданного радиуса поиска. При нахождении лучшего значения оно сохраняется, как текущее. Таким образом, комбинируя механизмы глобального и локального поиска на протяжении некоторого количества итераций, алгоритм позволяет получить улучшенное значение ЦФ. Алгоритм продолжает свою работу до тех пор, пока не будет достигнут критерий останова [53].

Достоинства: метод колонии пчел позволяет увеличить пространство поиска альтернативных решений без увеличения времени работы алгоритма; данный метод позволяет избежать преждевременной сходимости алгоритма к локальному оптимуму, а также улучшить качество квазиоптимальных решений.

Недостатки: метод колонии пчел не исключает попадания алгоритма в локальный оптимум при решении задач большой размерности.

Проведя анализ рассмотренных методов, ориентированных на решение комбинаторных задач, можно прийти к выводу, что все они обладают рядом существенных недостатков и не гарантируют получение качественных решений за приемлемое время. Таким образом, для повышения качества получаемых решения в данной диссертационной работе автор предлагает разрабатывать новые комбинированные архитектуры, в основе которых лежит параллельное и параллельно-последовательное выполнение современных модифицированных

алгоритмов, учитывающих все преимущества методов биоинспирированного поиска оптимальных решений.

1.5 Краткие выводы

1. Рассмотрены основные уровни проектирования СБИС и проведен анализ проектных процедур, реализуемых на стадии конструкторского проектирования. Автором выявлено, что проектная процедура разбиения схем является одной из наиболее важных при проектировании СБИС.

2. Рассмотрены графовые и гиперграфовые модели представления данных при проектировании СБИС. Отмечено, что при решении поставленной задачи важно применять математические модели, которые могут учитывать предварительные знания о решаемой задаче и наиболее адекватно моделирующие объект. В качестве модели представления решения задачи разбиения схем при проектировании СБИС выбрана гиперграфовая ММ, которая гарантирует наиболее адекватное отображение схемы.

3. Приведена постановка задачи разбиения схем и сформирована ЦФ с использованием комплексного критерия, учитывающего количество разрезов цепей и временную задержку сигнала.

4. Проведен обзор и сравнительный анализ существующих методов разбиения СБИС на части, на основе которого были выявлены их основные достоинства и недостатки. Для выполнения проектной процедуры разбиения схем при проектировании СБИС автор предлагает использовать комбинированную архитектуру на основе биоинспирированных эвристик, которые позволяют решать проблемы попадания алгоритмов в локальный оптимум и получать наборы квазиоптимальных решений за приемлемое время.

2 РАЗРАБОТКА МОДИФИЦИРОВАННЫХ БИОИНСПИРИРОВАННЫХ МЕТОДОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС

2.1 Общие положения теории биоинспирированного поиска

Для получения эффективных и качественных решений при разработке СБИС широко применяются современные стохастические методы оптимизации. Данный класс методов хорошо зарекомендовал себя при решении задач, возникающих на уровне конструкторского проектирования, в частности разбиении СБИС. В основе стохастических поисковых методов лежат так называемые биоинспирированные алгоритмы [47-49].

Принято выделять два класса поисковых алгоритмов глобальной оптимизации: траекторные и биоинспирированные. В классических траекторных алгоритмах происходит эволюция лишь одного альтернативного решения в то время, как биоинспирированные алгоритмы способны одновременно обрабатывать либо все, либо некоторое число альтернативных решений [54-56].

Все биоинспирированные алгоритмы глобального поиска являются эвристическими, т. е. не существует теоретического обоснования корректности метода и нахождение глобального экстремума не гарантировано. При этом на практике путем экспериментальных исследований было установлено, что в большинстве случаев алгоритм находит квазиоптимальное решение, достаточное для решения поставленной задачи [57, 58].

Так как существует огромное количество биоинспирированных алгоритмов, то для обобщения и систематизации основных понятий было введено понятие «агент», определяющее одного члена популяции. Под понятием агента могут пониматься различные объекты: особи, муравьи, бактерии, светлячки, пчелы и т. д., в зависимости от выбранного биоинспирированного алгоритма [59-61].

В общем случае биоинспирированные алгоритмы включают три этапа: инициализацию исходной популяции, миграцию агентов и завершение поиска [62].

На рисунке 2.1 приведена обобщенная схема работы биоинспирированных алгоритмов.



Рисунок 2.1 – Обобщенная схема работы биоинспирированных алгоритмов

Приведем более подробное описание основных этапов биоинспирированных алгоритмов [54].

1. *Инициализация популяции.* В поисковом пространстве формируется исходная популяция агентов в той или иной степени приближенных к глобальному решению поставленной задачи. На практике, как правило, для инициализации популяции применяют либо детерминированные, либо случайные алгоритмы.

2. *Миграция агентов популяции.* К сгенерированной начальной популяции агентов применяются так называемые миграционные операторы, специфичные для каждого биоинспирированного алгоритма. В результате каждый агент перемещается в поисковом пространстве в поисках глобального оптимума ЦФ.

3. *Завершение поиска.* На заключительном этапе проверяется условие выполнения критерия останова и, в случае его выполнения, поиск завершается, а агент с «лучшим» значением ЦФ принимается в качестве квазиоптимального решения. В противном случае поиск продолжается. В качестве критерия останова могут выступать достижение предварительно заданного числа итераций или

лимита времени, предоставленного на решение поставленной задачи, либо получение решения определенной точности.

Как видно из обобщенной схемы, все биоинспирированные алгоритмы имеют модульную структуру. Благодаря этому, путем комбинирования различных механизмов формирования начальной популяции, миграционных операторов и критериев останова поиска возможна генерация большого количества вариантов биоинспирированных алгоритмов.

Для всего класса биоинспирированных алгоритмов агенты характеризуются общим набором свойств: автономность, стохастичность, ограниченность представления, децентрализация и коммуникабельность [63].

Свойство автономности заключается в том, что в поисковом пространстве агенты движутся независимо друг от друга. Свойство стохастичности агентов объясняется отчасти случайным характером изменений в процессе миграции агентов популяции [61]. При этом каждый агент имеет ограниченное представление о пространстве поиска вокруг него, т. е. имеет данные лишь об исследуемой им области поискового пространства и, возможно, о соседних агентах. Поиск в популяции агентов децентрализован, т. е. нет агента, имеющего информацию обо всем процессе поиска и контролирующего его. Коммуникабельность агентов заключается в возможности обмена данными об оптимизируемой функции.

При сравнении классических методов поисковой оптимизации с биоинспирированными, последние с большей степенью вероятности определяют глобальный экстремум или квазиоптимальное решение при решении задач большой размерности, а также трудно формализуемых задач [62-64].

При решении задач невысокой размерности биоинспирированные алгоритмы покажут худший результат по сравнению с алгоритмом полного перебора. Также недостатком биоинспирированных алгоритмов является зависимость от значений достаточно большого значения свободных параметров.

Для оценки качества полученного решения (агентов популяции) используется понятие целевой функции. С точки зрения стратегии поиска, при

миграции каждый агент популяции стремится к глобальному оптимуму целевой функции.

В большинстве случаев целевая и оптимизируемая функция совпадают. В данном контексте можно утверждать, что целью биоинспирированных алгоритмов заключается в нахождении более высокого уровня приспособленности агентов популяции на текущей генерации по сравнению со значением их приспособленности на предыдущей генерации [63].

При разработке биоинспирированных алгоритмов одной из наиболее трудных проблем является поиск баланса между интенсификацией поиска, т. е. скоростью схождения алгоритма, и диверсификацией поиска, т. е. поиска в ширину. Интенсификация процесса поиска сводится к увеличению скорости сходимости биоинспирированного алгоритма, что приводит к уменьшению разнообразия агентов в популяции. При этом диверсификация процесса поиска обеспечивает исследование большего поискового пространства, что способствует увеличению вероятности нахождения глобального решения. Следовательно, целесообразно поддерживать разнообразие популяции агентов на протяжении большего числа генераций.

На первых генерациях биоинспирированного алгоритма агенты преимущественно исследуют поисковое пространство с целью нахождения новых альтернативных решений. В то время как, на последних генерациях происходит уточнение полученных ранее решений [64].

Для обеспечения баланса между интенсификацией и диверсификацией процесса поиска применяют механизмы адаптации и самоадаптации. Путем настройки значений свободных параметров биоинспирированных алгоритмов данные механизмы позволяют осуществить переход от диверсификации к интенсификации процесса поиска [12, 64].

На основе рассмотренных положений теории биоинспирированного поиска, можно сделать вывод, что для разработки новых более эффективных методов оптимизации целесообразно применять принципы комбинирования, гибридизации и многоуровневого подхода. Использование достоинств нескольких

биоинспирированных методов способствует получению набора квазиоптимальных решений за полиномиальное время и решению проблемы предварительной сходимости.

2.2 Построение комбинированной архитектуры биоинспирированного поиска

Задача разбиения СБИС относится к классу NP-полных комбинаторных оптимизационных задач, т.е. невозможно найти ее точное решение за полиномиальное время, т.к. размерность задач данного класса достаточно высока и современные ЭВМ не справляются с ее решением за приемлемое время. В связи с этим возникает необходимость поиска практически реализуемого результата и, соответственно, новых эффективных методов ее решения.

В данной диссертационной работе автор предлагает применение многоуровневого подхода для получения более качественного решения задачи разбиения СБИС. Преимуществом использования концепции многоуровневости является тот факт, что данные подходы способны легко адаптироваться к изменяющимся условиям внешней среды путем интеграции в свою структуру различных поисковых эвристик, направленных на повышения качества получаемого решения.

Исследованиями в данной области на мировом уровне занимаются G.Karypis [65], Alpert, C. J. [66], Burger M. [67], Faísca, N. P. [68], Kumar V. [69], T. Snijders [70], Garson, G. [71] и др. При этом, несмотря на очевидные преимущества применения многоуровневого подхода для решения комбинаторных задач, в РФ практически не проводятся исследования в данной области.

Для эффективного решения задачи разбиения СБИС автор предлагает использовать принципы комбинирования [72, 73], гибридизации [74-77], интеграции [78-79] и многоуровневости [80-82] для разработки новых эффективных многоуровневых подходов к решению поставленной задачи. На

рисунке 2.2 представлена комбинированная архитектура биоинспирированного поиска, основанная на многоуровневом подходе.

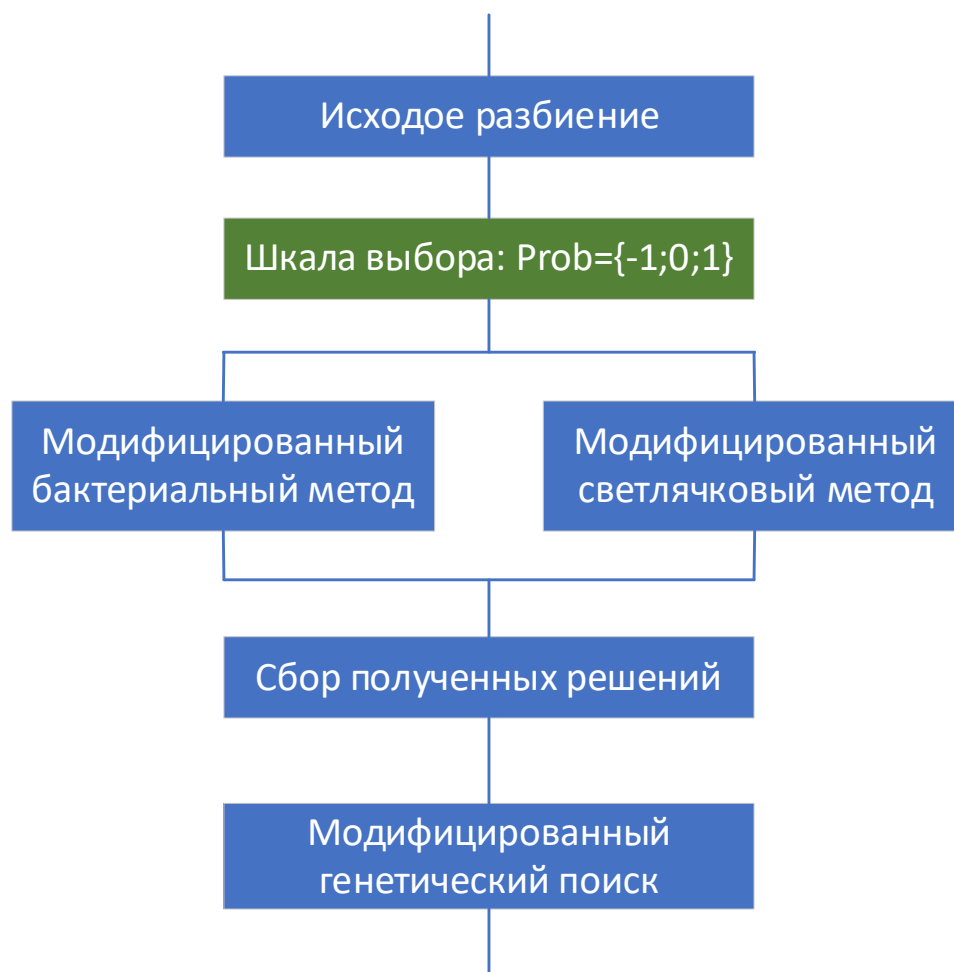


Рисунок 2.2 – Комбинированная архитектура биоинспирированного поиска на основе многоуровневого подхода

Для решения задачи разбиения схем при проектировании СБИС в данной диссертационной работе автор предлагает комбинированную поисковую архитектуру на основе методов бактериального, светлячкового и генетического поиска. Как и любой популяционный метод, предлагаемый подход работает с имеющимся начальным решением. В данной диссертационной работе начальное решение генерируется с помощью генератора случайных решений.

Далее вводится шкала выбора, позволяющая производить как последовательную, так и параллельную реализацию разработанных поисковых эвристик. Параметр шкалы выбора *prob* может принимать одно значение из

множества $\{-1,0,1\}$, где $prob = -1$ – последовательный поиск решения модифицированными светлячковым и генетическим методами, $prob = 0$ – последовательное выполнение модифицированных бактериального и генетического методов и $prob = 1$ параллельное выполнение модифицированных светлячкового и бактериального методов с последующим улучшением решения на основе модифицированного метода генетического поиска.

После их выполнения в блоке сбора полученных решений происходит отбор лучших решений на основе некоторого правила и выполняется редукция популяции до первоначального размера. Таким образом, формируется новая популяция как объект второго уровня. После этого проверяется первое условие останова: достигнуто ли заданное значение целевой функции. Очень часто при решении комбинаторных задач нет необходимости получить глобальный оптимум, а достаточно лишь приближенного к оптимуму квазиоптимального решения. В этом случае автор предлагает ввести пороговое значение ЦФ, по достижению которого алгоритм заканчивает выполнение. В данной диссертационной работе предложено использовать этот подход для сокращения времени работы разрабатываемого алгоритма.

В противном случае, если первое условие выхода не выполняется, производится попытка улучшения полученного решения с помощью модифицированного метода генетического поиска, в основе которого лежит модель эволюции Ламарка. В основе данной эволюционной модели лежит предположение о том, что характеристики, приобретаемые организмом в течение жизни, будут наследоваться потомками [48].

Разработанная в диссертационной работе комбинированная архитектура поиска позволяет эффективно использовать преимущества биоинспирированных поисковых методов, а также организовать работу методов в параллельном режиме, что способствует получению решений более высокого качества при решении задачи разбиения схем при проектировании СБИС.

2.3 Разработка модифицированного метода оптимизации на основе моделирования перемещения колонии бактерий

На сегодняшний день в области поисковой оптимизации наиболее перспективным принято считать класс оптимизационных методов, основанных на движении колонии бактерий в их среде обитания [83-85].

В основу данного класса методов легло моделирование перемещения колоний бактерий *E.coli*, т. к. она является наиболее изученной. Ее перемещение в пространстве обусловлено движением так называемых жгутиков, вращающихся с высокой частотой. Бактерия может моделировать два паттерна движения: прямолинейное и разворот в случайном направлении. В качестве основного стимула для прямолинейного движения бактерии выступает наличие или отсутствие в среде ее обитания полезных или вредных веществ (аттрактантов или репеллентов).

Данный тип двигательной активности бактерии называют хемотаксисом. Интеллектуальная составляющая в поведении бактерии обусловлена ее стремлением оказаться в среде, насыщенной питательными веществами, и избежать попадания в среду с вредными веществами [12].

В зависимости от характеристик среды бактерия демонстрирует различные схемы движения:

- в нейтральной среде бактерия движется произвольным образом, чередуя промежутки прямолинейного движения и разворотов в случайном направлении;
- в среде с повышенным содержанием аттрактантов движение направлено в сторону тех областей, где концентрация аттрактанта наибольшая;
- в среде с репеллентами бактерия стремится избежать попадания в область с высоким его содержанием.

Бактерия способна реагировать на достаточно незначительные изменения уровня полезных или вредных веществ благодаря специализированным сенсорам. В процессе моделирования поведения колонии бактерий необходимо учитывать, что она гибнет с течением времени. Таким образом, при соблюдении ряда условий,

бактерии объединяются в группы и демонстрируют интеллектуальность коллективного поведения [85].

Классический бактериальный метод активно применяется для решения широкого спектра задач поисковой оптимизации. В диссертационной работе автор предлагает для нахождения наиболее перспективных областей поискового пространства использовать концепцию процесса хемотаксиса в качестве случайного механизма поиска оптимального решения [86-88].

В основе оптимизационного процесса лежит последовательное выполнение нескольких процедур, при которых бактерия (агент) стремится оказаться в области с повышенным содержанием питательных веществ. К таким процедурам относятся хемотаксис, рассеивание и ликвидация.

Схема классического оптимизационного метода приведена на рисунке 2.3.



Рисунок 2. 3 – Классический метод бактериальной оптимизации

Предположим, $X_{i,r,l} - (|X| \times l)$ – координаты, которые определяют месторасположение бактерии $s_i \in S$ на t -ом шаге хемотаксиса (t – номер текущей итерации), r -м шаге репродукции и l -м шаге ликвидации.

Пусть $i \in [1:|S|]$, $t \in [1:\hat{t}]$, $r \in [1:\hat{t}^r]$, $l \in [1:\hat{t}^l]$, где $|S|$ – общее число особей в популяции S ; $\hat{t}$, $\hat{t}^r$, $\hat{t}^l$ – общее количество шагов хемотаксиса, репродукции и ликвидации. Таким образом, пусть значение ЦФ будет $\varphi_{i,r,l}$ [86].

В классическом методе оптимизации, инспирированном поведением колонии бактерий, процесс хемотаксиса лег в основу выполнения локальной оптимизации.

Последующее расположение бактерии $X'_{i,r,l}$ определяется следующим образом:

$$X'_{i,r,l} = X_{i,r,l} + \lambda_i \frac{V_i}{\|V_i\|_E}, \quad (2.1)$$

где V_i – текущий вектор шага хемотаксиса для s_i ; λ_i – текущее значение шага хемотаксиса. В случае прямолинейного движения бактерии значение вектора V_i остается постоянным на следующей итерации, т. е. верно равенство $V'_i = V_i$. В случае изменения направления движения вектор V'_i принимает случайное, значения в диапазоне $[-1;1]$.

В этом случае, при совершении бактерией разворота будет справедливо следующее выражение $V'_i = U_{|X|}(-1;1)$. Бактерия продолжает двигаться прямолинейно пока ЦФ продолжает увеличиваться.

Следующим этапом бактериального оптимизационного метода является механизм репродукции, направленный на увеличение скорости сходимости алгоритма путем сокращения объема поискового пространства. Пусть h_i – здоровье текущей бактерии s_i . Тогда ЦФ примет следующий вид:

$$h_i = \sum_{\tau=1}^t \varphi_{i,r,l}(\tau). \quad (2.2)$$

На следующем этапе вычисляется значение h_i , $i \in [1:|S|]$ и бактерии сортируются в порядке убывания их здоровья. Сортировка выполняется в

несколько шагов: на $(r+1)$ -м шаге репродукции из списка выбывают те агенты, у которых показатель здоровья является наименьшим (интерпретация гибели бактерии); после этого, каждый агент из списка преобразуется в два одинаковых агента с идентичными координатами [86-88]. В результате репродукции число агентов в популяции должно остаться постоянным.

В связи с тем, что вероятность предварительной сходимости остается достаточно высокой, выполнение хемотаксиса и репродукции не могут гарантировать нахождение глобального решения. Поэтому на заключительном этапе поиска выполняется механизм ликвидации, который заключается в следующем. После выполнения репродукции производится рандомный выбор n бактерий $s_{i_1}, s_{i_2}, \dots, s_{i_n}$ с заданной вероятностью ξ_e и последующее их уничтожение (удаление агентов из популяции). Для сохранения постоянного количества агентов в популяции на их место произвольно генерируется набор новых агентов под тем же порядковым номером.

Для повышения качества получаемых решений автор предлагает модифицировать классический метод бактериальной оптимизации. Для этого предлагается динамически варьировать число шагов, выполняемых в процессе хемотаксиса. Количество шагов при этом должно прямо зависеть от номера текущей генерации. Модифицированный метод бактериальной оптимизации представлен на рисунке 2.4.

В диссертационной работе на ранних этапах поиска автор предлагает использовать большое количество шагов хемотаксиса для поддержания высокого уровня диверсификации поиска. При этом количество шагов хемотаксиса обратно пропорционально номеру текущей итерации. Таким образом, на завершающих этапах поиска будет производиться поиск в глубину для увеличения точности текущего решения. Данная модификация позволяет получать решения более высокого качества по сравнению с классическим методом бактериальной оптимизации.



Рисунок 2.4 – Модифицированный метод бактериальной оптимизации

Отметим, что метод бактериальной оптимизации основан на мультиагентной эвристике, что позволяет сократить время поиска, т. к. поиск реализуется параллельно несколькими агентами. Характерной особенностью предложенного метода является его расширяемость, т. е. возможно добавление новых операторов в процесс поиска. Несмотря на очевидные преимущества, трудно оценить скорость схождения алгоритма. В большинстве случаев она носит случайный характер;

2.4 Разработка модифицированного метода оптимизации на основе поведения роя светлячков

Для решения задач оптимизации наука выделяет семейство светлячков Lampyridae, которые характеризуются наличием органов, обеспечивающих свечение на брюшных сегментах.

Механизм свечения у светлячков отвечает за коммуникацию между особями. Светлячки способны генерировать практически все виды сигналов, которые связаны с размножением, защитой и борьбой за территорию, но не все подвиды способны издавать весь спектр сигналов [89, 90].

Свечение происходит за счет выделения нескольких химических веществ, одним из которых является люциферин.

В 2005г. К. Кришнананд и Д. Гхоус предложили метод решения оптимизационных задач на основе поведения роя светлячков в среде их естественного обитания [91].

При разработке оптимизационного метода не моделировалась жизнь роя светлячков, однозначно копирующая существующую природную экосистему, а использовалась имитация колонии, как средство оптимизации, при котором система несколько отличается от естественной. Поэтому далее для формального описания модели используем вместо понятия «светлячок» понятие «агент».

При инициализации поиска все агенты произвольным образом распределены в поисковом пространстве целевой функции. Каждый агент выделяет определенное количество люциферина, которое пропорционально интенсивности свечения особи и имеет свою собственную динамическую область принятия решений r_d^i , ограниченную его областью восприятия r_s^i , т.е. $0 < r_d^i \leq r_s^i$. Схематичное представление областей восприятия агентов изображено на рисунке 2.5.

В данном случае $r_d^k < d(i, k) = d(i, j) < r_d^j < r_s^k < r_s^j$, т.е. агент i расположен в области видимости агентов j и k и на равном расстоянии от них. Но, т.к. область принятия решений агента j больше, то исключительно он имеет доступ

к информации относительно агента i .

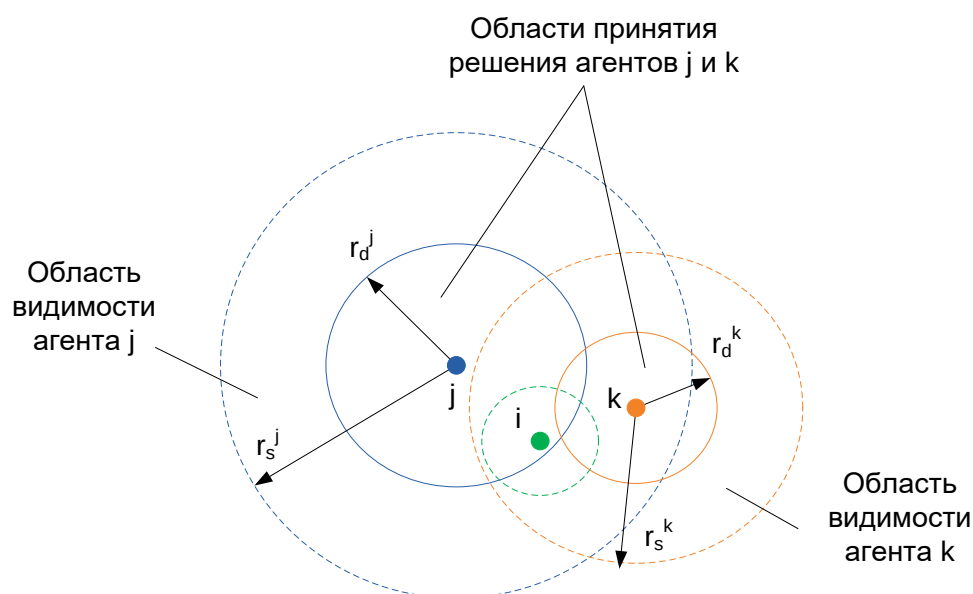


Рисунок 2.5 – Области восприятия агентов

Локальная область принятия решений задается радиусом окрестности поиска r_i для каждого i -го агента. Агент i рассматривает другого агента j как «соседа», если он находится в пределах радиуса окрестности поиска агента i и уровень люциферина l агента j выше, чем агента i , т.е. $l_j > l_i$. Используя вероятностный механизм, каждый агент выбирает соседнего агента, у которого уровень люциферина выше, чем его собственный, и движется в его направлении. Иными словами, каждый агент движется в направлении того агента, у которого уровень свечения выше. Интенсивность свечения каждого агента определяется значением целевой функции в текущем положении. Чем выше интенсивность свечения, тем больше значение целевой функции. Кроме того, радиус окрестности поиска каждого агента зависит от количества агентов в этой области. Если в окрестности поиска находится малое количество агентов, то ее радиус увеличивается. В противном случае радиус окрестности поиска сокращается [92].

Иными словами, данный алгоритм имеет 4 глобальных этапа: начальное распределение агентов в пространстве поиска, обновление уровня люциферина, перемещение агентов в более перспективную область поиска, обновление радиуса окрестности поиска каждого агента.

Базовая схема оптимизационного метода, инспирированного поведением роя светлячков, приведена на рисунке 2.6.

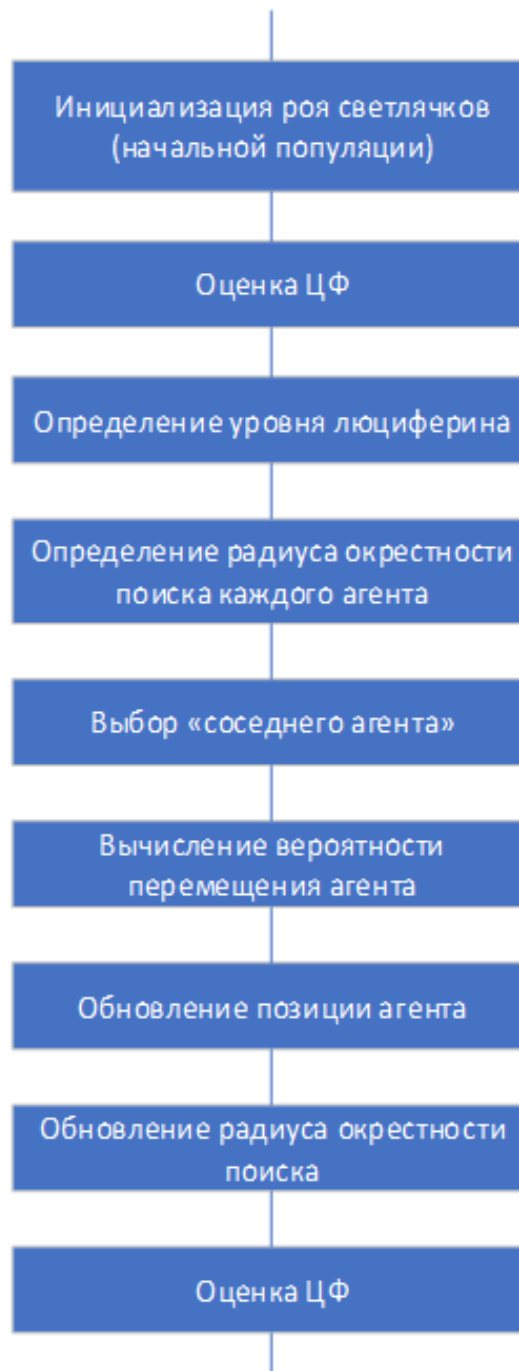


Рисунок 2.6 – Классический метод оптимизации, инспирированный поведением роя светлячков

Опишем процесс поиска оптимального решения пошагово.

Шаг 1. Инициализация входных параметров.

Шаг 2. Размещение в поисковом пространстве начальной (текущей) популяции, состоящей из n агентов. Изначально, все агенты имеют одинаковое количество люциферина. На каждой итерации происходит обновление уровня люциферина, а затем изменение положения агента в пространстве на основе заданных правил.

Шаг 3. Обновление уровня люциферина зависит от позиции агента в пространстве поиска (значения его целевой функции). Все агенты на начальной итерации имеют одинаковый уровень люциферина, поэтому значение целевой функции каждого агента зависит от его положения в пространстве поиска.

Уровень люциферина каждого агента увеличивается пропорционально измеряемым характеристикам агента (температура, уровень излучения). С точки зрения оптимизации это и является целевой функцией. Для моделирования процесса распада флуоресцирующего вещества производится вычитание части люциферина.

Вычисление уровня люциферина $l_i(t)$ (уровня свечения) i -го агента в момент времени t производится следующим образом:

$$l_j(t + 1) = (1 - \rho)l_j(t) + \gamma J_j(t + 1), \quad (2.3)$$

где ρ - коэффициент ослабления уровня люциферина для моделирования процесса распада флуоресцирующего вещества ($0 < \rho < 1$), γ - коэффициент привлекательности светлячка, J_j - значение целевой функции j -ого агента в момент времени t .

Шаг 4. Каждый агент выбирает того агента внутри радиуса окрестности поиска r_i , у которого уровень люциферина выше, чем его собственный, на основе «колеса рулетки». На данном определяются следующие значения: $N_i(t)$ - множества соседей i -го агента в момент времени t , r_i - радиуса окрестности поиска i -ого светлячка в момент времени t .

Шаг 5. При обновлении своего положения в пространстве поиска каждый агент на основе вероятностного механизма передвигается в направлении того агента, у которого уровень люциферина выше, чем его собственный.

На рисунке 2.7 приведен направленный граф на 6 вершин, отражающий зависимость движения каждого агента от уровня люциферина, влияющего на размер локальной области принятия решения. Агенты ранжируются в порядке возрастания уровня люциферина.

В данном случае у агентов a , b , c и d уровень люциферина больше, чем у агента e , который расположен в пределах области локального принятия решений только агентов c и d . Т.е. агент e имеет только два направления для перемещения. Учитывая, что уровня люциферина агента a больше, то, следовательно, агент a имеет наибольшую область принятия решений [15].

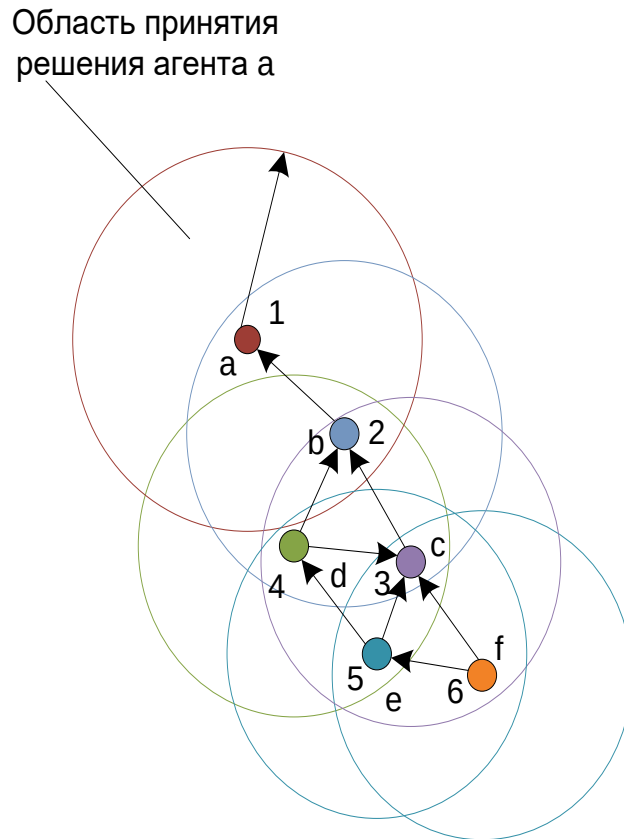


Рисунок 2.7 – Пример перемещения агента в пространстве поиска

Для каждого i -го агента вероятность перемещения в направлении агента j определяется по формуле:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2.4)$$

где $j \in N_i(t)$, $N_i(t) = \{j: d_{ij}(t) < r_d^i(t); l_i(t) < l_j(t)\}$; $d_{ij}(t)$ – евклидово расстояние между агентами i и j в момент времени t ; $l_j(t)$ – уровень люциферина агента j в момент времени t ; $r_d^i(t)$ – изменяемая локальная область принятия решений агента i в момент времени t .

Шаг 6. Агент i пытается распознать агента $j \in N_i(t)$ как «соседнего» с заданной вероятностью $p_{ij}(t)$, а затем передвигается в его сторону. В этом случае, обновленная позиция агента i задается следующим образом [91]:

$$x_i(t+1) = x_i(t) + s * \left\{ \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right\} \quad (2.5)$$

где s – шаг.

Для повышения качества решений оптимизационной задачи автор предлагает модифицировать способ обновления текущей позиции объекта путем оценивания показателя люциферина у двух соседних агентов вместо одного. Схема модифицированного оптимизационного метода на основе поведения роя светлячков изображена на рисунке 2.8.

Данная модификация позволяет уменьшить вероятность преждевременной сходимости алгоритма. На рисунке 2.9 проиллюстрирован пример выполнения модифицированного механизма обновления позиции агента.

Пусть движение агента b в направлении агента a выполняется с вероятностью $p_a = 0,5$, в направлении агента c – $p_c = 0,3$ и в направлении агентов d и e – $p_d = 0,1$ и $p_e = 0,1$. Тогда обновленная позиция агента b вычисляется следующим образом:

$$x_b(t+1) = x_b(t) + s * \left\{ \frac{x_u(t) - x_i(t)}{\|x_u(t) - x_i(t)\|} \right\}. \quad (2.6)$$

Здесь, $x_u(t)$ – координаты мнимого агента u , в направлении которого будет двигаться агент b . Координаты мнимого агента u определяются по формуле:

$$x_u(t) = \frac{p_a(x_a(t) + x_c(t))}{p_a + p_c}. \quad (2.7)$$

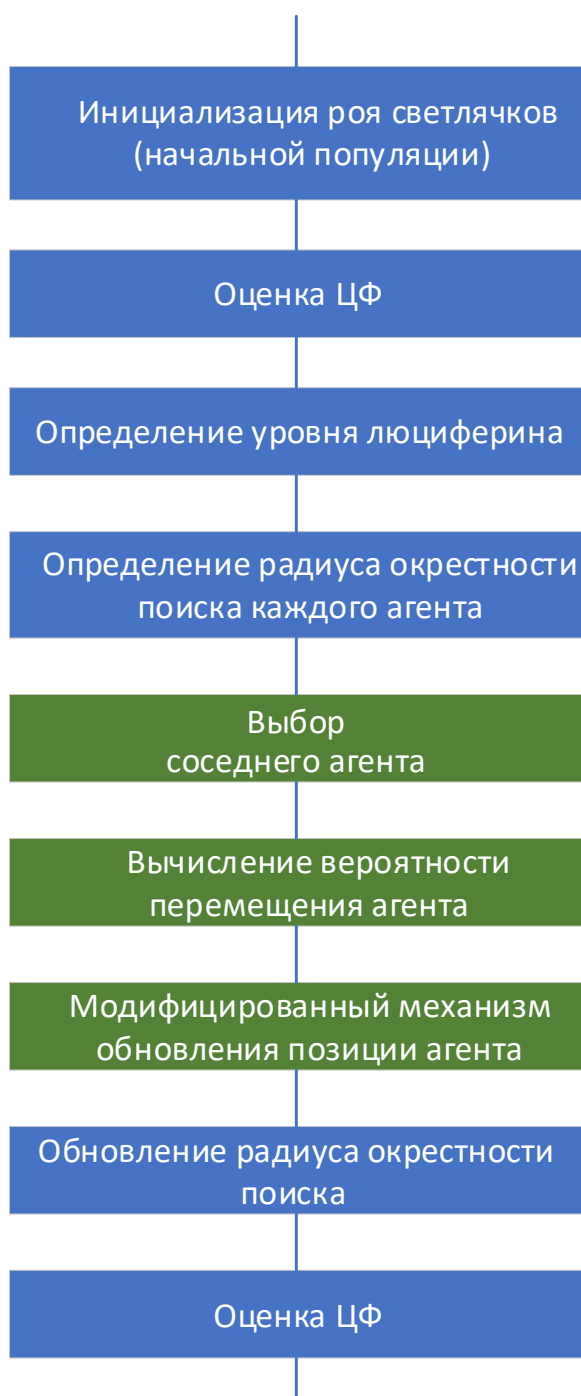


Рисунок 2.8 – Модифицированный метод оптимизации, инспирированный поведением роя светлячков

Шаг 7. Каждый агент для принятия решения об изменении своего положения в пространстве использует исключительно доступную ему локальную информацию. Поэтому целевая функция и, следовательно, количество локальных оптимумов, зависит от радиуса видимости каждого агента.

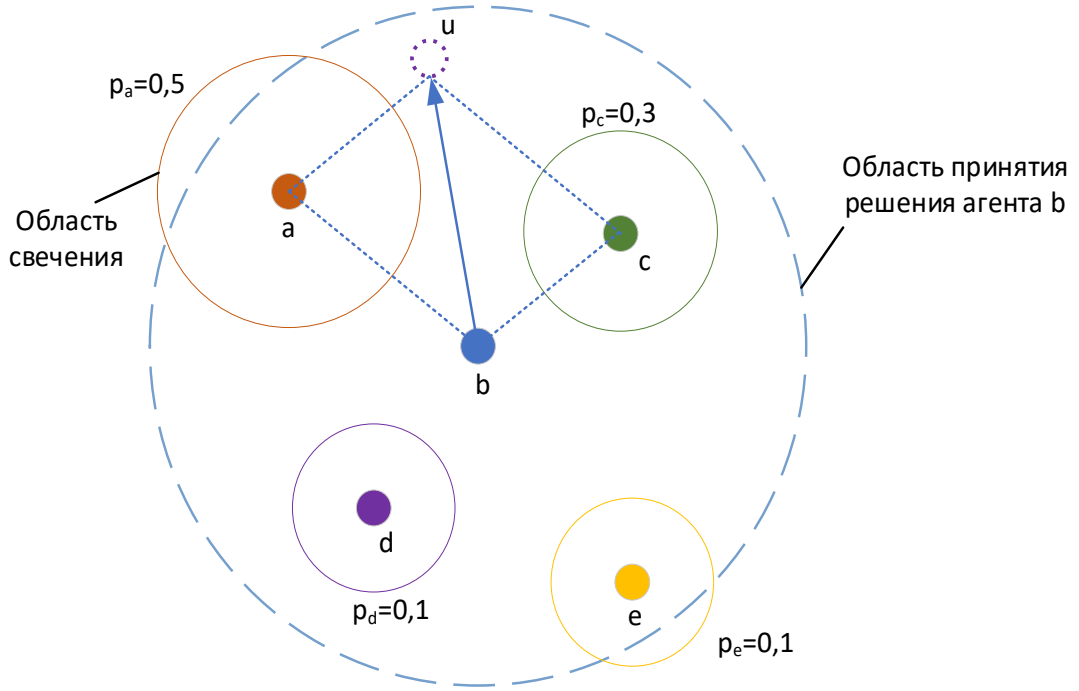


Рисунок 2.9 – Модифицированный механизм обновления позиции агента

Например, если области видимости всех агентов покрывают все пространство поиска, то все агенты перемещаются в точку глобального оптимума, игнорируя при этом локальные. Т. к. априори неизвестно количество локальных оптимумов ЦФ, то, чтобы найти глобальный оптимум, область видимости принимается изменяемым параметром. Т. е. каждый агент i имеет локальную область принятия решений r_d^i , динамически изменяющуюся в пределах $0 < r_d^i \leq r_s^i$. Для вычисления обновленного радиуса окрестности поиска (локальной области принятия решений) вводится явный пороговый параметр n_t для ограничения количества соседей агента на каждой итерации. Таким образом, радиус окрестности поиска r_d^i обновляется по следующей формуле:

$$r_d^i(t+1) = \min \left\{ r_s, \max \{ 0, r_d^i(t) + \beta (n_t - |N_i(t)|) \} \right\} \quad (2.8)$$

где β – const и n_t – параметр, устанавливающий число соседних агентов.

Следует отметить, что предложенный метод оптимизации роем светлячков и алгоритм, разработанный на его основе, имеет достаточно высокую скорость сходимости и высокую вероятность получения набора квази-оптимальных решений за приемлемое время.

2.5 Разработка модифицированного метода генетического поиска

При решении задачи разбиения СБИС для повышения качества решений, полученных на предыдущем этапе комбинированного поиска, автор предлагает использовать модифицированный метод генетического поиска, структура которого приведена на рисунке 2.10.



Рисунок 2.10 – Структура метода генетического поиска

В отличие от классической схемы метода генетического поиска, которая основывается на эволюционной модели Ч. Дарвина, автор использует многогранную теорию эволюции Ламарка, в основе которой лежит понятие, что приобретенные организмом характеристики будут унаследованы его потомками.

Суть процесса эволюции популяции альтернативных решений заключается в возможности «наилучших» решений влиять на состав следующей популяции за счет длительно выживания и большого количества потомков. [93, 94].

Ламарк утверждал, что процесс эволюции происходит постоянно без резких скачков. На основе данного факта он сделал вывод, что отдельных видов в природе не существует, а существуют лишь отдельные особи.

Ламарк выделил два ключевых фактора, которые оказывают влияние на эволюционные изменения животных и растений.

Согласно эволюционной теории Ламарка, первый фактор заключается в стремлении организма к постоянному изменению и улучшению своих признаков. Данный фактор принято называть «стремлением к прогрессу». Все живые организмы Ламарк распределяет на шесть уровней (ступеней или, как он называет, «градаций») в зависимости от сложности их организации. Кроме того, каждому живому организму свойственно стремление и продвигаться вверх по так называемым «ступеням». Таким образом, «законом градаций» - стремлением особи усложнять свою структуру –Ламарк объяснял прогрессивную эволюцию и возникновение более сложных форм.

Второй фактор, который может оказывать влияние на изменение признаков органического мира, состоит в предположение, что условия среды обитания воздействует на живые организмы. Под условиями среды обитания ученый подразумевал тепло, пищу, воздух и т. д. Также Ламарк утверждал, что среда обитания может влиять на органический мир прямо либо косвенно.

Все изменения особи являются последствием влияния окружающего мира и наследования признаков, которые организм приобрел в течение жизни, что, согласно Ламарку, объясняет способность организмов приспосабливаться к условиям окружающей среды.

Ламарк выделял следующие причины направленных изменений:

- способности каждой конкретной особи;
- направленное воздействие условий окружающей среды.

Ламарк утверждал, что каждая особь имеет возможность найти наилучший способ себя «изменить» самостоятельно, т. е. изменчивость является результатом эволюции, а не ее причиной. Именно такой тип эволюции принято рассматривать для моделирования искусственной эволюции, нашедшей применение в науке, т. к. она достаточно эффективна в том случае, когда популяция имеет свойство сходиться в области локального оптимума [94].

С помощью модели Ламарка невозможно однозначно представить процесс эволюции, т. к. она имеет теологический характер, т.е. наибольшая степень приспособленности потомков в популяции является результатом целенаправленной изменчивости родительских особей.

Данная эволюционная теория получила широкое распространение в области решения различных оптимизационных задач, хотя она абсолютно дискредитирована в области биологии. На подобии с естественной эволюцией, системы, разрабатываемые на основе искусственной эволюции, реализуют изменения в фенотипе и генотипе [95]. На основе теории Ламарка возможна организация локального поиска особей среди структур фенотипов.

При решении задачи разбиения СБИС такой вид поиска наиболее целесообразен при поиске локальных оптимумов в популяции решений. Размер популяции является одним из наиболее критичных параметров при реализации генетического поиска, т. к. от его размера напрямую зависит оптимальность найденного решения [96].

Упрощенная схема модели эволюции по Ламарку приведена на рисунке 2.11. В соответствии с моделью эволюции Ламарка, наиболее приспособленные организмы имеют большую вероятность на выживание и закрепление своего фенотипа. К тому же, данный фенотип может стать доминирующим при неизменных условиях окружающей среды.

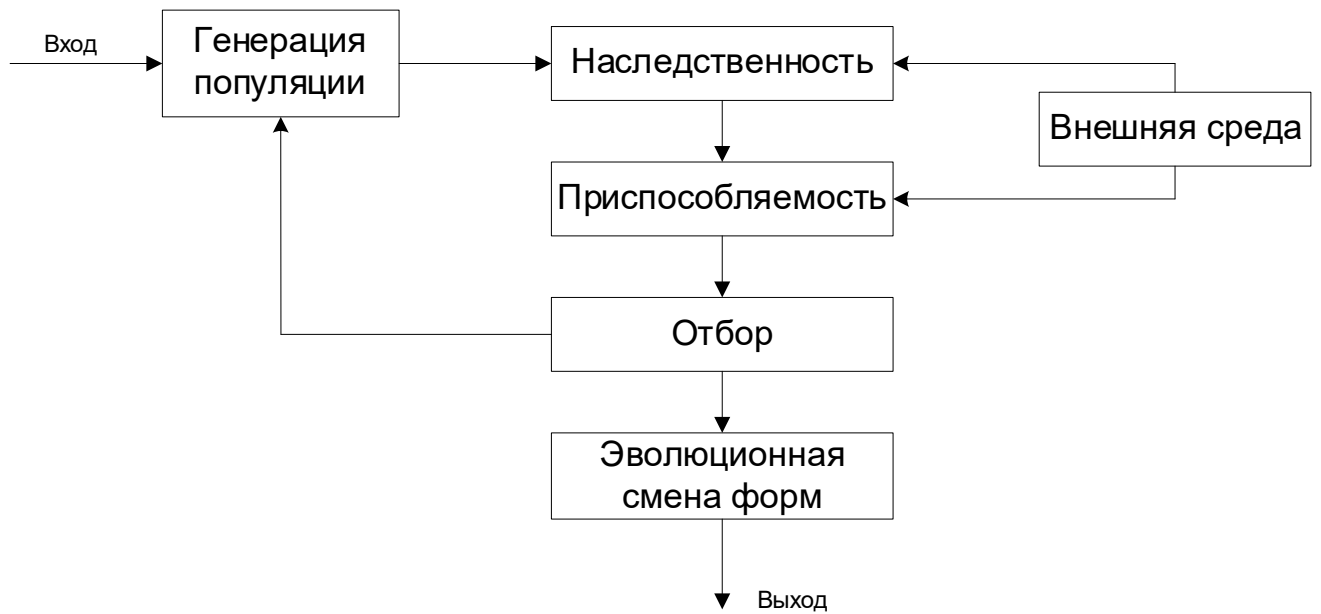


Рисунок 2.11 – Генетический поиск на основе модели эволюции Ламарка.

В диссертационной работе автор разработал модифицированный генетический алгоритм на основе эволюционной модели Ламарка [46], схема которого приведена на рисунке 2.12.

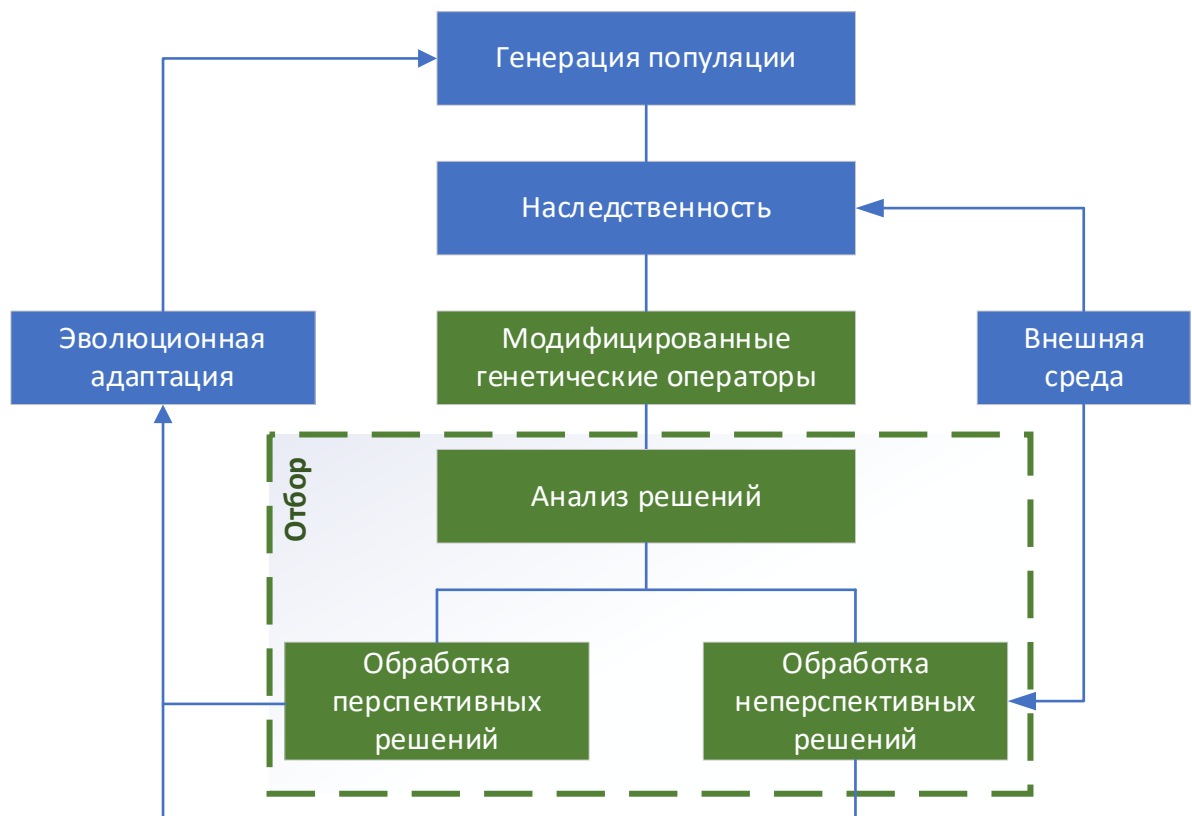


Рисунок 2.12 – Модифицированный метод генетического поиска на основе

модели эволюции Ламарка.

Применительно к задаче разбиения СБИС на этапе «Генерация популяции» генерируется набор альтернативных решений, т. е. вариант разбиения. В блоке «Наследственность» производится анализ набора альтернативных решений.

В качестве модификации классического метода генетического поиска на основе модели эволюции Ламарка автор ввел блок «Модифицированные генетические операторы» и модифицированную стратегию отбора неперспективных решений. Для сокращения альтернативных решений, участвующих в процессе поиска, вводится блок «Модифицированные генетические операторы», где выполняются модифицированные операторы кроссинговера и инверсии. Блок «Анализ решений» направлен на закрепление наследования благоприобретенных признаков, т. е. перспективные решения сразу попадают в блок «Обработка перспективных решений», а затем в следующую итерацию, что также способствует сокращению поискового пространства и уменьшению времени работы алгоритма.

На этапе «Обработка неперспективных решений» происходит преобразование альтернативных решений согласно требованиям конструктора. Далее, для сохранения численности популяции производится отбор лучших вариантов разбиения блоков. В качестве влияния внешней среды будем считать решения, принимаемые конструктором при проектировании. В блоке «Эволюционная адаптация» происходит принятие решения о целесообразности изменения размера текущей популяции.

Разработанный метод позволяет разделять альтернативные решения на перспективные и неперспективные, а также производить в них изменения за счет выполнения модифицированных генетических операторов. Данный метод позволяет сконцентрировать поиск вокруг перспективных решений, что позволяет получать решения высокого качества за приемлемое время. Для решения проблемы предварительной сходимости был введен блок «Эволюционная адаптация», где возможно изменение размера текущей популяции на каждой итерации.

2.6. Построение модифицированных генетических операторов, ориентированных на решение задачи разбиения при проектировании СБИС

Для получение более качественных результатов на этапе уточнения решения, полученного на предыдущей стадии комбинированного поиска, автор разработал модифицированный генетический алгоритм, основанный на эволюционной модели Ламарка.

В качестве модификации ГА в диссертационной работе разработаны новые технологии построения генетических операторов [97].

При реализации предложенного модифицированного ГА первым выполняется оператор репродукции. Оператор репродукции можно считать оптимальным, если он выполняет один из основных принципов бионического поиска, т. е. он способен создать возможность перехода от одного множества альтернативных решений к другому. Под селекцией понимается способ отбора пары хромосом для последующего скрещивания. От интенсивности селекции напрямую зависит скорость сходимости ГА [46]. Выбор вида селекции определяет возможность реализации операторов репродукции, что оказывает влияние на вероятность нахождения оптимального значения ЦФ.

В диссертационной работе реализована селекция на основе «колеса рулетки». Каждой хромосоме ch_i ($i = 1, 2, \dots, N$), где N – количество хромосом в популяции, соответствует сектор колеса рулетки $v(ch_i)$, что задается формулой формуле:

$$v(ch_i) = p_s(ch_i), \tag{2.9}$$

$$p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)}$$

где $F(ch_i)$ – значение ЦФ хромосомы ch_i , $p_s(ch_i)$ – вероятность селекции хромосомы ch_i .

В результате генерируется родительская популяция хромосом, численностью N , размер которой равен размеру текущей популяции.

Оператор кроссинговера (ОК) – это конструкция для получения хромосом-потомков, к основе которой лежит механизм преобразования целых хромосом или их частей из родительской популяции. В связи с тем, что структура ОК значительно влияет на получение квазиоптимальных решений, было разработано множества вариаций данного оператора [98, 99].

В данной диссертационной работе в качестве одного из механизмов генерации новых альтернативных решений выступает оператор кроссинговера, реализующий частично-соответствующий оператор кроссинговера. Механизм его работы заключается в следующем: перед началом работы частично-соответствующего ОК задается точка ОК случайным образом. Далее устанавливается частичное соответствие между родительскими хромосомами и будущими хромосомами-потомками. Т. е. правая часть хромосомы второго родителя P_2 переносится в левую часть хромосомы первого потомка, а левая часть первой родительской хромосомы P_1 переносится в первого потомка P'_1 с заменой повторяющихся генов на отсутствующие, которые находятся в частичном соответствии. Второй потомок P'_2 получается аналогичным образом [46]. Пример работы частично-соответствующего ОК приведен на рисунке 2.13.

P_1 : A B C D E F	G H I J
P_2 : E C I A D H	J B F G
P'_1 : A H C D E I	J B F G
P'_2 : E C F A D B	G H I J

Рисунок 2.13 – Пример частично-соответствующего ОК

Определение числа точек разрыва в хромосоме и их месторасположение значительно влияют на качество получаемых потомков: большое число точек

разрыва приводит к потере качественных альтернативных решений, в то время как, малое их число может привести в преждевременной сходимости алгоритма в локальном оптимуме.

С целью нахождения компромисса автор диссертационной работы предлагает для определения точек разрыва при выполнении ОК использовать теорию прямоугольных чисел [100]. Под прямоугольным числом понимается число, представляющее собой произведение двух последовательных целых чисел. Следовательно, последовательность прямоугольных чисел T_n можно задать следующим образом

$$T_n = n(n + 1), n = 1, 2, 3, \dots \quad (2.10)$$

Графически прямоугольные числа можно интерпретировать как количество элементов, расположенные в форме прямоугольника, как проиллюстрировано на рисунке 2.14.

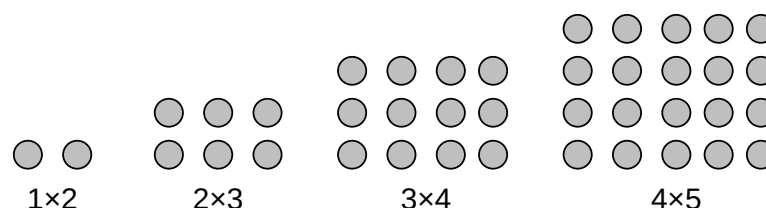


Рисунок 2.14 – Последовательность прямоугольных чисел

Рассмотрим более подробно реализацию модифицированного ОК, точки разрыва которого задаются с помощью теории прямоугольных чисел. При выборе длины последовательности чисел важно, чтобы последнее прямоугольное число было меньше длины альтернативного решения, т. е. при длине хромосомы $L=15$ последнее прямоугольное число будет равно 12.

Для демонстрации работы модифицированного ОК приведем пример его реализации для хромосомы с длиной 15 ($L=15$). Согласно теории прямоугольных чисел, точки кроссинговера располагаются после 2, 6 и 12 генов. Далее выполняется механизм частично-соответствующего оператора кроссинговера для получения хромосом-потомков. Пример реализации модифицированного ОК продемонстрирован на рисунке 2.15.

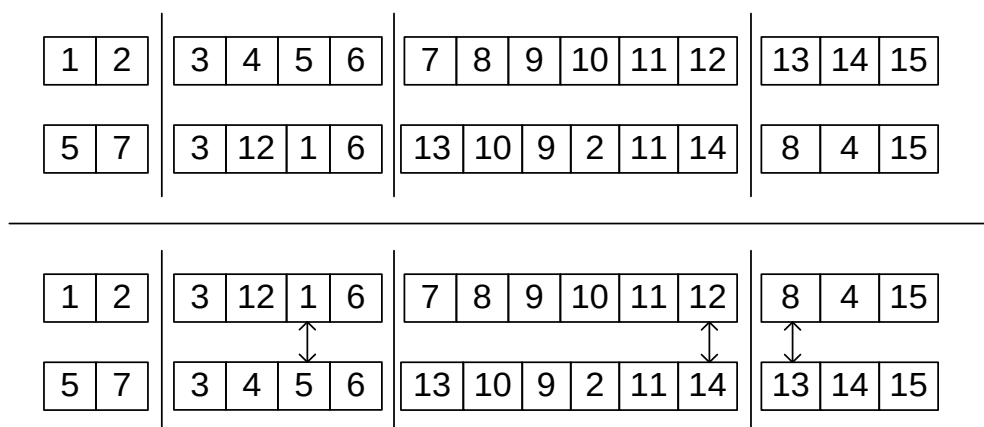


Рисунок 2.15 – Формирование хромосом-потомков в результате выполнения модифицированного ОК

После выполнение перестановок строительных блоков в хромосомах-родителях получаем нелегальное решение, т. к. дублируются гены 5, 12 и 13. Поэтому применяется механизм коррекции полученного решения путем перестановки дублирующихся генов в хромосомах-потомках. Результат выполнения модифицированного ОК после коррекции представлен на рисунке 2.16.

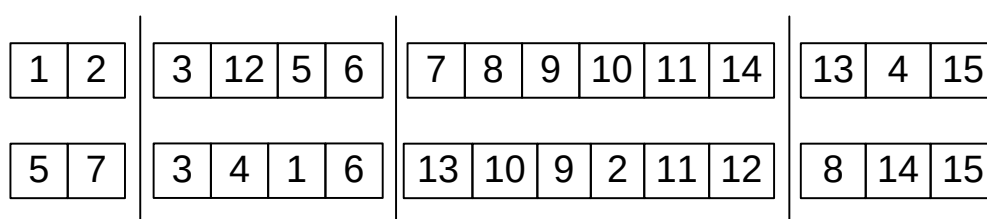


Рисунок 2.16 – Результат выполнения модифицированного ОК

При реализации генетического алгоритма важную роль играет *оператор мутации* (ОМ). Процесс мутационной изменчивости заключается в направленном изменении генотипа организма, при котором изменяется количество хромосом, их структура или строение, и непосредственно связан с возникновением мутаций. Под мутацией понимается изменение генотипа, которое приводит к новому проявлению генетических свойств за счет внезапных изменений в его структуре.

В диссертационной работе для решения задачи разбиения схем при проектировании СБИС наиболее целесообразно рассматривать модели мутаций, связанные с изменением генотипа. В алгоритмах эволюционного поиска, направленных на решение задач в области САПР, процесс мутации интерпретируется как генерация нового множества решений задачи. Затем инициализируется эволюционный поиск в исходной и новой популяциях альтернативных решений. В парадигме генетических алгоритмов под мутацией понимают конструкцию, которая позволяет получать хромосомы-потомки путем преобразования хромосомы-родителя (или ее части) [46].

При решении задачи разбиения СБИС целесообразно использовать знания о поставленной задаче, поэтому при реализации ОМ точки мутации выбираются направленно.

Одним из видов ОМ является оператор инверсии (ОИ), который выполняется путем инвертирования последовательности геном в хромосоме внутри точек разрыва. В диссертационной работе предложен новый гибридный механизм инверсии на основе распределения Больцмана, согласно которому в условиях термодинамического равновесия распределение частиц идеального газа зависит от значений потенциальной энергии. При высокой температуре молекулы газа распределены по высоте равномерно, а с течением времени с уменьшением температуры количество молекул на высотах сокращается.

Автор предлагает интерпретировать закон распределения Больцмана следующим образом. Пусть N – длина хромосомы P , I – количество итераций, i – номер текущей итерации. Тогда длина отрезка инвертирования будет определяться по формуле

$$L = N(1 - i/I). \quad (2.10)$$

На рисунке 2.17 приведен пример реализации оператора мутации при $N=10$, $I=100$.

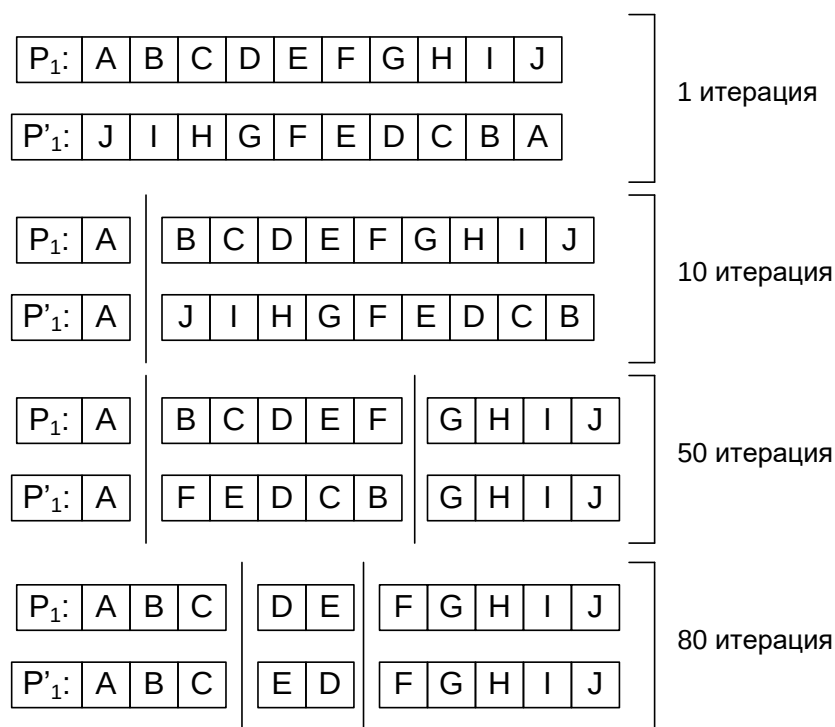


Рисунок 2.17 – Пример выполнения модифицированного оператора мутации.

Так на 1-й итерации ($i=1$) длина отрезка инвертирования определяется следующим образом:

$$L_1 = 10 \left(1 - \frac{1}{100} \right) = 9,9 \approx 10 \quad (2.11)$$

На 10-й итерации ($i=10$)

$$L_{10} = 10 \left(1 - \frac{10}{100} \right) = 9 \quad (2.12)$$

На 50-й итерации ($i=50$)

$$L_{50} = 10 \left(1 - \frac{50}{100} \right) = 5 \quad (2.13)$$

На 80-й итерации ($i=80$)

$$L_{80} = 10 \left(1 - \frac{80}{100} \right) = 2 \quad (2.14)$$

В заключение подраздела следует отметить, что разработанные модифицированные операторы эволюционного и генетического поиска направлены на решение задачи разбиения схем при проектировании СБИС и позволяют уменьшить интервал неопределенности, таким образом сокращая размер области поиска, а также могут квазиоптимально применяться при моделировании параллельного поиска.

2.7 Краткие выводы

1. В диссертационной работе приведены основные положения теории биоинспирированного поиска оптимальных решений, а также целесообразность использования популяционных алгоритмов для решения задачи разбиения схем при проектировании СБИС.

2. Предложена комбинированная архитектура биоинспирированного поиска, сочетающего в себе бактериальный и светлячковый оптимизационные методы, а также генетический алгоритм, используемый в качестве метода повышения качества решения, полученного на предыдущих этапах.

3. Разработан модифицированный метод поиска оптимальных решений на основе моделирования перемещения колонии бактерий, позволяющий сократить время поиска за счет параллельной работы нескольких агентов. Помимо этого, бактериальный метод обладает свойством расширяемости, т. е. позволяет вводить новые операторы в процессе синтеза и анализа проектных решений.

4. Разработан метод оптимизации, инспирированный поведением роя светлячков, который основан на создании динамической области принятия решений каждого агента, что гарантирует нахождение всех локальных оптимумов ЦФ за полиномиальное время. Предложенная автором модификация механизма обновления позиции агента на основе учета уровня люциферина двух соседних агентов позволяет избежать предварительной сходимости алгоритма и попадания в локальный оптимум.

5. Разработаны модифицированные генетические операторы, основанные на прямоугольных числах и законе распределения Больцмана, ориентированные на решение задачи разбиения схем при проектировании СБИС, которые позволяют уменьшить область неопределенности и направить поисковый процесс в сторону квазиоптимальных решений. Помимо этого, модифицированные ГО эффективно применяются при организации параллельного процесса поиска.

6. В диссертационной работе для повышения качества решений, полученных на предыдущем этапе поиска, был разработан модифицированный

генетический алгоритм, основанный на модели эволюции Ламарка. Его отличительной особенностью является использование модифицированных генетических операторов, что позволяет сократить интервал неопределенности и сузить поисковое пространство.

3. РАЗРАБОТКА МОДИФИЦИРОВАННЫХ АЛГОРИТМОВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС

3.1 Разработка унифицированного механизма представления решений при разбиении схем

Все биоинспирированные алгоритмы нуждаются в строгой формализации входных данных, что зачастую проблематично ввиду большого их количества. Поэтому при использовании биоинспирированных алгоритмов для решения задач на уровне конструкторского проектирования одной из основных проблем является механизм представление альтернативных решений [101]. Для ускорения процесса кодирования/декодирования решений, а также расчета целевой функции в данной диссертационной работе предлагается использовать единый подход к представлению данных для всех этапов работы многоуровневого алгоритма разбиения схем при проектировании СБИС. Автором предлагается кодировать решения в виде последовательности обработки элементов схемы. Альтернативное решение (хромосома) представляет собой последовательность номеров элементов СБИС, разбитых на группы. Например, разбиение схемы, состоящей из 10 элементов на 2 равные группы разбиения имеет вид, изображенный на рисунке 3.1:

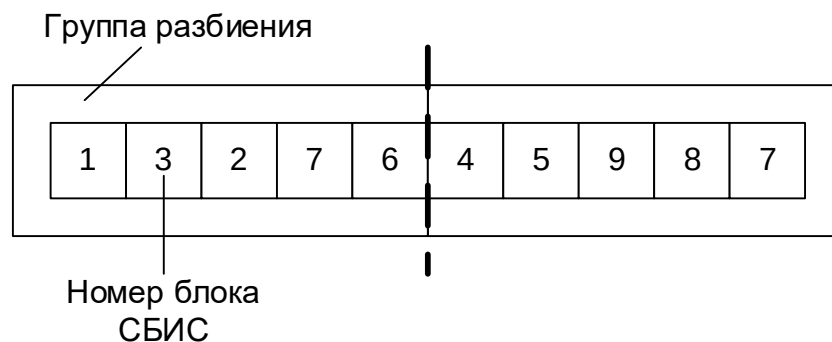


Рисунок 3.1 – Пример альтернативного решения

Рассмотрим более сложный пример. Предположим, что необходимо закодировать разбиение схемы, состоящей из 20 блоков и 4 групп разбиения

разного размера. Группа 1 содержит 5 элементов, группа 2 – 2 элемента, группа 3 – 10 элементов, группа 4 – 3 элемента. В данном случае решение будет иметь вид, представленный на рисунке 3.2.

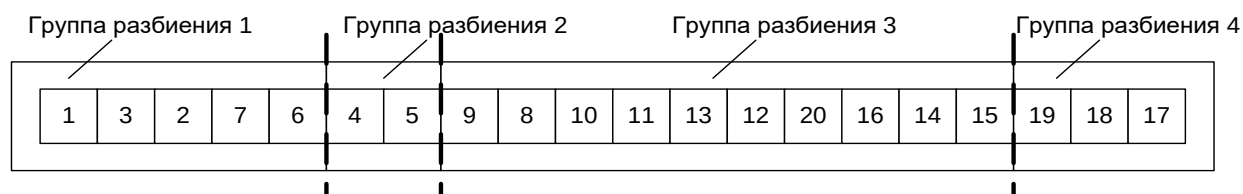


Рисунок 3.2 – Пример кодирования альтернативного решения

В данном примере в Группу 1 входят блоки с номерами 1, 3, 2, 7, 6; в группу 2 – блоки 4, 5; в группу 3 – блоки 9, 8, 10, 11, 13, 12, 20, 16, 14, 15 и в группу 4 – 19, 18, 17. Количество и размер каждой группы разбиения задаются лицом, принимающим решение в соответствии с техническим заданием.

Для иллюстрации всего процесса кодирования альтернативных решений в предлагаемом многоуровневом алгоритме разбиения схем при проектировании СБИС рассмотрим транзисторную схему, изображенную на рисунке 3.3. В качестве критерия рассмотрим количество межблочных соединений.

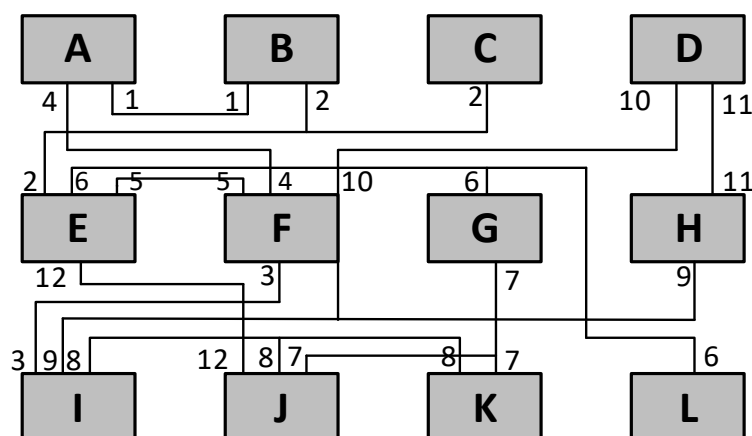


Рисунок 3.3 – Пример схемы

Предположим, что на основании технического задания необходимо разбить схему на 3 блока разбиения.

Представим данную схему в виде гиперграфа, изображенного на рисунке 3.4. Здесь, вершины графа $x_1 - x_{12}$ задают элементы схемы, а гиперребра $u_1 - u_{12}$ – цепи, соединяющие элементы схемы.

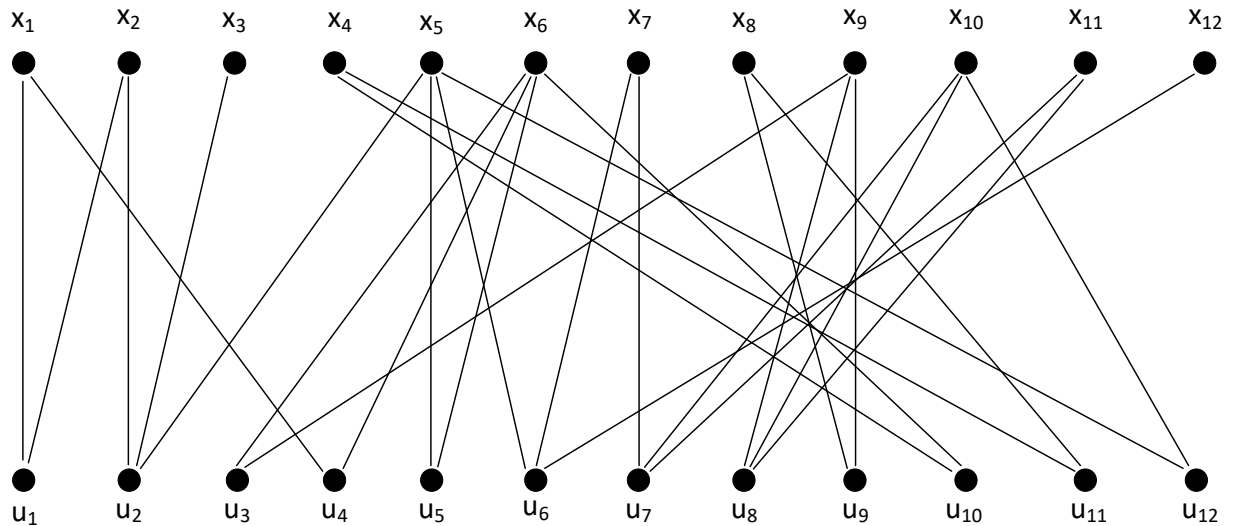


Рисунок 3.4 – Гиперграф

Как известно, для инициализации любого популяционного алгоритма требуется генерация начальной популяции решений. В данном случае начальное разбиение примет вид, предложенный на рисунке 3.5.

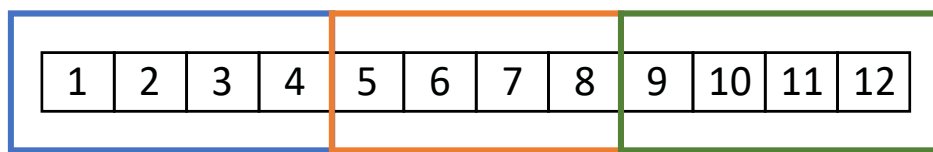


Рисунок 3.5 – Закодированное начальное разбиение схемы.

Первый блок разбиения включает вершины $x_1 - x_4$, второй блок – вершины $x_5 - x_8$, а третий блок – $x_9 - x_{12}$. Наглядное представление разбиения вершин и подсчета ЦФ начального разбиения представлено в виде двудольного графа на рисунке 3.6.

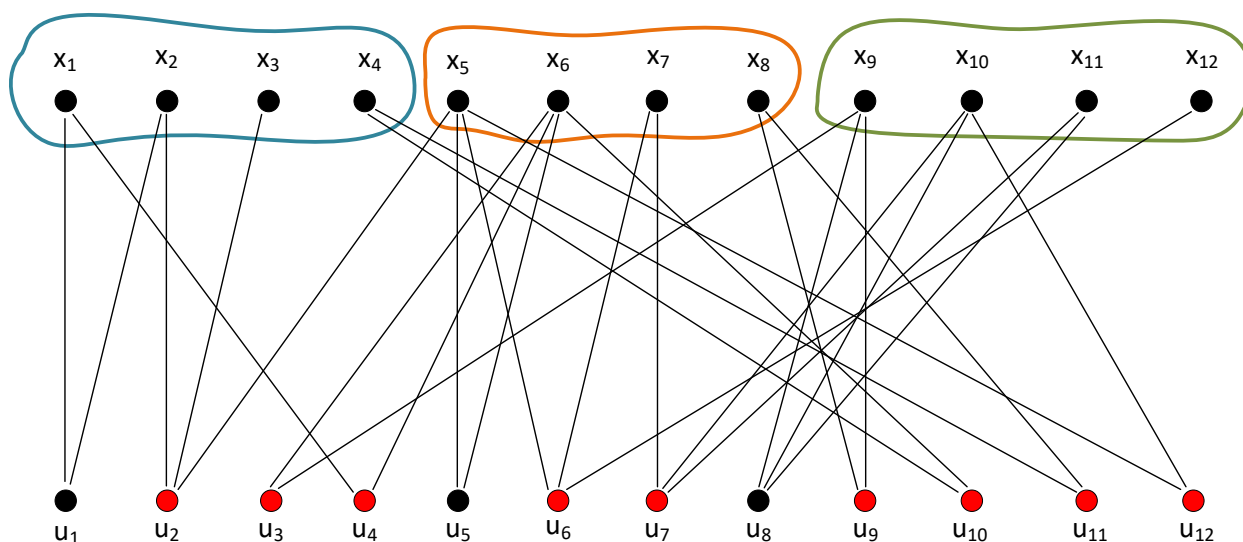


Рисунок 3.6 – Разбиение гиперграфа на 3 части

Из рисунка видно, что гиперребро u_1 соединяет вершины x_1 и x_2 , которые расположены в одном блоке разбиения. Гиперребро u_2 соединяет вершины x_1 и x_3 из первого блока разбиения и вершину x_5 , которая расположена во втором блоке разбиения. В этом случае получаем разрез гиперребра u_2 . Проанализировав оставшиеся гиперребра, получим 9 разрезанных ребер ($u_2, u_3, u_4, u_6, u_7, u_9, u_{10}, u_{11}, u_{12}$), т.е. ЦФ = 9.

После работы всех алгоритмов разбиения СБИС, которые подробно будут описаны позже, получим решение, представленное на рисунке 3.7.

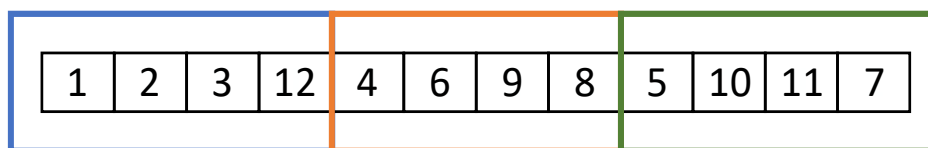


Рисунок 3.7 – Закодированное разбиение схемы после разбиения.

После выполнения алгоритмов разбиения гиперграф примет вид, представленный на рис. 3.8. После разбиения количество разрезов сократилось и составило ЦФ=6.

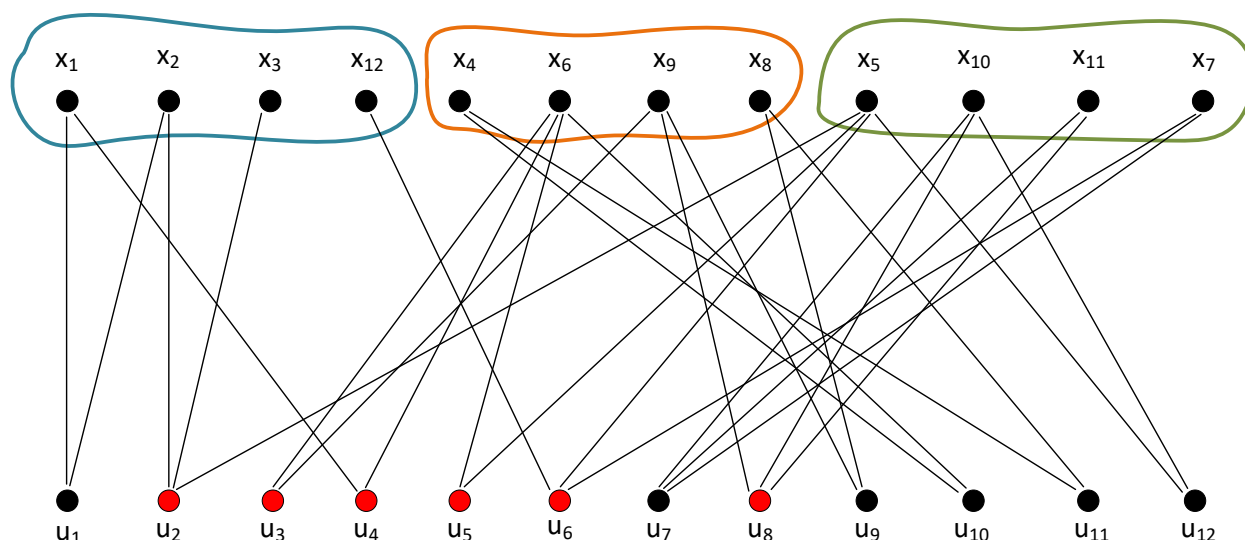


Рисунок 3.8 – Разбиение графа после выполнения алгоритма разбиения.

Следует отметить, что разработанный механизм кодирования и декодирования решений представляет собой унифицированный подход к представлению данных при реализации модифицированных бактериального, светлячкового и генетического алгоритмов в рамках многоуровневого алгоритма разбиения схем при проектировании СБИС. Кроме того, кодирование схемы в виде последовательности блоков позволяют сократить потери процессорного времени при передаче данных между различными уровнями многоуровневого алгоритма разбиения схем.

3.2 Разработка модифицированного бактериального алгоритма разбиения схем при проектировании СБИС

В рамках данной диссертационной работы автор предлагает модифицированный алгоритм разбиения схем, работающий на основе модели поведения колонии бактерий, впервые примененный для решения задачи разбиения схем при автоматизированном проектировании СБИС. Разработанный алгоритм включает следующие шаги: ввод настроечных параметров, генерация начальной популяции альтернативных решений, хемотаксис (скрещивание), репродукция и уничтожение [102, 103].

Управляющие параметры модифицированного бактериального алгоритма разбиения СБИС задаются разработчиком при инициализации алгоритма и включают в себя:

- P – количество агентов в популяции;
- I – общее число итераций;
- H_0 – значение целевой функции, интерпретируемое как начальное значение здоровья агента;
- N – общее число шагов хемотаксиса,
- $N_{\max}$ и $N_{\min}$ – максимальное и минимальное количество шагов хемотаксиса соответственно.

Схема работы модифицированного бактериального алгоритма разбиения схем при проектировании СБИС представлена на рисунке 3.9.

Входными данными является закодированное множество решений, полученных с помощью генератора случайных решений. После этого производится расчет значений целевых функций альтернативных решений.

Динамическое изменение количества шагов хемотаксиса автор предлагает интерпретировать как изменение количества парных перестановок на этапе хемотаксиса. Количество шагов хемотаксиса на текущей итерации определяется по формуле:

$$N = (N_{\max} - N_{\min}) \left(\frac{I - I_{cur}}{I} \right) + N_{\min} \quad (3.1)$$

где I_{cur} – номер текущей популяции.

После этого определяется стартовый индекс гена в хромосоме, который определяет начало движения для агента. Движение возможно в двух направлениях: в сторону уменьшения индекса гена и в сторону его увеличения. Изначально, агент движется в сторону увеличения индекса гена.

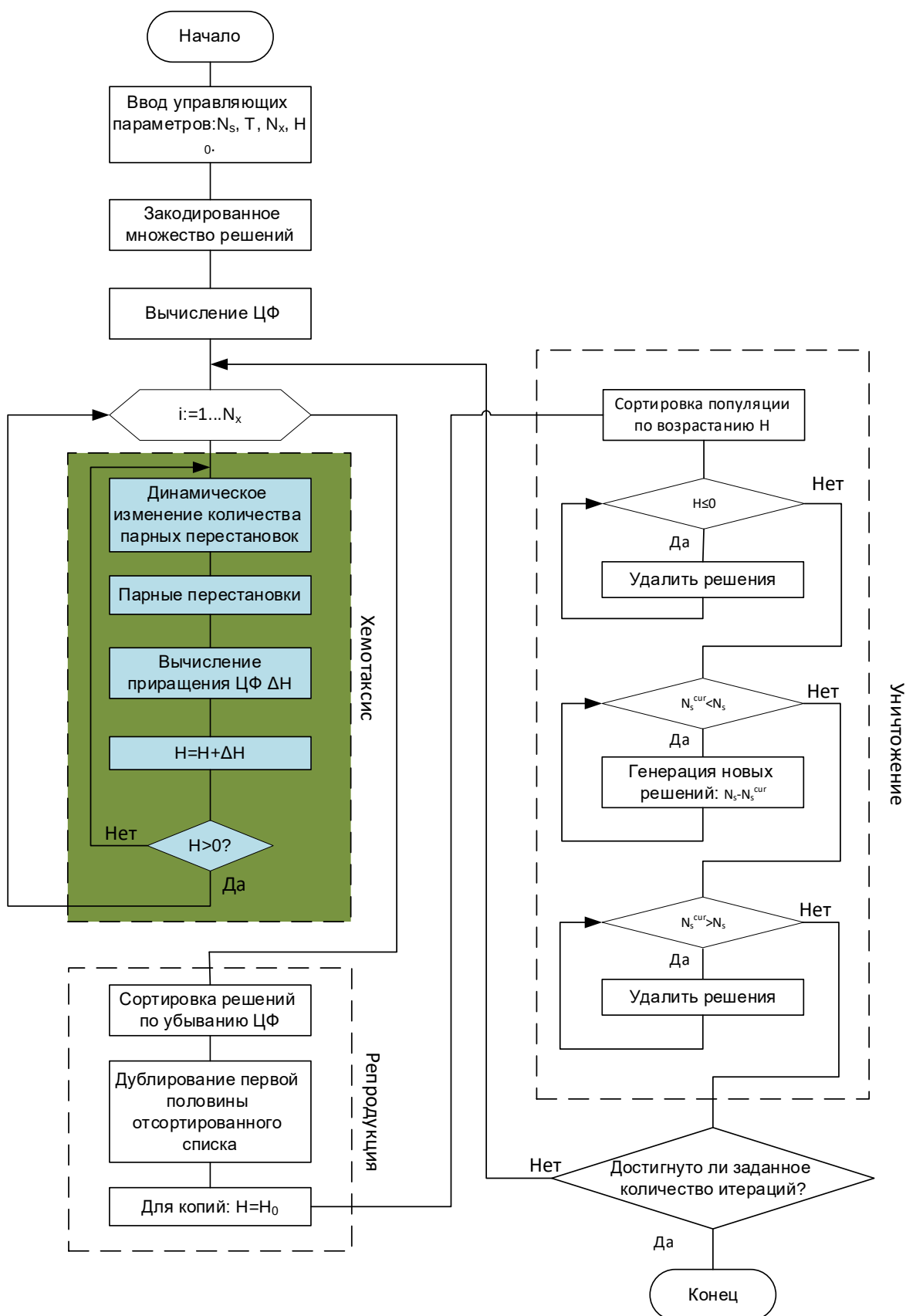


Рисунок 3.9 – Схема работы алгоритма разбиения СБИС на основе поведения колонии бактерий

Для обозначения расстояния (количества генов) между начальным и конечным индексами генов введем понятие шага агента. В том случае, если агент достигает первого или последнего гена хромосомы, то конечный индекс вычисляется циклически.

На рисунке 3.10 изображен пример движения агента в хромосоме с начальным индексом $I = 2$ с шагом равным 5.

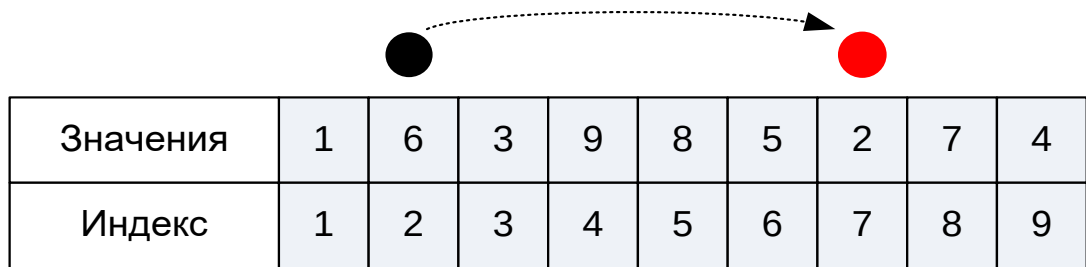


Рисунок 3.10 – Пример передвижения агента

На рисунке 3.11 приведено движение агента с начальным индексом 7 шагом 5.

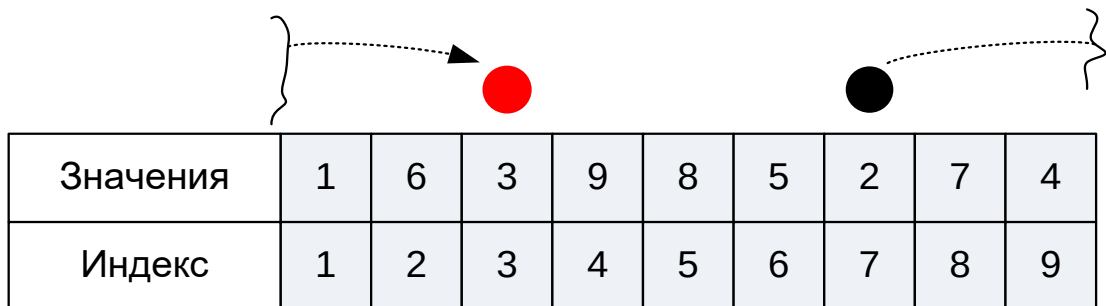


Рисунок 3.11 – Пример передвижения агента

В результате движения агента имеем два гена для последующей парной перестановки. На следующем этапе вычисляется приращение ЦФ ΔH до и после окончания перестановки генов, которое прибавляется к текущему значению здоровья H_0 , т. е.

$$H_1 = H_0 + \Delta H \quad (3.2)$$

В диссертационной работе здоровье каждого агента будет соответствовать общему приращению ЦФ на этапе выполнения хемотаксиса. При этом вводится требование не отрицательности значения здоровья агента. Если значение здоровья

агента принимает отрицательное значение, то перемещение агента останавливается и процесс поиска завершается.

Для предотвращения подобных ситуаций введена следующая особенность агента: он может с определенной долей вероятности изменять направление своего движения по генам в случае получения отрицательного приращения ЦФ. Данное свойство агента моделирует концепцию уклонения бактерии от репеллентов в естественной среде обитания.

Для наглядности приведем пример выполнения процесса хемотаксиса для одного из альтернативных решений. Входными параметрами являются:

- количество шагов агента варьируется в пределах отрезка $[2;5]$,
- количество шагов хемотаксиса - $N_x = 3$,
- начальное значение здоровье агента - $H_0=5$.
- Начальное значение ЦФ - $F_0 = 20$.

Начальное альтернативное решение приведено на рисунке 3.12.

Значения	8	3	9	4	6	5	2	7	1
Индекс	1	2	3	4	5	6	7	8	9

Рисунок 3.12 – Начальное альтернативное решение

Для начала движения агента случайным образом определяем начальный индекс $I_0 = 1$.

Первый шаг хемотаксиса:

1. Случайным образом генерируем агента $STEP_{cur} = 4$.
2. Вычисляем конечный индекс агента. $I_e = I_0 + STEP_{cur} = 1 + 4 = 5$.
3. Выполняем парную перестановку первого и пятого генов хромосомы.

На рисунке 3.13 приведена хромосома после первого шага хемотаксиса.

Значения	6	3	9	4	8	5	2	7	1
Индекс	1	2	3	4	5	6	7	8	9

Рисунок 3.13 – Представление решения после 1-го шага хемотаксиса

Значение ЦФ - $F_1 = 18$, а приращение ЦФ - $\Delta F_{01} = 2$. Прибавляем к текущему значению здоровья приращение ЦФ - ΔF_{01} . Тогда здоровье текущего агента принимает значение $H_1 = 7$. Принимая во внимание, что $\Delta F_{01} > 0$, следует учесть, направление передвижения агента остается неизменным на следующем шаге процесса хемотаксиса.

Второй шаг хемотаксиса выполняется следующим образом. Сначала генерируется шаг текущего агента $STEP_{cur} = 3$. Затем получаем конечный индекс агента $I_e = 8$. После этого производим парную перестановку выбранных генов (пятого и восьмого).

На рисунке 3.14 представлено решение после второго шага хемотаксиса.

Значения	6	3	9	4	7	5	2	8	1
Индекс	1	2	3	4	5	6	7	8	9

Рисунок 3.14 – Решение после 2-го шага хемотаксиса

В результате получаем $F_2 = 17$ и $\Delta F_{12} = 1$. Таким образом, здоровье агента после второго шага хемотаксиса стало равным $H_2 = 7 + 1 = 8$.

Третий шаг хемотаксиса.

1. Генерируем шаг агента $STEP_{cur} = 2$.
2. Согласно правилу циклического перехода получаем $I_e = 1$.
3. Выполняем парную перестановку для восьмого и первого генов.

На рисунке 3.15 представлено решение после третьего шага хемотаксиса.

Значения	8	3	9	4	7	5	2	6	1
Индекс	1	2	3	4	5	6	7	8	9

Рисунок 3.15 – Решение после 3-го шага хемотаксиса

В результате получаем значение ЦФ $F_3 = 19$, приращение ЦФ $\Delta F_{23} = -2$, обновленное значение здоровья агента $H_3 = 8 + (-2) = 6$. Т.к. $\Delta F_{23} < 0$, то на следующем шаге направление движения агента изменится на противоположное.

Затем после выполнения процесса хемотаксиса инициализируется процесс репродукции для каждого альтернативного решения. Основные шаги этапа репродукции приведены на рисунке 3.16:

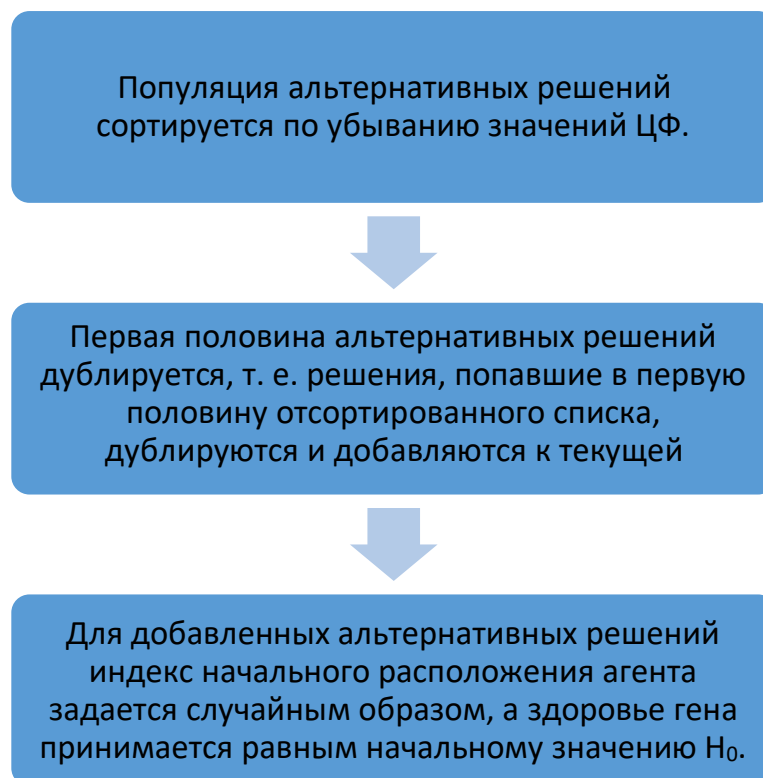


Рисунок 3.16 – Основные шаги этапа репродукции

После выполнения всех процедур этапа репродукции необходима инициализация процесса ликвидации для поддержания постоянного размера популяции. Основные шаги этапа ликвидации приведены на рисунке 3.17:

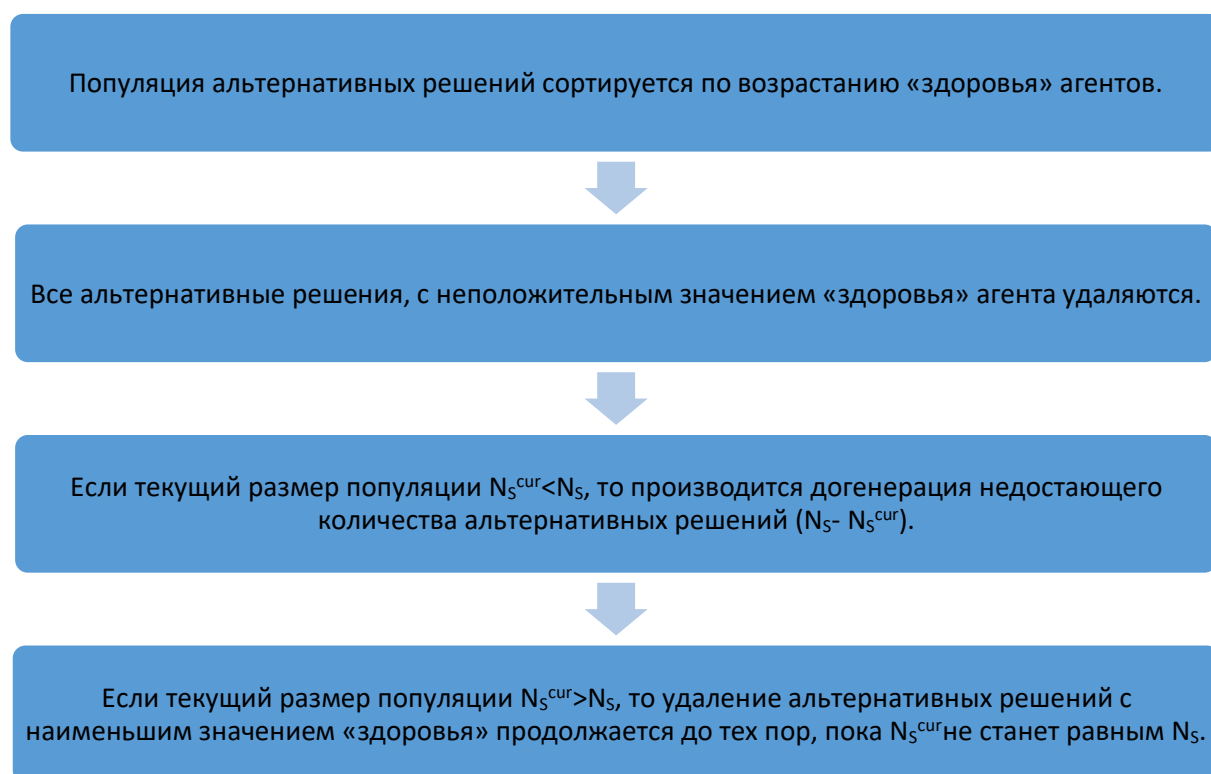


Рисунок 3.17 – Основные шаги этапа ликвидации

Затем процесс поиска продолжается итерационно по достижению критерия останова – предварительно заданного количества итераций алгоритма [102].

Разработанный в диссертационной работе модифицированный алгоритм бактериальной оптимизации, направленный на решение задачи разбиения схем при проектировании СБИС, отличается простотой выполнения, что способствует сокращению длительных итерационных вычислений. Модификация разработанного алгоритма заключается в динамическом изменении количества шагов хемотаксиса, т. е. количества парных перестановок, в зависимости от номера текущей итерации, за счет чего также производится уменьшение временных затрат при проектировании.

3.3 Разработка модифицированного светлячкового алгоритма разбиения схем при проектировании СБИС

В рамках данной диссертационной работы автор предлагает модифицированный алгоритм разбиения схем при проектировании СБИС,

работающий на основе поведения роя светлячков в естественной среде обитания [104-107].

Модифицированный светлячковый алгоритм разбиения СБИС включает следующие основные шаги:

- ввод управляющих параметров,
- инициализация множества альтернативных решений,
- расчет вероятностей выбора агентов для скрещивания на основании уровня люциферина,
- селекция, скрещивание (кроссинговер)
- отбор.

Схема работы алгоритма разбиения, основанного на поведении колонии светлячков, приведена на рисунке 3.18.

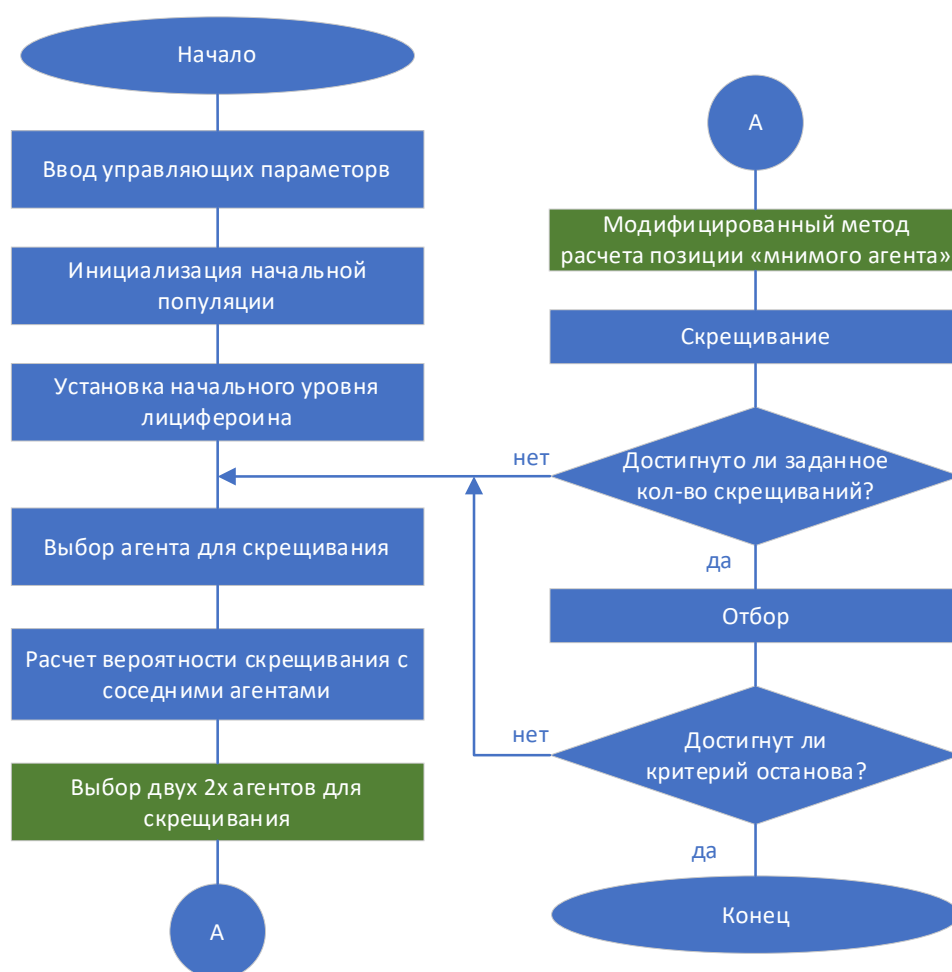


Рисунок 3.18 – Схема работы алгоритма разбиения СБИС, основанного на поведении колонии светлячков

В качестве основного механизма генерации новых альтернативных решений выступает оператор скрещивания, реализующий частично-соответствующий оператор кроссинговера. Механизм работы частично-соответствующего оператора кроссинговера заключается в следующем: перед началом работы частично-соответствующего ОК определяется место расположения разрезающей точки. Затем анализируются части обеих хромосом и производится формирование хромосом-потомков с установлением частичного соответствия между генами родителей. Механизм выполнения частично-соответствующего ОК был описан ранее.

На рисунке 3.19 представлен пример работы данного оператора кроссинговера. Его использование обусловлено тем, что в результате его работы получаются негомологичные альтернативные решения.

P ₁ :	1	2	3	4	5	6	7	8	9	10
P ₂ :	5	3	7	1	4	10	8	2	9	6
P' ₁ :	1	7	3	4	5	10	8	2	9	6
P' ₂ :	5	3	2	1	4	6	7	8	9	10

Рисунок 3.19 – Частично-соответствующий оператор кроссинговера

Модификация состоит в реализации метода выбора пары хромосом для скрещивания. После инициализации всех решений и расчете их целевых функций каждому из них присваивается начальный уровень люциферина равный L_0 . Далее выбирается произвольная хромосома i из популяции и для нее находится пара для скрещивания на основании вероятности, которая рассчитывается по формуле:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (3.3)$$

где $l_i(t)$ и $l_j(t)$ уровень люциферина i -ого и j -ого решения соответственно. После проведения всех расчетов производится выбор пары для скрещивания, на

основании известного механизма отбор на основе «Колесо рулетки» с использованием полученных вероятностей. После проведения заданного количества скрещиваний производится редукция особей популяции до первоначального размера. По окончании итерации пересчитываются уровни люциферина для каждого альтернативного решения по формуле:

$$l_j(t+1) = (1 - \rho)l_j(t) + \gamma J_j(t+1), \quad (3.4)$$

где ρ - коэффициент ослабления уровня люциферина для моделирования процесса распада флюоресцирующего вещества ($0 < \rho < 1$), γ - коэффициент привлекательности светлячка, J_j - значение ЦФ j -ого агента в момент времени t .

Поиск продолжается итерационно до достижения заданного количества итераций.

Приведем пример расчета позиции «мнимого агента» и последующего скрещивания с ним. На рисунке 3.20 изображено положение агентов a, c и u , «мнимого агента» ac , а также p_a и p_c - вероятности скрещивания агента u с агентами a и c соответственно.

Для корректного расчета позиции «мнимого агента» первоначально необходимо произвести нормализацию значений вероятностей скрещивания p_a и p_c по следующим формулам:

$$\begin{aligned} p'_a &= \frac{p_a}{p_a + p_c}; \\ p'_c &= \frac{p_c}{p_a + p_c}. \end{aligned} \quad (3.5)$$

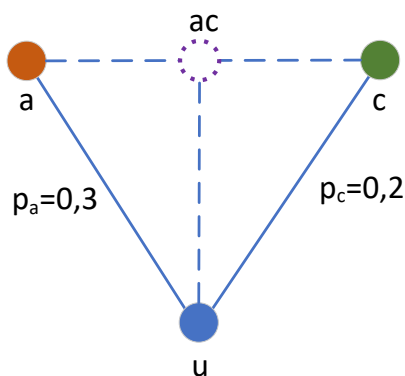


Рисунок 3.20 – Пример расчета позиции «мнимого агента»

Затем, сформируем «мнимого агента» ac на основе частично-соответствующего оператора кроссинговера. Точка разреза CUT определяется следующим образом:

$$CUT = (length(a), length(c)) * p'_a, \quad (3.6)$$

где $length(a)$ и $length(c)$ – длина решения (агента) a и c соответственно.

В рамках решения задачи разбиения схемы СБИС все решения имеют равную длину, т. е. $length(a) = length(c) = length$, то точка разреза CUT вычисляется по формуле:

$$CUT = length * p'_a. \quad (3.7)$$

Пусть $length = 5$, $p'_a = 0,6$. Тогда $CUT = 3$.

Выполним частично-соответствующий оператор кроссинговера для формирования «мнимого агента» ac , как продемонстрировано на рисунке 3.21.

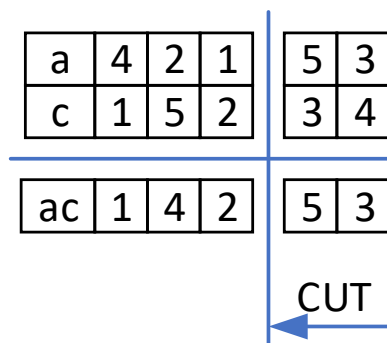


Рисунок 3.21 – «Мнимый агент» ac

Затем скрещиваем полученного «мнимого агента» ac с агентом u также с помощью частично-соответствующего ОК. Здесь точка разреза определяется случайным образом. Пример скрещивания «мнимого агента» ac и агента u приведен на рисунке 3.22.

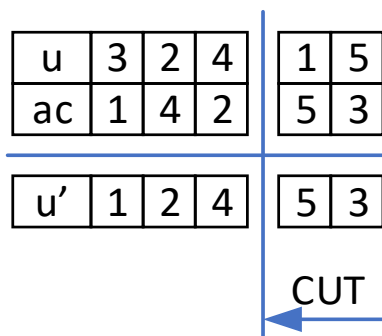


Рисунок 3.22 - Скрещивание «мнимого агента» ac и агента u

Отметим, что разработанный алгоритм позволяет выделять в пространстве поиска области с перспективными решениями, т. е. в высоком значении ЦФ, что способствует существенному сокращению времени работы алгоритма. Предложенный модифицированный метод выбора пары хромосом для последующего скрещивания позволяет частично избежать преждевременной сходимости разработанного алгоритма.

3.4 Разработка модифицированного генетического алгоритма разбиения при проектировании СБИС

В диссертационной работе с целью повышения качества решений предлагается применить модифицированный генетический алгоритм, который с некоторой долей вероятности позволит оптимизировать результаты, полученные с помощью модифицированных бактериального и светлячкового алгоритмов, на предыдущем этапе поиска. Модифицированный генетический алгоритм реализует модель эволюции естественного отбора, которая заключается в том, что особи с большей степенью приспособленности с большей долей вероятности попадают в следующую популяцию.

Основной идеей ГА является эволюционное преобразование некоторой популяции альтернативных решений. В природе главная цель эволюционных изменений заключается в стремление к наилучшей жизнеспособности и приспособленности популяции к условиям среды. В парадигме ГА целью эволюции является получение множества хромосом с высокими показателями по определенным критериям, которые наиболее полно характеризуют качество данной хромосомы. Таким образом, в общем случае цель генетического алгоритма состоит в поиске такой хромосомы (альтернативного решения), значение ЦФ, которой максимально приближено к оптимальному. Направленный поиск оптимального решения в ГА реализуется за счет выполнения операторов

генетического поиска [108].

В данной диссертационной работе для поиска альтернативных решений более высокого качества предлагается использовать модифицированный генетический алгоритм, схема работы которого приведена на рисунке 3.23.

Опишем работу модифицированного ГА более подробно. Изначально вводятся управляющие параметры алгоритма: T – число итераций, P_k – вероятность выполнения модифицированного ОК ($0 < P_k < 1$), P_m – вероятность выполнения модифицированного ОМ ($0 < P_m < 1$).

Затем на популяции, полученной после выполнения предыдущих этапов поиска, инициализируется выполнение модифицированного ОК. Поисковый процесс для всей популяции включает следующие шаги:

1. Счетчик выполненных ОК обнуляется.
2. Определяется пара хромосом для последующего скрещивания.
3. Выполняется скрещивание хромосом.
4. Проверяется условие останова для ОК.

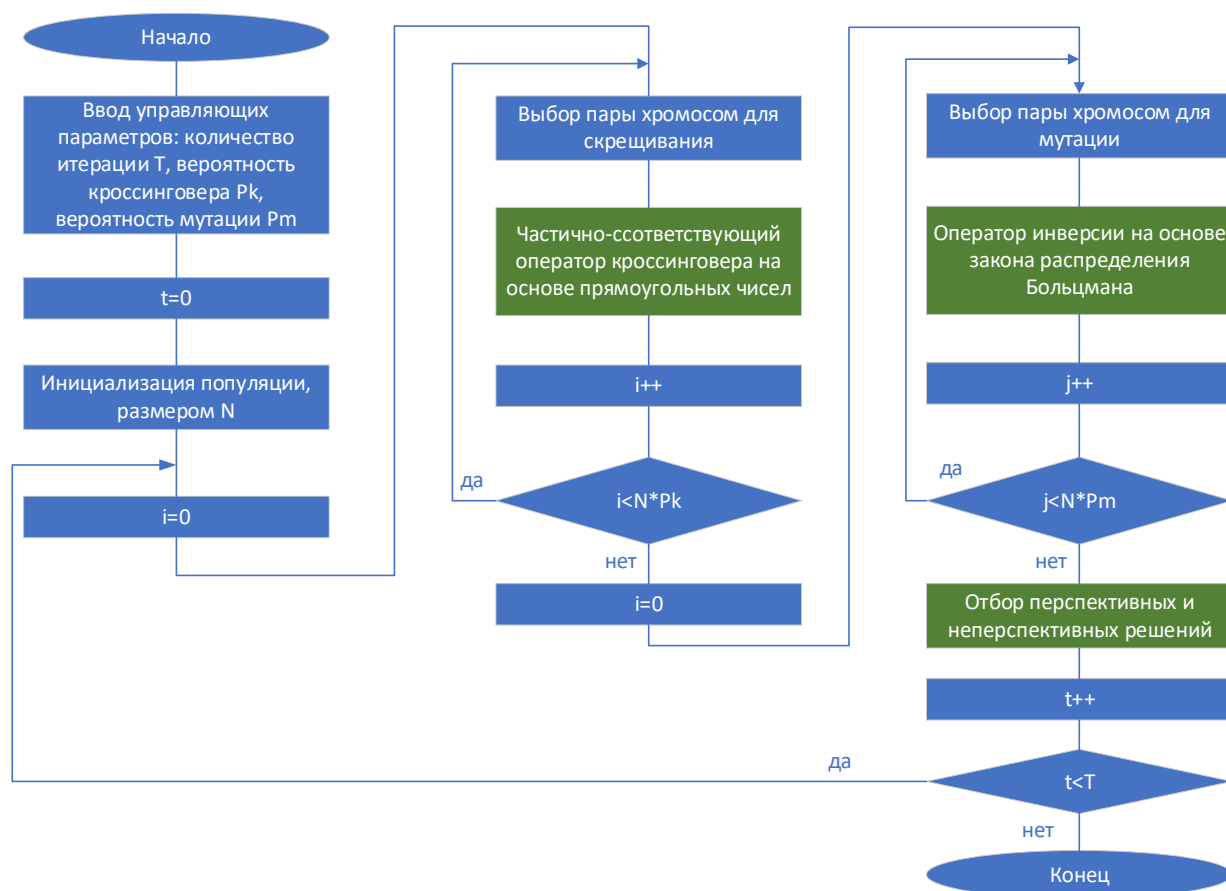


Рисунок 3.23 – Структурная схема модифицированного генетического алгоритма

Одной из ключевых операций при реализации модифицированного ОК является выбор пары хромосом для последующего скрещивания, т. к. это оказывает значительное влияние на качество получаемых решений. В диссертационной работе пара хромосом выбирается на основе стратегии колеса рулетки, т. к. этот метод позволяет с большей долей вероятности выбирать решения более высокого качества. Также снижается вероятность сходимости алгоритма в локальном оптимуме путем внесения разнообразия в генотип текущей популяции и увеличивается вероятность нахождения глобального оптимума за счет работы с большим количеством решений, приближенных к нему. Для управления скоростью сходимости алгоритма вводится настроечный параметр «Вероятность ОК», который позволяет контролировать количество получаемого генетического материала. Процедура получения новых альтернативных решений (модифицированный ОК) была детально описана выше и заключается в обмене генетическим материалом хромосом-родителей для генерации хромосом-потомков. После выполнения скрещивания полученные хромосомы добавляются в текущую популяцию и затем иницируется процесс выполнения модифицированного оператора инверсии.

Суть выполнения модифицированного ОИ заключается в следующем: из популяции альтернативных решений на основе пропорционального отбора выбирается хромосома для инвертирования. Получение нового решения заключается в инвертировании участка выбранной хромосомы внутри точек разрыва, принцип расположения которых инспирирован законом распределения Больцмана.

Затем для приведения размера популяции к исходному производится отбор наиболее перспективных альтернативных решений в новую популяцию. В диссертационной работе реализован отбор на основе колеса рулетки, т. к. такой вероятностный подход способствует формированию результирующей популяции пропорционально ЦФ хромосом, входящих в нее. Процесс поиска продолжается

итерационно, пока счетчик итераций не достигнет предварительно заданного значения.

Отметим, что разработанный модифицированный генетический алгоритм позволяет повысить качество решений, полученных на предыдущих этапах поиска, за счет выполнения модифицированных операторов кроссинговера, инверсии и отбора. Также данный алгоритм имеет гибкую и динамичную структуру, т. е. конструктор имеет возможность изменить вероятность выполнения того или иного модифицированного генетического оператора.

3.5 Разработка нового многоуровневого алгоритма разбиения схем при проектировании СБИС

Для достижения баланса между скоростью сходимости алгоритма и качества получаемых решений, а также уменьшения вероятности попадания в локальный оптимум был предложен многоуровневый алгоритм разбиения схем, основанный на модифицированных бактериальном, светлячковом и генетическом алгоритмах. Общая схема работы многоуровневого алгоритма разбиения представлена на рисунке 3.24.

Вводится параметр шкалы алгоритма *prob*, принимающий одно значение из множества $\{-1, 0, 1\}$, где -1 – выполнение только модифицированного светлячкового алгоритма, 0 – выполнение только модифицированного бактериального алгоритма, 1 параллельное выполнение модифицированных светлячкового и бактериального алгоритмов.

В общем случае для начального разбиения, в зависимости от значения параметра *prob*, выполняется бактериальный или светлячковый алгоритм или оба параллельно. В случае параллельного выполнения результатом являются две популяции, для которых выполняется объединение решений с помощью оператора миграции. Проверяется выполнение критерия останова алгоритма, и, если он не выполнен, объединенная популяция подается на вход модифицированного генетического алгоритма для получения более точного квазиоптимального

решения.

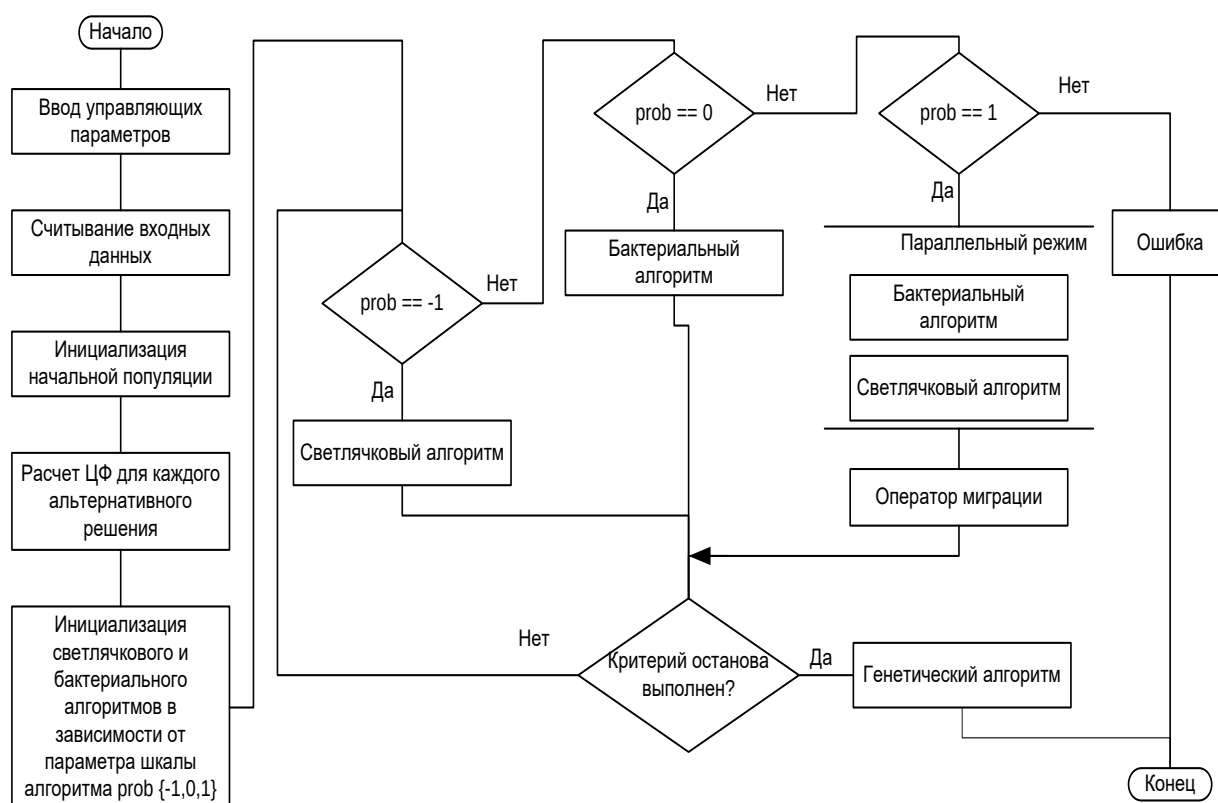


Рисунок 3.24 – Схема работы многоуровневого алгоритма разбиения СБИС

С целью уменьшения времени работы алгоритма в диссертационной работе предлагается использовать единый механизм представления данных. Единый подход к кодированию и декодированию решений позволит исключить потери процессорного времени при передаче данных между различными уровнями многоуровневого алгоритма разбиения схем при проектировании СБИС.

На текущем этапе развития науки и техники для решения сложных вычислительных задач и обработки большого количества данных часто используют многоядерные процессорные системы [109-113]. Это позволяет улучшить основные параметры системы, а также сократить время решения поставленной задачи. Таким образом, для увеличения скорости получения наборов квазиоптимальных решений целесообразна разработка параллельного алгоритма разбиения схем при проектировании СБИС.

Одной из особенностей предлагаемого в диссертационной работе многоуровневого алгоритма разбиения схем является «Параллельный режим».

Данный режим активируется при значении параметра шкалы алгоритма $prob = 1$. Особенностью предложенного режима работы является независимое параллельное выполнение модифицированных бактериального и светлячкового алгоритмов с последующим объединением полученных результатов путем реализации оператора миграции.

Эффективность работы оператора миграции обусловлена использованием единой оценки качества решений (ЦФ) и унифицированных механизмом кодирования и декодирования решений бактериального и светлячкового алгоритмов. В диссертационной работе выбор решений, которые попадут в результирующую популяцию, определяется модифицированным оператором отбора, в основе которого лежит принцип вероятностного отбора. Модификация производится за счет фиксирования изначальных секторов рулетки в соответствии с коэффициентом φ , $\varphi \in [0,1]$, который определяет процент вхождения в конечную популяцию результатов бактериального и светлячкового алгоритмов. Коэффициент вхождения φ задается лицом, принимающим решения (ЛПР).

Секторы колеса рулетки соответствуют сумме значений ЦФ всех хромосом. Каждой хромосоме $ch_i, i = 1, 2, \dots, N$ (N – количество хромосом в популяции) ставится в соответствие сектор колеса рулетки $v(ch_i)$ согласно формуле

$$v(ch_i) = p_s(ch_i) * 100\% \quad (3.8)$$

где $p_s(ch_i)$ – вероятность выбора хромосомы ch_i для попадания в результирующую популяцию, которая определяется по следующей формуле:

$$p_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)} \quad (3.9)$$

где $F(ch_i)$ – ЦФ хромосомы ch_i .

Рассмотрим механизм работы модифицированного оператора отбора для популяции, содержащей 5 хромосом. В таблице 3.1 представлены условные значения ЦФ после выполнения модифицированных бактериального и светлячкового алгоритмов, а размеры секторов, которые соответствуют значениям ЦФ.

Таблица 3.1. Пример работы модифицированного оператора селекции

Бактериальный алгоритм						Светлячковый алгоритм					
Номер хромосомы	ЦФ хромосомы	Доля ЦФ в общей популяции, %	Размер сектора, $\varphi=0,15, \%$	Размер сектора, $\varphi=0,5, \%$	Размер сектора, $\varphi=0,75, \%$	Номер хромосомы	ЦФ хромосомы	Доля ЦФ в общей популяции, %	Размер сектора, $\varphi=0,15, \%$	Размер сектора, $\varphi=0,5, \%$	Размер сектора, $\varphi=0,75, \%$
1	0,96	32	4,8	16	24	1	0,15	6	5,1	3	1,5
2	0,72	24	3,6	12	18	2	0,34	13	11,05	6,5	3,25
3	0,6	20	3	10	15	3	0,38	15	12,75	7,5	3,75
4	0,48	16	2,4	8	12	4	0,76	30	25,5	15	7,5
5	0,24	8	1,2	4	6	5	0,94	36	30,6	18	9

На рисунке 3.25 приведено колесо рулетки, построенное в соответствии с таблицей 3.1, при максимизации значения ЦФ.

С помощью оператора отбора из множества хромосом (как родителей, так и потомков) формируется новая популяция, включающая в себя наиболее приспособленных индивидов.

Одной из важнейших функций отбора является избегание вырождения популяции. Это происходит в том случае, когда потомки хромосомы с высоким значением ЦФ занимают практически всю популяцию, сокращая разнообразие генетического материала, что способствует преждевременной сходимости алгоритма.

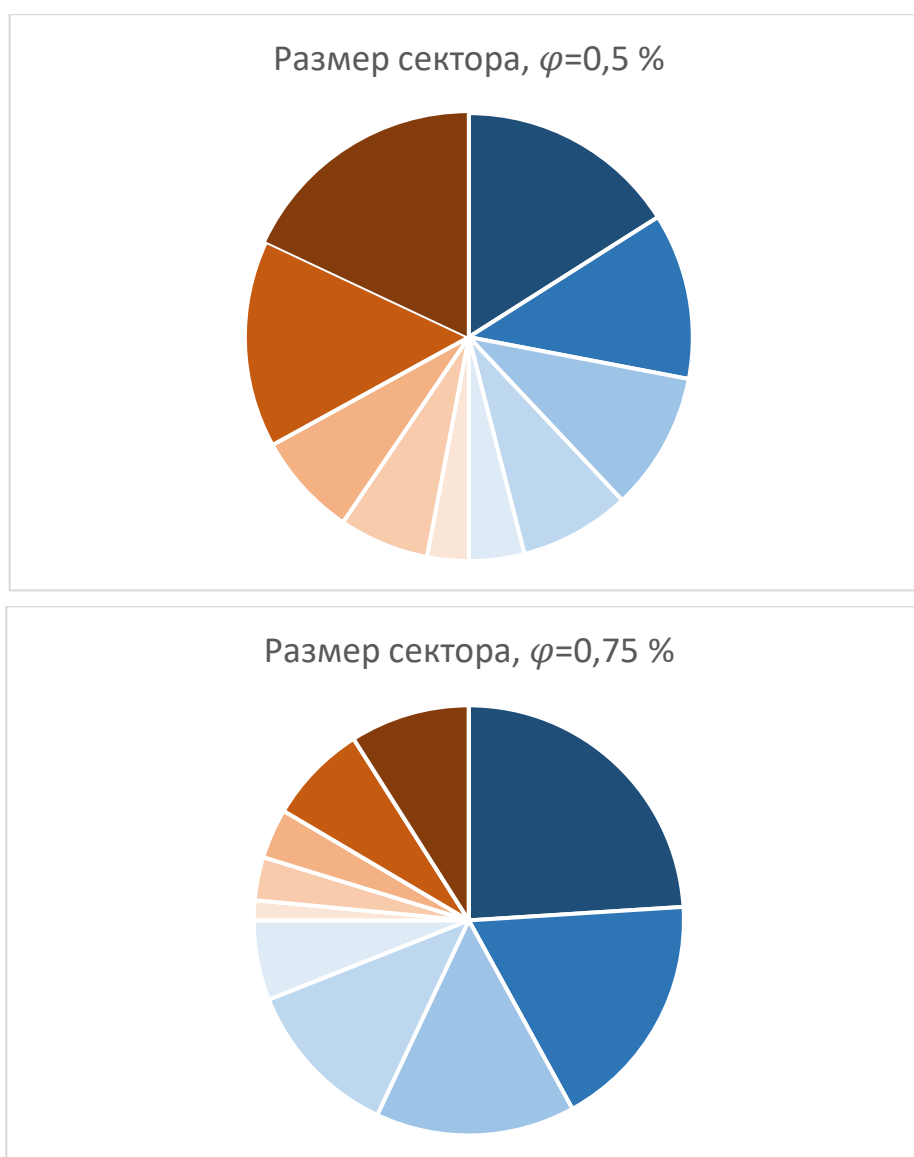


Рисунок 3.25 – Колесо рулетки

Применение в данной работе параллельного режима работы существенно сокращает время выполнения многоуровневого алгоритма разбиения схемы при сохранении качества получаемых наборов квазиоптимальных решений. Используемый оператор миграции позволяет сохранить высокий уровень диверсификации поиска, что способствует уменьшению вероятности попадания алгоритма в локальный оптимум. В то же время растет пространственная сложность алгоритма поиска, что приводит к ужесточению требований к используемым вычислительным ресурсам.

3.6 Краткие выводы

1. Разработан механизм кодирования и декодирования решений для организации единого подхода к представлению данных на всех этапах работы многоуровневого алгоритма разбиения схем при проектировании СБИС. Унифицированный механизм представления проектных решений позволил исключить потери процессорного времени при передаче данных между различными уровнями многоуровневого алгоритма разбиения схем.

2. Разработан модифицированный алгоритм бактериальной оптимизации для реализации проектной процедуры разбиения схем при проектировании СБИС. Модификация разработанного алгоритма заключается в динамическом изменении количества шагов хемотаксиса, т. е. количества парных перестановок, в зависимости от номера итерации, что позволяет уменьшить временные затраты при проектировании.

3. Разработан модифицированный светлячковый алгоритм, который способен выделять в поисковом пространстве области с наиболее перспективными решениями, что способствует значительному сокращению времени работы алгоритма. Модификация стратегии выбора пары хромосом позволяет частично решить проблему предварительной сходимости алгоритма.

4. Для улучшения качества решений, полученных модифицированными бактериальным и светлячковым алгоритмами, разработан модифицированный генетический алгоритм, выполняющийся на заключительном этапе работы многоуровневого алгоритма. Модифицированный генетический алгоритм реализует модель эволюции Ламарка, а модифицированные генетические операторы способствуют получению решений более высокого качества и сокращению области поиска.

5. Разработан многоуровневого алгоритма разбиения схем при проектировании СБИС. В основе алгоритма лежит параметр шкалы выбора, в зависимости от значения которого предусматривается реализация вариантов поисковых процедур на основе последовательных, последовательно-параллельных и параллельных

стратегий. Выполнение разработанных модифицированных алгоритмов в параллельном режиме позволяет существенно сократить время выполнения многоуровневого алгоритма разбиения схем при проектировании СБИС при сохранении качества получаемых наборов квазиоптимальных решений.

4 ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ И АНАЛИЗ ЭФФЕКТИВНОСТИ РАЗРАБОТАННЫХ МЕТОДОВ И СРЕДСТВ РАЗБИЕНИЯ СХЕМ ПРИ ПРОЕКТИРОВАНИИ СБИС

4.1 Теоретические оценки временных и пространственных характеристик методов и средств разбиения схем при проектировании СБИС

Временную и пространственную сложности разработанного многоуровневого алгоритма разбиения схем возможно оценить путем сложения данных показателей поисковых механизмов, входящих в его состав. В данной диссертационной работе были разработаны модифицированные бактериальный, светлячковый и генетический алгоритмы разбиения схем, оценка сложности которых зависти от базовых характеристик схемы, таких как N – количество элементов в схеме и M – количество цепей [114-116].

Пространственная сложность многоуровневого алгоритма определяется следующим образом. Каждый элемент СБИС характеризуется длиной L , шириной H и идентификатором I . Объем памяти, отводимый для их хранения, определяется следующим образом

$$X = (L+H+I+C) * B * N. \quad (4.1)$$

Здесь C – количество цепей с элементом с идентификатором I , B – количество ячеек, необходимых для хранения целого числа. Для хранения списка цепей СБИС необходимо

$$Y = B * N * M \quad (4.2)$$

ячеек памяти.

Одно альтернативное решение занимает $B*N$ ячеек памяти, а популяция альтернативных решений - $B*N*|P_i|$ ячеек, где $|P_i|$ - количество агентов в популяции

каждого алгоритма. Кроме этого, для временного хранения текущей популяции альтернативных решений на очередной итерации необходимо:

- $B * N * |A_f|$ ячеек для модифицированного светлячкового алгоритма, где $|A_f|$ - общее число агентов данного алгоритма;
- $B * N * |A_{bak}|$ ячеек для модифицированного бактериального алгоритма, где $|A_{bak}|$ - общее число агентов данного алгоритма;
- $-B * N * |P_{GA}| * (P_c + P_m)$ ячеек для модифицированного генетического алгоритма, где $|P_e|$ - общее число агентов данного алгоритма, P_c – вероятность выполнения модифицированного ОК, P_m - вероятность выполнения модифицированного ОИ.

В диссертационной работе предполагается три режима выполнения многоуровневого алгоритма разбиения схемы: последовательное выполнение модифицированных светлячкового и генетического алгоритмов, последовательное выполнение модифицированных бактериального и генетического алгоритмов, параллельное выполнение модифицированных светлячкового и бактериального алгоритмов с последующим выполнением модифицированного генетического алгоритмов.

Количество ячеек памяти, необходимых для последовательного выполнения модифицированных бактериального и генетического алгоритмов, вычисляется по формуле:

$$B * N * |A_{bak}| + B * N * |P_{GA}| * (P_c + P_m) + (L + H + I + C) * B * N + B * N * M + B * N * |P_i|. \quad (4.3)$$

После упрощения формула примет вид:

$$B * N * (|A_{bak}| + |P_{GA}| * (P_c + P_m) + (L + H + I + C) + M + |P_i|). \quad (4.4)$$

Количество ячеек памяти, необходимых для последовательного выполнения модифицированных светлячкового и генетического алгоритмов, вычисляется по формуле:

$$B * N * |A_f| + B * N * |P_{GA}| * (P_c + P_m) + (L + H + I + C) * B * N + B * N * M + B * N * |P_i|. \quad (4.5)$$

После упрощения формула примет вид:

$$B * N * (|A_f| + |P_{GA}| * (P_c + P_m) + (L + H + I + C) + M + |P_i|). \quad (4.6)$$

Количество ячеек памяти, необходимых для параллельного выполнения модифицированных светлячкового и бактериального алгоритмов с последующим выполнением модифицированного генетического алгоритма, вычисляется по формуле:

$$|T| * (B * N * |A_{bak}| + B * N * |A_f|) + B * N * |P_{GA}| * (P_c + P_m) + (L + H + I + C) * B * N + B * N * M + B * N * |P_i|. \quad (4.7)$$

где $|T|$ – общее число потоков.

После упрощения формула примет вид:

$$|T| * B * N * (|A_{bak}| + |A_f|) + |P_{GA}| * (P_c + P_m) + (L + H + I + C) + M + |P_i|. \quad (4.8)$$

Под временной сложностью алгоритма (BCA) понимается зависимость времени работы алгоритма от размерности (объема входных данных) поставленной задачи. Временная сложность разработанного многоуровневого алгоритма состоит из сложностей модифицированных биоинспирированных алгоритмов разбиения СБИС, сложности функций кодирования и декодирования данных и сложности оценочной функции [117].

Временная сложность модифицированного светлячкового алгоритма разбиения СБИС можно оценить по формуле $O(|P_f| * A_f * I_f)$, где $|P_f|$ – количество агентов в популяции, I_f – заданное число итераций.

Временная сложность модифицированного бактериального алгоритма разбиения СБИС можно оценить по формуле $O(|P_{bak}| * I_{bak} * T)$, где $|P_{bak}|$ – количество агентов в популяции, I_{bak} – заданное число итераций, T – количество шагов хемотаксиса $T \in [1, T_{max}]$, T_{max} – целое число.

Временная сложность модифицированного генетического алгоритма зависит от входных данных и вычисляется следующим образом:

$$O_{GA} = O(|P_{GA}| * (I_{GA} + P_c * O_c + P_m * O_m)) \quad (4.9)$$

где $|P_{GA}|$ – количество хромосом в популяции, P_c – вероятность выполнения ОК; P_m – вероятность выполнения ОМ; $P_m(0,1)$, I_{GA} – заданное число итераций, O_c –

временная сложность ОК, O_m – временная сложность ОМ; $O_c = O(N)$, $O_m = const$, где $const$ – постоянная величина.

Временная сложность процесса генерации новой популяции пропорциональна $O(N * |P|)$ для всех разработанных алгоритмов, а временная сложность вычисления ЦФ пропорциональна $O(N)$. ВСА кодирования и декодирования также пропорциональна $O(N * |P|)$.

Многоуровневый алгоритм разбиения СБИС поддерживает три режима работы, следовательно, временная сложность для каждого режима будет различна.

В случае последовательного выполнения модифицированных бактериального и генетического алгоритмов временная сложность вычисляется следующим образом:

$$\begin{aligned} O_{bak} = & O(N * |P|) + O(|P_{bak}| * I_{bak} * T) \\ & + O(|P_{GA}| * (I_{GA} + P_c * O_c + P_m * O_m) + O(N) + O(N * |P|) \\ & + O(N * |P|) + O(N * |P|)) \end{aligned} \quad (4.10)$$

В случае последовательного выполнения модифицированных светлячкового и генетического алгоритмов временная сложность вычисляется следующим образом:

$$\begin{aligned} O_f = & O(N * |P|) + O(|P_f| * A_f * I_f) \\ & + O(|P_{GA}| * (I_{GA} + P_c * O_c + P_m * O_m) + O(N) + O(N * |P|) \\ & + O(N * |P|) + O(N * |P|)) \end{aligned} \quad (4.11)$$

При параллельном выполнении светлячкового и бактериального алгоритмов с последующим выполнением модифицированного генетического алгоритмов временная сложность для одного потока пропорциональна следующему выражению:

$$\begin{aligned} O_{parallel} = & O(N * |P|) + \max(O(|P_f| * A_f * I_f), O(|P_{bak}| * I_{bak} * T)) \\ & + O(|P_{GA}| * (I_{GA} + P_c * O_c + P_m * O_m) + O(N) + O(N * |P|) \\ & + O(N * |P|) + O(N * |P|)) \end{aligned} \quad (4.12)$$

4.2 Цель экспериментальных исследований

В диссертационной работе был проведен ряд экспериментальных исследования для определения эффективности многоуровневого алгоритма, реализующего механизм разбиения схем при проектировании СБИС на заранее

известное множество фрагментов, по сравнению с известными алгоритмами решения данной задачи. В ходе эксперимента уточняются значения управляющих параметров для каждого алгоритма, входящего в состав многоуровневого, а также исследуются полученные квазиоптимальные решения. Эффективность разработанных методов и алгоритмов может быть доказана несколькими путями [118]:

- теоретически на основе временной и пространственной сложности разработанных алгоритмов;
- эмпирически с помощью сравнения результатов экспериментов.

В результате проведения экспериментальных исследований в данной диссертационной работе был получен и проанализирован набор данных.

Объектом исследований являются разработанный алгоритм разбиения схем при проектировании СБИС, содержащий блоки светлячкового и бактериального алгоритмов, включающий ряд биоинспирированных эвристик поиска альтернативных решений, а также модифицированный генетический алгоритм, состоящий из модифицированных генетических операторов (ОК, ОМ и т.д.) Исследование квазиоптимальности реализуемых в работе стратегий заключается в определении значений целевой функции и оценки ее сложности. Качество решения поставленной задачи во многом зависит от таких настроечных параметров модифицированных биоинспирированных алгоритмов, как размер начальной популяции, вероятность выполнения модифицированных ГО, количество итераций и т.д. Эффективность разработанного многоуровневого алгоритма определяется его временной сложностью, способностью алгоритма находить решения с высоким показателем ЦФ, сопоставление полученных решений с результатами hMetis и MLPart.

В ходе исследования многоуровневого алгоритма разбиения СБИС анализируются показатели приращения временных и пространственных затрат, необходимых для выполнения модифицированных биоинспирированных алгоритмов, по сравнению с ручным проектированием.

В рамках данной диссертационной работы была разработана программная подсистема, состоящая из трех модулей, для реализации многоуровневого алгоритма разбиения СБИС. При проведении испытаний в качестве начальных данных использовались СБИС и их графовые модели, полученные с помощью генератора случайных схем. Для генерации случайных СБИС на вход генератора поступал ряд настроечных параметров. При этом выходными параметрами разработанной подсистемы являются время работы и наилучшее полученное решение.

Сформулируем основные цели проведения экспериментальных исследований:

- вычисление зависимости времени работы многоуровневого алгоритма разбиения СБИС от количества элементов и цепей СБИС;
- вычисление зависимости объема памяти, необходимой для хранения информации, от количества элементов и цепей СБИС.

4.3 Планирование вычислительного эксперимента

Планирование эксперимента – подготовка набора данных, которые смогут отразить тенденции изменения исследуемых параметров [115]. В данной диссертационной работе проведение экспериментальных исследования планируется следующим образом:

1. Для определения ВСА и ПСА будет проведен ряд экспериментов при постоянных значениях настроечных параметров (Таблица 4.1).;

Таблица 4.1. План вычислительного эксперимента.

№ п/п серии	Кол-во элементов схемы, шт.	Кол-во цепей, шт.	Размер серии
1	12506	14111	5000
2	19342	19584	5000
3	22853	27401	5000

4	27220	31970	5000
5	28146	28446	5000
6	32332	34862	5000
7	45639	48117	5000
8	51023	50513	5000
9	53110	60902	5000
10	68685	75196	5000
11	70152	71076	5000
12	70439	77241	5000
13	83709	99666	5000
14	147088	152772	5000

2. Для получения наиболее эффективных решений будет проведена серия экспериментов, направленная на поиск решений с высоким значением ЦФ с помощью разработанного многоуровневого алгоритма разбиения СБИС [119, 120].

Сравнение качества решения, полученного с использованием подсистемы биоинспирированного поиска многоуровневого алгоритма решения задачи разбиения СБИС, проводится на основе схем, полученных с помощью генератора случайных схем.

Оценкой качества полученных решений проводится на основе количества межблочных соединений.

Сопоставление полученных решений будет проводиться с результатами, полученными с помощью известных алгоритмов разбиения СБИС MLPart [121], hMetis [122].

Результатом проведения экспериментальных исследований будут наборы данных для последующего анализа и сопоставления их с результатами, которые были получены сторонними алгоритмами.

4.4 Описание программного комплекса

Приведем краткое описание программного комплекса моделирования решения задачи разбиения схем при проектировании СБИС с минимизацией

интегрированного критерия – суммарного количества межблочных соединений. Программный комплекс состоит из трех модулей, представленных на рисунке 4.1:

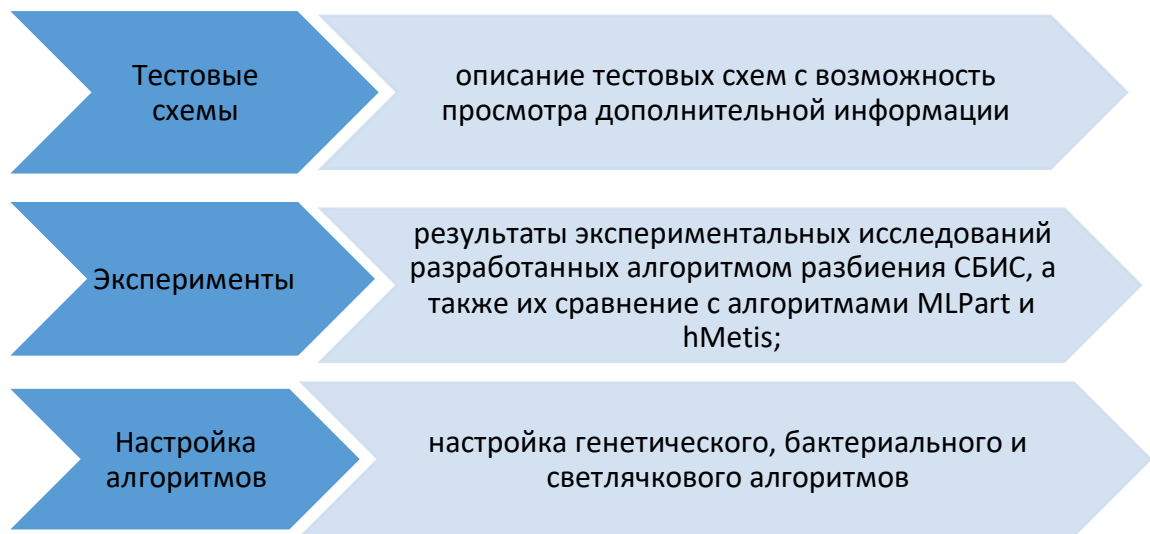


Рисунок 4.1 – Основные модули, входящие в состав программного комплекса

Исходными данными являются наборы тестовых схем *GEN_SCH_1-GEN_SCH_14*, сгенерированных генератором случайных схем *MultiGenerator*, представленные на рисунке 4.2.

Разработанная система позволяет моделировать решение задачи разбиения СБИС с использованием различных алгоритмов.

Тестовые схемы

Эксперименты

Настройки алгоритмов

Добавить

Id	Название	Генератор	Количество элементов	Количество целей	Действия
1	GEN_SCH_1	MultiGenerator	12757	14111	⚙️ -
2	GEN_SCH_2	MultiGenerator	19501	19583	⚙️ -
3	GEN_SCH_3	MultiGenerator	23134	27409	⚙️ -
4	GEN_SCH_4	MultiGenerator	27502	31971	⚙️ -
5	GEN_SCH_5	MultiGenerator	29348	28444	⚙️ -
6	GEN_SCH_6	MultiGenerator	32494	34825	⚙️ -
7	GEN_SCH_7	MultiGenerator	45927	48119	⚙️ -
8	GEN_SCH_8	MultiGenerator	53308	50514	⚙️ -
9	GEN_SCH_9	MultiGenerator	53391	60906	⚙️ -
10	GEN_SCH_10	MultiGenerator	69427	75197	⚙️ -
11	GEN_SCH_11	MultiGenerator	70554	81452	⚙️ -
12	GEN_SCH_12	MultiGenerator	71075	77241	⚙️ -
13	GEN_SCH_13	MultiGenerator	84198	99667	⚙️ -
14	GEN_SCH_14	MultiGenerator	147604	152773	⚙️ -

Рисунок 4.2 – Описание набора тестовых схем

В модуле «Эксперименты» реализовано выполнение многоуровневого алгоритма разбиения СБИС в трех режимах работы (светлячковый + генетический, бактериальный + генетический, параллельный + генетический), а также известные алгоритмы разбиения MLPart, hMetis, что представлено на рисунке 4.3.

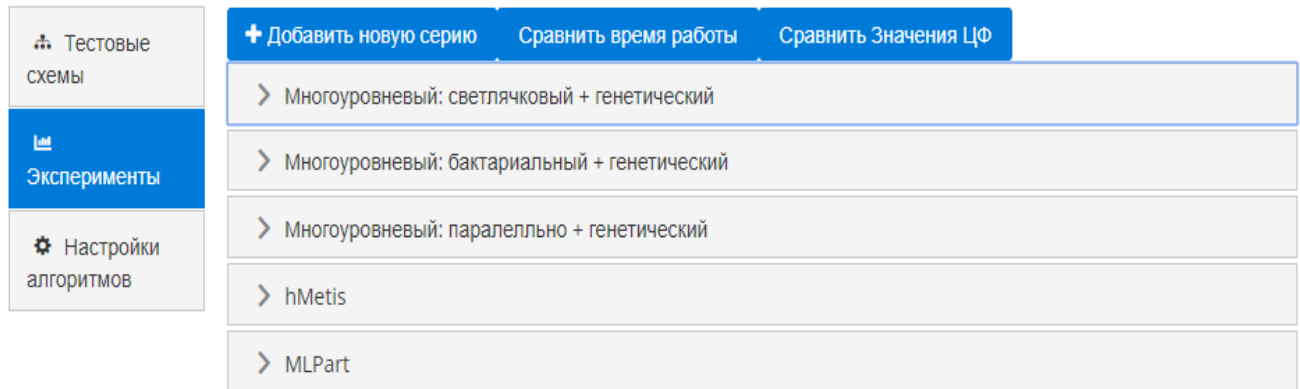


Рисунок 4.3 – Модуль «Эксперименты»

Здесь для каждого алгоритма приводятся сводные таблицы результатов проведенных экспериментов: среднее время выполнения серии экспериментов, время на выполнение дополнительных процедур, объем памяти, требуемый на выполнение алгоритма и дополнительных процедур, и значение ЦФ, отражающее количество межблочных соединений.

Результаты экспериментальных исследований для многоуровневого алгоритма разбиения при значении параметра $\text{prob} = -1$ приведены на рисунке 4.4.

Тестовые схемы

Эксперименты

Настройки алгоритмов

Добавить новую серию

Сравнить время работы

Сравнить значения ЦФ

Многоуровневый: светлячковый + генетический

Серия экспериментов для реализованного многоуровневого алгоритма компоновки при prob = -1

Схема	Среднее время, с	Дополнительное время, %	Память, КБ	Дополнительная память, %	Значение ЦФ, шт.
GEN_SCH_1: ат: 12757, цепи: 14111	120	2.24	12.34	14	247
GEN_SCH_2: ат: 19501, цепи: 19583	153	3.51	18.6	13	322
GEN_SCH_3: ат: 23134, цепи: 27409	184	4.86	32	11	890
GEN_SCH_4: ат: 27502, цепи: 31971	228	5.57	46	12	542
GEN_SCH_5: ат: 29348, цепи: 28444	231	5.55	96	14	1782
GEN_SCH_6: ат: 32494, цепи: 34825	287	5.81	147	13	703
GEN_SCH_7: ат: 45927, цепи: 48119	312	6.01	423	13	851
GEN_SCH_8: ат: 53308, цепи: 50514	378	6.51	593	12	1239
GEN_SCH_9: ат: 53391, цепи: 60906	374	6.87	788	11	560
GEN_SCH_10: ат: 69427, цепи: 75197	487	7.12	816	12	1359
GEN_SCH_11: ат: 70554, цепи: 81452	537	7.45	945	14	910
GEN_SCH_12: ат: 71075, цепи: 77241	542	7.65	1147	15	2409
GEN_SCH_13: ат: 84198, цепи: 99667	632	7.99	1562	14	1206
GEN_SCH_14: ат: 147604, цепи: 152773	812	8.13	3324	15	2040

Рисунок 4.4 –Результаты экспериментальных исследований для многоуровневого алгоритма разбиения при $prob = -1$

Кроме того, программный комплекс позволяет добавить новую серию экспериментов, сравнить время работы и значения ЦФ для всех разработанных алгоритмов. Визуализация полученных результатов приведена на рисунках 4.5 и 4.6. В результате проведения эксперимента рассчитываются показатели среднего времени работы выбранного алгоритма, дополнительное время, затраченное на передачу данных между алгоритмами, объем памяти, дополнительный объем памяти, а также значение ЦФ. На основе полученных результатов строятся гистограммы сравнения времени работы алгоритмов и значения ЦФ при различных значениях параметра $prob$.

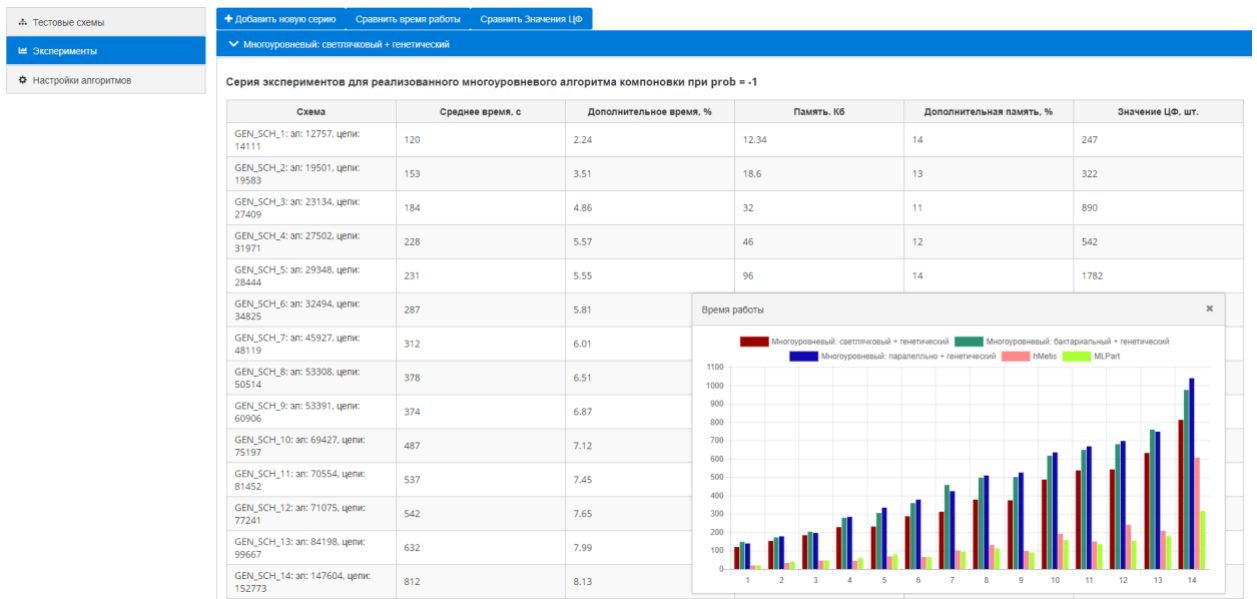


Рисунок 4.5 – Визуализация сравнения времени работы всех разработанных алгоритмов

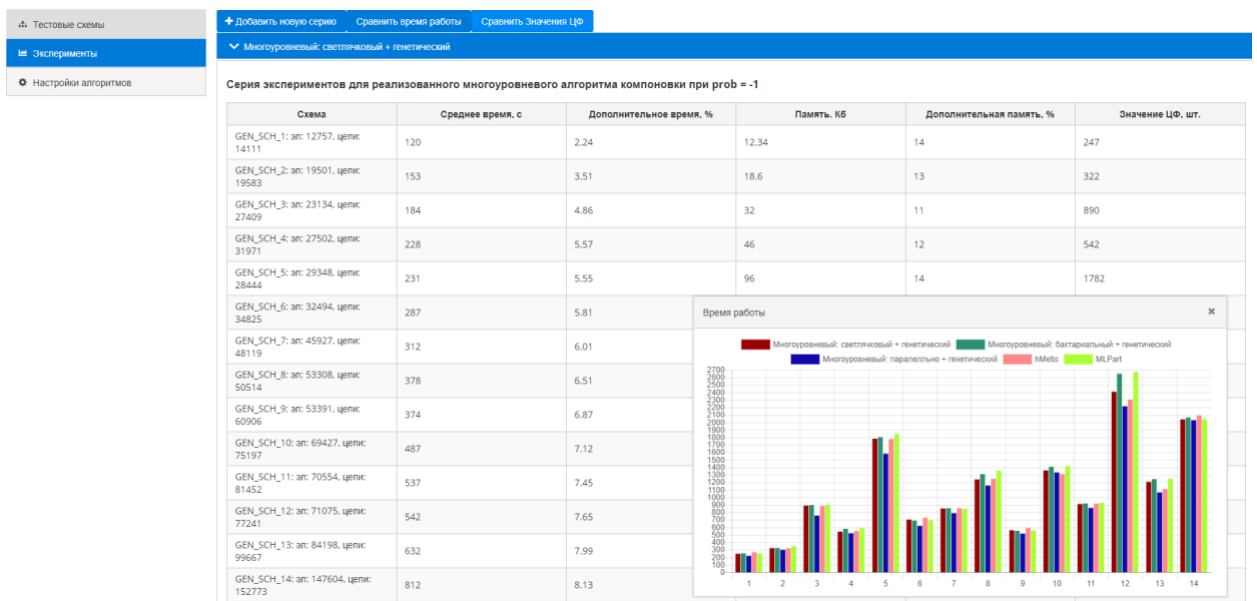
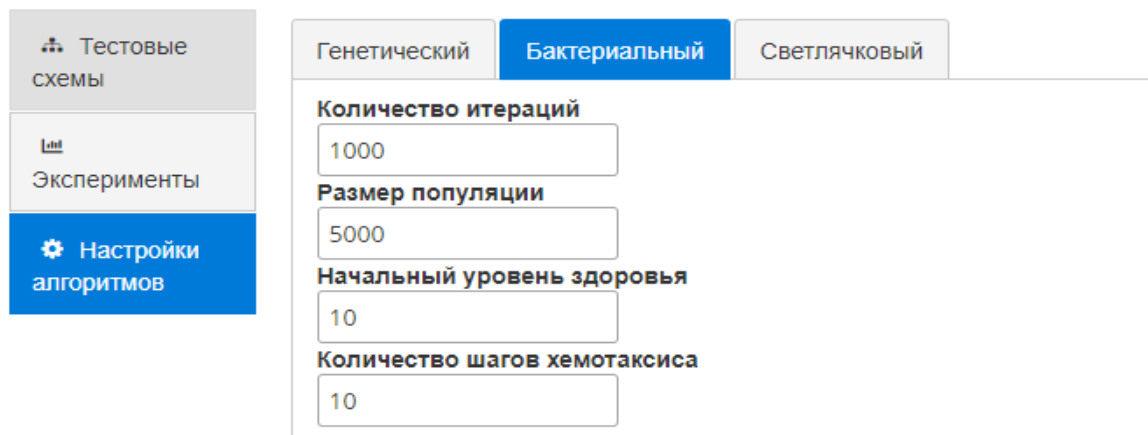


Рисунок 4.6 – Визуализация сравнения значения ЦФ всех разработанных алгоритмов

Модуль «Настройка алгоритмов» позволяет ввести настроечные параметры для бактериального, светлячкового и генетического алгоритмов. Для бактериального алгоритма необходимо ввести следующие параметры: количество итераций, размер популяции, начальный уровень здоровья и количество шагов

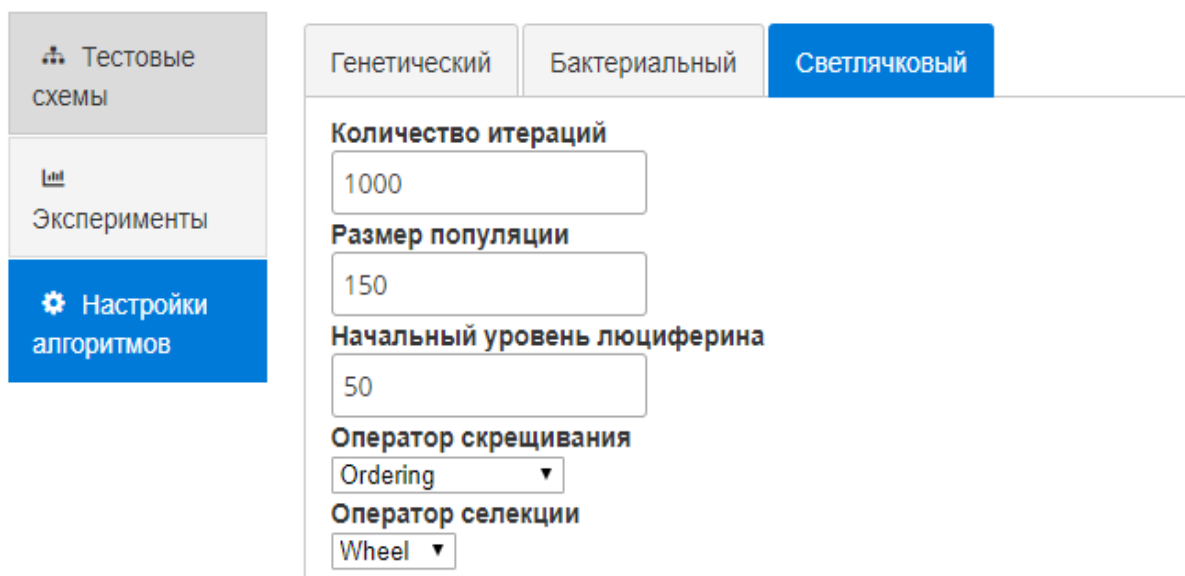
хемотаксиса. На рисунке 4.7 представлено Окно ввода настроечных параметров для бактериального алгоритма.



Генетический	Бактериальный	Светлячковый
Количество итераций		
1000		
Размер популяции		
5000		
Начальный уровень здоровья		
10		
Количество шагов хемотаксиса		
10		

Рисунок 4.7 – Окно ввода настроечных параметров для бактериального алгоритма

Для светлячкового алгоритма необходимо ввести следующие параметры: количество итераций, размер популяции, начальный уровень люциферина. Кроме того, возможен выбор оператора скрещивания (одноточечный, частично-соответствующий или упорядоченный) и оператора селекции (на основе «Колеса рулетки» или жадный). На рисунке 4.8 представлено окно ввода настроечных параметров для светлячкового алгоритма



Генетический	Бактериальный	Светлячковый
Количество итераций		
1000		
Размер популяции		
150		
Начальный уровень люциферина		
50		
Оператор скрещивания		
Ordering		
Оператор селекции		
Wheel		

Рисунок 4.8 – Окно ввода настроечных параметров для светлячкового алгоритма

Для модифицированного генетического алгоритма необходимо ввести следующие параметры: количество итераций, размер популяции, вероятность выполнения оператор мутации, вероятность выполнения оператора кроссинговера. Кроме того, возможен выбор оператора кроссинговера (одноточечный, частично-соответствующий или модифицированный ОК) и оператора селекции (на основе «Колеса рулетки»).

На рисунке 4.9 приведено окно ввода настроечных параметров для светлячкового алгоритма.

Параметр	Значение
Количество итераций	1000
Размер популяции	200
Вероятность выполнения мутации	0,2
Оператор кроссинговера	Partially matched
Вероятность выполнения кроссинговера	0,8
Оператор селекции	Wheel

Рисунок 4.9 – Окно ввода настроечных параметров для светлячкового алгоритма

Разработанная программная подсистема обладает свойством гибкости и достаточно хорошо настраивается. Для оценки ее быстродействия и эффективности разработанного многоуровневого алгоритма была реализована функция графического отображения зависимости ЦФ от времени работы алгоритма с учетом изменяющихся параметров биоинспирированного поиска.

4.5 Вычислительный эксперимент

С целью наглядной демонстрации эффективности, а также для определения оптимальных значений характеристик разработанных алгоритмов было проведено несколько серий экспериментов. Для этого были программно реализованы бактериальный, светлячковый и генетический алгоритмы разбиения СБИС, а также известные алгоритмы разбиения hMetis и MLPart.

Алгоритм hMetis относится к классу многоуровневых алгоритмов разбиения гиперграфа. В отличие от традиционных методов разбиения, где происходит разбиение непосредственно исходного графа, выполнение алгоритма hMetis происходит в три этапа.

На первом этапе производится сокращение размерности исходного гиперграфа путем объединения вершин или ребер таким образом, чтобы качество разбиения полученного гиперграфа не было значительно ниже, чем качество разбиения исходного гиперграфа.

Затем проводится разбиение данного гиперграфа Fiduccia-Mattheyses (FM) алгоритмом. На заключительном этапе производится разворачивание полученного гиперграфа до исходной размерности и затем применяется алгоритм улучшения разбиения для уменьшения количества межблочных соединений (FM алгоритм).

Алгоритм MLPart также принадлежит к классу многоуровневых алгоритмов разбиения. Его ключевой особенностью является возможность управления скоростью объединения вершин и ребер при сокращении размерности гиперграфа и, следовательно, количеством иерархических уровней [121, 122].

Экспериментальные исследования многоуровневого алгоритма разбиения СБИС проводились при постоянных параметрах алгоритмов биоинспирированного поиска. Размер популяции имеет фиксированную величину для всех алгоритмов и составляет $|P_i| = 5000$.

Параметры модифицированного бактериального алгоритма: $I_{bak} = 1000$ – число итераций, число шагов хемотаксиса $T = 10$.

Параметры модифицированного светлячкового алгоритма: $I_f = 1000$ – число итераций алгоритма, $A_f = 500$ – общее число агентов, выполняющих поиск, $L_0 = 50$ – исходный уровень люциферина каждого агента в популяции.

Параметры модифицированного генетического алгоритма: $I_{GA} = 1000$ – число итераций алгоритма, $P_c = 0,8$ – вероятность выполнения ОК, $P_m = 0,2$ – вероятность выполнения ОМ.

В ходе вычислительного эксперимента было проведено 5 типов тестов при различных значениях весовых коэффициентов ЦФ ω_1 и ω_2 , определяющих степень значимости частных критериев. Как было отмечено выше, частными критериями в данной диссертационной работе являются количество разрезов цепей и временная задержка схемы.

Помимо разработанного программного комплекса были реализованы алгоритмы hMetis и MLPart с учетом комплексного критерия оптимизации. В результате проведения вычислительных экспериментов были определены затраты процессорного времени и качество решений (значение ЦФ) при выполнении многоуровневого алгоритма при различных значениях параметра prob и алгоритмов разбиения MLPart, hMetis.

Первая серия экспериментов была проведена при значениях весовых коэффициентов $\omega_1 = 1$ и $\omega_2 = 0$. При таких значениях весовых коэффициентов временная задержка схемы не учитывается, а вычисляется только количество разрезов.

Результаты вычисления значений ЦФ и времени выполнения разработанных алгоритмов приведены в таблицах 4.2 и 4.3 соответственно.

Сравнительные диаграммы значений ЦФ и времени выполнения алгоритмов при $\omega_1 = 1, \omega_2 = 0$ приведены на рисунка 4.10 и 4.11 соответственно.

Таблица 4.2 Значение ЦФ при $\omega_1 = 1$ и $\omega_2 = 0$.

№ тестовой схемы	Кол-во элементов схемы	$\omega_1 = 1, \omega_2 = 0$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		шт.	шт.	шт.	шт.	шт.
1	12757	267	250	247	252	219
2	19501	320	348	322	324	300
3	23134	885	903	890	897	753
4	27502	550	592	542	580	520
5	29348	1777	1841	1782	1803	1582
6	32494	728	696	703	690	619
7	45927	855	846	851	855	788
8	53308	1246	1354	1239	1310	1158
9	53391	591	555	560	552	516
10	69427	1332	1419	1359	1407	1305
11	70554	914	926	910	918	859
12	71075	2304	2676	2409	2650	2215
13	84198	1110	1247	1206	1243	1065
14	147604	2092	2043	2040	2065	2029

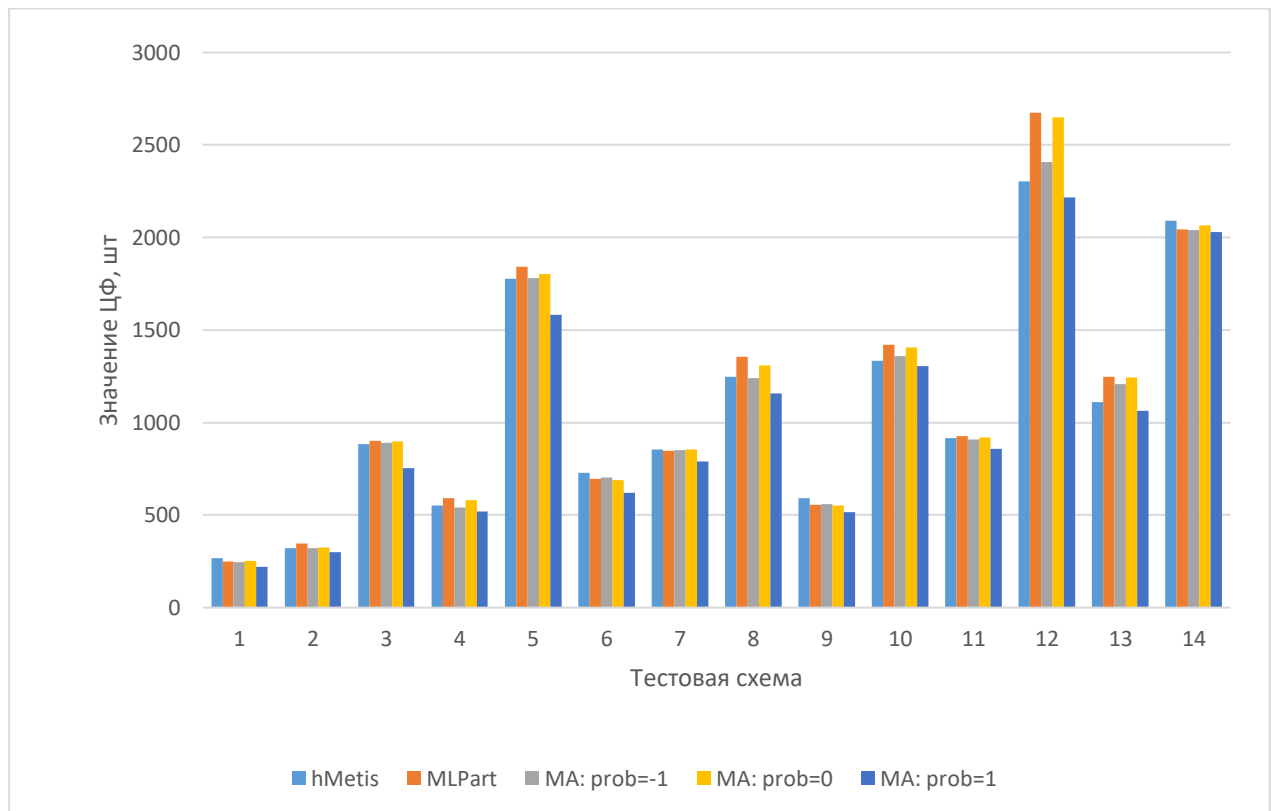
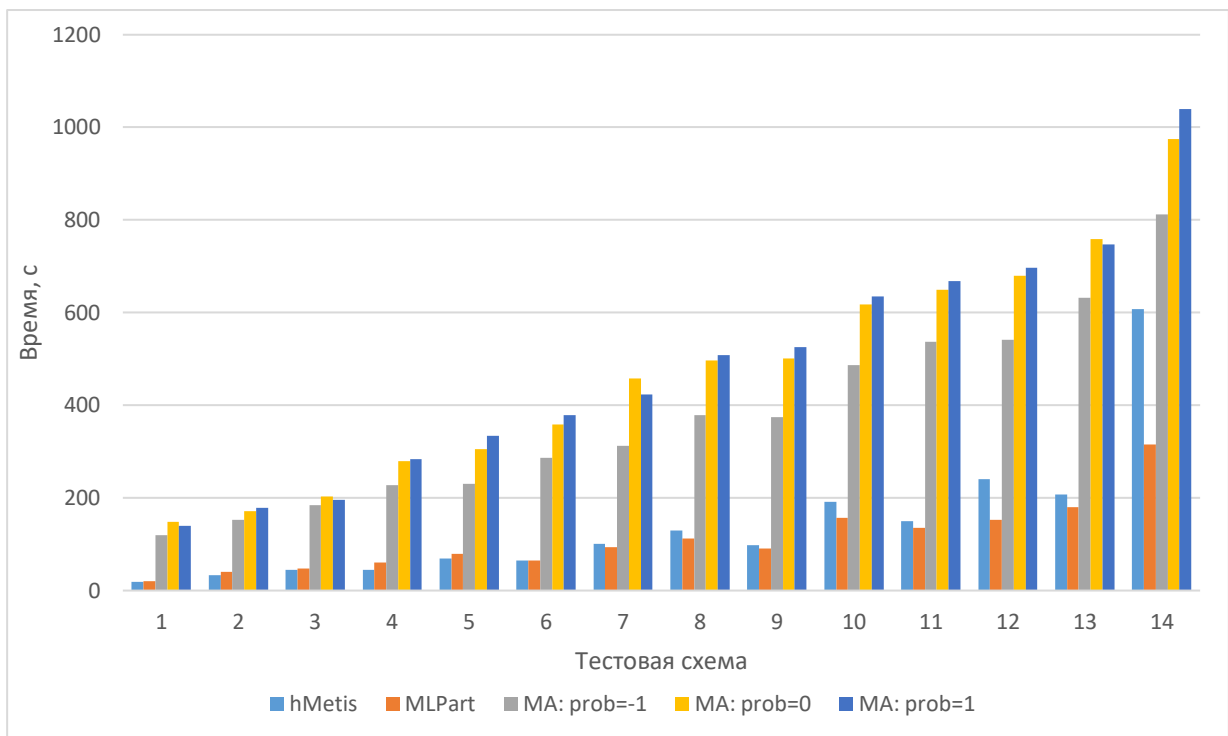
Рисунок 4.10 –Значения ЦФ алгоритмов при $\omega_1 = 1, \omega_2 = 0$

Таблица 4.3 Врем выполнения алгоритмов при $\omega_1 = 1$ и $\omega_2 = 0$.

№ тестовой схемы	Кол-во элементов в схеме	$\omega_1 = 1, \omega_2 = 0$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		с.	с.	с.	с.	с.
1	12757	19	20	120	148	139
2	19501	33	40	153	172	178
3	23134	45	48	184	203	196
4	27502	45	60	228	279	284
5	29348	69	79	231	305	334
6	32494	65	65	287	359	378
7	45927	101	94	312	458	424
8	53308	130	112	378	497	509
9	53391	98	91	374	501	525
10	69427	192	157	487	617	635
11	70554	150	136	537	649	668
12	71075	241	153	542	680	697
13	84198	208	180	632	759	748
14	147604	607	316	812	975	1039

Рисунок 4.11 – Сравнительная диаграмма времени работы алгоритмов при $\omega_1 = 1, \omega_2 = 0$

Проанализировав результаты первой серии экспериментов, можно сделать вывод, что независимо от количества элементов в схеме разработанный МА при параллельном режиме работы получает решения, превосходящие по качеству в среднем на 6,97% по сравнению с другими режимами работы и алгоритмами hMetis и MLPart. При этом разработанный алгоритм по времени решения уступает остальным алгоритмам, но его временная сложность лежит в пределах $O(n^2)$.

Вторая серия экспериментов проводилась при $\omega_1 = 0,75$ и $\omega_2 = 0,25$. Результаты вычисления ЦФ и затрат времени при выполнении алгоритмов приведены в таблицах 4.4 и 4.5.

Сравнительные диаграммы значений ЦФ и времени выполнения алгоритмов при $\omega_1 = 0,75, \omega_2 = 0,25$ приведены на рисунках 4.12 и 4.13 соответственно.

Таблица 4.4 Значение ЦФ при $\omega_1 = 0,75$ и $\omega_2 = 0,25$.

№ тестовой схемы	Кол-во элементов схемы	$\omega_1 = 0,75, \omega_2 = 0,25$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		шт.	шт.	шт.	шт.	шт.
1	12757	234	223	211	215	187
2	19501	281	310	274	276	256
3	23134	777	805	759	765	642
4	27502	483	528	462	494	443
5	29348	1560	1642	1519	1537	1348
6	32494	639	621	599	588	528
7	45927	751	755	725	729	672
8	53308	1094	1208	1056	1117	987
9	53391	519	495	477	470	440
10	69427	1169	1266	1158	1199	1112
11	70554	802	826	776	782	732
12	71075	2023	2387	2053	2259	1888
13	84198	975	1112	1028	1059	908
14	147604	1837	1822	1739	1760	1729

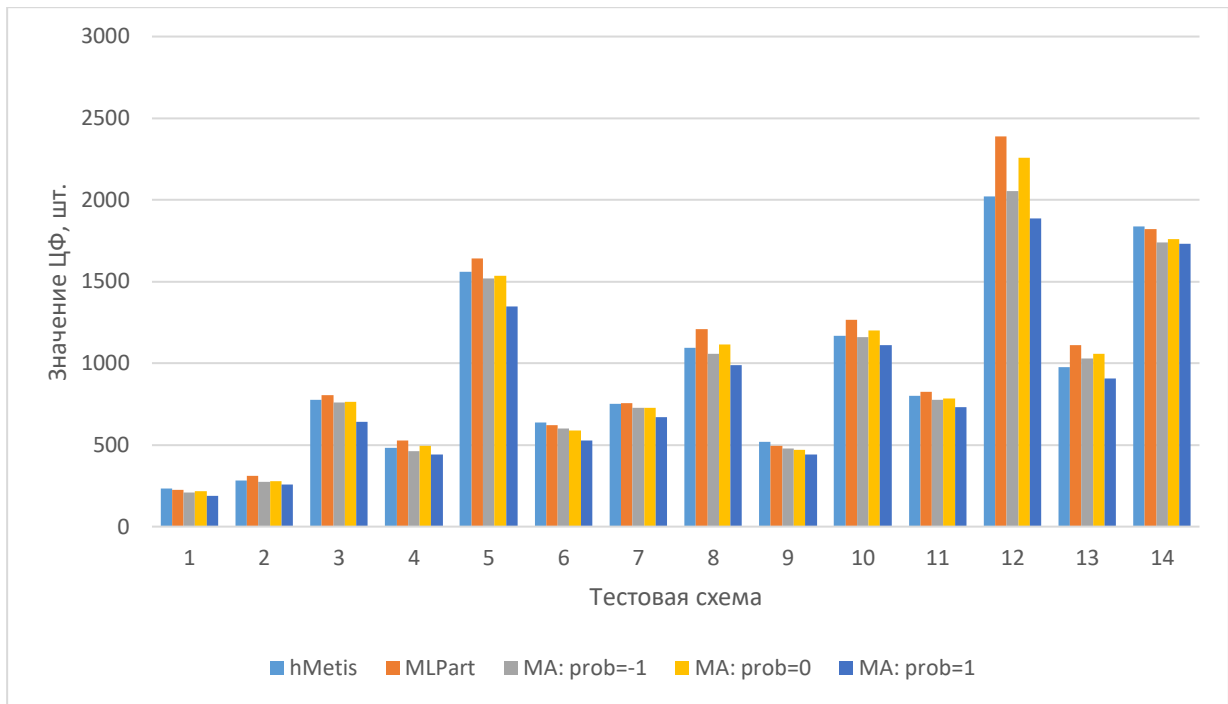


Рисунок 4.12 – Сравнительная диаграмма значений ЦФ алгоритмов при $\omega_1 = 0,75$, $\omega_2 = 0,25$.

Таблица 4.5 Время выполнения алгоритмов при $\omega_1 = 0,75$ и $\omega_2 = 0,25$.

№ тестовой схемы	Кол-во элементов в схеме	$\omega_1 = 0,75, \omega_2 = 0,25$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		с.	с.	с.	с.	с.
1	12757	23	26	146	178	195
2	19501	40	52	186	207	250
3	23134	54	62	223	244	275
4	27502	54	77	277	336	398
5	29348	83	102	280	367	469
6	32494	78	84	348	432	530
7	45927	122	121	379	551	595
8	53308	157	144	459	598	714
9	53391	118	117	454	603	737
10	69427	232	202	591	742	891
11	70554	181	175	652	781	937
12	71075	291	197	658	818	978
13	84198	251	232	767	913	1050
14	147604	732	407	986	1173	1458

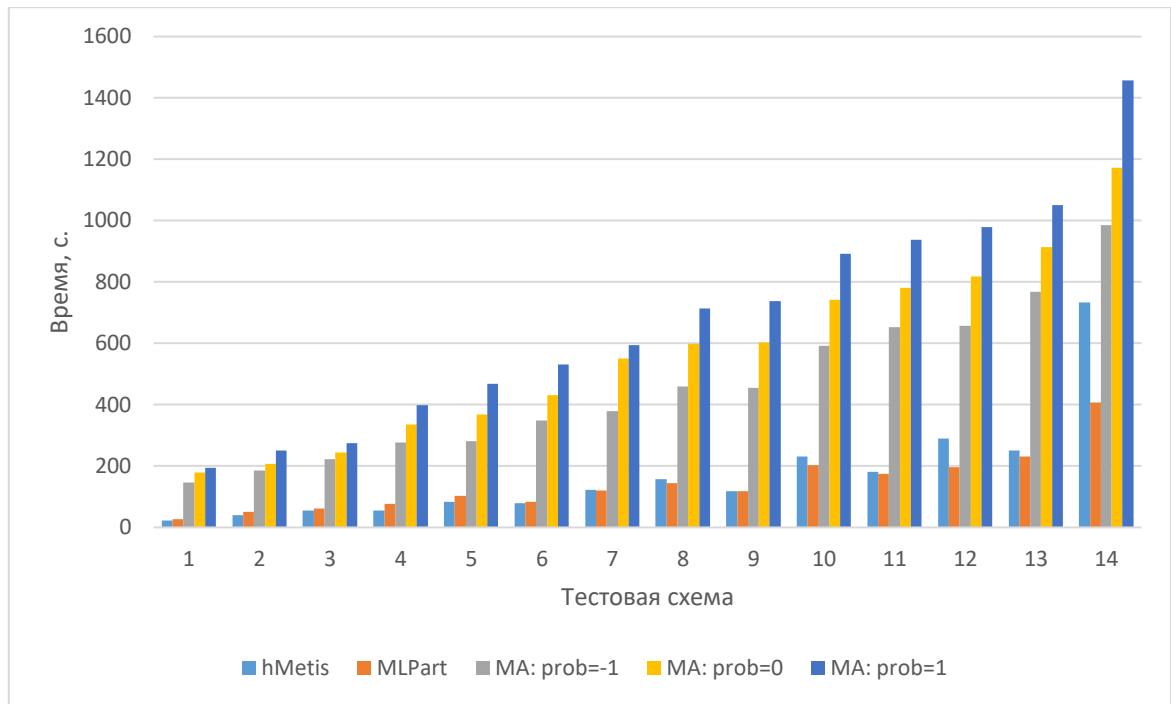


Рисунок 4.13 – Сравнительная диаграмма времени работы алгоритмов при $\omega_1 = 0,75$, $\omega_2 = 0,25$.

Проанализировав полученные результаты, можно сделать вывод, что многоуровневый алгоритм разбиения СБИС при параллельном выполнении модифицированных бактериального и светлячкового алгоритмов позволяет получать решения в среднем на 9,69% превышающие по качеству остальные алгоритмы, а его временная сложность лежит в пределах $O(n^2)$.

Третья серия экспериментов была выполнена при $\omega_1 = 0,5$ и $\omega_2 = 0,5$. Результаты вычисления ЦФ и затрат времени на выполнение алгоритмов приведены в таблицах 4.6 и 4.7 соответственно.

Сравнительные диаграммы значений ЦФ и времени выполнения алгоритмов на различных тестовых схемах при $\omega_1 = 0,5$, $\omega_2 = 0,5$ приведены на рисунках 4.14 и 4.15 соответственно.

Таблица 4.6 Значение ЦФ при $\omega_1 = 0,5$ и $\omega_2 = 0,5$.

№ тестовой схемы	Кол-во элементов схемы	$\omega_1 = 0,5, \omega_2 = 0,5$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		шт.	шт.	шт.	шт.	шт.
1	12757	240	226	212	217	191
2	19501	287	314	277	278	262
3	23134	795	815	765	771	658
4	27502	494	534	466	499	454
5	29348	1596	1661	1533	1550	1382
6	32494	654	628	605	593	541
7	45927	768	763	732	735	688
8	53308	1119	1221	1066	1126	1011
9	53391	531	501	482	474	451
10	69427	1196	1280	1169	1209	1140
11	70554	821	835	783	789	750
12	71075	2069	2414	2072	2278	1934
13	84198	997	1125	1037	1068	930
14	147604	1879	1843	1754	1775	1772

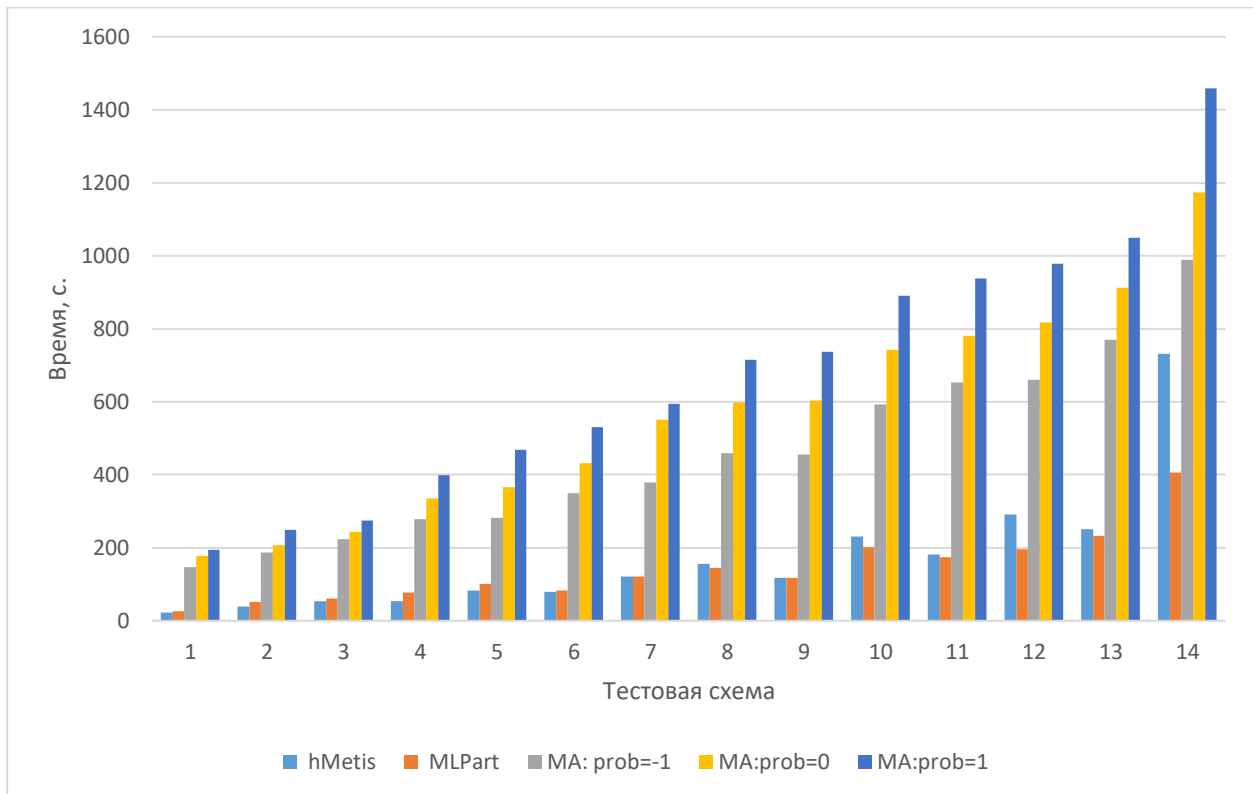
Рисунок 4.14 – Сравнительная диаграмма значений ЦФ при $\omega_1 = 0,5$, $\omega_2 = 0,5$.

Таблица 4.7 Время выполнения алгоритмов при $\omega_1 = 0,5$ и $\omega_2 = 0,5$.

№ тестовой схемы	Кол-во элементов в схеме	$\omega_1 = 0,5, \omega_2 = 0,5$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		с.	с.	с.	с.	с.
1	12757	23	26	146	178	195
2	19501	40	52	186	207	250
3	23134	54	62	224	244	275
4	27502	54	77	278	336	398
5	29348	83	102	281	367	469
6	32494	78	84	349	432	530
7	45927	122	121	380	551	595
8	53308	157	144	460	598	714
9	53391	118	117	455	603	737
10	69427	231	202	593	742	891
11	70554	181	175	654	781	937
12	71075	290	197	660	818	978
13	84198	251	232	769	913	1050
14	147604	732	407	988	1173	1458

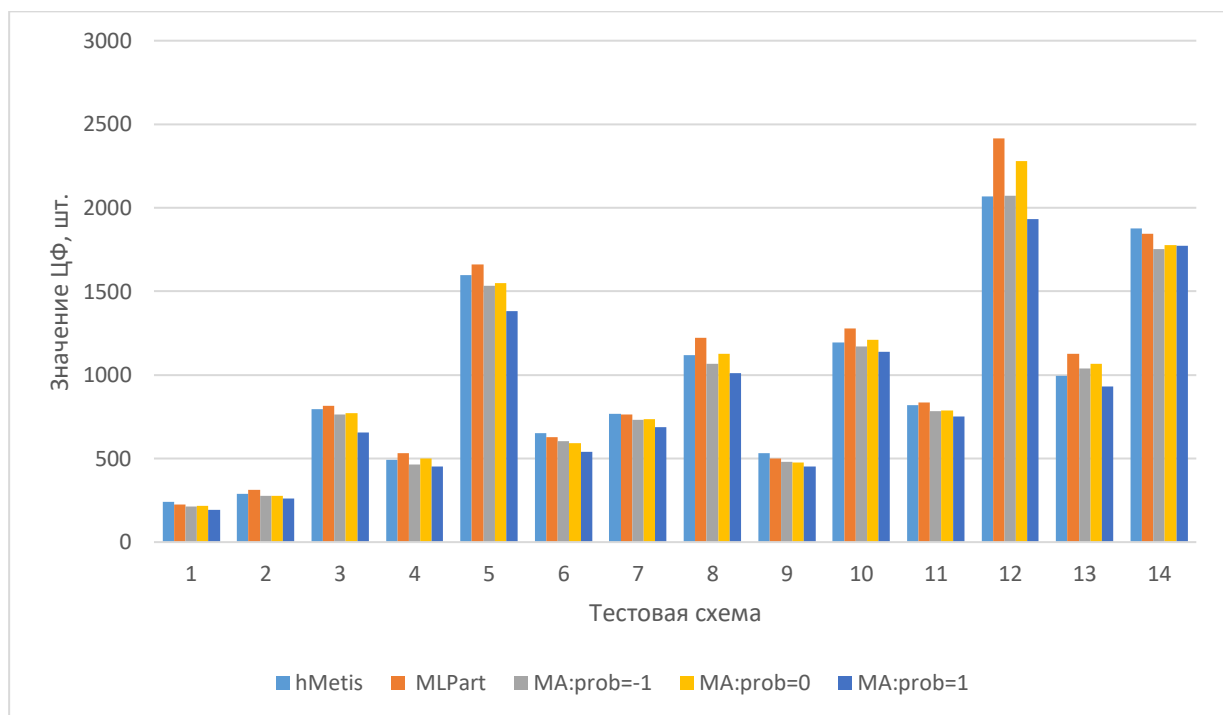


Рисунок 4.15 – Сравнительная диаграмма времени работы алгоритмов при $\omega_1 = 0,5$, $\omega_2 = 0,5$.

Проанализировав полученные таблицы и диаграммы, можно сделать вывод, что многоуровневый алгоритм разбиения СБИС при параллельном выполнении модифицированных бактериального и светлячкового алгоритмов позволяет получать решения в среднем на 9,53% превышающие по качеству остальные алгоритмы, а его временная сложность также, как и в предыдущих серия экспериментов, лежит в пределах $O(n^2)$.

Четвертая серия экспериментальных исследований была проведена при весовых параметрах $\omega_1 = 0,25$ и $\omega_2 = 0,75$. Результаты вычисления ЦФ и затрат процессорного времени приведены в таблицах 4.8 и 4.9.

Сравнительные диаграммы значений ЦФ и времени выполнения алгоритмов при $\omega_1 = 0,25$, $\omega_2 = 0,75$ приведены на рисунках 4.16 и 4.17 соответственно.

Таблица 4.8 Значение ЦФ при $\omega_1 = 0,25$ и $\omega_2 = 0,75$.

№ тестовой схемы	Кол-во элементов схемы	$\omega_1 = 0,25, \omega_2 = 0,75$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		шт.	шт.	шт.	шт.	шт.
1	12757	254	236	233	230	208
2	19501	305	329	303	295	285
3	23134	843	854	838	818	716
4	27502	524	560	511	529	495
5	29348	1692	1741	1679	1644	1505
6	32494	693	658	662	629	589
7	45927	814	800	802	780	750
8	53308	1187	1280	1167	1195	1101
9	53391	563	525	528	503	491
10	69427	1268	1342	1280	1283	1241
11	70554	870	876	857	837	817
12	71075	2194	2531	2269	2417	2107
13	84198	1057	1179	1136	1134	1013
14	147604	1992	1932	1922	1883	1930

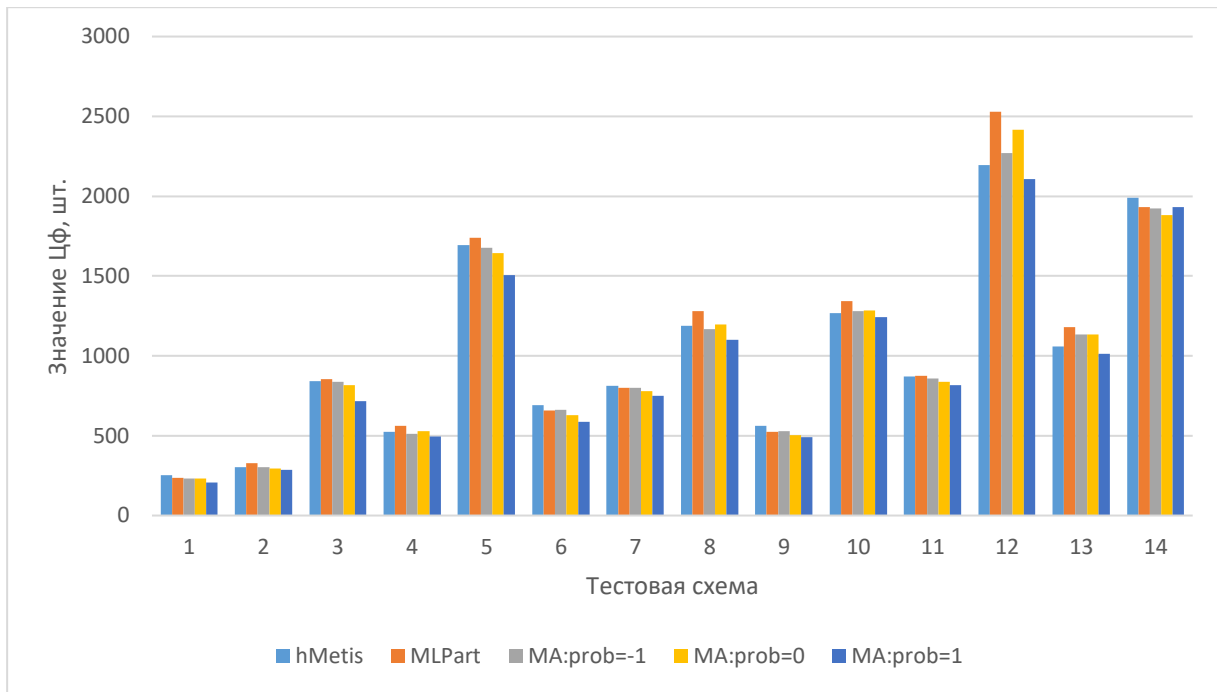


Рисунок 4.16 – Сравнительная диаграмма значений ЦФ алгоритмов при $\omega_1 = 0,25$, $\omega_2 = 0,75$.

Таблица 4.9 Время выполнения алгоритмов при $\omega_1 = 0,25$ и $\omega_2 = 0,75$.

№ тестовой схемы	Кол-во элементов в схеме	$\omega_1 = 0,25, \omega_2 = 0,75$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		с.	с.	с.	с.	с.
1	12757	23	26	146	178	195
2	19501	40	52	186	207	250
3	23134	54	62	224	244	275
4	27502	54	77	278	336	398
5	29348	83	102	281	367	469
6	32494	78	84	349	432	530
7	45927	122	121	380	551	595
8	53308	157	144	460	598	714
9	53391	118	117	455	603	737
10	69427	231	202	593	742	891
11	70554	181	175	654	781	937
12	71075	290	197	660	818	978
13	84198	251	232	769	913	1050
14	147604	732	407	988	1173	1458

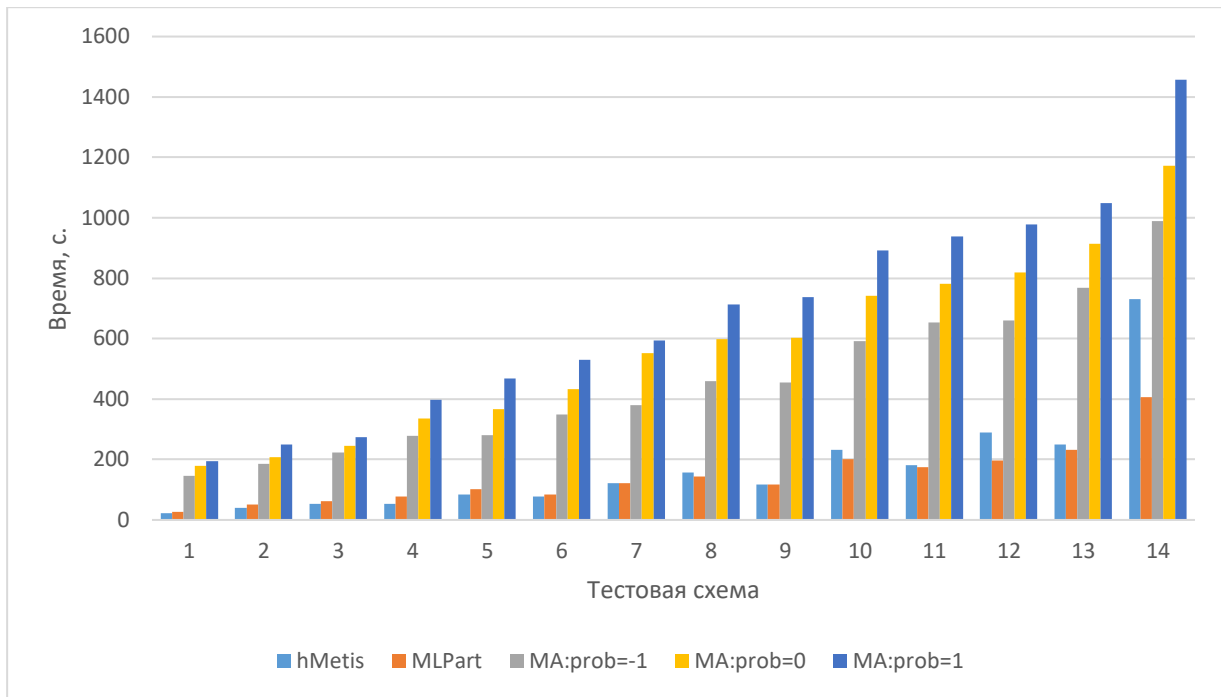


Рисунок 4.17 – Сравнительная диаграмма времени работы алгоритмов при $\omega_1 = 0,25$, $\omega_2 = 0,75$.

Проанализировав полученные таблицы и диаграммы, можно сделать вывод, что многоуровневый алгоритм разбиения СБИС при параллельном выполнении модифицированных бактериального и светлячкового алгоритмов позволяет получать решения в среднем на 7,07% превышающие по качеству остальные алгоритмы, а его временная сложность также, как и в предыдущих серия экспериментов, лежит в пределах $O(n^2)$.

Пятая серия экспериментов выполнялась при значении весовых параметров $\omega_1 = 0$ и $\omega_2 = 1$. При таких значениях весовых коэффициентов количество разрезов в схеме не учитывается, а вычисляется только временная задержка схемы. Полученные результаты приведены в таблицах 4.10 и 4.11.

Сравнительные диаграммы значений ЦФ и времени выполнения алгоритмов при $\omega_1 = 0$, $\omega_2 = 1$ приведены на рисунках 4.18 и 4.19 соответственно.

Таблица 4.10 Значение ЦФ при $\omega_1 = 0$ и $\omega_2 = 1$.

№ тестовой схемы	Кол-во элементов схемы	$\omega_1 = 0, \omega_2 = 1$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		шт.	шт.	шт.	шт.	шт.
1	12757	334	303	331	305	287
2	19501	400	421	432	393	394
3	23134	1107	1094	1194	1087	988
4	27502	688	717	727	703	682
5	29348	2223	2230	2390	2185	2076
6	32494	911	843	943	836	812
7	45927	1070	1025	1141	1036	1034
8	53308	1559	1640	1662	1588	1520
9	53391	739	672	751	669	677
10	69427	1666	1719	1823	1705	1713
11	70554	1143	1121	1220	1113	1127
12	71075	2882	3241	3231	3212	2907
13	84198	1389	1510	1617	1507	1398
14	147604	2617	2474	2736	2503	2663

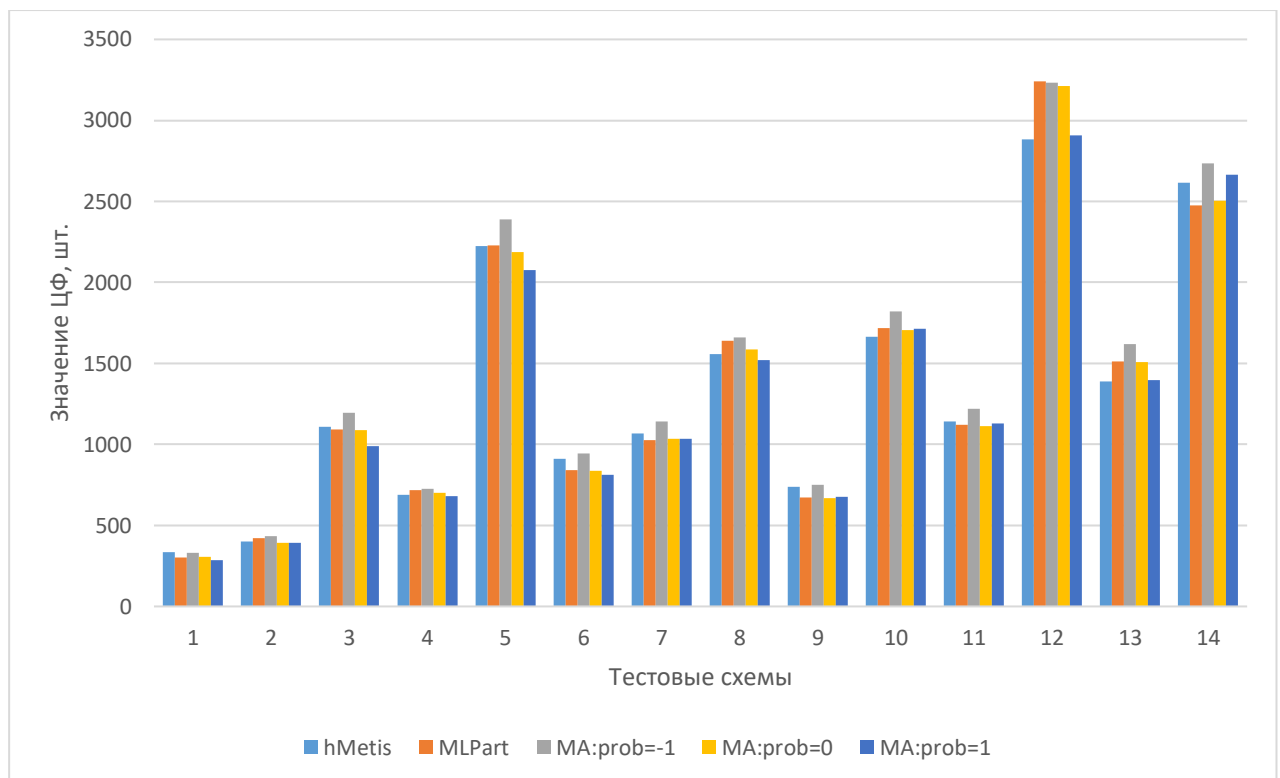
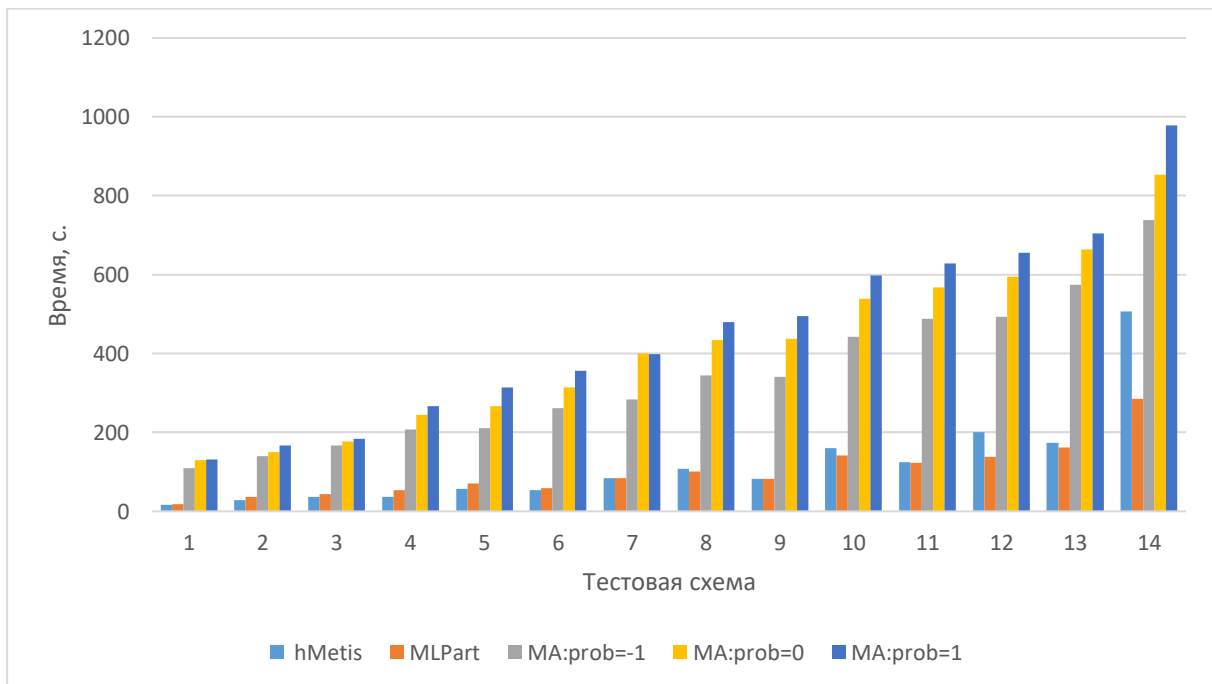
Рисунок 4.18– Сравнительная диаграмма значений ЦФ алгоритмов при $\omega_1 = 0$, $\omega_2 = 1$.

Таблица 4.9 Время выполнения алгоритмов при $\omega_1 = 0$ и $\omega_2 = 1$.

№ тестовой схемы	Кол-во элементов в схеме	$\omega_1 = 0, \omega_2 = 1$				
		hMetis	MLPart	Многоуровневый алгоритм		
				prob=-1	prob=0	prob=1
		с.	с.	с.	с.	с.
1	12757	16	18	109	129	131
2	19501	28	36	139	150	168
3	23134	38	43	167	178	184
4	27502	38	54	207	244	267
5	29348	58	71	210	267	314
6	32494	54	59	261	314	356
7	45927	84	85	284	400	399
8	53308	109	101	344	435	479
9	53391	82	82	340	438	494
10	69427	160	141	443	540	598
11	70554	125	123	489	567	629
12	71075	201	138	493	595	656
13	84198	174	162	575	664	704
14	147604	507	285	739	853	978

Рисунок 4.19 – Сравнительная диаграмма времени работы алгоритмов при $\omega_1 = 0$, $\omega_2 = 1$.

Проведя анализ полученных данных, можно сделать вывод, что многоуровневый алгоритм разбиения СБИС при параллельном выполнении модифицированных бактериального и светлячкового алгоритмов позволяет получать решения в среднем на 2,41% превышающие по качеству остальные алгоритмы, а его временная сложность также, как и в предыдущих серия экспериментов, лежит в пределах $O(n^2)$.

Полученные результаты позволяют сделать вывод, что во всех режимах работы временная сложность многоуровневого алгоритма разбиения СБИС при всех режимах работы лежит в пределах $O(n^2)$.

Исходя из приведенных экспериментальных исследований, можно сделать вывод, что разработанный в данной диссертационной работе многоуровневый алгоритм, ориентированный на решение задачи разбиения СБИС способствует получению квазиоптимальных решений за время близкое к полиномиальному (при этом ВСА стремится к $O(n^2)$), обладает свойством гибкости и достаточно высоким уровнем быстродействия при работе с одними и теми же исходными данными.

4.6 Краткие выводы

1. В ходе проведения экспериментальных исследований были разработаны описанные выше стратегии биоинспирированного поиска, что выявило их преимущество по сравнению с аналогичными методами решения задачи разбиения СБИС. В случае постоянства настроечных параметров разработанный алгоритм стремится к полиномиальной ВСА, что было подтверждено эмпирически, а значит данный алгоритм является эффективным.

2. Расходы памяти программной подсистемы состоят из нескольких частей: объем памяти, который необходим для данных многоуровневого алгоритма и зависит количества входных данных задачи, и дополнительный объем памяти, необходимый для работы разработанной подсистемы, который является постоянной величиной и не зависит от количества входных данных (около 15%).

3. Проведенные экспериментальные исследования способствовали уточнению теоретической ВСА для схем различной размерности. При реализации параллельного режима работы качество разбиения СБИС в среднем на 6,84% превосходит результаты, которые были получены алгоритмами hMetis и MLPart, что говорит не только о достаточной эффективности разработанного многоуровневого алгоритма разбиения СБИС, но и об эффективности разработанной программной подсистемы.

ЗАКЛЮЧЕНИЕ

Основным результатом проведенного исследования является повышение качества разбиения схем за счет использования разработанных алгоритмов, инспирированных природными системами, которые позволили получать квазиоптимальные решения в рамках полиномиального времени работы алгоритмов.

Проведенные теоретические и практические исследования позволили получить следующие научные результаты:

1. Проведен обзор и последующий анализ актуальных методов разбиения схем, определены их сильные и слабые стороны, а также обоснована актуальность и важность проведения данного исследования. Приведена постановка задачи разбиения схем, в основе которой лежит гиперграфовой модель, с учетом комплексного критерия оптимизации.

2. Проанализированы и разработаны биоинспирированные методы и средства оптимизации, направленные на поиск решений высокого качества при выполнении проектной процедуры разбиения схем.

3. Разработана комбинированная архитектура биоинспирированного поиска, которая позволяет эффективно использовать преимущества биоинспирированных методов оптимизации, а также организовать их работу в параллельном режиме, что способствует повышению качества получаемых решений и сокращению области поиска.

4. Разработаны модифицированные методы бактериальной, светлячковой и генетической оптимизации, позволяющие избегать локальных оптимумов.

5. Построены модифицированные генетические операторы на основе теории прямоугольных чисел и законе распределения Больцмана, направленные на увеличение вероятности сохранения в популяции решений с высоким значением ЦФ, что способствует уменьшению интервала неопределенности.

6. Разработан унифицированный механизм кодирования и декодирования альтернативных решений, позволяющий сократить потери процессорного времени

при передаче данных между различными этапами работы многоуровневого алгоритма.

7. Разработаны модифицированные биоинспирированные алгоритмы (бактериальный, светлячковый и генетический), направленные на решение задачи разбиения схем, которые позволяют частично избежать предварительной сходимости и находят решения высокого качества в пределах полиномиального времени работы алгоритма.

8. Разработан многоуровневый алгоритм поиска оптимальных решений, в основе которого лежат методы бактериального, светлячкового и генетического поиска. Данный алгоритм позволяет уменьшить время работы алгоритма за счет реализации параллельной работы нескольких биоинспирированных алгоритмов.

9. Разработан программный комплекс для исследования характеристик разработанных методов и алгоритмов и их сравнения с известными аналогами.

10. Проведено 5 типов экспериментов на тестовых схемах IBM. Качество решений, получаемых при параллельном режиме работы многоуровневого алгоритма, в среднем на 6,84% превосходит качество решений, полученных известными алгоритмами hMetis и MLPart, что говорит об эффективности разработанных в диссертации методов и алгоритмов.

Совокупность полученных в диссертации теоретических и практических результатов позволяет сделать вывод о том, что цель исследований достигнута, а сформулированная проблема полностью решена.

СПИСОК ЛИТЕРАТУРЫ

1. Власов, Д. В. Современные проблемы информатики: философский анализ / Д. В. Власов // Экономика, статистика и информатика. Вестник УМО. – 2011. – № 2. – С. 233-239.
2. Норенков, И.П. Основы автоматизированного проектирования: учебник / И. П. Норенков. – М.: Изд-во МГТУ имени Н.Э. Баумана, 2009. – 336 с.
3. Авдеев, В.А. Компьютерное моделирование цифровых устройств / В.А. Авдеев. – Москва: Изд-во «ДМК-Пресс», 2012. – 360 с.
4. Миловзоров, О. В. Электроника: учебник для вузов / О. В. Миловзоров, И. Г. Панков. – 6-е изд., перераб. и доп. – М.: Издательство Юрайт, 2023. – 344 с.
5. Царев, Р.Ю. Программные и аппаратные средства информатики / Царев Р.Ю., Прокопенко А.В., Князьков А.Н. - Краснояр.: СФУ, 2015. – 160 с.
6. Казеннов, Г.Г. Основы проектирования интегральных схем и систем / Г.Г. Казеннов. – М.: Бином. Лаборатория знаний, 2010. – 296с.
7. Colwell, R. The chip design game at the end of Moore's Law. / R. Colwell // Proceedings of the IEEE/ACM Symposium on High-Performance Chips (Hot Chips). – ACM Press, 2013. – с. 256-262.
8. Kureichik, V.V. Algorithms for Applied CAD Problems. / V.V. Kureichik, V.M. Kureichik, S.P. Malioukov, A.S. Malioukov. // Berlin Heidelberg: Springer-Verlag, 2009. – 487 p.
9. Sherwani, N.A. Algorithms for VLSI Physical Design Automation / N.A. Sherwani // Third Edition, Kluwer Academic Publisher. – USA: 2013. – 572p.
10. Краснощеков, П.С. Оптимизация в автоматизированном проектировании / П. С. Краснощеков, В.В. Морозов, Н.М. Попов. – М.: Ленанд, 2018. – 328с.
11. Батищев, Д.И. Оптимизация в САПР / Д.И. Батищев, Я.Е. Львович, В.Н. Фролов. – Воронеж: Изд-во ВГУ, 1997.

12. Карпенко, А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие / А.П. Карпенко – М: Издательство МГТУ им. Н.Э. Баумана, 2014. – 446 с.
13. Немудров, В., Системы-на-кристалле. Проектирование и развитие / В. Немудров, Г. Мартин. – М.: Техносфера, 2004.
14. Alpert, C. J. Handbook of Algorithms for Physical design Automation / C. J. Alpert, P. M. Dinesh, S. S. Sachin // Auerbach Publications Taylor & Francis Group, USA: 2009.
15. Weste, N. CMOS VLSI Design: A Circuits and Systems Perspective / N. Weste and D. Harris // Pearson, USA: 2010. – 880p.
16. Струченков В.И. Методы оптимизации в прикладных задачах / В.И. Струченков. – Солон-пресс, 2009 г. – 320 с.
17. Кормен, Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Вильямс, 2020. – 1328с.
18. Мелихов, А.Н. Применение графов для проектирования дискретных устройств: учебник / А.Н. Мелихов, Л.С. Берштейн, В.М. Курейчик. – М.: Наука, 1974.– 304 с.
19. Зыков, А. А. Основы теории графов: учебник / А. А. Зыков. – М.: Вузовская книга, 2016. – 664 с.
20. Андерсон, Д. Дискретная математика и комбинаторика. / Д. Андерсон. – М.: Вильямс, 2019. – 960с.
21. Гладков Л.А. Дискретная математика: учебник / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. –М.: Физматлит, 2014.– 496 с.
22. Овчинников, В.А. Математические модели объектов задач структурного синтеза: Наука и образование. / В. А. Овчинников. – Инженерное образование: эл. науч. издание, 2009.
23. Уилсон, Р. Введение в теорию графов / Р. Уилсон. – М.: Вильямс, 2020. – 240с.

24. Zaruba, D. Ultragraph Model for ECE Component Partitioning / D. Zaruba, E. Kuliev, D. Zaporozhets // 2020 IEEE East-West Design and Test Symposium, EWDTS 2020. – 2020. – P. 9225027. – DOI 10.1109/EWDTS50664.2020.9225027
25. Данилина, Т. И. Время задержки сигнала в сверхбольших интегральных схемах / Т. И. Данилина, И. А. Чистоедова, А. Д. Заречнев // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2019. – Т. 22. – № 1. – С. 20-24.
26. Gargini, P.A. Roadmap evolution: from NTRS to ITRS, from ITRS 2.0 to IRDS / P. A. Gargini // Proceedings Volume 10450, International Conference on Extreme Ultraviolet Lithography 2017. — 2017. — № 1045018.
27. Лежебоков, А. А. Проблемы учета временных задержек в задачах конструкторского проектирования/ А. А. Лежебоков, А. Н. Дуккардт, // Известия ЮФУ. Технические науки. – 2011. – № 7(120). – С. 108-113.
28. Заруба, Д.В., Разработка и исследование биоинспирированного алгоритма размещения с учетом временных задержек / Д.В. Заруба, Д.Ю. Запорожец, Д.В. Лещанов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'15». – 2015. – Т.3. – С. 27-33
29. Tutuianu, B. An explicit RC-circuit delay approximation based on the first three moments of the impulse response / B. Tutuianu, F. Dartu, L. Pileggi // Proceedings of the Design Automation Conference. – 1996. – pp. 611-616
30. Баринов, С. В. Компоновка МЭС на основе итерационной кластеризации с учетом временных задержек / С. В. Баринов, В. М. Курейчик, Л. А. Гладков // Известия ТРТУ. – 2006. – № 8(63). – С. 120-126.
31. Мартюгин, Д. А. Многокритериальная оптимизация планирования сверхбольших интегральных схем с учетом временных задержек / Д. А. Мартюгин, В. В. Воронова // Современные инновации в науке и технике. Том 3. – Курск: Закрытое акционерное общество "Университетская книга", 2014. – С. 72-79.
32. Илькив, А. Статический временной анализ в физическом проектировании цифровых интегральных схем / А. Илькив, С. Подрядчиков // Компоненты и технологии. – 2021. – № 2(235). – С. 96-103.

- 33.** Подиновский, В.В. Оптимизация по последовательно применяемым критериям / В.В. Подиновский, В.М. Гаврилов. – СПб.: Ленанд, 2016. – 194с.
- 34.** Зайцев, М.Г. Методы оптимизации управления и принятия решений / М.Г. Зайцев, С.Е. Варюхин. – М.: Дело, 2017. – 640 с.
- 35.** Анализ трудоемкости конструкторского проектирования СБИС средствами гибридной иерархической клиент-серверной подсистемы / В. М. Глушань, И. А. Дубровский, О. И. Красюк, М. В. Рыбальченко // Вестник Рязанского государственного радиотехнического университета. – 2017. – № 62. – С. 71-78.
- 36.** Харари, Ф. Теория графов / Ф. Харари. - СПб.: Ленанд, 2018. – 304с.
- 37.** Макаровских, Т.А. Комбинаторика и теория графов. Учебное пособие / Т.А. Макаровских. – СПб.: Ленанд, 2020. – 216 с.
- 38.** Корниенко В.П. Методы оптимизации. –М.:Высш.шк., 2007.
- 39.** Емельянов, В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик ; В.В. Емельянов, В.М. Курейчик, В.В. Курейчик. – Москва : РГБ, 2007. – 1 с.
- 40.** Каштаева, С. В. Методы оптимизации / С. В. Каштаева ; Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский государственный аграрно-технологический университет имени академика Д.Н. Прянишникова». – Пермь : ИПЦ Прокрость, 2020. – 84 с.
- 41.** Методы оптимизации : Учебник и практикум / Ф. П. Васильев, Л. А. Артемьева, М. М. Потапов, Б. А. Будак. – 1-е изд.. – Москва : Издательство Юрайт, 2020. – 1 с. – (Высшее образование).
- 42.** Зыков, А. А. Основы теории графов: учебник / А. А. Зыков. – М.: Вузовская книга, 2016. – 664 с.
- 43.** Овчинников, В. А. Модели и методы дискретной оптимизации : Модули 1 и 2 / В. А. Овчинников. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2019. – 278 с.

44. Гладков, Л. А. Методы решения задач оптимизации / Л. А. Гладков, Н. В. Гладкова. – Ростов-на-Дону - Таганрог : Южный федеральный университет, 2019. – 118 с.
45. Ионкин, Н. И. Численные методы : Курс лекций / Н. И. Ионкин. – Москва : ООО "МАКС Пресс", 2016. – 208 с.
46. Гладков, Л. А. Генетические алгоритмы : учебник / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. – 2-е издание, исправленное и дополненное. – Москва : ООО Издательская фирма "Физико-математическая литература", 2010. – 366 с.
47. Борзунов, Г. И. Биоинспирированные алгоритмы и их применение : Конспект лекций / Г. И. Борзунов, М. А. Куприяшин ; Международный научно-методический центр. – Москва : Национальный исследовательский ядерный университет "МИФИ", 2020. – 184 с.
48. Курейчик, В.В., Бионические информационные системы и их практические применения / В. В. Курейчик, В. М. Курейчик, Л. А. Зинченко [и др.]. – Москва : ООО Издательская фирма "Физико-математическая литература", 2011. – 287 с.
49. Курейчик, В. В. Биоинспирированный поиск при проектировании и управлении / В. В. Курейчик, В. В. Курейчик // Известия ЮФУ. Технические науки. – 2012. – № 11(136). – С. 178-183.
50. Тухбатуллин, М. С. Естественные алгоритмы. Пчелиный алгоритм / М. С. Тухбатуллин // Наука: прошлое, настоящее, будущее : Международная научно-практическая конференция, Уфа, 15 августа 2015 года / Том 1. – Уфа: Общество с ограниченной ответственностью "Аэтерна", 2015. – С. 86-88.
51. Xin-She, Yang Nature-Inspired Optimization Algorithms, Academic Press, 2021, pp.123-129.
52. Алаева, Д. Р. Метод пчелиного роя для решения задач на поиск экстремума функции / Д. Р. Алаева // Вестник науки и образования. – 2018. – Т. 1, № 4(40). – С. 14-19.

53. Кубанских, О. В. Алгоритмы оптимизации, основанные на симуляции социального поведения в природе / О. В. Кубанских, С. А. Хлебород, Д. А. Чалый // . – 2014. – № 16. – С. 132-137.

54. Курейчик, В.В. Биоинспирированные методы в оптимизации / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик, П. В. Сороколетов. – Москва : ООО Издательская фирма "Физико-математическая литература", 2009. – 381 с.

55. Заруба, Д. В. Генерация биоинспирированных поисковых процедур для решения оптимизационных задач / Д. В. Заруба, Д. Ю. Запорожец // Известия ЮФУ. Технические науки. – 2016. – № 6(179). – С. 13-24

56. Сороколетов П.В. и др. Методы и алгоритмы принятия решений на основе бионического поиска: Монография.– Таганрог: изд-во ТТИ ЮФУ, 2009.

57. Эволюционное моделирование и генетические алгоритмы : методические указания к выполнению цикла лабораторных работ / В. В. Курейчик, Вл. Вл. Курейчик, Д. Ю. Запорожец, Д. В. Заруба ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования "Южный федеральный университет", Инженерно-технологическая академия. Ч. 1. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 75 с.

58. Чуканов, С. Н. Многоагентные системы : учебно-методическое пособие / С. Н. Чуканов, Н. Н. Егорова. – Омск : Сибирский государственный автомобильно-дорожный университет, 2022. – 99 с.

59. Гладков, Л. А. Эволюционирующие многоагентные системы и эволюционное проектирование / Л. А. Гладков, Н. В. Гладкова // Известия ЮФУ. Технические науки. – 2020. – № 4(214). – С. 48-59.

60. Многоагентные системы в информационных технологиях как вершина развития / Д. А. Кошелев, А. П. Частиков, К. Г. Шевцова [и др.] // Успехи современной науки. – 2016. – Т. 2, № 7. – С. 68-70.

61. Мальцев, И. Ю. Многоагентные системы: современное состояние теории и практики / И. Ю. Мальцев // Научная сессия НИЯУ МИФИ-2014. Том 3. –

Москва: Национальный исследовательский ядерный университет "МИФИ", 2014. – С. 132.

62. Хорсик, И. А. Многоагентные системы в теории принятия решений / И. А. Хорсик // Хроники цифровых трансформаций. – Издательский дом "Сириус", 2022. – С. 81-84.

63. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. -М.: Эдиториал УРСС, 2002. -352с.

64. Zaporozhets, D. Y. Hybrid bionic algorithms for solving problems of parametric optimization / D. Y. Zaporozhets, D. V. Zaruba, V. V. Kureichik // World Applied Sciences Journal. – 2013. – Vol. 23, No. 8. – P. 1032-1036. – DOI 10.5829/idosi.wasj.2013.23.08.13127

65. Karypis, G. Introduction to Parallel Computing / A. Grama, A. Gupta, G. Karypis, V. Kumar // Addison-Wesley; 2nd edition, 2003 . – 664с.

66. Alpert, C. J. Multilevel circuit partitioning / C. J. Alpert, J. H. Huang, A. B. Kahng // Proc. of the 34th ACM/IEEE Design Automation Conference, 1997.

67. Burger, M. Iterative solution methods / M. Burger, B. Kaltenbacher, A. Neubauer // Handbook of Mathematical Methods in Imaging: Volume 1, Second Edition, 2015. – P. 431-470.

68. Faísca, N. P. Bilevel and Multilevel Programming / N. P. Faísca, B. Rustem, V. Dua // Process Systems Engineering. Vol. 1-7, 2014. – P. 129-149. – DOI 10.1002/9783527631209.ch6. – EDN WSCXQN.

69. Kumar, V. A Fast and High-Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karpis, V. Kumar. // SIAM Journal on Scientific Computing, 1998. – 20 (1) . – с.359—392.

70. Snijders, T. Multilevel Analysis: An Introduction to Basic And Advanced Multilevel Modeling / T. Snijders // SAGE Publications Ltd, 2nd ed, 2011. – 368с.

71. Garson, G. Multilevel Modeling: Applications in STATA (R), IBM (R) SPSS (R), SAS (R), R, & HLM (TM) / G. Garson // SAGE Publications, Inc; 1st edition, 2019. – 552 с.

72. Kureichik, V. Combined approach to place electronic computing equipment circuit elements / V. Kureichik, V. Jr. Kureichik, D. V. Zaruba // Proceedings of 2015 IEEE East-West Design and Test Symposium, EWDTs 2015. – 2015. – Art No 7493134 (pp. 1-5). – DOI 10.1109/EWDTs.2015.7493134
73. Zaporozhets, D. Distributed approach for optimization problems / D. Zaporozhets, D. Zaruba // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 734. – P. 12145. – DOI 10.1088/1757-899X/734/1/012145
74. Kureichik, V. V. Hybrid bioinspired search for schematic design / V. V. Kureichik, V. Kureichik Jr, D. Zaruba // Advances in Intelligent Systems and Computing. – 2016. – Vol. 451. – P. 249-255. – DOI 10.1007/978-3-319-33816-3_25
75. Заруба, Д.В., Решение задачи компоновки фрагментов СБИС на основе гибридного алгоритма / Д.В. Заруба, Д.Ю. Запорожец, Д.В. Лещанов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'16». – 2016. – Т.3. – С. 71-75
76. Kureichik, V. Hybrid approach for graph partitioning / V. Kureichik, D. Zaruba // Advances in Intelligent Systems and Computing. – 2017. – Vol. 573. – P. 64-73. – DOI 10.1007/978-3-319-57261-1_7
77. Курейчик, Вл. Вл. Оценка эффективности гибридного алгоритма при решении задач конструкторского проектирования на тестовых примерах (бенчмарках) / Вл. Вл. Курейчик, Д. В. Заруба // Международный научно-технический конгресс "Интеллектуальные системы и информационные технологии – 2020". "ИС & ИТ-2020" "IS&IT'20" : труды конгресса : в 2 т. Т. 1. – Таганрог : Изд-во Ступина С. А., 2020. — С. 129-134.
78. Kureichik, V. Integrated Approach for Partitioning of Electronic Computer Equipment Blocks / V. Kureichik, D. Zaruba // Proceedings - 2019 International Russian Automation Conference, RusAutoCon 2019. – 2019. – P. 8867613. – DOI 10.1109/RUSAUTOCON.2019.8867613
79. Kureichik, V. Integrated Approach to Combinatorial and Logic Graph Problems / V. Kureichik, D. Zaruba // Lecture Notes in Electrical Engineering. – 2022. – Vol. 857 LNEE. – P. 203-213. – DOI 10.1007/978-3-030-94202-1_20.

- 80.** Запорожец, Д. Ю. Параллельный популяционный алгоритм / Д. Ю. Запорожец, Д. В. Заруба // Известия ЮФУ. Технические науки. – 2018. – № 4(198). – С. 136-145. – DOI 10.23683/2311-3103-2018-4-136-145
- 81.** Курейчик, В. В. Компоновка фрагментов СБИС на основе модели поведения роя светлячков / В. В. Курейчик, Д. Ю. Запорожец, Д. В. Заруба // Международная конференция по мягким вычислениям и измерениям. – 2016. – Т. 1. – С. 463-466
- 82.** Курейчик, В. В. Двухуровневый алгоритм разбиения графа на части / В. В. Курейчик, Д. В. Заруба // Известия ЮФУ. Технические науки. – 2019. – № 2(204). – С. 6-15. – DOI 10.23683/2311-3103-2019-2-6-15
- 83.** Биосистемы: организация, поведение, управление : Тезисы докладов 74-й Всероссийской с международным участием школы-конференции молодых ученых, посвященной памяти проф. А. П. Веселова, Нижний Новгород, 20–23 апреля 2021 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2021. – 263 с.
- 84.** Ермилова, Е. В. Подвижность и поведение микроорганизмов / Е. В. Ермилова ; Е. В. Ермилова, Ж. М. Залуцкая, Т. В. Лапина; С.-Петерб. гос. ун-т. – СПб. : Изд-во С.-Петерб. ун-та, 2010. – 187 с.
- 85.** Архипова, Н. Д. Факторы адаптации бактерий / Н. Д. Архипова // Актуальные проблемы сельского хозяйства горных территорий : Материалы V-й Международной научно-практической конференции, Горно-Алтайск, 24–27 июня 2015 года / Ответственный за выпуск: Марченко В.А.. – Горно-Алтайск: Горно-Алтайский государственный университет, 2015. – С. 68-70.
- 86.** Zaruba, D. Parametric optimization based on bacterial foraging optimization / D. Zaruba, D. Zaporozhets, E. Kuliev // Advances in Intelligent Systems and Computing. – 2017. – Vol. 573. – P. 54-63. – DOI 10.1007/978-3-319-57261-1_6
- 87.** Zaporozhets, D. Bacterial foraging optimization for VLSI fragments placement / D. Zaporozhets, D. Zaruba // Advances in Intelligent Systems and Computing. – 2018. – Vol. 679. – P. 341-348. – DOI 10.1007/978-3-319-68321-8_35

- 88.** Multimodal function optimization using synchronous bacterial foraging optimization technique / K. Bakwad, S. Pattnaik, S. Devi [et al.] // . – 2010. – Vol. 56, No. 2. – P. 80-87.
- 89.** Panteleev, A. Multiagent self-organizing interval bacterial colony evolution optimization algorithm / A. Panteleev, V. Panovski // *Advances in Intelligent Systems and Computing*. – 2016. – Vol. 450. – P. 451-461.
- 90.** Яковлева, Е. А. Роевой интеллект в роботизированном решении пространственных задач / Е. А. Яковлева, А. А. Сорокин, Р. А. Коваленко. – Казань : Общество с ограниченной ответственностью "Бук", 2020. – 104 с.
- 91.** Krishnanand, K. A Glowworm Swarm Optimization Based Multi-robot System for Signal Source Localization / K. Krishnanand, D. Ghose // *Studies in Computational Intelligence*, 2009. – 177. – с. 49-68.
- 92.** Курейчик, В. В. Алгоритм параметрической оптимизации на основе модели поведения роя светлячков / В. В. Курейчик, Д. В. Заруба, Д. Ю. Запорожец // *Известия ЮФУ. Технические науки*. – 2015. – № 6(167). – С. 6-15
- 93.** Черникова, И. В. Идея универсальной эволюции и конвергентные процессы в современной науке / И. В. Черникова // *Революция и эволюция: модели развития в науке, культуре, обществе : труды II Всероссийской научной конференции, Нижний Новгород, 29 ноября – 01 2019 года*. – Нижний Новгород: Общество с ограниченной ответственность "Красная ласточка", 2019. – С. 16-18.
- 94.** Заруба, Д. В. Эволюционное моделирование и генетические алгоритмы / В.В. Курейчик, Д.Ю. Запорожец, Д.В. Заруба // *Учебно-методическое пособие к выполнению цикла лабораторных работ по курсу «Эволюционное моделирование и генетические алгоритмы»*. – Ростов-на-Дону: Изд-во ЮФУ, 2017. – 72 с.
- 95.** Утина, А. П. Эволюция живых систем / А. П. Утина, К. Р. Аушева // *Новые подходы к осмыслению старых проблем : Сборник статей Всероссийской научной конференции в рамках научной школы "Термодинамическая экология и информационная энтропия"*, Москва, 12 декабря 2019 года. – Москва: Общество с ограниченной ответственностью "Издательство "Мир науки", 2020. – С. 196-205.

96. Заруба, Д.В. Разработка и исследование биоинспирированного алгоритма размещения с учетом временных задержек / Д.В. Заруба, Д.Ю. Запорожец, Д.В. Лещанов // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'15». – 2015. – Т.3. – С. 27-33

97. Заруба, Д. В. Разработка биоэвристик для создания интеллектуальной подсистемы принятия эффективных решений NP-трудных и NP-сложных комбинаторно-логических задач на графах / Д. В. Заруба, Э. В. Кулиев, Д. Ю. Запорожец, М. М. Семенова // Известия ЮФУ. Технические науки. – 2021. – № 4(221). – С. 117-128. – DOI 10.18522/2311-3103-2021-4-117-128

98. Курейчик, В. М. Модифицированные генетические операторы / В. М. Курейчик // Известия ЮФУ. Технические науки. – 2009. – № 12(101). – С. 7-14.

99. Гладков, Л. А. Интегрированный алгоритм решения задачи разбиения / Л. А. Гладков, Н. В. Гладкова, А. И. Беседина // Информационные технологии, системный анализ и управление (ИТСАУ-2016) : сборник трудов XIV Всероссийской научной конференции молодых ученых, аспирантов и студентов, Таганрог, 16–19 ноября 2016 года / Том 1. – Таганрог: Южный федеральный университет, 2016. – С. 151-155.

100. Брежнева, О. В. Фигурные числа и их применение / О. В. Брежнева, С. В. Корнев // . – 2016. – № 5-2. – С. 39-41.

101. Запорожец, Д. Ю. Об одном способе кодирования решения для задачи размещения / Д. Ю. Запорожец, Д. В. Заруба, А. А. Лежебоков // Известия ЮФУ. Технические науки. – 2012. – № 11(136). – С. 183-188

102. Заруба, Д. В. Компоновка блоков ЭВА на основе биоинспирированных методов поиска / Д. В. Заруба // Современные компьютерные технологии : сборник статей научно-методической конференции, Таганрог, 25–29 февраля 2020 г.. – Таганрог, 2020. – С. 4-9.

103. Свидетельство о государственной регистрации программы для ЭВМ № 2017663572 Российская Федерация. Программная реализация бактериального алгоритма размещения разногабаритных элементов на плоскости : № 2017660364 :

заявл. 16.10.2017 : опубл. 07.12.2017 / Д. Ю. Запорожец, Д. В. Заруба ; правообладатель федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет» (Южный федеральный университет)

104. Zaruba, D. Glowworm Swarm Optimization Algorithm for Computer Equipment Partitioning / D. Zaruba, D. Zaporozhets, N. Kulieva // 2019 International Seminar on Electron Devices Design and Production, SED 2019. – 2019. – P. 8798435. – DOI 10.1109/SED.2019.8798435

105. Заруба, Д. В. Светлячковый алгоритм компоновки блоков ЭВА / Д. В. Заруба // Информационные технологии, системный анализ и управление : сборник трудов XVI Всероссийской научной конференции молодых ученых, аспирантов и студентов (Таганрог, 5–7 декабря 2018 г.) : в 3 томах. Т. 1 / сост. Е. Ю. Косенко, А. Я. Номерчук, А. Е. Титов, В. В. Шадрин [и др.]. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2018. – С. 17-22.

106. Курейчик, В. В. Компоновка фрагментов СБИС на основе модели поведения роя светлячков / В. В. Курейчик, Д. Ю. Запорожец, Д. В. Заруба // Международная конференция по мягким вычислениям и измерениям. – 2016. – Т. 1. – С. 463-466

107. Kureichik, V. V. Partitioning of VLSI fragments based on the model of glowworm's behavior / V. V. Kureichik, D. Y. Zaporozhets, D. V. Zaruba // Proceedings of the 19th International Conference on Soft Computing and Measurements, SCM 2016. – 2016. – P. 268-272. – DOI 10.1109/SCM.2016.7519750

108. Кулиева, Н. В. Модифицированный алгоритм биоинспирированной оптимизации в задачах конструкторского проектирования / Н. В. Кулиева, Д. В. Заруба, Д. Ю. Терещенко // V Всероссийская научно-техническая конференция молодых ученых, аспирантов, магистрантов и студентов "Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности" : сборник статей Всероссийской научно-технической конференции, 01-07 апреля 2019 г. – Таганрог, 2019. – С. 263-265

109. Гниденко, И. Г. Многопроцессорные системы и параллельное программирование: Учебное пособие / И. Г. Гниденко, С. К. Морозов, Д. Ю. Федоров. – Санкт-Петербург : Санкт-Петербургский государственный экономический университет, 2019. – 68 с.

110. Козаев, В. Ю. Многопроцессорные вычислительные системы / В. Ю. Козаев // Теория. Практика. Инновации. – 2018. – № 12(36). – С. 31-35.

111. Многопроцессорные вычислительные и управляющие системы : материалы Всероссийской научно-технической конференции (МВУС-2022), Таганрог, 27–30 июня 2022 года. – Ростов-на-Дону: Южный федеральный университет, 2022. – 252 с.

112. Филимонов, В. В. Методы оценки и повышения производительности многопроцессорных вычислительных систем / В. В. Филимонов, В. В. Сускин // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2021. – № 1. – С. 127-131.

113. Kureichik, V. Generation of bioinspired search procedures for optimization problems / V. Kureichik, D. Zaporozhets, D. Zaruba // Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings. – 2016. – P. 7991822. – DOI 10.1109/ICAICT.2016.7991822

114. Планирование и организация эксперимента / В. И. Балабанов, А. Ли, Н. Б. Мартынова [и др.]. – Ташкент : Бухарский институт управления природными ресурсами Национального исследовательского университета "Ташкентский институт инженеров ирригации и механизации сельского хозяйства", 2021. – 120 с.

115. Осипенко, С. А. Статистические методы обработки и планирования эксперимента / С. А. Осипенко. – Москва-Берлин : Директ-Медиа, 2020. – 62 с.

116. Симогин, А. А. Применение планирования эксперимента при построении математической модели в области оптимума / А. А. Симогин // Сборник научно-методических работ. Том Выпуск 11. – Донецк : Донецкий национальный технический университет, 2019.

117. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен ; Т. Кормен, Ч. Лейзерсон, Р. Ривест ; [пер. с англ. под ред. А. Шеня]. – [2-е изд., стер.]. – Москва : БИНОМ. Лаб. знаний ; 2004. – 955 с.

118. Бальчугов, А. В. Основы научных исследований, организация и планирование эксперимента / А. В. Бальчугов, А. В. Бадеников. – Ангарск : Ангарский государственный технический университет, 2021. – 179 с.

119. IBM-PLACE 2.0 benchmark suits <http://er.cs.ucla.edu/benchmarks/ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz>

120. Свидетельство о государственной регистрации программы для ЭВМ № 2018616455 Российская Федерация. Программная реализация гибридного алгоритма компоновки фрагментов СБИС на основе композиции генетической и муравьиной эвристик : № 2018613873 : заявл. 18.04.2018 : опубл. 01.06.2018 / Д. Ю. Запорожец, Д. В. Заруба, В. В. Бова ; правообладатель федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет» (Южный федеральный университет)

121. Alpert, C.J., Multilevel Circuit Partitioning / J. H. Huang, A. B. Kahng // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1998. – 17(8) . – с. 655– 667.

122. Karypis, hMETIS: A Hypergraph Partitioning Package Version 1.5, / G. Karypis ,V. Kumar // user manual, 1998.

Приложение №1

**СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ
ПРОГРАММ ДЛЯ ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2018616455

**Программная реализация гибридного алгоритма
компоновки фрагментов СБИС на основе композиции
генетической и муравьиной эвристик**

Правообладатель: *федеральное государственное автономное
образовательное учреждение высшего образования «Южный
федеральный университет» (Южный федеральный университет)
(RU)*

Авторы: *Запорожец Дмитрий Юрьевич (RU), Заруба Дарья
Викторовна (RU), Бова Виктория Викторовна (RU)*



Заявка № **2018613873**

Дата поступления **18 апреля 2018 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **01 июня 2018 г.**

*Руководитель Федеральной службы
по интеллектуальной собственности*

Г.П. Ивлиев

Г.П. Ивлиев

49/17

g. 56/14

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017663572

Программная реализация бактериального алгоритма
размещения разногабаритных элементов на плоскости

Правообладатель: *федеральное государственное автономное
образовательное учреждение высшего образования «Южный
федеральный университет» (Южный федеральный университет)*
(RU)

Авторы: *Запорожец Дмитрий Юрьевич (RU),
Заруба Дарья Викторовна (RU)*



Заявка № 2017660364

Дата поступления 16 октября 2017 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 07 декабря 2017 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

Приложение №2

АКТЫ О ВНЕДРЕНИЕ РЕЗУЛЬТАТОВ РАБОТЫ



НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЛАБОРАТОРИЯ
АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ, ОБЩЕСТВО
С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

пер. Биржевой Спуск, 8
г. Таганрог, 347900, Ростовская обл.,
Тел.: (8634) 47-70-40, 47-70-44
e-mail: info@reallab.ru,
http://www.reallab.ru

АКТ

Об использовании научных результатов диссертационной работы Заруба Д.В.
«Биоинспирированные методы и алгоритмы разбиения схем при автоматизированном
проектировании СБИС» на соискание ученой степени кандидата технических наук

Научные результаты, полученные в диссертационной работе Заруба Д.В., использовались в отделе
робототехники НИЛ АП.

Были использованы следующие результаты:

- модифицированный метод генетического поиска;
- модифицированный метод оптимизации на основе поведения роя светлячков.

Оба метода были использованы при разработке программы автоматической установки робота на
зарядную станцию. Модифицированный метод генетического поиска в сочетании с
разработанным в диссертации методом оптимизации позволил выбрать оптимальную траекторию
движения робота. Использование указанных выше результатов работы Заруба Д.В. позволило
повысить вероятность безошибочной установки робота в док.

Главный конструктор НИЛ АП,

доктор технических наук



29.03.2023

Денисенко В.В.



«УТВЕРЖДАЮ»
Директор ИКТИБ
Г.Е. Веселов
2023г.



АКТ

Об использовании научных результатов диссертационной работы Заруба Д.В.
«Биоинспирированные методы и алгоритмы разбиения схем при
автоматизированном проектировании СБИС» на соискание ученой степени
кандидата технических наук

Научные результаты, полученные в диссертационной работе Заруба Д.В., использовались в г/б 8.823.2014/К «Разработка теории и основных принципов эволюционных вычислений для поддержки принятия оптимальных решений при проектировании многоцелевых интеллектуальных систем» на выполнение проектной работы в части государственного задания МИНОБРНАУКИ России в сфере научной деятельности.

В частности, были использованы следующие результаты кандидатской диссертации Заруба Д.В.:

1. Принципы поиска оптимальных решений на основе биоинспирированных эвристик при проектировании интеллектуальных систем.
2. Комбинированный подход, основанный на биоинспирированных методах оптимизации, для принятия решений в конструкторском проектировании
3. Механизмы эволюционного моделирования и генетического поиска при принятии решений в задачах оптимизации при проектировании многоцелевых интеллектуальных систем.

Использование указанных теоретических и практических результатов, полученных в диссертационной работе Заруба Д.В., позволило сформулировать принципы построения системы для поддержки принятия решений для задач конструкторского проектирования.

Ответственный исполнитель
г/б 8.823.2014/К д.т.н., доцент



Ю.А. Кравченко

«УТВЕРЖДАЮ»

Директор ИКТИБ

Г.Е. Веселов

04 2023г.



АКТ

Об использовании научных результатов диссертационной работы Заруба Д.В.
«Биоинспирированные методы и алгоритмы разбиения схем при
автоматизированном проектировании СБИС» на соискание ученой степени
кандидата технических наук

Научные результаты, полученные в диссертационной работе Заруба Д.В., использовались при выполнении гранта РФФИ №13-01-00371 «Разработка теории и принципов построения интеллектуальных систем проектирования для решения задач разбиения СБИС на основе биоинспирированных методов».

В частности, были использованы следующие результаты кандидатской диссертации Заруба Д.В.:

1. Математическая модель схемы на основе гиперграфа.
2. Принципы и стратегии поиска оптимальных решений на основе методов биоинспирированной оптимизации, ориентированные на решение задачи разбиения СБИС.
3. Комбинированная архитектура поиска на основе биоинспирированных методов оптимизации.
4. Модифицированные биоинспирированные алгоритмы, ориентированные на решение задачи разбиения схем при проектировании СБИС.
5. Программная среда для решения задач разбиения СБИС на основе биоинспирированных методов.

Использование указанных теоретических и практических результатов, полученных в диссертационной работе Заруба Д.В., позволило разработать новые эффективные методы и алгоритмы, инспирированные природными системами, для построения интеллектуальных систем проектирования, ориентированных на решение задачи разбиения СБИС.

Ответственный исполнитель
РФФИ №13-01-00371 к.т.н., доцент

С.Н. Щеглов



АКТ

Об использовании научных результатов диссертационной работы Заруба Д.В.
«Биоинспирированные методы и алгоритмы разбиения схем при
автоматизированном проектировании СБИС» на соискание ученой степени
кандидата технических наук

Научные результаты, полученные в диссертационной работе Заруба Д.В., использовались при выполнении гранта РФФИ №19-01-00059 «Разработка биоэвристик для создания интеллектуальной подсистемы принятия эффективных решений NP-трудных и NP-сложных комбинаторно-логических задач на графах».

В частности, были использованы следующие результаты кандидатской диссертации Заруба Д.В.:

1. Биоинспирированные оптимизационные методы для решения NP-трудных и NP-сложных комбинаторно-логических задач.
2. Модифицированные алгоритмы биоинспирированной оптимизации, отличающиеся использованием новых механизмов поиска.
3. Разработаны модифицированные генетические операторы, ориентированные на решение задачи разбиения схем при автоматизированного проектировании СБИС, основанные на применении новых процедур.

Использование указанных теоретических и практических результатов, полученных в диссертационной работе Заруба Д.В., позволило сформулировать принципы построения программного комплекса для поддержки принятия решений NP-трудных и NP-сложных комбинаторно-логических задач на графах.

Ответственный исполнитель
РФФИ №19-01-00059 к.т.н., доцент

С.Н. Щеглов



«УТВЕРЖДАЮ»

Директор ИКТИБ

Г.Е. Веселов

04 2023г.

АКТ

Об использовании научных результатов диссертационной работы Заруба Д.В.
 «Биоинспирированные методы и алгоритмы разбиения схем при
 автоматизированном проектировании СБИС» на соискание ученой степени
 кандидата технических наук

Научные результаты, полученные в диссертационной работе Заруба Д.В., использовались при выполнении гранта РФФИ №16-01-00586 «Разработка инспирированных природой методов и принципов поиска оптимальных решений в задачах проектирования и управления».

В частности, были использованы следующие результаты кандидатской диссертации Заруба Д.В.:

1. Бактериальный и светлячковый методы биоинспирированной оптимизации, ориентированные на поиск оптимальных решений в задачах конструкторского проектирования.

2. Разработаны модифицированные бактериальный, светлячковый и генетический алгоритмы, позволяющие получать наборы квазиоптимальных решений.

3. Разработаны модифицированные генетические операторы и новые процедуры поиска оптимальных решений в задачах проектирования.

Использование указанных теоретических и практических результатов, полученных в диссертационной работе Заруба Д.В., позволило разработать новые эффективные методы и алгоритмы, инспирированные природными системами, для поиска оптимальных решений в задачах проектирования.

Ответственный исполнитель
 РФФИ №16-01-00586 к.т.н., доцент

С.Н. Щеглов



«УТВЕРЖДАЮ»

Директор ИКТИБ

Г.Е. Веселов

« 04 » 2023г.

АКТ

Об использовании результатов кандидатской диссертации Заруба Д.В.
«Биоинспирированные методы и алгоритмы разбиения схем при автоматизированном проектировании СБИС» в учебном процессе Института компьютерных технологий и информационной безопасности Южного федерального университета

Мы, ниже подписавшиеся, руководитель бакалаврского направления 09.03.02 «Информационные системы и технологии» к.т.н., профессор Нужнов Е.В., руководитель направления магистратуры 09.04.01 «Информатика и вычислительная техника» д.т.н., доцент Кравченко Ю.А., ответственный за направление аспирантуры 09.06.01 «Информатика и вычислительная техника» к.т.н., доцент Щеглов С.Н., зав. кафедры САПР, д.т.н. Курейчик В.В. составили акт в том, что в учебном процессе кафедры САПР Института компьютерных технологий и информационной безопасности используются следующие результаты, полученные в кандидатской диссертации Заруба Д.В.:

1. Унифицированный метод кодирования и декодирования данных при решении задачи разбиения схем при автоматизированном проектировании СБИС.
2. Комбинированная архитектура поиска оптимальных решений на основе биоинспирированных эвристик.
3. Модифицированные биоинспирированные методы оптимизации на основе поведения колонии бактерий и роя светлячков в живой природе.
4. Модифицированные алгоритмы биоинспирированного поиска.
5. Программный комплекс для решения задачи разбиения схем при проектировании СБИС.

Указанные результаты используются при проведении следующих курсов в ИКТИБ: «Интеллектуальные системы и технологии», «Роевые методы оптимизации», «Интеллектуальный анализ данных», «Перспективные методы и алгоритмы решения технических задач», «Эволюционное моделирование и генетические алгоритмы», «Машинное обучение и биоинспирированная оптимизация».

Внедрение в учебный процесс ряда теоретических и практических результатов диссертационной работы Заруба Д.В. позволило повысить качество подготовки бакалавров, магистров и аспирантов.

Руководитель направления 09.03.02

Е.В. Нужнов

Руководитель направления 09.04.01

Ю.А. Кравченко

Ответственный за направление 09.06.01

С.Н. Щеглов

Уч. секретарь кафедры САПР

В.И. Данильченко