

**МИНОБРНАУКИ РОССИИ**  
**Федеральное государственное бюджетное образовательное**  
**учреждение высшего образования**  
**«Ярославский государственный университет им. П.Г. Демидова»**

**Кафедра теоретической информатики**

Сдано на кафедру  
« 20 » июня 2025 г.  
Заведующий кафедрой,  
д.ф.- м.н., доцент  
\_\_\_\_\_ Кузьмин Е.В.

**Выпускная квалификационная работа**

**Разработка генетического алгоритма адаптивного**  
**круиз-контроля**

по направлению подготовки 02.03.02 Фундаментальная информатика и  
информационные технологии для бакалавров

Научный руководитель  
д.ф.-м.н., доцент  
\_\_\_\_\_ Башкин В.А.  
« 20 » июня 2025 г.

Студент группы ИТ-41  
\_\_\_\_\_ Петров Н.С.  
« 20 » июня 2025 г.

Ярославль 2025 г.

## РЕФЕРАТ

Объём 53 стр., 3 гл., 11 рис., 6 табл., 14 источников, 6 прил.

**Генетические алгоритмы, оптимизация, управление транспортом, модификация алгоритма, разработка, адаптивность, круиз-контроль.**

Решается задача построения оптимальной стратегии автоматического управления транспортным средством с целью минимизации расходования ресурсов на прохождение трассы (затрат топлива и времени).

Представлен самообучаемый (генетический) алгоритм, самостоятельно формирующий набор правил ускорения/замедления в зависимости от текущей скорости и свойств трассы (в частности, предстоящих подъёмов или спусков). Проведено обучение на различных классах тестовых трасс. Показано, что подобная эвристика является достаточно эффективной и гибкой, адаптируемой под различные новые факторы и критерии оптимальности.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ИСТОРИЧЕСКАЯ И ТЕОРЕТИЧЕСКАЯ СПРАВКА:	
1.1 История появления и основные идеи генетического алгоритма.....	7
1.2 Методы, параметры и особенности в генетическом алгоритме.....	9
1.3 Отличие генетического алгоритма от других алгоритмов.....	15
1.4 Историческая справка и идея круиз-контроля.....	18
1.5 Анализ существующих решений.....	20
РАЗРАБОТКА АЛГОРИТМА:	
2.1 Постановка задачи.....	21
2.2 Условия задачи.....	22
2.3 Описание структуры алгоритма.....	24
2.4 Реализация задачи.....	27
2.5 Результаты работы алгоритма.....	29
2.6 Выводы.....	35
МОДИФИКАЦИЯ АЛГОРИТМА:	
3.1 Дополнительные условия для задачи.....	36
3.2 Описание реализации.....	37
3.3 Результаты модифицированного алгоритма.....	38
3.4 Выводы.....	39
ЗАКЛЮЧЕНИЕ.....	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	43
ПРИЛОЖЕНИЯ.....	45

## ВВЕДЕНИЕ

В современном мире генетические алгоритмы стали одним из наиболее активно развивающихся методов в области оптимизации и машинного обучения. Их популярность объясняется универсальностью, способностью работать с большими и сложными пространствами решений.

Генетический алгоритм – это метод оптимизации, основанный на принципах эволюции и наследственности. Этот метод имитирует естественный процесс эволюции в природе, чтобы находить оптимальные решения для задач оптимизации. Генетические алгоритмы исследуют множество возможных решений параллельно, что повышает шансы нахождения наилучшего варианта. Они широко используются в различных областях, таких как инженерия, экономика, биология, информационные технологии и многие другие.

Главная идея генетического алгоритма заключается в том, чтобы создать популяцию особей и эффективно их эволюционировать, используя генетические операторы. Путем итеративного применения этих операторов алгоритм постепенно улучшает качество решений, стремясь к нахождению оптимального или близкого к оптимальному варианта.

В данной дипломной работе рассматривается основной принцип работы генетического алгоритма, его ключевые компоненты и особенности применения. Также будет рассмотрено использование генетического алгоритма для решения задач оптимизации и примеры его успешного применения.

**Цель** данной работы – разработка генетического алгоритма адаптивного круиз-контроля, позволяющего оптимизировать затраты топлива и времени при прохождении дистанции.

**Объектом** являются генетические алгоритмы многофакторной оптимизации и их свойства.

**Предметом** исследования является разработка генетического алгоритма для адаптивного круиз-контроля. Новый алгоритм должен быть достаточно эффективным (решать поставленную задачу за приемлемое число поколений), при этом он должен учитывать разнородные факторы (в частности, текущую скорость, профиль текущего и предстоящего участка трассы, расход топлива в зависимости от профиля, скорости и ускорения). Кроме того, схема алгоритма должна быть расширяема, то есть можно будет легко адаптировать и к другим возможным факторам, влияющим на эффективность прохождения трассы автомобилем (например, текущей погоде).

Цель, объект и предмет определяют следующие **задачи** реферата:

- изучить принципы работы генетического алгоритма и работы круиз-контроля;
- разработать генетический алгоритм адаптивного круиз-контроля;
- проанализировать результат алгоритма;
- модифицировать генетический алгоритм адаптивного круиз-контроля для дополнительных условий;
- проанализировать результат модифицированного алгоритма.

Цель и задачи реферата определили **структуру** работы, которая включает введение, три главы, заключение, список литературы и приложения. В первой главе «Историческая и теоретическая справка» рассматривается историческое происхождение генетических алгоритмов и систем круиз-контроля, основные этапы генетического алгоритма, различные методы и особенности его реализации, отличие генетического алгоритма от других алгоритмов оптимизации, его преимущества и недостатки, а также анализ существующих решений. Вторая глава «Разработка алгоритма» посвящена собственной разработке генетического алгоритма от постановки задачи до получения результата. В заключительной главе «Модификация алгоритма» приводится пример модификации генетического алгоритма для дополнительных условий задачи из второй главы.

Таким образом, в рамках данной дипломной работы разработан собственный генетический алгоритм на языке Python для задачи, связанной с системой адаптивного круиз-контроля. Продемонстрирована эффективность нового алгоритма, в том числе для разных классов трасс («равнинные», «горные» и «смешанные»). Всего лишь за 200 поколений во всех экспериментах получается устойчивое решение-чемпион. Также разработано расширение алгоритма для «электромобилей с солнечными батареями», учитывающее дополнительные специфические факторы (в частности, количество облаков на небе). Таким образом, задача решена успешно, была показана эффективность генетических алгоритмов для решения оптимизационных задач с большим количеством разнородных факторов.

# ИСТОРИЧЕСКАЯ И ТЕОРЕТИЧЕСКАЯ СПРАВКА

## 1.1 История появления и основные идеи генетического алгоритма

История генетических алгоритмов началась в 1950-х и 1960-х годах, когда такие исследователи, как Джон Холланд и его коллеги из Мичиганского университета, начали изучать вычислительные модели, основанные на биологической эволюции, для решения сложных задач оптимизации. Отсюда и пришла идея генетического алгоритма, где естественный отбор и мутации помогают видам адаптироваться к изменяющимся условиям окружающей среды.

Термин "генетический алгоритм" был впервые предложен Джоном Холландом в 1960-х годах как метод оптимизации, использующий принципы естественного отбора. Его основополагающая работа «Адаптация в естественных и искусственных системах», опубликованная в 1975 году, заложила основу для генетических алгоритмов, которые мы знаем сегодня.

На протяжении многих лет генетические методы успешно применялись для решения важных задач по оптимизации и поиску в различных областях, включая инженерию, информатику, экологические проблемы и биологию. Одним из основных преимуществ алгоритмов является то, что они способны эффективно решать большие пространственные задачи и находить высококачественные решения сложных проблем, которые трудно решить с помощью традиционных методов оптимизации.

В рамках развития этой области исследователи разработали различные расширения и улучшения базового алгоритма, такие как многокритериальная оптимизация, традиционные генетические алгоритмы и гибридные подходы, сочетающие генетические алгоритмы с методами других оптимизаций. Эти достижения еще больше расширяют применимость и внедряют генетические методы при решении некоторых проблем.

В последние годы, с появлением мощных вычислительных ресурсов и достижений в области машинного обучения и искусственного интеллекта,

генетические методы продолжают оставаться ценными для решения сложных задач по оптимизации. Исследователи постоянно ищут новые способы повышения производительности и масштабируемости генетических алгоритмов, что делает их необходимыми для исследований в области эволюционных вычислений.

Основные принципы генетического алгоритма были сформулированы Джоном Холландом в его работе «Адаптация в естественных и искусственных системах». Он предложил использовать механизмы естественного отбора и генетической рекомбинации для решения задач оптимизации. Генетический алгоритм базируется на следующих идеях:

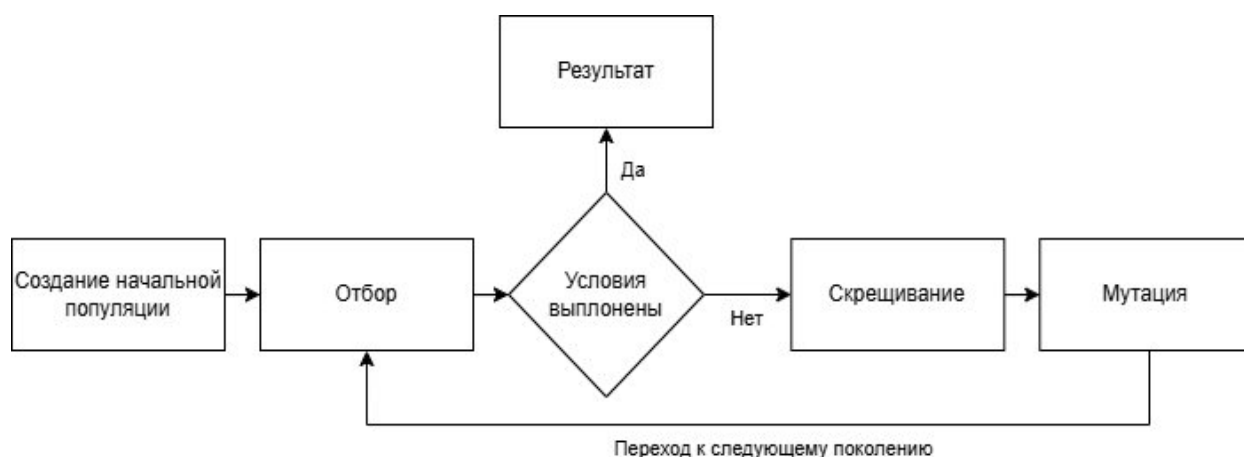
- Кодирование решений: в генетическом алгоритме каждое решение представляется в виде генов, которые кодируют определенные характеристики или параметры.
- Популяция решений: начальная популяция решений создается случайным образом. В каждом поколении осуществляется отбор лучших решений, которые используются для создания новых решений путем скрещивания и мутаций.
- Отбор: лучшие решения из текущей популяции отбираются для передачи своих генов следующему поколению. Этот процесс называется селекцией и напоминает естественный отбор.
- Скрещивание и мутации: для создания новых решений происходит скрещивание генов лучших решений, что приводит к получению комбинаций параметров. Также происходят случайные мутации, изменяющие некоторые гены.

Таким образом, генетический алгоритм представляет из себя набор повторяющихся действий, которые будут продолжаться несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма. Критерием остановки может быть нахождение конкретного решения, исчерпание числа поколений или времени, определённого для работы всего алгоритма.



## 1.2 Методы, параметры и особенности в генетическом алгоритме.

После знакомства с историей возникновения генетического алгоритма и основных идей необходимо более детально углубиться для цельного понимания. По своей идее генетический алгоритм может показаться простым в реализации и понимании, но на самом деле разработка генетического алгоритма имеет множество аспектов, которые могут сильно повлиять как на качество получаемого результата, так на время и нагрузку выполнения алгоритма. Если не обращать внимания на детали алгоритма, которые зависят от поставленной задачи, то стандартную схему генетического алгоритма можно представить таким образом:



**Рис. 1** – Стандартная схема генетического алгоритма

Это одна из вариаций стандартной схемы, но она отлично отображает основные этапы и параметры для дальнейшего рассмотрения всех нюансов и возможностей при разработке генетического алгоритма.

Любой генетический алгоритм начинается с создания начальной популяции. Начальная популяция – это набор из различных решений, которые являются начальной точкой для нахождения оптимального решения в поставленной задаче. Создание начальной популяции может оказать сильное влияние на поиск оптимального решения, поэтому правильная постановка задачи может сэкономить время как для разработки, так и для поиска решения. Если для задачи нет возможности качественно определить особей в начальном поколении, то используется случайная генерация.

Количество особей в популяции зависит от множества условий, и нет точного варианта для всех задач. Если поставленная задача имеет высокую сложность и большую область поиска оптимальных решений, то большая популяция позволит охватить множество решений за счёт разнообразия особей в популяции, но в то же время потребует множество ресурсов системы и времени для обработки каждой особи и итерации алгоритма. Если задача небольшая и имеет только одно оптимальное решение, то небольшое количество особей в популяции быстрее справится с меньшими затратами ресурсов. Однозначного выбора для этого значения нет, но выбор размера популяции имеет важную роль для работы всего генетического алгоритма.

Каждая особь является решением поставленной задачи, но в зависимости от задачи это решение может представлять собой различный вид данных: бинарный, численный, списочный, древовидный. В генетическом программировании такое решение называется «генотипом». Каждый генотип состоит из набора некоторого количества генов. Гены представляют собой небольшую часть от общего решения и позволяют создавать разнообразие среди решений. Также они участвуют при скрещивании и мутации особей для получения более оптимальных решений задачи. Выбор структуры генов зависит от конкретной задачи и параметров, которые возможно оптимизировать и структурировать, но именно это позволяет создавать генетические алгоритмы для различного рода задач в разных сферах жизни.

Первым из этапов итерационной цикличности алгоритма является отбор. Данный этап играет роль в выборе приспособленных особей для формирования следующего поколения на основании предыдущего поколения, но приспособленность новых особей может варьироваться для разнообразия генофонда. Основной целью является выделение лучших особей для дальнейших этапов генетического алгоритма и избавление от вариантов решения, которые либо точно не подходят для оптимального

решения, либо замедляют процесс получения этого оптимального результата.

Перед отбором результат каждой особи оценивается функцией приспособленности, которая определяется из условий и целей поставленной задачи для проверки на соответствие ожидаемого результата. Такая функция принимает некий объект данных, который является одним из решений задачи, и возвращает числовое значение, по которому уже делается заключение о приспособленности данной особи среди текущего поколения.

Когда все результаты особей поколения будут интерпретированы к конкретному и понятному значению, тогда можно приступить к отбору среди этих особей. Для этого существуют следующие методы:

- Правило рулетки – этот метод заключается в случайном выборе особей исходя из процентного представления её результата среди всех особей поколения. У особей с лучшей приспособленностью шанс пройти отбор выше, так как процентная доля больше, но из-за случайного выбора есть вероятность появления особей с меньшей приспособленностью в следующем поколении.
- Турнирный отбор – этот метод заключается в случайном выборе группы особей из поколения и выбора из этой группы самой лучшей особи. Данное действие повторяется до тех пор, пока не будет выбрано необходимое количество особей. Этот метод позволяет обходиться без функции приспособленности, если возможно сравнить результаты особей между собой.
- Элитарный отбор – этот метод заранее определяет прохождение небольшого процента приспособленных особей в следующее поколение, а остальной набор особей осуществляется другим методом отбора. Данный метод удобно подходит для сохранения хорошо приспособленных особей в следующем поколении до момента появления более приспособленных особей.

- Отбор вытеснением – этот метод определяет выбор особи не только по её приспособленности, но и по тому, есть ли особь с такими же генами среди уже отобранных особей для следующего поколения. Данный метод позволяет найти решение, которое может достигаться различным набором генов, и сохранить разнообразие в поколении.

Выбор метода для этапа отбора может быть разным и зачастую зависит от условий самой задачи и от формирования генотипа особей, поэтому весь выбор определяется формулировкой задачи, но метод отбора должен позволять алгоритму добиться оптимального решения задачи с возможностью минимальных потерь при отборе.

Скрещивание – один из важных этапов работы генетического алгоритма. Оно определяет новых особей на основе текущего поколения для дальнейшей проверки на соответствие искомому результату. Из отобранной популяции берутся пары особей без повторения, и путём обмена генов выбранных особей создаётся одна или две особи. Новые особи должны быть лучше приспособлены к решению поставленной задачи, чем один из родителей, поэтому такой родитель может отбрасываться в следующем поколении. Для скрещивания особей для новой популяции существует множество методов:

- Одноточечное скрещивание – этот метод случайным образом определяет деление генотипа пары особей на две части. Затем от одной особи берут первую часть генов, а от второй другую часть генов (или наоборот) и соединяют для получения генотипа новой особи.
- k-точечное скрещивание – этот метод идейно похож на метод одноточного скрещивания, но различается в k-разбиениях генотипов у пары особей.
- Равномерное скрещивание – этот метод заключается в формировании новой особи путём случайного отбора генов у пары особей. Случайный отбор можно ограничить условиями, если это необходимо.

- Упорядоченное скрещивание – этот метод использует метод k-точечного скрещивания для разбиения генотипов пары особей, а затем выбирает ген из каждой части разбитых генотипов с проверкой на повторение. Такой метод подходит в случаях, когда в генотипах новых особей могут повторяться значения, тем самым противореча условиям поставленной задачи.
- Имитация двоичного скрещивания – этот метод работает с генами, которые представлены двоичным кодом. Ключевой особенностью этого метода является равенство среднего значения хромосом у родителей и у потомков.

Методы скрещивания никак не ограничены только этими методами. Для скрещивания можно использовать любой собственный метод, но он должен создавать новых особей, которые будут лучше себя проявлять в решении поставленной задачи, иначе есть большая вероятность не достигнуть оптимального результата.

Мутация является заключительным этапом итерации генетического алгоритма в стандартном понимании. Данный этап позволяет с некоторой вероятностью изменить генотип новой особи для разнообразия генов относительно родителей этой особи. Вероятность наступления мутации имеет маленькое значение, чтобы сильно не изменить результат прошлых поколений, так как это может привести к отдалению от оптимального результата. Также мутация влияет только на небольшое количество генов новой особи по такой же причине. Этап мутации может быть реализован одним из возможных методов:

- Инвертирование битов – этот метод подходит для задач, где генотип представлен двоичным кодом. Его сутью является изменение случайным образом одного или более генов на противоположное значение.

- Мутация обменом – этот метод подходит для задач, где у генотипа важен порядок записанных генов. Такая мутация просто меняет этот порядок генов.
- Мутация обращением – этот метод подходит для задач, где у генотипа важен порядок записанных генов. Такая мутация схожа с мутацией обменом, но только меняет порядок у какой-то выбранной части генов.
- Мутация заменой – этот метод заменяет случайно выбранный ген на какое-то другое значение, которое обусловлено условиями задачи.

Как и методы скрещивания, методы мутации не ограничиваются только этим списком методов. Можно реализовать свои собственные методы мутации, но важно использовать небольшую вероятность наступления мутации и затрагивать только небольшую часть генов особи, чтобы добавить именно разнообразие в генотип новой особи, а не кардинально изменить его. Также выбор метода мутации зависит от поставленной задачи и от способа, которым представлен генотип особей.

Отбор, скрещивание и мутация участвуют в каждой итерации алгоритма до момента наступления точки останова или критерия останова. Это условие определяет завершение работы алгоритма и влечёт за собой получение результата работы всего алгоритма. Точкой останова может являться достижение указанного количества поколений, достижение значения результата в определённых пределах, отсутствие долгого изменения среди новых популяций или истечение определённого времени работы алгоритма. Как и на всех этапах генетического алгоритма, выбор подводящего критерия остановки зависит от поставленной задачи и необходимой оптимизации, но он должен позволять алгоритму достичь необходимого оптимального результата.

Результатом является возможное оптимальное решение поставленной задачи, так как именно на это рассчитан генетический алгоритм при корректной разработке всех этапов и параметров алгоритма. Также результатом может являться сам процесс поиска этого оптимального

результата для проверки быстродействия алгоритма или для использования полученных данных в разработке других алгоритмов.

Рассмотрев все особенности генетического алгоритма, можно сделать выводы о том, что генетические алгоритмы представляют собой гибкий и универсальный инструмент для решения разнообразных задач, однако их разработка требует тщательного подхода из-за множества параметров, методов и факторов, которые необходимо учитывать перед разработкой. Формирование начальной популяции, выбор методов для отбора, скрещивания и мутации, корректное определение генотипа особи и критерия останова влияют на эффективность работы всего алгоритма. При этом именно эта вариативность позволяет адаптировать генетические алгоритмы под самые разные по сложности и масштабу задачи. Таким образом, несмотря на сложность разработки, генетический алгоритм остаётся универсальным методом, способным находить эффективные решения даже в условиях неопределённости и высокой вычислительной сложности. Главное достоинство генетических алгоритмов в возможности не только находить оптимальные варианты решения, но и в возможности предоставить способ поиска этих вариантов, благодаря чему генетические алгоритмы находят применение в разных областях человеческой жизни.

### **1.3 Отличие генетического алгоритма от других алгоритмов**

Генетические алгоритмы занимают особое место среди методов оптимизации. В отличие от классических методов, опирающихся на математический анализ и градиенты, генетический алгоритм использует принципы наследственности, изменчивости и отбора, что делает его универсальным инструментом для задач, где другие подходы оказываются неэффективными. Сравним генетический алгоритм с другими алгоритмами и методами оптимизации.

В сравнении с градиентными алгоритмами, такими как градиентный спуск или метод Ньютона, генетические алгоритмы демонстрируют

принципиально иную идею работы. Градиентные методы движутся, постепенно приближаясь к локальному оптимуму, а генетические алгоритмы работают с целой популяцией решений, что позволяет им эффективно исследовать пространство поиска и избегать застревания в локальных оптимумах, поэтому генетические алгоритмы превосходят градиентные алгоритмы при оптимизации сложных, невыпуклых функций, где градиентные алгоритмы часто оказываются неэффективными.

Генетические алгоритмы обладают более сложной эволюционной структурой по сравнению с алгоритмом оптимизации роем частиц (PSO). Если алгоритм PSO имитирует коллективное поведение частиц, стремящихся к глобальному оптимуму, то генетический алгоритм используют методы скрещивания и мутации, что позволяет эффективнее исследовать пространство решений. Благодаря этому генетические алгоритмы демонстрируют большую универсальность, хотя и требуют более тщательного подбора параметров.

Генетические алгоритмы принципиально отличаются от метода имитации отжига тем, что оперируют не единичным решением, а целой популяцией кандидатов. В то время как алгоритм метода отжига постепенно снижает вероятность выбора худших решений по мере "охлаждения" системы, генетический алгоритм активно поддерживает разнообразие в популяции. Это позволяет им лучше анализировать сложные пространства решений. Хотя генетические алгоритмы могут потребовать много ресурсов, но они зачастую оказываются эффективнее в поиске глобальных оптимумов для многомерных задач.

Особенное отличие генетических алгоритмов проявляется при сравнении с детерминированными алгоритмами, такими как метод ветвей и границ. Детерминированные алгоритмы гарантируют нахождение оптимального решения для задач, где это возможно, но они становятся непрактичными для задач высокой размерности из-за экспоненциального роста вычислительной сложности. Генетические алгоритмы, напротив,



предлагают подход, который жертвует гарантией оптимальности в пользу масштабируемости и возможности работы с задачами большой размерности.

Важным отличительным свойством генетических алгоритмов является возможность выполнения вычислений в параллельном режиме. Оценка приспособленности особей в популяции может выполняться независимо друг от друга, поэтому генетические алгоритмы идеально подходят для реализации в параллельном режиме, что дает им существенное преимущество перед многими последовательными алгоритмами оптимизации.

Сравнив с некоторыми другими алгоритмами оптимизации, можно сделать вывод о том, что генетический алгоритм обладает рядом уникальных преимуществ, которые выделяют его среди других алгоритмов. К ключевым достоинствам генетического алгоритма относится его способность эффективно работать с задачами, где традиционные методы оптимизации оказываются несостоятельными. Прежде всего, генетический алгоритм не требует дифференцируемости целевой функции, что позволяет применять его для оптимизации разрывных и дискретных систем. Методы скрещивания и мутации позволяют обеспечить устойчивость к проблеме локальных оптимумов для многомерных задач и позволяют эффективнее исследовать пространство решений. Алгоритм демонстрирует хорошую масштабируемость и возможность реализации вычислений в параллельном режиме, поскольку оценка особей в популяции может выполняться независимо.

Однако генетический алгоритм имеет и существенные недостатки, ограничивающие область его применения. Главным минусом является высокая вычислительная стоимость из-за необходимости оценки целой популяции решений на каждой итерации, что делает алгоритм затратным по сравнению с градиентными методами. Генетический алгоритм требует тщательного подбора параметров, причем их оптимальные значения сильно зависят от конкретной задачи. Алгоритм не гарантирует нахождения

глобального оптимума и может преждевременно сходиться к субоптимальным решениям. Кроме того, поведение алгоритма плохо предсказуемо и трудно поддается теоретическому анализу, что осложняет его применение в критически важных системах.

Таким образом, генетический алгоритм занимает особое место среди методов оптимизации, демонстрируя универсальность и эффективность при решении определенного класса задач. Однако его применение требует тщательного анализа, так как выбор в пользу данного метода должен основываться не только на его очевидных преимуществах, но и на четком понимании требований к качеству конечного решения и допустимым вычислительным затратам.

#### **1.4 Историческая справка и идея круиз-контроля**

Круиз-контроль был создан американским инженером и механиком Ральфом Титором. Идея пришла ему в голову во время поездки с адвокатом, который, отвлекаясь на разговоры, постоянно менял скорость автомобиля. Это раздражало Титора и побудило его задуматься о системе, автоматически поддерживающей заданную скорость. В начале 1950-х годов он разработал первый автомобильный круиз-контроль и запатентовал свое изобретение. Однако, хотя Титор вошел в историю как создатель именно автомобильной системы, сама концепция автоматического регулирования скорости появилась гораздо раньше. Ещё в 1788 году шотландец Джеймс Уатт и англичанин Мэтью Болтон изобрели центробежный регулятор для паровых двигателей. Его принцип работы был схож с современным круиз-контролем: при увеличении оборотов механизм закрывал дроссельную заслонку, снижая скорость, а при падении оборотов — открывал ее, ускоряя движение. Таким образом, этот механизм можно считать прообразом современных систем поддержания постоянной скорости.

Круиз-контроль — это система, предназначенная для автоматического поддержания заданной скорости автомобиля без необходимости постоянного

участия водителя. Она эффективно работает как на ровных дорогах, так и на участках с подъемами и спусками. Активируется система по команде водителя, после чего датчики отслеживают положение педалей газа и тормоза, а электронный блок управления регулирует подачу топлива, изменяя положение дроссельной заслонки. Это позволяет сохранять постоянную скорость независимо от рельефа дороги. Использование круиз-контроля особенно удобно во время длительных поездок на высокой скорости: он не только снижает нагрузку на ноги водителя, но и способствует экономии топлива. При грамотном применении система помогает сократить расход на 5–10%. Но современные автомобили часто оснащаются адаптивным круиз-контролем.

Адаптивный круиз-контроль представляет собой усовершенствованную систему, которая не только поддерживает заданную водителем скорость, но и автоматически регулирует её для предотвращения возможных столкновений. Эта интеллектуальная система использует радарные или лазерные датчики, постоянно сканирующие пространство перед автомобилем и идентифицирующие потенциально опасные объекты на пути движения. При обнаружении препятствия система плавно снижает скорость транспортного средства, поддерживая безопасную дистанцию, а при устранении опасности - автоматически возвращается к первоначально заданному скоростному режиму. Такая функциональность значительно снижает нагрузку на водителя и минимизирует риск возникновения аварийных ситуаций.

Таким образом, круиз-контроль является важным этапом на пути к созданию полностью автономных транспортных средств. Данная технология демонстрирует оптимальный баланс между автоматизацией и человеческим фактором, что особенно важно на современном этапе развития транспортных систем. Сегодня это один из основных элементов практически каждого современного автомобиля. Это не только обеспечивает комфортную езду, но и позволяет лучше контролировать скорость.

## 1.5 Анализ существующих решений

Для проверки уникальности поставленной задачи и способа решения был проведён анализ существующих решений для задач с похожей тематикой и использованием генетического алгоритма.

В одной из обнаруженных задач используется генетический алгоритм в качестве способа для корректировки веса связей между нейронами для улучшения предсказаний сети [13]. Это означает, что веса сети представлены в виде генов, и гены эволюционируют для достижения оптимального решения. Целью этой работы является создание нейросети, которая будет управлять транспортным средством для прохождения гоночной трассы. Условия задачи определяют извилистую трассу, которая поделена на контрольные точки для измерения производительности, возможность поворотов транспорта и выхода за территорию трассы. Также стоит отметить, что данная задача решается на языке C# с использованием среды разработки Unity.

В другой найденной задаче [14] генетический алгоритм искал решение для оптимального результата прохождения гоночной трассы Формулы-1, то есть нахождения стратегии с наилучшим результатом по времени. Условия этой задачи основываются на реальных условиях, а также используются реальные данные для нахождения оптимальной стратегии.

Мною не было найдено работ, решающих при помощи генетических алгоритмов (или других близких подходов) задачу многофакторной оптимизации затраты ресурсов на прохождение трассы. Все найденные работы оптимизируют только время прохождения и не учитывают рельеф. Таким образом, можно сделать вывод о новизне рассматриваемой задачи и предлагаемых для её решения подходов.

## РАЗРАБОТКА АЛГОРИТМА

### 2.1 Постановка задачи

В поставленной задаче используется транспортное средство, которое может управляться с помощью алгоритма адаптивного круиз-контроля. Суть этого алгоритма заключается в увеличении или уменьшении скорости транспортного средства в зависимости от текущей скорости, типа ближайшего и следующего участка трассы.

Транспортному средству будет необходимо пройти фиксированную дистанцию из различных участков трассы с наиболее эффективным результатом в рамках определённых условий и параметров. Результат прохождения дистанции определяется через количество потраченного времени и топлива, но исходя из скорости транспортного средства и структуры дистанции могут получиться разные стратегии работы круиз-контроля, которые могут дать различный эффект.

Из-за различной структуры дистанций могут сильно отличаться результаты работы алгоритма и порядок действий транспортного средства. Для решения данной проблемы дистанции будут поделены на типы, а получение результатов будет выявляться как для каждого типа отдельно, так и для всех типов сразу.

По завершению работы алгоритма должен получиться набор действий для прохождения дистанции с оптимальными затратами в определённых условиях. Для этого определена система, которая сможет в процессе «самообучиться», по схеме генетического программирования с подобранными параметрами для поставленной задачи.

Использование схемы генетического программирования позволит гибко настроить необходимые условия для решения поставленной задачи в упрощённом виде и с дополнительными условиями и факторами.

## 2.2 Условия задачи

Рассматривая поставленную задачу на реальных условиях, можно сделать выводы о множестве различных факторов, которые непосредственно будут влиять на прохождение трассы, поэтому было решено принять меры по упрощению ситуации для транспортного средства.

Было решено поделить все трассы на три типа: равнинные, горные и смешанные. Такое деление охватывает большую часть трасс, встречающихся в реальных условиях, но не учитывает различные крайние случаи, например, полностью ровную дорогу. У каждого типа трасс определяется закономерность в последовательности участков, а также имеется чёткий предел по общей высоте всей трассы. Так, у трасс равнинного типа не будет разницы между низшей и высшей точкой больше 2, у горных трасс это значение будет от 6 до 8, а смешанные трассы получают разность от 4 до 5. Также стоит отметить, что все трассы имеют прямой вид дороги, то есть на всём протяжении трассы транспортное средство не совершает поворотов, а просто движется прямо.

Каждая трасса состоит из фиксированного количества участков. Каждый участок имеет один из пяти типов возможных значений:

- Резкий подъём (+2)
- Подъём (+1)
- Ровная поверхность (0)
- Спуск (-1)
- Резкий спуск (-2)

Участки различаются по наклону относительно трассы и влияют на количество необходимого топлива для изменения и поддержания скорости в конкретных значениях. Название типов участков имеет условное обозначение, а их количества хватит для составления трасс различного типа и получения результата от прохождения. Длина каждого участка 1 км.

Одним из важных параметров для получения оптимального результата является скорость транспортного средства. Было решено ограничить этот параметр до трёх значений: 20, 40 и 60 км/ч. Этого набора достаточно для проверки правильности работы алгоритма и получения результата для различных типов трасс.

Для задачи также определены некоторые условия, обоснованные удобством реализации работы алгоритма, но при этом не оказывающие существенного влияния на общий результат прохождения трассы. Одним из таких условий является то, что транспортное средство должно начинать и завершать прохождение трассы на одной скорости, а именно на скорости 40 км/ч. Другим условием стал момент, что после прохождения трассы следующий участок трассы будет типа “Ровная поверхность”.

Качество адаптивности транспортного средства к определённым трассам определяется из количества потраченного топлива и времени, умноженного на сто (1):

$$f + t * 100, \quad (1)$$

где  $f$  – количество потраченного топлива;  $t$  – время прохождения трассы.

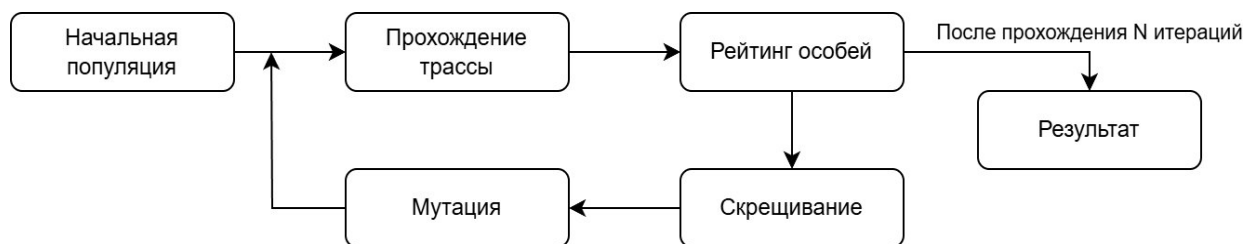
Транспортное средство никак не ограничено ни по времени, ни по топливу, но эти значения имеют разумный предел. Время прохождения трассы высчитывается как суммарное время прохождения всех участков, при этом время прохождения отдельного участка определяется как частное длины участка и средней скорости на участке. Средняя скорость на участке определяется как среднее значение из скорости, с которой транспорт начинает проходить участок, и скорости, с которой он завершит прохождение участка. Расход топлива определяется через таблицы скоростей и зависит от типа участка трассы. Таблица скоростей определяет количество топлива, которое нужно потратить транспортному средству для перехода от одной скорости к другой в зависимости от текущего типа участка. Время умножается на сто для более наглядной значимости этого параметра. Данная

формула не является единственной из возможных для данной задачи. Она является всего лишь необходимым параметром генетического алгоритма для определения качества адаптивности и возможности для сравнения результатов прохождения трасс.

В этом заключаются выбранные параметры задачи. Они упрощают разработку генетического алгоритма на данном этапе, но это никак не мешает дальнейшему масштабированию задачи под дополнительные условия или под изменение текущих. Данные условия позволяют решить поставленную задачу и получить оптимальный результат работы алгоритма.

### 2.3 Описание структуры алгоритма

После постановки задачи и выявления условий можно перейти к разработке генетического алгоритма. В структуре генетического алгоритма зачастую присутствуют следующие этапы: создание начальной популяции, отбор, скрещивание, мутация и результат. Мной разработанный алгоритм содержит похожие этапы, но при этом этап отбора разбит на два этапа: прохождение трассы и рейтинг особей. Таким образом, будет получена следующая структура алгоритма:



**Рис. 2** – Структура алгоритма

Рассмотрим каждый этап алгоритма по отдельности и разберёмся в их сути в алгоритме.

Начальная популяция представляет собой набор особей, каждая из которых имеет собственный генотип, то есть данные, позволяющие особи продемонстрировать свою способность в решении поставленной перед ней задачей. В данном алгоритме генотип каждой особи будет представлять собой таблицу из 4 столбцов: текущая скорость, текущий участок трассы,



следующий участок трассы и изменение скорости. Первый столбец принимает только значения, соответствующие 3 видам скорости. Вторым и третьим столбцом принимают значения 5 видов участков трассы. Значения в этих столбцах описаны в условиях задачи. Таким образом, после перемножения всех значений получается 75 строк комбинаций, то есть генов у каждой особи. Последний столбец никак не будет влиять на количество генов в генотипе. Он принимает следующие значения: -20, 0, +20. Это значение определяет изменение скорости после прохождения текущего участка трассы. Оно будет изменяться среди особей, тем самым различать их и определять лучшую особь. Это значение будет меняться в процессе скрещивания и мутации у новых особей. В начальном поколении у всех особей одинаковый генотип, так как они должны быть в одинаковых условиях. В их генотипе у каждого гена значение, отвечающее за изменение скорости, будет равно 0.

После составления начального поколения начинаются этапы, которые будут происходить последовательно после каждой итерации алгоритма. Первым таким этапом является «Прохождение трассы». В условиях было решено рассматривать результаты работы алгоритма по разным типам трасс. На этом этапе для каждой группы предоставляются её типовые трассы, которые определены заранее. Последовательно берётся каждая трасса, и её последовательно проходят особи из группы. В соответствии с генами для каждой особи определяется её скорость на каждом участке трассы и количество потраченного бензина и времени. После прохождения всех трасс получается набор данных с результатом каждой особи на каждой пройденной трассе. Эти данные переходят на этап «Рейтинг особей».

На этапе «Рейтинг особей» сортируется результат особей по каждой трассе в порядке возрастания. Далее составляется рейтинг особей. Относительно количества особей в группе и позиции особи после сортировки результата каждой особи начисляются баллы, то есть если в группе 30 особей, то первая особь в отсортированном списке получит 30

баллов, вторая - 29 баллов и т.д. Так начисляются баллы для каждой особи по всем пройденным трассам, а затем они суммируются, и получается общий список с рейтингом особей, который сортируется в порядке убывания. Такой список показывает насколько хорошо каждая особь прошла трассы относительно своей типовой группы и набора предоставляемых трасс.

Полученный список с рейтингом особей переходит на этап «Скрещивание». Полученный ранее список с рейтингом особей делится пополам на два списка. Поочерёдно каждая особь из первого получившегося списка будет скрещиваться с особью из второго списка в соответствии с одинаковой позицией в списках. Для скрещивания особей был выбран метод равномерного скрещивания в процентном соотношении 50%, то есть от каждой особи-родителя берётся по 50% случайных генов, но так как их в генотипе всего 75, то на один ген больше берётся от особи-родителя из первого списка. Получившаяся особь заменяет особь-родителя из второго списка в следующем поколении.

Именно такой выбор особей для скрещивания обоснован тем, что в среднем в новом поколении будет равная адаптивность генов, несмотря на возможность изменений генов у новых особей на этапе мутации. Такая мутация у некоторых особей не так сильно отразится на общем результате группы, как если бы было скрещивание только между ближайшими лучшими особями. При таком скрещивании мутация сильных новых особей могла бы сильнее повлиять на общую адаптивность группы особей в следующем поколении. Несмотря на случайность возможных изменений, ухудшающих последующий результат поколения, было решено оставить написанный выше порядок выполнения этапа скрещивания для нового поколения.

Последним этапом в итерационном процессе является «Мутация». На этот этап попадают только новые особи после скрещивания с вероятностью 25%. Мутация заключается в изменение одного гена среди каждых пяти генов особи, то есть мутация затронет 15 генов новой особи. Мутация

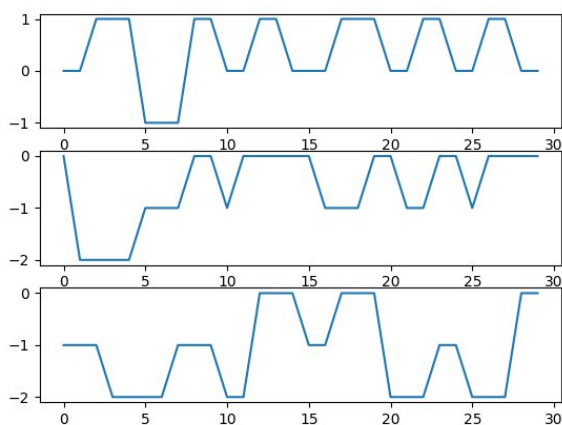
изменяет только значение, отвечающее за изменение скорости после прохождения текущего участка, соответствуя ограничениям по этому значению. Если текущая скорость будет равна 60, то изменение скорости может быть изменено только на 0 или -20, так как по условию скорость не может быть больше 60. То же самое и с минимальной скоростью. Если текущая скорость равна 20, то изменение скорости может быть изменено только на 0 или +20, так как по условию скорость не может быть ниже 20.

Этапы «Прохождение трассы», «Рейтинг особей», «Скрещивание» и «Мутация» повторяются N-ое количество итераций или поколений, что является точкой остановки генетического алгоритма и переходом на этап «Результат». После указанного количества поколений предоставляется результат по каждой группе особей относительно пройденных поколений по определённым типовым трассам. Можно заметить видимое улучшение результата относительно начальной популяции и конечной. Это позволит сделать выводы о корректности работы алгоритма.

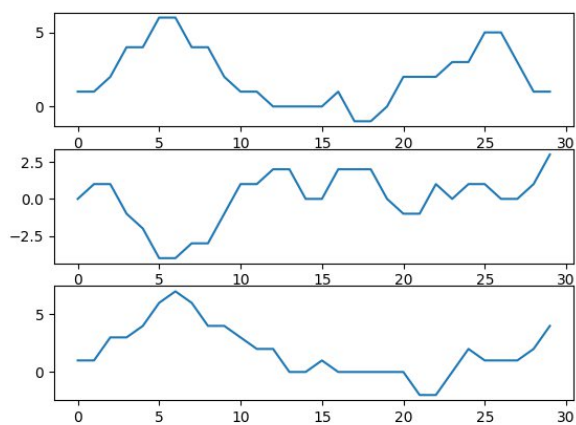
## **2.4 Реализация задачи**

Для тестирования реализованного алгоритма были определены следующие значения параметров. Всего в алгоритме будет участвовать 4 группы особей. Три из них будут соответствовать типовым трассам, которые были определены в условиях задачи, а последняя группа особей будет обучаться на всех типовых трассах сразу. В каждой группе по 50 особей. Такое количество хорошо покажет быстроту адаптивности особей к прохождению трасс и будет оптимальным для визуализации получения наилучшего результата.

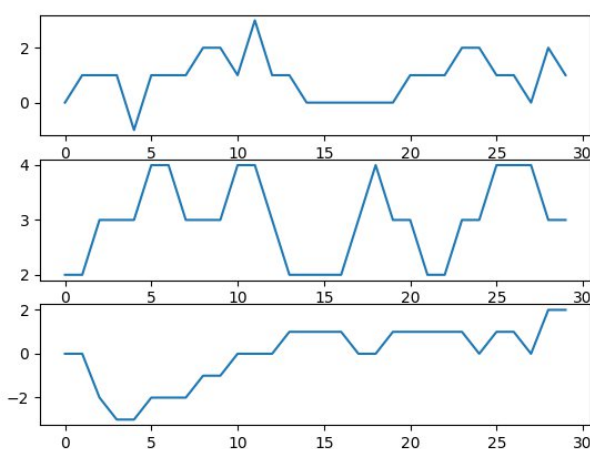
Для каждой группы определено по 3 типовых трассы. Они будут определены самостоятельно исходя из прописанных ранее условий и будут иметь следующий вид:



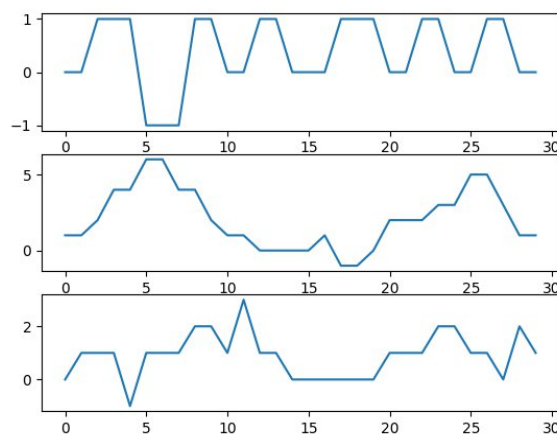
**Рис. 3 – Вид равнинных трасс**



**Рис. 4 – Вид горных трасс**



**Рис. 5 – Вид смешанных трасс**



**Рис. 6 – Вид всех трасс**

Каждая трасса имеет длину в 30 участков, то есть 30 километров. Данное количество трасс (и их длина) для каждой группы особей также достаточно для адаптации особей к прохождению трасс и будет оптимальным для визуализации получения наилучшего результата.

Ещё одним из параметров, влияющих на результат работы алгоритма, является количество топлива, которое необходимо будет тратить особи для преодоления каждого участка трассы. Для всех особей оно будет одинаковым и будет отображаться в качестве таблицы, где индексы в строках будут определять текущую скорость транспортного средства, а индексы в столбцах будут определять новую скорость после прохождения участка трассы. На пересечении строки и столбца будет значение, равное топливу, которое необходимо потратить для перехода от одной скорости к другой или для поддержания текущей скорости. Данное значение может

быть положительным или нулевым. Для каждого участка трассы своя таблица:

Таблица 1  
**Резкий спуск**

	20	40	60
20	0	1	2
40	0	0	1
60	0	0	0

Таблица 2  
**Спуск**

	20	40	60
20	1	3	5
40	0	1	3
60	0	0	2

Таблица 3  
**Ровный участок**

	20	40	60
20	5	7	12
40	2	6	10
60	1	4	8

Таблица 4  
**Подъём**

	20	40	60
20	10	14	25
40	8	12	20
60	6	10	16

Таблица 5  
**Резкий подъём**

	20	40	60
20	15	21	37
40	12	18	30
60	8	15	24

Все значения в таблицах условны и не опираются на какие-то реальные значения, но при этом имеют логическую основу реальных ситуаций.

Последним параметром является количество поколений. Было решено оставить 500 поколений, так как почти все группы особей успевали адаптироваться за это количество и предоставить оптимальный результат.

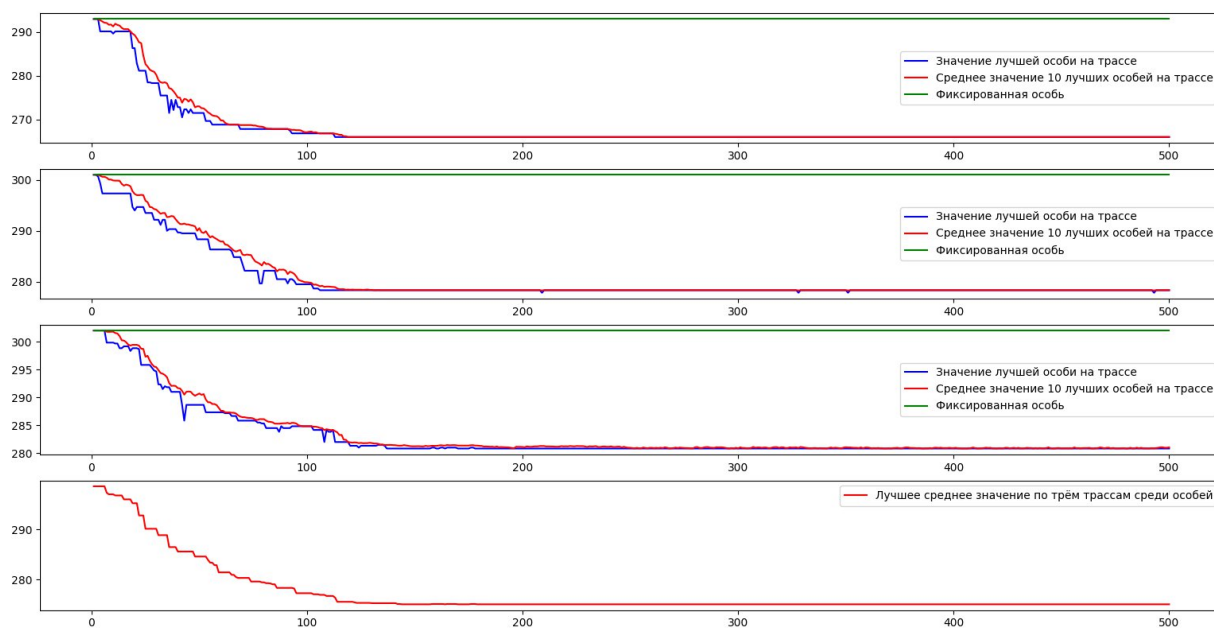
## 2.5 Результаты работы алгоритма

После прохождения 500 поколений от каждой группы особей были получены следующие результаты. Каждый рисунок является результатом группы особей с определённым типом трасс.



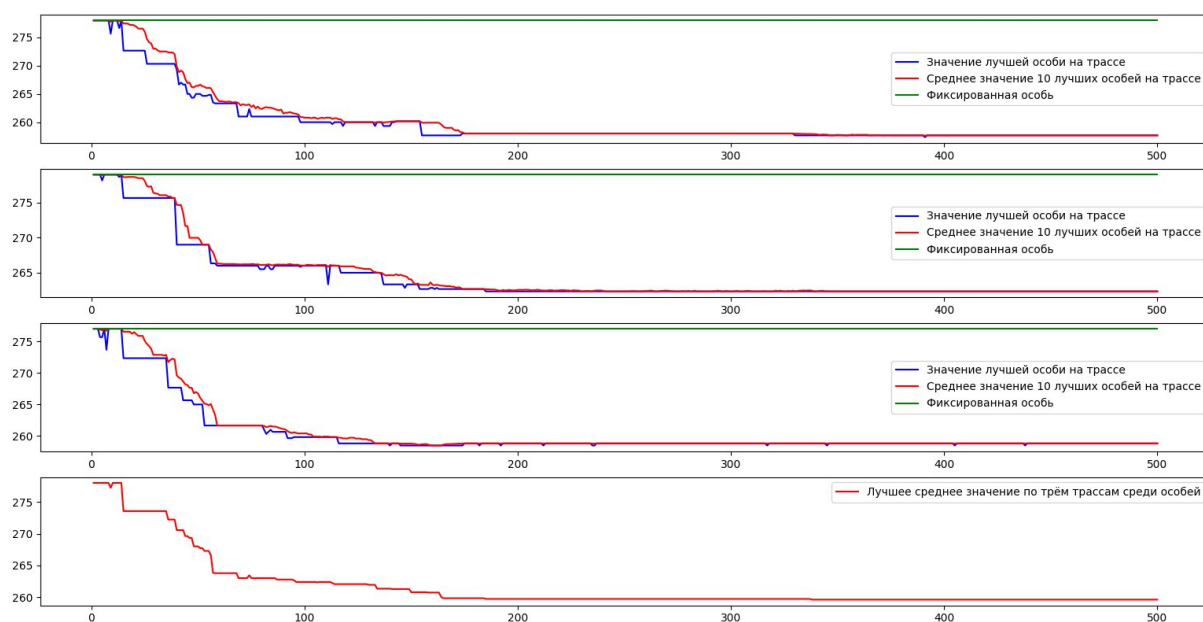
**Рис. 7 – Равнинные трассы**

Первые три графика отображают результат прохождения по конкретной трассе группой особей. Последний график отображает лучшее среднее значение по трём трассам среди всех особей группы.



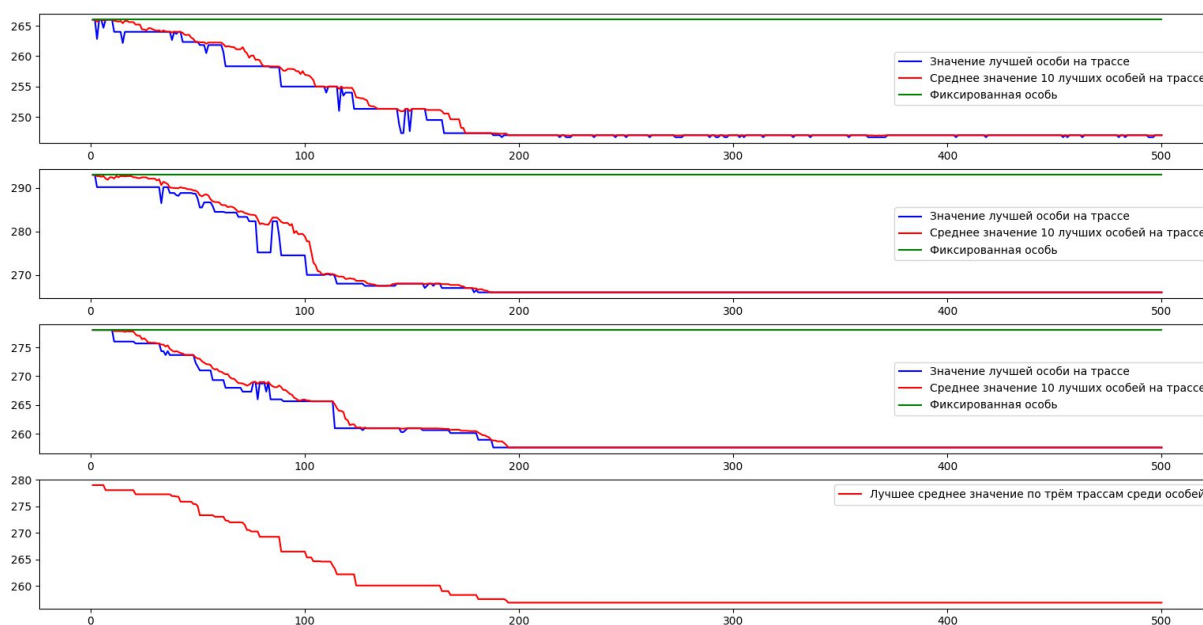
**Рис. 8 – Горные трассы**

График зелёного цвета отображает особь, которая двигалась всю трассу с одинаковой начальной скоростью (40 км/ч). График красного цвета отображает среднее значение 10 лучших особей на трассе. График синего цвета отображает результат лучшей особи на трассе в каждом поколении.



**Рис. 9 – Смешанные трассы**

Скачки на графиках трасс возникают по причине того, что особь показала хороший результат только на одной трассе, поэтому в общем рейтинге она попадает во вторую половину списка и удаляется в следующем поколении.



**Рис.10 – Все трассы**

В 500-м поколении у особей с лучшим результатом прохождения всех типовых трасс конкретной группы получились следующие генотипы.

Таблица 6

## Генотипы лучших особей из 500 поколения по всем типам трасс

Номер гена	Тек. скорость	Тек. участок трассы	След. участок трассы	Равнин. трассы	Горные трассы	Смешанные трассы	Все трассы
1	20	-2	-2	0	0	0	20
2	20	-2	-1	20	0	20	20
3	20	-2	0	20	20	0	20
4	20	-2	1	0	0	0	20
5	20	-2	2	0	0	20	20
6	20	-1	-2	0	20	0	20
7	20	-1	-1	0	20	20	20
8	20	-1	0	0	20	0	20
9	20	-1	1	0	20	0	0
10	20	-1	2	0	20	0	0
11	20	0	-2	0	20	20	20
12	20	0	-1	20	20	20	20
13	20	0	0	20	20	20	20
14	20	0	1	20	20	20	20
15	20	0	2	20	20	0	20
16	20	1	-2	0	0	0	0
17	20	1	-1	20	20	20	20
18	20	1	0	0	20	0	0
19	20	1	1	0	20	0	20
20	20	1	2	20	0	0	20
21	20	2	-2	20	0	0	20
22	20	2	-1	0	0	20	0
23	20	2	0	20	0	0	0
24	20	2	1	0	20	0	20
25	20	2	2	0	20	0	0
26	40	-2	-2	-20	20	-20	20
27	40	-2	-1	-20	20	20	20
28	40	-2	0	20	20	20	20
29	40	-2	1	0	0	0	-20
30	40	-2	2	0	-20	0	0
31	40	-1	-2	-20	0	0	20
32	40	-1	-1	20	0	0	-20
33	40	-1	0	20	20	20	20
34	40	-1	1	0	0	0	0
35	40	-1	2	-20	-20	0	20
36	40	0	-2	-20	0	0	-20
37	40	0	-1	0	0	0	0
38	40	0	0	-20	-20	-20	-20
39	40	0	1	0	0	0	0
40	40	0	2	0	0	-20	0
41	40	1	-2	-20	0	0	0
42	40	1	-1	0	-20	-20	0
43	40	1	0	-20	-20	-20	-20
44	40	1	1	20	0	-20	20
45	40	1	2	20	0	0	0
46	40	2	-2	-20	0	0	-20
47	40	2	-1	20	-20	-20	-20
48	40	2	0	-20	-20	-20	-20
49	40	2	1	0	-20	-20	0



50	40	2	2	-20	-20	20	0
51	60	-2	-2	0	-20	-20	0
52	60	-2	-1	0	0	-20	-20
53	60	-2	0	0	0	0	0
54	60	-2	1	-20	0	0	0
55	60	-2	2	0	0	0	0
56	60	-1	-2	-20	-20	-20	-20
57	60	-1	-1	0	0	0	-20
58	60	-1	0	0	0	0	0
59	60	-1	1	0	0	0	-20
60	60	-1	2	0	0	0	0
61	60	0	-2	0	-20	-20	-20
62	60	0	-1	0	-20	-20	-20
63	60	0	0	-20	0	-20	-20
64	60	0	1	-20	-20	-20	-20
65	60	0	2	-20	-20	-20	0
66	60	1	-2	0	0	0	-20
67	60	1	-1	0	-20	0	0
68	60	1	0	0	-20	-20	0
69	60	1	1	0	0	-20	0
70	60	1	2	0	0	0	-20
71	60	2	-2	-20	0	-20	-20
72	60	2	-1	0	-20	-20	0
73	60	2	0	-20	0	0	0
74	60	2	1	0	0	0	0
75	60	2	2	0	0	-20	0

Рассмотрим конкретные гены среди всех лучших особей и предположим гипотезы к таким ситуациям. В гене №16 можно заметить, что все особи оставляют текущую скорость без изменений перед резким спуском после небольшого подъёма, но при этом в гене №17 все особи увеличивают скорость, если следующий участок представляет собой небольшой спуск. Такая закономерность наблюдается только при скорости 20 км/ч.

На участке «Ровная поверхность» почти во всех случаях особи всех типов принимают решение увеличить скорость перед следующим участком трассы. Можно сделать вывод, что участок «Ровная поверхность» при скорости 20 км/ч отлично подходит для ускорения независимо от следующего участка трассы. Однако в генах №36-40 видно, что не следует увеличивать текущую скорость до 60 км/ч, а в некоторых случаях её можно снизить с 40 до 20 км/ч. В генах №61-65 также в большинстве случаев наблюдается резкое снижение скорости до 40 км/ч.

Интересное наблюдение касается генов №1, 4, 29, 57, 59, 65, 66, 70. У этих генов одинаковые значения для всех типов трасс, кроме «Все трассы». Это показывает, насколько генотип оптимального результата может отличаться при разном наборе даже одинаковых трасс, а также насколько разные виды трасс одного типа могут влиять на генотип оптимального результата.

У генов типа «Горные трассы» явно заметно стремление увеличить скорость с 20 км/ч до 40 км/ч, а не оставлять её без изменений, в отличие от всех других типов трасс. Можно сделать вывод, что для трасс с сильными подъёмами и спусками скорости 20 км/ч зачастую недостаточно для оптимального прохождения.

Можно сделать общий вывод по всем генам каждого типа трасс. Больше всего увеличивала скорость особь из группы «Все трассы», а меньше всего особь из группы «Смешанные трассы». Без изменений скорости больше всего у особи из группы «Равнинные трассы», а меньше всего у особи из группы «Все трассы». Понижала текущую скорость больше всего особь из группы «Смешанные трассы», а меньше всего особи из групп «Равнинные трассы» и «Горные трассы». Эти результаты сильно зависят от трасс, к которым адаптировались особи, и от процесса адаптации на протяжении всех поколений, поэтому они неоднозначны и могут измениться при другом наборе трасс для всех типовых групп или при повторном запуске генетического алгоритма.

По результатам работы алгоритма для всех 4-х групп можно увидеть стремление особей к адаптации к определённым трассам и получение оптимального значения, с которым преодолевается та или иная трасса. Это подтверждает корректность работы алгоритма в рамках заданных условий. Можно заметить, что большинство особей адаптируется уже к 200-му поколению. Это особенно хорошо видно на итоговых графиках для каждой группы.

Если сравнить работу данного генетического алгоритма с методом перебора, можно заметить его явное преимущество в эффективности. Общее число возможных особей  $12^{25}$ , так как количество возможных генотипов именно таково. Поскольку почти на всех типах трасс оптимальный результат достигается к 200-му поколению, к этому моменту было проанализировано  $50 + 25 \times 199 = 5025$  особей, что составляет крайне малую долю от общего числа возможных вариантов. Такие результаты работы алгоритма позволяют перейти к дальнейшим исследованиям.

## 2.6 Выводы

Разработанный генетический алгоритм проверен на работоспособность в соответствии с поставленными условиями задачи. Продемонстрирована требуемая эффективность. Весь код написан на Python в кроссплатформенной интегрированной среде разработки PyCharm. Каждый этап алгоритма представляет собой функцию. Некоторые основные функции включают дополнительные вспомогательные функции для вычислений, а также позволяют удобно изменять или добавлять условия выполнения того или иного этапа алгоритма. Каждая основная функция состоит от 20 до 50 строк кода. Количество основных функций соответствует количеству этапов алгоритма. Все функции вызываются в главной функции, которая структурно представляет собой сам алгоритм. Эта функция вызывается для каждого типа трассы отдельно, что позволяет тестировать алгоритм выборочно, а не для всех типов сразу, если это необходимо.

После разработки алгоритма и получения результатов можно сделать выводы о корректности его работы и возможности дальнейшего масштабирования. Можно добавить условия для большей реалистичности, например, учесть качество покрытия трассы, погоду, пробки или плотность движения.

## МОДИФИКАЦИЯ АЛГОРИТМА

### 3.1 Дополнительные условия для задачи

Представленный выше алгоритм ориентирован на решение задачи без учета дополнительных ограничений для транспортного средства. Теперь мы усложним модель, введя новые параметры и условия для прохождения трассы, чтобы проверить, насколько эффективно алгоритм сможет адаптироваться к измененным требованиям и какие результаты покажет в новых условиях.

В обновленной постановке задачи учтены два новых фактора: влияние погодных условий на прохождение трассы и риск перегрева двигателя. Теперь транспортное средство использует не бензин, а электричество, частично получаемое через солнечные батареи. При этом начального заряда аккумулятора достаточно для полного прохождения трассы, поэтому солнечные панели влияют лишь на отдельные особенности движения, но не на принципиальную возможность финишировать. Всего определено 3 вида погоды: солнечная, облачная и пасмурная. Рассмотрим каждое погодное условие отдельно.

При солнечной погоде возникает риск перегрева двигателя, если транспортное средство движется со скоростью 60 км/ч на двух последовательных участках трассы. В случае перегрева после прохождения второго участка происходит обязательная остановка для охлаждения. Время простоя рассчитывается как утроенное максимальное время прохождения одного участка, что в алгоритме составляет 0.15 условных единиц. Ограничений на количество возможных перегревов в течение трассы не предусмотрено, поэтому данная ситуация может возникать многократно.

При облачной погоде транспортное средство получает возможность движения на 60 км/ч при выполнении одного из двух условий: либо если солнце появлялось хотя бы один раз ранее, либо если с начала трассы еще не было участков с такой скоростью. Для формирования появления солнца

генерируется случайный булев массив длиной, равной количеству участков трассы, где с вероятностью 20% будет появление солнца на соответствующем участке трассы, но с запретом на два солнечных участка подряд. При появлении солнца транспорт может развить скорость 60 км/ч на текущем или следующем участке (согласно генотипу), причем данный эффект не накапливается, поэтому если солнце появлялось несколько раз, то транспортное средство может проехать участок трассы со скоростью 60 км/ч только один раз до следующего появления солнца.

В условиях пасмурной погоды вводятся ограничения на скоростное прохождение трассы: транспортному средству разрешается преодолеть не более двух участков со скоростью 60 км/ч, при этом только первые два участка трассы (при наличии соответствующей записи в генотипе). Все последующие участки, где определено движение на скорости 60 км/ч, будут автоматически пройдены со сниженной скоростью 40 км/ч.

Такие дополнительные условия интегрируются в существующую постановку задачи, и это позволяет модифицировать ранее созданный генетический алгоритм с минимальными изменениями, наглядно демонстрируя гибкость и масштабируемость генетического подхода.

### **3.2 Описание реализации**

Для реализации новых условий потребовались минимальные изменения в структуре алгоритма. Общее количество основных этапов осталось неизменным. Модификации подверглись лишь этапы, отвечающие за «Прохождение трассы» и «Рейтинг особей». Также в качестве дополнения появился новый циклический процесс, добавленный в алгоритм для обработки различных погодных условий. При этом базовая архитектура решения осталась неизменной.

На этапе «Прохождение трассы» были добавлены три функции, каждая из которых соответствует определенному погодному условию. Эти функции сохраняют базовую логику алгоритма движения по трассе, но адаптируют её

под особенности каждого погодного условия. Такой подход обеспечивает разделение функционала в зависимости от погодных условий, что позволяет удобно и наглядно реализовывать поведение для каждого случая.

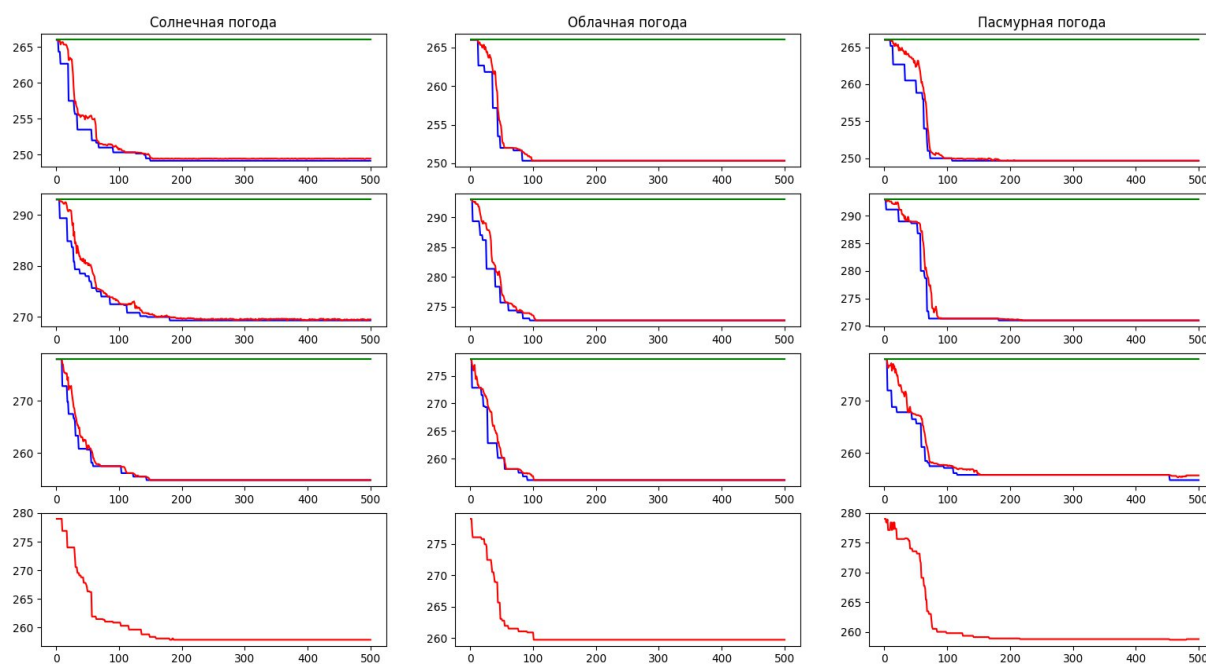
Модификации этапа «Рейтинг особей» были направлены на устранение проблемы алгоритма, проявлявшейся в резких скачках результатов на графиках. Основная причина заключалась в том, что некоторые особи демонстрировали высокую эффективность лишь на отдельных трассах, из-за чего не попадали в следующее поколение по общему рейтингу. Было решено для каждой особи, которая заняла первое место среди других особей по конкретной трассе, начислять баллы, равные количеству особей в популяции, умноженному на количество всех трасс. Такое изменение позволило сгладить графики результатов, сохранив при этом качество получаемых решений. Однако следует отметить, что данный подход позволяет проходить в следующее поколение особям, не обладающим оптимальным генотипом для адаптации к типам трасс.

Внедрение дополнительных условий потребовало лишь добавления новых строк кода, не затрагивая существенно основную структуру алгоритма. Такой подход наглядно демонстрирует ключевое преимущество генетических алгоритмов – их гибкость и масштабируемость при условии наличия грамотно спроектированной структуры алгоритма.

### **3.3 Результаты модифицированного алгоритма**

Результаты исследования сохранили прежние критерии оценки; все графические показатели имеют ту же интерпретацию, что и у предыдущих графиков. Количество графиков увеличилось до 12 для каждого типа трасс. Столбцы отражают различные погодные условия, а строки (кроме последней) соответствуют конкретным структурам трасс для данного типа. Последняя строка отображает наилучший средний результат среди всех особей для каждого вида трасс и каждого погодного условия. Введённые модификации оказали незначительное влияние на динамику оптимизации. В

большинстве случаев оптимальный результат достигается уже на 200-м поколении. Ниже представлены результаты для типа трасс «Все трассы» по каждому погодному условию, а данные по остальным типам трасс приведены в Приложении (А-В). Также в Приложении (Г-Е) приведены результаты лучших генотипов по каждому типу трассы для каждого погодного условия.



**Рис. 11** – Все трассы с погодными условиями

### 3.4 Выводы

При модификации генетического алгоритма под новые условия задачи удалось обойтись минимальными изменениями в существующем коде. Данная модификация наглядно иллюстрирует преимущества генетического алгоритма: высокую степень масштабируемости и простоту модификации при изменении или дополнении условий. Залогом такой гибкости является грамотный выбор параметров и методов для каждого этапа генетического алгоритма. Это позволяет в дальнейшем вносить точечные корректировки без переработки всей структуры, что не только ускоряет процесс доработки, но и снижает вероятность появления ошибок, так как основные механизмы алгоритма остаются нетронутыми. Таким образом, сочетание масштабируемости и адаптивности делает генетический алгоритм

эффективным инструментом для решения задач, в которых важно учитывать изменяющиеся условия без потери качества и скорости получения оптимального результата.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы был изучен материал по теме «Генетические алгоритмы», рассмотрены особенности настройки методов и параметров для генетического алгоритма, его преимущества перед другими алгоритмами оптимизации. Также был изучен материал, связанный с системой адаптивного круиз-контроля. Данные знания позволили реализовать собственный генетический алгоритм для задачи, связанной с прохождением дистанции транспортным средством с системой адаптивного круиз-контроля с оптимальным результатом по затратам топлива и времени. Удалось достичь оптимального результата работы алгоритма за небольшое количество итераций и при малом количестве рассмотренных вариантов решения для поставленной задачи, что является явным преимуществом использования генетического алгоритма для данной задачи оптимизации.

В данной работе был продемонстрирован ещё один плюс генетического алгоритма, заключающийся в масштабируемости и адаптивности к новым условиям задачи. Такими дополнениями стали погодные условия для прохождения дистанции и возможность перегрева двигателя транспортного средства. Новые условия не оказали сильного влияния на изменение получаемого результата работы алгоритма, но при этом позволили усложнить поставленную задачу, что отражает качество подобранных методов и параметров для генетического алгоритма.

Стоит отметить, что примеров решения данной задачи (или подобных ей) с помощью генетического алгоритма не было найдено нами в интернет-источниках; таким образом, по всей видимости, данная постановка задачи и её реализация являются уникальными.

Перспективы дальнейшего исследования могут заключаться в усложнении условий для более реалистичной модели, адаптации к реальным транспортным дистанциям, моделирования потока движения, а не только

единичного транспортного средства, а также в усовершенствовании алгоритма с использованием других методов и параметров.

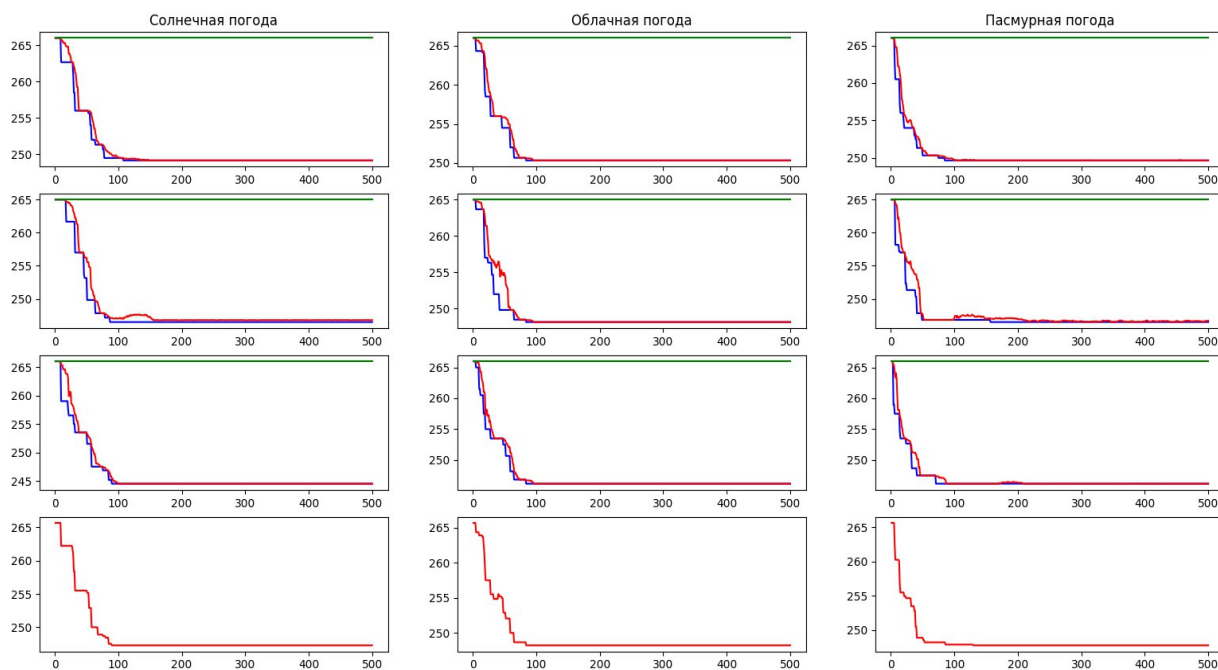
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

- 1) A Field Guide to Genetic Programming [Электронный ресурс] / Riccardo Poli / William B. Langdon / Nicholas F. McPhee – URL: [https://www.researchgate.net/publication/216301261\\_A\\_Field\\_Guide\\_to\\_Genetic\\_Programming](https://www.researchgate.net/publication/216301261_A_Field_Guide_to_Genetic_Programming)
- 2) Генетические алгоритмы [Электронный ресурс] / Т.В. Панченко // Изд-во Астраханский университет 2007 – URL: <https://masters.donntu.ru/2019/fknt/domina/lib/genalg.pdf?ysclid=mabkazue1k323368727>
- 3) Генетические алгоритмы [Электронный ресурс] – URL: <https://algolist.ru/ai/ga/index.html>
- 4) Круиз-контроль [Электронный ресурс] – URL: <https://dzen.ru/a/X8IKu9V-6SdSJYyA?ysclid=maa6v2ntx6563987902>
- 5) Кто придумал круиз-контроль? [Электронный ресурс] – URL: <https://www.drive2.ru/b/464635747907928116/?ysclid=maa6tu2wa9529548020>
- 6) Обзор методов отбора, скрещивания и мутации [Электронный ресурс] – URL: <https://proporprogs.ru/ga/ga-obzor-metodov-otbora-skreshchivaniya-i-mutacii?ysclid=mabgda3zkw335217791>
- 7) Genetic Algorithms and its use-cases in Machine Learning [Электронный ресурс] – URL: <https://www.analyticsvidhya.com/blog/2021/06/genetic-algorithms-and-its-use-cases-in-machine-learning/>
- 8) Генетический алгоритм vs алгоритм роя частиц [Электронный ресурс] – URL: <https://habr.com/ru/articles/551890/>
- 9) Сравнительное исследование классических методов оптимизации и генетических алгоритмов [Электронный ресурс] – URL: <https://cyberleninka.ru/article/n/sravnitelnoe-issledovanie-klassicheskikh-metodov-optimizatsii-i-geneticheskikh-algoritmov/viewer>
- 10) Сравнение генетического алгоритма и алгоритма оптимизации роя частиц на примере одной задачи структурной оптимизации

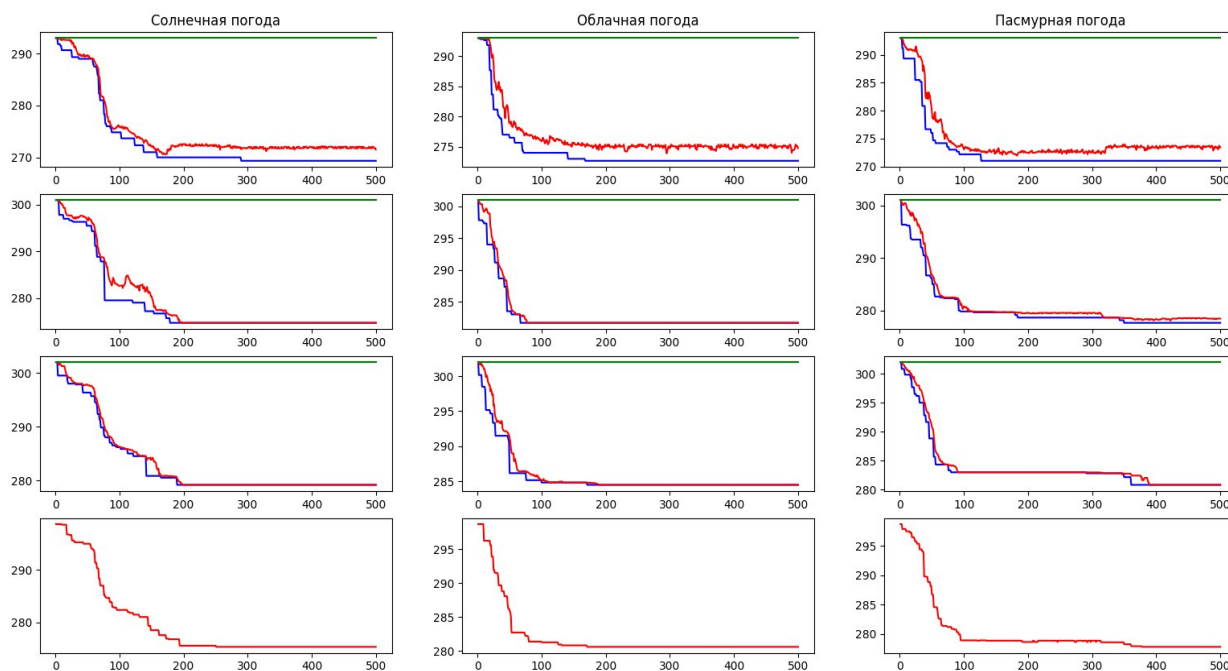
[Электронный ресурс] – URL: [https://masters.donntu.ru/2020/fknt/pashchenko/library/my\\_article2.htm?ysclid=mahfm9fiyk910247291](https://masters.donntu.ru/2020/fknt/pashchenko/library/my_article2.htm?ysclid=mahfm9fiyk910247291)

- 11) Как генетический алгоритм находит решения [Электронный ресурс] – URL: <https://propoprogs.ru/ga/ga-kak-geneticheskiy-algoritm-nahodit-resheniya>
- 12) Racing with genetic algorithms [Электронный ресурс] – URL: <https://www.ihomer.nl/en/blog/racen-met-genetische-algoritmen>
- 13) Evolutionary F1 Race [Электронный ресурс] – URL: [Strategyhttps://dl.acm.org/doi/pdf/10.1145/3583133.3596349](https://dl.acm.org/doi/pdf/10.1145/3583133.3596349)
- 14) Петров Н.С. Разработка генетического алгоритма адаптивного круиз-контроля / Н.С. Петров // Путь в науку: прикладная математика, информатика и информационные технологии – 2025

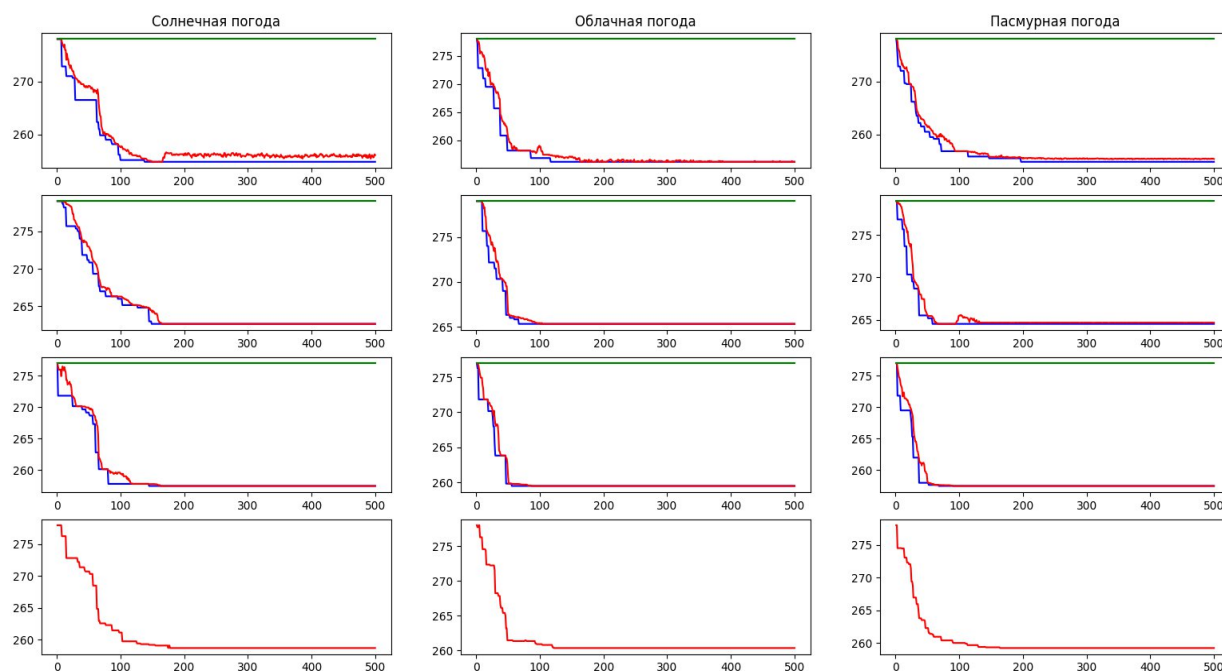
## Результат по Равнинным трассам с погодными условиями



## Результат по Горным трассам с погодными условиями



## Результат по Смешанным трассам с погодными условиями



### Лучшие генотипы при солнечной погоде по всем типам трасс

Номер гена	Тек. скорость	Тек. участок трассы	След. участок трассы	Равнин. трассы	Горные трассы	Смешанные трассы	Все трассы
1	20	-2	-2	0	0	20	0
2	20	-2	-1	0	20	20	20
3	20	-2	0	20	20	0	20
4	20	-2	1	20	0	0	0
5	20	-2	2	20	20	20	20
6	20	-1	-2	0	0	20	20
7	20	-1	-1	0	20	0	0
8	20	-1	0	20	20	0	20
9	20	-1	1	20	20	20	0
10	20	-1	2	20	20	0	20
11	20	0	-2	0	20	20	20
12	20	0	-1	20	20	20	20
13	20	0	0	20	20	20	20
14	20	0	1	20	20	20	20
15	20	0	2	20	20	0	20
16	20	1	-2	20	0	0	0
17	20	1	-1	0	20	20	20
18	20	1	0	20	0	0	20
19	20	1	1	0	0	0	0
20	20	1	2	0	20	0	0
21	20	2	-2	20	20	0	0
22	20	2	-1	20	0	0	0
23	20	2	0	0	0	20	0
24	20	2	1	0	0	0	0
25	20	2	2	20	0	20	0
26	40	-2	-2	20	20	0	-20
27	40	-2	-1	20	0	0	0
28	40	-2	0	20	20	20	-20
29	40	-2	1	-20	0	-20	0
30	40	-2	2	0	20	0	0
31	40	-1	-2	-20	0	0	0
32	40	-1	-1	0	0	0	0
33	40	-1	0	20	20	20	20
34	40	-1	1	0	0	0	-20
35	40	-1	2	0	0	0	0
36	40	0	-2	-20	-20	20	-20
37	40	0	-1	0	-20	-20	0
38	40	0	0	-20	-20	-20	-20
39	40	0	1	-20	-20	0	0
40	40	0	2	0	-20	0	0
41	40	1	-2	20	-20	0	-20
42	40	1	-1	-20	-20	0	0
43	40	1	0	-20	-20	-20	-20
44	40	1	1	0	0	-20	20
45	40	1	2	20	-20	0	0
46	40	2	-2	0	0	0	-20
47	40	2	-1	-20	-20	-20	-20



48	40	2	0	-20	-20	-20	-20
49	40	2	1	0	0	-20	0
50	40	2	2	0	-20	0	0
51	60	-2	-2	0	-20	0	0
52	60	-2	-1	-20	-20	0	0
53	60	-2	0	-20	0	-20	0
54	60	-2	1	0	-20	0	-20
55	60	-2	2	-20	0	0	0
56	60	-1	-2	0	-20	-20	0
57	60	-1	-1	0	-20	0	-20
58	60	-1	0	-20	-20	-20	-20
59	60	-1	1	-20	0	-20	-20
60	60	-1	2	-20	-20	0	0
61	60	0	-2	0	-20	-20	0
62	60	0	-1	-20	-20	-20	-20
63	60	0	0	-20	-20	-20	-20
64	60	0	1	-20	-20	-20	-20
65	60	0	2	0	-20	-20	-20
66	60	1	-2	0	-20	0	-20
67	60	1	-1	-20	0	0	-20
68	60	1	0	-20	-20	-20	0
69	60	1	1	0	-20	0	0
70	60	1	2	-20	0	-20	-20
71	60	2	-2	0	-20	0	0
72	60	2	-1	-20	0	0	0
73	60	2	0	-20	0	0	0
74	60	2	1	-20	-20	0	0
75	60	2	2	0	-20	0	0

### Лучшие генотипы при облачной погоде по всем типам трасс

Номер гена	Тек. скорость	Тек. участок трассы	След. участок трассы	Равнин. трассы	Горные трассы	Смешанные трассы	Все трассы
1	20	-2	-2	0	20	20	0
2	20	-2	-1	20	20	0	20
3	20	-2	0	20	20	0	20
4	20	-2	1	20	20	20	0
5	20	-2	2	0	0	0	20
6	20	-1	-2	0	20	0	20
7	20	-1	-1	0	20	0	20
8	20	-1	0	20	20	20	20
9	20	-1	1	0	20	20	0
10	20	-1	2	0	0	20	20
11	20	0	-2	0	20	20	20
12	20	0	-1	20	20	20	20
13	20	0	0	20	20	20	20
14	20	0	1	20	20	20	20
15	20	0	2	20	20	0	20
16	20	1	-2	20	0	0	20
17	20	1	-1	0	20	20	20
18	20	1	0	0	0	20	0
19	20	1	1	20	0	0	20
20	20	1	2	0	20	20	20
21	20	2	-2	0	0	0	0
22	20	2	-1	0	0	0	0
23	20	2	0	0	0	0	20
24	20	2	1	20	0	0	20
25	20	2	2	0	0	0	20
26	40	-2	-2	0	0	0	20
27	40	-2	-1	0	0	0	0
28	40	-2	0	-20	20	0	0
29	40	-2	1	20	0	0	0
30	40	-2	2	0	20	0	0
31	40	-1	-2	0	0	20	20
32	40	-1	-1	20	20	0	0
33	40	-1	0	20	0	0	0
34	40	-1	1	20	0	0	0
35	40	-1	2	-20	0	0	0
36	40	0	-2	-20	-20	0	-20
37	40	0	-1	-20	-20	-20	-20
38	40	0	0	-20	-20	-20	-20
39	40	0	1	0	-20	0	0
40	40	0	2	0	0	20	0
41	40	1	-2	-20	20	0	20
42	40	1	-1	0	0	-20	0
43	40	1	0	-20	-20	-20	-20
44	40	1	1	0	0	0	0
45	40	1	2	0	-20	-20	20
46	40	2	-2	0	0	0	-20
47	40	2	-1	-20	-20	-20	-20

48	40	2	0	-20	-20	-20	-20
49	40	2	1	0	0	0	0
50	40	2	2	0	-20	0	0
51	60	-2	-2	-20	0	0	-20
52	60	-2	-1	-20	0	0	-20
53	60	-2	0	-20	0	0	0
54	60	-2	1	0	0	0	0
55	60	-2	2	0	-20	0	0
56	60	-1	-2	0	0	0	0
57	60	-1	-1	0	-20	-20	-20
58	60	-1	0	0	-20	0	-20
59	60	-1	1	-20	-20	0	0
60	60	-1	2	0	0	0	-20
61	60	0	-2	0	0	-20	0
62	60	0	-1	-20	0	0	-20
63	60	0	0	0	0	0	-20
64	60	0	1	0	0	0	0
65	60	0	2	0	0	0	-20
66	60	1	-2	0	0	-20	0
67	60	1	-1	-20	-20	-20	0
68	60	1	0	0	0	0	-20
69	60	1	1	0	0	-20	0
70	60	1	2	0	-20	0	0
71	60	2	-2	-20	-20	0	-20
72	60	2	-1	0	0	0	0
73	60	2	0	-20	0	0	0
74	60	2	1	0	0	0	0
75	60	2	2	0	0	0	0

### Лучшие генотипы при пасмурной погоде по всем типам трасс

Номер гена	Тек. скорость	Тек. участок трассы	След. участок трассы	Равнин. трассы	Горные трассы	Смешанные трассы	Все трассы
1	20	-2	-2	0	0	0	20
2	20	-2	-1	0	20	20	0
3	20	-2	0	20	20	20	20
4	20	-2	1	0	0	0	0
5	20	-2	2	0	0	0	20
6	20	-1	-2	20	0	0	20
7	20	-1	-1	0	0	20	20
8	20	-1	0	20	20	20	20
9	20	-1	1	20	20	20	0
10	20	-1	2	20	0	20	20
11	20	0	-2	20	20	20	20
12	20	0	-1	20	20	20	20
13	20	0	0	20	20	20	20
14	20	0	1	20	20	20	20
15	20	0	2	20	20	0	20
16	20	1	-2	0	0	20	20
17	20	1	-1	0	0	0	0
18	20	1	0	20	0	20	20
19	20	1	1	20	0	20	0
20	20	1	2	20	0	20	20
21	20	2	-2	0	20	0	0
22	20	2	-1	0	0	0	0
23	20	2	0	20	0	0	0
24	20	2	1	0	0	20	0
25	20	2	2	0	0	20	20
26	40	-2	-2	-20	0	0	20
27	40	-2	-1	20	0	20	20
28	40	-2	0	20	20	0	20
29	40	-2	1	0	-20	0	0
30	40	-2	2	0	0	20	-20
31	40	-1	-2	0	0	0	0
32	40	-1	-1	20	0	20	-20
33	40	-1	0	20	20	20	-20
34	40	-1	1	0	0	-20	20
35	40	-1	2	0	20	0	0
36	40	0	-2	0	-20	0	-20
37	40	0	-1	-20	-20	-20	-20
38	40	0	0	-20	-20	-20	-20
39	40	0	1	0	0	0	0
40	40	0	2	0	0	0	0
41	40	1	-2	20	0	0	-20
42	40	1	-1	20	-20	-20	20
43	40	1	0	-20	-20	-20	-20
44	40	1	1	-20	-20	0	20
45	40	1	2	0	-20	20	0
46	40	2	-2	0	0	-20	-20
47	40	2	-1	-20	-20	-20	-20

48	40	2	0	-20	-20	-20	-20
49	40	2	1	20	0	0	0
50	40	2	2	-20	-20	0	0
51	60	-2	-2	0	0	-20	0
52	60	-2	-1	0	0	0	-20
53	60	-2	0	0	-20	0	0
54	60	-2	1	0	0	0	-20
55	60	-2	2	0	0	0	-20
56	60	-1	-2	-20	0	0	0
57	60	-1	-1	-20	0	0	0
58	60	-1	0	0	0	-20	-20
59	60	-1	1	0	-20	0	-20
60	60	-1	2	0	0	0	0
61	60	0	-2	-20	-20	-20	-20
62	60	0	-1	0	-20	0	-20
63	60	0	0	-20	0	-20	-20
64	60	0	1	0	-20	-20	-20
65	60	0	2	-20	-20	0	0
66	60	1	-2	-20	-20	-20	-20
67	60	1	-1	0	0	0	0
68	60	1	0	-20	0	0	-20
69	60	1	1	0	0	-20	0
70	60	1	2	0	0	-20	0
71	60	2	-2	0	0	0	-20
72	60	2	-1	-20	0	0	0
73	60	2	0	0	-20	-20	-20
74	60	2	1	0	-20	0	0
75	60	2	2	-20	0	-20	0