



**«Московский государственный технический  
университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ      Информатика и системы управления  
КАФЕДРА        Компьютерные системы и сети (ИУ6)

**РАСЧЁТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к научно-исследовательской работе на тему:**

**«Анализ генетических алгоритмы с использованием цифровых  
автоматов»**

Студент ИУ6-75  
(группа)

\_\_\_\_\_  
(Подпись, дата)

**В.Д. Шульман**  
(И.О.Фамилия)

Руководитель работы

\_\_\_\_\_  
(Подпись, дата)

**О.Ю. Еремин**  
(И.О.Фамилия)

## Оглавление

Реферат .....	4
Определения, обозначения и сокращения .....	5
Введение .....	7
<b>1. Актуальность НИР, цели и задачи .....</b>	<b>8</b>
1.1. Актуальность генетических алгоритмов .....	8
1.2. Цели и решаемые задачи.....	9
<b>2. Сравнение генетических алгоритмов с детерминированными алгоритмами.....</b>	<b>9</b>
<b>3. Классификация генетических алгоритмов .....</b>	<b>10</b>
3.1. Стационарные генетические алгоритмы .....	11
3.2. Динамические генетические алгоритмы .....	12
3.3. Поколенческие генетические алгоритмы .....	12
3.4. Адаптивные генетические алгоритмы.....	12
3.5. Многоуровневые генетические алгоритмы .....	13
3.6. Параллельные генетические алгоритмы .....	13
<b>4. Анализ предметной области .....</b>	<b>14</b>
4.1. Основные объекты предметной области .....	15
4.1.1. Агент.....	15
4.1.2. Хранилище генов.....	15
4.2. Основные информационные структуры предметной области .....	16
4.2.1. Генетический код .....	16
4.2.2. Генетическое древо .....	16
4.2.3. Поле визуализации.....	16
4.3. Основные процедуры предметной области .....	17
4.3.1. Генетический алгоритм .....	17
4.3.2. Метод отбора .....	17
4.3.3. Сеанс.....	17
<b>5. Необходимые информационные технологии .....</b>	<b>18</b>
5.1. Централизованная обработка информации .....	18
5.2. Децентрализованная обработка информации .....	18
5.3. Информационная технология поддержки принятия решений.....	19
<b>6. Понятие цифрового автомата как математической абстракции .....</b>	<b>19</b>
6.1. Классификация абстрактного автомата .....	20

6.2. Базовая модель конечного автомата .....	23
<b>7. Понятие клеточного автомата .....</b>	<b>24</b>
<b>8. Эволюционирующий клеточный автомат.....</b>	<b>27</b>
<b>Заключение.....</b>	<b>29</b>
<b>Список литературы .....</b>	<b>30</b>

## Реферат

Записка с. ?, рис. ?, табл. ?, источников ?, прил. А, Б, В, Г

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ, СЕЛЕКЦИОННЫЙ МЕТОД, ОТБОР, АГЕНТ, ХРАНИЛИЩЕ ГЕНОВ.

Рассматриваемыми объектами в данной исследовательской работы являются генетические алгоритмы и методы отбора при решении прикладных задач, где в основе лежит использование цифровых автоматов.

Основной целью исследования в данной области является поиск и нахождения примеров эффективного применения цифровых автоматов в генетических алгоритмах, а также формулирование основных принципов при построении моделей и использованием цифровых автоматов. В работе рассматриваются различные способы представления цифрового автомата и также подходы к его модификации в процессе работы генетического алгоритма. Также в данном исследовании рассматриваются различные подходы к выработке метода отбора и выявлению оптимальны параметров для него.

## **Определения, обозначения и сокращения**

Генетический алгоритм – алгоритм, осуществляющий поиск решения задачи оптимизации или моделирования путем случайного комбинирования параметров и отбора лучших решений.

Селекция – процесс, в рамках которого производится отбор по определенным критерия наиболее оптимального решения.

Селекционный метод – алгоритм, согласно которому производится оценка решений и формирование нового поколения

Агент – элементарная эволюционная единица, осуществляющая решение поставленной перед ней задачи

Популяция – множество агентов

Поколение – текущая популяция агентов, сформированная путем на основании предыдущих поколений, прошедших через селекционный отбор.

Жизненный цикл – период времени от появления агента в системе до его исключения из системы в результате работы селекционного алгоритма.

Генетический код – набор параметров и инструкций, которые выполняют агент в процессе своего жизненного цикла.

Мутация – случайное изменение генетического кода агента, приводящее к приближению или отдалению агента от оптимального решения

Оптимальное решение – решение, которое по набору признаков является наиболее предпочтительным

ГА – генетический алгоритм

ПГА – параллельный генетический алгоритм

Эвристический алгоритм – алгоритм решения задачи, который не является гарантированно точным или оптимальным

Детерминированный алгоритм – алгоритмический процесс, который выдает уникальный и предопределенный результат для заданных входных данных

Цифровой автомат – математическая модель дискретного устройства, которое принимает и выдает сигналы, принимая различные состояния

АА – абстрактный автомат

КА – конечный автомат

БА – бесконечный автомат

ЭА – элементарный автомат

## Введение

На данный момент достаточно широкое распространение получили методы решения задач, основанные на использовании генетических алгоритмов. Эти методы в основном используются для решения задач, решение для которых либо слишком сложно найти аналитически, составив алгоритм, либо они являются настолько абстрактными, что решение для них в виде детерминированного алгоритма вовсе не существует. Генетические алгоритмы позволяют решать такие задачи.

Сама идея генетических алгоритмов не нова. Еще в 60-ых годах прошлого века начались первые симуляции эволюционных процессов на тогдашних маломощных огромных ЭВМ. В 70-ых годах была разработана концепция эволюционного программирования с использованием конечных автоматов. Достаточно большую популярность генетические алгоритмы обрели после экспериментов Джона Холланда с так называемыми клеточными автоматами

Однако, к сожалению, из-за скромных вычислительных мощностей длительно время генетические алгоритмы оставались в основном в теоретической области и почти не использовались для решения прикладных задач. Лишь к концу 80-годов с ростом вычислительных мощностей стало внедряться оборудование, которое включало в себя генетические алгоритмы для решения поставленных перед ними задач. В 89-ом году компания Axcelis выпустила свой первый продукт для персональных компьютеров, в котором применялись генетические алгоритмы. Носил он символичное название “Evolver”.

Сейчас существует много вариантов реализации генетических алгоритмов, каждый из которых больше подходит для решения определенных задач. Например, нейронные сети хорошо решают задачу распознавания изображений, но решить задачу самообучающегося ИИ нейронные сети так и не смогли, несмотря на то, что прошло несколько витков их популярности.

## **1. Актуальность НИР, цели и задачи**

Генетические алгоритмы (ГА) являются подклассами более общей группы методов, называемыми эволюционными алгоритмами, которые объединяют различные варианты использования эволюционных принципов. В отличие от других технологий оптимизации ГА содержат популяцию пробных решений, которые конкурентно управляются с помощью определенных операторов. Генетическим алгоритмам присуще итеративное обучение популяции индивидов.

Генетические алгоритмы относятся к классу эвристических алгоритмов поиска, т.е. они решают задачи, используя практические методы, которые не являются гарантированно точными и оптимальными, однако достаточно точны для решений поставленной перед ними задачи.

### **1.1. Актуальность генетических алгоритмов**

На волне популярности искусственного интеллекта генетические алгоритмы являются поводом для постоянных дискуссий и исследований. Несмотря на то, что ИИ представляет из себя целый комплекс различных систем, такие как системы распознавания, анализа, обучения, адаптации и т.п., генетические алгоритмы вполне могут решить часть задач, связанных с адаптацией и обучением.

Генетические алгоритмы актуальны на данный момент тем, что уже сейчас способны решать определенный спектр вполне реальных задач. Генетические алгоритмы могут использоваться для управления процессорами, балансируя нагрузку в многопроцессорной системе, они могут использоваться для решения задач по поиску наилучшего соотношения веса/прочности/размера/плотности, для решения задач по размещению объектов различных форм на определенной площади (трассировка плат или нарезка ткани с наименьшими потерями, как пример).

Один из недавних коммерческих примеров: израильская компания Schema разработала программный продукт Channeling для оптимизации



работы сотовой связи путем выбора оптимальной частоты, на которой будет вестись разговор. В основе этого программного продукта и используются генетические алгоритмы.

Генетические алгоритмы приобрели большую популярность на фоне новой волны подъема интереса к нейронным сетям. Сейчас довольно быстро развиваются технологии по распознаванию изображения и звука. Для обучения нейронных сетей также используется генетический алгоритм, где после каждого раунда нейронные сети сравниваются по своей эффективности, после чего часть из них отсеивается, а на основании оставшихся генерируется новое поколение нейронных сетей и измененными весами переходов.

Одно из больших достоинств генетических алгоритмов – это возможность распараллеливания вычислений при поиске решений. Т.к. агенты, которые пытаются решить задачу, чаще всего не должны взаимодействовать с друг-другом, то обработку каждого из них можно вести параллельно. Раньше это не давало больших преимуществ из-за одноядерных процессоров, но сейчас с развитием многопроцессорных архитектур и асинхронных языков программирования поиск решения с использованием генетических алгоритмов может проходить в разы быстрее.

## **1.2. Цели и решаемые задачи**

Основной целью НИР является решение задач по поиску оптимальных подходов к реализации генетических алгоритмов с использованием цифровых автоматов и определению круга задач, для которых использование такого вида генетических алгоритмов было бы оправдано и эффективно

## **2. Сравнение генетических алгоритмов с детерминированными алгоритмами**

Генетические алгоритмы применяются чаще всего для поиска не оптимального решения, а более лучшего по сравнению с имеющимся. Это обосновывается тем, что для сложной задачи чаще всего требуется найти не идеальное решение, а то решение. Которое удовлетворяло бы имеющимся

требованиям. При этом достижения идеального «оптимального» решения отходит на второй план. Однако при этом другие методы, ориентированные на нахождение оптимального решения, из-за их чрезвычайно высокой сложности становятся и вовсе нереализуемыми.

Основные отличия генетических алгоритмов от традиционных методов:

1) Генетические алгоритмы работают с кодами, в которых представлен набор параметров, напрямую зависящих от аргументов целевой функции. При этом интерпретация этих кодов происходит только перед началом работы алгоритма и после завершения для получения результата. В процессе работы редактирования кодов проходит независимо от их трактования, код рассматривается просто как битовая строка.

2) Для поиска генетический алгоритм использует несколько точек поискового пространства одновременно, а не переходит от точки к точке, как это делается в традиционных методах. Это позволяет преодолеть один из их недостатков – опасность попадания в локальный экстремум целевой функции, если она не является унимодальной, т.е. имеет несколько таких экстремумов.

3) Генетические алгоритмы в процессе работы не используют никакую дополнительную информацию. Это повышает скорость работы. Единственной используемой информацией область допустимых значений параметров и целевой функции в произвольной точке.

4) Генетический алгоритм использует как вероятностные правила для порождения новых точек, так и детерминированные правила для перехода от одних точек к другим. Одновременное использование элементов случайности и детерминированности дает значительно больший эффект, чем раздельное.

### **3. Классификация генетических алгоритмов**

Генетические алгоритмы являются частным случаем еще более обобщенной сферы – генетического моделирования (рис. 1), но генетические алгоритмы достаточно разнообразны и обладают собственной классификацией.

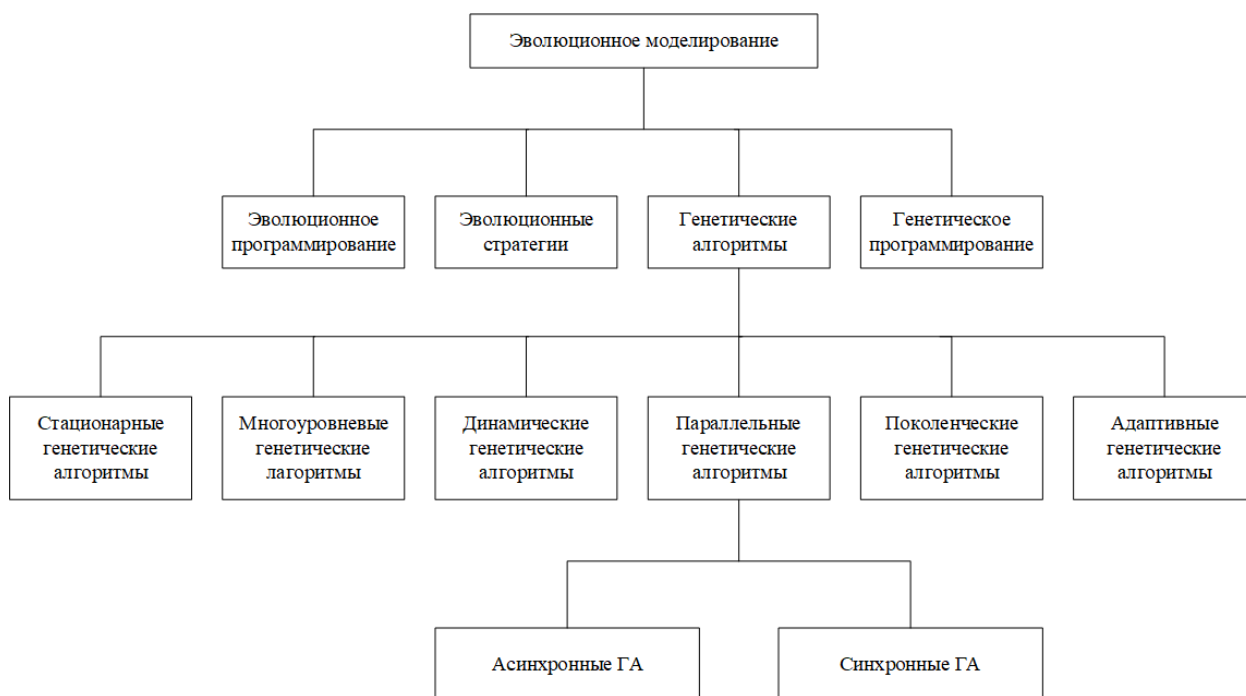


Рисунок 1 – Классификация алгоритмов эволюционного моделирования

ГА не обязательно относится только лишь к одно из возможных классификаций, т. к. не все из них являются взаимоисключающими. Генетический алгоритм может быть отнесен сразу к нескольким категориям, которые представлены на рисунке 1.

### 3.1. Стационарные генетические алгоритмы

Стационарные генетические алгоритмы подразумевают использовать подход, в котором популяция обновляется частями, а не вся сразу, Идея заключается в итеративном создании одного или двух потомков и добавления их непосредственно в популяцию, с предварительным исключением некоторых существующих агентов, чтобы освободить необходимое пространство.

Данный генетический алгоритм, по сравнению с поколенческим, использует примерно в два раза меньше памяти, т.к. всегда рассматривается только одна популяция. Также в этом алгоритме агенты-родители остаются в популяции на более длительное время, т. к. длительность их жизненного цикла обусловлена не «жесткими параметрами моделирования», а параметрами

самого агента, которые формируются под воздействием генетического алгоритма случайным образом.

Такой тип алгоритмов более характерен для реального мира, поколения не имеют между собой четких временных границ, а сменяемость поколений происходит не резко, «между раундами», а плавно, в процессе конкуренции.

### **3.2. Динамические генетические алгоритмы**

Противоположен по своим идеям стационарному генетическому алгоритму. Здесь подразумевается использования итеративного подхода, где после каждой итерации популяция проходит полное обновление, в рамках которого удаляются и добавляются агенты.

### **3.3. Поколенческие генетические алгоритмы**

Поколенческие генетические алгоритмы подразумевают итеративное следованию определенной последовательности действий:

- 1) Создание первоначальной популяции.
- 2) Вычисление функций приспособленности для агентов популяции (оценивание)
- 3) Выборка агентов из текущей популяции (селекция)
- 4) Мутации агентов
- 5) Вычисление функций приспособленности для всех агентов
- 6) Формирование нового поколения
- 7) Оценить результаты и, если они не удовлетворительны, то повторить начиная с 3 пункта.

### **3.4. Адаптивные генетические алгоритмы**

Основная идея адаптивного генетического алгоритма заключается в том, что такой алгоритм должен уметь изменять свои параметры в процессе работы. В качестве параметров здесь выступает размер популяции, вероятность мутации, количество удаляемых и добавляемых агентов, а также множество других параметров, которые зависят от специфики самой задачи. Целью

создания адаптивного алгоритма является ускорение поиска решения или уменьшения затрат ресурсов при его поиске (например, память).

### **3.5. Многоуровневые генетические алгоритмы**

Такие алгоритмы представляют из себя многоуровневую систему, где каждый уровень генетического алгоритма оптимизирует работу вышестоящего генетического алгоритма и так по цепочке. Говоря более формально, многоуровневый генетический алгоритм – это система, в которой генетические алгоритмы нижнего уровня (оптимизируемые генетические алгоритмы) оптимизируют генетические алгоритмы верхнего уровня.

Этот принцип часто используется в нейронных сетях, где вершины нейронной сети разбиты по слоям, представляющие собой уровни генетического алгоритма. При этом возникает очень сложная задача оптимизации, где необходимо вычислить оптимальное количество слоев для наиболее быстрого поиска решения задачи, т.к. никакой универсальной формулы, дающая ответ на этот вопрос, не существует.

### **3.6. Параллельные генетические алгоритмы**

Такие алгоритмы основаны на разбиении популяции на несколько отдельных подпопуляций, каждая из которых будет, независимо от других подпопуляций, обрабатываться генетическим алгоритмом. Одновременно с этим, разнообразные миграции агентов порождают обмен генетическим кодом среди популяций, которые, как правило, улучшают точность и эффективность алгоритма.

Выделяют три типа параллельных генетическим алгоритмов:

- Глобальные однопопуляционные ПГА (master-slave)
- Однопопуляционные ПГА
- Многопопуляционные ПГА

Модель «хозяин-раб» характеризуется тем, что в алгоритмах такого типа селекция принимает во внимание целую популяцию, в отличие от двух других моделей.

В алгоритмах второго класса существует главная популяция, но оценка целевой функции распределена среди нескольких процессоров. Хозяин хранит популяцию, выполняет операции ГА и распределяет агентов между подчиненными. Они же лишь оценивают агентов. Однопопуляционные ГА пригодны для массовых параллельных компьютеров и состоят из одной популяции. Данный класс ПГА может быть эффективно реализован на параллельных компьютерах.

Третий класс - многопопуляционные ГА более сложная модель, так как она состоит из нескольких подпопуляций, которые периодически, по установленным правилам, обмениваются агентами. Такой обмен агентами называется миграцией и управляется несколькими параметрами. Такие ГА очень популярны, но достаточно сложны как для понимания, так и для реализации, потому что последствия от эффекта миграции, на данный момент, не полностью исследованы. В то же время многопопуляционные ГА имеют сходство с «островной моделью» в популяционной генетике, которая рассматривает относительно изолированные общины; поэтому параллельные ГА в некоторых случаях называют «островными» параллельными ГА.

#### **4. Анализ предметной области**

Предметная область информационной системы, реализующая генетические алгоритмы, включает в себя ряд объектов, данных и функций, которые её описывают. Они представлены на рис. 1.

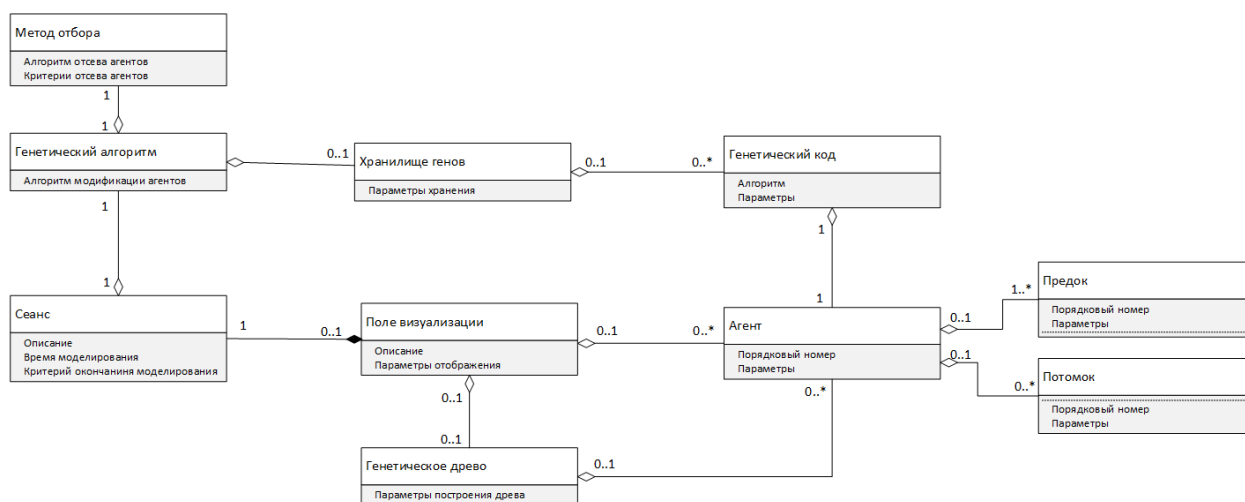


Рисунок 2 – Концептуальная схема предметной области

## 4.1. Основные объекты предметной области

Объекты представляют собой действующие лица, между которыми идет обмен информацией и которые осуществляют некоторые функции над друг-другом.

### 4.1.1. Агент

Агент представляет из себя элементарную рабочую единицу ИС. Перед агентом ставятся задачи, для которой в системе идет поиск решения, как можно более приближенного к оптимальному. Существует множество агентов, параллельно решающих поставленную перед ними задачу. В начале процесса нахождения оптимального решения задачи все агенты находятся в схожих условиях и имеют идентичную конфигурацию. В течение некоторого отрезка времени в модели порождаются новые агенты с несколько отличной конфигурацией согласно генетическому алгоритму. Периодически с помощью метода отбора часть агентов удаляется из процесса поиска оптимального решения.

### 4.1.2. Хранилище генов

Хранилище генов представляет из себя банк с данными. В этих данные содержится информацией о конфигурациях агентов. Хранилище генов содержит в себе информацию как о тех агентах, которые участвуют в текущем сеансе. Данные о конфигурации агентов из хранилища генов используются для

формирования конфигурации новых агентов в процессе работы генетического алгоритма.

## **4.2. Основные информационные структуры предметной области**

Информационные структуры представляют из себя совокупность данных, которыми обмениваются объекты предметной области. Также данные используются функциями (алгоритмами) для получения результата, который может представлять из себя набор еще каких-то данных.

### **4.2.1. Генетический код**

Включает в себя набор параметров и инструкций, который использует агент в процессе своего жизненного цикла. Генетический код агента также можно определить, как его конфигурацию. Генетический код изменяется случайным образом в процессе работы генетического алгоритма. Генетический код наиболее агента, который наиболее эффективно решает поставленную перед ним задачу, является тем самым оптимальным решением, поиск которого производится в системе.

### **4.2.2. Генетическое древо**

Представляет из себя совокупность информации о родителях и потомках каждого агента, когда-либо появившегося в процессе сеанса поиска решения. Генетическое древо показывает какие ветви развития агентов имели какую численность в зависимости, и какая последовательность изменений привела к текущим генетическим кодам агентов

### **4.2.3. Поле визуализации**

Поле визуализации показывает текущее состояние сеанса работы генетического алгоритма.

Оно может в себя включать различные графические элементы:

- 1) Графики
- 2) Диаграммы
- 3) Графы состояний и переходов
- 4) Поля пространств, где размещаются и перемещаются агенты



## 5) Визуализации наполнения банков данных

Информация в поле призвана давать возможность объективно оценить текущие состояние сеанса и определить, был ли достигнут требуемый результат работы генетического алгоритма.

### 4.3. Основные процедуры предметной области

Процедуры (алгоритмы), реализуемые в системе, использующие генетические алгоритмы, необходимы для обработки имеющихся данных и формирования новых.

#### 4.3.1. Генетический алгоритм

Наиболее важная часть ИС, определяющая всю логику ее работы и множество задач, которые эта система может решать.

Генетический алгоритм определяет каким образом будет происходить наследование генетического кода от агентов одного поколения к агентам последующего, а также какие случайные изменения будут происходить в генетическом коде новых агентов и с какой вероятностью.

#### 4.3.2. Метод отбора

Это алгоритм, который осуществляет отсев агентов в процессе моделирования, решения поставленной задачи которых оказалось наименее эффективно согласно критериям оптимальности решения этой задачи.

#### 4.3.3. Сеанс

Сеанс представляет из себя весь временной отрезок, в рамках которого агенты решают представленную перед ними задачу. Сеанс в зависимости от условий и специфики задач, стоящих перед ними, может проходить в различных формах:

- 1) Агенты решают поставленную задачу независимо, никак не контактируя и не взаимодействуя друг с другом
- 2) Агенты решают поставленную задачу в одной среде, весь сеанс поделен на периоды времени, представляющие из себя циклы, после каждого

из которых происходит удаление и добавление агентов в зависимости от эффективности решения каждого.

3) Агенты решают задачу одновременно в одной среде соревнуясь друг с другом. При таком виде сеанса отбор осуществляется естественным образом в ходе конкуренции между агентами. При этом наиболее продвинутые агенты порождают других агентов, согласно генетическому алгоритму, поддерживая тем самым их общее количество. Этот тип сеанса больше всего напоминает естественные процессы, проходящие в живой природе.

## **5. Необходимые информационные технологии**

Для успешного построения системы, которая будет обрабатывать данные с использованием генетического алгоритма, необходимо использование различных информационных технологий

### **5.1. Централизованная обработка информации**

Централизованная обработка подразумевает, что система работы системы будет осуществляться на удаленном высокопроизводительном сервере.

Такой подход к обработке эффективен, если решаемые задачи требуют затрат больших вычислительных ресурсов или специализированного программного обеспечения. В случае с генетическими алгоритмами использование такого подхода будет преследовать цель нахождения оптимального решения в максимально сжатые сроки.

Ожидаемый эффект использования технологии – быстрое моделирование работы генетического алгоритма.

### **5.2. Децентрализованная обработка информации**

Децентрализованная обработка информации подразумевает использование распределенной системы, где в роли вычислительных мощностей выступают компьютеры абонентов.

Этот подход подходит, если скорость работы генетического алгоритма не столь важна, сколько возможность проводить параллельно множество таких процессов моделирования. При этом это дает каждому абоненту более широкие возможности по настройке параметров моделирования.

Ожидаемый эффект использования технологии – возможность вести множество моделирований работы генетического алгоритма, не ограничивая себя вычислительными мощностями центрального сервера системы.

### **5.3. Информационная технология поддержки принятия решений**

Эта технология организует взаимодействие пользователя и вычислительной системы, где в результате обработки входных данных от пользователя, выдается рекомендация.

Технология поддержки принятия решения ориентирована на решение плохо структурированных задач. Использование генетического алгоритма позволяет решать такие задачи, где не найдено аналитическое решение, но при этом можно составить математическую модель, которая описывала бы предметную область и критерии оптимальности решения.

Ожидаемый эффект использования технологии – возможность принимать от пользователя набор данных, описывающий задачу, и находить максимально приближенное к оптимальному решение этой задачи с использованием генетического алгоритма.

Особенность применения – необходимо обеспечивать дополнительный анализ информации от пользователя для составления математической модели решаемой задачи.

## **6. Понятие цифрового автомата как математической абстракции**

Понятие абстрактного автомата [АА], позволяет рассматривать дискретные объекты с точки зрения алгоритмов их функционирования, то есть реализуемых последовательностей действий по преобразованию дискретной информации.

Абстрактным автоматом называют модель, которая описывается кортежем, состоящим из 5 элементов:

$$A = (X, Y, S, f_y, f_s)$$

Здесь первые три компонента – это непустые множества:

$X$  – множество входных сигналов,

$Y$  – множество выходных сигналов,

$S$  – множество состояний.

Остальные два компонента кортежа являются характеристическими функциями:

$f_y$  – функция выходов,

$f_s$  – функция переходов АА из одного состояния в другое

Если множества  $X$ ,  $Y$ ,  $S$  – конечные, то такой АА называется конечным автоматом [КА]. В том случае, если бы хотя бы одно из множеств автомата было бесконечным, то и сам АА являлся бы бесконечным.

В данной исследовательской работе основным объектом для анализа являются именно КА, который, в отличие от БА, программно-реализуемым и может быть полностью представлен с помощью таблицы или графа переходов, что дает возможность их практического применения.

### **6.1. Классификация абстрактного автомата**

Для классификации автомата рассматривается множество признаков, например, определенность функции переходов и функции выходов, однозначность заданных функций, устойчивость состояний. Все три признака перечислены и определены в таблице 1.

Классификация абстрактных автоматов может быть представлена в виде схемы на рисунке 3.

Таблица 1

Признак	Определение
Определенность характеристических функций	В автоматах полностью определенных областью определения функций $f_s$ и $f_y$ является множество всех пар $(s_i, x_k) \in S \times X$ , где $s_i \in S$ , $x_k \in X$ . В автоматах частично определенных либо обе характеристические функции, либо одна из них имеют областью определения строгое подмножество декартова произведения $S \times X$ . Таким образом, характеристические функции подобных автоматов определены не для всех пар $(s_i, x_k)$ .
Однозначность функции переходов	В детерминированных автоматах выполняется условие однозначности переходов: если АА находится в некотором состоянии $s_i \in S$ , то под воздействием произвольного входного сигнала $x_k \in X$ автомат может перейти в одно и только одно состояние $s_j \in S$ , причем ситуация $s_i = s_j$ вовсе не исключается. В автоматах вероятностных при воздействии одного и того же входного сигнала возможны переходы из состояния $s_i$ в различные состояния из множества $S$ с заданной вероятностью.
Устойчивость состояний	В устойчивых автоматах выполняется условие устойчивости: если автомат под воздействием входного сигнала $x_k \in X$ оказался в состоянии $s_i \in S$ , то выход из него и переход в иное состояние возможен только при поступлении на вход автомата другого сигнала $x_z \in X$ , $x_z \neq x_k$ . Если условие устойчивости не выполняется хотя бы для одного состояния $s_j \in S$ , то такой автомат называют неустойчивым.

Для практического применения чаще всего используются те автоматы, которые по перечисленным выше признакам являются полностью определенными, детерминированными и устойчивыми конечными автоматами.



Рисунок 3 – Классификация абстрактных автоматов

Также можно классифицировать автоматы по виду ее характеристической функции. Если аргументы характеристической функции являются только текущее значение входного сигнала и текущего состояния, то такой автомат является автоматом 1 рода или автоматом Мили. Если выходные сигналы автомата зависят исключительно от текущего состояния автомата, то такой автомат принято считать автоматом 2 рода или автоматом Мура.

Пример графов переходов для автомата Мура и автомата Мили представлены на рисунках 4 и 5 соответственно/

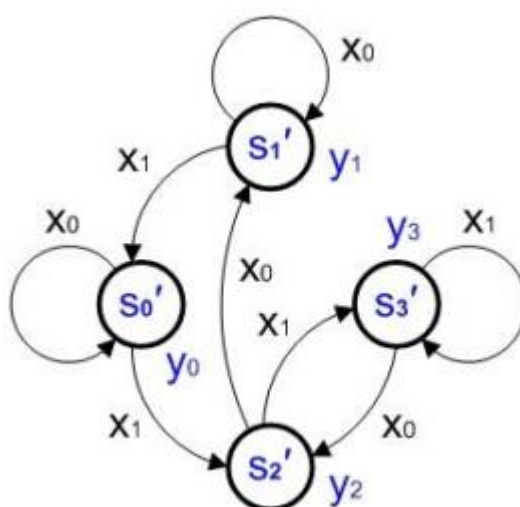


Рисунок 4 – граф переходов автомата Мура

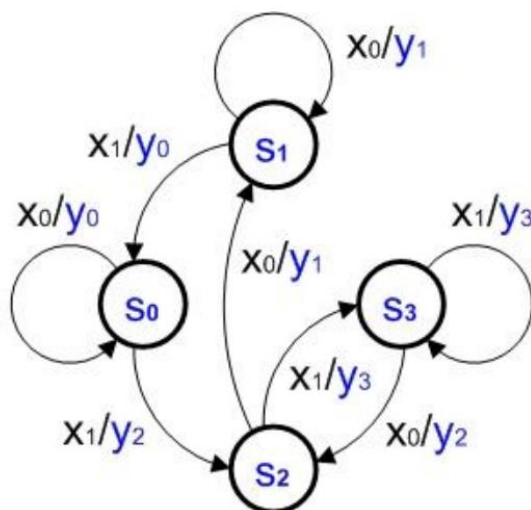


Рисунок 5 – граф переходов автомата Мили

Следует при этом отметить, что автомат Мили запаздывает на один дискретный момент времени по входному сигналу по отношению к автомату Мура

Если в автомате присутствуют как свойства автомата Мура, так и автомата Мили, то такой автомат можно считать смешанным.

## 6.2. Базовая модель конечного автомата

Конечный автомат, в описание которого входят таким образом определенные множества, называют  $(n, p, q)$ -автоматом, а самим множествам усваивают наименование векторов, например: вектор входных сигналов, вектор состояний.

Все автоматы, и в том числе конечные, функционируют в дискретном исчислении времени. Моменты времени образуют ряд целых неотрицательных чисел:  $t = 0, 1, 2, 3, \dots$  В каждый дискретный момент времени КА находится в одном и только одном состоянии  $S_i$ , воспринимает одно значение вектора  $X$  и выдает на выходе одно значение вектора  $Y$ .

Принято считать, что в момент времени  $t = 0$  автомат находится в начальном состоянии  $S_0$ , которое можно включить в кортеж отдельным, шестым компонентом:  $A = (X, Y, S, f_y, f_s, S_0)$ .

Автомат с выделенным начальным состоянием называют инициальным.

Общую схему автомата можно представить в виде «черного ящика», осуществляющего преобразование вектора входных сигналов в вектор выходных.

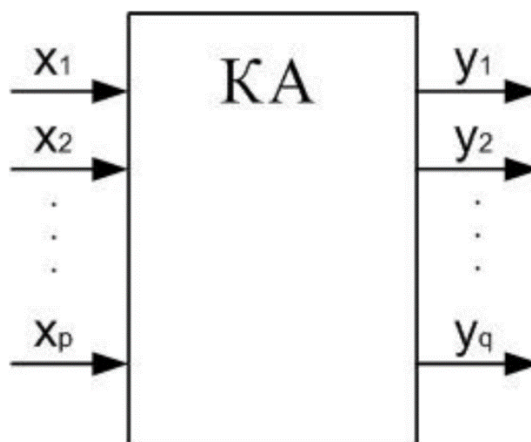


Рисунок 6 – Общая схема конечного автомата

Исходя из этого, можно записать следующее уравнение:  $Y = f_y(X, t)$ .

Фактор времени в приведенном уравнении учитывается введением вектора состояний  $S$ , как своего рода «памяти о прошлом». Действительно, на один и тот же набор входных сигналов (значений компонентов вектора  $X$ ) автомат будет выдавать разные выходные сигналы (значения компонентов вектора  $Y$ ) в зависимости от состояния, в котором он находится в данный момент времени. Текущее состояние, в свою очередь, определяется алгоритмом функционирования автомата.

## 7. Понятие клеточного автомата

Клеточный автомат – это дискретная модель, которая представляет из себя сетку произвольной размерности. Каждая клетка в каждый момент времени может принимать одно из конечного множества состояний, при этом существует правило, по которому осуществляется переход клеток из одного состояния в другое.

Клеточные автоматы могут быть охарактеризованы критериями, представленными в таблице .



Таблица 2

Критерий	Возможные варианты
Размерность решетки	Одномерная, двумерная, трехмерная и т.д.
Количество возможных состояний	Бинарные, троичные и т.д.
Определение окрестности клетки	Окрестность Фон-Неймана, окрестность Мура и др.
Синхронизация	Синхронные и асинхронные клеточные автоматы
Тип поведения	4 класса

Клеточные автоматы можно разделить на 4 класса по типу поведения:

1) Все клетки быстро принимают одинаковое состояние, после чего автомат стабилизируется.

2) Состояние всех клеток быстро стабилизируется, либо возникают периодические колебания состояний клеток

3) Автомат порождает хаотические, непериодические структуры. Небольшие изменения исходного состояния влекут за собой значительные изменения в будущем.

4) Автомат порождает сложные, взаимодействующие между собой структуры, способные выживать длительное время. Однако при этом автомату не удастся достичь стабильного состояния.

Самой простой, так называемый простейший клеточный автомат, - это одномерный бинарный клеточный автомат, где состояния клетки в каждый момент времени зависят только от ее собственного состояния и состояний смежных с ней клеток в предыдущий момент времени.

Простейших клеточных автоматов существует всего 256, и поведение некоторых из них дублирует другие. Но, несмотря на это, широко известный в узких кругах Стивен Вольфрам посвятил годы жизни их изучению.

Вариантов простейших автоматов всего 256. Каждый из вариантов таким автоматов принято называть по порядковому номеру, или же «Правило N». Возьмем для примера наиболее интересное из них, правило 110.

Двоичный код десятичного числа 110 представляет собой последовательность бит 01101110. Данная последовательность бит формирует функцию переходов клетки.

Таблица 3

111	110	101	100	011	010	001	000
0	1	1	0	1	1	1	0

В зависимости от состояний соседа слева, самой клетки, соседа справа (первая строка таблицы) на следующем шаге клетка примет одно из состояний, указанных во второй строке таблицы 3.

Графическая иллюстрация представлена на рисунке .



Рисунок 7 – графическое представление правила 110

Клеточные автоматы, в отличие от классических конечным автоматов, являются Тьюринг-полными. Это делает возможным, посредством клеточного автомата реализовывать любую вычислимую функцию, т.е. реализовывать любой возможный алгоритм.

Стоит заметить, что каждая клетка клеточного автомата представляет простой конечный автомат. В простом случае, когда имеется двумерное поле и каждая клетка способна принимать всего два состояния, эта самая клетка реализует автомат, который способен принимать два возможных состояния под воздействием входных сигналов. В данном случае входными сигналами будут являться окрестность клетки, т.е. текущие состояния клеток, входящих

в эту окрестность. При этом текущее состояние клетки-автомата будет являться входным сигналом для других клеток.

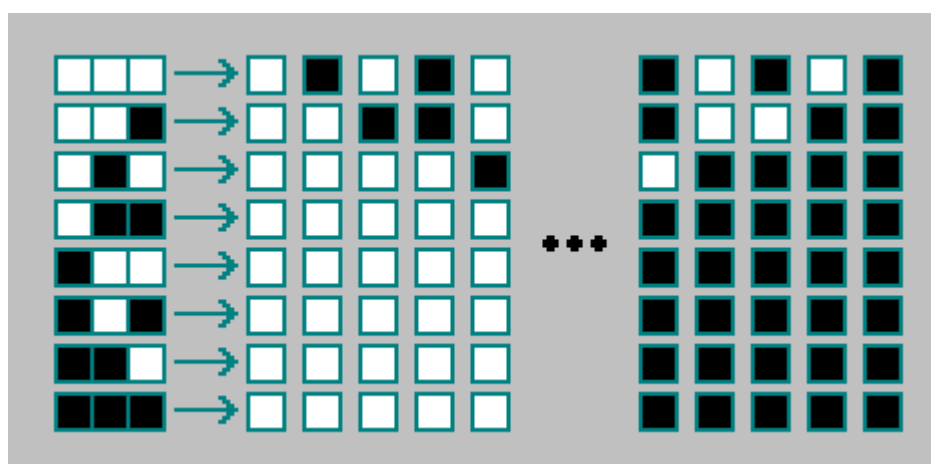
## 8. Эволюционирующий клеточный автомат

Клеточный автомат весьма мощное средство моделирования. Он позволяет строить модели различных динамических систем. Даже простейшие одномерные клеточные автоматы являются Тьюринг-полными и способны генерировать сложные неперiodические структуры.

Однако все клетки такого автомата равны между собой и работает каждая из них по заранее определенному правилу. Соединение идеи клеточного автомата и генетического алгоритма позволит избежать статичности правила и позволит его видоизменять при каждой итерации алгоритма.

В этом случае правило автомата фактически превращается в генетический код, а сам клеточный автомат становится агентом.

В случае простейшего генетического автомата, где правило формируется по состояниям двух смежных клеток и состоянию самой клетки, существует 8 возможных комбинаций, из которых можно породить 256 правил (рис ). Как раз переход в одно из двух возможных состояний для каждой из 8 комбинаций будет определять один из соответствующих им 8 генов.



Для решения задач моделирования чаще приходится использовать моделирование в двумерном или трехмерном пространстве. Практических

задач для одномерного пространства значительно меньше чем, например, задачи на плоскости или связанных с объемным пространством. Поэтому стоит рассмотреть работу генетического алгоритма с использованием двумерного клеточного автомата для большей наглядности и практичности.

У двумерного клеточного автомата для каждой клетки, если использовать окрестность Мура, существует 8 соседних клеток. Существует порядка  $2^9 = 512$  возможных комбинаций состояний клетки и её соседей, из чего следует, что поведение данного клеточного автомата определяет комбинация из 512 генов. Комбинация этих генов дает  $2^{512}$  возможных генетических конфигураций клеточного автомата (рис 8), что является огромным множеством всевозможных комбинаций.

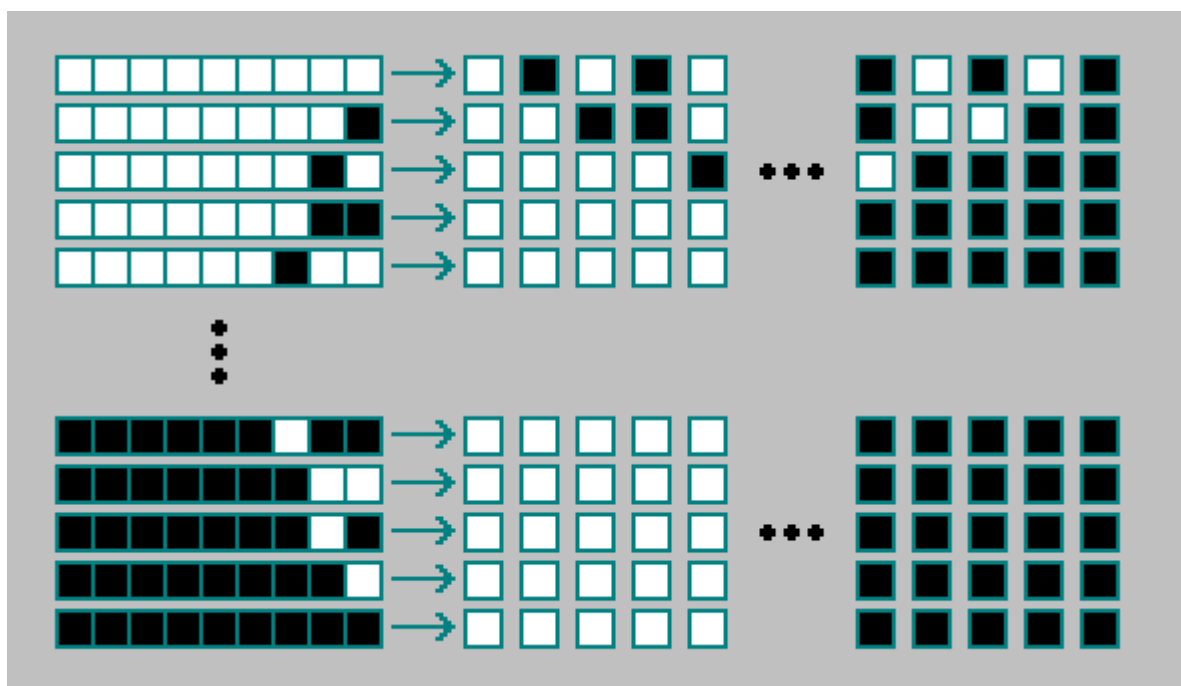


Рисунок 8 – Возможные комбинации состояний двумерного клеточного автомата

## Заключение

Существует вопрос о том, насколько целесообразно применение генетических алгоритмов для решения различных задач. С одной стороны, в математике существует достаточно большой класс абсолютно надежных (в смысле гарантии получения точного решения) методов решения различных задач. С другой стороны, речь идет о действительно сложных практических задачах, в которых эти надежные методы часто неприменимы. Нередко эти задачи выглядят настолько необозримыми, что не предпринимается даже попыток их осмысленного решения.

Генетические алгоритмы — реализация одной из наиболее популярных идей такого рода. Таким образом, задав условия жизни в некотором виртуальном мире и заселив его представителями с определенными свойствами, после процессов скрещивания, мутации и естественного отбора, аналоги которых происходят и в реальном мире, мы стабильно получаем особь, свойства которой отвечают ранее заданным требованиям. Этот факт говорит о том, что понимание проверенных веками законов природы позволяет использовать их при решении, казалось бы, и далеких от нее задач, частным случаем которых являются задачи оптимизации.

Генетические алгоритмы достаточно эффективный способ решения непростых оптимизационных задач, поскольку сочетает в себе комбинацию переборного и градиентного методов. Механизмы кроссинговера (скрещивания) и мутации реализуют переборную часть, а отбор лучших решений — градиентный спуск.

Выбирая приемлемое время расчета, получаем лучшие решения, которые можно получить за это время.

## Список литературы

1. Генетические алгоритмы, Гладков Л.А., Курейчик В.В., Курейчик В.М., М.: ФИЗМАТЛИТ, 2006.
2. Гуренко В.В. Введение в теорию автоматов. Электронное учебное издание. – М.: МГТУ имени Н.Э. Баумана, 2013. – 62 с
3. Скобцов Ю. А. Основы эволюционных вычислений. — Донецк: ДонНТУ, 2008. — 326 с
4. Исаев С.А. Генетические алгоритмы - эволюционные методы поиска – Интернет: <<http://rv.ryazan.ru/~bug/library/ai/isaev/2/part1.html>>
5. Редько В.Г. Курс лекций "Эволюционная кибернетика" – Интернет: <<http://inet.keldysh.ru/BioCyber/Lectures.html>>
6. Кантор И.А.(перевод) Введение в ГА и Генетическое Программирование – Интернет: <<http://www.algolist.manual.ru>>
7. Атлас простейших клеточных автоматов Стивена Вольфрама – Интернет: <<http://atlas.wolfram.com>>
8. Голубин А.В. Определение параметров генетического алгоритма для оптимизации многопараметрических функций// Прогрессивные технологии, конструкции и системы в приборо- и машиностроении. / Сб. статей. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. - С. 65-67.