

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

На правах рукописи



**Белых Михаил Алексеевич**

**ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ДЛЯ АДАПТИВНОЙ СИСТЕМЫ  
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ  
ПРИ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ  
ТРАНСПОРТНОЙ ЗАДАЧИ**

Специальность 2.3.1. Системный анализ, управление и обработка  
информации, статистика

**Диссертация**

на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук, профессор  
Барabanов Владимир Федорович

Воронеж – 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
ГЛАВА 1. СИСТЕМНЫЙ АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ .....	12
1.1 Общий анализ проблематики задач оптимизации.....	12
1.2 Задачи поиска оптимального маршрута как класс задач оптимизации .	14
1.2.1 Задача коммивояжера с временными ограничениями .....	15
1.2.2 Классическая и многокритериальная транспортные задачи .....	17
1.2.3 Многокритериальная оптимизация по Парето .....	20
1.3 Систематизация эволюционных алгоритмов для решения оптимизационных задач .....	24
1.3.1 Генетические алгоритмы в задачах оптимизации .....	25
1.3.2 Муравьиные алгоритмы как инструмент решения оптимизационных задач.....	27
1.3.3 Пчелиный алгоритм для решения задач поиска оптимального маршрута.....	32
1.3.4 Сравнительный анализ параметров эволюционных алгоритмов.....	34
1.4 Обзор современных средств решения задач оптимизации на основе эволюционных алгоритмов .....	35
1.5 Применение систем поддержки принятия решений с использованием адаптивных алгоритмов.....	38
1.6 Постановка задачи исследования.....	40
ГЛАВА 2. ФОРМАЛИЗАЦИЯ МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧИ ОПТИМИЗАЦИИ И МЕХАНИЗМЫ МОДИФИКАЦИЙ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ .....	42
2.1 Схемы работы эволюционных алгоритмов.....	42
2.2 Математическая постановка многокритериальной транспортной задачи с временными ограничениями .....	47

2.3	Механизмы модификации эволюционных алгоритмов .....	50
2.3.1	Модификации алгоритма муравьиной колонии .....	51
2.3.2	Модификации генетического алгоритма .....	52
2.3.3	Модификации алгоритма пчелиной колонии .....	55
2.4	Представление базы знаний адаптивной системы .....	56
2.5	Разработка механизмов переключения эволюционных алгоритмов .....	57
2.6	Выводы .....	60
ГЛАВА 3. СТРУКТУРА АДАПТИВНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ .....		62
3.1	Общая структура работы адаптивной системы поддержки принятия решений .....	62
3.2	Описание правил базы знаний для взаимодействия с данными адаптивной системы поддержки принятия решений .....	68
3.3	Структура работы модулей эволюционных алгоритмов на примере алгоритма муравьиной колонии .....	71
3.4	Формирование пользовательского интерфейса адаптивной системы .....	78
3.5	Выводы .....	82
ГЛАВА 4. РЕАЛИЗАЦИЯ АДАПТИВНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ .....		84
4.1	Апробация разработанной адаптивной системы поддержки принятия решения при многокритериальной оптимизации с использованием эволюционных алгоритмов .....	84
4.1.1	Многокритериальная транспортная задача без ограничений .....	86
4.1.2	Многокритериальная транспортная задача с маршрутными ограничениями .....	94
4.1.3	Многокритериальная транспортная задача с временными ограничениями .....	97
4.2	Обсуждение результатов .....	101

4.3 Перспективы развития адаптивной системы поддержки принятия решений.....	104
4.4 Выводы.....	104
ЗАКЛЮЧЕНИЕ .....	106
СПИСОК ЛИТЕРАТУРЫ.....	108
ПРИЛОЖЕНИЕ А Свидетельство о государственной регистрации программы для ЭВМ .....	123
ПРИЛОЖЕНИЕ Б Акты о внедрении результатов диссертационного исследования.....	124

## ВВЕДЕНИЕ

### **Актуальность темы исследования.**

Многокритериальная оптимизация является одной из ключевых задач в производстве, логистике, информационных технологиях и т. д. Решение задачи многокритериальной оптимизации позволяет обеспечить снижение расходов и издержек, определить оптимальный план действий или эффективнее распределить имеющиеся ресурсы, учитывая множество возможных факторов.

Для решения задач многокритериальной оптимизации нередко используются эволюционные алгоритмы, в число которых входят, например, генетические алгоритмы (ГА), алгоритм муравьиной колонии (АМК) и алгоритм пчелиной колонии (АПК). Данный тип алгоритмов применяется как в своих классических формулировках, так и с применением различных модификаций, направленных на улучшение эффективности поиска. Для решения каждой задачи оптимизации может быть пригодно несколько алгоритмов, однако для подбора наиболее оптимального алгоритма имеет место применение адаптивных систем поддержки принятия решений, способных предоставить рекомендации относительно наиболее подходящего для решения конкретной задачи алгоритма.

Также для адаптивных систем характерным и обязательным является наличие математического описания решаемой задачи. Это позволяет системе получить предельно четкие формулировки требований и ограничений, накладываемых на конкретную задачу, а также оценить приемлемость применения для ее решения конкретного алгоритма.

Для адаптивной системы поддержки принятия решений имеет место необходимость разработки механизмов проверки текущего решения и переключения активного алгоритма. В данном контексте под адаптацией понимается возможность системы настроить (или предложить) инструменты для решения задачи оптимизации, основываясь на результатах предыдущих

решений схожих по критериям задач, а также с возможностью внесения изменений в алгоритм (или его замены), который уже используется для проведения вычислений, основываясь на анализе получаемых решений в предыдущих итерациях. Механизм проверки текущего решения подразумевает отслеживание состояния решения в процессе работы алгоритма. Механизм переключения характеризуется возможностью замены активного алгоритма.

Таким образом, актуальность темы диссертационного исследования обозначается необходимостью разработки алгоритмического обеспечения для системы, способной получать оптимальные решения многокритериальных задач оптимизации, адаптироваться к изменениям результатов получаемых решений с целью их улучшения и обладающей механизмами проверки текущего решения и модификации алгоритмов в ходе выполнения вычислений.

Разработками и исследованиями в области эволюционных алгоритмов, задач оптимизации и адаптивных систем занимаются такие ученые, как Громов Ю.Ю., Золотарюк А.В., Кажанов А.А., Курейчик В.М., Ногин В.Д., Пересветов В.В., Подвальный С.Л., Подиновский В.В., Штовба С.Д., Ben Hamida S, Dorigo M., Hwang C.L., Karaboga D., Li J.-Y., Seif M.S., Wierzbicki A.P., Zhang Q. и др. Все перечисленные ученые отмечают сложность и важность поиска оптимального решения многокритериальных транспортных задач.

Вышесказанное определяет **практическую задачу** – повышение эффективности решения многокритериальных транспортных задач оптимизации за счет сокращения времени расчета (переключением активного алгоритма) путем создания адаптивной системы поддержки принятия решений с использованием эволюционных алгоритмов.

При этом необходимо рассмотреть **научную задачу**, заключающуюся в разработке специализированных моделей, механизмов модификации

эволюционных алгоритмов, структуры адаптивной системы при многокритериальной оптимизации транспортной задачи.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

**Объект исследования:** системы поддержки принятия решений при многокритериальной оптимизации транспортной задачи с использованием эволюционных алгоритмов.

**Предмет исследования:** эволюционные алгоритмы для адаптивных систем поддержки принятия решений при многокритериальной оптимизации транспортной задачи.

**Целью работы** является повышение эффективности адаптивной системы поддержки принятия решения при многокритериальной оптимизации транспортной задачи за счет модифицированных эволюционных алгоритмов.

**Задачи исследования.** Для достижения поставленной цели решались следующие задачи исследования.

1. Провести комплексный анализ существующих эволюционных алгоритмов, применяемых для решения многокритериальной транспортной задачи оптимизации.

2. Разработать специализированное математическое описание многокритериальной транспортной задачи оптимизации.

3. Оценить эффективность и предложить модификации эволюционных алгоритмов: генетического алгоритма, алгоритма муравьиной колонии и алгоритма пчелиной колонии. Разработать механизмы проверки текущего решения.

4. Разработать структуру адаптивной системы поддержки принятия решений, основанной на модифицированных эволюционных алгоритмах.

5. Провести апробацию адаптивной системы поддержки принятия решений многокритериальной оптимизации транспортной задачи.

**Методы исследования.** Для решения поставленных задач в ходе работы над диссертационным исследованием использовались методы теории системного анализа, принятия решения, оптимизации, эволюционных алгоритмов, адаптивных систем, специального математического обеспечения.

**Тематика исследований** соответствует следующим пунктам паспорта специальности 2.3.1 «Системный анализ, управление и обработка информации, статистика»: п. 5. «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта»; п. 9. «Разработка проблемно-ориентированных систем управления, принятия решений и оптимизации технических объектов».

**Научная новизна.** К результатам работы, отличающимся научной новизной, относятся следующие материалы:

1. Специализированное математическое описание многокритериальной транспортной задачи оптимизации с временными ограничениями, позволяющее применять адаптивный механизм поиска решения и отличающееся использованием модифицированных эволюционных алгоритмов.

2. Механизм модификации эволюционных алгоритмов (генетический, муравьиной колонии, пчелиной колонии), позволяющий ускорить процесс решения многокритериальных задач оптимизации (посредством варьирования правил скрещивания и мутаций, отбора, формирования новых популяций, изменения феромона и т.д.) и отличающийся учетом результатов предыдущих решений.

3. Адаптивный механизм проверки текущего решения, отличающийся учетом скорости на каждой итерации результатов для последующей



обработки и внесения оперативных корректировок в системе поддержки принятия решений.

4. Структура адаптивной системы поддержки принятия решений при многокритериальной оптимизации транспортной задачи, позволяющей производить ускоренный поиск и оптимизацию и отличающейся использованием базы знаний, содержащей данные о полученных ранее решениях.

**Теоретическая значимость исследования** заключается в развитии методов системного анализа для построения адаптивной системы поддержки принятия решений на базе эволюционных алгоритмов, разработке модификаций эволюционных алгоритмов с целью улучшения их вычислительных возможностей и формировании механизмов анализа решения и подбора эффективного алгоритма для решения транспортной задачи оптимизации.

**Практическая значимость исследования** заключается в реализации специализированного программного обеспечения в виде адаптивной системы поддержки принятия решений, направленной на эффективное решение многокритериальных транспортных задач оптимизации.

**Положения, выносимые на защиту.**

1. Специализированное математическое описание многокритериальной задачи оптимизации (транспортной задачи поиска оптимального маршрута) предполагает использование модифицированных эволюционных алгоритмов.

2. Механизм модификации эволюционных алгоритмов (генетический, муравьиный и пчелиный) для решения многокритериальной транспортной задачи поиска оптимального маршрута обеспечивает большую эффективность и скорость проведения расчетов.

3. Адаптивный механизм проверки текущего решения позволяет отслеживать скорость получаемых в процессе работы результатов и влиять на выбор алгоритма, модифицируя или заменяя его на другой.

4. Структура адаптивной системы поддержки принятия решений при поиске оптимального маршрута базируется на модифицированных эволюционных алгоритмах и обеспечивает адаптивное переключение алгоритмов при расчете, основываясь на анализе предыдущих решений.

**Реализация и внедрение результатов работы.** В рамках диссертационной работы был реализован программный компонент «Адаптивная система поддержки принятия решений». Разработанные материалы внедрены в образовательный и научно-исследовательский процесс кафедры автоматизированных и вычислительных систем Воронежского государственного технического университета, а также в компаниях «Девелоперс», «АЙТИ Комфорт», «КИИНАЙ».

**Апробация работы.** Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: XIV Международной научно-практической конференции «Антропоцентрические науки: Инновационный взгляд на образование и развитие личности» (Воронеж, Воронежский государственный технический университет, 19-20 апреля 2021), на 61-ой, 62-ой, 64-ой научно-технических конференциях профессорско-преподавательского состава, сотрудников, аспирантов и студентов в рамках Дней науки ВГТУ (Воронеж, Воронежский государственный технический университет, 2021, 2022, 2024 гг.), на Международной научно-практической конференции «НаБиТэм-2024» (Липецк, Липецкий государственный технический университет, 27-28 февраля 2024).

Достоверность результатов обусловлена корректным использованием теоретических методов исследования и подтверждена результатами сравнительного анализа данных вычислительных экспериментов.

**Публикации.** По результатам диссертационного исследования опубликовано 12 научных работ (1 – без соавторов), в том числе 4 – в изданиях, рекомендованных ВАК РФ, а также 1 свидетельство о регистрации программы для ЭВМ. В работах, опубликованных в соавторстве и

приведенных в конце реферата, лично автором получены следующие результаты: [5, 32, 33] – структура интеллектуальной системы поддержки эволюционных алгоритмов, описание подсистем; [65] – алгоритмическое обеспечение эволюционных алгоритмов; [25, 28, 29, 35] – модификации муравьиного алгоритма и формализация задач оптимизации; [38] – формализация задачи оптимизации на примере мониторинга состояний; [34] – математическая формулировка эволюционных алгоритмов.

**Структура и объем работы.** Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 125 наименований, приложений. Основная часть изложена на 128 страницах с 42 рисунками и 6 таблицами.

# ГЛАВА 1. СИСТЕМНЫЙ АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

## 1.1 Общий анализ проблематики задач оптимизации

Оптимизация в широком смысле представляет собой задачу достижения наибольшей эффективности процесса в данных конкретных условиях, то есть получение заданного объема продукции или выполненных работ с наименьшими затратами либо получение наибольшего объема продукции или выполненных работ при данных конкретных условиях и затратах [24, 105].

В зависимости от конкретной постановки цели оптимизации и параметров принятия оптимального решения, определяющих область действия оптимизации, задачи можно разделить на задачи глобального и локального типа, а также дополнительно стратегического и тактического.

Задачи глобального типа направлены на обеспечение достижения какой-либо всеобщей, единой для широкого спектра областей цели по некоторому единому параметру. Задачи стратегического типа направлены на достижения наибольшей эффективности в какой-то отдельной области. Стратегический тип задач имеет широкое распространение, поскольку он определяет общую направленность деятельности каждой отрасли, давая возможность правильно решать вопросы выбора и материальной базы, организации и технологии производственных процессов. Задачи тактического типа позволяют решать более узкие по целям задачи, охватывающие ограниченную деятельность отрасли и/или производства. Локальные задачи преследуют цель обеспечить наибольшую эффективность для какого-то конкретного процесса и/или операции [24].

Любая оптимизация из описанных выше, так или иначе, приводит к необходимости математического описания и формулировке задачи

оптимизации. В свою очередь, всякая математическая задача оптимизации сводится к следующей концепции: нахождение экстремума функции (выражающегося в ее максимуме или минимуме) в некоторой области конечного векторного пространства, области которого сформированы набором равенств и неравенств [48, 56, 94]. Целевая функция (ЦФ) в задачах оптимизации представляет собой набор критериев качества, которые должны быть оптимизированы одновременно.

В общем виде ЦФ представляет собой набор переменных, как настраиваемых, так и формирующихся самостоятельно под воздействием собственных условий и обстоятельств. Постановка задачи оптимизации требует осмысления оптимизируемого объекта или системы, а также цели оптимизации. При этом необходимо, чтобы в каждой задаче определялось экстремальное значение только лишь одного свойства объекта, поскольку в большинстве случаев оптимизируемые свойства находятся в смысловом противоречии друг с другом. Свойство объекта или системы, определяющее цель оптимизации, обозначается как критерий оптимизации [106].

В общем виде задача оптимизации может быть формализована следующим образом [48]: среди элементов  $x$ , образующих множество  $X$ , найти такой элемент  $x^*$ , который доставляет максимальное/минимальное значение  $f(x^*)$ , заданное функцией  $f(x)$ .

Для того чтобы корректно сформулировать задачу оптимизации необходимо обозначить:

– допустимое множество

$$(X = \{x = (x_1, x_2, \dots, x_n) | g_i(x) \leq 0, i = 1, \dots, m\} \subset R^n);$$

– ЦФ как отображение  $(f(x), X), X \rightarrow R$ ;

– критерий поиска *min/max*.

Решением задачи  $f(x) \rightarrow \min$  либо  $f(x) \rightarrow \max$  является выполнение одного из следующих действий:

– установление того, что  $X = \emptyset$ ;

- установление того, что ЦФ  $f(x)$  не ограничена снизу/сверху;
- нахождение такого  $x^* \in X$ , что  $f(x^*) \rightarrow \min f(x)$  либо  $f(x^*) \rightarrow \max f(x)$ .

Для функций, не являющихся выпуклыми, при поиске оптимума ограничиваются поиском локальных значений, т. е. таких  $x^*$ , в окрестностях которых  $f(x) \geq f(x^*)$  для минимума или  $f(x) \leq f(x^*)$  для максимума. В том случае, если допустимое множество  $X \in R^n$ , то речь идет о задаче безусловной оптимизации; в отношении условной оптимизации допустимое множество характеризуется как  $\{x = (x_1, x_2, \dots, x_n) | g_i(x) \leq 0, i = 1, \dots, m\}$ .

Задача оптимизации в широком смысле рассматривается при обсуждении проблем в различных сферах, что подчеркивает ее актуальность и на сегодняшний день. Примером этого могут служить эти [12, 15, 19, 43, 49, 77, 99, 100, 125] и другие работы, так или иначе затрагивающие задачи оптимизации.

Типовое формирование задач оптимизации приводит к разнообразию классов, от которых зависит подбор эффективных методов решения. Для классификации задачи служит формируемая ЦФ задачи, допустимая область поиска, задаваемая системой равенств и неравенств или более сложными алгоритмами.

## **1.2 Задачи поиска оптимального маршрута как класс задач оптимизации**

Классическая задача маршрутизации транспортных средств заключается в поиске набора маршрутов автотранспорта с минимальной суммарной длиной с учетом того, что все клиенты должны быть обслужены. Наряду с данной задачей (Heterogeneous Fixed Fleet Vehicle Routing Problem, сокр. HFFVRP) также часто рассматривают ситуацию с безлимитным по

численности автопарком, гибридный (смешанный) вариант и вариант с идентичными транспортными средствами.

Задача маршрутизации транспортных средств в широком смысле может быть решена различными способами, а исследования и разработки в области формирования методов и алгоритмов решения ведутся на регулярной основе [27].

Все эти задачи являются NP-трудными [78], что означает экспоненциальный рост сложности. Как следствие, для их решения требуются алгоритмы, способные предложить наиболее приближенные к оптимальным из полученных решений.

### 1.2.1 Задача коммивояжера с временными ограничениями

Задача коммивояжера (Traveling Salesman Problem, сокр. TSP) является одной из вариаций оптимизационной задачи. TSP возникает в обширном классе таких приложений, например, как распознавание траекторий, образов, построение оптимальных схем движения и т. д. Сама по себе она представляет задачу отыскания кратчайшего гамильтонова пути в полном конечном графе с  $N$  вершинами. Все известные методы нахождения точного решения включают в себя поиск пространства решений, которое увеличивается экспоненциально в зависимости от  $N$  [58].

Задача коммивояжера с временными ограничениями (Traveling Salesman Problem with Time Windows, сокр. TSPTW), являющаяся более сложной вариацией классической, формулируется следующим образом [84, 85].

Имеется конечный граф вида  $G = (V, A)$ , где  $V = N \cup \{o, d\}$ ;  $N = \overline{1, n}$  – множество вершин  $n$ ,  $A = N \times N$  – множество ребер,  $o$  – начальная точка пути,  $d$  – конечная точка в пути.

Зафиксируем  $D_{ij} = \langle y_i, y_j \rangle$ ,  $D_{ij} \in A$  – длина ребра, соединяющего вершины  $y_i, y_j \in N$ , причем согласно следующей системе:

$$x_{ij} = \begin{cases} 1, & \text{если } D_{ij} \text{ входит в оптимальный маршрут;} \\ 0, & \text{в обратном случае.} \end{cases}$$

Каждой вершине  $x_i$  соответствует временной интервал  $[a_i, b_i]$ , где  $a_i$  – начало временного интервала,  $b_i$  – его окончание. Соответственно,  $a_o$  – время выезда автотранспорта из начальной вершины  $o$ ,  $b_d$  – время прибытия в конечную вершину  $d$ .

Учитывая, что допускается ожидание начала временного интервала, переменная  $t_i$  определяет время прибытия в вершину  $y_i \in N \cup \{o\}$ , а переменная  $t_d$  – время прибытия в конечную вершину  $d$ ,  $t_o$  соответственно принимается за время начала маршрута.

Также отметим, что каждому ребру графа  $D_{ij} \in A$ , соответствует стоимость  $c_{i,j}$ , принимаемая как  $c_{i,j} = l_1 T_{ij} + l_2 U_{ij} + l_3 D_{ij}$ , где:

–  $T_{ij} = t_j - (t_i + s_i)$  – разница во времени между концом обслуживания вершины  $x_i$  и началом обслуживания вершины  $x_j$ ;

–  $U_{ij} = b_j - (t_i + s_i + D_{ij})$  – актуальность доставки в вершину  $j$ , что подразумевает собой время, оставшееся до последней возможности обслужить вершину  $j$ ;

–  $l_1 + l_2 + l_3 = 1$ , ( $l_1 \geq 0, l_2 \geq 0, l_3 \geq 0$ ) – коэффициенты, позволяющие влиять на стоимость, исключая или дополняя параметры.

Также для каждого ребра  $D_{ij} \in A$  должно выполняться следующее условие:

$$(t'_i + s_i + D_{ij}) \leq b_j, \quad t'_i = \max\{a_i, t_i\} \quad (i \neq j).$$

Задача математического программирования должна содержать два типа переменных: множество вершин двоичной матрицы переходов  $X = \{x_i\}$  ( $x_i \in N$ ) и множество переменных времени  $T = \{t_i\}$ .

Минимальный по стоимости маршрут начинается в интервале времени  $[a_o, b_o]$  из начальной вершины  $o$ , в дальнейшем проходя через все вершины



из множества  $N$  только один раз в течение соответствующих временных окон, и заканчивается в конечной вершине  $b_d$ .

Математически ЦФ данной задачи может быть сформулирована в виде

$$\min \sum_{i,j \in A} c_{ij} x_{ij}.$$

Решение TSPTW заключается в построении такого маршрута движения транспортного средства, чтобы результат не только являлся наиболее приближенным к оптимальному с позиции основного требования (например, посещения всех складов или магазинов), но и укладывался в рамки приложенных к задаче ограничений.

Задача коммивояжера имеет различные решения и не теряет своей актуальности и сегодня, о чем говорят систематические публикации на эту тему [64, 67, 83, 84].

### 1.2.2 Классическая и многокритериальная транспортные задачи

Более широкой и сложной в сравнении с задачей коммивояжера является транспортная задача (она же – задача Монжа-Канторовича [13, 18, 120]), которая традиционно решается на предприятиях логистами. Вопросы логистики рассматривают такие аспекты, как снижение расходов на транспортировку, выбор кратчайшего маршрута перевозок, сокращение затрат времени, упрощение сложной схемы доставки продукции, уменьшение разного рода расходов [55, 91, 112] и пр. Рассмотрению данного рода задач оптимизации посвящена, например, монография [57], в которой рассматриваются как сама задача, так и методы ее решения.

Классическая транспортная задача является задачей линейного программирования и математической моделью многих задач, встречающихся в различных областях [107]. Транспортные задачи разделяют на закрытые (сумма запасов товаров у поставщиков равносильна сумма потребностей потребителей в товарах) и открытые (баланс между значениями не

соблюдается, т. е. запрос превышает предложение либо предложение превышает запрос).

Исходные данные транспортной задачи могут быть сформированы в виде таблицы (таблица 1.1), где имеются  $m$  пунктов производства однородных товаров  $A_i$  ( $i = 1, \dots, m$ ) и  $n$  пунктов потребления однородных товаров  $B_j$  ( $j = 1, \dots, n$ ), а также заданы объемы производства  $a_i$  и величина спроса  $b_j$  в одних и тех же единицах измерения.

Таблица 1.1 – Исходные данные классической транспортной задачи

		Пункты потребления					
		$B_1$	$B_2$	...	$B_n$		
Пункты производства	$A_1$	$c_{11}$	$c_{12}$	...	$c_{1n}$	$a_1$	Объем производства
	$A_2$	$c_{21}$	$c_{22}$	...	$c_{2n}$	$a_2$	
	...	...	...	...	...	...	
	$A_m$	$c_{m1}$	$c_{m2}$	...	$c_{mn}$	$a_m$	
		$b_1$	$b_2$	...	$b_n$	Объем потребления	

Допустимость такой задачи характеризуется выполнением условия совпадения объемов продукции в пунктах производства и пунктах потребления  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ . В этом случае расходы на перевозку  $c_{ij}$  представляются в виде следующей матрицы:

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{pmatrix}. \quad (1.1)$$

Требуется обеспечить перевозку всей продукции из пунктов производства в пункты потребления таким образом, чтобы расходы на перевозку были минимальными, т. е. план перевозки был наиболее экономным.

Искомое количество продукции определяется как  $x_{ij} \geq 0$  и представляется в виде следующей матрицы:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}. \quad (1.2)$$

Данная задача характеризуется как задача об оптимальном плане перевозок некоторого типового продукта из  $m$  типовых пунктов хранения  $A_i$  в объемах  $a_i$  в  $n$  типовых пунктов размещения  $B_j$  в объемах  $b_j$  на типовых транспортных средствах (ТС) фиксированного количества со статичными данными [81, 92, 107]

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}, \quad (1.3)$$

где  $x_{ij}$  понимается как объем перевозки от  $i$ -го пункта хранения в  $j$ -ый пункт размещения,  $c_{ij}$  – коэффициент затрат на перевозки и  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ .

Замкнутость транспортной модели подразумевает перевозку всей продукции, имеющейся в пунктах производства и полное удовлетворение спроса на продукцию в пунктах потребления. Математически эти условия выглядят следующим образом:

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, & \sum_{i=1}^m x_{ij} &= b_j, \\ x_{ij} &\geq 0 \quad (i = 1, \dots, m; j = 1, \dots, n). \end{aligned} \quad (1.4)$$

Допустимость такой задачи обеспечивается условием

$$\sum_{j=1}^n b_j = \sum_{i=1}^m a_i, \quad (1.5)$$

которое обозначает, что объём продукции в пунктах производства должен совпадать с объемом продукции, требуемым в пунктах потребления.

Транспортная задача имеет различные модификации, одной из которых является ее многокритериальная версия [69, 75, 79, 103], оформление которой можно представить на примере задачи с двумя целевыми функциями.

Рассмотрим матрицу  $H$  размерностью  $m \times n$ , в которой каждый элемент  $h_{ij}$  характеризуется степенью важности перевозки продукта из  $i$ -го пункта производства в  $j$ -ый пункт потребления. В соответствии с этим, экспертами может быть сформирована степень важности перевозки для сочетания  $(i, j)$ , а именно

$$H = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{pmatrix}. \quad (1.6)$$

Отсюда ЦФ, обозначающая максимизацию степени важности перевозок, формируется как

$$\max \sum_{i=1}^m \sum_{j=1}^n h_{ij} x_{ij}. \quad (1.7)$$

Данная транспортная задача является примером многокритериальной транспортной задачи (МТЗ), так как включает в себя минимизации суммарных затрат на перевозку и максимизации степени важности перевозок одновременно [79].

Задача многокритериальной оптимизации заключается в существовании альтернативных математических вариантов, представляющих область допустимых решений, а также целевые функции, которые должны быть минимизированы (максимизированы) в данной области [68]. В математической формулировке она имеет следующее представление:

$$F = \min(\max)\{F_1, F_2, \dots, F_n\} \quad (n \geq 2). \quad (1.8)$$

Также стоит отметить, что задачу минимизации или максимизации можно преобразовать в обратную ей задачу (максимизации или минимизации соответственно) путем операции домножения ЦФ на  $-1$ .

### 1.2.3 Многокритериальная оптимизация по Парето

Решение задачи многокритериальной (или многоцелевой) оптимизации иногда понимается как аппроксимация либо вычисление всех или репрезентативного набора оптимальных решений по Парето.

Оптимизацией по Парето называется область многокритериального принятия решений, которая связана с задачами математической оптимизации с двумя и более функциями, подлежащими одновременной оптимизации [100].

Оптимальность по Парето определяется следующим образом. Точка  $x^* \in X$  называется оптимальной по Парето (также парето-оптимальной или эффективной), если  $\nexists x \in X$ , для которой выполнено неравенство  $f(x) \geq f(x^*)$ . В таком случае вектор  $f(x^*)$  именуют оптимальным (или эффективным) по Парето.

В многоцелевой оптимизации обычно не существует приемлемого решения, которое минимизировало бы все целевые функции одновременно, вследствие чего оптимальности по Парето уделяется существенное внимание. В случае однокритериальной задачи оптимальная по Парето точка становится точкой максимума ЦФ. Из этого следует, что определение парето-оптимальной точки представляет собой обобщение точки максимума скалярной числовой функции для варианта с векторным критерием.

В связи с тем, что  $f(x) \geq f(x^*)$  является ложным только в том случае, если истинно соотношение  $f(x) \not\geq f(x^*)$ , то верна и другая формулировка: оптимальной по Парето (парето-оптимальной) точкой  $x^* \in X$  является та, которая для  $\forall x \in X$  удовлетворяет соотношению  $f(x^*) \not\geq f(x)$ .

Один из методов оптимизации по Парето – лексикографический метод (lexicographic method) – подразумевает, что все критерии многокритериальной задачи должны быть распределены в порядке убывания степени важности – идея последовательных уступок или масштабирования [98, 100, 101]. Другой метод – скаляризации (scalarizing) – основывается на формулировке задачи однокритериальной оптимизации, оптимальные

решения которой являются парето-оптимальными решениями многокритериальной задачи; т. е. критерии оптимальности сводятся в одну целевую функцию задачи оптимизации [11, 41].

Рассмотрим подробнее метод последовательных уступок. На начальном этапе решается однокритериальная задача оптимизации по первому критерию:

$$y_1^* = \max_{X \in D} f_1(X); \quad (1.9)$$

После этого вводится определенная на основании каких-либо условий или заключений уступка  $\Delta y_1$ . С учетом уступки формируется и решается однокритериальная задача оптимизации по второму критерию:

$$y_2^* = \max_{\substack{X \in D \\ f_1(X) \geq y_1^* - \Delta y_1}} f_2(X). \quad (1.10)$$

Предоставление уступок по каждому критерию и решения однокритериальных задач продолжается до тех пор, пока не будет достигнут последний этап. Данный метод именуется методом  $\varepsilon$ -ограничений ( $\varepsilon$ -constrain method), который в общем виде может быть представлен как

$$\min_{x \in X} (\max_{f_i(x) \leq (\geq) \varepsilon_i, i \in \{\overline{1, k}\} \setminus \{j\}} f_j(x)). \quad (1.11)$$

Более простым является подход линейной скаляризации (linear scalarization), который заключается в объединении различных функций оптимизации в одну функцию. Однако этот метод позволяет находить только поддерживаемые решения проблемы (т. е. точки на выпуклой оболочке набора целей), в противном случае не все эффективные решения могут быть найдены [16]. На рисунке 1.1 представлено несколько итераций поиска эффективного решения при невыпуклом наборе результатов данным подходом. В данном примере минимизируется функция

$$\min F(x) = \omega_1 f_1(x) + \omega_2 f_2(x),$$

где  $f_1(x) = x_1$ ,  $f_2(x) = 1 + x_2^2 - x_1 - a \sin(b\pi x_1)$ , такая, что  $x_1 = [0; 1]$ ,  $x_2 = [-2; 2]$ . Для получения невыпуклого набора результатов, параметры  $a$  и

$b$  принимают значения  $a = 0.1$ ,  $b = 3$ . При этом  $\omega_1 + \omega_2 = 1$ ,  $\omega_1 \in [0,1]$ ,  $\omega_2 = 1 - \omega_1$ .

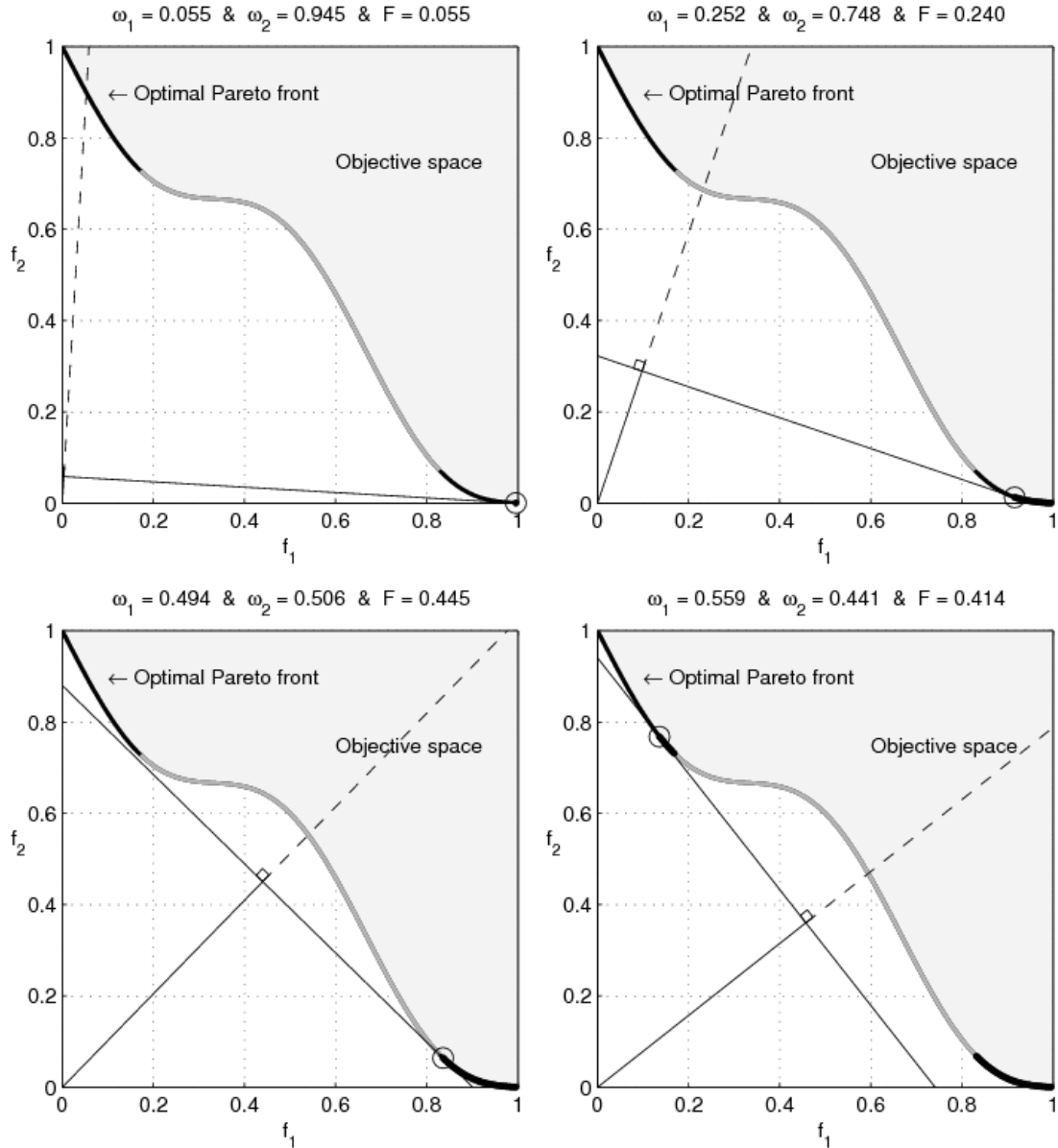


Рисунок 1.1 – Поиск эффективного решения при невыпуклом наборе результатов методом линейной скаляризации

Более сложными и продвинутыми примерами скаляризации являются достижение скаляризирующих задач Вержбицкого [23], многоцелевое программирование Сена [7] и гипербъемная (или Чебышевская) скаляризация [10].

Парето-оптимизация применяется во многих сферах, где присутствуют многокритериальные задачи оптимизации. К таковым относятся оптимизация процессов и управления, финансы, экономика, дизайн, распределение ресурсов и пр. [31].

### **1.3 Систематизация эволюционных алгоритмов для решения оптимизационных задач**

Эволюционные алгоритмы (ЭА) являются одним из эффективных и перспективных инструментов, применяемых для решения задач оптимизации в различных сферах деятельности, например, при решении задач глобальной оптимизации [47, 72], расчетах трудоемкости работ [61] и размеров памяти многоуровневой системы [89], исследовании и обработке облаков точек [36, 37], решении задачи коммивояжера [117] или транспортной задачи.

Эффективность представителей этого класса алгоритмов зависит от поставленной перед ними задачи. Так, один будет успешен при работе с задачами с небольшим числом параметров, но оказывается неработоспособным при обработке больших массивов данных. Другой, наоборот, превосходно работает с задачами, основанными на необходимости многокритериальной оптимизации, но не способен выдать удовлетворительного решения для менее сложных задач.

Отдельным вопросом является время поиска решения: в некоторых случаях он может не просто затянуться на часы, а уйти в бесконечный поиск и перебор решений. И даже в этом случае ЭА не дает гарантии, что по итогу будет найдено оптимальное или квазиоптимальное решение.

Чтобы не допускать такого исхода, необходимо устанавливать ограничения, в рамках которых алгоритм будет вынужден предоставить решение или набор решений, вписывающихся в эти ограничения.

Также одной из основных проблем, с которой сталкиваются разработчики ЭА, является проблема обеспечения баланса между скоростью



сходимости алгоритма и диверсификацией поиска. Быстрая сходимость алгоритма означает снижение разнообразия популяции. Напротив, диверсификация поиска расширяет пространство поиска и увеличивает вероятность локализации глобального экстремума задачи. Эти проблемы являются фундаментальными ввиду их теоретической и практической важности [111].

Несмотря на это, ЭА широко используются в решении задач оптимизации [47, 55, 61, 72, 90, 92, 112, 117]. В частности, в качестве такого инструмента применяются генетические алгоритмы, алгоритм муравьиной колонии и алгоритм пчелиной колонии.

### 1.3.1 Генетические алгоритмы в задачах оптимизации

Генетический алгоритм (Genetic Algorithm, ГА) представляет собой эволюционный алгоритм, основанный на эволюционном признаке выживания наиболее приспособленных особей. Данный алгоритм применяет методы, аналогичные природному естественному отбору, такие как скрещивание, мутацию и кроссинговер. Наиболее приспособленные особи обладают повышенными шансами на выживание и продолжение рода, и, как следствие, потомство в среднем будет более приспособленным, нежели предыдущее поколение [80, 108].

Основные понятия, используемые в ГА, следующие:

- особь, определяемая как искомый оптимальный путь  $S_{\text{ОПТ}}$ ;
- популяция, характеризующаяся как набор всех маршрутов  $f(x) = F(S_1, S_2, \dots, S_k)$ :
- хромосома, представляющая собой любой маршрут  $S_k$ ;
- гены, сформированные как точки некоторого графа  $G$  с  $N$  узлами и  $N(N - 1)$  ребрами от  $A$  до  $k$ , которые необходимо пройти.

Реализация ГА начинается с создания первичной маршрутной матрицы, представляющей собой начальную популяцию. Далее путем перемешивания генов генерируются  $n$  маршрутов, производится расчет приспособленности (длина маршрута) каждой хромосомы. Следом определяются два родителя (маршрута) и проводится процесс размножения, при котором сперва генерируется точка разрыва, с которой начинается процесс скрещивания, а затем формируются первый и второй потомки [121].

Процесс расчетов методом генетического алгоритма содержит в себе несколько основных этапов:

– Создание новой популяции. На данном этапе осуществляется создание начальной популяции, которая может не содержать в себе решение. Основным критерием формирования является предельно возможное разнообразие особей.

– Отбор. На этом этапе производится выборка наиболее подходящей популяции для их дальнейшего скрещивания, при этом особи, содержащие одинаковый набор генов, не учитываются.

– Скрещивание. На данном этапе осуществляется скрещивание двух особей, в первую очередь – наилучших. Результатом являются также две особи, состоящие из компонентов от объектов скрещивания. Целью данной операции является распространение благоприятных генов по всей популяции и сведение ее плотности к тем участкам, где она уже велика.

– Мутация. На этом этапе выполняется изменение произвольного количества элементов в особи на другие случайные.

Не считая первого, все рассмотренные этапы повторяются до тех пор, пока не будет выполнено условие нахождения наиболее оптимального решения либо подойдет к концу заявленное время на выполнение или число обращений к ЦФ.

В списке достоинств ГА числятся эффективное распараллеливание процессов и значительное количество свободных параметров, которые дают

возможность эффективного встраивания дополнительных эвристик. В то же время данный алгоритм уступает в скорости работы другим алгоритмам в отношении простых ЦФ, а также нельзя однозначно утверждать, что сходимость в том или ином случае вообще будет достигнута.

Различные модификации ГА применяются для решения не только задач оптимизации [25, 95, 110]. Данный алгоритм может применяться, например, при решении задачи о распределении веб-источников по группам при извлечении из них ЦФ [47] или расчете кинетических параметров с минимизацией функционального отклонения расчетных значений концентраций компонентов от экспериментальных данных [22]. Также можно привести в пример работы [59, 60], в которых рассматривается вопрос применения ГА для решения задачи размещения компонентов сверхбольших интегральных схем.

На сегодняшний день существует множество различных видов генетических алгоритмов. Каждая конкретная модификация определяется в зависимости от решаемой задачи и представляет собой набор генетических операторов с определенными параметрами. Для каждой задачи может быть составлен собственный уникальный ГА, а могут применяться и наиболее распространенные варианты алгоритма.

### **1.3.2 Муравьиные алгоритмы как инструмент решения оптимизационных задач**

Алгоритм муравьиной колонии (Ant Colony Algorithm, муравьиный алгоритм, АМК) является одним из наиболее эффективных методов нахождения решений задачи коммивояжера: это первая комбинаторная задача, решенная данным методом.

Он представляет собой имитацию организации поведения муравьиной колонии. Колонией является мультиагентная система, в которой каждый отдельный представитель данной системы действует самостоятельно по

некоторым определенным правилам. Алгоритм работы представляет вероятностную эвристику, в которой вероятности устанавливаются в соответствии с информацией о качестве решения на основе предыдущих решений [96, 125].

АМК можно описать следующей последовательностью действий [70]:

1. Создание муравьев. Методика расположения муравьев является основополагающей и привязана к условиям задачи, в зависимости от которой муравьи колонии могут располагаться как в одном месте, так и в разных. Также в момент создания муравьев необходимо указать первоначальный уровень феромона, который характеризуется некоторым небольшим положительным числом. Это необходимо для того, чтобы обеспечить отличную от нуля вероятность перехода в следующую точку на первом шаге.

2. Поиск решения. Сам маршрут представляет собой совокупность вершин графа, соединенных тропами – ребрами. Вероятность перехода муравья из вершины  $i$  в вершину  $j$  определяется по следующей формуле

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{h \notin tabu_k} \tau_{ij}^{\alpha} \eta_{ij}^{\beta}}, & \text{если } j \notin tabu_k, \\ 0, & \text{в противном случае,} \end{cases} \quad (1.8)$$

где:

- $\tau_{ij}$  – количество феромона на ребре  $(i, j)$ , «обоняние» муравья;
- $\eta_{ij}$  – привлекательность ребра  $(i, j)$ ,  $\eta_{ij} = 1/d_{ij}$ , где  $d_{ij}$  обозначает расстояние между вершинами  $i$  и  $j$ , «зрение» муравья;
- $\alpha, \beta$  – регулируемые параметры, определяющие важность составляющих (вес ребра и уровень феромонов) при выборе пути;
- $tabu_k$  – список уже посещенных вершин, «память» муравья.

3. Обновление феромонов. Когда все муравьи завершили свой путь, количество феромона должно измениться. Данный процесс включает два этапа:

- уменьшение значения феромона на всех дугах на определенную фиксированную величину;
- увеличение уровня феромона на тех ребрах, которые посещали муравьи.

Имитация испарения феромона производится по формуле

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad (1.9)$$

где  $\rho$  обозначает параметр, который отвечает за контроль скорости испарения феромона.

Данный параметр  $\rho$  дает возможность нивелировать ситуацию с безостановочным накоплением феромонов на ребрах пути. Это, в свою очередь, может привести к тому, что алгоритм не будет «забывать» неудовлетворительные решения, полученные на предыдущих этапах. В том случае, если ребро не было выбрано муравьями, то связанный с ним уровень феромона с каждой последующей итерацией будет экспоненциально уменьшаться. После окончательного испарения все муравьи изменяют уровень феромона на ребрах, которые они посетили.

Для ребра  $(i, j)$  количество феромона, откладываемого на нем, задается в следующем виде:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad (1.10)$$

где  $\Delta\tau_{ij}^k$  обозначает количество феромона, оставляемого  $k$ -ым муравьем на ребре, которое он посетил, и рассчитывается по следующей формуле:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L, & \text{если ребро } (i, j) \in L, \\ 0, & \text{в противном случае,} \end{cases} \quad (1.11)$$

где  $Q$  – константа, искусственно добавляющая феромон,  $L$  – общая длина пройденного пути.

Согласно формуле (1.11), чем лучше является путь, тем большее количество феромона будет располагаться на ребрах этого пути. В общем случае те ребра, по которым передвигается большее количество муравьев и

которые входят в состав наиболее короткого пути, получают больше феромонов. По этой причине такие ребра чаще всего будут выбираться муравьями в последующих итерациях.

Данный итерационный процесс будет продолжаться до тех пор, пока не произойдет выполнение одного из определенных условий завершения:

- произведено заданное количество итераций;
- все заданное количество муравьев завершило поиск;
- получено необходимое или наиболее близкое к необходимому качеству решение;
- закончилось отведенное для расчетов время.

Настроечными параметрами для АМК являются коэффициент  $\alpha$ , определяющий, насколько сильно количество феромона оказывает влияние на выбор муравья, и коэффициент  $\beta$ , определяющий, насколько сильно муравей при принятии решения будет ориентироваться на близость следующей вершины, а также число муравьев в колонии [121].

Модификацией данного алгоритма, которая предназначена для повышения эффективности решения задач маршрутизации, является система муравьиных колоний (СМК). В качестве основных отличий от классического АМК отмечаются [83, 84]:

- стратегия выбора пути модифицирована для обеспечения возможности достижения баланса между исследованием нового ребра и использованием накопленных знаний о проблеме;
- правило глобального обновления феромона применяется только к ребрам, которые принадлежат к лучшим решениям;
- правило локального обновления феромона будет применяться, когда агент закончит маршрут.

Стратегия выбора пути в СМК состоит из двух отдельных стратегий, а именно: эксплуатации и эксплорации пути. Стратегия эксплуатации пути является детерминированным правилом, которое всегда выбирает ребро с

наибольшим значением произведения веса феромона на видимость вершины. С другой стороны, стратегия эксплорации является стохастическим правилом

$$P_{ij}^k(t) = \frac{[\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}{\sum_{k \in J_{i,l}} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}, \quad j \in J_{i,k}, \quad (1.12)$$

где  $P_{ij}^k(t)$  – вероятность движения агента  $k$  от вершины  $x_i$  к вершине  $x_j$ ;  $\eta_{il}(t)$  – привлекательность пути от вершины  $x_i$  к вершине  $x_j$ , согласно [17] является априорной степенью осуществления перехода, определяется как  $\eta_{ij} = 1/D_{ij}$ , т. е. от вершины  $x_i$  к вершине  $x_j$ ;  $D_{ij}$  – геометрическое расстояние в двумерном пространстве между вершинами  $x_i$  и  $x_j$ , определяется как

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (1.13)$$

где  $(x_i, y_i)$  – координаты вершины  $i$ , а  $(x_j, y_j)$  – координаты вершины  $j$ ;  $J_{i,k}$  – множество не посещенных вершин для агента  $k$ , расположенного в вершине  $x_i$ ;  $\alpha, \beta$  – параметры, определенные пользователем, они же – параметры привлекательности перехода в вершину, определяющие относительную важность феромона в зависимости от расстояния (при  $\alpha = 0$  агент стремится выбирать кратчайшее ребро, при  $\beta = 0$  – ребро с наибольшим количеством феромона).

Использование термина привлекательности пути подразумевает собой то, что близкие друг к другу вершины имеют большую вероятность быть выбранными для посещения агентом. Стоит отметить, что данный фактор не наблюдается в поведении реальных муравьев, но используется здесь как искусственное средство повышения эффективности алгоритма [84].

При нахождении нового решения используется стратегия, которая является этапом принятия решения о переходе от текущей вершины  $x_i$  и  $x_j$ . В данном случае стратегия пути формируется как [83, 84]

$$s = \begin{cases} \operatorname{argmax} (\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)), & q \leq q_0 \text{ (эксплуатация)}, \\ P_{ij}^k \text{ (эксплорация)}, & \end{cases} \quad (1.14)$$

где  $q_0 \in (0,1)$  – определенный пользователем параметр, отвечающий за вероятность эксплуатации перехода;  $q$  – случайное число, равномерно распределенное в интервале  $[0, 1]$ .

АМК также применяется во многих областях, не исключая задачи оптимизации, причем как в роли самостоятельного инструмента (например, при решении задачи нахождения оптимального маршрута ТС с временными окнами [50]), так и вместе с каким-либо другим алгоритмом. В качестве примеров последнего можно привести решение задачи классификации (задачи поиска и выделения классификационных правил) с помощью связки муравьиного и генетического алгоритмов [115], а также исследование и разработку управления мобильным роботом на основе комбинирования алгоритма управления, базирующегося на методе потенциальных полей и муравьином алгоритме [6].

Также в решении задач маршрутизации транспорта объединение АМК с методом лучевого поиска и локальным методом является результативным в отношении построения полных решений с последующей оценкой нижних границ [62-64].

### **1.3.3 Пчелиный алгоритм для решения задач поиска оптимального маршрута**

Метод пчелиного роя (Bees Algorithm, алгоритм пчелиной колонии, пчелиный алгоритм, АПК) является одним из новейших методов, относящихся к рассматриваемому направлению. Первые статьи, в которых был предложен данный метод, были опубликованы в 2005 году [14]. Метод представляет собой эволюционный итеративный мультиагентный метод случайного поиска, основная идея которого состоит в моделировании поведения пчел при поиске нектара [117]. АПК состоит из трех основных



фаз: отправки разведчиков, отбора лучших источников и поиска в окрестности лучших источников.

АПК для решения задачи коммивояжера может быть математически представлен в виде формулы

$$p(i, j, k) = (\tau(i, j, k)^\alpha) \cdot \frac{(\dot{\tau}(j, k)^\beta)}{\Sigma[(\tau(i, j, k)^\alpha) \cdot (\dot{\tau}(j, k)^\beta)]}. \quad (1.14)$$

Для каждой пчелы  $i$  в колонии, начиная со случайно выбранного города  $j$ , выбирается следующий город  $k$  из всех не посещенных городов, основываясь на значениях привлекательности городов  $(\tau(i, j, k))$  и на значениях феромонов на ребрах графа  $(\dot{\tau}(j, k))$ , где пробегает все не посещенные города,  $\alpha$  и  $\beta$  – параметры алгоритма, задающие веса для значений привлекательности и феромонов соответственно.

После выбора следующего города, обновляются значения феромонов на ребрах графа

$$\dot{\tau}(j, k) = (1 - \rho) \cdot \dot{\tau}(j, k) + \rho \cdot \Delta \dot{\tau}(j, k), \quad (1.15)$$

где  $\rho$  – коэффициент испарения феромона,  $\Delta \dot{\tau}(j, k)$  – количество феромона, отложенного пчелой  $i$  на ребре  $(j, k)$  [92].

Если все города посещены, возвращение в начальный город  $j$  и окончание маршрута. Описанные операции повторяются для каждой пчелы в колонии, сохраняя лучший найденный маршрут.

К достоинствам АПК относятся простота в реализации и понимании, возможность адаптации к изменениям среды и балансировки между исследованием и использованием пространства поиска, а также быстрое действие, что подразумевает нахождение приближенных к оптимальным решениям за приемлемое время.

Среди недостатков упоминания заслуживает большое число свободных параметров метода, от значений которых, с одной стороны, зачастую сильно зависит эффективность метода, а с другой – отсутствуют какие-либо содержательные основания для выбора этих значений. Также отмечается

существенное падение скорости работы алгоритма в случае использования последовательной обработки.

Данный алгоритм находит свое применение в сфере оптимизационных задач, в частности, для решения задачи планирования беспроводной сети [117, 118], параметрической оптимизации на основе распараллеливания процесса поиска [87], поиска разливов нефти группой БПЛА [66], моделирования процессов оптимизации структуры сил и средств при пожаре [26] и т. д.

#### **1.3.4 Сравнительный анализ параметров эволюционных алгоритмов**

В таблице 1.2 представлена обобщенная информация о ГА, АМК и АПК, определенная на основе проведенного анализа каждого из описанных выше представителей ЭА, среди которой имеются материалы, являющиеся основанием для включения данных алгоритмов в состав интеллектуальной системы. Данная таблица не является исчерпывающей, однако она характеризует алгоритмы по основным параметрам.

Таблица 1.2 – Обобщенная информация рассмотренных эволюционных алгоритмов

	<b>Генетический алгоритм</b>	<b>Алгоритм муравьиной колонии</b>	<b>Алгоритм пчелиной колонии</b>
<b>Область применения алгоритма</b>	Робототехника, обучение нейронных сетей, <b>решение задач оптимизации</b> , компьютерные игры, модели искусственной жизни, оптимизация запросов к базам данных и т.д.	Решение комбинаторных задач, <b>решение задач оптимизации</b> , решение задач проектирования (развитие инфраструктуры предприятий) и т.д.	<b>Решение задач оптимизации</b> , криптография, инженерия и промышленность, планирование и проектирование сетей и коммуникаций и т.д.
<b>Размерность решаемых задач</b>	От тысячи и более обрабатываемых в поставленной задаче элементов.	До нескольких тысяч обрабатываемых в поставленной задаче элементов.	До нескольких сотен обрабатываемых в поставленной задаче элементов.
<b>Отличительная особенность</b>	Возможность эффективного оперирования и обработки больших объемов данных.	Повышенная эффективность при решении задач оптимизации маршрута.	Получение быстрого решения для сравнительно малых задач оптимизации.
<b>Основной недостаток</b>	Получение оптимального решения не гарантируется (квазиоптимальное решение).	Время (гарантируемой) сходимости невозможно определить или предсказать.	Большая потеря во времени и качестве решения при увеличении сложности решаемой задачи.

#### 1.4 Обзор современных средств решения задач оптимизации на основе эволюционных алгоритмов

Существуют готовые программные пакеты, реализованные на основе описанных выше алгоритмов и спроектированные для выполнения конкретных задач, в том числе – задач оптимизации, а также инструменты, которые позволяют их запрограммировать.

Самым известным из последних является пакет прикладных программ (ППП) Matlab, в котором неоднократно были реализованы как ГА [8, 53, 54,

115], так и АМК [3, 40, 97] и АПК [89]. Имеются как пользовательские реализации с применением встроенных в пакет инструментов, так и непосредственно инструкции по реализации этих алгоритмов в Matlab.

При всей своей практической применимости касательно реализации ЭА, данный ППП имеет недостатки, которые не могут быть устранены: высокая требовательность к навыкам пользователей, невозможность одновременной реализации нескольких алгоритмов, равно как и передача данных, и переключение между ними – Matlab не позволяет объединить несколько алгоритмов под одним интерфейсом для решения задачи более чем одним способом или переключать алгоритмы для улучшения получаемых результатов.

Из готовых программных решений, в основе которых лежит ГА, АМК или АПК, можно рассмотреть следующий набор.

1. Программное средство (ПС) AntColonySystem – консольная реализация АМК, написанная на языке C# [4]. Достоинством программного средства является возможность свободного доступа к программному коду для изменения начальных параметров. Недостатком является отсутствие графического интерфейса, из-за чего результаты расчетов представляются в виде итогового значения наилучшего маршрута без подробностей или какого-либо визуального представления.

2. ПС ABC2 производит вычисления, основываясь на АПК [2, 33]. Интерфейс программы позволяет не только установить параметры в необходимые значения, но и выбрать функцию для расчета, а также визуализирует результаты в виде графика, как показано на рисунке 1.2. Недостатком данного программного средства является наличие только англоязычного интерфейса.

3. ПС Python code of the basic Artificial Bee Colony является базовой реализацией алгоритма Artificial Bee Colony (ABC) на языке Python. Она была разработана с целью предоставить пользователям простой и эффективный способ использования алгоритма ABC для решения задач

оптимизации [20]. На рисунке 1.3 можно увидеть примеры работы алгоритма в 1D- и 2D-проекции, где осуществляется поиск наилучшего источника.

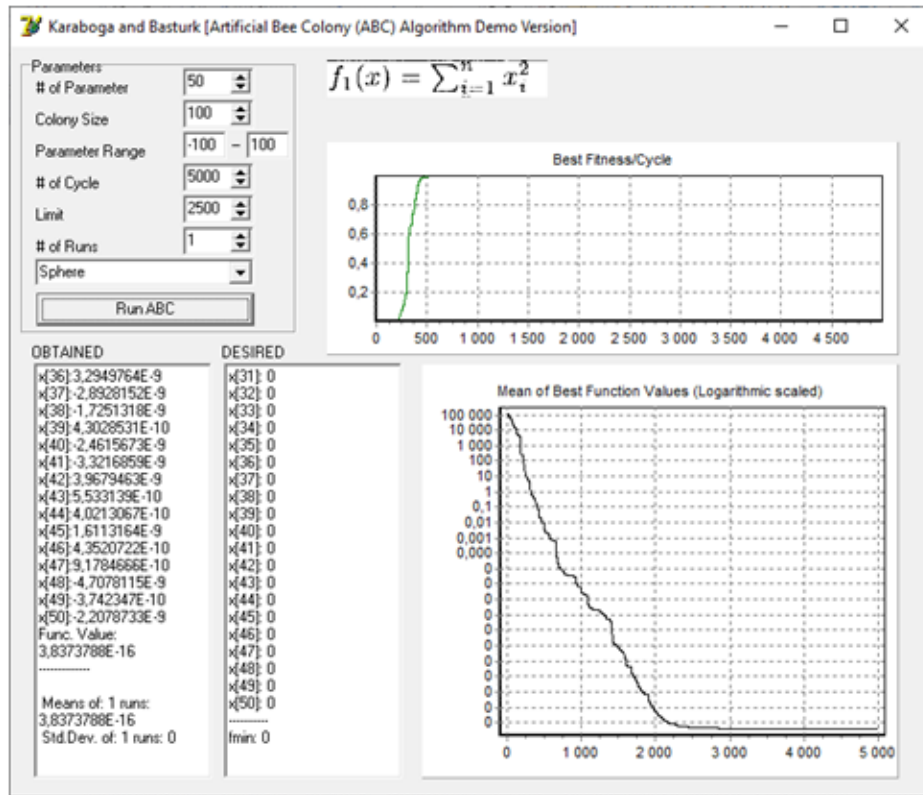


Рисунок 1.2 – Интерфейс программного средства ABC2

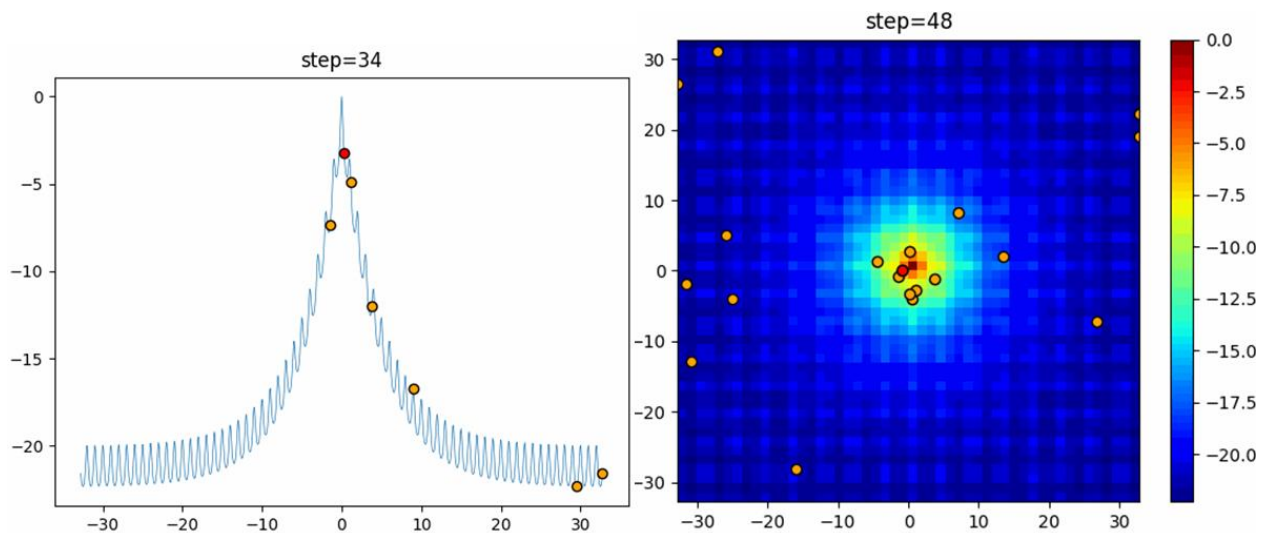


Рисунок 1.3 – Проекция 1D-работы алгоритма пчелиной колонии

4. ПС ABC Algorithm Source Code by Delphi for Constrained Optimization представляет собой реализацию АПК на языке программирования Delphi. Она имеет predetermined параметры, которые могут быть изменены в зависимости от требуемой точности решения. Приложение применимо для решения задач оптимизации с ограничениями, например, в областях промышленности и инженерии [1]. На рисунке 1.4 представлено окно программы в процессе работы.

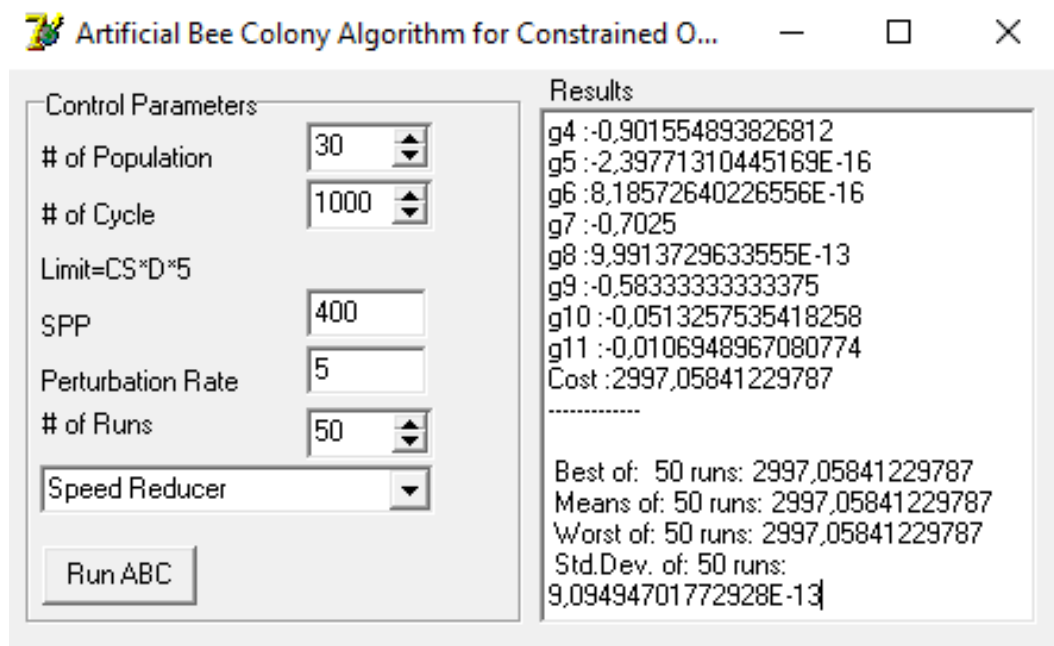


Рисунок 1.4 – Окно программы ABC Algorithm Source Code

Рассмотренные системы и решения справляются с возложенной на них задачей. При этом, как уже отмечалось выше, они не являются универсальными в широком смысле, что является их недостатком.

### **1.5 Применение систем поддержки принятия решений с использованием адаптивных алгоритмов**

Вопрос принятия решений в широком смысле является пропорционально сложным количеству параметров, на него влияющих.

Первоначально для решения данной проблемы использовались один или несколько профессионально подготовленных, опытных в конкретной сфере людей – экспертов. Позднее, с развитием информационных технологий, знания и навыки человека дополнились возможностями вычислительных машин, способных облегчить процесс принятия решений, что привело к появлению систем и подсистем [51, 52] поддержки принятия решений. Данные системы (как интеллектуальные [46], так и адаптивные [86]) в процессе проведения вычислений нередко опираются на экспертные системы (ЭС) – программные системы, которые при решении задач, трудных или времязатратных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом [119].

В научной литературе предлагаются различные определения для экспертных систем [76, 119, 123]. Несмотря на различия в определении понятия экспертной системы, общий смысл остается неизменным: экспертная система является аналогом либо помощником специалиста-эксперта в проблемных ситуациях.

ЭС предназначены для решения так называемых неформализованных задач, отличающихся ошибочностью, неоднозначностью, неполнотой, противоречивостью и изменчивостью как исходных данных, так и предметной области в целом. Пространство решений таких задач имеет очень большую размерность и перебору не поддается.

Чтобы программу можно было отнести к категории ЭС, она должна отвечать следующим требованиям:

- обладание знаниями, то есть информацией, на основе которой программа может принимать решения или получать новую информацию;
- концентрация на определенной предметной области, предполагающая специфическую организацию знаний, отношений между ними и принципов построения логических выводов, характерных для данной предметной области;

– решение проблем на основе имеющихся знаний, на которые должна быть ориентирована программа.

Важной частью любой экспертной системы является база знаний (БЗ), которую принято считать ее ядром, и которая представляет собой совокупность знаний предметной области в форме, понятной для эксперта и пользователя, причем обычно на языке, близком к естественному [88, 109]. Создание и использование БЗ реализуется с помощью комплекса программных, языковых и интеллектуальных средств, который носит название системы управления базой знаний (СУБЗ) [40].

На практике во многих экспертных системах и БЗ содержимое базы знаний подразделяют на то, что называют «фактами» и «правилами». Факты – элементарные единицы знания (простые утверждения о характеристиках объекта), правила служат для выражения связей, зависимостей между фактами и их комбинациями. Отсюда классификацию знаний можно представить следующим образом:

- понятия (математические и нематематические);
- факты;
- правила, зависимости, законы, связи;
- алгоритмы и процедуры.

Прямое использование знаний из БЗ для решения задач обеспечивается механизмом получения решений – процедурой поиска, планирования, решения. Механизм решения дает возможность извлекать из базы знаний ответы на вопросы, получать решения, формулируемые в терминах понятий, хранящихся в базе.

## **1.6 Постановка задачи исследования**

Целью данной работы является повышение эффективности системы поддержки принятия решения при многокритериальной оптимизации



транспортной задачи средствами модифицированных эволюционных алгоритмов.

В соответствии с этим, можно выделить следующие задачи:

- изучение предметной области, формирование проблематики;
- изучение эволюционных алгоритмов: анализ принципов работы, формирование структуры алгоритмов, определение сильных и слабых сторон;
- формирование структуры адаптивной системы, определение входящих в нее компонентов и подсистем, описание взаимодействия между ними;
- проектирование механизмов определения эффективности решений, переключения и модификации активного алгоритма;
- разработка программной составляющей адаптивной системы: интерфейса пользователя, алгоритмических модулей, системы взаимодействия пользователя и модулей;
- апробация разработанного программного решения на многокритериальной транспортной задаче;
- подведение итогов проведенного исследования.

## ГЛАВА 2. ФОРМАЛИЗАЦИЯ МНОГОКРИТЕРИАЛЬНОЙ ЗАДАЧИ ОПТИМИЗАЦИИ И МЕХАНИЗМЫ МОДИФИКАЦИЙ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

### 2.1 Схемы работы эволюционных алгоритмов

Принцип работы адаптивной системы поддержки принятия решений подразумевает взаимодействие с пользователем, которое характеризуется следующими аспектами:

- постановка пользователем многокритериальной задачи оптимизации, в которую входят формирование ЦФ, определение условий задачи и ее ограничений;

- анализ системой определенных пользователем значений, обращение к базе знаний, сопоставление их с материалами базы;

- определение алгоритма, решавшего схожую задачу, и его параметров, установка его в качестве рекомендуемого с преднастройкой параметров;

- смена пользователем активного алгоритма либо изменение его настроек;

- запуск процедуры расчета с применением определенного алгоритма;

- систематический анализ процесса решения и получаемых результатов;

- замена используемого расчетного (активного, т. е. того, который в текущий момент времени используется системой для осуществления расчетов) алгоритма с целью улучшения результатов расчета;

- визуализация результатов расчета алгоритма пользователю с занесением в базу данных результатов;

- сохранение результатов в базе знаний адаптивной системы по запросу пользователя.

Каждый из этих аспектов включает в себя определенную последовательность действий – алгоритм работы, завязанный на операциях, проводимых либо пользователем, либо системой.

Схема работы АМК представлена на рисунке 2.1. Начальным этапом является инициализация муравьев колонии, который также включает в себя их размещение, после чего следует построение маршрутов муравьями колонии и оценка построенных маршрутов по качеству. Следующим этапом является определение и сохранение лучшего решения из полученных и определение изменения величины уровня феромона. После этого производится обновление значений феромона на маршрутах. Данная очередность действий (за исключением инициализации) повторяется до тех пор, пока не будут выполнены условия завершения алгоритма.



Рисунок 2.1 – Схема классического АМК

Схема с описанием работы ГА представлена на рисунке 2.2. На начальном этапе случайным образом создается некоторое количество начальных векторов, называемых начальной популяцией. Далее к маршрутам множества применяются операции скрещивания и мутации, целью которых является создание нового «поколения». Исключающим фактором попадания в локальный экстремум является тот факт, что ГА использует одновременно несколько точек поискового пространства [115].

Схема работы АПК представлена на рисунке 2.3.

В начале алгоритма создается популяция из  $N$  пчел-разведчиков, каждая из которых представляет потенциальное решение задачи оптимизации. Каждая пчела-разведчик случайным образом выбирает точку в пространстве поиска и вычисляет значение ЦФ (качество нектара) в этой точке. Затем все пчелы-разведчики возвращаются в улей и делятся информацией о своих находках с другими пчелами.

На основе полученной информации выбирается  $M$  лучших источников нектара, которые будут дальше исследоваться. Количество пчел, отправляемых к каждому из лучших источников, пропорционально их качеству. Таким образом, алгоритм предпочитает более перспективные регионы пространства поиска.

Каждая пчела, отправленная к одному из лучших источников, выполняет локальный поиск в небольшой окрестности этого источника. Для этого она выбирает случайную точку в окрестности и сравнивает ее качество с качеством исходного источника. Если новая точка лучше, то пчела запоминает ее и забывает старую. В противном случае пчела остается при своем источнике. Затем пчела возвращается в улей и сообщает о своем результате.

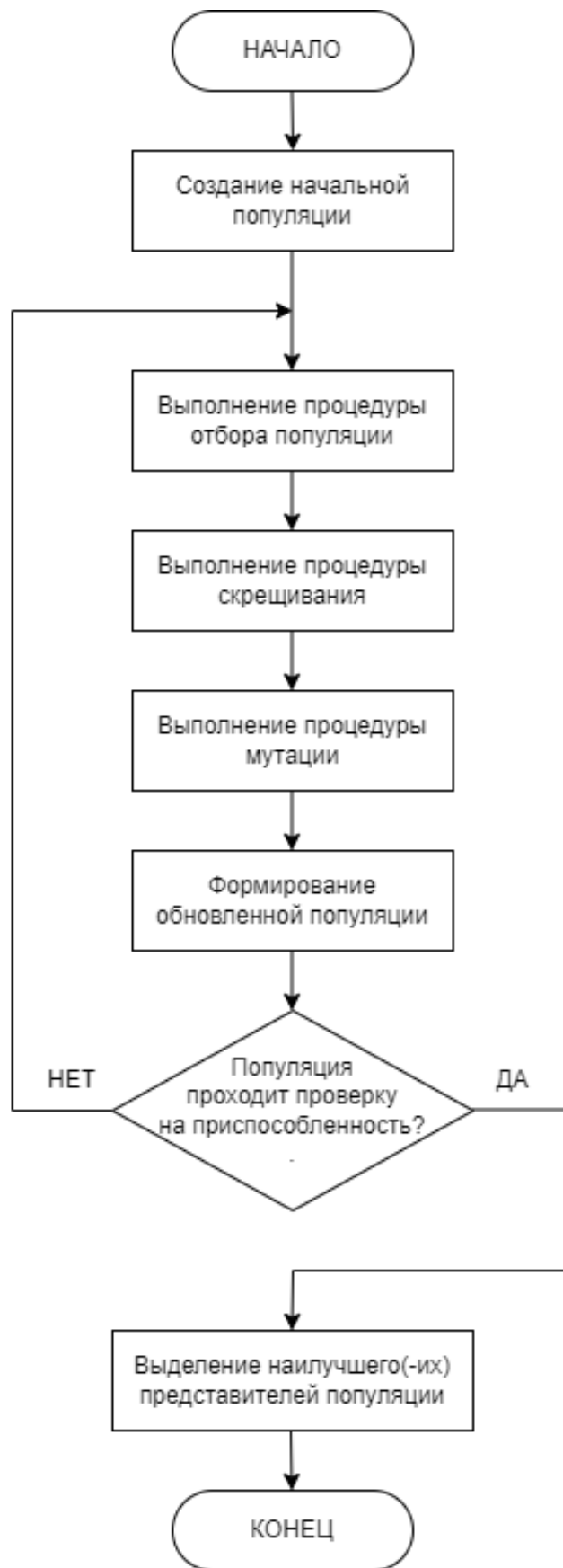


Рисунок 2.2 – Схема работы ГА

После того, как все пчелы завершили свой локальный поиск, происходит отбор лучших источников снова. Те источники, которые не улучшились за заданное количество итераций, считаются исчерпанными и забрасываются. Вместо них создаются новые пчелы-разведчики, которые начинают поиск с новых случайных точек. Таким образом, алгоритм избегает застревания в локальных оптимумах и поддерживает разнообразие решений.

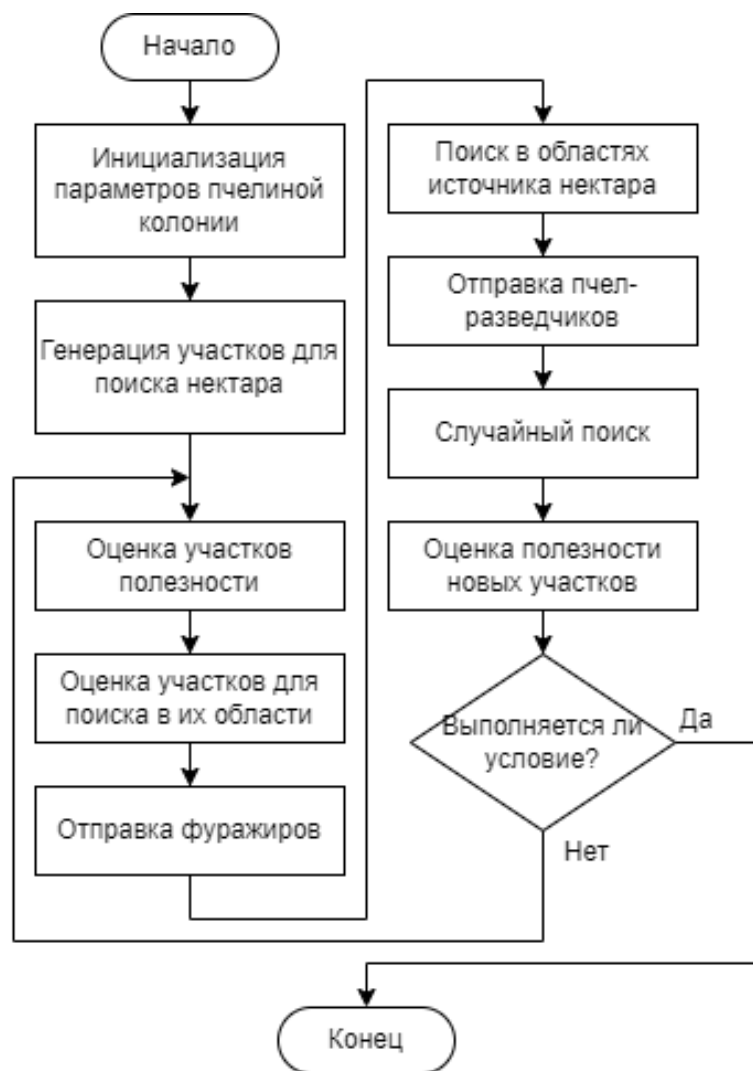


Рисунок 2.3 – Схема работы АПК

Алгоритм продолжает повторять эти три фазы до тех пор, пока не будет достигнут критерий остановки, например, максимальное количество итераций или желаемая точность решения. На выходе алгоритм возвращает лучшее найденное решение за все время работы.

## 2.2 Математическая постановка многокритериальной транспортной задачи с временными ограничениями

В п. 1.2.2 был рассмотрен вариант многокритериальной транспортной задачи. На его примере сформулируем следующую МТЗ [31].

Пусть на момент времени  $t$  имеется автопарк из набора однотипных транспортных средств (ТС) в количестве  $k$  единиц ( $k = \overline{1, K}$ ) и набор заказов  $j = \overline{1, J}$ . Каждое ТС, как и каждый заказ, характеризуется определенным набором координат  $(x_i, y_i)$ .

Также имеется некоторый набор узлов, каждый из которых является некоторой точкой из имеющихся координат, и временное окно  $t_j^* = [t_j^{opn}, t_j^{cls}]$  для каждого заказа, характеризуемое отрезком времени со значениями открытия и закрытия доступа ( $t_j^{opn}$  и  $t_j^{cls}$  соответственно).

Первую ЦФ можно сформулировать следующим образом: минимизация стоимости суммарного маршрута всех ТС автопарка. Спецификой данной задачи является фактор ее решения в любой момент времени, что означает возможность изменения координат ТС. В связи с этим введем в рассмотрение функцию  $\rho(k, j, t)$ , которая будет определять расстояние от ТС  $k$  до поставщика  $j$  в момент времени  $t$ . Также введем функцию  $f(\cdot)$ , определяющую стоимость маршрута для данного ТС по данному расстоянию.

Искомая матрица может быть представлена в виде

$$X = \begin{pmatrix} x_{11} & \dots & x_{1j} \\ \dots & \dots & \dots \\ x_{k1} & \dots & x_{kj} \end{pmatrix}, \quad (2.1)$$

где каждый элемент  $x_{kj}$  определяется как

$$x_{kj} = \begin{cases} 1, & \text{если ТС } k \text{ следует к заказу } j, \\ 0, & \text{в обратном случае} \end{cases} \quad (k = \overline{1, K}, j = \overline{1, J}). \quad (2.2)$$

Также стоит отметить допущение, что выполнение заказа может быть связано с поездкой в некоторую точку.

Одновременно с этим имеется матрица

$$H = \begin{pmatrix} h_{11} & \dots & h_{1J} \\ \dots & \dots & \dots \\ h_{K1} & \dots & h_{KJ} \end{pmatrix}, \quad (2.3)$$

где каждый элемент  $h_{kj} \in (0,1]$  обозначает степень важности выполнения конкретного заказа. Соответственно, чем больше значение элемента, тем выше приоритет выполнения заказа.

В дополнение имеется матрица

$$T = \begin{pmatrix} t_{11} & \dots & t_{1J} \\ \dots & \dots & \dots \\ t_{K1} & \dots & t_{KJ} \end{pmatrix}, \quad (2.4)$$

где  $t_{kj} \geq 0$  характеризуется временем выполнения ТС  $k$  заказа  $j$ . В случае, если  $t_{kj} = 0$  подразумевается, что ТС  $k$  не имеет возможности выполнить заказ  $j$ .

На основании этих данных ЦФ может быть сформирована как

$$F_1 = \min \sum_{k=1}^K \sum_{j_1=1}^J x_{kj} \left( f(k, \rho(k, j_1, t)) + f(k, \rho(j_1, j_2, t)) \right), \quad (2.5)$$

где  $j_1$  – некоторый заказ из множества заказов  $J$ , а  $j_2$  – конечная точка для заказа  $j_1$ .

На основании этого вторую ЦФ можно сформулировать как максимизацию выполнения заказов с наибольшими приоритетами, т. е.

$$F_2 = \max \sum_{k=1}^K \sum_{j=1}^J x_{kj} h_{kj}. \quad (2.6)$$

Отсюда третья ЦФ может быть сформулирована следующим образом:

$$F_3 = \min \sum_{k=1}^K \sum_{j_1=1}^J x_{kj} t_{kj}, t_{kj} > 0. \quad (2.7)$$

Также необходимо ввести ограничение на единственность выбираемого ТС, которое представимо в виде



$$\sum_{k=1}^K x_{kj} = 1 \quad (j = \overline{1, J}). \quad (2.8)$$

Дискретность обозначенной задачи формулируется следующим ограничением:

$$x_{kj} \in \{0, 1\}. \quad (2.9)$$

Далее опишем ограничения, связанные со временем. Пусть функция  $T(\cdot)$  определяет длительность пребывания в пути, задаваемом расстоянием  $\rho$ .

Тогда

$$\sum_{k=1}^K x_{kj} T(\rho(k, j, t)) \leq t_j^{cls} \quad (j = \overline{1, J}) \quad (2.10)$$

и дополнительно, с учетом существования временных окон, накладывается ограничение-запрет на посещение вне временного окна

$$x_{kj} T(\rho(k, j, t)) \in t_j^* \quad (k = \overline{1, K}, j = \overline{1, J}). \quad (2.11)$$

На основании п. 1.2.3 сформулированная МТЗ может быть определена в виде

$$F = \min\{F_1, -F_2, F_3\} \quad (2.12)$$

либо

$$F = \max\{-F_1, F_2, -F_3\} \quad (2.13)$$

в зависимости от того, какой из критериев (либо минимизация стоимости (2.5) или минимизация времени (2.7), либо максимизация важности (2.6)) будет являться превалирующим.

Отсюда возможных вариантов парето-оптимизации может быть несколько. В качестве примера можно рассмотреть следующий вариант.

Пусть для некоторого предприятия по решению коллегии экспертов принято решение, что во главе должна быть необходимость в минимизации временных затрат с учетом степеней важности, а стоимость перевозки ставится на последнее место. В таком случае за основу берется формула (2.12), которая может быть переписана с учетом выдвинутых экспертами важностей критериев как

$$F = \min\{F_3, -F_2, F_1\}. \quad (2.14)$$

Тогда на основании (2.12)-(2.14), а также (1.11) можно обозначить следующее:

$$y_1^* = F_3, \quad y_2^* = \begin{matrix} -F_2, \\ F_3 \leq y_1^* - \Delta y_1 \end{matrix}, \quad y_3^* = \begin{matrix} F_1. \\ F_3 \leq y_1^* - \Delta y_1 \\ F_2 \leq y_2^* - \Delta y_2 \end{matrix}. \quad (2.15)$$

Представленные операции, равно как и формирование самой МТЗ, требуют наличия одного либо нескольких экспертов. От их решений и заключений будет зависеть последующий ход решения данной задачи.

В рамках данного диссертационного исследования решено остановиться на методе последовательных уступок, который интересен, в первую очередь, гибкостью и универсальностью.

Помимо возможности применения в отношении различных по требованиям многокритериальных задач оптимизации, он позволяет учитывать различные факторы, как, например, упомянутые временные ограничения. Особенно актуальным данное свойство будет в тех случаях, когда имеется большое количество ограничений и критериев.

### 2.3 Механизмы модификации эволюционных алгоритмов

Модификации эволюционных алгоритмов преследуют одну основную цель: видоизменить механизм работы алгоритма так, чтобы на выходе был получен наиболее приближенный к оптимальному вариант решения поставленной задачи за наименьшее возможное время, иными словами – обеспечение увеличения скорости сходимости с минимальными потерями в качестве решений.

Ниже представлены описания модификаций эволюционных алгоритмов, рассмотренных в п. 1.3, которые направлены на улучшение скорости вычислений и могут быть применены в адаптивной системе поддержки принятия решений при многокритериальной оптимизации.

### 2.3.1 Модификации алгоритма муравьиной колонии

Основные известные модификации АМК (например, ММАС [74] и система элитных муравьев [82]) предлагают ускорение процесса поиска путем сокращения числа рассматриваемых маршрутов посредством влияния на формирование и изменение принципов откладывания и испарения феромонов, как следствие – увеличение скорости сходимости [104]. Однако изменение основных коэффициентов алгоритма  $(\alpha, \beta, p, q)$  также непосредственно влияет на то, каким образом будут производиться расчеты. В контексте АМК коэффициенты характеризуются так:  $\alpha$  и  $\beta$  – коэффициенты «видимости» муравьем города и «чувствительности» к оставленному феромону соответственно, а  $p$  и  $q$  – коэффициенты «оставления» и «испарения» феромона.

В рамках диссертационного исследования было решено использовать как возможность изменения коэффициентов классического алгоритма, так и обозначенной выше модификации с применением системы элитных муравьев. Данная модификация позволит ярче выделять наиболее благоприятные маршруты, что будет способствовать увеличению скорости сходимости.

Для формирования феромона элитных муравьев (т. е. тех из колонии, чей маршрут оказался лучше остальных) за основу берется формула (1.11) и принимает следующий вид:

$$\Delta\tau_{elite} = A_{elite} Q/L, \quad (2.16)$$

где элемент  $A_{elite}$  характеризуется как «авторитет» данного муравья.

Как следствие – с помощью формулы (2.16) (в частности, с помощью коэффициента  $A_{elite}$ ) можно осуществлять воздействие на влияние элиты колонии в отношении остальных муравьев.

Также стоит отметить, что существует несколько методов начального расположения колонии:

- «дробовик», когда муравьи распределяются случайным образом;
- «фокусировка», когда все муравьи колонии размещаются в одной точке (вершине графа);
- «одеяло», которое считается стандартным размещением колонии в формате «один муравей – одна вершина»;
- «блуждающая колония», когда каждую следующую итерацию алгоритма начальное положение колонии изменяется [68].

С учетом постановки МТЗ с временными ограничениями, обозначенной в п. 2.2, более приемлемым вариантом при расчетах будет использование гибридного метода, сочетающего в себе элементы всех методов, поскольку расположение ТС может быть как разрозненным, так и частично либо полностью сгруппированным в одной точке.

### **2.3.2 Модификации генетического алгоритма**

Модификации ГА позволяют влиять на различные его аспекты: правила скрещивания и мутации, принципы осуществления отбора, алгоритмы формирования новых популяций и т.д. Изменения алгоритма направлены как на снижение времени поиска (например, TLP [21]), так и на повышение качества решения (например, модель Голденберга [73]).

Для модификации механизма скрещивания предлагается использование метода Order-based Crossover (OX), в частности, его первую реализацию OX operator (OX1). Данный механизм является вариацией равномерного пересечения, введенной с целью сохранения относительного порядка символов в комбинируемых последовательностях. Первая реализация OX1 генерирует две точки пересечения для обоих родительских элементов. Символы между двумя точками вырезания копируются в дочерние элементы. Затем, начиная со второй точки вырезания одного родительского элемента,

символы из другого родительского элемента копируются в один дочерний элемент в том же порядке, исключая те, которые уже существуют [6].

Для изменения методов отбора предлагается использование следующих трех модификаций:

– рулетка (или пропорциональный отбор, рисунок 2.4) – решения (особи) отображаются в отрезке линии (т. н. секторе рулетки) так, чтобы их размер был пропорционален значению целевой функции для данного решения (особи), после чего генерируются случайные числа в диапазоне  $[0,1]$ , и в промежуточную популяцию отбираются те решения (особи), в чей отрезок попадают полученные случайные числа;

– стохастический (стохастическая универсальная выборка) – метод для отбора потенциально полезных решений для рекомбинации, предполагающий полностью случайный отбор решений (особей) из множества;

– турнирный (рисунок 2.5) – случайным образом производится отбор решений (особей), среди которых определяется лучшее, после чего процесс повторяется, пока не будет заполнена популяция.

На рисунках 2.4-2.5 используются следующие обозначения:  $n$  – численность популяции,  $t$  – численность отобранных индивидов (родителей),  $P$  – множество индивидов размером  $n$ ,  $P_n$  – множество отобранных индивидов размером  $t$ ,  $k$  – размер турнира,  $C$  – шанс индивида пройти отбор.

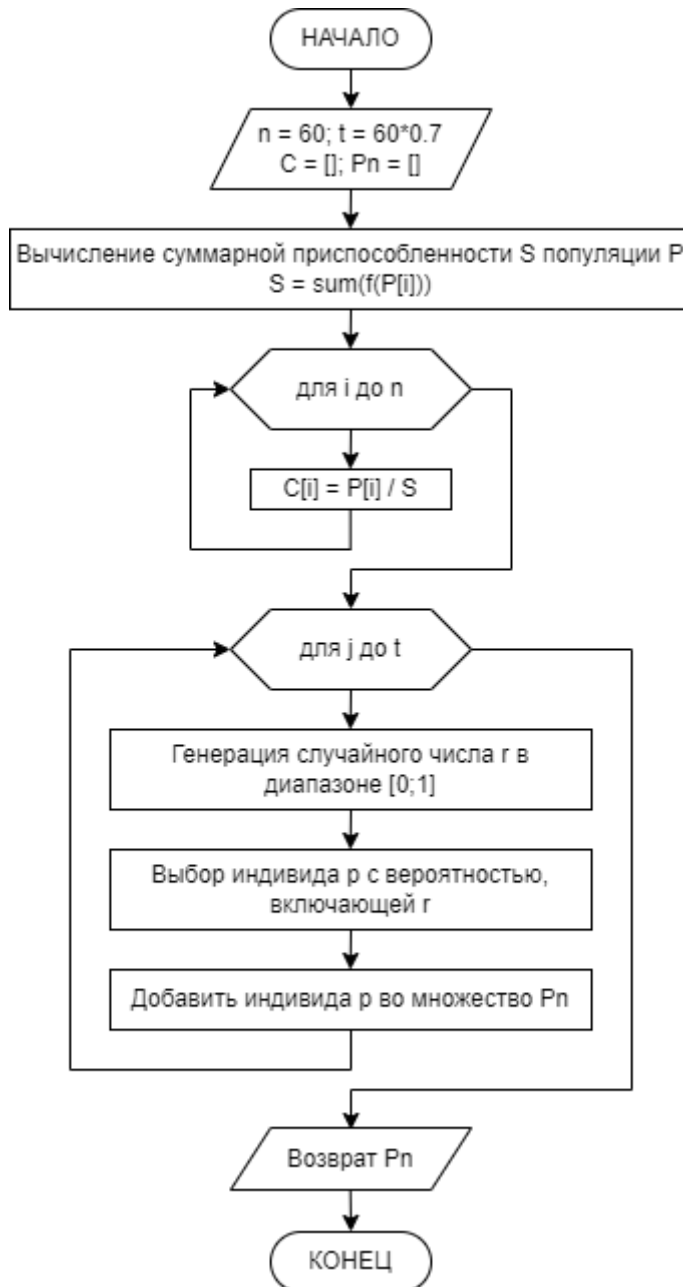


Рисунок 2.4 – Схема метода отбора «рулетка»

Для описания используемых методов мутации, исходное множество  $S$  представлено в соответствии с формулой

$$S = \{x_1, x_2, \dots, x_n\}, \quad (2.17)$$

где  $x_i (i = \overline{1, n})$  – элемент подмножества,  $n$  – длина подмножества. Измененные методы мутации формулируются как «смена индексов» и «реверс подмножества». Метод смены индексов может быть представлен в виде следующего выражения

$$S' = \{x_1, x_2, \dots, x_{i-1}, x_j, x_{i+1}, \dots, x_{j-1}, x_i, x_{j+1}\},$$

$$i, j \in n, i \neq j, \quad (2.18)$$

где  $i$  и  $j$  – некоторые числа.

Метод реверса подмножества описывается равенством

$$S' = \{x_1, x_2, x_{b-1}, x_{b+l-1}, x_{b+l-2}, \dots, x_b, \dots, x_n\},$$

$$b - l \leq 0, b + j \geq n, \quad (2.19)$$

где  $b$  – случайно сгенерированное число (индекс начала),  $l$  – длина подмножества.

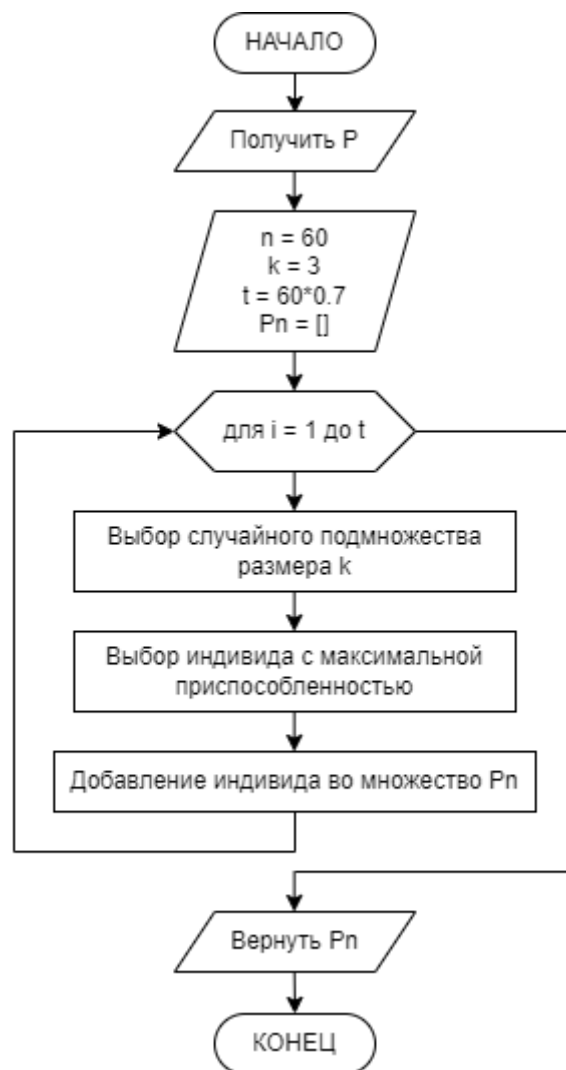


Рисунок 2.5 – Схема турнирного метода отбора

### 2.3.3 Модификации алгоритма пчелиной колонии

Модификации АПК нередко представляют из себя изменения алгоритма в соответствии с задачей (например,  $N(Sol)$ ) [117, 118] и работают в тандеме, например, с другими алгоритмами и нейронными сетями.

В нашем случае для АПК наиболее оптимальным вариантом является использование модификаций, связанных с использованием измененных методов поиска нового источника нектара. В контексте адаптивной системы выбор пал на два метода: «смена индексов» и «реверс подмножества». Поскольку данные методы аналогичны подобным для ГА, то за основу их математических формулировок принимаются формулы (2.18) для смены индексов и (2.19) для реверса подмножеств.

## 2.4 Представление базы знаний адаптивной системы

Как было отмечено в п. 1.5, БЗ является важной частью экспертной системы (не в последнюю очередь как элемент искусственного интеллекта), а потому она должна быть обязательной составляющей структуры адаптивной системы.

БЗ включает в себя две основные составляющие: базу данных (БД), содержащую некоторые данные в определенной предметной области, и правила вывода, с помощью которых осуществляется взаимодействие с хранящимися данными.

В рамках адаптивной системы поддержки принятия решений при многокритериальной оптимизации в БД подразумевается хранение следующей информации:

- постановка задачи оптимизации;
- условия задачи оптимизации;
- ограничения задачи оптимизации;
- алгоритм, использованный для решения задачи оптимизации;
- параметры использованного алгоритма;



- модификация использованного алгоритма (при наличии);
- настройки модификации алгоритма (при наличии);
- время поиска решения задачи оптимизации;
- полученный результат;
- был ли прерван алгоритм или нет;
- был ли заменен алгоритм или нет.

Работа БЗ с представленными материалами позволяет адаптивной системе на этапе подготовки к вычислениям определить, насколько близка поставленная задача к тем, данные о которых содержатся в БД, и предоставить информацию о том, каким алгоритмом решалась подобная задача, с какими параметрами и результатами.

Относительно правил вывода БЗ должна содержать такой набор правил, который позволяет не только проводить анализ и сопоставление при начальных настройках задачи и алгоритма, но и обеспечивать отслеживание ситуации в процессе вычислений.

Например, с помощью БЗ пользователю могут быть предоставлены рекомендации относительно модификации или замены активного алгоритма в тех случаях, когда время расчета на текущей итерации превышает допустимое или разница в качестве получаемых решений в течение нескольких итераций неизменно либо меньше минимально допустимого значения.

Также каждое последующее вычисление (вместе с результатами) может быть зафиксировано в БЗ, тем самым пополняя банк знаний адаптивной системы, улучшая процессы адаптации для последующих решений.

## **2.5 Разработка механизмов переключения эволюционных алгоритмов**

Многокритериальная оптимизация включает в себя широкий спектр различных задач, к каждой из которых может быть применен любой из существующих алгоритмов. Однако каждая задача, даже имея множество схожестей с другими, является уникальной, из-за чего не всякий алгоритм может найти ее решение, не говоря уже о времени поиска.

В связи с этим адаптивная система должна иметь возможность переключать активный алгоритм в процессе поиска решения задачи оптимизации в случае, если поиск затягивается слишком сильно или находимые в ряде итераций решения несущественно отличаются друг от друга и одновременно не являются оптимальными.

Для адаптивной системы был разработан механизм контроля текущего решения (рисунок 2.6). Он представляет собой процесс проверки и анализа, проводимый после каждой итерации в ходе решения задачи. Его целью является выявление наличия или отсутствия улучшения получаемых результатов в сравнении с предыдущими в течение нескольких итераций подряд. Количество итераций задается пользователем.

В случае отсутствия видимых улучшений результатов (разница между полученным в текущей и предыдущей итерациях решений составляет меньше, чем установленный пользователем либо рекомендованный системой порог) в течение нескольких итераций подряд передает данные основной системе и пользователю для дальнейшей корректировки – применения модификации для активного алгоритма, либо его замена на другой доступный (рисунок 2.7).

Под модификацией алгоритма понимается как внесение корректировок в текущие настройки (например, изменение значений коэффициентов для АМК), так и применение, например, других методов отбора в ГА.

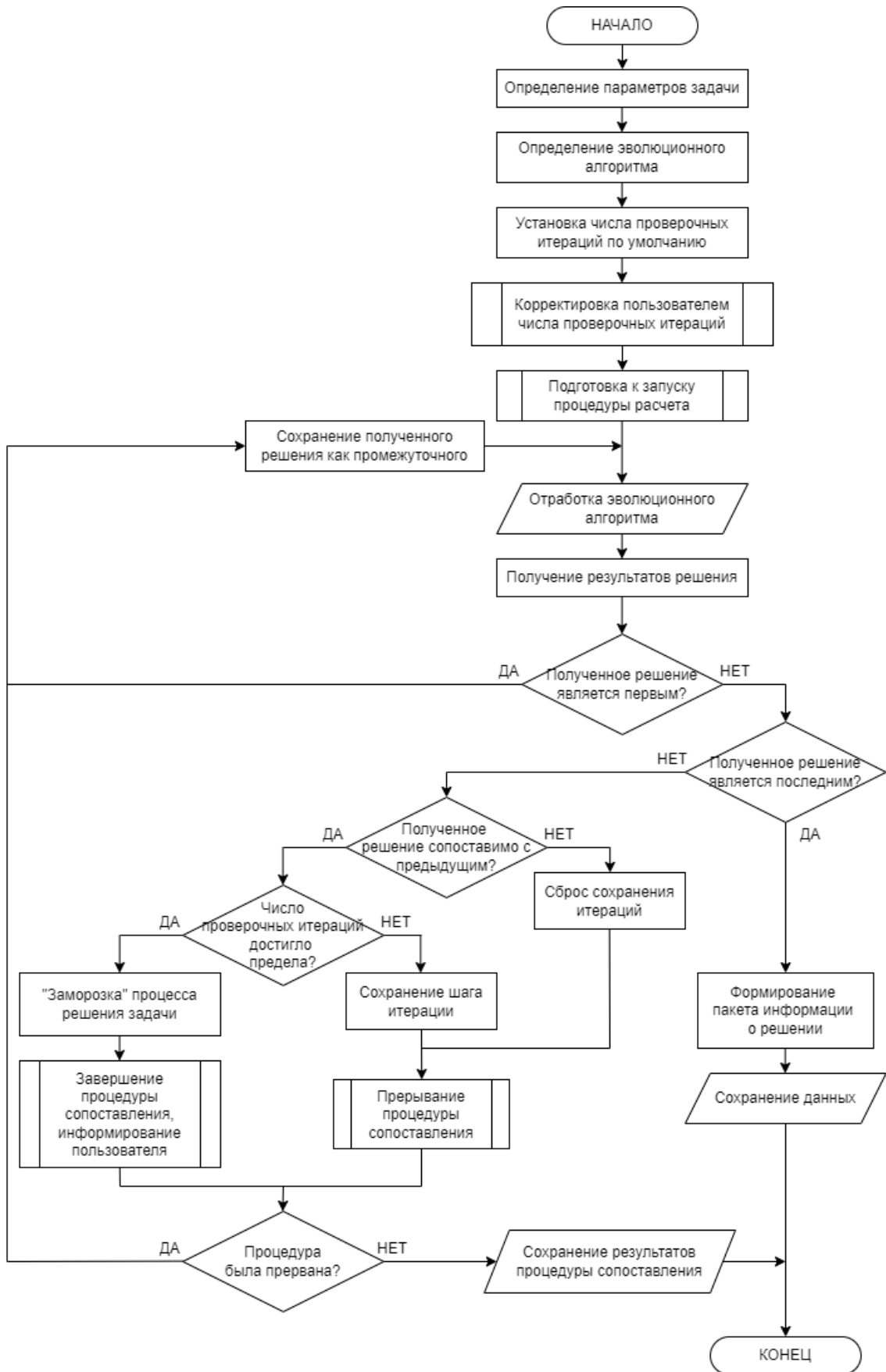


Рисунок 2.6 – Схема контроля текущего решения

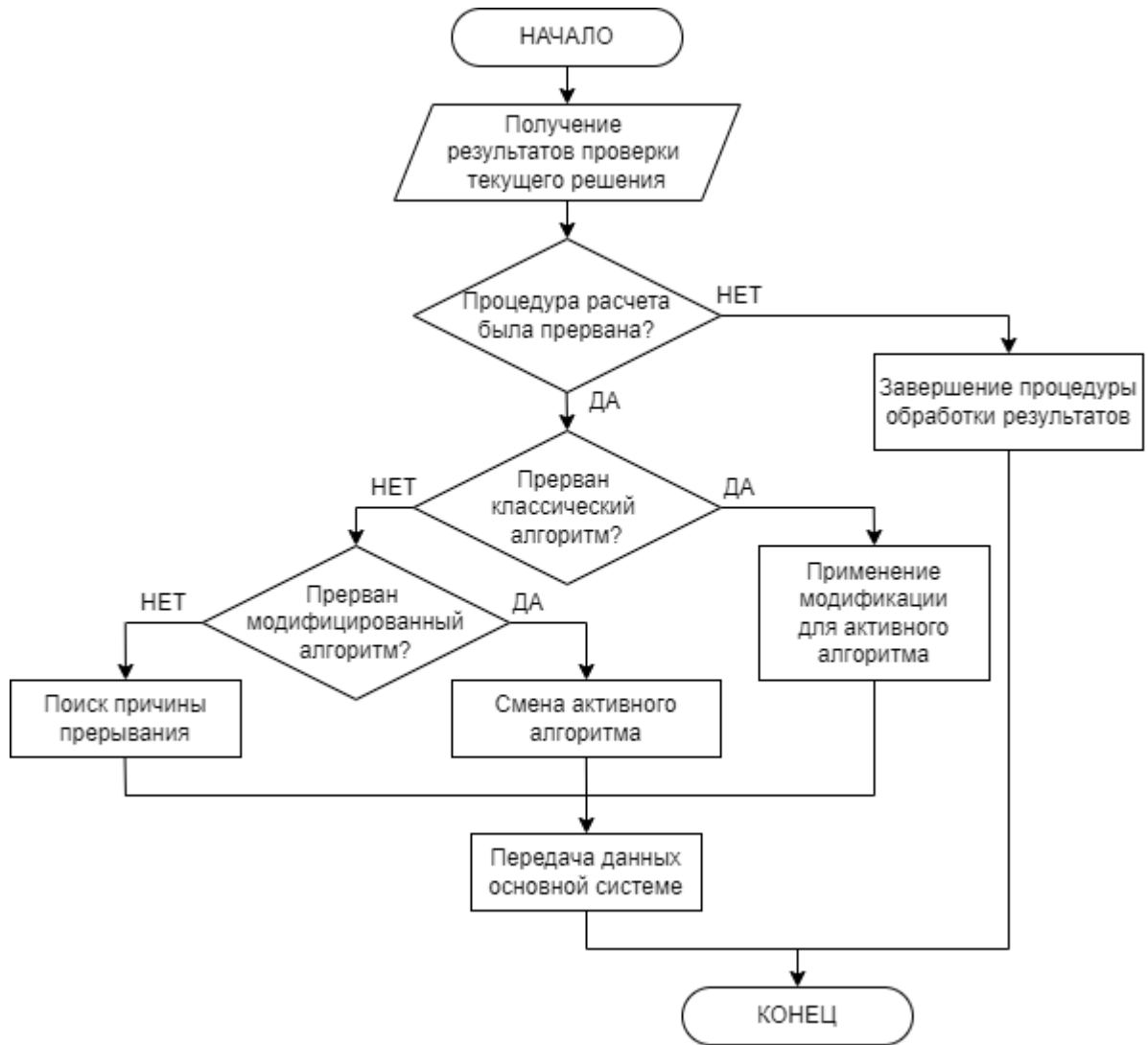


Рисунок 2.7 – Схема обработки результатов проверки текущего решения

## 2.6 Выводы

В данной главе рассмотрены вопросы формализации многокритериальной задачи оптимизации и механизмы модификаций эволюционных алгоритмов, характеризующиеся следующими основными результатами.

1. Проведен комплексный анализ существующих эволюционных алгоритмов (ГА, АМК, АПК), применяемых для решения многокритериальных задач оптимизации, в частности – многокритериальной

транспортной задачи, а также рассмотрены варианты их модификаций, направленных на их качественные улучшения.

2. Сформулировано специализированное математическое описание многокритериальной транспортной задачи с временными ограничениями, предполагающее применение при ее решении модифицированных эволюционных алгоритмов, механизмов проверки текущего решения и оперативной замены активного алгоритма.

3. Приведено представление базы данных адаптивной системы поддержки принятия решений, в частности, сформулирована хранящаяся в ней информация о формулировках задач и результатах их решения.

4. Разработаны схемы контроля и обработки результатов текущего решения с учетом информации о решении на предыдущем шаге (итерации) и ограничений по количеству итераций и формирующие рекомендации для модификации либо смены алгоритма.

## **ГЛАВА 3. СТРУКТУРА АДАПТИВНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ**

### **3.1 Общая структура работы адаптивной системы поддержки принятия решений**

В рамках данного исследования была разработана структура адаптивной системы поддержки принятия решений (АСППР), представленная на рисунке 3.1 [2, 32, 33].

В состав данной системы входят следующие элементы:

- модуль пользовательского интерфейса;
- модуль работы с документами;
- модуль математического ядра поддержки ЭА;
- модуль настройки системы;
- модуль формирования ЦФ;
- модуль справочной системы;
- модуль визуализации.

Модуль пользовательского интерфейса предоставляет пользователю необходимые инструменты, в том числе возможность настройки алгоритмов, а также просмотр результатов работы. Помимо этого, на данный модуль возлагается задача по обеспечению взаимодействия между остальными модулями АСППР.

Каждый модуль должен быть независимым от работы других модулей. Для связи между модулями используются модули, отвечающие за их взаимодействие. Интерфейс системы предназначен для приема и передачи сообщений. При таком подходе имеет смысл заблокировать собственные данные модуля от перезаписи другими компонентами во время его работы.

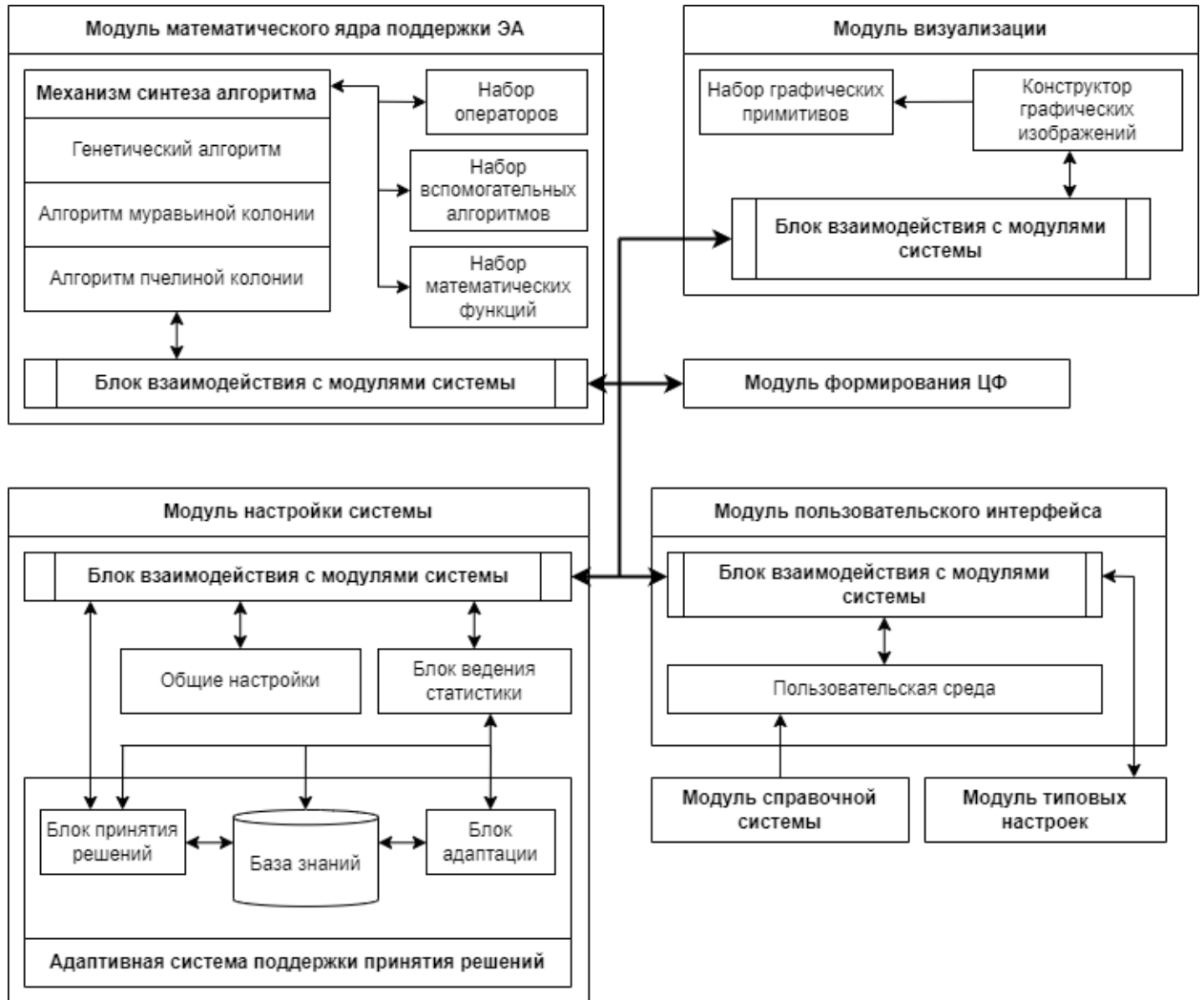


Рисунок 3.1 – Структурная схема АСППР

Также интерфейс не должен работать с другими частями АСППР, т. к. в случае дополнения и модификации системы имеются сложности с изменением и добавлением исходного кода сразу в несколько модулей.

Также необходимо исключить возможности непосредственного взаимодействия интерфейса с остальными элементами АСППР: это связано с проблематичным внесением изменений в код программы при дальнейшей доработке в случае возникновения потребности изменения более чем одного модуля.

Ядро поддержки АСППР занимается обработкой данных, выполнением ЭА и вспомогательными процедурами. К нему выдвигаются следующие требования:

- наличие в составе трех алгоритмов (ГА, АМК и АПК), а также вспомогательных функций;
- обеспечение механизма вызова любого алгоритма по запросу пользователя;
- обеспечение возможности получения и передачи данных другим модулям в составе системы;
- наличие в составе инструментов получения и обработки статистических материалов в ходе работы алгоритмов для пополнения аналитической базы.

Наличие в составе ядра упомянутых выше алгоритмов позволит решать задачи оптимизации с различными требованиями и условиями, находя наиболее приближенный к оптимальному вариант решения. С учетом всех факторов, математическое ядро должно состоять из:

- набора операторов ЭА;
- механизма синтеза ЭА и отработки каждого из них;
- вспомогательных алгоритмов, включая математические функции;
- блока обмена данными с другими модулями.

Настроечный модуль должен отвечать за детализацию настроек системы в целом и ЭА в частности. В число таких настроек входят задание начальных параметров алгоритмов и общие настройки их работы. Также этот модуль предоставляет для пользователя набор рекомендаций, связанных с настройкой каждого алгоритма, опираясь на начальные условия.

Формат экспертной системы с диалоговым окном настройки отлично подходит для реализации описанного функционала. В такой реализации справочная система будет предоставлять необходимую информацию, а база данных будет выступать в роли аналитической базы знаний. Данная база



должна содержать в себе различные преднастроенные конфигурации для каждого алгоритма в соответствии с категориями задач для повышения эффективности, причем подразумевается, что данная база будет пополняться в ходе сбора статистики работы АСППР.

Модуль формирования ЦФ ответственен за создание функции для выбранного алгоритма. Он предназначен для работы в связке с АСППР и обеспечивает обработку введенной пользователем функции и вычисление ее значения. Входными данными являются:

- одна или несколько ЦФ в текстовом виде;
- условия и ограничения, накладываемые на ЦФ;
- массив (матрица) вещественных чисел, представляющий собой набор аргументов функции на текущем шаге алгоритма;
- массив (матрица) вещественных чисел, представляющий собой список дополнительных параметров целевой функции.

Размер первого массива может колебаться от одного до нескольких тысяч, размер второго массива колеблется от нуля до нескольких тысяч.

В качестве выходных данных выступает вещественное число, представляющее собой результат вычисления ЦФ. Обобщенная структура модуля представлена на рисунке 3.2.

Для разработки модуля было рекомендовано спроектировать формальный язык, обеспечивающий возможность задания основных видов ЦФ. Укрупненный алгоритм взаимодействия обработчика ЦФ с АСППР представлен на рисунке 3.3.



Рисунок 3.2 – Обобщенная структура модуля генерации ЦФ

Основные математические функции, поддерживаемые представленным модулем, следующие:

- «+», «-» (как знак числа и как математическая операция), «\*», «/»;
- « $\Sigma$ » (с возможностью задания интервала изменения переменной « $i$ », в том числе в виде « $n$ », « $n+1$ », а также использования нескольких переменных (« $i$ », « $j$ », « $k$ »));
- « $\Pi$ » (аналогично  $\Sigma$ );
- « $\sqrt{\quad}$ » (желательна возможность задания не только квадратного корня);
- «возведение в степень» (в произвольную степень и произвольного числа);
- « $\min$ » и « $\max$ » (с возможностью задания количества сравниваемых элементов по индексам);
- « $\sin$ », « $\cos$ », « $\operatorname{tg}$ », « $\operatorname{ctg}$ » (желательна возможность их вычисления от других функций);

- «*log*» (логарифм);
- дополнительные символы (числа  $\pi$ ,  $e$  и т.д.).

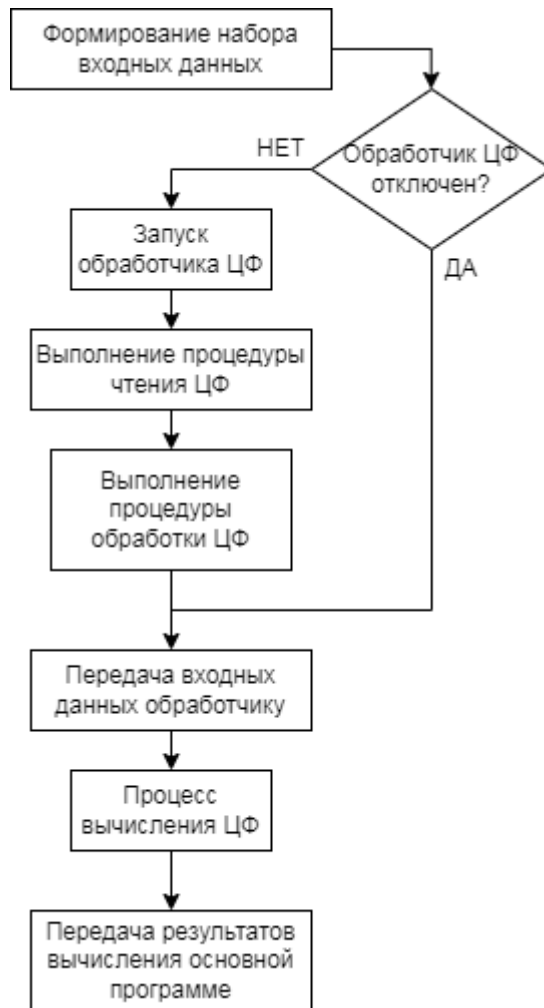


Рисунок 3.3 – Укрупненный алгоритм взаимодействия обработчика ЦФ с АСППР

Взаимодействие с графическим модулем основывается на преобразовании исходных данных для АСППР и результатов работы каких-либо ЭА из данных, понятных машине, в какой-либо графический вид, например, изображение (рисунок), график, блок-схема и т.д. В данном случае подразумевается возможность отрисовки данным модулем графических примитивов.

Модуль работы с документами отвечает за создание входных и выходных данных системы, их сохранение и редактирование. Файл входной информации предполагает содержание следующих данных:

- информацию об алгоритме и его настройках;
- информацию о пространстве решений задачи, а также ее ограничений;
- информацию о целевой функции.

Предполагается, что общие настройки системы располагаются в обособленном файле. В качестве выходных данных выступает решение задачи в символьном либо графическом виде, если результаты возможно визуализировать средствами АСППР. Модуль должен осуществлять обмен информацией только с интерфейсом системы.

Работа модуля справочной системы заключается в предоставлении пользователю информации об эксплуатации АСППР, поддерживаемых ею алгоритмах, особенностях функционирования системы и осуществления настроек для каждого алгоритма в составе системы, а также в предоставлении информации о текущей версии программы.

### **3.2 Описание правил базы знаний для взаимодействия с данными адаптивной системы поддержки принятия решений**

Правила вывода в БЗ представляют собой логические выражения, которые применяются для получения новых знаний или заключений на основе имеющихся данных. Данные правила могут быть записаны в виде логических операций, предоставляемых инструментами используемой системы.

Для использования механизмов базы знаний (как элемента искусственного интеллекта), включающих в себя базу данных решенных задач с указанием использованного алгоритма и результатов решения задачи оптимизации и правила оценивания, был сформирован ряд правил в формате

«ЕСЛИ [условие] ТО [действие]», в основе которых лежит продукционная модель представления знаний. Данный формат является простым с точки зрения формулировки правил и универсальным в том смысле, что с его помощью можно сформулировать ряд последовательных правил, в процессе выполнения которых могут быть произведены сложные операции.

В качестве примеров можно привести следующие правила:

– ЕСЛИ [прерван классический ГА] ТО [использовать другую модификацию ГА]; данное правило срабатывает в случае, если в процессе решения задачи оптимизации был прерван ГА без модификаций; в качестве результата производится замена модификаций ГА на другую из имеющихся в адаптивной системе;

– ЕСЛИ [прерван классический АМК] ТО [использовать другую модификацию АМК]; данное правило срабатывает в аналогичном ключе, но в отношении муравьиного алгоритма;

– ЕСЛИ [прерван классический АПК] ТО [использовать другую модификацию АПК]; данное правило срабатывает в аналогичном ключе, но в отношении пчелиного алгоритма;

– ЕСЛИ [время решения активного алгоритма больше, чем  $T_{calc}$ ] ТО [заменить активный алгоритм]; данное правило срабатывает в случае, если время вычислений активного алгоритма превышает  $T_{calc}$  – время расчета, отведенное системой для получения решения задачи оптимизации и установленное пользователем;

– ЕСЛИ [прерван модифицированный АМК] ТО [определить причину прерывания процедуры решения]; данное правило срабатывает в случае, если в процессе выполнения программы был прерван АМК с примененной к нему модификацией; в качестве результата выступает запуск механизмов определения причины прерывания;

– ЕСЛИ [прерван модифицированный АПК] ТО [определить причину прерывания процедуры решения]; данное правило срабатывает в аналогичном ключе, но в отношении пчелиного алгоритма;

– ЕСЛИ [прерван модифицированный ГА] ТО [определить причину прерывания процедуры решения]; данное правило срабатывает в аналогичном ключе, но в отношении генетического алгоритма;

– ЕСЛИ [запрос на изменение активного алгоритма] ТО [определить активный алгоритм] И [предложить альтернативный алгоритм]; данное правило срабатывает в случае, если системой выполнен запрос на изменение активного алгоритма по какой-либо причине; результатом является определение текущего активного алгоритма и предложение альтернативы ему из имеющихся с усредненными настройками;

– ЕСЛИ [выбран активный алгоритм] ТО [выбрать модификацию активного алгоритма]; данное правило срабатывает, если для проведения расчетов пользователем или системой выбран активный алгоритм; результатом является подбор доступных модификаций для этого алгоритма и предложение пользователю использования одной из них;

– ЕСЛИ [не выбран активный алгоритм] ТО [выбрать алгоритм]; данное правило срабатывает, если пользователем или системой не установлен активный алгоритм (это относится к ситуации, когда идет подготовка к расчетам); результатом является обращение системы к БД, поиск схожих по параметрам задач, их решений и использованных алгоритмов, и предложение на основе этих данных наиболее подходящего алгоритма с точки зрения качества полученных решений;

– ЕСЛИ [получен сигнал от ТС о невозможности продолжать маршрут] ТО [убрать ТС из списка доступных]; данное правило срабатывает, если конкретное ТС не в состоянии продолжать движение по какой-либо причине; результатом является исключение данного ТС из списка доступных и помещение в отдельный список;

– ЕСЛИ [получен сигнал от ТС о возможности работать] ТО [вернуть ТС в список доступных]; данное правило срабатывает, если помещенное в отдельный список ТС может вернуться к работе; результатом является возвращение ТС в основной список с возможностью дальнейшего использования при расчетах.

### **3.3 Структура работы модулей эволюционных алгоритмов на примере алгоритма муравьиной колонии**

В момент запуска системы формируется колония из определенного количества муравьев. Основную роль в системе играет цикл, ограниченный заданным числом итераций, в теле которого идет проход по всей колонии [29].

После каждого своего шага, муравей формирует феромон  $f_\varphi$ , сила которого формируется в соответствии с формулой

$$f_\varphi = \frac{v}{b} c, \quad (3.1)$$

где  $v$  – полученное муравьем значение ЦФ,  $b$  – последний лучший результат в системе на момент вычисления,  $c$  – коэффициент, полученный из конфигурации системы для стимуляции силы феромонов в случае получения слишком маленького значения функции.

В начале работы функции «*gип*» вызывается вероятностная функция для определения вектора движения муравья, в соответствии с которым на протяжении нескольких итерационных циклов он будет двигаться в сторону одного из феромонов или «по своему усмотрению».

Принцип работы вероятностной функции реализован следующим образом.

Вероятности интенсивностей всех «чужих» феромонов в системе распределены в пределах от нуля до разницы между единицей и шансом муравья выдвигаться по своему усмотрению согласно формуле

$$p(\varphi) = \frac{f_\varphi}{\sum_{i=0}^{\bar{n}_a} f_{\varphi_n}} (1 - \delta), \quad (3.2)$$

где  $f_\varphi$  – сила полученного в качестве аргумента функции феромона,  $\delta$  – шанс муравья пойти по своему усмотрению, которое является статическим значением из конфигурации системы,  $\bar{n}$  – количество «чужих» для муравья феромонов.

Вероятностная функция формируется следующим образом:

$$f = \begin{array}{l|l} \varphi_1 & i \in [0; p(\varphi_1)), \\ \varphi_2 & i \in [p(\varphi_1); p(\varphi_2)), \\ \dots & \dots \\ \varphi_n & i \in [p(\varphi_{n-1}); p(\varphi_n)), \\ \emptyset & i \in [1 - \delta; 1], \end{array} \quad (3.3)$$

где  $\varphi_j$  – экземпляр класса  $j$ -го феромона,  $i$  – сгенерированное случайное вещественное значение в пределах  $[0;1]$ ,  $\emptyset$  – отсутствие феромона, т. е. муравей будет двигаться по своему усмотрению.

Вероятностная функция (2.3) получает на вход случайное, равномерно распределенное значение в пределах  $[0;1]$  и возвращает либо феромон, либо пустое значение, что будет означать выбор в пользу движения «по своему усмотрению».

Муравей формирует вектор следующего шага самостоятельно в зависимости от своей целевой точки. В случае наличия феромона, принцип вычисления приведен в соответствии с формулой

$$\bar{c}_a = |c_\varphi - c_a| \cdot \omega, \quad (3.4)$$

где  $\bar{c}_a$  – новые координаты местоположения муравья,  $c_\varphi$  – координаты местоположения феромона,  $c_a$  – прежние координаты местоположения муравья,  $\omega$  – коэффициент передвижения (скорость муравья), генерируемый случайным образом и имеющий равномерное распределение.

После движения муравей получает новые значения своих координат, которые становятся новыми аргументами ЦФ, производится вычисление значения ЦФ. Если данное значение является лучшим, то значения



аргументов функции и само значение записываются в тело класса как «лучший результат».

Система состоит из трех основных классов: Ant, AntColonyAlgorithm и Pheromone. UML-диаграмма классов системы представлена на рисунке 3.4.

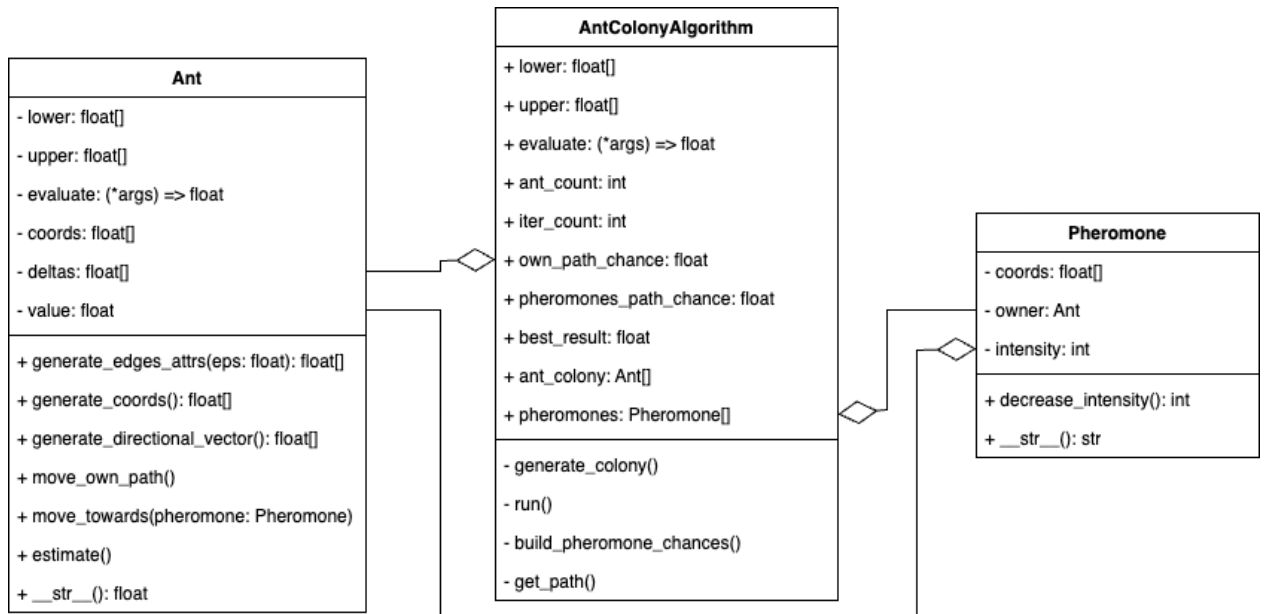


Рисунок 3.4 – UML-диаграмма классов

Класс Ant формирует собой субъект муравья, содержащего в себе текущее местоположение в виде вектора значений, ограничивающих условия задачи, значение ЦФ на момент работы и функции для передвижения как «по своему усмотрению», так и к определенному феромону. Кроме того, в теле класса муравья производится вычисление координат следующего шага при передвижении.

Феромон, с точки зрения работы алгоритма, представляет собой специальное химическое вещество, которое муравьи откладывают на своем пути [83]. Когда муравей выбирает направление движения, он учитывает не только свои собственные предпочтения в поиске кратчайшего пути, но и опыт других муравьев. Он получает этот опыт (информацию о нем) непосредственно через уровень феромонов, оставленных на каждом пути.

Отсюда следует, что концентрация феромона служит показателем желания муравья выбрать определенный путь: муравьи, следуя за следами других, могут принимать более информированные решения и находить оптимальные пути для общей пользы.

В системе за работу феромонов отвечает класс `Pheromone`, внутри которого хранится информация о «хозяине» и сила воздействия, которая декрементируется с каждым шагом итерации.

Основные расчеты системы производятся в классе `AntColonyAlgorithm`, который служит для прямого диалога с пользователем и содержит в себе ЦФ, множество конфигурационных переменных (количество итераций, размер колонии, шанс выбора муравьем собственного пути и лучший результат вычисления ЦФ в колонии), числовые ограничения задачи, а также элементы, отвечающие за построение и вычисление вероятностной функции выбора муравьем следующего шага пути, генерация колонии и основная функция запуска алгоритма.

Схема алгоритма работы функции «run» приведена на рисунке 3.5. В теле функции инициализируется колония муравьев и входные значения: ЦФ, ограничения, численность колонии и т. д.

В процессе итераций выполняется процедура обхода по всем муравьям сформированной колонии, в ходе которой производится выбор пути между одним из имеющихся феромонов или «собственным». После произведения шага на случайную величину по одному из путей вычисляется значение ЦФ из значений, присущих муравью.

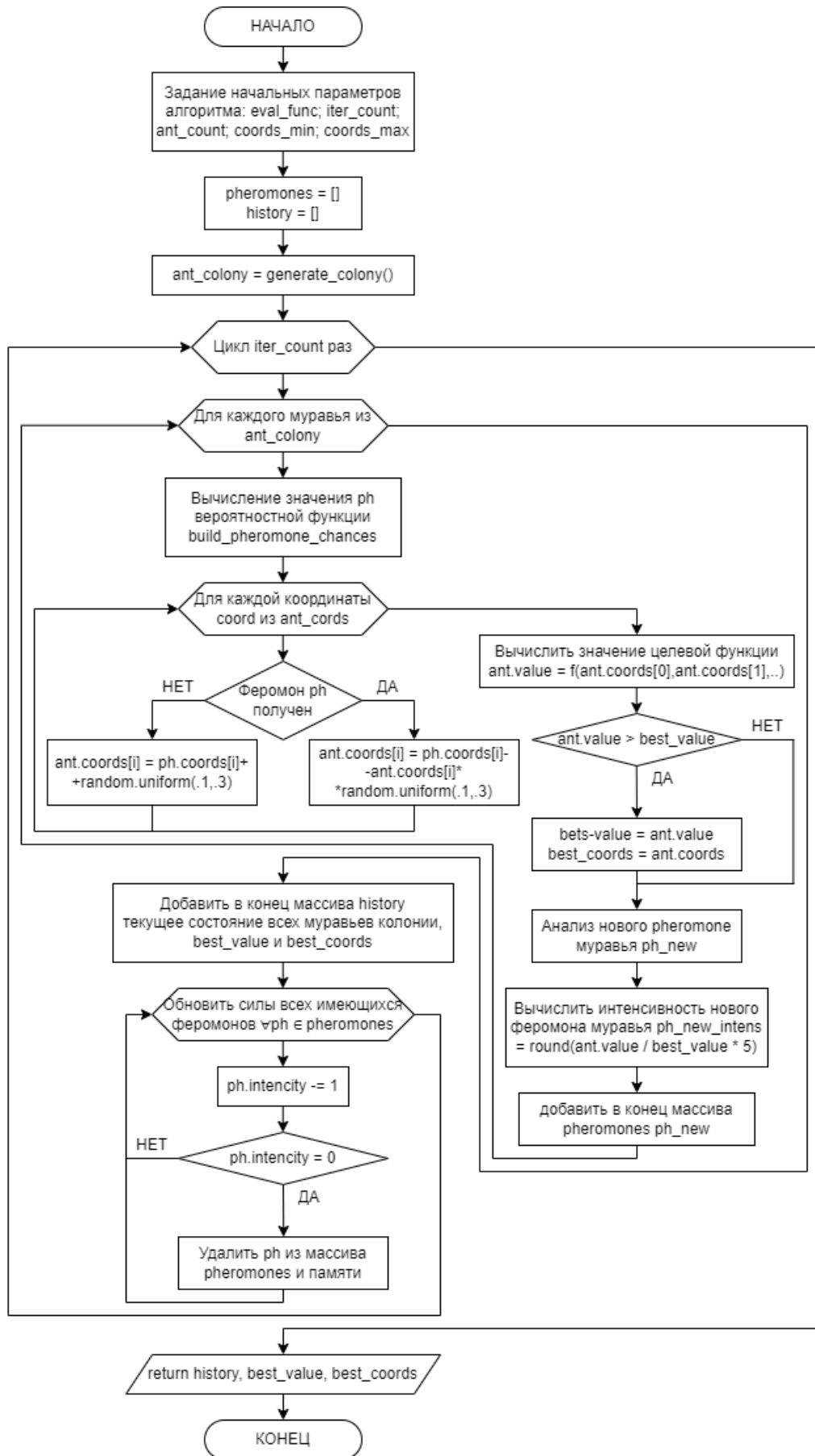


Рисунок 3.5 – Схема алгоритма работы функции «run»

После этого выполняется сравнение полученных результатов с лучшими на момент вычисления зафиксированными результатами в системе. В случае если муравей превзошел лучший результат, системой выполняется обновление значения.

По завершении обхода муравьи инициализируют новый феромон, который получает координаты каждого муравья на момент инициализации и интенсивность – соотношение полученного результата к лучшему. За пределами прохода по муравьям значения каждого феромона в системе обновляется посредством декрементации значения интенсивности.

В качестве тестовых запусков были поставлены следующие задачи: вычислить максимальное значение функции и точки для функций

$$f(x, y) = \frac{3x^2 + 2xy + 4y^2}{x + y} \rightarrow \max \quad (3.5)$$

и

$$f(x, y) = \cos(xy) + e^{-(x^2+y^2)} + \frac{1}{(x-1)^2 + (y-2)^2 + 0.1} \rightarrow \max \quad (3.6)$$

на промежутке  $[-5, 5]$  для каждой переменной. Визуализация работы муравьиного алгоритма по каждой из обозначенных задач в формате отдельного приложения представлена на рисунках 3.6 и 3.7 соответственно.

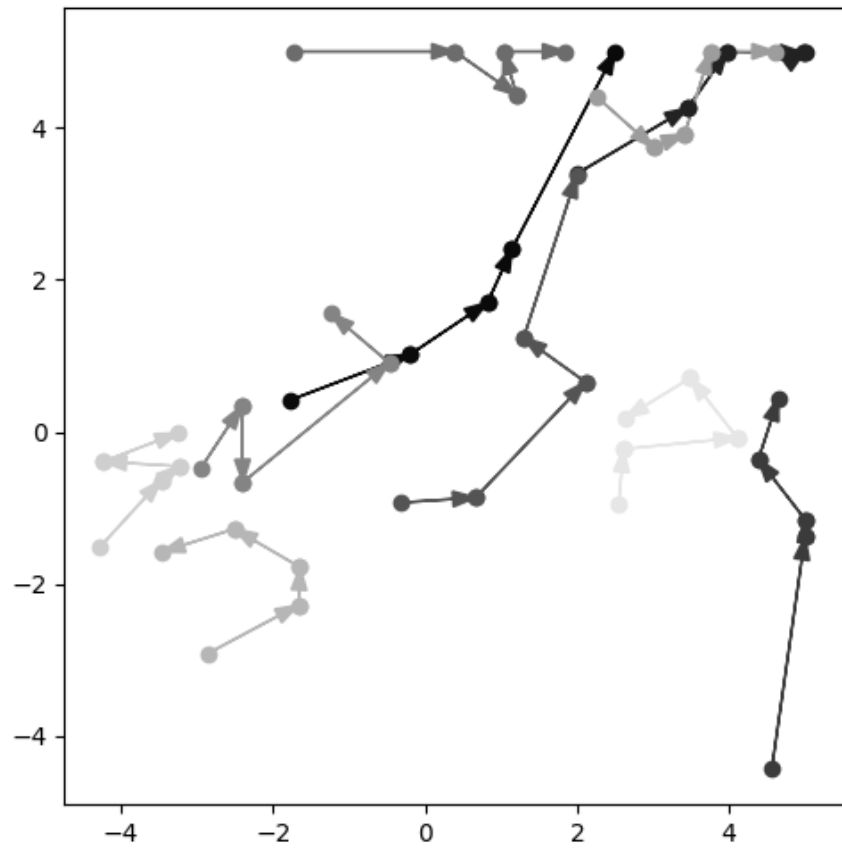


Рисунок 3.6 – Визуализация работы муравьиного алгоритма для функции (3.5)

Отдельно можно отметить, что и в первом, и во втором случае колония в большинстве своем движется в область, расположенную в квадрате  $([4, 4], [5, 5])$ , что подразумевает расположение в этом квадрате удовлетворительных значений для обеих функций.

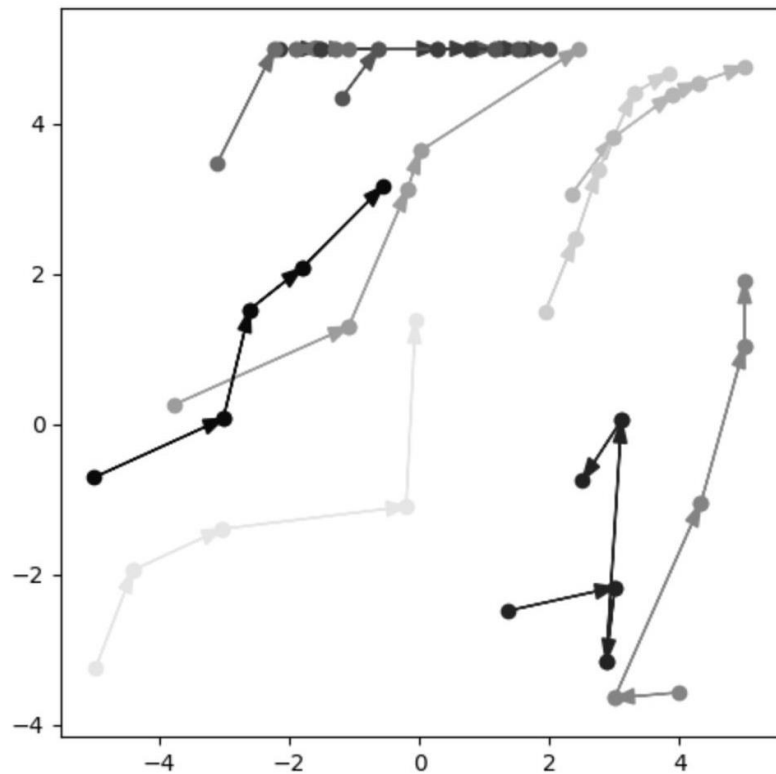


Рисунок 3.7 – Визуализация работы муравьиного алгоритма для функции (3.6)

### 3.4 Формирование пользовательского интерфейса адаптивной системы

Ниже приведены макеты пользовательского интерфейса для адаптивной системы поддержки принятия решений (рисунки 3.8-3.11).

На основе данных макетов был спроектирован интерфейс пользователя для адаптивной системы. На рисунке 3.12 представлено рабочее пространство (интерфейс пользователя) разработанного программного средства «Адаптивная Система Поддержки Принятия Решений» (АСППР). Оно включает в себя область настройки условий и ограничений многокритериальной задачи, визуальное представление, а также журнал событий, в котором отображаются итерации (шаги) решений в хронологическом порядке. На данном рисунке также можно увидеть пример

отображения ошибки расчета в журнале событий, результатом которого станет переключение активного алгоритма.

Адаптивная Система Поддержки Принятия Решений					
<b>Настройки</b>		<input type="button" value="Импорт из файла"/>		<input type="button" value="Сброс всех настроек"/>	
<b>Аппаратные</b>					
Ограничения по числу ядер CPU	<input type="text" value="10"/>			<input type="checkbox"/> Использовать ускорение графического процессора	
Ограничения по ОЗУ (мб)	<input type="text" value="4096"/>				
Максимальное время простоя (с)	<input type="text" value="60"/>			<input type="button" value="Сброс"/>	
<b>Импорт/экспорт в файл</b>					
Способ сохранения в файл		<input checked="" type="radio"/> Поточковый		<input type="radio"/> Статичный	
				<input type="button" value="Сброс"/>	
<b>Муравьиный алгоритм</b>					
Число особей	<input type="text" value="100"/>			<input checked="" type="checkbox"/> Включен	
Коэффициент испарения феромонов	<input type="text" value="0.1"/>			<input type="checkbox"/> Авто	
Мин. число итераций перед заменой	<input type="text" value="5"/>	Макс. число итераций перед заменой	<input type="text" value="10"/>		
				<input type="button" value="Сброс"/>	
<b>Генетический алгоритм</b>					
Число особей	<input type="text" value="100"/>			<input checked="" type="checkbox"/> Включен	
Коэффициент мутации	<input type="text" value="0.4"/>	Шанс мутации	<input type="text" value="0.4"/>	<input type="checkbox"/> Авто	
Способ отбора	<input type="text" value="Турнирный"/>	Способ скрещивания	<input type="text" value="Авто"/>		
Мин. число итераций перед заменой	<input type="text" value="5"/>	Макс. число итераций перед заменой	<input type="text" value="10"/>		
				<input type="button" value="Сброс"/>	
<b>Пчелиный алгоритм</b>					
Число разведчиков	<input type="text" value="100"/>			<input checked="" type="checkbox"/> Включен	
Число рабочих	<input type="text" value="100"/>			<input type="checkbox"/> Авто	
Число наблюдателей	<input type="text" value="100"/>				
Мин. число итераций перед заменой	<input type="text" value="5"/>	Макс. число итераций перед заменой	<input type="text" value="10"/>		
Вероятность отказа от источника	<input type="text" value="5"/>	Лимит оставления от источника	<input type="text" value="10"/>		
				<input type="button" value="Сброс"/>	

Рисунок 3.8 – Макет окна общих настроек решения и эволюционных алгоритмов

**Адаптивная Система Поддержки Принятия Решений**

**1** Выбор задачи

Навигационная    Раскря

**2** Выбор используемого алгоритма

Генетический    Муравьиный    Имитации отжига

**3** Выбор последовательности выбора алгоритмов

1 Муравьиный	Авто
2 Генетический	Генетический
3 Муравьиный	Муравьиный
4 Пчелиный	Пчелиный
5 Авто	
6 Генетический	

**4** Задание постановки задачи или [загрузка постановки из файла](#)

Наименование	Длина	Ширина	Количество
Материал	400	800	5000
Деталь А	3	4	

[Добавить деталь](#)

[Загрузить параметры из файла](#)    [Открыть настройки](#)

При выборе задачи коммивояжера

**4** Задание постановки задачи или [загрузка постановки из файла](#)

Таблица расстояний    Карта

	A	B	C	D	E
A		3	1	5	1
B	3		7	2	5
C	1	7		3	6
D	5	2	3		14
E	1	5	6	14	
G	3	4	3	1	3

Рисунок 3.9 – Макет окна формирования общих настроек задачи

**Адаптивная Система Поддержки Принятия Решений**

Задача: Коммивояжера    Способ представления: Таблица расстояний

**Условия**

	A	B	C	D	E
A		3	1	5	1
B	3		7	2	5
C	1	7		3	6
D	5	2	3		14
E	1	5	6	14	
G	3	4	3	1	3

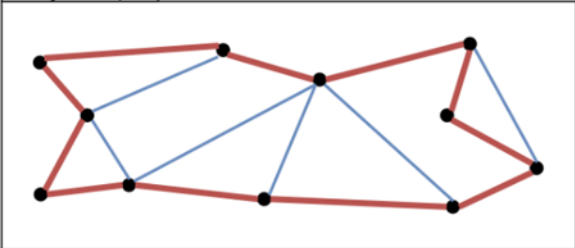
**Заданные точки**

A  
B  
C  
D  
E  
G

**Ограничения**

Выбрать: Без повторов, Макс расстояние, D2C2, B1C1, C2D2, C1D1, Если C2D2 > 20 То C3D4 < 10, C2D2, Коэффициент расстояния, C1D1

**Визуализация**    Тепловая карта



**Журнал**

Шаг 99, алгоритм переключён на генетический с 100 особями, соревновательным отбором и коэффициентом мутации 0.1  
Шаг 100...

**Решение**

Рейтинг оптимальных передвижений:  
1) A - D - B - C - E - G - A  
2) B - D - A - C - E - G - B

Рисунок 3.10 – Макет окна визуализации решения задачи коммивояжера





Рисунок 3.11 – Макет окна визуализации решения задачи раскрой

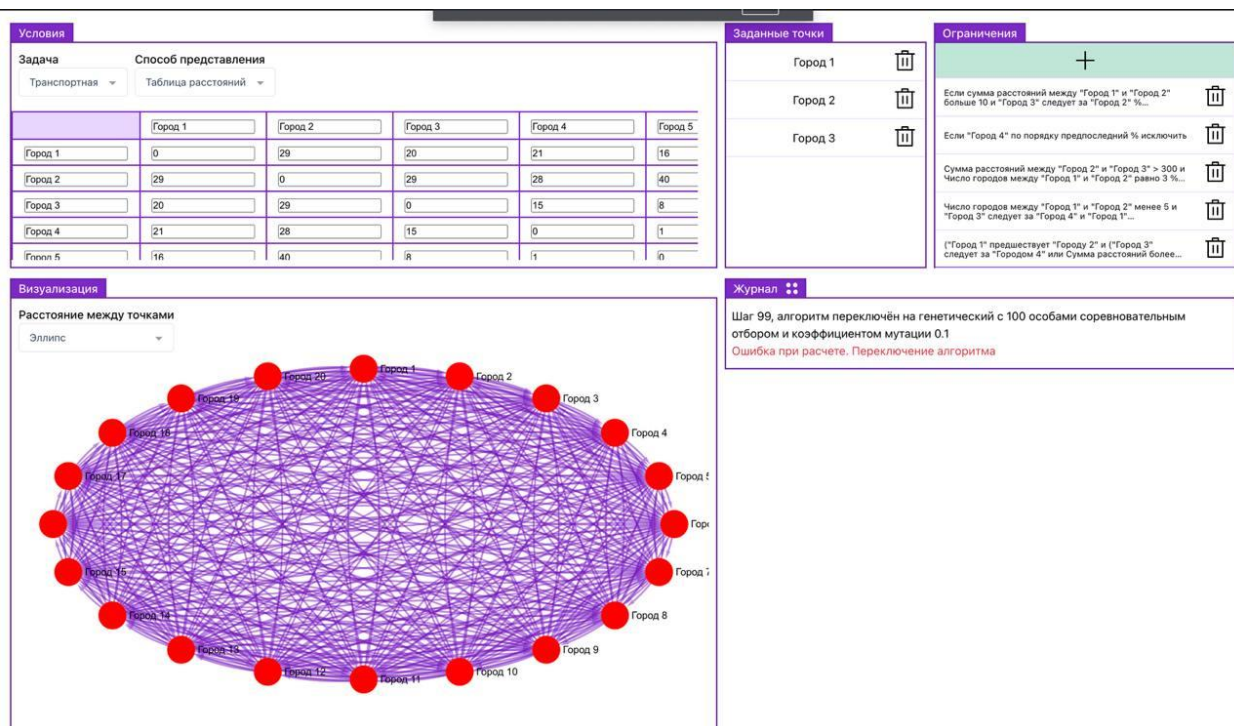


Рис. 3.12 – Рабочее пространство (интерфейс) АСППР

Представленная адаптивная система была реализована с использованием языка программирования Python [38, 42, 44, 98, 113].

Данный язык является современным и мощным инструментом разработки, при этом относительно простым в изучении и имеющим самую большую библиотеку вспомогательных решений.

Выгрузка данных о населенных пунктах осуществляется посредством обращения к базе данных, где хранятся названия пунктов и координаты их расположения (широта и долгота), на основании которых при помощи формулы евклидова расстояния рассчитывается расстояние между ними.

На рисунке 3.13 приводится пример содержимого результирующего файла обработчика, где отображаются матрица значений (маршрутов), матрица временных окон, характеристики наиболее результативного и наиболее быстрого алгоритмов и значения полученных решений.

	A	B	C	D	E	F	G	H
1	matrix	time_matrix	most_effective_result_algo	most_effective_result_fitness	most_effec	most_fast_result_algo	most_fast_result_fitness	most_fast_re
1530	(0.0, 7116.0, 8437.0, 4092.0)	[[ 224, 288, 400, 576, 667, 4103	[Type: 'AC', alpha: 1.5, beta: 0.75, q: 1.25, p: 0.75]	55817.412871437125	1.822	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	89101.02116855264	0.022
1531	(0.0, 4913.0, 5321.0, 3489.0)	[[ 1200, 1600, 2028, 2000, 12	[Type: 'AC', alpha: 1.5, beta: 1.25, q: 1, p: 0.5]	12188.07688295052	1.184	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	28268.58334495266	0.018
1532	(0.0, 2000.0, 14142.0, 2000)	[[ 4000, 1000, 14000, 16000, 1	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	60128.99029445196	1.418	[Type: 'BC', workers_part: 0.75, regenerate_func: 'swap_indexes']	82914.0519750461	0.020
1533	(0.0, 3893.0, 4247.0, 1490.0)	[[ 2000, 4000, 6000, 8000, 1000	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	11474.058920627066	1.578	[Type: 'BC', workers_part: 0.75, regenerate_func: 'reverse_element']	15840.023442848196	0.022
1534	(0.0, 12649.0, 4472.0, 2690.0)	[[ 0, 176, 304, 432, 544, 656, 880	[Type: 'GA', p_mutation: 0.6, select_func: 'tournament', mutate_func: 'swap_indexes']	134234.11564373976	0.539	[Type: 'BC', workers_part: 0.75, regenerate_func: 'swap_indexes']	270969.6207492318	0.022
1535	(0.0, 224.0, 288.0, 400.0, 57	[[ 40, 79, 119, 158, 198, 238, 27	[Type: 'GA', p_mutation: 0.2, select_func: 'tournament', mutate_func: 'swap_indexes']	18845.574156849545	2.216	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	28160.760689816787	0.022
1536	(0.0, 36.6, 57.0, 25.0, 51.0)	[[ 0, 51, 57, 72, 92, 114, 137, 181	[Type: 'GA', p_mutation: 0.4, select_func: 'roulette', mutate_func: 'swap_indexes']	679.039406509786	2.213	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	1265.538822735355	0.029
1537	(0.0, 40.0, 79.0, 119.0, 158	[[ 0, 200, 4000, 6000, 8000, 1000	[Type: 'GA', p_mutation: 0.2, select_func: 'stochastic', mutate_func: 'reverse_elements']	1188.8600000000002	0.097	[Type: 'BC', workers_part: 0.75, regenerate_func: 'swap_indexes']	3011.9599999999995	0.027
1538	(0.0, 51.0, 57.0, 72.0, 92.0)	[[ 40, 79, 119, 1622, 1698, 1699	[Type: 'GA', p_mutation: 0.2, select_func: 'tournament', mutate_func: 'reverse_elements']	691.3421095729984	0.236	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	1362.1076079945215	0.021
1539	(0.0, 1172.0, 718.0, 1385.0)	[[ 0, 718, 768, 333, 112, 91, 394, 1	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	5378.06003340312	0.687	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	8956.11231750554	0.018
1530	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 4765, 2679, 4782, 5036, 821	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	44000.0	1.099	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	80000.0	0.017
1531	(0.0, 41.0, 314.0, 274.0, 234	[[ 1184, 7186, 3999, 6994, 104	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 1.0]	2436.0783868602385	0.182	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	4033.155070614985	0.022
1532	(0.0, 176.0, 304.0, 432.0, 54	[[ 0, 2000, 4000, 6000, 8000, 1000	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	4064.0	3.259	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	5465.0	0.017
1533	(0.0, 12000.0, 16000.0, 282	[[ 2008, 4862, 7698, 870, 4735	[Type: 'AC', alpha: 1.5, beta: 1.5, q: 1.25, p: 1.25]	54128.34420254716	3.132	[Type: 'BC', workers_part: 0.75, regenerate_func: 'reverse_element']	92304.4512128015	0.021
1534	(0.0, 2068.0, 4862.0, 7698.0)	[[ 2000, 4000, 6000, 8000, 1000	[Type: 'AC', alpha: 1.5, beta: 1.5, q: 1.25, p: 1.25]	17998.811027797405	1.863	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	57605.34673590766	0.017
1535	(0.0, 5962.0, 3228.0, 434.0)	[[ 0, 9132, 3842, 7639, 8741, 8729	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 1, p: 0.75]	16836.051015681423	2.009	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	40570.01577285419	0.022
1536	(0.0, 4000.0, 6000.0, 14142.0)	[[ 2000, 16122, 12186, 10158, 2	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 1, p: 0.75]	46472.13359499598	1.757	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	77012.16448024174	0.017
1537	(0.0, 2000.0, 4000.0, 12000)	[[ 0, 4000, 14142, 12186, 2000, 4	[Type: 'AC', alpha: 1.5, beta: 1.5, q: 1.25, p: 1.25]	69410.9625604668	3.406	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	150921.9004049566	0.022
1538	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 5962, 3228, 434, 1579, 5895	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	42828.42712474619	2.055	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	52094.28325545101	0.021
1539	(0.0, 9132.0, 3842.0, 7639.0)	[[ 0, 3144, 9254, 6798, 14646, 135	[Type: 'AC', alpha: 1.5, beta: 0.75, q: 1.25, p: 0.75]	59070.12884651917	1.974	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	101476.20304604607	0.021
1540	(0.0, 4000.0, 10000.0, 14000)	[[ 0, 4000, 6000, 8000, 10000, 140	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	47416.40784699875	2.883	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	90256.4241831682	0.017
1541	(0.0, 718.0, 768.0, 353.0, 11	[[ 0, 25, 51, 76, 102, 127, 152, 178	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 1, p: 0.75]	1760.322670705657	0.752	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	2024.5141963360158	0.021
1542	(0.0, 4000.0, 14142.0, 2184.0)	[[ 0, 6588, 5505, 10397, 6066, 893	[Type: 'AC', alpha: 1.5, beta: 1.5, q: 1.25, p: 1.25]	75934.0682180759	2.705	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	131101.51244607079	0.022
1543	(0.0, 2000.0, 16122.0, 12186.0)	[[ 182, 71, 455, 212, 62, 212, 3	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 1, p: 0.75]	42246.1122523521	2.406	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	96791.86889739791	0.017
1544	(0.0, 25.0, 51.0, 78.0, 102.0)	[[ 0, 1269, 1321, 3129, 921, 115	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	95200.00000000007	1.082	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	762.0000000000009	0.017
1545	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 64, 208, 320, 416, 480, 816, 9	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	48000.0	1.207	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	84000.0	0.017
1546	(0.0, 1269.0, 1122.0, 31.0, 2	[[ 0, 40, 78, 116, 154, 193, 233, 27	[Type: 'AC', alpha: 1.5, beta: 0.75, q: 1.25, p: 0.75]	4782.007505054798	0.848	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	8892.003545252624	0.018
1547	(0.0, 40.0, 78.0, 116.0, 154.0)	[[ 0, 4000, 6000, 8000, 10000, 140	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	638.1089372633888	0.744	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	687.6629454520026	0.018
1548	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 3641, 3085, 1012, 924, 89, 6	[Type: 'GA', p_mutation: 0.8, select_func: 'roulette', mutate_func: 'reverse_elements']	42957.17329231446	0.093	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	81683.12088384605	0.022
1549	(0.0, 3144.0, 9254.0, 6798.0)	[[ 2000, 4000, 6000, 8000, 1000	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	65570.4000717056	0.758	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	103950.7552171484	0.021
1550	(0.0, 3641.0, 3085.0, 1012.0)	[[ 0, 2000, 4000, 6000, 12166, 101	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 0.5]	42045.8545421237	1.872	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	115605.73068587286	0.021
1551	(0.0, 4765.0, 2679.0, 4782.0, 5036.0)	[[ 77, 155, 163, 117, 193, 227, 2	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 1, p: 0.75]	12900.031454822945	1.158	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	25636.00618834765	0.018
1552	(0.0, 1184.0, 718.0, 353.0, 1099.0)	[[ 0, 2000, 4000, 6000, 8000, 1000	[Type: 'GA', p_mutation: 0.4, select_func: 'tournament', mutate_func: 'swap_indexes']	66212.40705382131	0.813	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	108260.36145362927	0.022
1553	(0.0, 40.0, 79.0, 119.0, 162.0)	[[ 0, 571, 585, 597, 609, 621, 633	[Type: 'GA', p_mutation: 0.8, select_func: 'tournament', mutate_func: 'swap_indexes']	3655.9562826498399	0.246	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	4658.824726181482	0.022
1554	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 12, 851, 863, 875, 887, 899, 9	[Type: 'AC', alpha: 1.5, beta: 0.5, q: 0.75, p: 1.0]	80000.0	2.441	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	178000.0	0.022
1555	(0.0, 4000.0, 6000.0, 8000.0)	[[ 0, 2712, 513, 2995, 1989, 2793	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 0.75, p: 1.0]	80000.0	0.685	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	92000.0	0.018
1556	(0.0, 573.0, 585.0, 597.0, 60	[[ 0, 2000, 6000, 8000, 12166, 101	[Type: 'GA', p_mutation: 0.2, select_func: 'tournament', mutate_func: 'reverse_elements']	1501.67913329713	0.235	[Type: 'BC', workers_part: 0.9, regenerate_func: 'swap_indexes']	2305.697381005044	0.022
1557	(0.0, 6588.0, 5505.0, 10397.0)	[[ 0, 8737, 5147, 1839, 4725, 3662	[Type: 'GA', p_mutation: 0.2, select_func: 'stochastic', mutate_func: 'reverse_elements']	60904.70989243149	0.255	[Type: 'BC', workers_part: 0.75, regenerate_func: 'swap_indexes']	107868.70586274177	0.028
1558	(0.0, 2000.0, 4000.0, 6000.0)	[[ 0, 96, 190, 352, 464, 539, 583, 8	[Type: 'GA', p_mutation: 0.4, select_func: 'tournament', mutate_func: 'swap_indexes']	48975.201455727285	2.144	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	117568.31361709448	0.022
1559	(0.0, 4173.0, 2310.0, 868.0, 11	[[ 0, 220, 1384, 5003, 6413, 9237	[Type: 'AC', alpha: 1.25, beta: 0.5, q: 1, p: 0.75]	11962.02334282654	0.811	[Type: 'BC', workers_part: 0.9, regenerate_func: 'reverse_element']	17116.02602396027	0.019

Рис. 3.13 – Содержимое результирующего файла обработчика

Данный файл имеет формат .csv, однако его также можно открыть для просмотра в формате .xlsx для обзора с помощью Microsoft Office Excel.

### 3.5 Выводы

В данной главе описана структура адаптивной системы поддержки принятия решений при многокритериальной оптимизации с предоставлением следующих результатов.

1. Разработана структура адаптивной системы поддержки принятия решений при многокритериальной оптимизации транспортной задачи, в основе вычислительного инструмента которой заложены модифицированные эволюционные алгоритмы (ГА, АМК, АПК), а также механизм их переключения в процессе решения задачи.

2. Разработаны механизмы и проведена модификация эволюционных алгоритмов в адаптивной системе и правила базы знаний для взаимодействия с данными о решенных задачах оптимизации, позволяющие ускорить процесс решения многокритериальной транспортной задачи.

3. Проиллюстрирована структура работы модулей эволюционных алгоритмов на примере алгоритма муравьиных колоний.

4. Разработаны макеты адаптивной системы поддержки принятия решений, иллюстрирующие основные окна с настройками и вариантами визуализации результатов решений в виде долей, тепловой карты или схематичного графа.

5. Приведено разработанное рабочее пространство (интерфейс пользователя) адаптивной системы поддержки принятия решений, включающее в себя область настройки условий и ограничений многокритериальной задачи, визуальное представление, а также журнал событий, в котором отображаются итерации (шаги) решений в хронологическом порядке.

## ГЛАВА 4. РЕАЛИЗАЦИЯ АДАПТИВНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

### 4.1 Апробация разработанной адаптивной системы поддержки принятия решения при многокритериальной оптимизации с использованием эволюционных алгоритмов

Для проведения апробации разработанной адаптивной системы поддержки принятия решений было решено провести ряд вычислительных экспериментов.

Целью экспериментов является сравнение эффективности различных параметрических вариаций АМК, АПК и ГА при решении многокритериальной транспортной задачи.

Изначальная численность постановок составляет 28025. Постановки представляют собой квадратные матрицы размерностью от  $10 \times 10$  до  $20 \times 20$ . Численность исходных и полученных после проведения расчетов матриц приведена в таблице 4.1. Матрицы были получены путем парсинга файлов из открытого источника [9]. В результате были получены 95 постановок, начиная с размерности с 17 точек и заканчивая размерностью в 85900 точек. Полученные в виде матриц постановки были разбиты на множество подматриц размерностью от 10 до 19 точек включительно, как показано в формуле

$$B = A_{i,i+j:i,i+j} \mid i \in [0; S - 20); j \in [0; 20); i, j \in Z, \quad (4.1)$$

где  $A$  – полученная путем парсинга матрица,  $S$  – размерность матрицы по одной «оси».

Таблица 4.1 – Численность матриц, использованных в вычислительных экспериментах

Количество точек	Количество матриц	
	До расчетов	После расчетов
10	2745	2698
11	2760	2758
12	2777	2764
13	2798	2691
14	2781	2742
15	2773	2712
16	2796	2788
17	2766	2893
18	2931	2857
19	2898	2771
20	1975	1964
Всего	30000	29638

Для каждого алгоритма (включая модификации) были приняты следующие условия прерывания расчетов:

- достижение предельного количества итераций;
- недостижение ни одним из акторов локального решения задачи;
- полное уничтожение всех особей (например, в смене поколений);
- иные исходы, приведшие к ошибкам работы алгоритмов.

В таблице 4.2 приведены общие значения, использованные для параметров ГА, АМК и АПК в рамках вычислительных экспериментов.

Таблица 4.2 – Значения общих параметров алгоритмов

Параметр	Значение
Численность итераций	60
Численность особей (акторов)	60
Численность предлагаемых решений	1
Вероятность ( $P$ )	0.7

Смежные матрицы содержали между собой расстояния между городами в общем диапазоне [223;15000]. На рисунке 4.1 приводится график распределения средних значений матриц смежности.

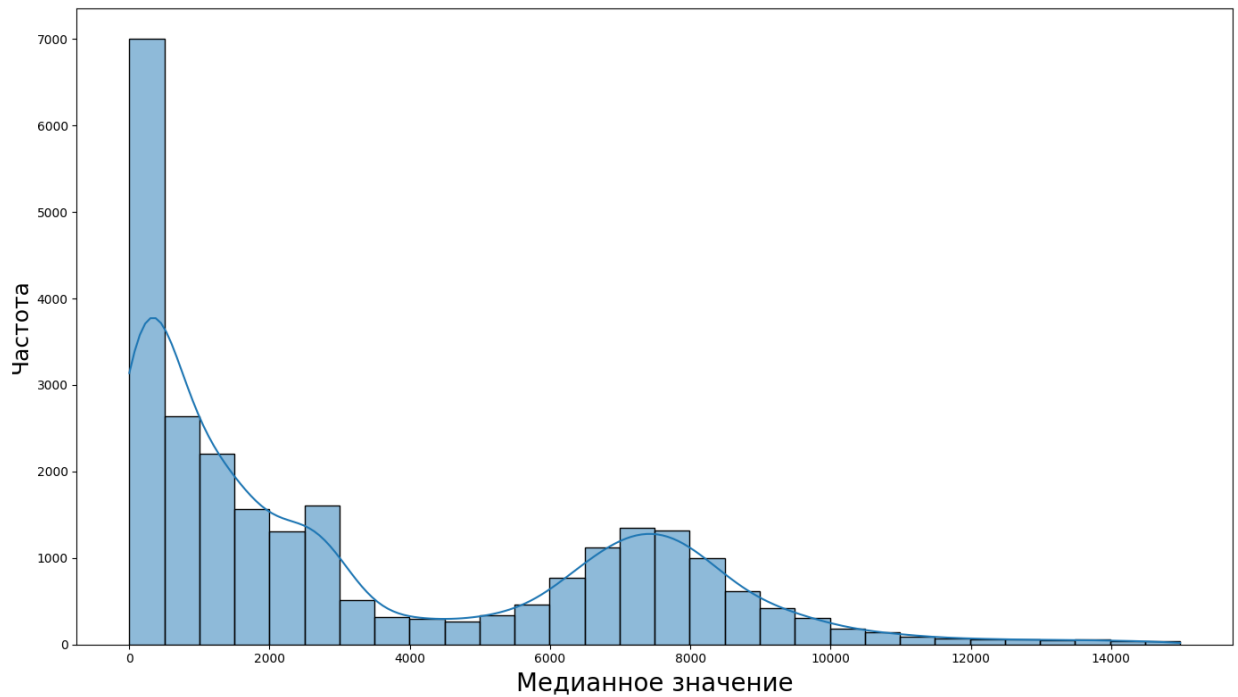


Рис. 4.1 – Распределение средних значений матриц смежности

#### 4.1.1 Многокритериальная транспортная задача без ограничений

За основу была принята постановка МТЗ из п. 2.2, но без применения к ней ограничений. Общее число сформулированных задач было сведено к 10000, каждая из которых характеризовалась квадратными матрицами с размерностями в диапазоне  $[10, 20]$ . Для проведения вычислительного эксперимента использовались программные реализации ГА, АМК и АПК с рядом заданных параметров. Вариации алгоритмов с параметрами приведены в таблице. Каждый из приведенных алгоритмов выполнял 60 итераций с 60 особями.

Использованные вариации алгоритмов в виде использованных параметрических значений приводятся в таблице 4.3. В данной таблице можно увидеть следующее:

– для АМК коэффициенты принимают значения  $\alpha \in [0.25, 1.5]$  с шагом 0.25,  $\beta \in [0.5, 1.25]$  с шагом 0.25,  $p \in [0.75, 1.25]$  с шагом 0.25,  $q \in [0.5, 1.25]$  с шагом 0.25;

– для АПК доля пчел-работниц колонии находится в диапазоне  $[0.15, 0.9]$  с шагом 0.1 и 0.15 для каждого из двух обозначенных методов поиска нового источника;

– для ГА шанс мутации располагается в диапазоне  $[0.2, 0.6]$  с шагом 0.2 для методов мутации «смена индекса» и «реверс подмножества» и методов отбора типов «рулетка», «стохастический» и «турнирный».

На рисунке 4.2 приведена диаграмма, отображающая, какой алгоритм для какого количества задач показал лучший результат среди всех. В целях предотвращения «перегрузки» графика на данной и последующих диаграммах приводятся первые 20 модификаций в порядке убывания качества иллюстрируемых результатов. Каждый алгоритм в приведенных ниже диаграммах имеет сокращения (муравьиный – МУР, пчелиный – ПЧЕ, генетический ГЕН).

Таблица 4.3 – Параметрические значения вариаций алгоритмов для МТЗ без ограничений

№	Алгоритм / наименование параметра / значение параметра			
	Алгоритм муравьиных колоний			
	$\alpha$	$\beta$	$\rho$	$q$
1	1,25	0,5	0,75	1
2	0,75	1	1,25	0,5
3	1,5	0,5	0,75	0,5
4	1,25	0,5	1	0,75
5	0,5	0,5	1	0,75
6	0,25	0,5	1	0,75
7	1,5	0,5	1	0,75
8	1,5	0,75	1,25	0,75
9	1,5	1,5	1,25	1,25
10	1,5	1,25	1	0,75
11	0,25	1	1	0,75
	Алгоритм пчелиных колоний			
	Доля пчел-работниц		Метод поиска нового источника	
1	0,15		Смена индексов	
2	0,15		Реверс подмножества	
3	0,25		Смена индексов	
4	0,25		Реверс подмножества	
5	0,4		Смена индексов	
6	0,4		Реверс подмножества	

## Продолжение таблицы 4.3

7	0,5	Смена индексов
8	0,5	Реверс подмножества
9	0,6	Смена индексов
10	0,6	Реверс подмножества
11	0,65	Смена индексов
12	0,65	Реверс подмножества
13	0,75	Смена индексов
14	0,75	Реверс подмножества
15	0,9	Смена индексов
16	0,9	Реверс подмножества
Генетический алгоритм		
	Шанс мутации	Метод мутации
		Метод отбора
1	0,2	Смена индексов
2	0,2	Реверс подмножества
3	0,2	Реверс подмножества
4	0,2	Смена индексов
5	0,4	Смена индексов
6	0,4	Реверс подмножества
7	0,4	Реверс подмножества
8	0,4	Смена индексов
9	0,6	Смена индексов
10	0,6	Реверс подмножества
11	0,6	Реверс подмножества
12	0,6	Смена индексов

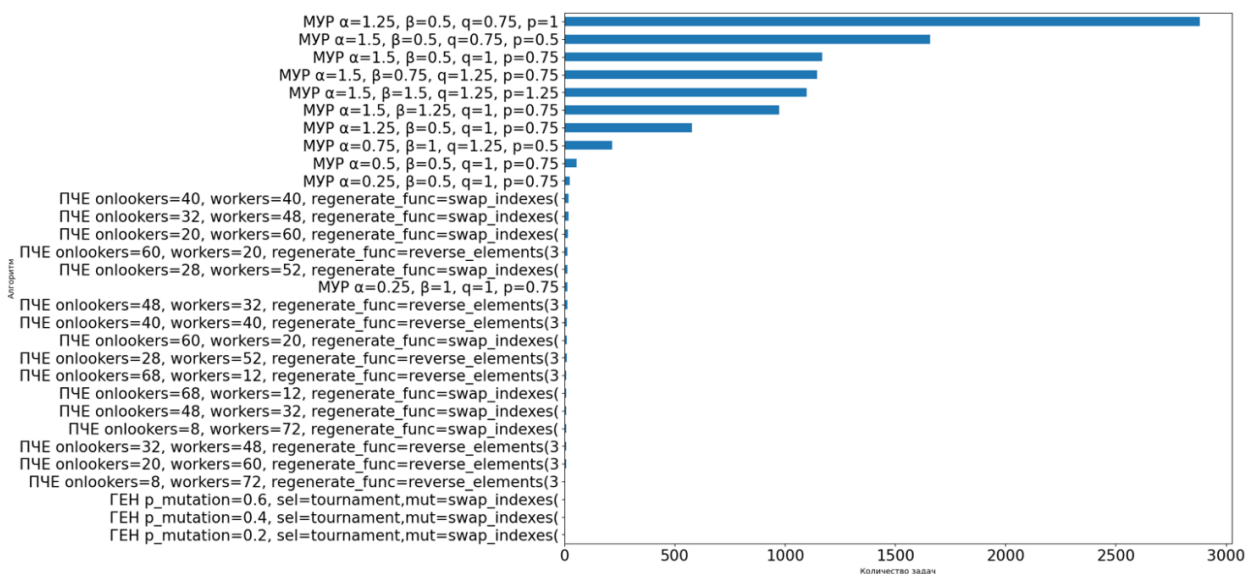


Рис. 4.2 – Диаграмма эффективности алгоритмов по количеству решенных задач



Для наибольшей численности задач, наиболее эффективным оказались модификации АМК, на втором месте – модификации АПК. Для малого количества задач эффективными оказались модификации ГА, использующие турнирный метод отбора.

На рисунке 4.3 приведены медианные значения времени работы самых эффективных алгоритмов. На рисунке 4.4 показана диаграмма медианы общего времени работы эффективных алгоритмов за исключением АМК. Можно заметить, что некоторые модификации ГА «отработали» немного быстрее модификаций АПК.

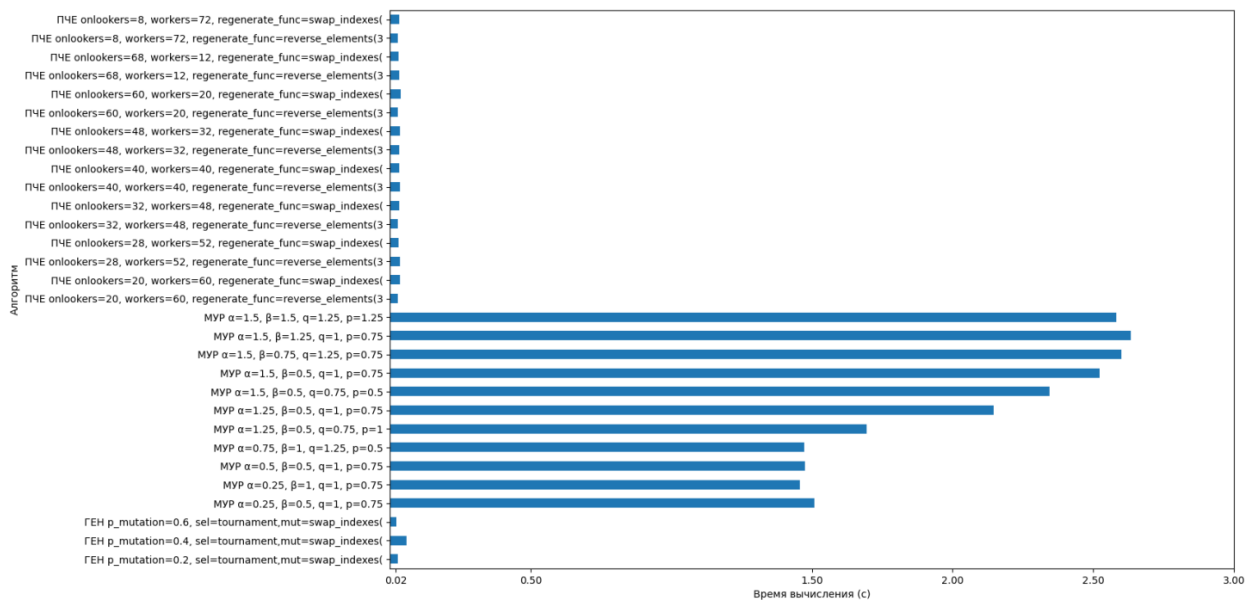


Рис. 4.3 – Диаграмма медианы общего времени работы эффективных алгоритмов

На рисунке 4.5 показана диаграмма эффективности модификаций алгоритмов по скорости решения задач. В контексте скорости прогона, лидирующие позиции заняли модификации ГА. Второе место заняли некоторые модификации АПК.

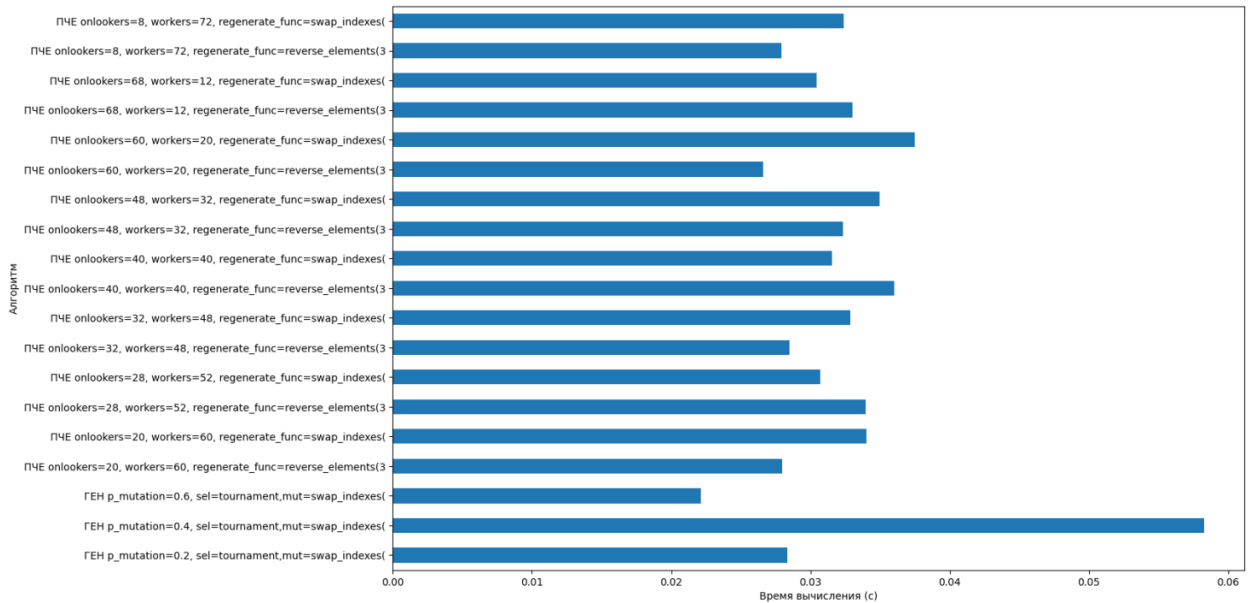


Рис. 4.4 – Диаграмма медианы общего времени работы алгоритмов без учета АМК

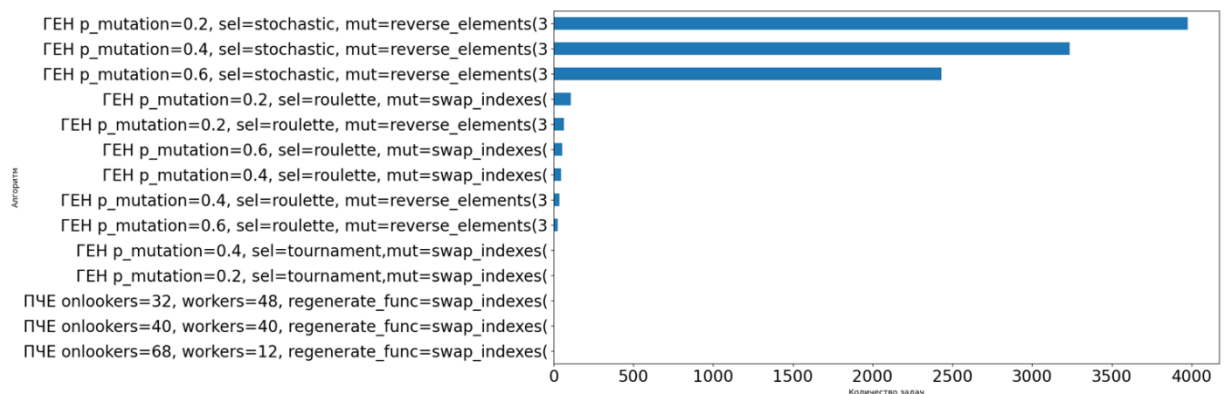


Рис. 4.5 – Диаграмма эффективности алгоритмов по скорости решения задач

Было принято решение выявить соотношение результатов самых быстрых алгоритмов к самым эффективным по скорости решения. Для каждой задачи результат самого эффективного алгоритма был поделен на результат самого быстрого. Из 5000 задач сформировано всего 144 уникальные пары типа «самый эффективный – самый быстрый». Полученные пары были сгруппированы по алгоритмам, как это показано на рисунке 4.6.

В большинстве задач АМК показал себя эффективным. Однако наиболее эффективным среди «быстрых» алгоритмов является АПК.

Следующим был проведен анализ эффективности алгоритмов по численности шагов, за которые был достигнут локальный минимум (скорость сходимости). На рисунке 4.7 приведена диаграмма эффективности алгоритмов по скорости сходимости. Лидирующие позиции по скорости сходимости делят между собой модификации ГА и АМК.

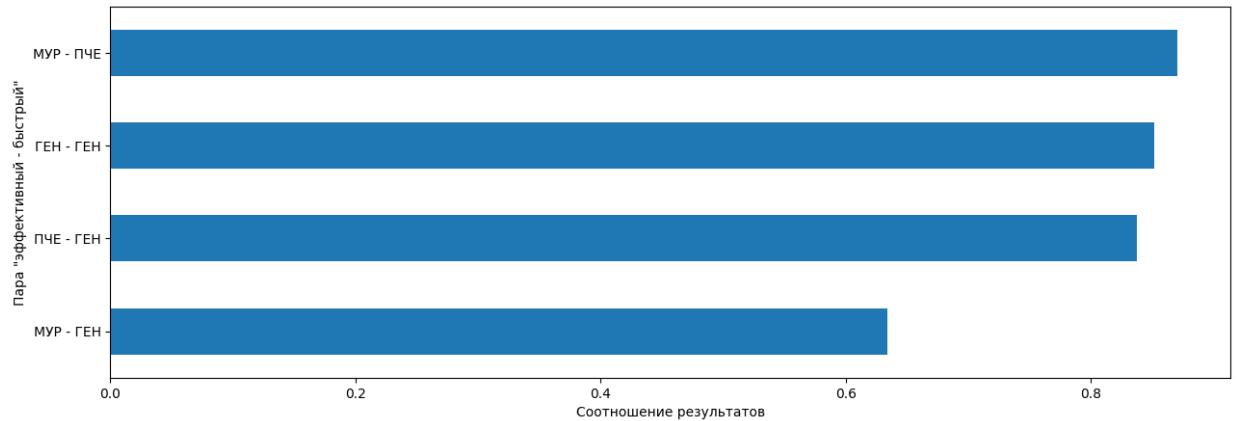


Рис. 4.6 – Диаграмма медиан соотношений решений самого эффективного и самого быстрого алгоритмов

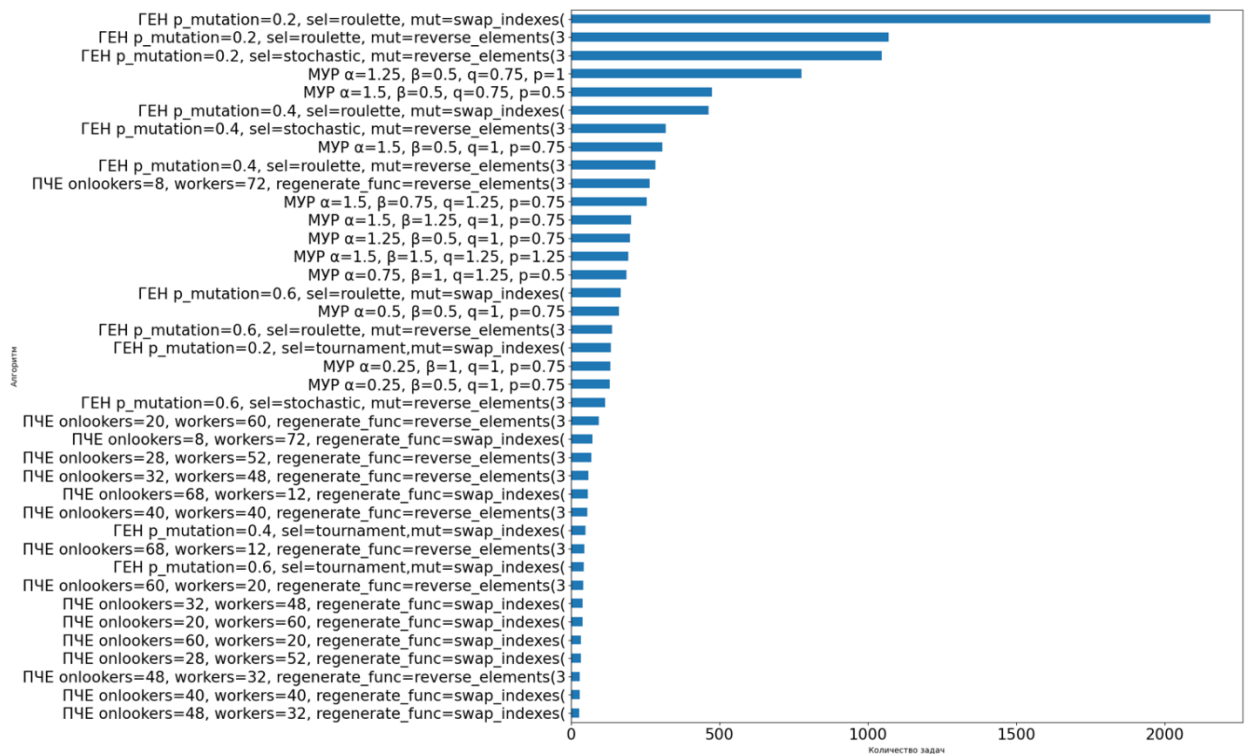


Рис. 4.7 – Диаграмма эффективности алгоритмов по скорости сходимости

На рисунке 4.8 показаны диаграмма средних значений скоростей сходимости алгоритмов (среди тех, который показал наибольшую скорость). На рисунке 4.9 приведены медианные скорости сходимости тех алгоритмов, которые показали себя наиболее эффективно в нахождении минимума.

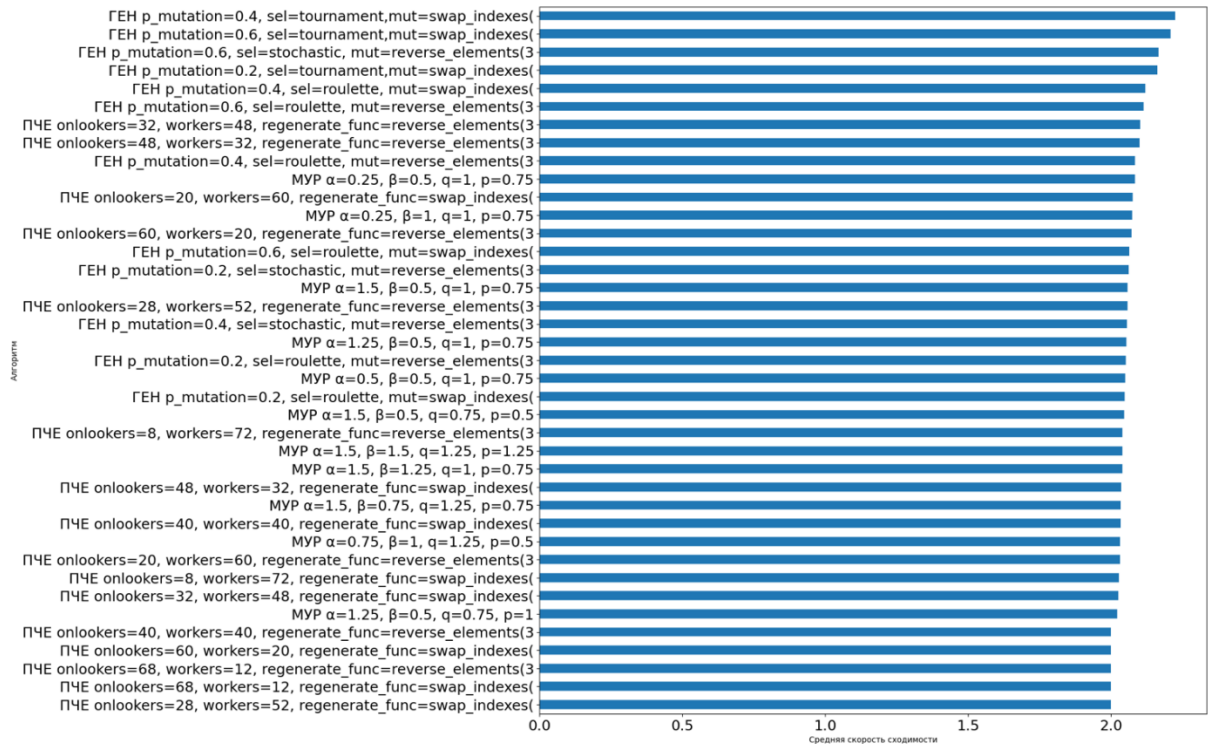


Рис. 4.8 – Диаграмма средних скоростей сходимости алгоритмов

В подавляющем большинстве задач (среди используемых эволюционных алгоритмов) самые оптимальные решения удается найти АМК. Лучшие вариации алгоритма варьируются в следующих пределах: 1.25-1.5 для  $\alpha$ , 0.5-1.5 для  $\beta$ , 0.75-1.25 для  $q$ , 0.75 для  $p$  соответственно. В зависимости от поставленной задачи, изменение этих параметров существенно влияет на эффективность алгоритма. Однако АМК работает в несколько раз медленнее, чем АПК и ГА.

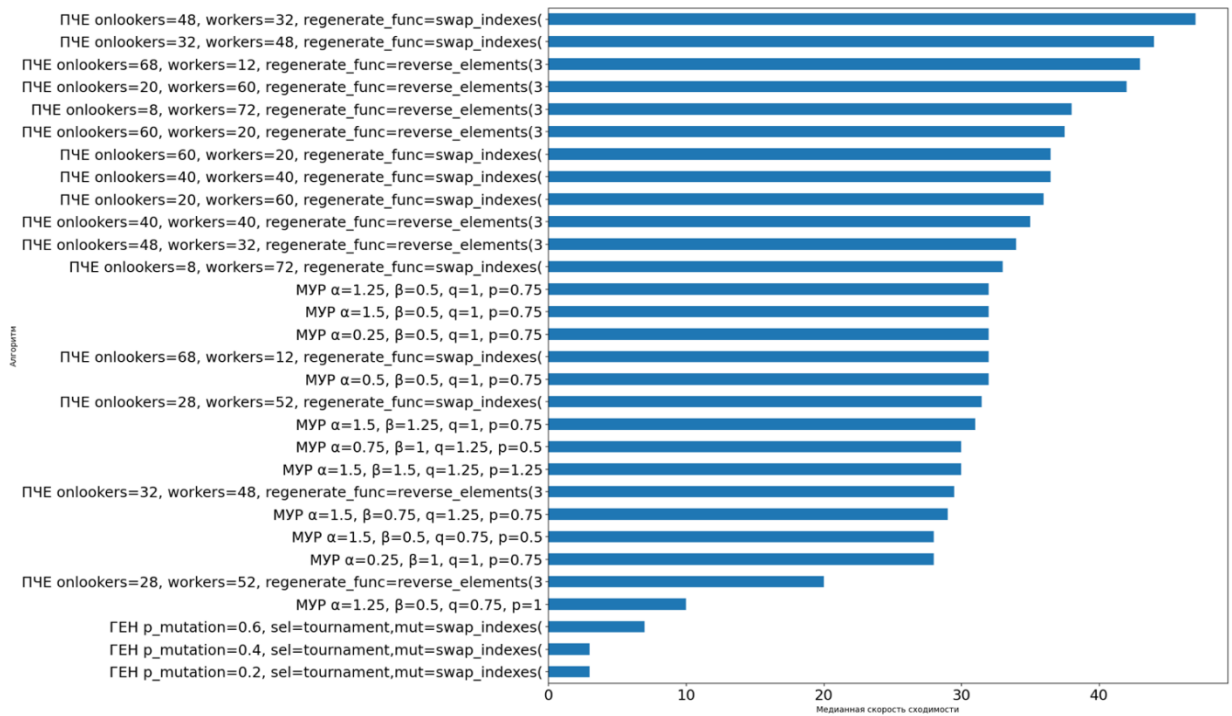


Рис. 4.9 – Медианная скорость сходимости лучших в нахождении минимума алгоритмов

ГА «выделяется» самой высокой скоростью работы как в плане сходимости, так и в плане общего времени прогона, однако в плане эффективности представляет меньшинство. Наиболее эффективными оказались вариации алгоритма, использующие в своей работе турнирный отбор и метод мутации «смена индексов» – данный метод мутации не вносил коренных изменений в решение задачи между итерациями, что помогло сохранить «удачные» варианты. При этом шанс мутации у самой эффективной вариации равняется 0.6. Наибольшую эффективность в плане вычислений показали вариации, имеющие шанс мутации 0.2, стохастический метод отбора и метод отбора «рулетка»: это связано с относительно низкой вычислительной сложностью используемых методов. Вариации с шансом мутации выше 0.6 не показали себя достаточно эффективными.

АПК проявил себя как самый сбалансированный по характеристикам эффективности и времени работы. Его вариации относительно часто показывали наибольшую эффективность вслед за вариациями АМК, а

скорость работы и сходимости во многих случаях сравнима с вариациями ГА. При этом не было найдено каких-либо существенных закономерностей между используемыми параметрами и показателями скорости и эффективности, из-за чего можно сделать вывод, что их выбор не оказывает существенного влияния на показатели эффективности.

#### 4.1.2 Многокритериальная транспортная задача с маршрутными ограничениями

Для тестирования МТЗ с маршрутными ограничениями за основы были взяты данные, использованные в п. 4.1.1 с сокращением численности задач до 1000, что связано с ростом сложности вычислений и времени, необходимого для проведения расчетов. Для данной задачи были введены маршрутные ограничения, согласно которым несколько маршрутов были однонаправленными, т. е. можно было попасть, например, из города А в город В, но не наоборот.

В таблице 4.4 представлены использованные вариации алгоритмов в виде параметрических значений. Для данной МТЗ был изменен перечень вариаций ГА, в частности, значение шанса мутации теперь располагается в диапазоне [0.2, 0.8] с идентичным шагом.

Таблица 4.4 – Параметрические значения вариаций алгоритмов для МТЗ с маршрутными ограничениями

№	Алгоритм / наименование параметра / значение параметра			
	Алгоритм муравьиных колоний			
	$\alpha$	$\beta$	$p$	$q$
1	1,25	0,5	0,75	1
2	0,75	1	1,25	0,5
3	1,5	0,5	0,75	0,5
4	1,25	0,5	1	0,75
5	0,5	0,5	1	0,75
6	0,25	0,5	1	0,75
7	1,5	0,5	1	0,75
8	1,5	0,75	1,25	0,75

Продолжение таблицы 4.4

9	1,5	1,5	1,25	1,25
10	1,5	1,25	1	0,75
11	0,25	1	1	0,75
	Алгоритм пчелиных колоний			
	Доля пчел-работниц		Метод поиска нового источника	
1	0,15		Смена индексов	
2	0,15		Реверс подмножества	
3	0,25		Смена индексов	
4	0,25		Реверс подмножества	
5	0,4		Смена индексов	
6	0,4		Реверс подмножества	
7	0,5		Смена индексов	
8	0,5		Реверс подмножества	
9	0,6		Смена индексов	
10	0,6		Реверс подмножества	
11	0,65		Смена индексов	
12	0,65		Реверс подмножества	
13	0,75		Смена индексов	
14	0,75		Реверс подмножества	
15	0,9		Смена индексов	
16	0,9		Реверс подмножества	
	Генетический алгоритм			
	Шанс мутации	Метод мутации	Метод отбора	
1	0,2	Смена индексов	Рулетка	
2	0,2	Реверс подмножества	Рулетка	
3	0,2	Смена индексов	Стохастический	
4	0,2	Реверс подмножества	Стохастический	
5	0,2	Смена индексов	Турнирный	
6	0,2	Реверс подмножества	Турнирный	
7	0,4	Смена индексов	Рулетка	
8	0,4	Реверс подмножества	Рулетка	
9	0,4	Смена индексов	Стохастический	
10	0,4	Реверс подмножества	Стохастический	
11	0,4	Смена индексов	Турнирный	
12	0,4	Реверс подмножества	Турнирный	
13	0,6	Смена индексов	Рулетка	
14	0,6	Реверс подмножества	Рулетка	
15	0,6	Смена индексов	Стохастический	
16	0,6	Реверс подмножества	Стохастический	
17	0,6	Смена индексов	Турнирный	
18	0,6	Реверс подмножества	Турнирный	
19	0,8	Смена индексов	Рулетка	
20	0,8	Реверс подмножества	Рулетка	
21	0,8	Смена индексов	Стохастический	
22	0,8	Реверс подмножества	Стохастический	
23	0,8	Смена индексов	Турнирный	
24	0,8	Реверс подмножества	Турнирный	

На рисунке 4.10 приведена диаграмма вариаций алгоритмов на количество задач, на которой ось  $Y$  – вариации алгоритмов, ось  $X$  – численность задач, в которых вариация алгоритма выдала наиболее оптимальное решение. В подавляющем большинстве задач наиболее эффективным оказался АМК. Следует заметить, что перечень наиболее эффективных вариаций АМК практически не изменился относительно аналогичного эксперимента без ограничений. Вторым по эффективности показал себя АПК.

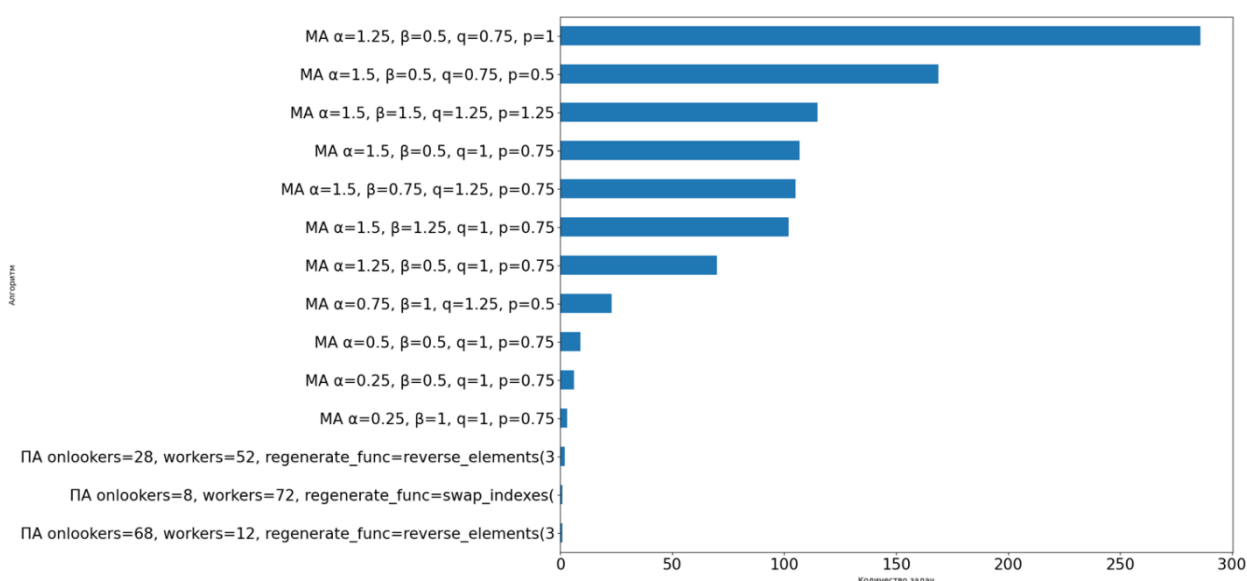


Рис. 4.10 – Диаграмма вариаций алгоритмов на количество задач

На рисунке 4.11 показана диаграмма медианного вычислительного времени алгоритмов, решения которых оказались наиболее эффективными. В условиях маршрутных ограничений, скорость работы вариаций МА значительно ниже по сравнению с условиями отсутствия ограничений. Это связано, в первую очередь, с увеличением вычислительной сложности. При этом, скорость работы ПА сохранилась примерно на том же уровне.



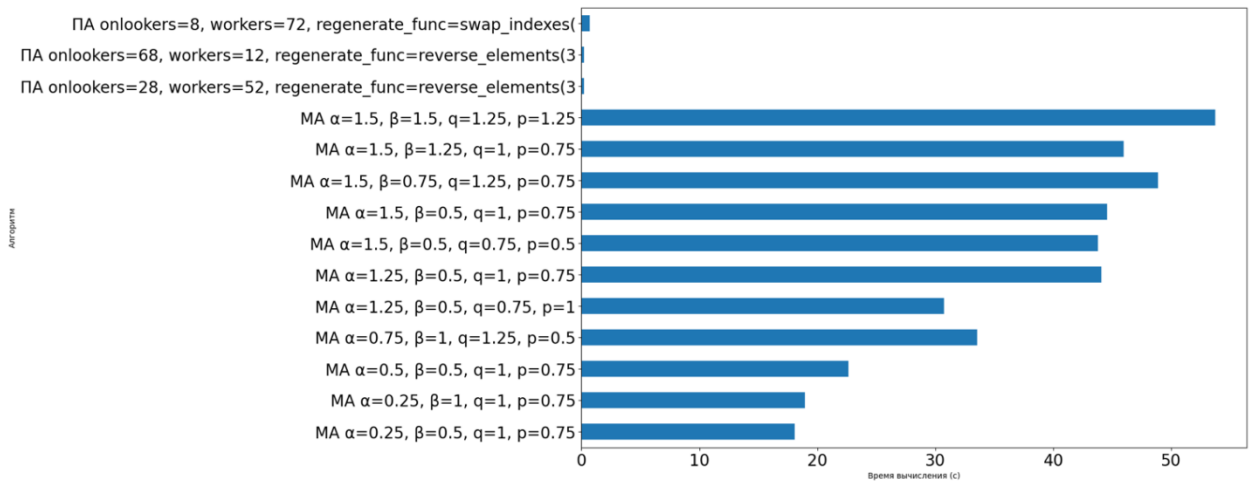


Рис. 4.11 – Диаграмма медианного вычислительного времени эффективных алгоритмов

#### 4.1.3 Многокритериальная транспортная задача с временными ограничениями

В основу была принята МТЗ с временными ограничениями, описанная в п. 2.2. В рамках вычислительных экспериментов МТЗ с временными окнами было решено порядка 76,7% вариаций постановок: результаты по остальным не были получены до окончания времени, отведенного на обработку, что подразумевает необходимость в данных случаях переключения модификации или расчетного алгоритма в целом.

Рисунок 4.13 содержит диаграмму медианного времени решения МТЗ эффективными вариациями алгоритмов. Наиболее быстрым алгоритмом среди остальных оказался ГА. При этом, менее эффективные его вариации, использующие такие методы отбора, как «рулетка» и «стохастический», оказались примерно в 2 раза быстрее, чем вариации, использующие турнирный отбор. Самым медленным оказался АМК – скорость его работы уступает скорости работы ГА в 4-16 раз. Сильную переменчивость во времени решении имеют вариации АПК: как среди «быстрых», так и «медленных» вариаций имеются те, что имеют большие и меньшие значения параметров «доля пчел-рабочих» и используют одинаковые алгоритмы

поиска нового источника. Предположительно, это связано с крайне малым количеством задач, в которых АПК оказался наиболее эффективным.

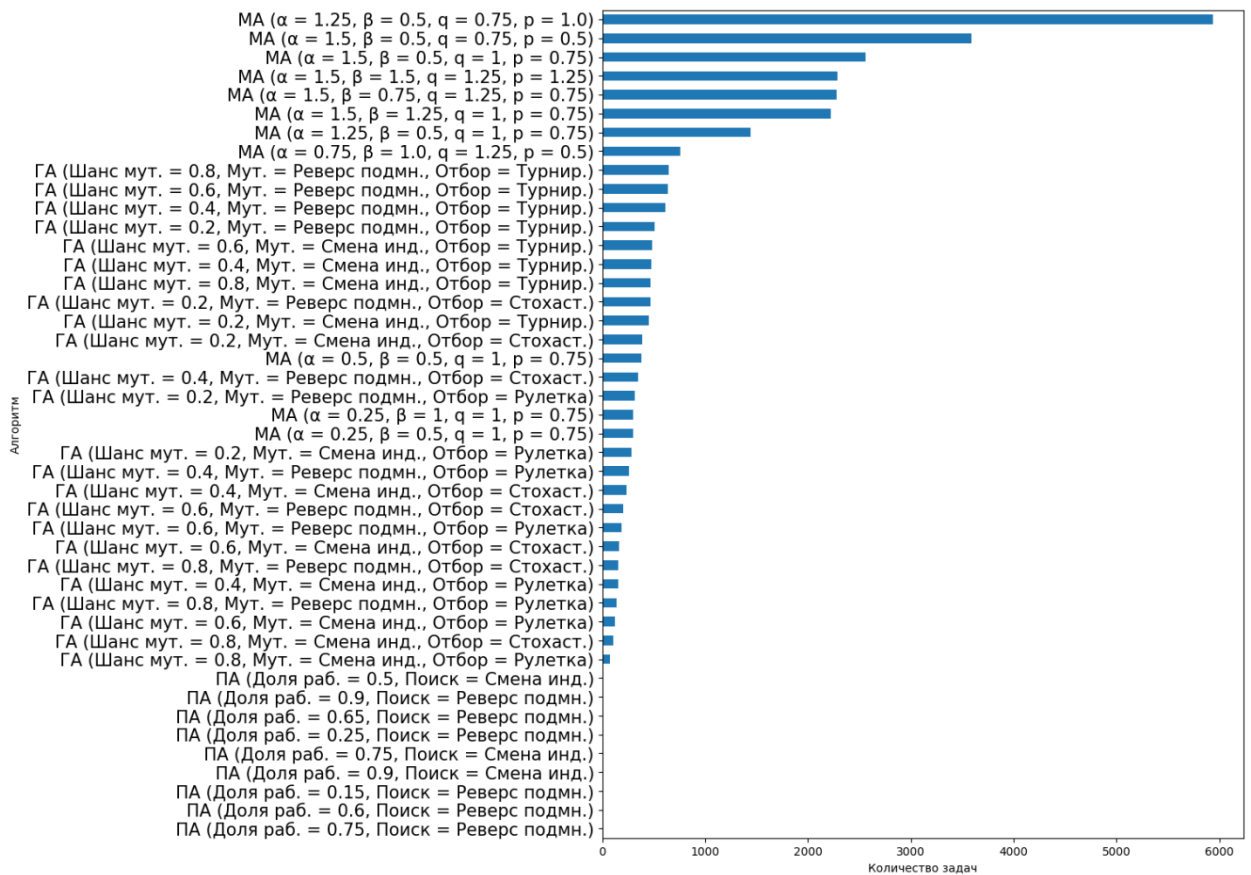


Рис. 4.12 – Диаграмма эффективности вариаций эволюционных алгоритмов по численности эффективных решений МТЗ

Результаты анализа решений задач с точки зрения скорости получения решения приведены в виде диаграммы на рисунке 4.14. Наиболее быстрыми показали себя модификации АПК, в частности, «смена индекса» и «реверс подмножества» с долей рабочих пчел от общего числа колонии 0.9. Значительно уступающие результаты по числу решенных задач показали модификации ГА и часть модификаций АПК.

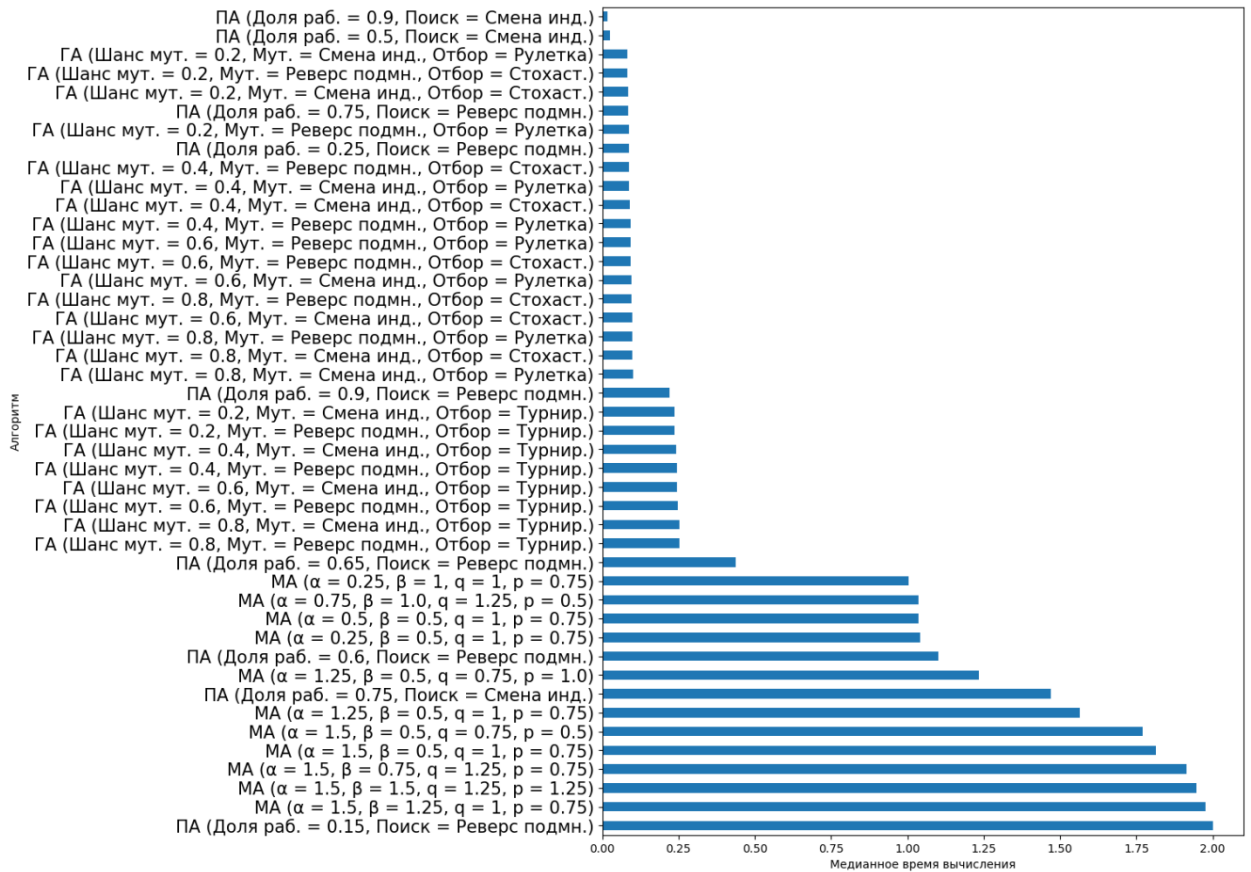


Рисунок 4.13 – Диаграмма медианных времен решения многокритериальной транспортной задачи эффективными вариациями алгоритмов

На рисунке 4.15 показана диаграмма количества задач эффективных вариаций алгоритмов, которые хотя бы раз оказывались и самыми быстрыми в какой-либо из задач. Места поделили между собой вариации АПК и ГА, причем вариации последнего с методами отбора «Стохастический» и «Рулетка» оказались наиболее быстрыми с учетом того, что большинство из них имеет относительно низкий шанс мутации.



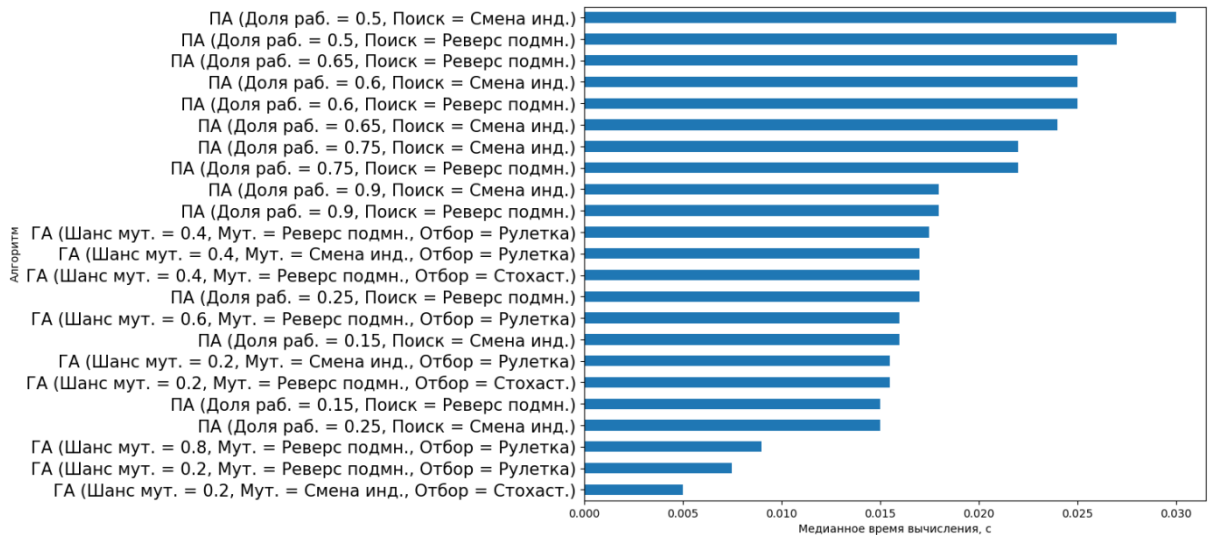


Рис. 4.16 – Диаграмма медианных времен решения многокритериальной транспортной задачи быстрыми вариациями алгоритмов

В соответствии с полученными результатами, можно сделать вывод, что время решения МТЗ с временными ограничениями вариациями ГА меньше времени, затрачиваемого АПК, в 2-6 раз, хотя оба алгоритма имеют крайне малое время вычисления, что может быть использовано для быстрого нахождения какой-либо сходимости. ПА для работы с данной задачей может подходить только в тех случаях, когда результирующее время расчетов не является критичным.

## 4.2 Обсуждение результатов

На рисунке 4.17 приводится медианное значение итерационного шага для алгоритмов, при котором достигнуто оптимальное значение. На нем можно увидеть, что АМК получал оптимальные значения на более поздних итерациях в сравнении с ГА и АМК.

Рисунок 4.18 содержит график усредненных значений результатов расчетов эволюционных алгоритмов по трем критериям.

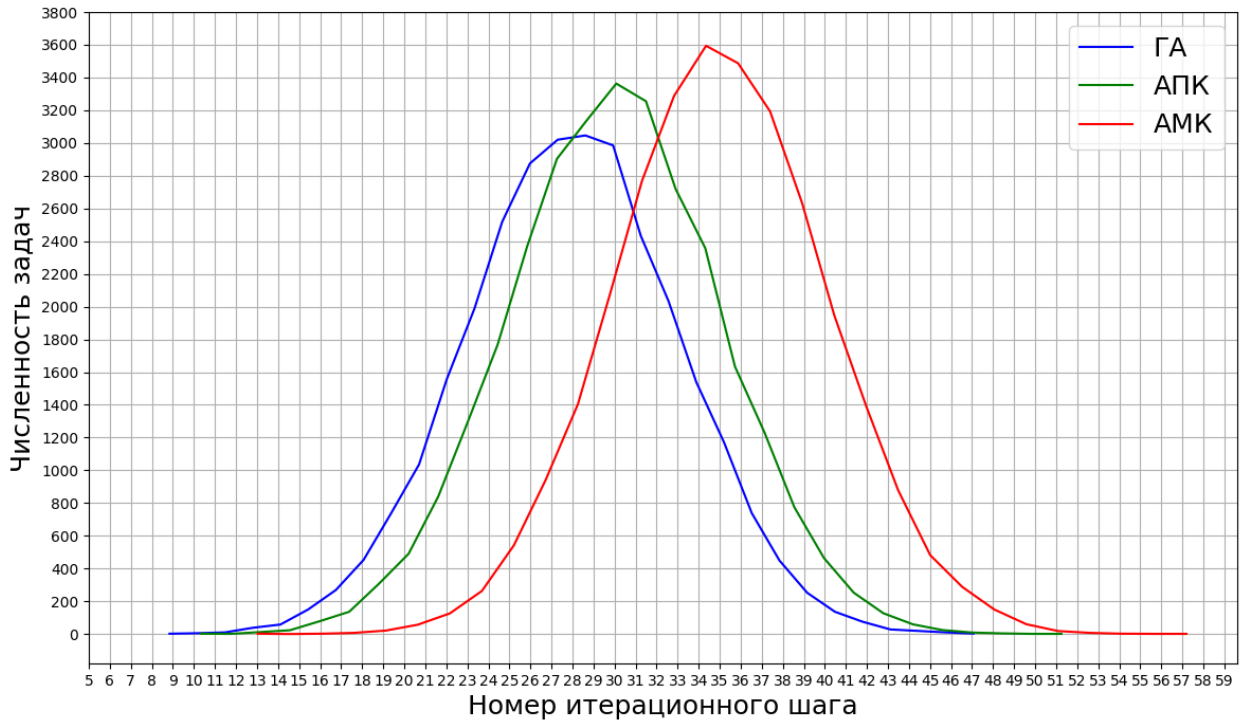


Рис. 4.17 – Медианное значение итерационного шага для алгоритмов, при котором достигнуто оптимальное значение

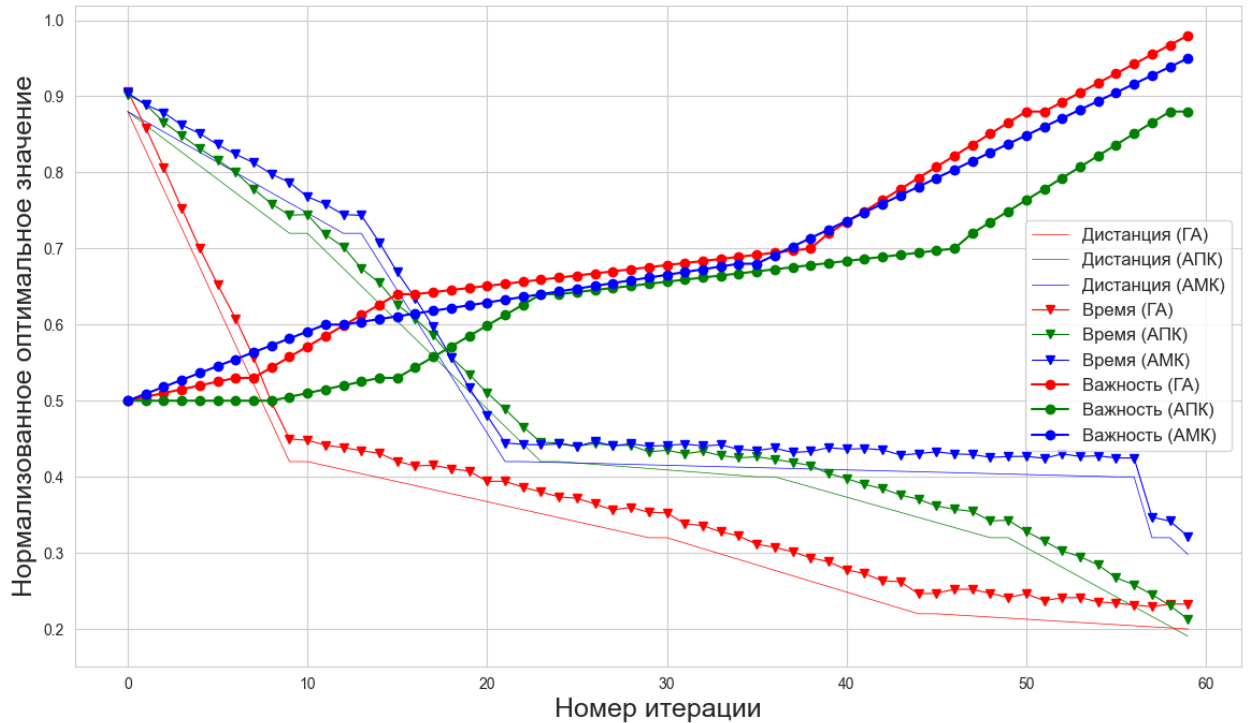


Рис. 4.18 – Усредненные значения результатов расчетов эволюционных алгоритмов по трем критериям

Результаты вычислительных экспериментов демонстрируют, что каждый из алгоритмов пригоден при решении определенного спектра задач оптимизации со своими особенностями:

– АМК предлагает высокую скорость получения парето-оптимальных решений, особенно в условиях многокритериальной транспортной задачи с временными ограничениями, однако в силу своей специфической особенности (высокой чувствительности к настроечным параметрам и начальным условиям размещения муравьев колонии) рекомендуется его использование в случаях, когда в распоряжении имеется достаточно времени для проведения нескольких просчетов с различными параметрами; в противном случае рекомендуется переключение на ГА либо АМК в зависимости от ситуации;

– ГА обеспечивает «баланс» между скоростью получения решений и их качеством, при этом стоит помнить о том, что в ряде постановок время поиска решения может возрасти в несколько раз; в таких случаях рекомендуется переключение на АПК, если требуется предельно ускорить процесс решения, либо на АМК, если необходимо получить более «качественное» решение;

– АПК предлагает наивысшую скорость получения результатов решения задач среди трех алгоритмов, однако полученные решения могут не соответствовать критериям парето-оптимальности; тем не менее, данный алгоритм имеет наивысший приоритет в предпочтительности в случаях небольших по размерностям задач, в ином случае рекомендуется его замена на АМК либо ГА в зависимости от ситуации.

Одновременно с этим, использование механизмов адаптации системы с модификацией и переключением активного алгоритма позволило добиться улучшения получаемых результатов в среднем на 4,5-11,2% в отношении каждого алгоритма.

### **4.3 Перспективы развития адаптивной системы поддержки принятия решений**

Разработанная адаптивная система имеет широкий спектр возможных модификаций и расширений. В качестве таковых можно привести следующие примеры:

- расширение списка эффективных в решении задач многокритериальной оптимизации алгоритмов, доступных для проведения оптимизационных расчетов;
- расширение параметров настройки для увеличения спектра и сложности решаемых задач многокритериальной оптимизации;
- расширение списка возможных модификаций алгоритмов в составе адаптивной системы, а также расширение возможностей их настроек;
- интеграция в систему элементов нейронных сетей для улучшения механизмов адаптации процесса решения задач многокритериальной оптимизации;
- перенос вычислительных мощностей в облачные ресурсы с целью ускорения вычислительных процессов и снижению нагрузки на устройство пользователя.

### **4.4 Выводы**

В данной главе представлена реализация адаптивной системы поддержки принятия решений при многокритериальной оптимизации с приведением следующих результатов.

1. В рамках вычислительных экспериментов многокритериальной транспортной задачи с временными окнами в отведенное ограниченное время было решено порядка 76,7% вариаций постановок, что подтверждает необходимость в данных случаях переключения варианта модификации эволюционного алгоритма.



2. Рассмотрены вариации алгоритмов с параметрами, произведена оценка их эффективности по количеству решаемых задач, численности эффективных решений, скорости решения.

3. Проведенная апробация адаптивной системы поддержки принятия решений посредством решения многокритериальных транспортных задач оптимизации, показала, что АМК предлагает высокую скорость получения решения, ГА обеспечивает баланс между скоростью решения и качеством, АПК при высокой скорости решения для задач небольшой размерности, а также продемонстрировали улучшение показателей получаемых результатов в среднем на 4,5-11,2% по сравнению с базовыми алгоритмами.

4. Произведен анализ полученных результатов апробации, в частности, разбор результатов вычислительных экспериментов для многокритериальной транспортной задачи без ограничений и с ограничениями.

5. Приводятся перспективы дальнейшего развития адаптивной системы поддержки принятия решений посредством расширения списка алгоритмов и их модификаций, параметров настройки и интеграции нейронных сетей.

## ЗАКЛЮЧЕНИЕ

Проведенное диссертационное исследование направлено на разработку адаптивной системы поддержки принятия решений, позволяющей решать задачи многокритериальной оптимизации (в частности, многокритериальную транспортную задачу) с применением эволюционных алгоритмов и механизмов адаптации.

По результатам диссертационного исследования необходимо выделить следующие основные аспекты:

1. Проведен комплексный анализ существующих эволюционных алгоритмов (ГА, АМК, АПК), применяемых для решения многокритериальных задач оптимизации, в частности – многокритериальной транспортной задачи, а также рассмотрены варианты их модификаций, направленных на их качественные улучшения.

2. Сформулировано специализированное математическое описание многокритериальной транспортной задачи с временными ограничениями, предполагающее применение при ее решении модифицированных эволюционных алгоритмов, механизмов проверки текущего решения и оперативной замены активного алгоритма.

3. Разработаны механизмы и проведена модификация эволюционных алгоритмов в адаптивной системе и правила базы знаний для взаимодействия с данными о решенных задачах оптимизации, позволяющие ускорить процесс решения многокритериальной транспортной задачи.

4. Разработаны схемы контроля и обработки результатов текущего решения с учетом информации о решении на предыдущем шаге (итерации) и ограничений по количеству итераций и формирующие рекомендации для модификации либо смены алгоритма.

5. Разработана структура адаптивной системы поддержки принятия решений при многокритериальной оптимизации транспортной задачи, в основе вычислительного инструмента которой заложены модифицированные

эволюционные алгоритмы (ГА, АМК, АПК), а также механизм их переключения в процессе решения задачи.

6. В рамках вычислительных экспериментов многокритериальной транспортной задачи с временными окнами в отведенное ограниченное время было решено порядка 76,7% вариаций постановок, что подтверждает необходимость в данных случаях переключения варианта модификации эволюционного алгоритма.

7. Проведенная апробация адаптивной системы поддержки принятия решений посредством решения многокритериальных транспортных задач оптимизации, показала, что АМК предлагает высокую скорость получения решения, ГА обеспечивает баланс между скоростью решения и качеством, АПК при высокой скорости решения для задач небольшой размерности, а также продемонстрировали улучшение показателей получаемых результатов в среднем на 4,5-11,2% по сравнению с базовыми алгоритмами.

Разработанную в ходе выполнения диссертационной работы адаптивную систему поддержки принятия решений при многокритериальной оптимизации транспортной задачи целесообразно применять для внедрения в организации и компании, связанные с обеспечением транспортной логистики, а также могут служить основой для последующего развития систем поддержки принятия решений.

**СПИСОК ЛИТЕРАТУРЫ**

1. ABC Algorithm Source Code by Delphi for Constrained Optimization [Электронный ресурс] – Режим доступа: <https://abc.erciyes.edu.tr/pub/ABCAlgorithmDelphiCodesforConstrainedOptimization.rar>
2. ABC2 [Электронный ресурс] – Режим доступа: <https://abc.erciyes.edu.tr/>
3. Ant Colony Optimization [Электронный ресурс] – Режим доступа: <https://www.mathworks.com/matlabcentral/fileexchange/52859-ant-colony-optimization-aco>
4. AntColonySystem [Электронный ресурс] – Режим доступа: <https://github.com/mbalchanowski/Ant-Colony-System>
5. Belykh M.A, Komarova E.P. Structure of intellectual support system solution using evolutionary algorithms // Материалы XIV Международной научно-практической конференции – Воронеж: «Научная книга», 2021. – С. 330-333.
6. Ben Hamida S., Gorsane R., Mestiri K. Towards a Better Understanding of Genetic operators for Ordering Optimization-Application to the Capacitated Vehicle Routing Problem // 15th International Conference on Software Technologies, Jul 2020, Lieusaint – Paris, France – P. 461-469.
7. Chandra Sen. A new approach for multi-objective rural development planning // The Indian Economic Journal, 1983. – Vol. 30 (4). – P. 91-96.
8. Genetic Algorithm [Электронный ресурс] – Режим доступа: <https://www.mathworks.com/discovery/genetic-algorithm.html>.
9. TSPLIB [Электронный ресурс] – Режим доступа: <https://github.com/ryanjoneil/tsplib/tree/master/elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp>

10. Golovin D., Zhang Q. Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization // ICML 2020 [Электронный ресурс] – Режим доступа: <https://arxiv.org/abs/2006.04655>

11. Hwang C.L., Masud A.S.Md., Paidy S.R., Paul Yoon K. Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey / Berlin, New York: Springer-Verlag, 1979. – 358 p.

12. Jayarathna D.G.N.D., Lanel G.H.J., Juman Z.A.M. Industrial vehicle routing problem: a case study // Journal of Shipping and Trade, 2022. – №7. – 27 p. [Электронный ресурс] – Режим доступа: <https://jshippingandtrade.springeropen.com/articles/10.1186/s41072-022-00108-7>

13. Kantorovich L. On the translocation of masses // C. R. (Doklady) Acad. Sci. URSS (N. S.), 37:199-201, 1942 [Электронный ресурс] – Режим доступа: <https://www.math.toronto.edu/mccann/assignments/477/Kantorovich42.pdf>

14. Karaboga D. An idea based on honey bee swarm for numerical optimization // Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005. – P. 1-10.

15. Koledina K.F., Alexandrova A.A. Solving the problem of multi-criteria optimization of the synthesis reaction of benzylalkyl esters by the method of "ideal" point and lexicographic ordering // Computational Mathematics And Information Technologies, 2022. – V. 1. – I. 1. – P. 12-19.

16. Li J.-Y., Zhan Zh.-H., Li Y., Zhang J. Multiple Tasks for Multiple Objectives: A New Multiobjective Optimization Method via Multitask Optimization // IEEE Transactions on Evolutionary Computation, 2023 [Электронный ресурс] – Режим доступа: <https://ieeexplore.ieee.org/document/10178002>

17. Maneggio V., Gambardella L.M., Fabio de L. Ant Colony Optimization, 2004 [Электронный ресурс] – Режим доступа: [https://www.researchgate.net/publication/2883153\\_Ant\\_Colony\\_Optimization](https://www.researchgate.net/publication/2883153_Ant_Colony_Optimization)

18. Monge G. Mémoire sur la théorie des déblais et de remblais. Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et

de Physique pour la même année, pages 666-704, 1781 [Электронный ресурс] – Режим доступа: <https://archive.org/details/histoiredelacad00germgoog/mode/2up>

19. Nurmagamedi S.T. Practical materials for training optimization study problems by math // Science, Education And Innovations In The Context Of Modern Problems, 2021. – V. 2. – I. 2. – P. 15-20.

20. Python code of the basic Artificial Bee Colony is released – [Электронный ресурс] – Режим доступа: <https://abc.erciyes.edu.tr/pub/ABCPython-master.zip>

21. Seif M.S., Mohammad Reza Tabeshpour, Akbar Golafshani A., Hayatdavoodi Masoud. Geometrical optimization of TLP hull using genetic algorithm method to minimize down time // 4th Int'l Conf. on Innovations in Engineering, Technology, Computers and Industrial Applications (IETCIA-17) Pattaya (Thailand). Aug. 3-4, 2017.– P. 26-29.

22. Shaimardanova G.F., Koledina K.F. Genetic algorithm for solving the inverse problem of chemical kinetics // Computational Mathematics And Information Technologies, 2022. – V. 1. – I. 1. – P. 41-49.

23. Wierzbicki A.P. A mathematical basis for satisficing decision making // Mathematical Modelling, 1982. – № 3 (5). – P. 391-405.

24. Абляимов О.С. О классификации задач оптимизации процессов // Universum: Технические науки, 2020. – №. 8 (77). – С. 20-22.

25. Андреев М.П., Сергеев М.Ю., Белых М.А. Применение генетического алгоритма для решения транспортной задачи // Сборник трудов «Научная опора Воронежской области», Воронеж, 2022 г. – С. 146-148.

26. Арифжанов А.Ш., Мухамедиева Д.К., Хасанов У.У. Моделирование процессов оптимизации структуры сил и средств на пожаре с использованием параллельных вычислений // Высокопроизводительные вычислительные системы и технологии, 2021. – Т. 5. – № 2. – С. 35-41.

27. Барабанов В.Ф., Гребенникова Н.И., Коваленко С.А. Разработка алгоритма для решения задачи маршрутизации транспорта в городских условиях // Вестник ВГТУ, 2017. – Т. 13. – № 5. – С. 22-26.

28. Баранов Д.А., Белых М.А., Барабанов В.Ф. Программная реализация задачи линейной оптимизации на примере муравьиного алгоритма // Оптимизации и моделирование в автоматизированных системах: труды Международной научной школы – Воронеж: ВГТУ, 2023. – С. 23-27.

29. Баранов Д.А., Белых М.А., Барабанов В.Ф., Гребенникова Н.И., Черников В.Н. Программная реализация задачи линейной оптимизации на базе муравьиного алгоритма // Вестник ВГТУ, 2023. – Т. 19. – №6. – С. 53-58.

30. Басинский В.М., Степин Ю.Г. Алгоритм муравьиной колонии для решения задачи классификации и использование генетического алгоритма для подбора его параметров // Информационно-коммуникационные технологии: достижения, проблемы, инновации (ИКТ-2018), Электронный сборник статей I международной научно-практической конференции, посвященной 50-летию Полоцкого государственного университета, 2018 – С. 118-122.

31. Белых М.А. Формализация многокритериальной транспортной задачи с временными ограничениями // Моделирование, оптимизация и информационные технологии, 2024. – Т. 12. – № 2. – С. 1-9.

32. Белых М.А., Барабанов А.В. Схема работы выбора эволюционного алгоритма интеллектуальной системы // Информационные технологии моделирования и управления. Научно-технический журнал. Воронеж: Научная книга, 2022. – №2 (128). – С. 114-117.

33. Белых М.А., Барабанов В.Ф., Подвальный С.Л., Донских А.К. Структура интеллектуальной системы поддержки эволюционных алгоритмов // Вестник ВГТУ, 2021. – Т. 17. – №3. – С. 7-13.

34. Белых М.А., Баранов Д.А. Разработка интеллектуальной системы оптимизации на основе эволюционных алгоритмов // Нано-био-технологии. Теплоэнергетика. Математическое моделирование. Сборник статей

международной научно-практической конференции. Липецк, 2024. – С. 158-164.

35. Белых М.А., Баранов Д.А. Решение задачи коммивояжера вариативным муравьиным алгоритмом // Информационные технологии моделирования и управления. Научно-технический журнал. Воронеж: Научная книга, 2024. – №2 (136). – С. 116-119.

36. Белых М.А., Вдовин Д.А., Нужный А.М., Гребенникова Н.И. Алгоритмы и программные средства обработки облака точек // Сборник трудов «Научная опора Воронежской области», Воронеж, 2020. – С. 13-15.

37. Белых М.А., Вдовин Д.А., Нужный А.М., Гребенникова Н.И. Анализ аспектов изучения технологии обработки облака точек // Инженерные Системы и Сооружения. Воронеж, 2020. – №1 (38) – С. 105-113.

38. Бизли Д. Python. Подробный справочник. – Пер. с англ. / СПб.: Символ-Плюс, 2010. – 864 с., ил.

39. Бобровников Б.Н., Барабанов А.В., Белых М.А. Мониторинг состояния вычислительной системы инструментами языка разработки // Информационные технологии моделирования и управления. Научно-технический журнал. Воронеж: Научная книга, 2024. – №2 (136) – С. 100-104.

40. Борисевич М.Н. Основы информационных технологий для специалистов АПК [Электронный ресурс] – Режим доступа: <https://www.vsavm.by/knigi/kniga3/1460.html>

41. Брахман Т.Р. Многокритериальность и выбор альтернативы в технике / М.: Радио и связь, 1984. – 287 с.

42. Буйначев С.К., Бокланг Н.Ю. Основы программирования на языке Python: учебное пособие / Екатеринбург: Изд-во Урал. ун-та, 2014. – 91 с.

43. Бурый А.С., Шевкунов М.А. Интеллектуализация процессов принятия решений в эргатических системах // Transport Business in Russia, 2015. – №4. – Р. 48-50.

44. Васильев А.Н. Программирование на Python в примерах и задачах / М.: Эксмо, 2021. – 616 с.



45. Вдовин И.В. Применение генетического алгоритма к задаче оптимизации извлечения данных из веб-источников // Прикладная математика и фундаментальная информатика, 2015. – №2. – С. 107-113.

46. Виссия Х., Краснопрошин В.В., Вальвачев А.Н. Интеллектуализация принятия решений на основе предметных коллекций // Вестник БГУ. Сер. 1. 2011. – № 3. – С. 84-90.

47. Витковский Д.И. Эволюционный алгоритм многокритериальной оптимизации // Сборник трудов Международной научной конференции «Актуальные проблемы прикладной математики, информатики и механики». ФГБОУ ВО «Воронежский государственный университет», 2020. – С. 1818-1820.

48. Власова И.А. Методы одномерной оптимизации: методические указания / Самара: СамГУ, 2015. – 90 с.

49. Гасников А.В., Двуреченский П.Е. Стохастический промежуточный градиентный метод для задач выпуклой оптимизации // ДОКЛАДЫ АКАДЕМИИ НАУК. – М.: Институт проблем передачи информации им. А.А. Харкевича РАН, 2016. – Т. 467. – № 2. – С. 131-134.

50. Гвоздев Л.Р., Медведева Т.А. Решение задачи маршрутизации транспортных средств с временными окнами с помощью алгоритма муравьиных колоний // ДонГТУ, Молодой исследователь Дона, 2022. – №3 (36). – С. 58-61.

51. Громов Ю.Ю., Потапов А.Н., Началов А.Л. Формирование обобщенной структуры нечеткого логического вывода при обработке разнородной информации подсистемы поддержки принятия решений проблемноориентированной системы управления информационным обеспечением авиации // Промышленные АСУ и контроллеры, 2023. – № 9. – С. 39-46.

52. Громов Ю.Ю., Бунин А.В., Потапов А.Н., Началов А.Л., Хасанов В.Р. Архитектура подсистемы поддержки принятия решений проблемноориентированной системы управления информационным обеспечением

авиации // Приборы и системы. Управление, контроль, диагностика, 2024. – № 7. – С. 42-50.

53. Генетические алгоритмы в MATLAB [Электронный ресурс] – Режим доступа: <https://habr.com/ru/articles/111417/>

54. Генетические алгоритмы в MATLAB. Global Optimization Toolbox [Электронный ресурс] – Режим доступа: <https://www.youtube.com/watch?v=c7RMPJ6acY>

55. Гетманова А.Д., Никифоров А.Л., Панов М.И. и др. Логистика: учебное пособие для ВУЗов / М.: Дрофа, 1995. – 156 с.

56. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. Пер. с англ. / М.: Мир, 1985. – 509 с.

57. Гольштейн Е.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа / М.: Наука. Главная редакция физико-математической литературы, 1969. – 384 с.

58. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. / М.: Мир, 1982. – 419 с.

59. Данильченко В.И., Данильченко Е.В., Курейчик В.М. Модифицированные генетические операторы, ориентированных на решение задачи размещения компонентов СБИС // Сборник трудов XX Всероссийской научной конференции молодых ученых, аспирантов и студентов ИТСАУ-2022 – Ростов-на-Дону, Таганрог, 2022. – С. 15-17.

60. Данильченко В.И., Курейчик В.М. Кодирование и декодирование в задаче формирования топологии СБИС в условиях разной ориентации разногабаритных компонентов // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии – 2022» («ИС & ИТ-2022», «IS&IT'22») Таганрог, 2022. – С. 179-186.

61. Демин Д.С. Эвристический алгоритм расчета трудоемкости работ на предприятии по фактически отработанному времени // Инновации и инвестиции, 2014. – № 12. – С. 137-140.

62. Долгова О.Э., Пересветов В.В. Лучевой поиск и муравьиный алгоритм в решении задачи маршрутизации транспорта // Информатика и системы управления, 2016. – Т. 48. – № 2. – С. 47-57.

63. Долгова О.Э., Пересветов В.В. Муравьиный алгоритм и метод локальных улучшений в решении задач маршрутизации транспорта с временными окнами кластерного типа // Материалы IV всероссийской научно-практической конференции «Информационные технологии и высокопроизводительные вычисления» – Хабаровск: ТоГУ, 2017. – С. 50-54.

64. Долгова О.Э., Пересветов В.В. Муравьиный алгоритм с ослаблением ограничений по временным окнам в решении задачи маршрутизации транспорта // Вычислительные технологии, 2018. – Т. 23. – № 5. – С. 49-62.

65. Донских А.К., Барабанов В.Ф., Гребенникова Н.И., Белых М.А. Обзор архитектуры систем управления интеллектом на основе полезности и дерева поведения // Воронеж: Вестник ВГТУ, 2021. – Т. 17. – №3. – С. 36-41.

66. Егорова К.В., Соколов С.С. Алгоритм пчелиной колонии как метод оптимизации при поиске разливов нефти группой беспилотных летательных аппаратов // Материалы Всероссийской студенческой научно-практической конференции «Математические модели техники, технологий и экономики» – Санкт-Петербург, СПбГЛУ им. С.М. Кирова, 2023. – С. 3-6.

67. Ершова К.А. Задача оптимизации маршрута движения выездной метрологической группы // Международный научный журнал «Вестник Науки», 2023. – Т. 2. – №1 (58). – С. 236-245.

68. Зобнина О.В., Дю А.И., Бабаева Ю.А. Многокритериальная оптимизация // Научно-образовательный журнал для студентов и преподавателей «StudNet», 2021. – Т. 4. – №1. – С. 87-93.

69. Золотарюк А.В. Математическая модель многокритериальной оптимизации транспортных перевозок // Инновационные технологии в науке и образовании, 2015. – № 1. – С. 317-320.

70. Кажанов А.А., Курейчик В.М. Муравьиные алгоритмы для решения транспортных задач // Таганрог: Известия РАН. Теория и системы управления, 2010. – № 1 – С. 32-45.

71. Кажаров А.А., Курейчик В.М. Обзор задач коммивояжера и маршрутизации автотранспорта // Сборник трудов Международной научно-технической конференции по интеллектуальным системам AIS'13 в 4-х томах. – М.: Физматлит, 2013 – Т. 2. – С. 117-123.

72. Карпенко А.П., Воробьева Е.Ю. Ко-эволюционный алгоритм глобальной оптимизации на основе алгоритма роя частиц // Наука и Образование: Научное издание МГТУ Им. Н.Э. Баумана, 2013. – №11. – С. 431-474.

73. Кобак В.Г., Титов Д.В., Плешаков Д.В., Золотых О.А. Повышение эффективности генетического алгоритма на базе модели Голденберга за счет применения элиты / Известия вузов. Северо-Кавказский регион, 2014. – № 3. – С. 12-15.

74. Ковалев И.В., Карасева М.В., Соловьев Е.В. Модификация муравьиного алгоритма для задачи формирования мультиверсионного программного обеспечения // Вестник СибГАУ, 2014. – №1 (53). – С. 19-24.

75. Константинова М.А. К вопросу многокритериальной задачи в транспортной логистике // Научное сообщество студентов XXI столетия. Технические науки: сб. ст. по мат. XVIII междунар. студ. науч.-практ. конф. – Новосибирск: «СибАК», 2014. – № 3(18). – С. 49-54.

76. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: «Нолидж», 2000. – 352 с.

77. Коробов А.А., Петин А.Е. Приближенное решение задачи параметрической оптимизации // Материалы XII Всероссийского совещания по проблемам управления ВСПУ-2014 – М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. – С. 2424-2428.

78. Кочетов Ю.А., Хмелев А.В. Гибридный алгоритм локального поиска для задачи маршрутизации разнородного ограниченного автопарка // Дискретный анализ и исследование операций, 2015. – Т. 22. – № 5 – С. 5-29.

79. Кошкин Б.П., Носков С.И., Оленцевич В.А., Рязанцев А.И. О многокритериальной транспортной задаче // Фундаментальные исследования, 2017. – № 7 – С. 35-38.

80. Кремер О.Б., Подвальный С.Л. Программная реализация решения оптимизационных задач методом генетического алгоритма // Вестник Воронежского государственного технического университета, 2012. – Т. 8. – № 3. – С. 21-24.

81. Кузнецов А.В., Холод Н.И., Костевич Л.С. Руководство к решению задач по математическому программированию. / Минск: Высшая школа, 1978. – 256 с.

82. Курейчик В.М., Кажанов А.А. О некоторых модификациях муравьиного алгоритма // Известия ЮФУ. Технические науки, 2008 – №4 (81). – С. 7-12.

83. Курейчик В.М., Мартынов А.В. Об алгоритмах решения задачи коммивояжера в сети интернет // Вестник РГРТУ, 2019. – № 68 – С. 37-43.

84. Курейчик В.М., Мартынов А.В. Об алгоритмах решения задачи коммивояжера с временными ограничениями // Информатика, вычислительная техника и инженерное образование, 2014. – № 1 (16). – С. 1-13.

85. Кушнир А.Ю. Многокритериальная оптимизация транспортных перевозок // Сборник статей участников V Международного научного студенческого конгресса, 2014. – С. 826-830.

86. Лавриченко О.В. Адаптивная система поддержки принятия решений на основе неманипулируемых механизмов // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника», 2015. – Т. 15. – № 2. – С. 109-114.

87. Лежебоков А.А., Нагоев З.В., Ошхунов М.М., Пшеноков А.Ю., Шугушхов Х.М. Биоинспирированный алгоритм решения задачи параметрической оптимизации // Известия Кабардино-Балкарского Научного Центра РАН, 2014. – № 5 (61). – С. 33-39.

88. Лекция 13. Экспертные системы: технология, этапы создания, применение [Электронный ресурс] – Режим доступа: <https://voronina-ai.github.io/docs/l-13/>

89. Максимова Н.Н., Колтунов Н.С. Поиск оптимального кольцевого маршрута с использованием пчелиного алгоритма // Вестник АмГУ, 2020. – № 89. – С. 16-21.

90. Маличенко Д.А. Эвристический алгоритм расчета размеров памяти в многоуровневой системе хранения // Информационно-управляющие системы, 2015. – № 5 (78). – С. 100-105.

91. Маргарян Е.А., Семашко М.А. Транспортные задачи как инструмент решения логистических проблем предприятия // Международный Научный журнал «Символ Науки», 2016. – №4. – С.135-137.

92. Математическая модель транспортной задачи [Электронный ресурс] – Режим доступа: <https://studfile.net/preview/6179691/page:8/>

93. Математическая форма алгоритма пчелиной колонии [Электронный ресурс] – Режим доступа: [https://www.researchgate.net/publication/3418512\\_Gambardella\\_LM\\_Ant\\_Colony\\_System\\_A\\_cooperative\\_learning\\_approach\\_to\\_the\\_Traveling\\_Salesman\\_Problem\\_IEEE\\_Tr\\_Evol\\_Comp\\_1\\_53-66](https://www.researchgate.net/publication/3418512_Gambardella_LM_Ant_Colony_System_A_cooperative_learning_approach_to_the_Traveling_Salesman_Problem_IEEE_Tr_Evol_Comp_1_53-66)

94. Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации / М.: Наука, 1978. – 352 с.

95. Моров В.А. Применение генетического алгоритма к задачам оптимизации. Реализация генетического алгоритма для задачи коммивояжера // Вестник АмГУ, 2012. – 6 с [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/primenenie-geneticheskogo-algoritma-k->

zadacham-optimizatsii-realizatsiya-geneticheskogo-algoritma-dlya-zadachi-kommivoyazhera/viewer

96. Муравьиные алгоритмы [Электронный ресурс] – Режим доступа: <https://blog.bullgare.com/wp-content/uploads/2019/05/aca.pdf>

97. Муравьиный алгоритм для задачи коммивояжера в Matlab [Электронный ресурс] – Режим доступа: <https://www.youtube.com/watch?v=ZJjT3hrsfgo>

98. Мэтиз Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / СПб.: Питер, 2020. – 512 с.

99. Нечаев Г.И., Рябичев В.Д., Скриннокова А.В., Киричевский А.Р. Решение задачи оптимальной загрузки автотранспорта для различных видов груза на основе модели условной оптимизации // Транспорт Азиатско-Тихоокеанского региона, 2023. – № 3 (36).– С. 94-99.

100. Ногин В.Д. Множество и принцип Парето: Учебное пособие. – 2-е издание, исправленное и дополненное / СПб.: Издательско-полиграфическая ассоциация высших учебных заведений, 2022. – 110 с.

101. Оптимизация на примере. Муравьиный алгоритм (ACS) против Метода отжига. Часть 2 [Электронный ресурс] – Режим доступа: <https://www.pvsm.ru/matlab/204887>

102. Остроух Е.Н., Демьянов А.М., Панасенко П.А. Переход от задачи условной оптимизации к задаче безусловной оптимизации // Материалы Всероссийской научной конференции. Министерство образования и науки Российской Федерации, Российский фонд фундаментальных исследований, Донской государственный технический университет, 2018 – С. 28-29.

103. Осыкина Ю.А., Чернышова Г.Д. Многокритериальная транспортная задача с разрывной целевой функцией // Вестник Воронежского государственного университета. Серия: системный анализ и информационные технологии, 2008. – № 2. – С. 10-12.

104. Павленко А.И., Титов Ю.П. Сравнительный анализ модифицированных методов муравьиных колоний // Прикладная информатика, 2012. – №4 (40). – С. 100-112.

105. Пахомова А.В., Баширзаде Р.Р. Разработка модели оптимизации транспортных затрат предприятия на основе концепции архитектуры интегрированных информационных систем (ARIS) // Вестник Пермского университета / Экономика, 2015. – № 3. – Т. 26. – С. 104-114.

106. Певнева А.Г., Калинкина М.Е. Методы оптимизации. Учебное пособие / СПб.: Университет ИТМО, 2020. – 66 с.

107. Петин П.С. О некоторых модификациях классической транспортной задачи в условиях нестабильной рыночной экономики // Материалы областного профильного семинара по проблемам естественных наук «Школа молодых ученых» – Липецк: ЛГПУ им. П.П. Семенова-Тян-Шанского, 2019. – С. 87-90.

108. Подвальный С.Л., Вдовин Д.А. Разработка специального программного обеспечения решения транспортных задач модифицированным генетическим алгоритмом с использованием многопоточности // Вестник ВГТУ, 2020. – Т. 16. – №4. – С. 7-12.

109. Представление знаний в экспертных системах: учебное пособие / сост. Морозова В. А., Паутов В. И. / Екатеринбург: Изд-во Урал. Ун-та, 2017. – 120 с.

110. Прохорова И.А., Аверьянова С.С. Применение генетических алгоритмов для решения многокритериальных задач // Наука ЮУрГУ: материалы 72-й научной конференции, май 2020. – 8 с [Электронный ресурс] – Режим доступа: [https://www.researchgate.net/publication/351023529\\_PRIMENENIE\\_GENETICESKIH\\_ALGORITMOV\\_PRI\\_RESENI\\_MNOGOKRITERIALNYH\\_ZADAC](https://www.researchgate.net/publication/351023529_PRIMENENIE_GENETICESKIH_ALGORITMOV_PRI_RESENI_MNOGOKRITERIALNYH_ZADAC)

111. Родзин С.И., Родзина О.Н. Поиск оптимальных решений комбинаторных задач: теория, эволюционные алгоритмы и их приложения для проблемно-ориентированных информационных систем // Информатика,



вычислительная техника и инженерное образование, 2014. – №4 (19) – С. 1-15.

112. Родников А.Р. Логистика: терминологический словарь. / М.: ИНФРА-М, 2000. – 350 с.

113. Руководство по Python [Электронный ресурс] – Режим доступа: [https://translated.turbopages.org/proxy\\_u/en-ru.ru.7834ac77-65ee9a58-45c3e01d-74722d776562/https/docs.python.org/3/tutorial/index.html](https://translated.turbopages.org/proxy_u/en-ru.ru.7834ac77-65ee9a58-45c3e01d-74722d776562/https/docs.python.org/3/tutorial/index.html)

114. Саймон Д. Алгоритмы эволюционной оптимизации / М.: ДКМ-Пресс, 2020. – 940 с.

115. Седых И.А., Сметанникова А.М. Применение пакета Matlab для параметрической идентификации окрестностных моделей на основе генетических алгоритмов // Вестник ВГУ, 2017. – №4. – С. 25-30.

116. Семенов С. С., Педан А. В, Воловиков В. С., Климов И. С. Анализ трудоемкости различных алгоритмических подходов для решения задачи коммивояжера // Системы управления, связи и безопасности. 2017. – № 1. – С. 116-131.

117. Скаков Е.С., Малыш В.Н. Модифицированный алгоритм пчелиной колонии ABC для проектирования топологии беспроводной сети // Труды Международного симпозиума «Надежность и качество». – 2016 – Т. 1. – С. 293-296.

118. Скаков Е.С., Малыш В.Н. Пчелиный алгоритм оптимизации для решения задачи планирования беспроводной сети // Программные продукты и системы / Software & Systems. – 2016 – № 3. – Т. 29. – С. 67-73.

119. Сотник С.Л. Конспект лекций по курсу «Основы проектирования систем искусственного интеллекта», 1997-1998 [Электронный ресурс] – Режим доступа: <https://coollib.com/b/542549-s-l-sotnik-konspekt-lektsiy-po-kursu-osnovyi-proektirovaniya-sistem-iskusstvennogo-intellekta/read>

120. Тема 1. Транспортные задачи. Случай конечных пространств [Электронный ресурс] – Режим доступа: <http://dfgm.math.msu.su/files/ivanov-tuzhilin/2018-2019/lt01.pdf>

121. Федина А.А., Нургалиев А.И., Скворцова Д.А. Сравнение результатов применения различных эволюционных алгоритмов для решения задачи оптимизации маршрута беспилотных аппаратов // Компьютерные исследования и моделирование. – 2022. – Т. 14. – № 1. – С. 45-62.

122. Хуссейн Ф.А., Финаев В.И. Исследование эффективности алгоритма искусственных потенциалов, муравьиного алгоритма и их комбинации при планировании траектории движения мобильного робота // Ростов-на-Дону: материалы Всероссийской научно-технической конференции с международным участием. В двух томах. 2020. – Т. 2. – С. 39-48.

123. Черненко В.В., Пискорский С.Ю. Экспертные системы // Актуальные проблемы авиации и космонавтики, 2012. – С. 322-323.

124. Штовба С.Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях, 2003. – №4. – С. 70-75.

125. Эйдельштейн М.Ю., Шапорова З.Е. Задача оптимизации транспортной схемы // Вестник КрасГАУ, 2015. – №1. – С. 94-97.

**ПРИЛОЖЕНИЕ А Свидетельство о государственной регистрации  
программы для ЭВМ**

**РОССИЙСКАЯ ФЕДЕРАЦИЯ**



**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

**№ 2022681169**

**Оптимизация управления производственными  
системами на базе генетических алгоритмов**

Правообладатели: *Олейникова Светлана Александровна (RU),  
Барabanов Владимир Федорович (RU), Гребенникова  
Наталья Ивановна (RU), Белых Михаил Алексеевич (RU),  
Селищев Иван Алексеевич (RU)*

Авторы: *Олейникова Светлана Александровна (RU), Барabanов  
Владимир Федорович (RU), Гребенникова Наталья Ивановна  
(RU), Белых Михаил Алексеевич (RU), Селищев Иван  
Алексеевич (RU)*

Заявка № **2022680852**

Дата поступления **30 октября 2022 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **10 ноября 2022 г.**

*Руководитель Федеральной службы  
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат 68b80077e14e3f0da94e0bd24145d5c7  
Владимир Зубов Юрий Сергеевич  
Действителен с 2013 по 26.05.2023

*Ю.С. Зубов*



## ПРИЛОЖЕНИЕ Б Акты о внедрении результатов диссертационного исследования



### АКТ

#### о внедрении результатов диссертационного исследования

**Белых Михаила Алексеевича**

Данный Акт удостоверяет, что материалы диссертационного исследования Белых М.А. были применены при проектировании систем адаптивной оптимизации с использованием эвристических алгоритмов.

В ходе диссертационного исследования автором были получены результаты, которые позволили улучшить качество работы отладочных модулей, связанных с задачами оптимизации. Разработанный автором адаптивный механизм проверки текущего решения позволяет производить переключение эвристических алгоритмов в процессе работы для увеличения эффективности расчетов и сокращения расходуемого времени.

Старший специалист-консультант Кошевцова О.А.

6 ноября 2023 г.



911@itcomf.ru

t.me/itcomf\_ru

vk.com/itcomf

394029, Воронежская обл., г. Воронеж,

Ленинский проспект, д. 15, оф. 317

+7 (473) 2-911-511

УТВЕРЖДАЮ

Руководитель проектов ООО «Девелоперс»

Нечепалева Е.С.

«24» июня 2024 г.**АКТ****о внедрении результатов диссертационного исследования****Белых Михаила Алексеевича**

Данным Актом удостоверяется, что теоретические и практические результаты диссертационного исследования Белых М.А. были использованы при разработке систем поддержки принятия решений при многокритериальной оптимизации транспортных задач.

Спроектированный автором модифицированный генетический алгоритм позволяет добиться сокращения времени поиска решения и повысить производительность программных комплексов.

Данный алгоритм представляется перспективным и рекомендуется к дальнейшему использованию при решении задач оптимизации.

Руководитель проектов ООО «Девелоперс»

Нечепалева Е.С.





**ООО «Киинай»**  
ИНН 6451015321 КПП 645101001

Юридический адрес: Саратовская обл., г. Саратов,  
ул. Фабричная, д. 1А литера А офис 609

АО «Тинькофф Банк»  
БИК: 044505576  
И/С: 40702810810000492037

Тел: +7 (906) 300-9432  
Электронная почта: office@keeneye.pro

20.02.2024

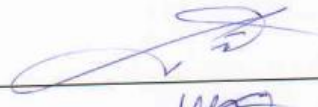
### АКТ


#### о внедрении полученных результатов диссертационного исследования Белых Михаила Алексеевича

Комиссия ООО «КИИНАЙ» в составе Курман Л.А. и Малиновкина В.А. настоящим актом подтверждает использование результатов диссертационного исследования Белых М.А. в деятельности ООО «КИИНАЙ».

Разработанный Белых М.А. механизм модификации эвристических алгоритмов (муравьиного и пчелиного) позволяет увеличить скорость обработки и снизить время вычислений оптимизационных задач, не снижая качество итогового результата.

Предложенный автором программный продукт рекомендован к внедрению в качестве дополнительного модуля для систем, специализирующихся на решении задач оптимизации.

Члены комиссии :  Курман Лилия Андреевна

 Малиновкин Владислав Алексеевич

УТВЕРЖДАЮ

Проректор по учебной работе

ФГБОУ ВО «ВГТУ»

А.И. Колосов

« » 2024 г.



## А К Т

**внедрения результатов кандидатской диссертации в учебный процесс  
ФГБОУ ВО «Воронежский государственный технический университет»**

**Тема диссертации:** «Эволюционные алгоритмы для адаптивной системы поддержки принятия решений при многокритериальной оптимизации транспортной задачи».

**Автор:** Белых Михаил Алексеевич

**Научный руководитель:** д.т.н., проф. Барабанов Владимир Федорович

Выполненной в ФГБОУ ВО «Воронежский государственный технический университет» на кафедре автоматизированных и вычислительных систем в рамках основного научного направления «Информатика и вычислительная техника»

В период с «15» апреля 2024 г. по н.в. внедрены в учебный процесс кафедры по направлению подготовки 09.06.01 «Информатика и вычислительная техника» на основании решения кафедры АВС от «10» сентября 2024 г., протокол № 2.


**1. Вид результатов, внедренных в учебный процесс:** совокупность знаний и представлений по теме диссертационного исследования.

**2. Область применения:** лабораторный практикум и лекционный курс по дисциплине «Проектная деятельность», выполнение курсовых проектов, выпускных квалификационных работ.

**3. Форма внедрения:** разработанные в диссертационном исследовании схемы и механизмы были внедрены в образовательный процесс в виде изучения особенностей проектирования адаптивных систем поддержки принятия решений, освоения механизмов адаптации эвристических алгоритмов и особенностей решения задач многокритериальной оптимизации.


**4. Эффект от внедрения.** Повышение качества образования: применение новых механизмов адаптации эвристических алгоритмов при решении задач многокритериальной оптимизации, позволяющих отслеживать качество решений на основе анализа результатов предыдущих итераций.

Научный руководитель диссертанта

  
Барabanов В.Ф.  
(подпись, Ф.И.О.)

«\_\_» \_\_\_\_\_ 2024 г.

Начальник УМУ

  
Скляров К.А.  
(подпись, Ф.И.О.)

«\_\_» \_\_\_\_\_ 20~~24~~г.

Диссертант

  
Белых М.А.  
(подпись, Ф.И.О.)


«\_\_» \_\_\_\_\_ 2024 г.

Декан ФИТКБ

  
Бредихин А.В.  
(подпись, Ф.И.О.)

«\_\_» \_\_\_\_\_ 20~~24~~г.

И.о. заведующего кафедрой АВС

  
Барabanов В.Ф.  
(подпись, Ф.И.О.)

«\_\_» \_\_\_\_\_ 2024 г.