

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
(МИНОБРНАУКИ)

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»  
(СПБГУ ИТМО)

УДК 004.4'242  
№ госрегистрации 0120.0 710295  
Инв. №

УТВЕРЖДАЮ  
Ректор СПбГУ ИТМО,  
докт. техн. наук, профессор  
В. Н. Васильев

«\_\_\_» \_\_\_\_\_ 2008 г.

ТЕХНОЛОГИЯ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ  
ДЛЯ ГЕНЕРАЦИИ АВТОМАТОВ УПРАВЛЕНИЯ  
СИСТЕМАМИ СО СЛОЖНЫМ ПОВЕДЕНИЕМ

ПРОМЕЖУТОЧНЫЙ ОТЧЕТ ПО III ЭТАПУ  
«ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ  
ПОСТАВЛЕННЫХ ПЕРЕД НИР ЗАДАЧ»

ЛИСТОВ 104

Декан факультета «Информационные  
технологии и программирование»  
докт. техн. наук, профессор  
\_\_\_\_\_ В. Г. Парфенов

Руководитель темы  
заведующий кафедрой «Технологии программирования»,  
докт. техн. наук, профессор  
\_\_\_\_\_ А. А. Шалыто

Ответственный исполнитель  
магистрант кафедры «Компьютерные технологии»,  
\_\_\_\_\_ Ф. Н. Царев

САНКТ-ПЕТЕРБУРГ  
2008

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

## СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель темы докт. техн. наук, профессор	А. А. Шалыто	Отчет в целом
Ответственный исполнитель магистрант	Ф. Н. Царев	Отчет в целом
Нормоконтролер	Г. Р. Туктарова	Отчет в целом
Ведущий научный сотрудник, докт. техн. наук, профессор	В. В. Антипов	Разделы 1 и 2.
Ведущий научный сотрудник, канд. техн. наук	В. В. Киселев	Раздел 1.
Ведущий научный сотрудник, канд. техн. наук	Р. Н. Котляр	Раздел 3.
Ведущий научный сотрудник, канд. техн. наук	Ю. П. Московцев	Раздел 4.
Ведущий научный сотрудник, канд. техн. наук, доцент	В. А. Третьяков	Раздел 2.
Ведущий научный сотрудник, канд. техн. наук	Г. М. Файкин	Раздел 4.
Доцент, канд. техн. наук	Г. А. Корнеев	Раздел 3.
Канд. техн. наук	Е. М. Кузнецова	Раздел 2.
Канд. техн. наук	А. С. Пресняк	Раздел 1.
Доцент, канд. техн. наук	Г. А. Корнеев	Раздел 4.
Доцент, канд. техн. наук	Э. А. Опалева	Раздел 2.
Руководитель временного трудового коллектива (ВТК), ассистент кафедры КТ	А. П. Мельничук	Раздел 2.
Член ВТК, профессор кафедры ИС	Е. Ю. Михайлова	Раздел 2.2.
Член ВТК, доцент кафедры КТ	В. Д. Наумчик	Раздел 2.1.
Член ВТК, доцент кафедры КТ	М. Ю. Осипов	Раздел 2.2.
Член ВТК, доцент кафедры КТ	А. Н. Воробьев	Раздел 2.1.
Член ВТК, доцент кафедры КТ	Ю. А. Щупак	Раздел 2.3.
Член ВТК, доцент кафедры КТ	С. В. Чириков	Раздел 2.4.
Член ВТК, доцент кафедры КТ	А. С. Сегаль	Раздел 2.5.
Член ВТК, доцент кафедры КТ	Д. Г. Шопырин	Раздел 2.5.
Член ВТК, доцент кафедры ИС	Д. А. Зубок	Раздел 2.1.
Член ВТК, ассистент кафедры ИС	В. В. Повышев	Раздел 2.4.
Член ВТК, ассистент кафедры ИС	В. В. Ильин	Раздел 2.5.
Член ВТК, ассистент кафедры ИС	М. Г. Холин	Раздел 2.5.
Аспирант	П. Г. Лобанов	Разделы 2 и 4.
Магистрант	Н. И. Поликарпова	Раздел 3.2.
Магистрант	В. Н. Точилин	Раздел 3.2.
Магистрант	Ю. Д. Бедный	Раздел 3.3.
Магистрант	В. Р. Данилов	Раздел 3.3.
Студент	Е. А. Мандриков	Разделы 3.2. и 3.3.
Студент	В. А. Кулев	Разделы 3.2. и 3.3.
Студент	А. А. Давыдов	Разделы 3.2, 3.3. и 4.
Студент	Д. О. Соколов	Разделы 3.2, 3.3. и 4.
Студент	В. О. Клебан	Разделы 1 и 3.2.
Студент	С. О. Попов	Раздел 3.3.

## РЕФЕРАТ

Отчет 104 с., 4 гл., 78 рис., 27 табл., 6 приложений, 30 источников.

### УПРАВЛЯЮЩАЯ ПРОГРАММА, КОНЕЧНЫЙ АВТОМАТ, ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ, АВТОМАТНОЕ ПРОГРАММИРОВАНИЕ, ПОВЕДЕНИЕ.

Объектом исследования являются методы генерации автоматов управления системами со сложным поведением.

Цель этапа — апробация и экспериментальное исследование методов генерации автоматов управления системами со сложным поведением, разработанных на предыдущих этапах работ по контракту на примере системы управления моделью беспилотного летательного аппарата.

В результате выполнения третьего этапа работ была исследована модель беспилотного летательного аппарата, позволяющая оценить разработанные методы. Были построены системы управления моделью беспилотного летательного аппарата, построенные на основе методов, предложенных на предыдущих этапах работы. В качестве прототипа для анализа без применения методов автоматизации была разработана система управления моделью беспилотного летательного аппарата.

Результаты, полученные на третьем этапе, позволяют перейти к детальной оценке разработанных методов и работы в целом.

До окончания работы в целом оценить экономическую эффективность не представляется возможным.

Прогнозные предположения о развитии объекта исследования: предполагается внедрение разработанных методов в практику построения управляющих программ систем со сложным поведением.

В первой главе отчета описана тестовая модель беспилотного летательного аппарата и разработаны критерии оценки качества управления.

Во второй главе без применения методов автоматизации реализована система управления беспилотным летательным аппаратом, применяемая для оценки систем управления, построенных на основе разработанных методов.

Третья глава содержит описание систем управления беспилотным летательным аппаратом, построенные на основе методов, разработанных на втором этапе работы: метода сокращенных таблиц переходов, метод представления автоматов деревьями решений и метода оптимизации генетических алгоритмов для построения автоматов.

В четвертой главе приведен сравнительный анализ полученных систем управления и результаты экспериментальных исследований.

Отчет содержит шесть приложений: А. Отчет о патентных исследованиях, Б. Спецификация В. Текст программы, Г. Описание программы, Д. Программа и методика экспериментов, Е. Протоколы экспериментов.

## СОДЕРЖАНИЕ

СПИСОК ИСПОЛНИТЕЛЕЙ.....	2
РЕФЕРАТ .....	3
СОДЕРЖАНИЕ .....	4
ВВЕДЕНИЕ.....	7
1. ОПИСАНИЕ МОДЕЛИ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА.....	9
1.1. ПРАВИЛА СОРЕВНОВАНИЙ .....	9
1.2. ДИНАМИКА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА.....	11
1.3. АЭРОДИНАМИЧЕСКОЕ ВЗАИМОДЕЙСТВИЕ МЕЖДУ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ .....	12
1.4. СТОЛКНОВЕНИЕ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ .....	13
1.5. МОДЕЛИРОВАНИЕ ГОНКИ.....	14
1.6. КРИТЕРИИ ОЦЕНКИ СИСТЕМ УПРАВЛЕНИЯ.....	15
1.7. ВЫВОДЫ .....	16
2. СИСТЕМА УПРАВЛЕНИЯ, ПОСТРОЕННАЯ БЕЗ ПРИМЕНЕНИЯ МЕТОДОВ АВТОМАТИЗАЦИИ.....	17
2.1. КРАТКОЕ ОПИСАНИЕ СТРАТЕГИИ .....	17
2.2. ДИАГРАММА СВЯЗЕЙ.....	17
2.3. ПОСТАВЩИКИ СОБЫТИЙ .....	18
2.3.1. Поставщик событий Environment .....	18
2.3.2. Поставщик событий Radar .....	19
2.4. АВТОМАТЫ.....	19
2.4.1. Автомат AFP – Состояние агента .....	19
2.4.2. Автомат AL – Режим полета.....	20
2.4.3. Автомат A1 - Уклонение от границ коридора и агентов справа и слева .....	21
2.4.4. Автомат AR – Радар.....	21
2.4.5. Автомат A3 – Уклонение от летательных аппаратов спереди и сзади .....	22
2.5. ОБЪЕКТ УПРАВЛЕНИЯ FLYINGPLATE .....	22
2.6. СИСТЕМА УПРАВЛЕНИЯ С «АГРЕССИВНОЙ» СТРАТЕГИЕЙ .....	23
2.7. СИСТЕМА УПРАВЛЕНИЯ С «ПРОСТОЙ» СТРАТЕГИЕЙ.....	23
2.8. ВЫВОДЫ.....	24
3. АВТОМАТИЗИРОВАННОЕ ПОСТРОЕНИЕ СИСТЕМ УПРАВЛЕНИЯ .....	25
3.1. СОВМЕСТНОЕ ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ.....	25
3.1.1. Некоторые проблемы, возникающие при использовании генетического программирования для построения конечных автоматов.....	25
3.1.2. Искусственные нейронные сети.....	25
3.1.2.1. Элементы нейронных сетей .....	26
3.1.2.2. Структура нейронных сетей .....	27
3.1.2.3. Применение нейронных сетей .....	27
3.1.3. Структура системы управления беспилотным летательным аппаратом .....	27
3.1.4. Алгоритм генетического программирования для построения системы управления беспилотным летательным аппаратом .....	28
3.1.4.1. Структура особи в используемом алгоритме .....	29
3.1.4.2. Создание начального поколения .....	30
3.1.4.3. Операция мутации .....	30
3.1.4.4. Операция скрещивания .....	31
3.1.4.5. Формирование следующего поколения .....	34

3.1.4.6. Функция приспособленности.....	35
3.1.4.7. Результаты применения метода совместного применения нейронных сетей, генетического программирования и конечных автоматов.....	35
3.2. СОВМЕСТНОЕ ПРИМЕНЕНИЕ МЕТОДА СОКРАЩЕННЫХ ТАБЛИЦ ПЕРЕХОДОВ И АЛГОРИТМА ВОССТАНОВЛЕНИЯ СВЯЗЕЙ МЕЖДУ СОСТОЯНИЯМИ .....	38
3.2.1. Структура системы управления беспилотным летательным аппаратом .....	38
3.2.2. Представление особи в генетическом алгоритме.....	39
3.2.3. Восстановление связей между состояниями .....	41
3.2.4. Оператор мутации .....	42
3.2.5. Оператор скрещивания .....	43
3.2.6. Особенности применения сокращенных таблиц.....	45
3.2.7. Генетический алгоритм.....	46
3.2.7.1. Создание начального поколения .....	46
3.2.7.2. Формирование следующего поколения .....	46
3.2.7.3. Миграции .....	46
3.2.7.4. Вычисление функции приспособленности.....	46
3.2.7.5. Большие мутации .....	50
3.2.7.6. Особенности применения островного генетического алгоритма .....	50
3.2.8. Результат применения метода представления автоматов с помощью сокращенных таблиц переходов .....	51
3.3. ПРИМЕНЕНИЕ МЕТОДА ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ДЕРЕВЬЯМИ РЕШЕНИЙ.....	52
3.3.1. Представление особи в генетическом алгоритме.....	52
3.3.2. Генерация случайной особи .....	54
3.3.3. Алгоритм обрезки недостижимых ветвей дерева .....	54
3.3.4. Оператор мутации .....	56
3.3.5. Оператор скрещивания .....	57
3.3.6. Особенности представления при помощи деревьев решений .....	58
3.3.7. Генетический алгоритм.....	58
3.3.7.1. Создание начального поколения .....	58
3.3.7.2. Вычисление функции приспособленности.....	58
3.3.8. Результат применения метода представления автоматов с помощью деревьев решений.....	59
3.4. ВЫВОДЫ.....	64
4. СРАВНЕНИЕ СИСТЕМ УПРАВЛЕНИЯ .....	65
4.1. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ С СИСТЕМОЙ УПРАВЛЕНИЯ, ПОСТРОЕННОЙ БЕЗ ПРИМЕНЕНИЯ МЕТОДОВ АВТОМАТИЗАЦИИ.....	65
4.1.1. Соревнование с командой беспилотных летательных аппаратов, реализующей «агрессивную» стратегию .....	65
4.1.2. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.....	68
4.1.3. Соревнование командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода сокращенных таблиц переходов .....	70
4.1.4. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода представления автоматов деревьями решений.....	72

4.2. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМЫХ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА СОВМЕСТНОГО ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ, АВТОМАТОВ И НЕЙРОННЫХ СЕТЕЙ.....	74
4.2.1. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации .....	74
4.2.2. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, реализующей «агрессивную» стратегию.....	76
4.2.3. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода сокращенных таблиц переходов .....	79
4.2.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.....	81
4.3. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМОЙ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА СОКРАЩЕННЫХ ТАБЛИЦ ПЕРЕХОДОВ .....	83
4.3.1. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, реализующей «агрессивную» стратегию .....	83
4.3.2. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной без применения методов автоматизации .....	86
4.3.3. Соревнование с командой, беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.....	88
4.3.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.....	90
4.4. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМОЙ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ДЕРЕВЬЯМИ РЕШЕНИЙ.....	92
4.4.1. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной без применения методов автоматизации .....	92
4.4.2. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, реализующей «агрессивную» стратегию .....	94
4.4.3. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.....	96
4.4.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода сокращенных таблиц переходов .....	98
4.5. ОБЩАЯ ОЦЕНКА СИСТЕМ УПРАВЛЕНИЯ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ.....	100
4.6. ВЫВОДЫ.....	101
ЗАКЛЮЧЕНИЕ .....	102
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	103

## ВВЕДЕНИЕ

Технология генетического программирования для генерации автоматов управления системами со сложным поведением разрабатывается в рамках проведения научно-исследовательской работы по лоту «Разработки в области языков программирования и моделирования программного обеспечения, технологий и инструментальных средств проектирования программ» шифр «2007-4-1.4-18-01-033» по теме «Технология генетического программирования для генерации автоматов управления системами со сложным поведением», выполняемой в рамках Федеральной целевой научно-технической программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2002 годы» по государственному контракту № 02.514.11.4044 от 18.05.2007, заключенному между Федеральным агентством по науке и инновациям и Государственным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» на основании решения Конкурсной комиссии Роснауки № 14 (протокол от 28.04.2007 г. № 14).

Целью настоящего этапа является апробация и экспериментально исследование методов генерации автоматов управления системами со сложным поведением. Задачами этапа является:

1. Описание тестовой модели беспилотного летательного аппарата и определение критериев оценки качества управления.
2. Апробация методов генерации автоматов управления системами со сложным поведением на примере создания системы управления моделью беспилотного летательного аппарата.
3. Экспериментальное исследование и оценка построенной системы управления.

Достижение цели настоящего этапа работы позволит перейти к оценке разработанных генерации автоматов управления системами со сложным поведением и работы в целом.

Как показал патентный поиск, в настоящее время работы по построению автоматов на основе генетического программирования проводятся в ряде ведущих университетов мира, в том числе в Массачусетском технологическом институте и Университете Южной Калифорнии. Знакомство с результатами этих работ показало, что в них строятся автоматы, которые не могут быть использованы в системах со сложным поведением.

Обычно генетические алгоритмы в рамках эволюционного моделирования используются для настройки нейронных сетей. Однако, если настроенная нейронная сеть функционирует в рассматриваемой среде недостаточно эффективно, то без применения методов автоматизации ее перенастроить невозможно. Поэтому приходится вновь и вновь настраивать ее при помощи генетических алгоритмов. При этом человеку весьма трудно направить процесс в требуемом направлении, а также при необходимости понять полученный результат.

Как показано на предыдущих этапах исследования, при применении генетических алгоритмов для настройки автоматов ситуация принципиально изменяется, так как построив с помощью генетических алгоритмов автоматы, их в дальнейшем обычно удается модифицировать.

Поиск приемлемого по выбранным критериям управляющего автомата перебором практически невозможен из-за огромного размера пространства, в котором осуществляется поиск. Например, в такой простой задаче как задача об «Умном муравье», число возможных автоматов с семью состояниями около  $3,2 \times 10^{18}$ .

Применение генетического программирования позволяет сделать перебор направленным, однако и в этом случае трудоемкость построения автоматов с требуемыми свойствами остается большой.

Указанная проблема была решена за счет учета специфики автоматов, применяемых в системах управления, состоящей в том, что состояния декомпозируют входные воздействия на группы, в каждую из которых обычно входит небольшое число переменных. Это позволяет строить

хромосомы только для подмножества класса всех автоматов, что существенно сокращает пространство возможных решений, и как следствие, время поиска. Апробации и экспериментальному исследованию методов, основанных на этих идеях, рассматривается в настоящем отчете.

Метрологическое обеспечение НИР не предусмотрено техническим заданием.

Дополнительные патентные исследования, проведенные в рамках третьего этапа работы (отчет о патентных исследованиях № 2008.05.30-1 входит в состав отчетной документации по этапу), позволяют утверждать, что в настоящее время отсутствуют патенты и иные охраняемые документы, которые могут препятствовать применению в Российской Федерации результатов научных исследований, проводимых по контракту.

Изложенное позволяет утверждать, что результаты выполнения научно-исследовательской работы будут соответствовать мировому уровню разработок в рассматриваемой области.



## 1. ОПИСАНИЕ МОДЕЛИ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

Проводится соревнование [21, 27, 26] между двумя командами беспилотных летательных аппаратов. Цель соревнований состоит в том, чтобы один из летательных аппаратов команды переместился на максимальное расстояние от линии старта. Состязание проходит на трассе, представляющей собой полубесконечную (бесконечную в одну сторону) полосу шириной 40 метров. Маневры, связанные с изменением высоты полета, не допускаются (таким образом, трасса соревнования двумерна).

Каждая команда состоит из  $N$  летательных аппаратов. В дальнейшем, кроме термина «соревнование», будем использовать термин «гонка».

В начале гонки аппараты первой команды располагаются в воздухе случайным образом на некотором расстоянии от линии старта в левой половине трассы, которая на экране расположена горизонтально. Вторая команда размещается симметрично первой на правой половине трассы.

Для каждого аппарата заданы начальная скорость и направление движения. В простейшем случае начальные скорости всех аппаратов одинаковы, а направления – строго вперед. Система также позволяет делать начальные скорости и направления различными. Летательные аппараты в процессе полета могут поворачивать.

Каждый летательный аппарат имеет определенный запас топлива, расходуемого в процессе движения. По команде «Старт» все аппараты начинают движение с целью максимально удалиться от линии старта. Они в процессе полета могут изменять скорость своего движения за счет изменения расхода топлива.

Беспилотные летательные аппараты, покинувшие трассу, считаются прекратившими гонку. Выходом за пределы коридора считается пересечение центром аппарата границы трассы.

Управление каждой командой выполняет программа, написанная на языке программирования *Java*.

### 1.1. ПРАВИЛА СОРЕВНОВАНИЙ

В каждом соревновании каждая из команд на старте имеет  $N$  беспилотных летательных аппаратов с полным запасом топлива. Исходно аппараты первой команды случайным образом располагаются на первых 25 метрах левой половины трассы. Летательные аппараты второй команды располагаются симметрично им в правой половине трассы (рис. 1).

Жизненный цикл беспилотного летательного аппарата в рассматриваемой задаче может быть описан графом переходов автомата, который может находиться в одном из трех состояний: «Полет», «Нормальное завершение гонки», «Аварийное завершение гонки» (рис. 2).

Обозначения, используемые на рис. 2, приведены в табл. 1.

Таблица 1. Используемые обозначения

Обозначение	Описание
v1	Летательный аппарат покинул пределы трассы (его центр пересек границу трассы)
v2	Скорость летательного аппарата стала меньше, чем один м/с
v3	Летательный аппарат столкнулся с другим летательным аппаратом
v_rel	Относительная скорость столкновения летательных аппаратов
fuel	Количество топлива, которое осталось у летательного аппарата

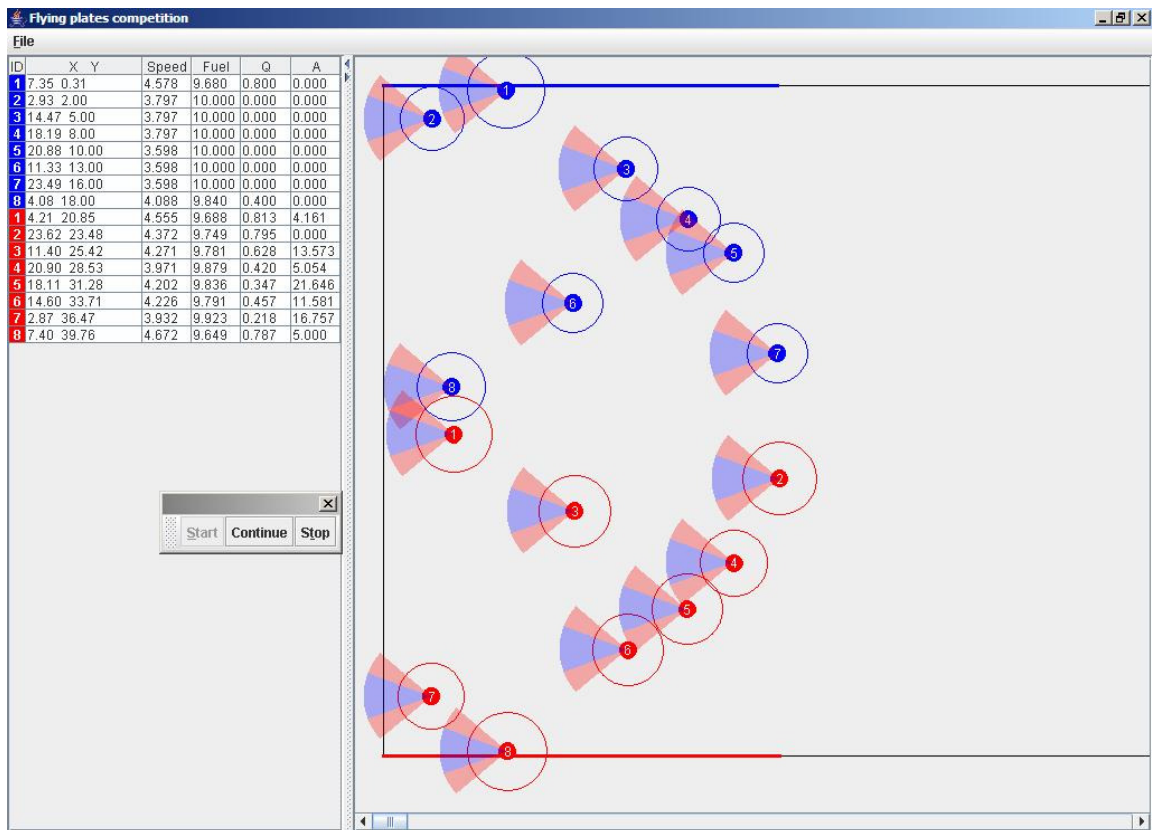


Рис. 1. Беспилотные летательные аппараты на старте

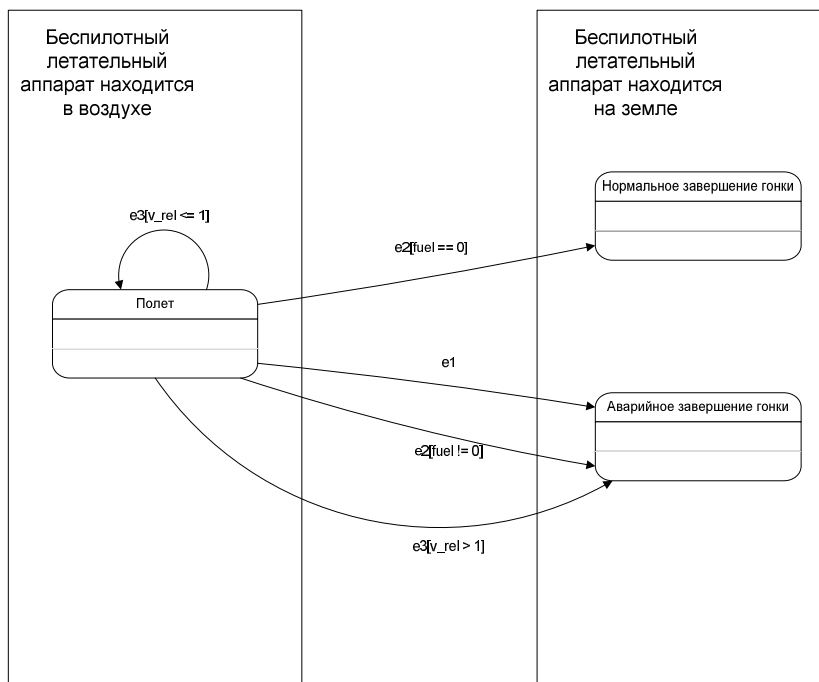


Рис. 2. Возможные состояния беспилотного летательного аппарата и переходы между ними

Поясним поведение беспилотного летательного аппарата. В начале гонки он находится в воздухе, исправен и способен продолжать участие в гонке. Этому соответствует состояние «Полет».

При выходе летательного аппарата за пределы трассы (событие  $e1$ ) он завершает гонку аварийно.

Если скорость летательного аппарата падает ниже одного м/с (событие  $e2$ ), и его топливный бак не пуст (условие  $fuel \neq 0$ ), то он завершает гонку аварийно. Если же при падении скорости ниже одного м/с (событие  $e2$ ) топливный бак летательного аппарата пуст (условие  $fuel == 0$ ), то он нормально завершает гонку.

Если летательный аппарат сталкивается с другим аппаратом (событие  $e3$ ), то при относительной скорости столкновения, большей одного м/с (условие  $v_{rel} > 1$ ), он аварийно завершает гонку. При относительной скорости столкновения, не превышающей одного м/с (условие  $v_{rel} \leq 1$ ), беспилотный летательный аппарат продолжает полет.

Заметим, что поскольку начальный запас топлива у каждой летательного аппарата конечен, то рано или поздно все беспилотные летательные аппараты обеих команд завершат гонку.

При подведении итогов гонки учитываются только результаты летательных аппаратов, нормально ее завершивших. Результатом команды считается наибольшее из расстояний, на которое удалились от линии старта ее летательные аппараты, которые нормально завершили гонку. Если все летательные аппараты команды вышли из гонки аварийно, результат команды считается равным нулю. Победителем признается команда, прошедшая наибольшее расстояние. В случае равенства результатов гонка считается завершившейся вничью.

## 1.2. ДИНАМИКА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

Беспилотный летательный аппарат представляет собой дискообразное «летающее крыло» радиусом один метр. На рис. 3 представлен вид сверху описываемого летательного аппарата.

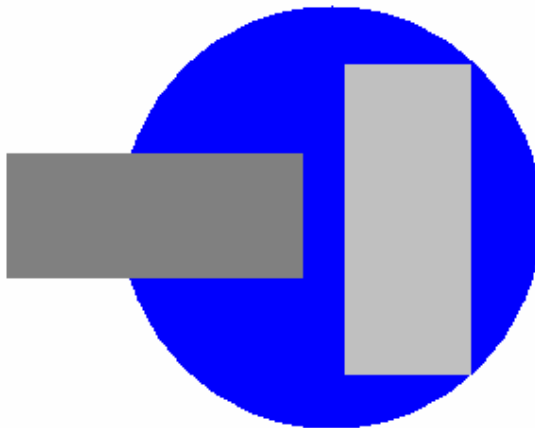


Рис. 3. Беспилотный летательный аппарат

Беспилотный летательный аппарат имеет реактивный двигатель (горизонтальный прямоугольник на рис. 3), топливный бак (вертикальный прямоугольник), аэродинамические рули и бортовой компьютер, способный регулировать расход топлива (и, как следствие, тягу двигателя) и положение аэродинамических рулей. Рули позволяют летательному аппарату маневрировать. Летательный аппарат может передвигаться со скоростями от одного метра в секунду. Его максимальная скорость зависит от запаса топлива и сопротивления воздуха. Ограничение в один метр в секунду вызвано тем, что летательный аппарат с меньшей скоростью не может держаться в воздухе.

Беспилотный летательный аппарат движется в соответствии со вторым законом Ньютона. Его движение определяется двумя силами: сопротивлением воздуха  $F$  и тягой двигателя  $T$ . Если тяга не

равна сопротивлению воздуха, то летальный аппарат движется с ускорением, которое может быть положительным (если тяга больше сопротивления воздуха) или отрицательным (если сопротивление воздуха больше тяги).

Ускорение определяется по формуле (1):

$$a = \frac{T - F}{m}, \quad (1)$$

где  $m$  – масса беспилотного летательного аппарата. При этом считается, что изменение его массы за счет выгорания горючего пренебрежимо мало.

Сопротивление воздуха определяется по формуле (2):

$$F = c_1 + c_2 v^2, \quad (2)$$

где  $v$  – скорость беспилотного летательного аппарата, а коэффициенты  $c_1$  и  $c_2$  определяются его аэродинамическими характеристиками и одинаковы для всех аппаратов обеих команд.

Тяга двигателя определяется по формуле (3):

$$T = c_4 q, \quad (3)$$

где  $q$  – расход топлива в сантиметрах кубических в секунду. Расход топлива находится под контролем бортового компьютера летательного аппарата, что позволяет изменять расход от нуля до единицы. Константа  $c_4$  определяется характеристиками двигателя беспилотного летательного аппарата и одинакова для всех аппаратов обеих команд.

Аэродинамические рули позволяют летательному аппарату поворачивать относительно его текущего направления движения на угол, не превышающий  $25^\circ$ .

### 1.3. АЭРОДИНАМИЧЕСКОЕ ВЗАИМОДЕЙСТВИЕ МЕЖДУ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ

При полете беспилотного летательного аппарата от траектории его полета в направлениях назад и в стороны под углом около  $30^\circ$  распространяются конические вихревые потоки воздуха. Если другой аппарат попадет в этот вихрь, то сопротивление воздуха его полету резко *снизится* (рис. 4).

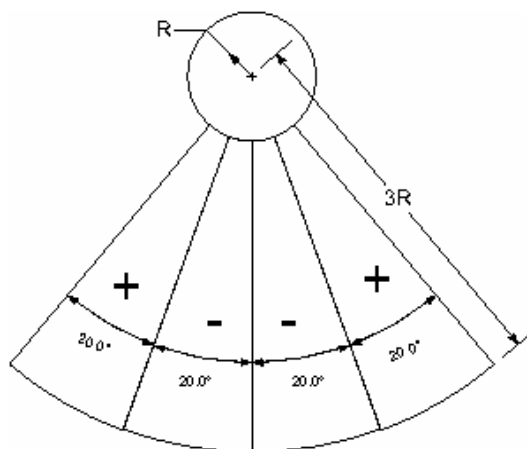


Рис. 4. Зоны повышенного и пониженного сопротивления воздуха

Отметим что, летательный аппарат, находящийся за хвостом (два сектора по  $20^\circ$ ) другого беспилотного летательного аппарата, испытывает *дополнительное сопротивление* движению, обусловленное реактивной струей.

Поясним, как учитывается изменение сопротивления воздуха. Если центр второго летательного аппарата находится в областях, отмеченных на рис. 5 знаком «+», сопротивление воздуха его движению падает на 50%. Если же центр второй летательного аппарата находится в области, помеченной знаком «-», сопротивление воздуха возрастает на 50%.

Аэродинамические воздействия от нескольких летательных аппаратов складываются, так что в зоне, отмеченной на рис. 5 знаками «+++», сопротивление воздуха вообще отсутствует, а в зонах, помеченных знаком «0», воздействия компенсируют друг друга. При этом в результате наложения зон воздействия от трех и более летательных аппаратов сопротивление воздуха не может стать отрицательным.

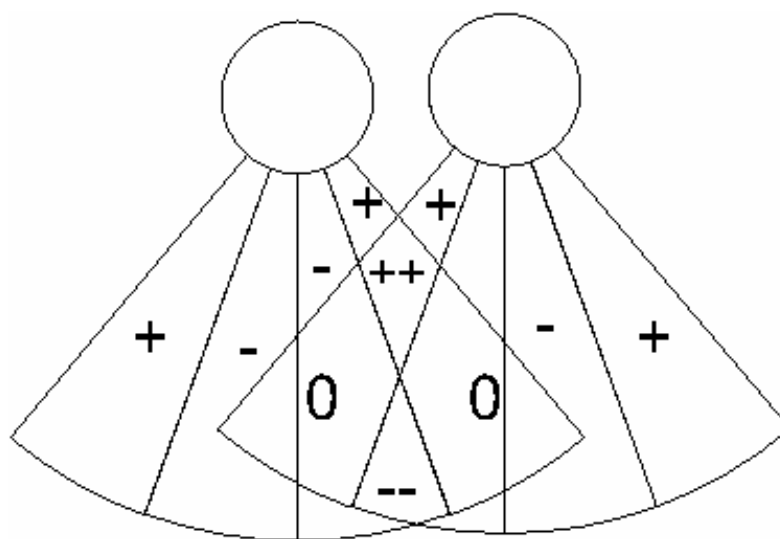


Рис. 5. Наложение областей аэродинамического взаимодействия двух летательных аппаратов

Учитывая изложенное, вычисление сопротивления воздуха происходит следующим образом. Пусть  $N_+$  – число летательных аппаратов, уменьшающих сопротивление воздуха в этой области, а  $N_-$  – число аппаратов, увеличивающих сопротивление воздуха. Пусть  $\Delta N = N_+ - N_-$ . Если  $\Delta N = 0$ , то в этой области нормальное аэродинамическое сопротивление, если  $\Delta N = 1$  или  $\Delta N = 2$ , то сопротивление понижается на  $50\Delta N$  процентов. Если  $\Delta N$  отрицательно, то сопротивление в этой области повышается на  $50|\Delta N|$  процентов.

#### 1.4. СТОЛКНОВЕНИЕ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

При столкновении двух беспилотных летательных аппаратов происходит их абсолютно упругое соударение без передачи вращательного момента. Если относительная скорость столкновения была более одного метра в секунду, то оба участвовавшие в столкновении беспилотных летательных аппарата повреждаются и аварийно завершают гонку. В рамках рассматриваемой модели поврежденные летающие аппараты не взаимодействуют между собой и аппаратами, продолжающими полет.

Под *относительной скоростью столкновения* понимается величина проекции векторной разности скоростей летательных аппаратов на прямую, проходящую через центры аппаратов в момент столкновения (рис. 6). На этом рисунке вектор  $V_{rel}$  соответствует относительной скорости.

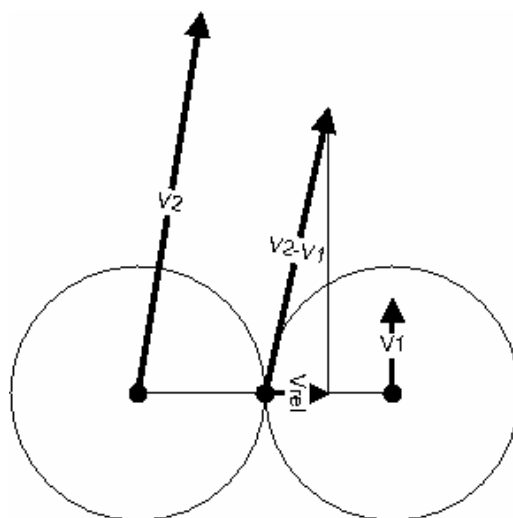


Рис. 6. Относительная скорость столкновения двух летательных аппаратов

### 1.5. МОДЕЛИРОВАНИЕ ГОНКИ

Моделирование гонки происходит по ходам, каждый из которых занимает  $t$  миллисекунд (параметр  $t$  читается из конфигурационного файла). В начале каждого хода игроки обладают информацией о координатах и скоростях всех беспилотных летательных аппаратов. Каждому игроку предоставляется возможность установить расход топлива и угол поворота аэродинамических рулей каждого летательного аппарата своей команды.

Каждые  $t$  миллисекунд (один ход) происходит обновление параметров. Покажем, как выполняется моделирование полета беспилотных летательных аппаратов за время одного хода.

Во-первых, производится снятие с соревнования летательных аппаратов, движущихся со скоростью, меньшей одного метра в секунду. При завершении полета летательными аппаратами с пустыми баками пройденные ими расстояния засчитываются в результат команды.

Во-вторых, производится расчет ускорений летательных аппаратов в соответствии с установленными расходами топлива и углами поворотов, а также аэродинамическим сопротивлением.

В-третьих, производится расчет новых скоростей летательных аппаратов по формуле (4):

$$\vec{V}_{temp} = \vec{V}_{old} + \vec{a} \cdot \Delta t, \quad (4)$$

где  $V_{temp}$  – вектор скорости летательного аппарата после учета ускорения,  $V_{old}$  – вектор старой скорости летательного аппарата,  $a$  – вектор ускорения. После этого происходит поворот вектора скорости на угол равный углу поворота аэродинамических рулей (рис. 7). В результате поворота получается вектор новой скорости беспилотного летательного аппарата  $V_{new}$ .

В-четвертых, производится снятие с соревнования беспилотных летательных аппаратов, движущихся медленнее одного метра в секунду. Как и ранее, при завершении полета летательными аппаратами с пустыми баками, пройденные ими расстояния засчитываются в результат команды.

В-пятых, происходит равномерное прямолинейное движение летательных аппаратов (считается, что за время шага моделирования их скорости не меняются). Если при этом происходит соударение аппаратов – расстояние между центрами каких-либо двух аппаратов становится меньше двух метров, то их скорости и координаты изменяются в соответствии с законами сохранения импульса и энергии. При этом летательные аппараты, относительная скорость столкновения которых превосходила один метр в секунду, выбывают из гонки.

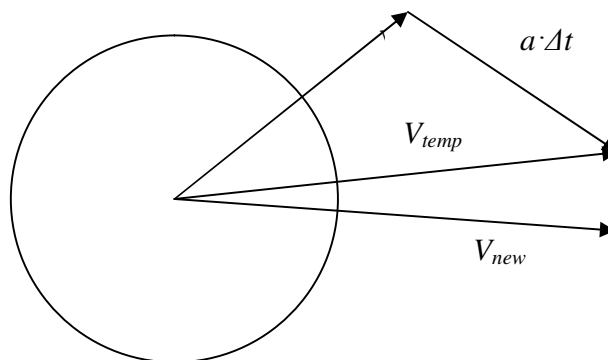


Рис. 7. Пересчет скорости летательного аппарата на шаге моделирования

В-шестых, производится проверка того, что все беспилотные летательные аппараты находятся в пределах трассы. При выходе центра аппарата за пределы трассы, он выбывает из гонки.

Гонка продолжается до тех пор, пока ее не завершил хотя бы один летательный аппарат. После того, как ее закончит и этот аппарат, гонка завершается.

Отметим, в чем состоит различие понятий «ход» и «шаг» в настоящей работе. Под **шагом** моделирования подразумевается квант времени в программе. Все временные промежутки в программе должны быть ему кратны. При этом передача управления системам управления летательными аппаратами (разрешение каждой из них произвести *ход*) выполняется через промежутки времени, равные шагу моделирования. Между двумя соседними ходами состояние внешней среды изменяется так, как будто между ними прошло время, равное шагу моделирования.

## 1.6. КРИТЕРИИ ОЦЕНКИ СИСТЕМ УПРАВЛЕНИЯ

Критериями оценки систем управления в рамках поставленной задачи являются:

1. Результат (максимальное расстояние, на которое удалился от линии старта нормально завершивший гонку беспилотный летательный аппарат команды), показываемый командой беспилотных летательных аппаратов, управляемых построенной системой управления.
2. Число беспилотных летательных аппаратов команды, нормально завершивших соревнование.
3. Степень автоматизации построения системы управления.
4. Временные затраты на построение системы управления.
5. Сложность системы управления.

Первый из указанных критериев отражает эффективность оцениваемой системы конечных автоматов для решаемой задачи управления беспилотным летательным аппаратом. Оценка системы по этому критерию проводится в результате проведения нескольких соревнований оцениваемой системы управления моделью беспилотного летательного аппарата с другими системами. При этом в качестве значения параметра полагается средний результат, показанный оцениваемой системой. Отметим, что в расчет берется не только указанное среднее значение, но и распределение показанных результатов. Этот критерий, наряду со вторым критерием (число беспилотных летательных аппаратов команды, нормально завершивших соревнование), является основным при оценке системы управления беспилотным летательным аппаратом в данной задаче.

Второй критерий (число беспилотных летательных аппаратов команды, нормально завершивших соревнование) отражает устойчивость результатов, показываемых системой управления моделью беспилотного летательного аппарата, относительно противодействия команды соперника.

Оценка системы по этому критерию проводится в результате проведения нескольких соревнований оцениваемой системы управления моделью беспилотного летательного аппарата с другими системами. При этом в качестве значения параметра полагается среднее число успешно завершивших соревнования летательных аппаратов. Отметим, что в расчет берется не только указанное среднее значение, но и распределение числа успешно завершивших гонку летательных аппаратов. Этот критерий, наряду с результатом, показываемым командой беспилотных летательных аппаратов, управляемых построенной системой управления, является основным при оценке системы управления моделью беспилотного летательного аппарата

Третий критерий (степень автоматизации построения системы управления) оценивает число этапов построения системы управления моделью беспилотного летательного аппарата, которые проводятся без участия человека или при его минимальном участии. Этапами построения, которые могут быть автоматизированы, являются: построение поставщиков событий, построение объекта управления, построение системы взаимодействующих управляющих автоматов.

Четвертый критерий (временные затраты на построение системы управления) отражает время, потраченное на построение оцениваемой системы управления моделью беспилотного летательного аппарата. Оно включает в себя все время, потраченное на разработку и программную реализацию поставщиков событий, объектов управления, системы управляющих автоматов. При использовании генетических алгоритмов для построения системы автоматов в это время включается время работы алгоритма.

Под сложностью системы управления понимаются такие показатели, как число суммарное состояний автоматов, входящих в систему управления, суммарное число переходов в этих автоматах. Кроме этого, необходимо учитывать наличие или отсутствие в системе управления других компонентов, таких, как, например, нейронная сеть, и сложность их структуры.

## 1.7. Выводы

Разработанная модель беспилотного летательного средства позволит оценить качество систем управления. С одной стороны, модель достаточно проста, что обеспечивает возможность эффективно строить системы управления и оценивать их качество. С другой стороны, она учитывает не только аэродинамику отдельных летательных аппаратов, но и взаимодействие между ними, в том числе, столкновения. Специально отметим, что задача самоорганизации беспилотных летательных аппаратов является весьма актуальной с практической точки зрения, в частности, с военной точки зрения, о чем свидетельствуют соответствующие исследования [22].

Разработаны критерии, которые позволят объективно сравнивать качество систем управления моделью беспилотного летательного аппарата. Это позволит оценить методы генерации автоматов управления системами со сложным поведением.



## 2. СИСТЕМА УПРАВЛЕНИЯ, ПОСТРОЕННАЯ БЕЗ ПРИМЕНЕНИЯ МЕТОДОВ АВТОМАТИЗАЦИИ

В работе [27] описана система управления беспилотными летательными аппаратами, построенная с использованием парадигмы автоматного программирования [24, 31], SWITCH-технологии и инструментального средства *UniMod* [19, 8]. Все автоматы в указанной работе были построены без применения методов автоматизации, а такое построение было достаточно трудоемким. Этот подход кратко описан в настоящем разделе.

Описываемая система взаимодействует с ядром программы моделирования соревнований беспилотных летательных аппаратов, которое также описано в работе [27].

### 2.1. КРАТКОЕ ОПИСАНИЕ СТРАТЕГИИ

При разработке системы управления беспилотными летательными аппаратами было сделано предположение, что победной будет стратегия, при которой одна половина аппаратов движется вперед и уклоняется от границ коридора и рядом находящихся аппаратов, а вторая половина аппаратов разбивается на пары. При этом аппараты в парах движутся, поддерживая взаимное расположение для использования аэродинамического взаимодействия. Эта стратегия почти всегда оказывается победной по сравнению со стратегией, реализованной для противника: один аппарат летит вперед, а другие – пытаются сбить аппараты другой команды.

### 2.2. ДИАГРАММА СВЯЗЕЙ

В данной версии системы управления беспилотным летательным аппаратом поведение всех летательных аппаратов «нашей» команды описывается одной и той же системой взаимодействующих автоматов. Диаграмма связей, в которую входят эти автоматы, показана на рис. 8. Автомат *Состояние агента AFP* имеет три состояния: «Летит», «Нормальное завершение», «Аварийное завершение».

В первое состояние этого автомата вложен автомат *Режим полета AL* и автомат *Радар AR*. В состоянии «Полет в одиночку» автомата *Режим полета* вложены автомат *Уклонение от границ трассы и аппаратов справа и слева A1* и автомат *Уклонение от аппаратов спереди и сзади A3*.

В состоянии «Полет первым в паре» и «Полет вторым в паре» автомата *Режим полета* также вложен автомат *Уклонение от границ трассы и аппаратов справа и слева A1* и автомат *Уклонение от аппаратов спереди и сзади A3*

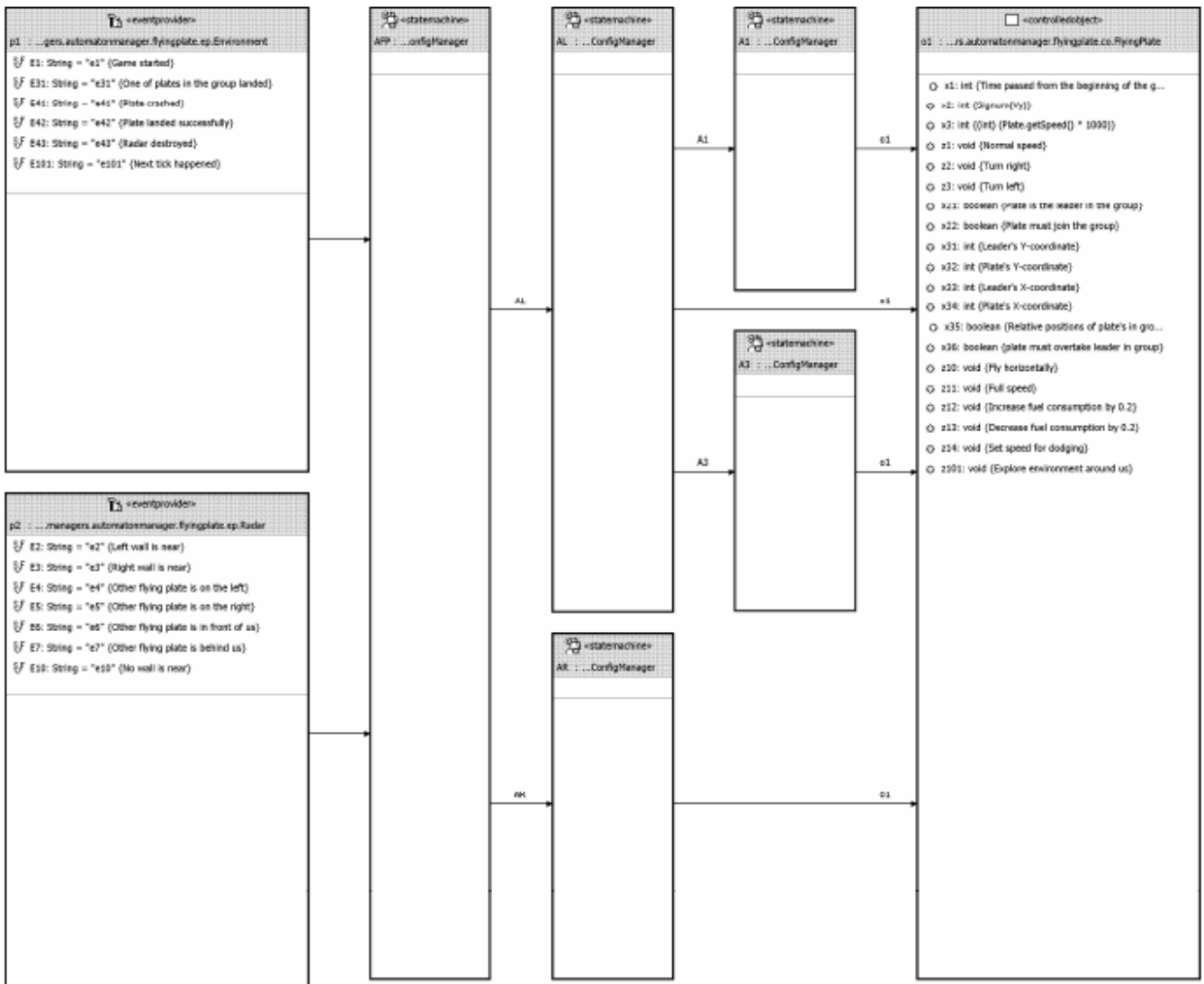


Рис. 8. Диаграмма связей системы управления беспилотным летательным аппаратом

### 2.3. ПОСТАВЩИКИ СОБЫТИЙ

В настоящем разделе описаны поставщики событий

#### 2.3.1. Поставщик событий Environment

Этот поставщик событий служит для передачи воздействий на беспилотный летательный аппарат от игрового мира. События, генерируемые этим поставщиком событий, перечислены в табл. 2.

Таблица 2. События поставщика событий Environment

	Событие	Описание	Параметр	Комментарий
1.	e1	Соревнование началось	Нет параметров	
2.	e31	Второй из аппаратов в паре разбился или приземлился	Нет параметров	
3.	e41	Беспилотный летательный аппарат разбился	нет параметров	
4.	e42	Беспилотный летательный аппарат благополучно приземлился	Нет параметров	
5.	e43	Радар сломался	Нет параметров	
6.	e101	Поставщик событий ConfigurableTimer из ядра программы создал событие e101	Нет параметров	Таким образом, событие e101 из ядра программы ретранслируется системе управления моделью беспилотного летательного аппарата

### 2.3.2. Поставщик событий Radar

Поставщик событий Radar содержит события (табл. 3), связанные с изменением взаимного расположения данного беспилотного летательного аппарата относительно других летательных аппаратов и стен.

Таблица 3. События поставщика событий Radar

	Событие	Описание	Параметр	Комментарий
1.	e2	Аппарат находится близко к левой границе трассы	Нет параметров	
2.	e3	Аппарат находится близко к правой границе трассы	Нет параметров	
3.	e4	Другой аппарат находится слева	Нет параметров	
4.	e5	Другой аппарат находится справа	Нет параметров	
5.	e6	Другой аппарат находится спереди	Нет параметров	
6.	e7	Другой аппарат находится сзади	Нет параметров	
7.	e10	Рядом нет границ трассы	Нет параметров	

## 2.4. АВТОМАТЫ

В этом разделе описаны конечные автоматы, реализующие искусственный интеллект системы управления беспилотным летательным аппаратом.

### 2.4.1. Автомат АФР – Состояние агента

Этот автомат предназначен для управления беспилотным летательным аппаратом на самом верхнем уровне. Его граф переходов показан на рис. 9.

Автомат АФР содержит состояния, соответствующие всем трем возможным этапам жизненного цикла беспилотного летательного аппарата. Состояние *Alive* соответствует тому, что беспилотный летательный аппарат находится в воздухе, – в него вложен автомат AL, отвечающий за управление полетом. Состояние *Landed successfully* соответствует тому, что летательный аппарат благополучно приземлился, а состояние *Crashed* – тому, что он разбился.

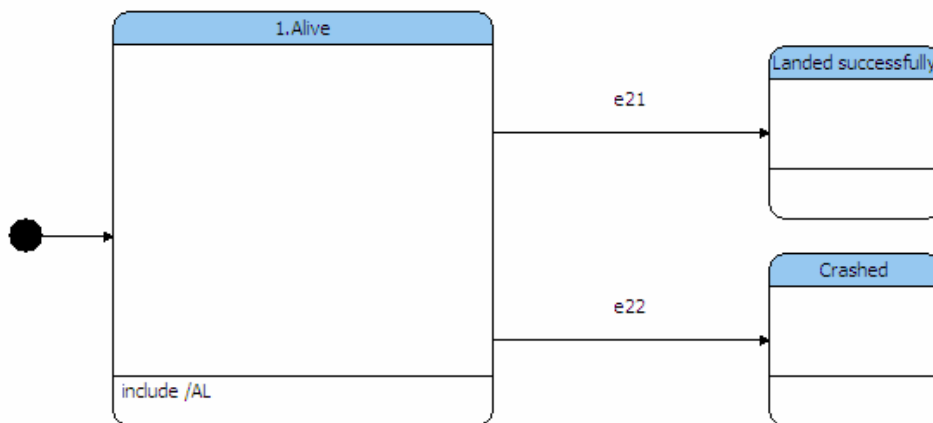


Рис. 9. Граф переходов автомата AFP

В начале гонки автомат находится в состоянии *Alive*. При поступлении события  $e_{21}$  (аппарат благополучно приземлился) автомат переходит в состояние *Landed successfully*, а при поступлении события  $e_{22}$  (аппарат разбился) – в состояние *Crashed*.

#### 2.4.2. Автомат AL – Режим полета

Автомат AL «отвечает» за полет беспилотного летательного аппарата. Его граф переходов представлен на рис. 10.

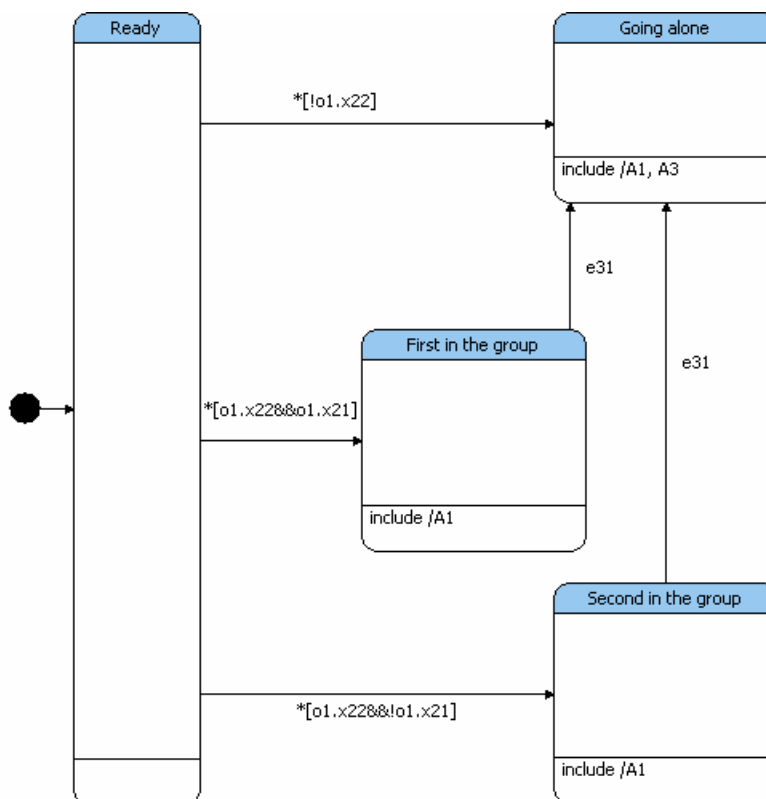


Рис. 10. Граф переходов автомата AL

### 2.4.3. Автомат А1 - Уклонение от границ коридора и агентов справа и слева

Этот автомат «следит» за тем, чтобы беспилотный летательный аппарат не выходил за границы трассы и не сталкивался с другими летательными аппаратами. Его граф переходов показан на рис. 11.

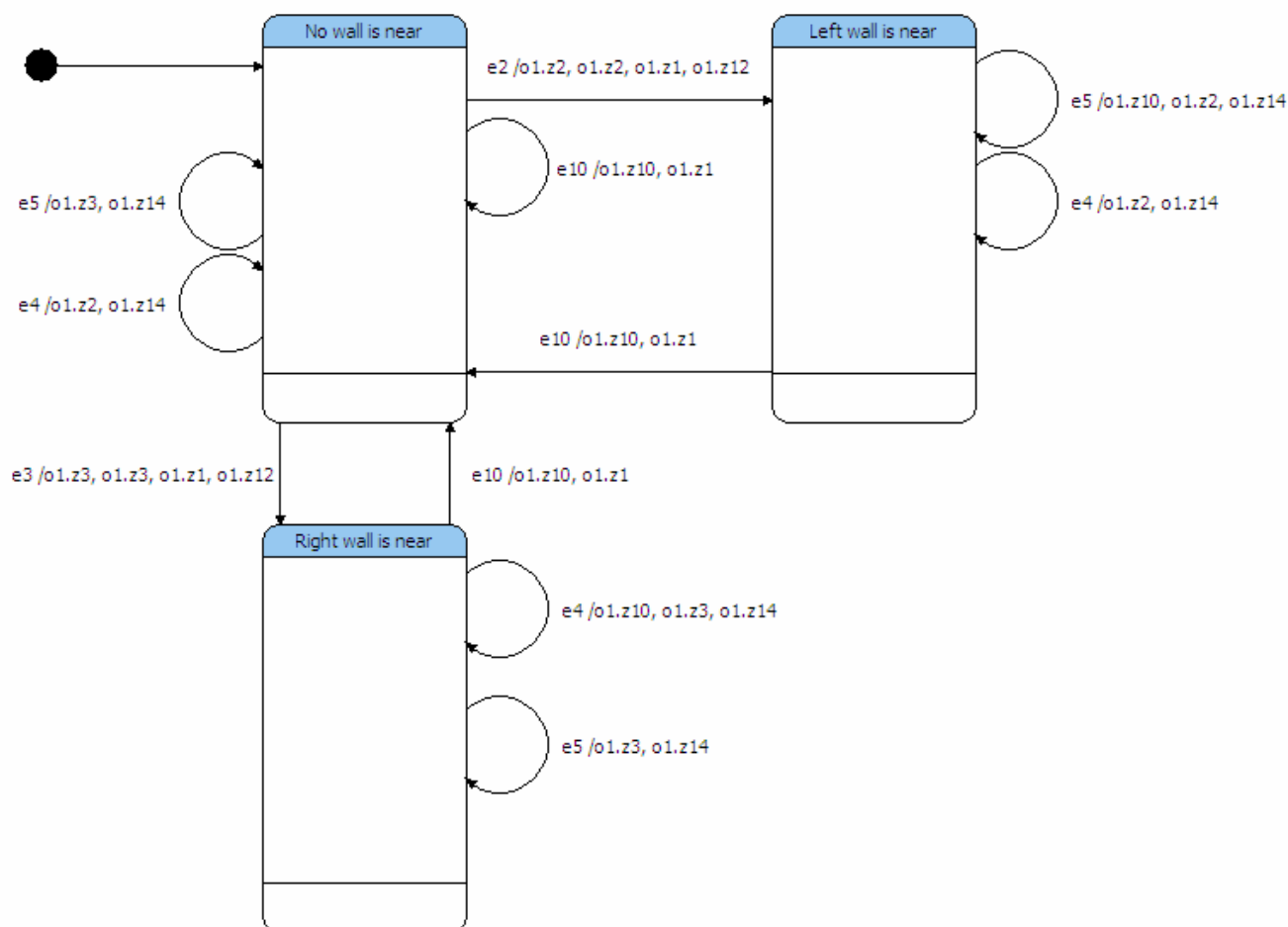


Рис. 11. Граф переходов автомата А1

Автомат А1 имеет три состояния. Состояние «*No wall is near*» соответствует тому, что агент находится далеко от границ коридора, в котором происходят соревнования. Находясь в нем, автомат обрабатывает события, связанные с приближением управляемого аппарата к границам трассы, либо с появлением рядом с ним других летательных аппаратов.

При приближении управляемого этим автоматом летательного аппарата к левой границе трассы (событие  $e_2$ ) автомат переходит в состояние *Left wall is near*. При этом переходе агент поворачивает направо, а расход топлива устанавливается равным 0.6.

Аналогично происходит обработка события  $e_3$  (аппарат находится вблизи правой границы трассы), только вместо поворота направо производится поворот влево.

### 2.4.4. Автомат AR – Радар

Автомат AR, граф переходов которого показан на рис. 12, управляет радаром беспилотного летательного аппарата.

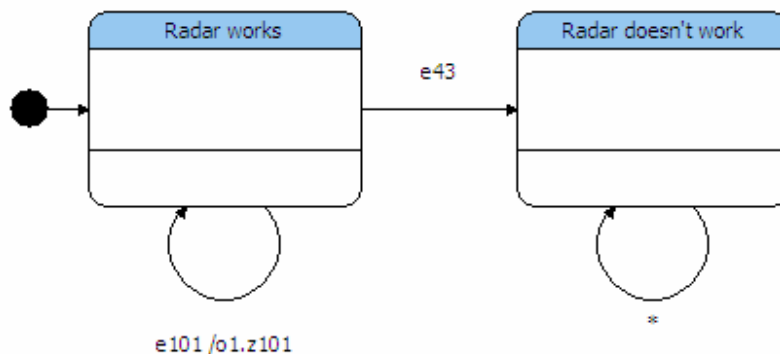


Рис. 12. Граф переходов автомата AR

Автомат AR имеет два состояния. Состояние *Radar works* соответствует тому, что радар исправен. Находясь в нем, автомат по событию e101 вызывает метод o1.z101, который выполняет «исследование» окружающей среды. В результате этого генерируются события, связанные с приближением летательного аппарата к границам трассы или к другим летательным аппаратам. По событию e43 (радар поврежден) автомат переходит в состояние *Radar doesn't work*, в котором он при поступлении любого события не выполняет никаких действий.

#### 2.4.5. Автомат A3 – Уклонение от летательных аппаратов спереди и сзади

Этот автомат «следит» за летательными аппаратами, находящимися в непосредственной близости спереди, либо сзади. Его граф переходов показан на рис. 13.

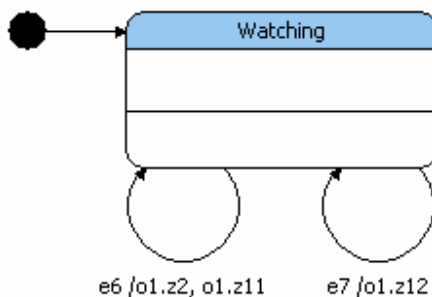


Рис. 13. Граф переходов автомата A3

Автомат A3 имеет одно состояние и два перехода. Один из переходов соответствует обнаружению другого беспилотного летательного аппарата спереди – в этом случае управляемый этим автоматом беспилотный летательный аппарат уменьшает расход топлива (и, как следствие, свою скорость). Обнаружение другого беспилотного летательного аппарата сзади обрабатывается аналогично, но в этом случае расход топлива увеличивается.

### 2.5. ОБЪЕКТ УПРАВЛЕНИЯ FLYINGPLATE

Этот объект управления представляет собой беспилотный летательный аппарат, которым производится управление. Он содержит выходные воздействия, связанные с управлением аэродинамическими рулями и расходом топлива. Выходные воздействия этого объекта управления перечислены в табл. 4.

Таблица 4. Выходные воздействия объекта управления FlyingPlate

Выходное воздействие	Описание	Комментарий
z1	Установить расход топлива равным 0.4	
z2	Повернуть направо	
z3	Повернуть налево	
z10	Лететь горизонтально	
z11	Максимальный расход топлива	
z12	Увеличить расход топлива на 0.2	
z13	Уменьшить расход топлива на 0.2	
z14	Установить расход топлива равным 0.8	

Входные воздействия объекта управления FlyingPlate перечислены в табл. 5.

Таблица 5. Входные воздействия объекта управления FlyingPlate

Входное воздействие	Описание	Комментарий
x1	Время, прошедшее с начала гонки	Измеряется числом событий e101
x2	Знак проекции скорости агента на ось Oy	
x3	Величина скорости агента	
x21	Летательный аппарат должен лететь первым в паре	
x22	Летательный аппарат должен лететь в паре с другим аппаратом	
x31	Y-координата первого в паре	
x32	Y-координата летательного аппарата	
x33	X-координата первого в паре	
x34	X-координата летательного аппарата	
x35	Взаимное расположение аппаратов в паре правильно	
x36	Летательный аппарат должен обогнать первого в паре	

## 2.6. СИСТЕМА УПРАВЛЕНИЯ С «АГРЕССИВНОЙ» СТРАТЕГИЕЙ

Система управления с «агрессивной» стратегией реализует следующий алгоритм: один беспилотный летательный аппарат движется строго вперед, а другие беспилотные летательные аппараты пытаются сбить или вытолкнуть за пределы трассы беспилотные летательные аппараты соперника.

## 2.7. СИСТЕМА УПРАВЛЕНИЯ С «ПРОСТОЙ» СТРАТЕГИЕЙ

Система управления с «простой» стратегией реализует следующий алгоритм: все беспилотные летательные аппараты движутся строго вперед

## **2.8. Выводы**

Система управления моделью беспилотного летательного аппарата построена на основе методов проектирования, принятых в автоматном программировании. Отметим, что выделенные поставщики событий и объекты управления могут быть применены при генерации системы управления.

Разработанная система управления будет являться точкой отсчета при сравнении систем управления, построенных на основе методов генерации автоматов управления системами со сложным поведением. Это позволит оценить указанные методы и провести их экспериментально исследование.

Рассмотрены системы управления с «агрессивной» и «простой» стратегиями, которые будут использоваться при автоматическом построении других систем управления.



### 3. АВТОМАТИЗИРОВАННОЕ ПОСТРОЕНИЕ СИСТЕМ УПРАВЛЕНИЯ

В настоящем разделе описано три подхода к автоматическому построению управляющих систем для рассмотренной выше модели беспилотного летательного аппарата. Все эти подходы основаны на применении генетических алгоритмов (генетического программирования) [17, 18, 23, 25, 30, 1–16], но используют различные способы представления автоматов и генерируют управляющие системы различной структуры.

#### 3.1. СОВМЕСТНОЕ ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

В настоящем разделе описан подход, основанный на совместном применении генетического программирования, искусственных нейронных сетей и конечных автоматов.

##### 3.1.1. Некоторые проблемы, возникающие при использовании генетического программирования для построения конечных автоматов

Генетическое программирование [10] наиболее эффективно в тех случаях, когда оптимизируемый объект (например, конечный автомат) имеет небольшой размер (небольшое число состояний). В то же время, число различных вариантов значений входных переменных может быть достаточно большим, а сами переменные могут быть не только логическими, но и числовыми. Например, в рассматриваемой задаче такие входные переменные, как скорость беспилотного летательного аппарата или его координаты являются вещественными.

Обычно с помощью генетических алгоритмов строятся автоматы с входными переменными логического типа. Для того чтобы применять такой алгоритм, необходимо разработать способ перехода от произвольных входных переменных к логическим входным переменным (или, хотя бы, к переменным, множество значений которых конечно и содержит небольшое число элементов).

Одним из вариантов решения этой задачи является введение соответствующих переменных. Например, если исходно были две вещественные переменные  $x$  и  $y$ , то, например, можно ввести две новые логические переменные  $A$  равную  $x > 100$  и  $B$  равную  $y < 200$ .

Второй вариант решения состоит в том, чтобы разбить множество значений входных переменных на несколько областей и использовать в качестве значения входной переменной номер области, в которой лежат текущие значения входных переменных. Таким образом, перед тем, как подавать данные на вход автомата, необходимо определять, в какой из областей лежит набор текущих значений входных переменных. Это – *задача классификации*. Если для ее решения применять автоматический классификатор (нейронная сеть, дерево принятия решений, и т.д.), то возникает идея настраивать этот классификатор совместно с автоматом, с которым он связан.

В настоящем разделе описывается одна из возможных реализаций второго подхода. В качестве классификатора используется нейронная сеть. Ее настройка и построение автомата производится совместно с помощью генетического программирования.

##### 3.1.2. Искусственные нейронные сети

*Нейрон* – это клетка головного мозга или нервной системы, основной функцией которой является сбор, обработка и распространение электрических сигналов. Считается, что способность мозга к обработке информации обусловлена функционированием сетей, состоящих из нейронов.

Одна из первых математических моделей нейрона предложена Мак-Каллоком (McCulloch) и Питтсом (Pitts) [13]. С 1943 года были разработаны более подробные и реалистичные модели, как нейрона, так и более крупных систем мозга. Это привело к созданию новой научной области – *вычислительной неврологии*. С другой стороны, исследователи в области искусственного интеллекта

исследовали более абстрактные свойства нейронных сетей: способность выполнять распределенные вычисления, справляться с зашумленными входными данными, обучаться и т. д.

Со временем стало ясно, что похожими свойствами обладают и другие системы (такие, как, например, байесовские сети). Однако *искусственные нейронные сети* (в дальнейшем, нейронные сети) [30] по сей день остаются одним из наиболее изученных и широко применяемых методов искусственного интеллекта.

### 3.1.2.1. Элементы нейронных сетей

Нейронные сети состоят из *узлов* (искусственных нейронов), соединенных между собой *связями*. Связь от элемента  $i$  к элементу  $j$  предназначена для распространения *активации*  $a_j$  от  $j$  к  $i$ . Каждая связь имеет назначенный ей числовой *вес*  $W_{i,j}$ . Каждый элемент вычисляет взвешенную сумму своих входных данных (5):

$$in_i = \sum_{j=0}^n W_{j,i} a_j \quad (5)$$

и применяет к ней *функцию активации*  $g$  (6):

$$a_i = g(in_i) = g\left(\sum_{j=0}^n W_{j,i} a_j\right). \quad (6)$$

Обратим внимание на то, что в эту формулу входит *смещенный вес*  $W_{0,i}$ , относящийся к постоянному входному значению  $a_0 = -1$ .

Основными видами функций активации являются:

- *пороговая функция* (7)

$$g(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}; \quad (7)$$

- *знаковая функция* (8)

$$g(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}; \quad (8)$$

- *сигмоидальная (логистическая) функция* (9)

$$g(x) = \frac{1}{1 + e^{-x}}. \quad (9)$$

Отметим, что эти функции имеют пороговое значение около нуля, а смещенный вес  $W_{0,i}$  фактически задает пороговое значение для данного элемента.

Важным свойством элементов с пороговой функцией активации является то, что с их помощью можно представить логические функции *AND*, *OR* и *NOT* [30]. Таким образом, с помощью нескольких таких элементов можно выразить любую булеву функцию от входов сети.

### 3.1.2.2. Структура нейронных сетей

Существуют две основные категории структур нейронных сетей: ациклические сети (*сети с прямым распространением*) и циклические (*рекуррентные*) сети.

Сети с прямым распространением реализуют некоторую функцию от своих входов, в то время как в рекуррентной сети выходы ее элементов могут подаваться на вход. В связи с этим уровни активации в рекуррентной сети могут находиться в устойчивом состоянии, могут переходить в колебательный или даже в хаотический режим.

Рекуррентные сети представляют собой более сложную для понимания и исследования модель. В настоящем разделе в дальнейшем будут использоваться только сети с прямым распространением.

### 3.1.2.3. Применение нейронных сетей

Наиболее часто нейронные сети применяются для решения следующих задач:

- *классификация образов* – указание принадлежности входного образа, представленного вектором признаков, одному или нескольким предварительно определенным классам;
- *кластеризация* – классификация образов при отсутствии обучающей выборки с метками классов;
- *прогнозирование* – предсказание значения  $y_{n+1}$  при известной последовательности  $y_1, y_2 \dots y_n$ .

В настоящей работе, как отмечалось выше, нейронные сети используются для получения по вещественным значениям входных переменных логических переменных, которые подаются на вход конечного автомата.

### 3.1.3. Структура системы управления беспилотным летательным аппаратом

Каждый беспилотный летательный аппарат управляется системой, состоящей из нейронной сети и конечного автомата. Таким образом, можно говорить, что используется *мультиагентный подход* [30] – каждый летательный аппарат представляет собой агента, взаимодействующего с внешней средой и другими агентами.

При этом, как отмечалось выше, нейронная сеть используется для классификации значений вещественных входных переменных и выработки входных логических переменных для автомата, а автомат – для выработки выходных воздействий на беспилотный летательный аппарат (рис. 14).

Структура нейронной сети и способ ее взаимодействия с конечным автоматом показаны на рис. 15.

Символами **S** на рис. 15 обозначены нейроны с сигмоидальной функцией активации, символом **L** – нейроны с пороговой функцией активации. Рядом с нейронами указаны их номера (они используются при описании операции скрещивания нейронных сетей). На каждый из трех выходов нейронной сети поступает число равное нулю или единице. Таким образом, существует восемь вариантов комбинаций выходных сигналов нейронной сети (000, 001, 010, 011, 100, 101, 110, 111), подаваемых на вход конечного автомата.

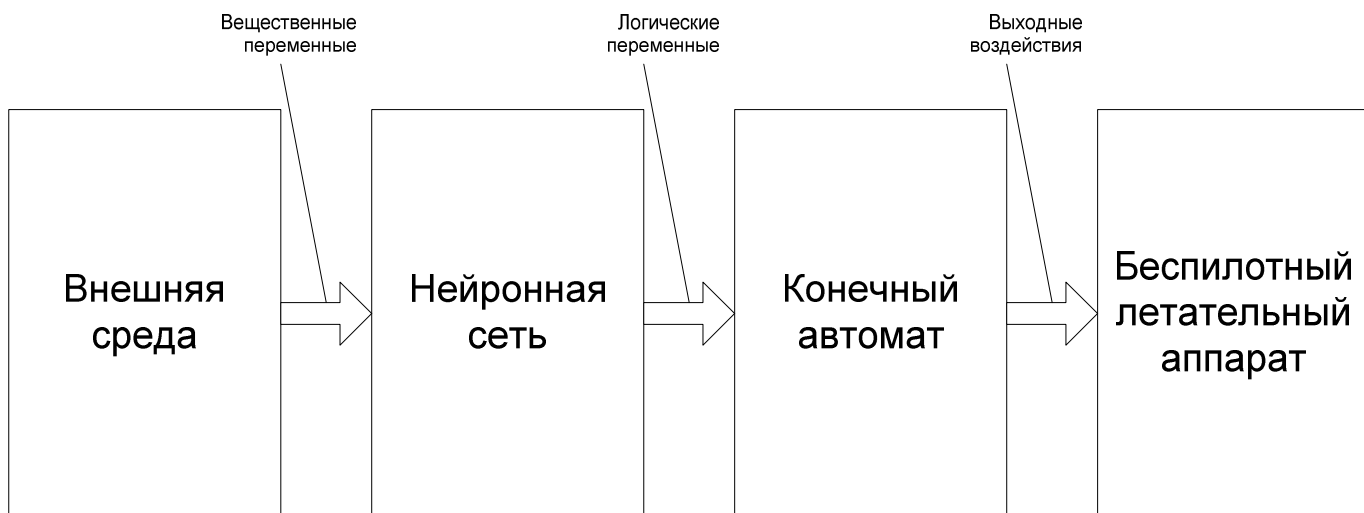


Рис. 14. Структурная схема системы управления беспилотным летательным аппаратом

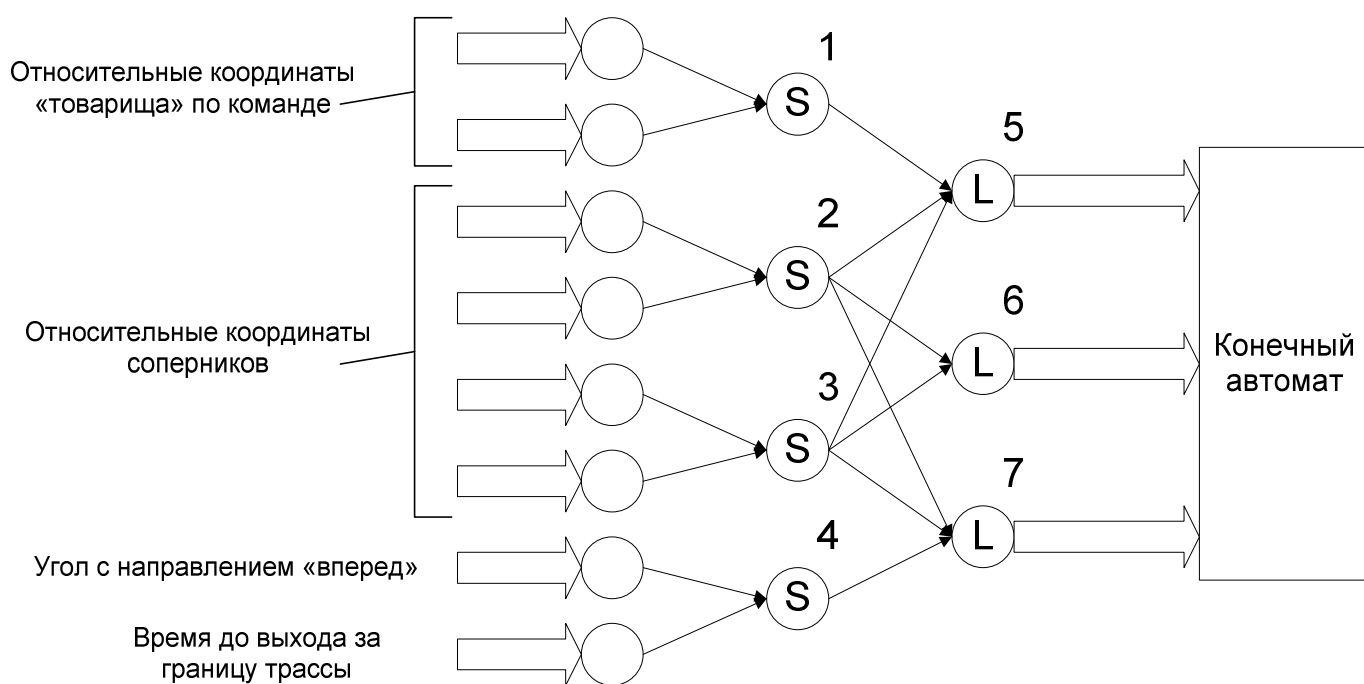


Рис. 15. Нейронная сеть и ее взаимодействие с конечным автоматом

### 3.1.4. Алгоритм генетического программирования для построения системы управления беспилотным летательным аппаратом

В настоящем разделе описан алгоритм генетического программирования, используемый для построения системы управления беспилотным летательным аппаратом. *Алгоритм генетического программирования* состоит из пяти частей:

- создание начального поколения;
- мутация;

- скрещивание (кроссовер);
- отбор особей для формирования следующего поколения;
- вычисление функции приспособленности (фитнес-функции).

### 3.1.4.1. Структура особи в используемом алгоритме

В связи с тем, что генетическое программирование эффективно только в случае небольшого размера особи, а число летательных аппаратов в каждой команде может быть достаточно велико, возникла идея совместно строить управляющие системы только для двух аппаратов. При этом предполагается, что беспилотный летательный аппарат может достаточно «хорошо» управляться даже при наличии небольшого объема информации о внешней среде. Поэтому была выбрана структура нейронной сети, указанная в предыдущем разделе. При этом отметим, что два летательных аппарата были выбраны только для генерации систем управления с помощью генетического программирования, а в дальнейшем построенные системы управления будут размножены в количестве, необходимом для формирования команды – шесть или восемь аппаратов.

Таким образом, особь в описываемом алгоритме генетического программирования состоит из двух систем управления беспилотным летательным аппаратом (рис. 16). Это означает, что она состоит из систем управления аппаратом, которые в дальнейшем будут входить в одну команду при проведении соревнований.

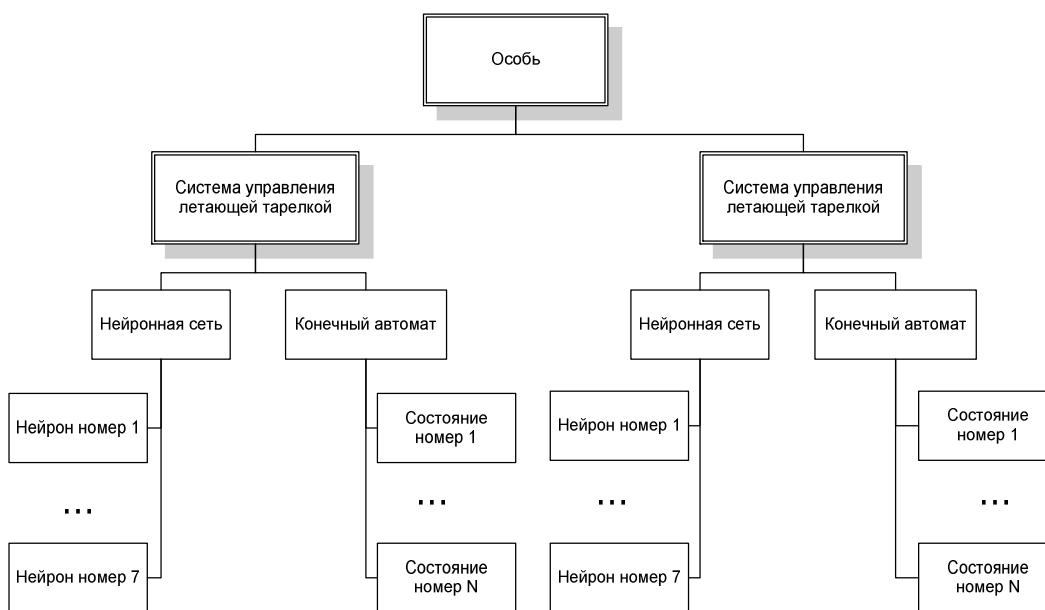


Рис. 16. Структура особи

Каждая система управления беспилотным летательным аппаратом состоит из нейронной сети и конечного автомата. Нейронная сеть состоит из четырех нейронов с сигмоидальной функцией активации и трех нейронов с пороговой функцией активации. Каждый из нейронов характеризуется порогом активации и весами связей, которые соединяют другие элементы сети с рассматриваемым. На языке программирования *Java* нейрон представляется следующим образом:

```

public abstract class Neuron {
    protected Neuron[] inputs;
    protected int inputsCnt;
    protected double[] w;
}
  
```

Описание конечного автомата состоит из номера начального состояния и описания состояний. Описание состояния состоит из описаний восьми переходов, соответствующих восьми вариантам значений входных переменных автомата. Описание каждого перехода состоит из номера состояния, в которое ведет этот переход, и двух действий, которые выполняются при выборе этого перехода – изменения расхода топлива и изменения угла поворота аэродинамических рулей. Каждое из этих действий характеризуется одним вещественным числом – соответственно, новым расходом топлива и углом поворота аэродинамических рулей:

```
public class Individual {
    protected PlateControlSystem[] pcs;
}
public class PlateControlSystem {
    private NeuralNet neuralNet;
    private Automaton automaton;
}
```

### 3.1.4.2. Создание начального поколения

Начальное поколение заполняется случайно сгенерированными системами управления. При этом в каждой системе управления случайным образом генерируется конечный автомат и нейронная сеть – веса связей в ней инициализируются случайными числами от минус единицы до единицы.

### 3.1.4.3. Операция мутации

**Мутация особи.** При мутации особи с равной вероятностью мутирует либо одна система управления летательным аппаратом, либо вторая.

**Мутация системы управления беспилотным летательным аппаратом.** При мутации системы управления беспилотным летательным аппаратом мутирует либо нейронная сеть, либо конечный автомат.

**Мутация нейронной сети.** При мутации нейронной сети мутирует случайно и равновероятно выбирается один элемент (искусственный нейрон) сети и мутирует.

**Мутация элемента сети.** При мутации элемента сети случайно выбирается один из весов связей, и к нему прибавляется случайное число из отрезка  $[-0.05; 0.05]$ . Кроме этого, с вероятностью 0.5 аналогичная операция производится с лимитом активации нейрона.

Мутация нейронной сети проиллюстрирована на рис. 17.

**Мутация конечного автомата.** При мутации конечного автомата равной вероятностью производится либо изменение начального состояния, либо мутация случайно выбранного перехода.

**Изменение начального состояния.** Начальное состояние изменяется на случайно выбранное состояние автомата.

**Мутация перехода.** При мутации перехода с равной вероятностью происходит либо изменение номера состояния, в которое ведет переход, либо мутация одного из действий, связанных с переходом – может мутировать действие, связанное с изменением расхода топлива (прибавляется случайное число от  $-0.05$  до  $0.05$ ) или с изменением угла поворота аэродинамических рулей (уменьшается или увеличивается на  $5^\circ$ ).

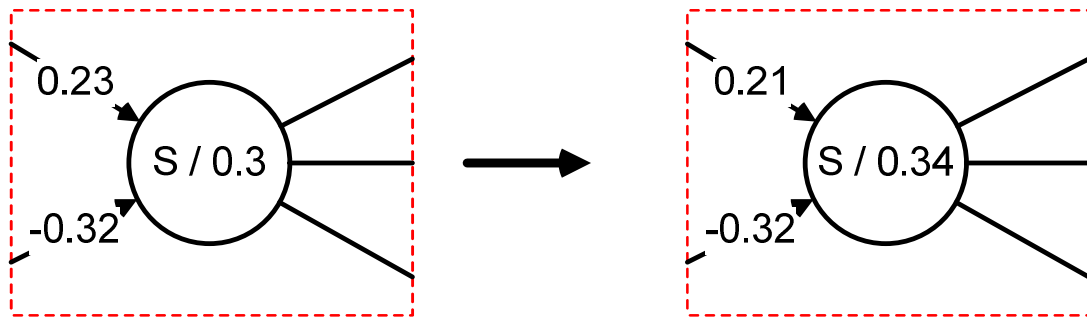
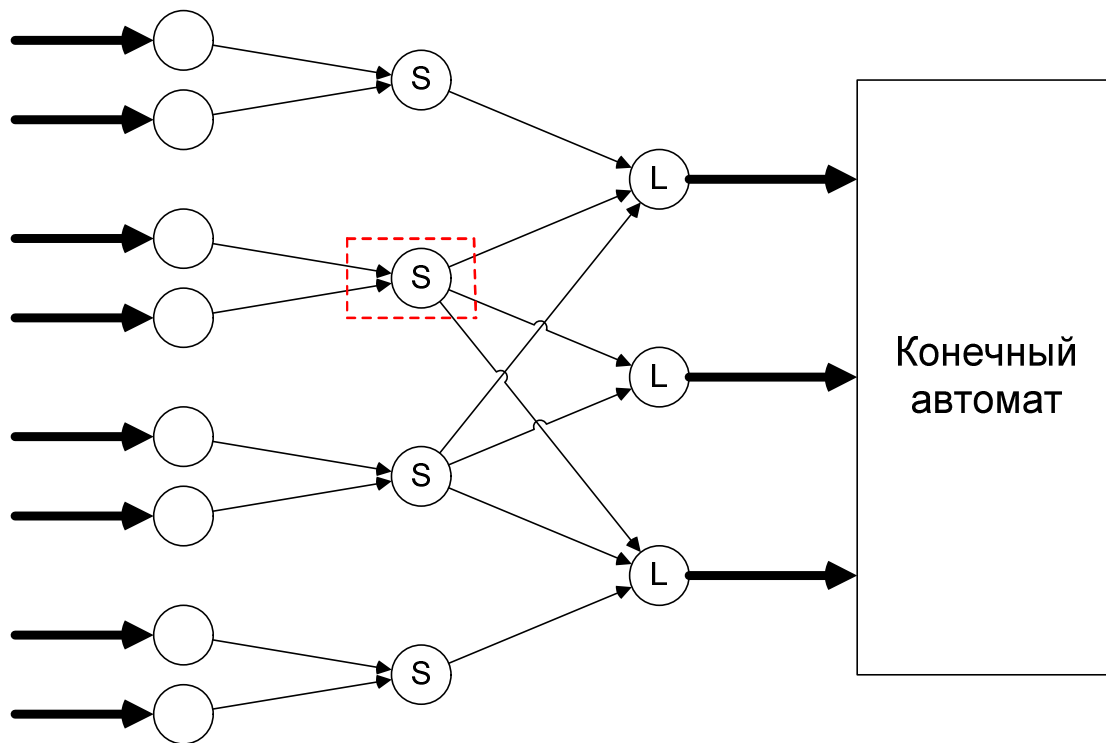


Рис. 17. Мутация нейронных сетей

#### 3.1.4.4. Операция скрещивания

**Оператор скрещивания** получает на вход две особи ( $P1, P2$ ) и выдает две особи ( $S1, S2$ ). Пусть  $X$  – некоторая особь. Пусть  $X.s1$  и  $X.s2$  – системы управления беспилотными летательными аппаратами, входящие в эту особь. Пусть  $s$  – некоторая система управления аппаратом. Обозначим как  $s.ns$  входящую в нее нейронную сеть, а как  $s.a$  – входящий в нее автомат.

**Скрещивание особей.** При скрещивании особей происходит скрещивание систем управления летательными аппаратами:  $P1.s1$  и  $P2.s1$ ,  $P1.s2$  и  $P2.s2$ . Обозначим системы, получившиеся в результате первого скрещивания,  $s11$  и  $s12$ , а в результате второго –  $s21$  и  $s22$ . Тогда для особей-потомков будет справедливо:  $S1.s1 = s11$ ,  $S1.s2 = s21$ ,  $S2.s1 = s21$ ,  $S2.s2 = s22$ .

**Скрещивание систем управления беспилотными летательными аппаратами.** При скрещивании систем управления беспилотными летательными аппаратом  $s1$  и аппаратом  $s2$  происходит скрещивание автоматов  $s1.a$  и  $s2.a$  и скрещивание нейронных сетей  $s1.ns$  и  $s2.ns$ . Обозначим получающиеся в результате описанных скрещиваний автоматы  $a1$  и  $a2$ , а нейронные сети –  $ns1$  и  $ns2$ . В результате скрещивания системы управления летательными аппаратами получают системы управления  $s3$  и  $s4$ , содержащие следующие элементы:  $s3$  содержит  $a1$  и  $ns1$ ,  $s4$  –  $a2$  и  $ns2$ .

**Скрещивание автоматов.** Обозначим автоматы, поступающие на вход оператора скрещивания автоматов,  $A1$  и  $A2$ . Начальное состояние некоторого автомата  $A$  обозначим  $A.is$ , а переход из состояния  $i$  по значению входной переменной  $j$  как  $A(i, j)$ . Обозначим автоматы, получающиеся в результате скрещивания  $A3$  и  $A4$ . Для их начальных состояний справедливо:

- либо  $A3.is = A1.is$  и  $A4.is = A2.is$ ;
- либо  $A3.is = A2.is$  и  $A4.is = A1.is$ .

Опишем переходы автоматов  $A3$  и  $A4$ . Скрещивание производится отдельно для каждого состояния  $i$  и для каждого значения  $j$  входной переменной. В каждом случае возможно два равновероятных варианта:

- $A3(i, j) = A1(i, j)$  и  $A4(i, j) = A2(i, j)$ ;
- $A3(i, j) = A2(i, j)$  и  $A4(i, j) = A1(i, j)$ .

Проиллюстрируем скрещивание автоматов на примере случая одной входной переменной. Обозначим переход из состояния номер  $i$  в автомате  $A1$  по значению входной переменной «1» как  $A1(i, 1)$ , а по значению «0» как  $A1(i, 0)$ . Аналогичный смысл придадим обозначениям  $A2(i, 0)$  и  $A2(i, 1)$ . Тогда для переходов из состояния с номером  $i$  в автоматах-потомках  $A3$  и  $A4$  будет справедливо одно из четырех соотношений:

- либо  $A3(i, 0) = A1(i, 0)$ ,  $A4(i, 1) = A2(i, 1)$  и  $A4(i, 0) = A2(i, 0)$ ,  $A4(i, 1) = A1(i, 1)$ ;
- либо  $A3(i, 0) = A2(i, 0)$ ,  $A4(i, 1) = A1(i, 1)$  и  $A4(i, 0) = A1(i, 0)$ ,  $A4(i, 1) = A2(i, 1)$ ;
- либо  $A3(i, 0) = A1(i, 0)$ ,  $A4(i, 1) = A1(i, 1)$  и  $A4(i, 0) = A2(i, 0)$ ,  $A4(i, 1) = A2(i, 1)$ ;
- либо  $A3(i, 0) = A2(i, 0)$ ,  $A4(i, 1) = A2(i, 1)$  и  $A4(i, 0) = A1(i, 0)$ ,  $A4(i, 1) = A1(i, 1)$ .

Все четыре варианта соотношений равновероятны. Возможные варианты переходов изображены на рис. 18.

В левой части этого рисунка показаны переходы из состояния номер  $i$  автоматов, поступающих на вход операции скрещивания, а в правой части – четыре возможных варианта переходов автоматов, которые будут получены в результате ее применения. Переходы в левой части рисунка пронумерованы числами от одного до четырех. Переходы в правой части также пронумерованы, причем нумерация переходов соответствует нумерации в левой части рисунка.



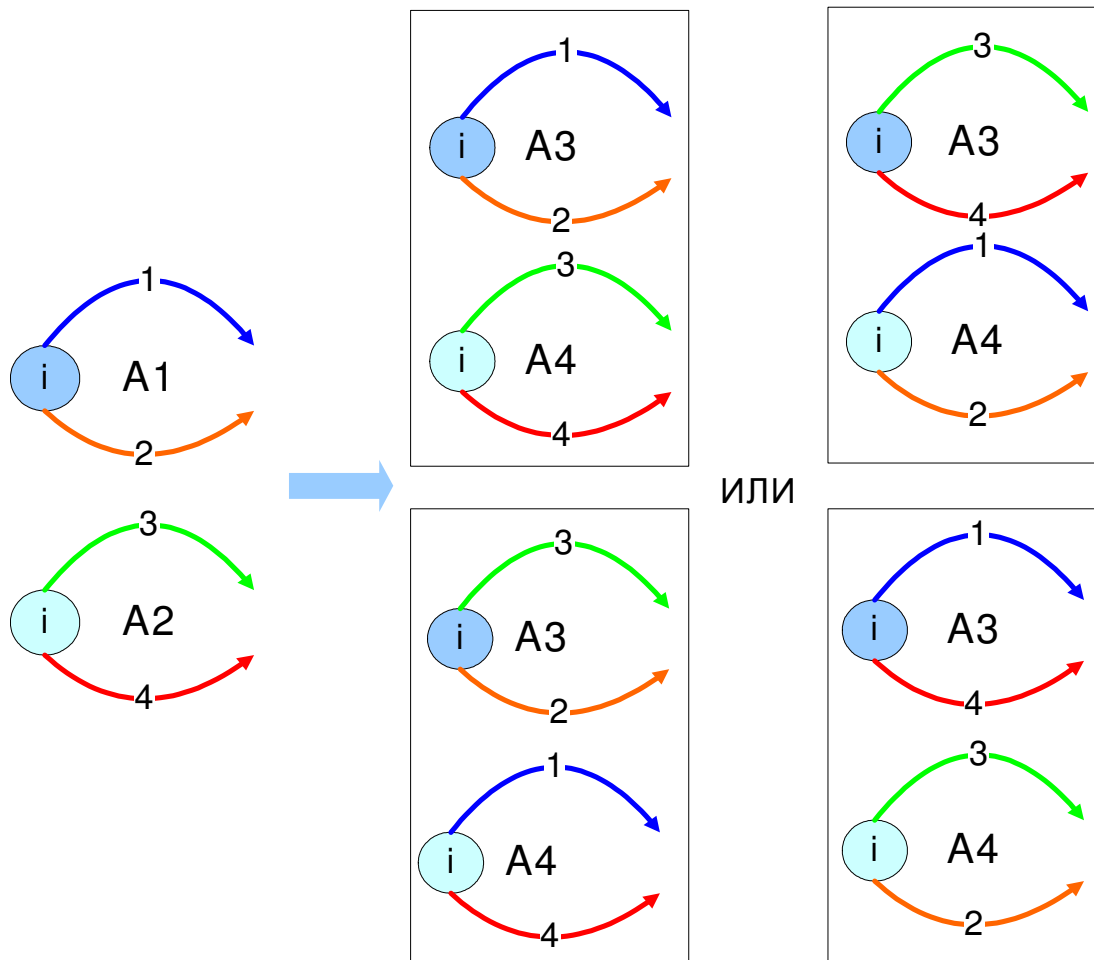


Рис. 18. Варианты переходов при скрещивании

**Скрещивание нейронных сетей.** Обозначим нейронные сети, поступающие на вход оператора скрещивания нейронных сетей,  $NS1$  и  $NS2$ , а получающиеся в результате его применения –  $NS3$  и  $NS4$ . Опишем их устройство. Обозначим  $NS(i)$  нейрон с номером  $i$  сети  $NS$  (рис. 15). Для нейронов с номерами  $NS3(i)$  и  $NS4(i)$  возможны два варианта:

- $NS3(i) = NS1(i)$  и  $NS4(i) = NS2(i)$ ;
- $NS3(i) = NS2(i)$  и  $NS4(i) = NS1(i)$ .

Скрещивание нейронных сетей проиллюстрировано на рис. 19.

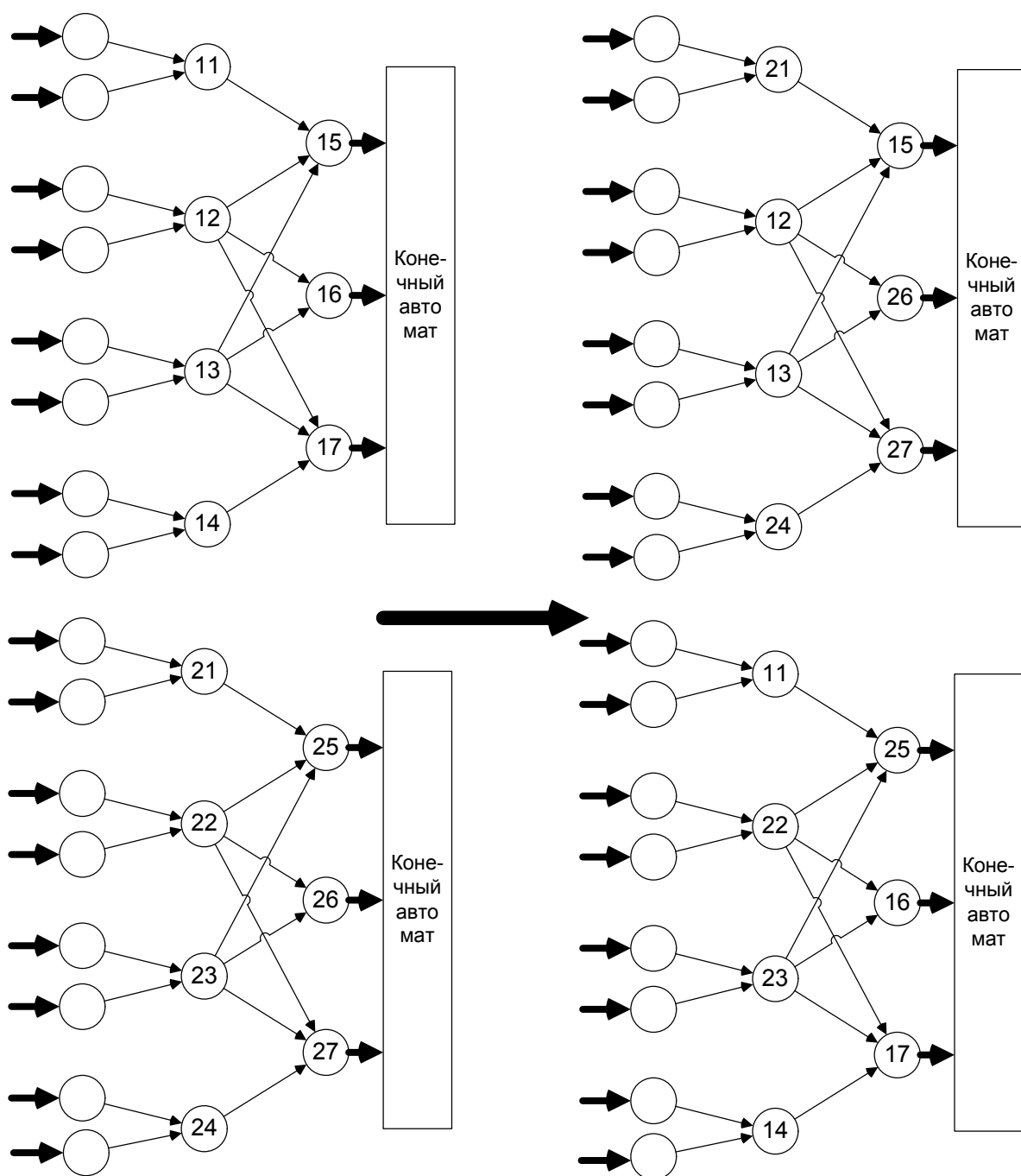


Рис. 19. Скрещивание нейронных сетей

### 3.1.4.5. Формирование следующего поколения

В качестве основной стратегии формирования следующего поколения используется элитизм. При обработке текущего поколения отбрасываются все особи, кроме нескольких наиболее приспособленных. Доля выживающих особей постоянна для каждого поколения и является одним из параметров алгоритма.

Эти особи переходят в следующее поколение. После этого оно дополняется до требуемого размера следующим образом: пока оно не заполнено выбираются две особи из текущего поколения, и

они с некоторой вероятностью скрещиваются или мутируют. Обе особи, полученные в результате мутации или скрещивания, добавляются в новое поколение.

Кроме этого, если на протяжении достаточно большого числа поколений не происходит увеличения приспособленности, то применяются «малая» и «большая» мутации поколения. При «малой» мутации поколения ко всем особям, кроме 10% лучших, применяется оператор мутации. При «большой» мутации каждая особь либо мутирует, либо заменяется на случайно сгенерированную.

Число поколений до «малой» и «большой» мутации постоянно во время работы алгоритма, но может быть различным для разных его запусков.

#### 3.1.4.6. Функция приспособленности

Функция приспособленности особи вычисляется в ходе соревнований команды, аппараты которой управляются описываемыми особью системами управления беспилотным летательным аппаратом, с некоторыми командами, управляемыми системами управления, реализующими выбранную стратегию поведения. В качестве таких систем в настоящей работе используются системы, реализующие «агрессивную» и «простую» стратегию [27].

Проводилось по пять соревнований с каждой из стратегий при следующих параметрах летательных аппаратов (разд. 1):

$$c_1 = 0.625; \quad (10)$$

$$c_2 = 0.025; \quad (11)$$

$$c_4 = 3.125; \quad (12)$$

$$\Delta t = 0.3; \quad (13)$$

$$L = 7. \quad (14)$$

Результатом вычисления функции приспособленности является сумма результатов команды, аппараты которой управляются описываемыми особью системами управления беспилотным летательным аппаратом, во всех соревнованиях, к которой прибавлено число побед, деленное на число соревнований, увеличенное на единицу.

#### 3.1.4.7. Результаты применения метода совместного применения нейронных сетей, генетического программирования и конечных автоматов

Системы управления беспилотным летательным аппаратом строились с помощью описанного алгоритма генетического программирования, а далее тестировались в среде, разработанной в работе [27]. Тестирование проводилось с помощью соревнования построенной системы с командой, аппараты которой управляются системой, описанной в работах [26, 27].

При этом соревнования проводились при числе летательных аппаратов в каждой команде, равном шести или восьми. Для того, чтобы построенные с помощью генетического программирования системы управления беспилотным летательным аппаратом могли работать в этом случае, на первые два входа нейронной сети (рис. 16) подавались относительные координаты ближайшего летательного аппарата из «своей» команды, а на входы с третьего по шестой подавались координаты двух ближайших аппаратов из обеих команд. При этом летательные аппараты с нечетными номерами управлялись системой, построенной для первого аппарата, а с четными номерами – построенной для второго аппарата.

С помощью описанного алгоритма генетического программирования была построена особь, содержащая две системы управления беспилотными летательными аппаратами, каждая из которых

содержит автомат с шестью состояниями. Их построение заняло около суток на компьютере с процессором *Intel Celeron 2.53 GHz*.

Функция переходов и действий автомата, входящего в систему управления первым беспилотным летательным аппаратом, приведена в табл. 6. Строки этой таблицы соответствуют состояниям автомата (пронумерованы числами от 0 до 5), столбцы – возможным комбинациям значений трех входных переменных. Серым цветом отмечена строка, соответствующая начальному состоянию.

Таблица 6. Функция переходов и функция действий автомата, управляющего первым беспилотным летательным аппаратом

	000		001		010		011		100		101		110		111	
0	0		5		3		5		4		2		5		4	
	5	0,8	25	0,4	-5	0,8	15	0,8	5	0,4	-5	0,8	-10	0,4	25	0,8
1	3		2		3		5		4		2		0		5	
	-20	0,4	25	0,4	15	0,8	-15	0,4	5	0,4	20	0,4	-10	0,8	0	0,8
2	2		3		4		5		1		5		2		0	
	20	0,4	-5	0,8	15	0,8	10	0,4	20	0,8	10	0,8	5	0,4	-20	0,4
3	5		3		5		4		5		2		0		4	
	-20	0,4	-20	0,8	10	0,8	-10	0,4	-25	0,8	5	0,4	-5	0,8	-25	0,8
4	0		2		5		2		5		0		1		2	
	0	0,4	5	0,4	-10	0,8	-10	0,44	-10	0,8	10	0,8	-15	0,4	-5	0,8
5	2		1		2		3		1		4		2		2	
	-25	0,78	-15	0,4	-20	0,4	10	0,4	-20	0,8	-5	0,8	0	0,8	-20	0,4

Ячейки имеют формат, показанный в табл. 7, – в верхней строке указано новое состояние, а в нижней – выполняемое действие.

Таблица 7. Формат ячеек табл. 6

Новое состояние	
Изменение угла поворота	Новый расход топлива

В табл. 8 приведены веса нейронной сети, входящей в построенную систему управления первым беспилотным летательным аппаратом.

Таблица 8. Веса связей и смещенные веса нейронов, входящих в нейронную сеть, управляющую первым беспилотным летательным аппаратом

Номер нейрона	Номера нейронов, соединенных с данным	Вес связи	«Смещенный вес»
1	Вход-1	0,7820135561918911	0,0000823910892
	Вход-2	0,9843639098567366	
2	Вход-3	0,8621015893426324	0,0000342189215
	Вход-4	0,49004984673483654	
3	Вход-5	0,31478285017906643	0,0000238490014
	Вход-6	0,9111794712756234	
4	Вход-7	0,9383238994033571	0,0000341289013
	Вход-8	0,5244096391670159	
5	1	-0,9357005374202394	0,4971125943935544
	2	-0,4004385439577506	
	3	0,97490945048913	
6	2	0,9562669142389417	-0,568426590601512
	3	-0,2903272439699007	
7	2	0,1400255181142766	-0,859430022111886
	3	-0,315932752555524	
	4	-0,5112115605370626	

В табл. 9 показаны значения функций переходов и действий для автомата, управляющего вторым беспилотным летательным аппаратом. Обозначения и формат ячеек в этой таблице такие же, как и в табл. 6.

Таблица 9. Функция переходов и функция действий автомата, управляющего вторым беспилотным летательным аппаратом

	000	001	010	011	100	101	110	111
0	2	3	3	4	0	4	0	3
	-15   0,4	5   0,4	10   0,4	-20   0,4	-5   0,4	15   0,4	15   0,4	-25   0,4
1	0	3	3	3	3	5	2	5
	-25   0,8	-15   0,4	25   0,4	-10   0,4	-25   0,8	-5   0,4	-15   0,8	-10   0,4
2	4	0	0	0	2	0	3	2
	-15   0,8	5   0,8	25   0,4	15   0,4	-25   0,4	-5   0,4	0   0,4	10   0,8
3	3	3	2	3	3	5	2	1
	0   0,4	20   0,8	20   0,8	-25   0,8	-15   0,8	20   0,4	5   0,4	15   0,4
4	4	4	5	0	4	2	3	2
	5   0,8	-20   0,4	25   0,8	25   0,4	0   0,38	-15   0,4	0   0,8	-15   0,36
5	3	4	0	0	4	4	4	2
	-15   0,4	-20   0,4	0   0,8	10   0,4	-10   0,8	10   0,35	0   0,4	0   0,8

В табл. 10 приведены веса связей нейронной сети, входящей в построенную систему управления вторым беспилотным летательным аппаратом.

Таблица 10. Веса связей и смещенные веса нейронов, входящих в нейронную сеть, управляющую вторым беспилотным летательным аппаратом

Номер нейрона	Номера нейронов, соединенных с данным	Вес связи	«Смещенный вес»
1	Вход-1	0,23234941730420006	0,0003223789
	Вход-2	0,07485700920671275	
2	Вход-3	0,3056025308692568	0,0030210123478
	Вход-4	0,30661997660895013	
3	Вход-5	0,17219120661378778	0,0000728973223
	Вход-6	0,03018455303178147	
4	Вход-7	0,23636134465917025	0,00430412789122
	Вход-8	0,806175323870498	
5	1	0,9180350210858039	-0,8415486736799431
	2	-0,4638322712743177	
	3	-0,5253829213469569	
6	2	-0,8334567897770782	0,05661834053390668
	3	-0,7931512598511974	
7	2	0,3096107059232607	0,2372057991691212
	3	0,13115163005946284	
	4	-1,0	

### 3.2. СОВМЕСТНОЕ ПРИМЕНЕНИЕ МЕТОДА СОКРАЩЕННЫХ ТАБЛИЦ ПЕРЕХОДОВ И АЛГОРИТМА ВОССТАНОВЛЕНИЯ СВЯЗЕЙ МЕЖДУ СОСТОЯНИЯМИ

В этом разделе описывается совместное применение метода сокращенных таблиц переходов [28] и алгоритма восстановления связей между состояниями, описанных в отчете по II этапу научно-исследовательских работ по настоящему государственному контракту [29].

#### 3.2.1. Структура системы управления беспилотным летательным аппаратом

В настоящем разделе все беспилотные летательные аппараты управляются одинаковыми системами управления, каждая из которых представляет собой конечный автомат. Таким образом, как отмечалось выше, можно говорить, что используется *мультиагентный подход* [30] – каждый беспилотный летательный аппарат представляет собой агента, взаимодействующего с внешней средой и другими агентами.

Отметим, что при такой структурной схеме в отличие от схемы, представленной на рис. 15, программирование входных логических воздействий осуществляется вручную, а не автоматически с помощью искусственной нейронной сети.

В течение соревнования каждый летательный аппарат испытывает на себе воздействие среды и других летательных аппаратов.



Рис. 20. Структурная схема системы управления беспилотным летательным аппаратом

Для описания этих воздействий служат ниже перечисленные логические входные переменные:

- граница трассы справа;
- граница трассы слева;
- другой аппарат слева;
- другой аппарат справа;
- другой аппарат спереди;
- другой аппарат сзади.

Эти переменные подаются на вход управляющего автомата, который, в свою очередь, формирует последовательность выходных воздействий. Приведем список выходных воздействий:

- увеличить расход топлива на фиксированную величину;
- уменьшить расход топлива на фиксированную величину;
- сделать расход топлива максимально возможным;
- изменить направление аэродинамического руля на фиксированный угол налево;
- изменить направление аэродинамического руля на фиксированный угол направо;
- лететь прямо.

### 3.2.2. Представление особи в генетическом алгоритме

Каждая особь хранит следующие параметры:

- массив состояний;
- число состояний;
- число обрабатываемых входных переменных;
- число возможных действий.

Каждое состояние хранит сокращенную таблицу переходов:

- массив значимых входных переменных;
- массив состояний для всех переходов;
- массив выполняемых действий для всех переходов.

На рис. 21 приведен пример сокращенной таблицы для одного из состояний автомата. В этой таблице столбец  $S$  – массив состояний,  $Z1-Z4$  массивы действий,  $Z1[i]$  равно единице, если данное

действие выполняется. Массив `variables` указывает, какие переменные значимы, если переменной соответствует единица, то переменная значима.

S	Z1	Z2	Z3	Z4
1	0	0	1	0
3	1	0	1	1
5	1	0	1	0
1	1	0	1	1

variables	A	B	C	D	E	F	G
	0	0	1	0	0	1	0

Рис. 21. Пример сокращенной таблицы переходов

*Значимыми* называются входные переменные, значение которых обрабатываются при нахождении в данном состоянии (значения других переменных игнорируются).

Когда на вход поданы переменные, определяется строка, соответствующая данным значениям, по следующему правилу:

- в битовую строку записываются значения всех значимых входных переменных в том порядке, в котором они входят в массив `variables`;
- полученная битовая строка переводится в десятичное число, к которому необходимо прибавить единицу. Результатом будет искомый номер строки (рис. 22).

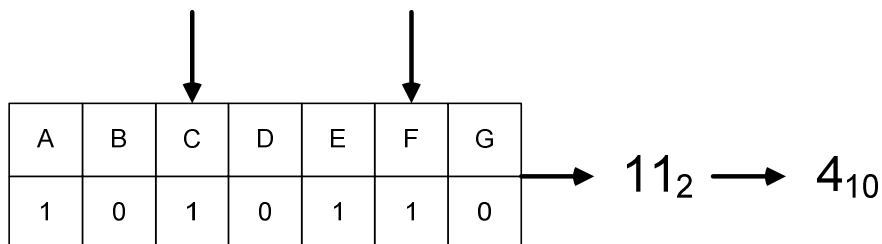


Рис. 22. Подсчет номера строки

На языке *Java* особь в этом случае имеет следующее представление:

```
public class TableAutomaton {
    private int numberVariables;
    private int numberActions;
```



```

private int numberStates;
private int initialState;
private State[] state;

public class State {

    private int[] endState;
    private boolean[][] transitionTable;
    private boolean[] variable;

}
}

```

### 3.2.3. Восстановление связей между состояниями

В данном разделе описан алгоритм восстановления связей между состояниями автомата, который обеспечивает достижение большего числа состояний автомата из начального за счет изменения конечного состояния некоторых переходов. Для поиска достижимых состояний используется алгоритм *поиска в ширину (Breadth First Search)*. Данный алгоритм описан ниже на псевдокоде.

```

TableAutomaton repairedAutomaton() {
    for (повторить N раз) {
        Запустить bfs от стартового состояния. Массив пометок mark
        содержит информацию о том, было ли посещено состояние.
        for (для всех i: состояний автомата) {
            if (!mark[i]) {
                State st = случайное достижимое состояние;
                Установить случайный переход из st в состояние i;
            }
        }
        if (не изменилось ни одно из состояний) {
            выйти из цикла;
        }
    }
    return this;
}

```

Данная процедура не гарантирует достижимость всех состояний из начального состояния для графа переходов, а лишь восстанавливает ее с вероятностью, возрастающей с константой N. В данной работе используется N, равное десяти.

Рис. 23 демонстрирует работу данного алгоритма, серым отмечены состояния, достижимые из стартового.

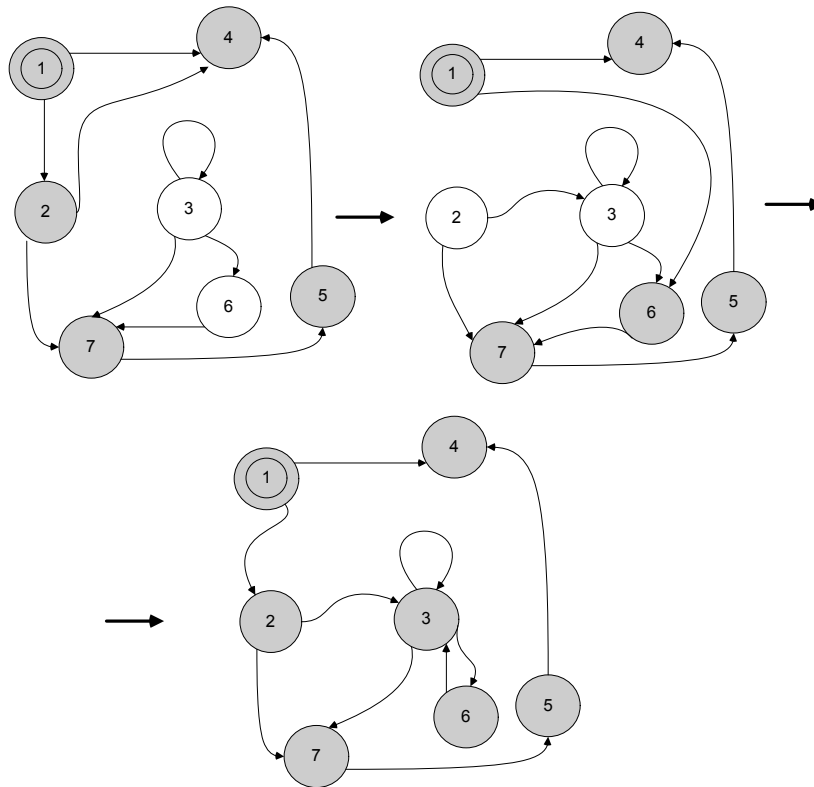


Рис. 23. Пример работы алгоритма восстановления связей между состояниями

### 3.2.4. Оператор мутации

При мутации особи выполняются следующие действия:

- с вероятностью 0.5 случайное изменение начального состояния;
- мутация случайного состояния.

Мутация состояния для случая представления особи в виде сокращенных таблиц осуществляется следующим образом:

для каждой строки таблицы переходов выполнить:

- с некоторой заранее заданной вероятностью  $p_1$  изменить переход из данного состояния по текущим значениям входных переменных на случайное значение (из допустимых);
- с некоторой заранее заданной вероятностью  $p_2$  изменить все действия на данном переходе.

Каждое действие выполняется на данном переходе с вероятностью  $\frac{n_1}{k}$ , где  $n_1$  – число вы-

полняемых действий на данном переходе до начала мутации,  $k$  – число возможных действий.

Выбор *значимых переменных* состояния. Этот выбор осуществляется следующим образом:

- выбрать случайным образом две переменные;
- если одна из них значима в данном состоянии, а другая нет, то они меняются местами. Иначе ничего не изменяется.

На рис. 24 приведен пример мутации. Серым цветом отмечены получившиеся части состояния.

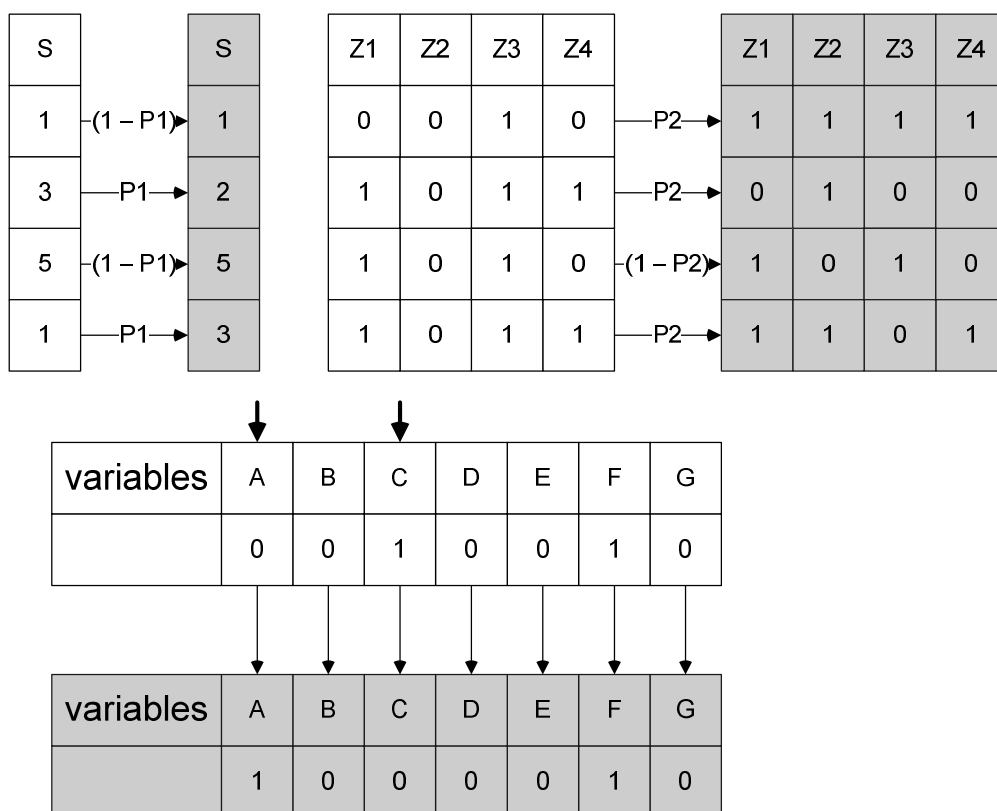


Рис. 24. Мутация сокращенных таблиц

По окончании работы оператора мутации для получившейся особи запускается алгоритм восстановления связей между состояниями. Это необходимо потому, что в результате мутации некоторые состояния могут стать недостижимыми.

### 3.2.5. Оператор скрещивания

Оператор скрещивания особей представляется описанным ниже образом.

Обозначим родительские особи –  $P1$  и  $P2$ , а детей –  $S1$  и  $S2$ . Обозначим  $k$ -ое состояние автомата  $A$  как  $A.a[k]$ . Как  $c1[k]$  и  $c2[k]$  обозначим результат скрещивания состояний  $P1.a[k]$  и  $P2.a[k]$ . Тогда для любого  $k$  будет верно утверждение  $S1.a[k] = c1[k]$ ,  $S2.a[k] = c2[k]$ .

Алгоритм скрещивания состояний представляет собой *одноточечное* скрещивание соответствующих столбцов таблицы переходов. Для каждого столбца выполняются следующие действия:

- случайный выбор границы разделения столбца;
- соответствующий столбец состояния  $c1$  составляется из первой части столбца состояния  $P1.a[k]$  и второй части столбца особи  $P2.a[k]$ ;
- соответствующий столбец состояния  $c2$  составляется из второй части столбца особи  $P1.a[k]$  и первой части столбца особи  $P2.a[k]$ .

Данный алгоритм проиллюстрирован на рис. 25.

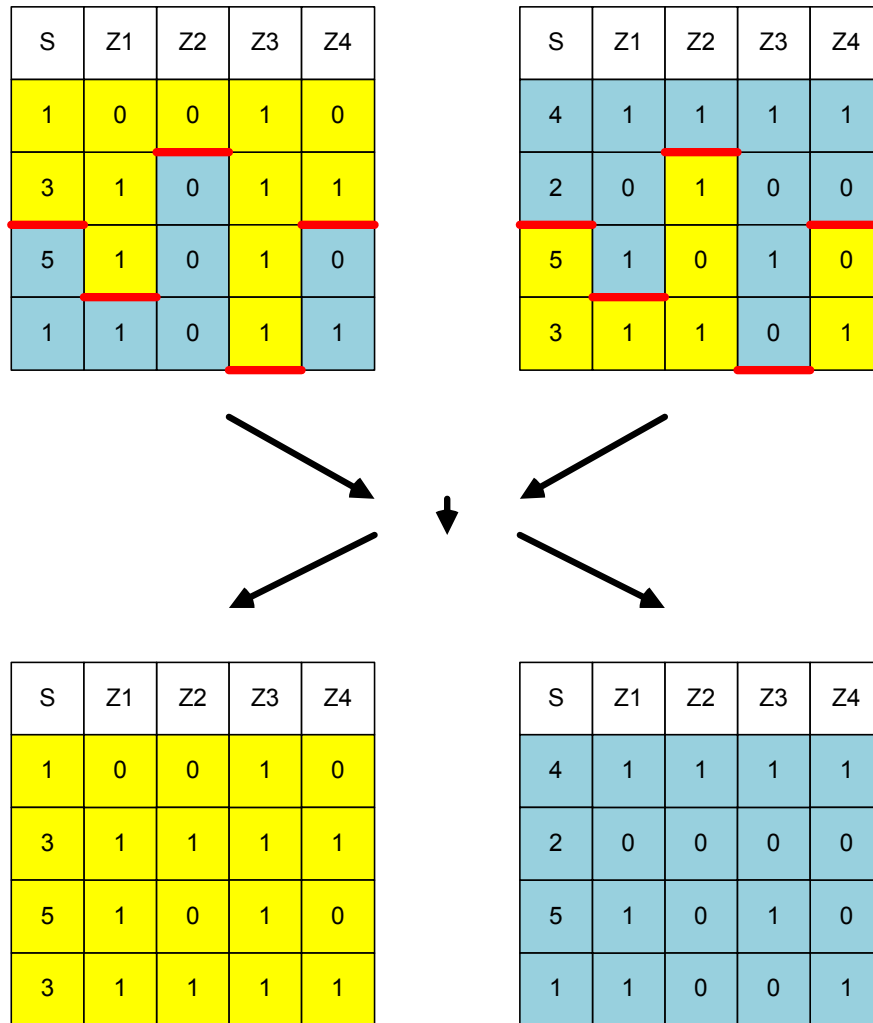


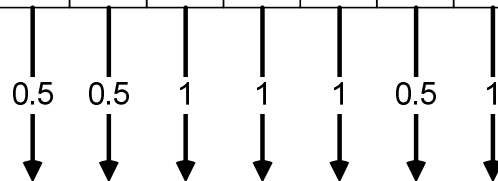
Рис. 25. Скрещивание сокращенных таблиц

Для того чтобы выбрать значимые переменные (рис. 26)  $c1[k]$  и  $c2[k]$ , выполняется просмотр списка значимых переменных родителей. Обозначим за  $v[k]$  –  $k$ -ый элемент в списке значимых переменных. При этом справедливо одно из следующих утверждений:

- $P1.v[k] = 0, P2.v[k] = 0 \Rightarrow S1.v[k] = 0, S2.v[k] = 0$ ;
- $P1.v[k] = 1, P2.v[k] = 1 \Rightarrow S1.v[k] = 1, S2.v[k] = 1$ ;
- $P1.v[k] = 0, P2.v[k] = 1$  или  $P1.v[k] = 1, P2.v[k] = 0 \Rightarrow$  возможен один из трех случаев:
  - 1) списки значимых переменных обоих детей еще не заполнены (число значимых переменных в обоих списках не превышает число значимых переменных у родителей), тогда с вероятностью 0.5 выполняются соотношения  $P1.v[k] = 0, P2.v[k] = 1$ , иначе  $P1.v[k] = 1, P2.v[k] = 0$ ;
  - 2) списки значимых переменных первого ребенка заполнены, тогда  $P1.v[k] = 0, P2.v[k] = 1$ ;
  - 3) списки значимых переменных второго ребенка заполнены, тогда  $P1.v[k] = 0, P2.v[k] = 1$ ;

variables	A	B	C	D	E	F	G
	1	0	1	0	1	1	0

variables	A	B	C	D	E	F	G
	0	1	1	0	1	0	1



variables	A	B	C	D	E	F	G
	1	0	1	0	1	1	0

variables	A	B	C	D	E	F	G
	0	1	1	0	1	0	1

Рис. 26. Выбор значимых переменных детей

Отметим, что списки значимых переменных обоих потомков могут быть заполнены одновременно тогда, и только тогда, когда в списках обоих родителей не осталось не просмотренных значимых переменных.

### 3.2.6. Особенности применения сокращенных таблиц

Опишем преимущества метода сокращенных таблиц переходов применительно к рассматриваемой задаче:

- несовместность некоторых входных переменных (граница трассы слева, граница трассы справа), делает неиспользуемыми некоторые состояния. При использовании сокращенных таблиц вероятность такого состояния, по сравнению с полными таблицами значительно снижается;
- сокращение используемой памяти, а также некоторое ускорение работы алгоритма.

При применении сокращенных таблиц все значимые переменные имеют равный приоритет, так как решение принимается сразу на основе всех переменных, которые используются в данном состоянии (альтернативой являются деревья решений, рассматриваемые далее, в которых решение принимается поэтапно).

### 3.2.7. Генетический алгоритм

В данной работе для генерации системы управления беспилотным летательным аппаратом используется *островной генетический алгоритм*. Общая схема (рис. 27) данного алгоритма заключается в том, что существует несколько популяций – *островов*. Большую часть времени развитие популяций на каждом острове происходит независимо.

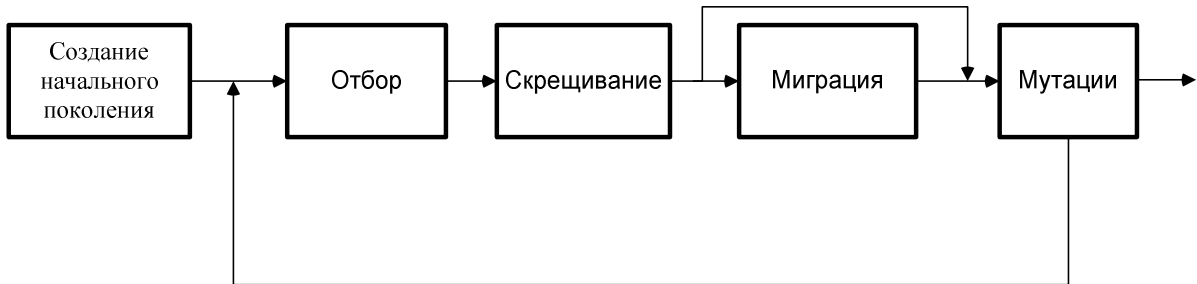


Рис. 27. Схема островного генетического алгоритма

Через заданное число поколений происходит *миграция* – часть особей с каждого острова передается другому острову.

#### 3.2.7.1. Создание начального поколения

Все острова заполняются случайно сгенерированными особями. Все особи имеют заранее заданное число состояний. При применении сокращенных таблиц для хромосом всех особей задается число значимых переменных, которое одинаково для всех состояний.

#### 3.2.7.2. Формирование следующего поколения

В качестве основной стратегии формирования следующего поколения используется элитизм. При рассмотрении текущего поколения отбрасываются все особи, кроме некоторой доли наиболее приспособленных – «элиты». Эти особи переходят в следующее поколение. После этого оно дополняется в определенной пропорции случайными особями, особями из текущего поколения, которые мутировали, и результатами скрещивания особей из текущего поколения (отдельно отметим, что скрещиваться могут не только элитные особи, а все). Особи, «имеющие право» давать потомство, определяются «в поединке»: выбираются две случайные пары особей, и более приспособленная особь в каждой из них становится одним из родителей.

#### 3.2.7.3. Миграции

Через фиксированное число поколений каждый остров меняется с другим случайным числом случайно выбранных элитных особей.

Также через некоторое число поколений происходит *большая мутация*, называемая также *мутацией острова*.

#### 3.2.7.4. Вычисление функции приспособленности

Функция приспособленности особи должна зависеть от результата команды беспилотных летательных аппаратов, использующей стратегию, описываемую этим автоматом. Для того, чтобы задать вид зависимости, требуется определить условия соревнования: начальные координаты летательных аппаратов и команду соперника. В этом-то и состоит проблема, так как, если при каждом запуске автомата начальные координаты и соперника выбирать случайным образом, то функция приспособленности будет «необъективна». Противоположный вариант – зафиксировать соперника и

начальные координаты (например, считать их нулевыми). В качестве соперника выбрана команда с «агрессивной» стратегией. Эта стратегия, как отмечалось в разделе 2, заключается в том, что один летательный аппарат летит только вперед с постоянным оптимальным расходом топлива. Оптимальный расход топлива – такой расход топлива, при котором достигается наибольшая дальность полета. Остальные аппараты пытаются сбить летательные аппараты соперника. Однако даже в этом случае функция приспособленности не является некоторой абсолютной величиной, из-за недетерминированности агрессивной стратегии.

Для решения этих проблем предлагается вычислять функцию приспособленности как среднее арифметическое результатов нескольких соревнований. При этом начальные координаты – случайные, но одинаковые для всех команд реализующих стратегию, заданную генерируемыми особями. Число соревнований выбиралось равным трем из соображений экономии времени работы алгоритма.

При этом функция приспособленности имеет вид (15):

$$F = \frac{\sum_{i=0}^k r_i}{k}, \quad (15)$$

где  $k$  – число соревнований, а  $r_i$  – результат команды в  $i$ -ом соревновании.

В качестве соперников выбиралась команда с агрессивной стратегией и команда со стратегией заданной, автоматом, граф переходов которого показан на рис. 28. Этот автомат был построен с помощью описываемого генетического алгоритма, но при вычислении функции приспособленности только команда с «агрессивной» стратегией.

Опишем обозначения, используемые в этом рисунке графе. Двойной окружностью на этом рисунке отмечено стартовое состояние. Пометки на вершинах имеют следующий вид: *номер состояния*/*массив значимых переменных*. Переменные перечисляются в следующем порядке:

1. Граница трассы справа.
2. Граница трассы слева.
3. Другой аппарат слева.
4. Другой аппарат справа.
5. Другой аппарат спереди.
6. Другой аппарат сзади.

Пометки на переходах имеют следующий вид *{значения входных переменных}*/*вектор действий*. Действия перечисляются в следующем порядке:

1. Установить нормальный расход топлива.
2. Изменить направление аэродинамического руля на фиксированный угол налево.
3. Изменить направление аэродинамического руля на фиксированный угол направо.
4. Лететь прямо.
5. Сделать расход топлива максимально возможным.
6. Увеличить расход топлива на фиксированную величину.
7. Уменьшить расход топлива на фиксированную величину.

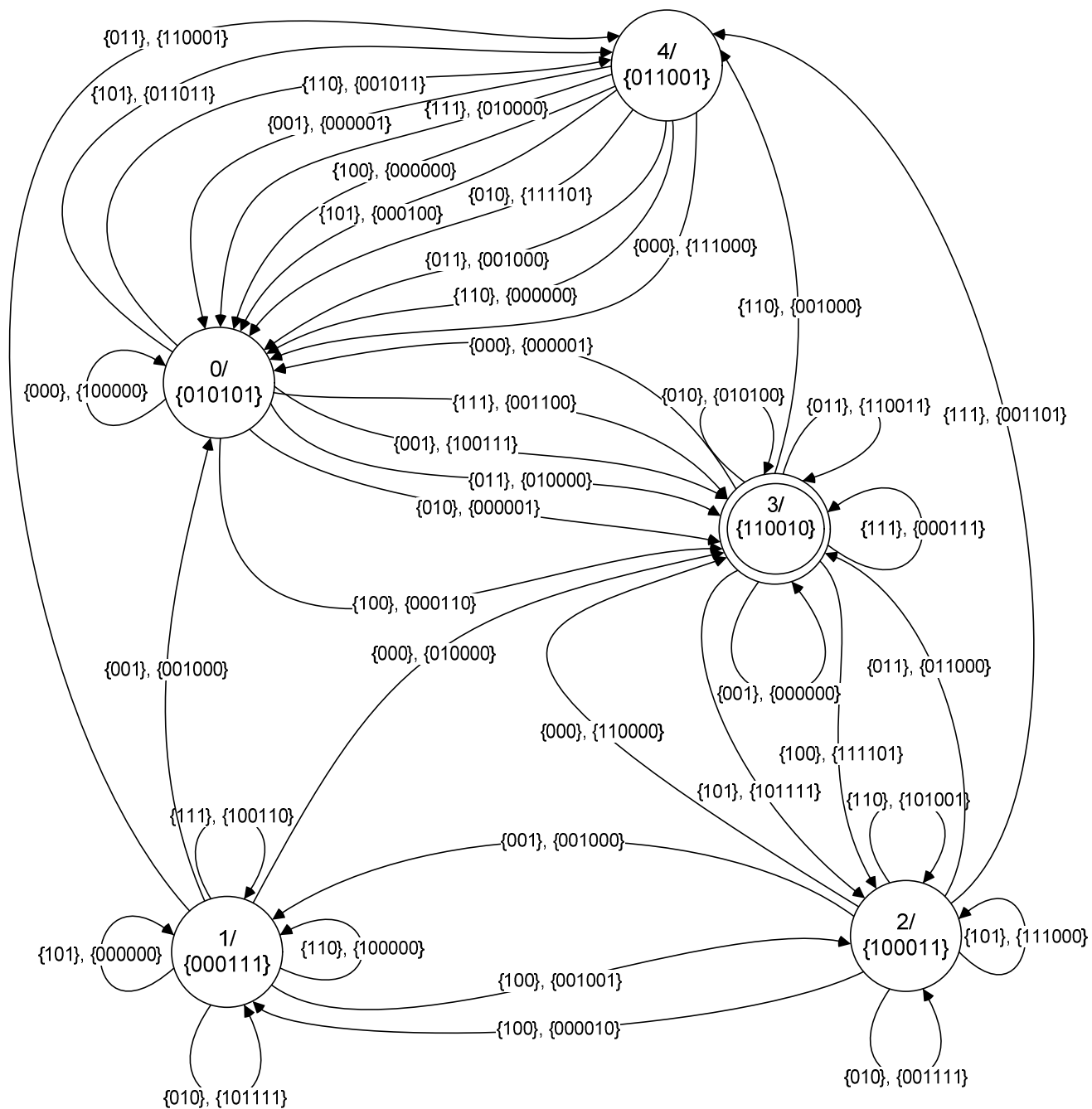


Рис. 28. Управляющий автомат, используемый при вычислении функции приспособленности

Этот автомат был выбран по двум причинам. Во-первых, он «продемонстрировал свою силу» в соревнованиях с «агрессивной» командой и командой, реализующей стратегию, заданную системой автоматов, построенных без применения методов автоматизации (описана в разд. 2). Из двадцати соревнований с первой из них построенный с помощью генетического алгоритма автомат выиграл восемнадцать, со второй – девятнадцать. Во-вторых, в отличие от этих двух команд он реализует детерминированную стратегию. И, наконец, этот автомат реализует «дружелюбную» стратегию, которая заключается в том, что летательные аппараты помогают лететь друг другу, а не пытаются



сбить аппараты команды соперника. Это позволяет «тренировать» новые управляющие автоматы сразу на двух принципиально различных стратегиях: «агрессивной» и «дружелюбной».

Проблемой является принципиальная невозможность оценить полученную особь по абсолютной шкале, как уже упоминалось ранее. Это происходит из-за недетерминированности начальных параметров задачи, а также противников. Этот эффект можно наблюдать на рис. 29, где изображены графики функции приспособленности особи, представляющей собой систему управления беспилотными летательными аппаратами от индекса поколения.

Светлым цветом на этих графиках изображена средняя дальность летательного аппарата при проведении 10 соревнований (против трех при вычислении функции приспособленности), а черным – функция приспособленности.

Число поколений работы генетического алгоритма ограничено лишь теми соображениями, что генерируемые на больших поколениях особи не демонстрируют универсального поведения, а приспосабливаются к соперникам и начальным условиям.

Также существует теоретический максимум дальности полета, определяемый исходя из условий задачи о беспилотных летательных аппаратах.

Однако если задан конкретный противник, то вырастить автомат, который, по крайней мере, играет не хуже его, не составляет особого труда. Естественно, результат соревнования все равно будет зависеть от начальных условий, но эта зависимость будет не слишком значимой. Выше (рис. 28) уже приводился пример автомата, который побеждает «агрессивную» стратегию с соотношением 9:1.



Рис. 29. Эволюция.

Зеленый цвет – средняя дальность, черный – функция приспособленности

### 3.2.7.5. Большие мутации

Для того чтобы избежать «вырождения» автоматов (попадания в локальный максимум функции приспособленности), предлагаемый генетический алгоритм использует описанную ниже процедуру.

Через заранее заданное число поколений фиксированная доля островов заменяется островами со случайными особями. Это является причиной провала на 41-ом поколении (рис. 30).

Проведение мутации в момент, когда функция приспособленности (имеются в виду только элитные особи) изменяется незначительно, невозможно, так как за счет миграции особей между островами среднее значение функции приспособленности постоянно изменяет свое значение.

### 3.2.7.6. Особенности применения островного генетического алгоритма

Островной генетический алгоритм имеет следующие достоинства (перед алгоритмами, где поколение развивается как единое целое):

- более быстрая сходимость к максимуму за счет миграций;
- возможность выхода из локального максимума без потери результатов предыдущих вычислений за счет «изолированности» островов;

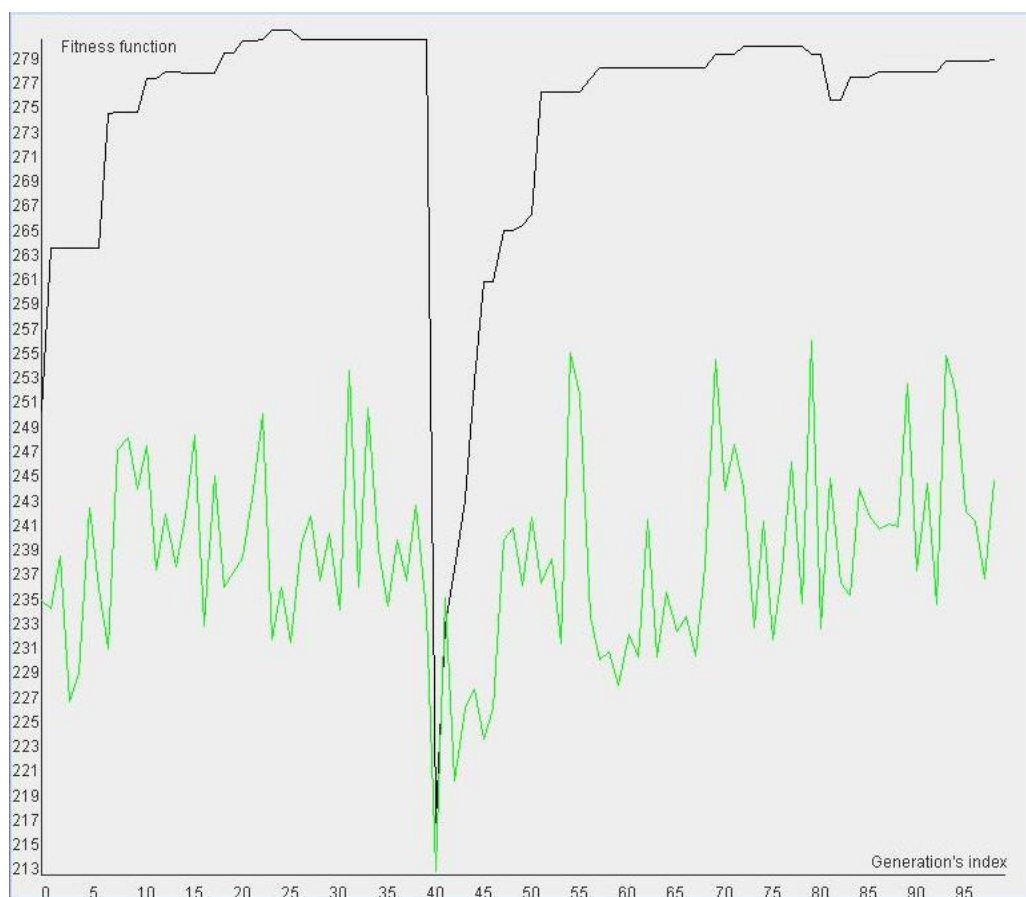


Рис. 30. Большая мутация.

Зеленый цвет – средняя дальность, черный – функция приспособленности

К недостаткам островного алгоритма можно отнести более частое попадание в локальные максимумы по сравнению с традиционным генетическим алгоритмом.

### 3.2.8. Результат применения метода представления автоматов с помощью сокращенных таблиц переходов

При помощи алгоритма генетического программирования с использованием метода представления автоматов с помощью сокращенных таблиц переходов построен конечный автомат управления моделью беспилотного летательного аппарата. Его граф переходов приведен на рис. 31. Построение этого автомата заняло 62 поколения при следующих значениях параметров алгоритма:

- число островов – 10;
- размер острова – 100 особей;
- процент «элиты» – 10%;
- вероятность мутации – 10%;
- процент особей, участвующих в миграции между островами – 3%;
- число поколений, проходящее между миграциями особей между островами – 10;
- число поколений, проходящее между «большими» мутациями – 41;
- процент островов, на которых все особи заменяются случайно сгенерированными в процессе «большой» мутации – 85%.

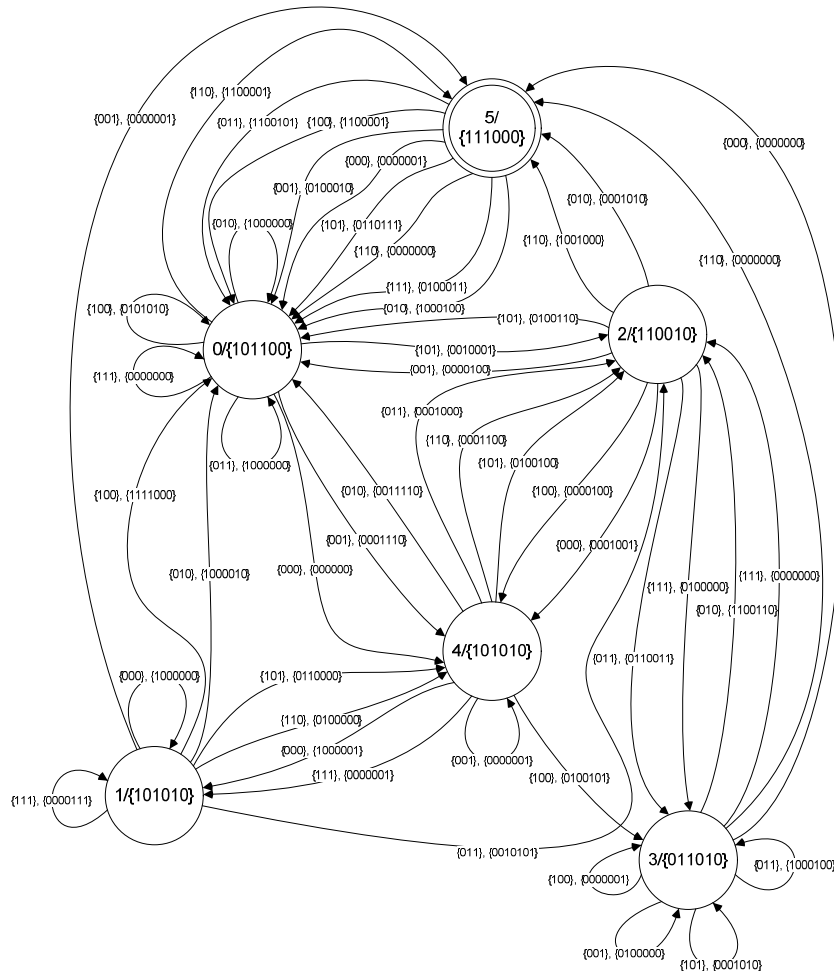


Рис. 31. Управляющий автомат, построенный алгоритмом генетического программирования

Двойной окружностью на этом рисунке показано начальное состояние автомата.

На рис. 31 используются указанные ниже обозначения.

Пометки на вершинах имеют следующий вид *номер/массив значимых переменных*.

Переменные перечисляются в следующем порядке:

1. Граница трассы справа.
2. Граница трассы слева.
3. Другой аппарат слева.
4. Другой аппарат справа.
5. Другой аппарат спереди.
6. Другой аппарат сзади.

Пометки на переходах имеют следующий вид *{значения переменных}/{вектор действий}*.

Действия перечисляются в следующем порядке:

1. Установить нормальный расход топлива.
2. Изменить направление аэродинамического руля на фиксированный угол налево.
3. Изменить направление аэродинамического руля на фиксированный угол направо.
4. Лететь прямо.
5. Сделать расход топлива максимально возможным.
6. Увеличить расход топлива на фиксированную величину.
7. Уменьшить расход топлива на фиксированную величину.

### 3.3. ПРИМЕНЕНИЕ МЕТОДА ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ДЕРЕВЬЯМИ РЕШЕНИЙ

В настоящем разделе описывается применение метода представления автоматов деревьями решений [20] для построения управляющих автоматов в задаче управления беспилотными летательными аппаратами.

#### 3.3.1. Представление особи в генетическом алгоритме

При использовании метода представления автоматов деревьями решений каждая особь хранит следующие параметры:

- массив состояний;
- «рекомендуемая» высота дерева.

Каждое состояние представляет собой дерево решений для функции вида (16):

$$f : \{0,1\}^n \rightarrow N \times \{0,1\}^k, \quad (16)$$

где  $n$  – число входных переменных,  $k$  – число возможных действий беспилотного летательного аппарата. Данное дерево по значениям входных переменных, выдает номер состояния, в которое необходимо перейти автомату, а также вектор из нулей и единиц. При этом если  $i$ -ая компонента этого вектора равна единице, то аппарату необходимо выполнить  $i$ -ое действие. Далее будем называть этот вектор *вектором действий*.

На рис. 32 приведен пример дерева решений для функции от двух булевых переменных ( $A, B$ ).

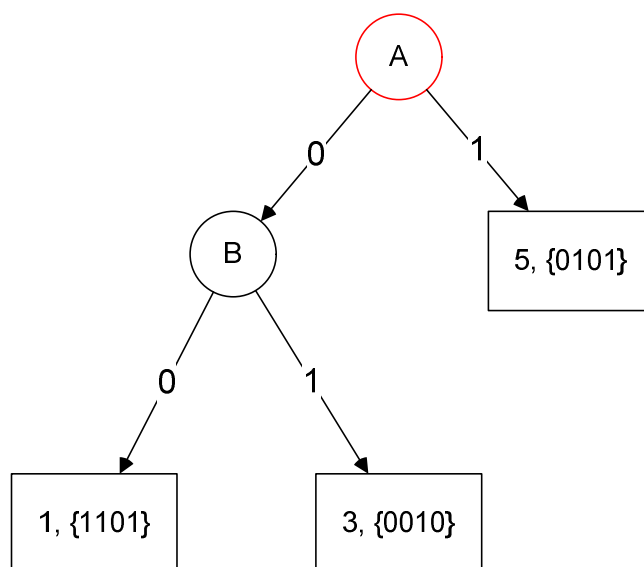


Рис. 32. Пример дерева решений

Каждое состояние представляется деревом решений, которое состоит из *узлов*. Среди узлов выделяется *корень* – стартовый узел дерева решений (на рис. 32 изображен сверху).

Каждый узел дерева решений состоит из следующих частей:

- указателя на левого ребенка (для листьев этот указатель пуст);
- указателя на правого ребенка (для листьев этот указатель пуст);
- переменной, по которой происходит расщепление в данном узле (для листьев это значение не используется);
- номера состояния автомата, в которое ведет переход из данной вершины (используется только в листьях);
- вектор действий (для внутренних вершин этот вектор пуст).

Левый ребенок внутреннего узла соответствует нулевому значению переменной расщепления, а правый – единичному.

Таким образом, особь представляет собой объект на языке *Java* следующего вида:

```

public class TreeAutomaton extends Automaton{

    private Tree[] state;
    private int initialState;
    private int height;

}

public class Tree{

    private TreeNode start;

    public static class TreeNode{

        private TreeNode left;
        private TreeNode right;
    }
  
```

```

        private int endState;
        private boolean[] actions;
        private int variable;
    }
}

```

### 3.3.2. Генерация случайной особи

Для того, чтобы сгенерировать внутреннюю вершину дерева применяется следующий алгоритм:

- выбрать переменную, по которой в данной вершине дерева будет проводиться расщепление;
- сгенерировать левого ребенка;
- сгенерировать правого ребенка.

Генерация листа происходит следующим образом:

- случайным образом выбрать номер состояния, в которое будет вести переход из данного листа;
- случайным образом сгенерировать вектор действий.

Для генерации случайного дерева решений применяется следующий алгоритм: с вероятностью 0.5 генерируется лист или внутренний узел.

Данный алгоритм может продолжать свою работу бесконечно долго. Для того, чтобы этого избежать введено ограничение на высоту дерева. Если при генерации высота дерева начинает превышать удвоенное число возможных входных переменных, то обязательно генерируется лист.

Для того, чтобы сгенерировать особь выполняются следующие действия:

- сгенерировать необходимое число случайных состояний (деревьев решений);
- выбрать случайным образом начальное состояние автомата.

### 3.3.3. Алгоритм обрезки недостижимых ветвей дерева

Если по пути из корня до некоторого узла переменная, по которой происходит расщепление в этом узле, встречается дважды, то ветвь, соответствующая значению противоположному тому, которое было выбрано при первом расщеплении, будет *недостижимой*. Рис. 33 поясняет это утверждение.

Светло-серым цветом на рис. 33 отмечены вершины с повторяющейся переменной расщепления, а темно-серым – недостижимая ветвь.

Для обрезки недостижимых ветвей используется модификация алгоритма *поиска в глубину (Depth First Search)*. При рекурсивном спуске запоминаются переменные, по которым уже проводилось расщепление на пути из корня в текущий узел, а также значения этих переменных, соответствующие этому пути. Если переменная встречается второй раз, то текущий узел, заменяется на корень достижимого поддерева этого узла. Решение о том, какое из поддеревьев достижимо, принимается на основании запомненных значений переменных.

Работу данного алгоритма иллюстрирует рис. 34.

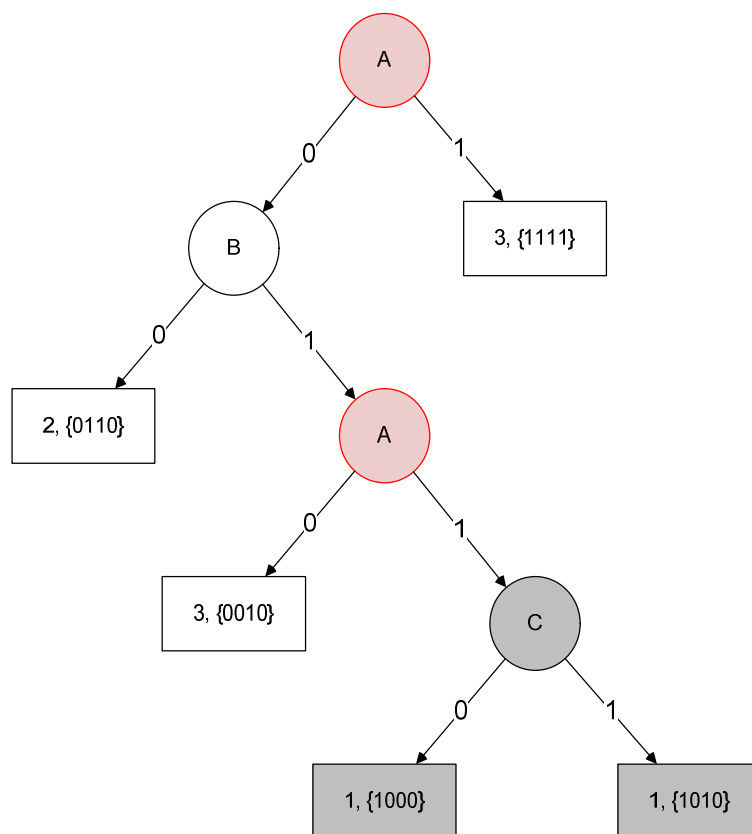


Рис. 33. Недостижимые ветви дерева решений

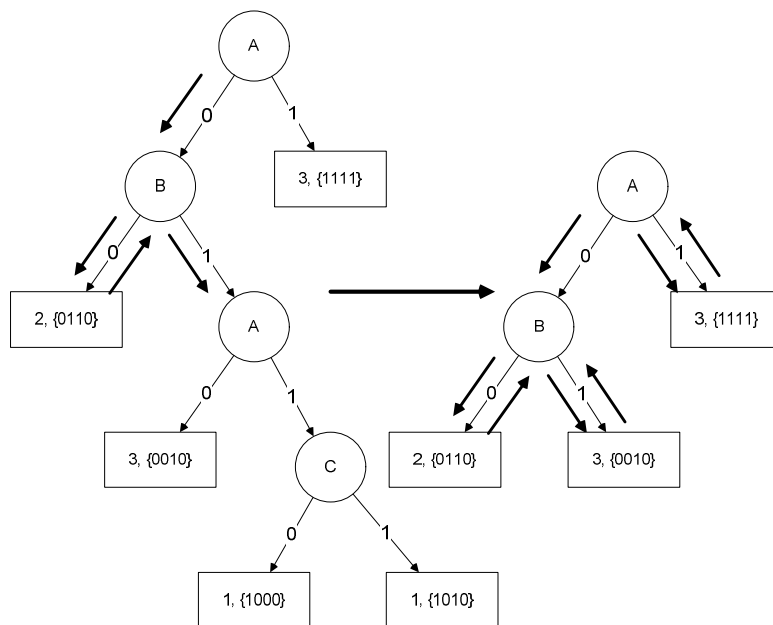


Рис. 34. Удаление недостижимых ветвей

### 3.3.4. Оператор мутации

Оператор мутации выполняет:

- с вероятностью 0.5 случайное изменение стартового состояния;
- мутацию случайного состояния.

Оператор мутации состояния, также как и алгоритм обрезки недостижимых ветвей, представляет собой модификацию алгоритма поиска в глубину. При каждом рекурсивном вызове этого алгоритма выполняются следующие действия:

- если текущий узел является листом, то обязательно, а для внутренних узлов с вероятностью  $P$ , вернуть случайно сгенерированное дерево решений;
- иначе равновероятно пойти в одно из поддеревьев.

Запуск описанной рекурсивной процедуры производится с корня дерева.

Фактически данный алгоритм случайно выбирает некоторое поддерево и заменяет его на случайно сгенерированное. Выбор поддерева происходит не равновероятно – чем выше узел, тем больше вероятность выбора его поддерева. Создание случайного поддерева происходит по алгоритму описанному выше.

По окончании мутации для мутированной особи необходимо запустить алгоритм обрезки недостижимых ветвей дерева. Это иллюстрирует рис. 35. На нем изображена ситуация, в которой после описанной операции мутации существует путь из корня до листа, проходящий дважды через вершину с переменной расщепления  $A$ .

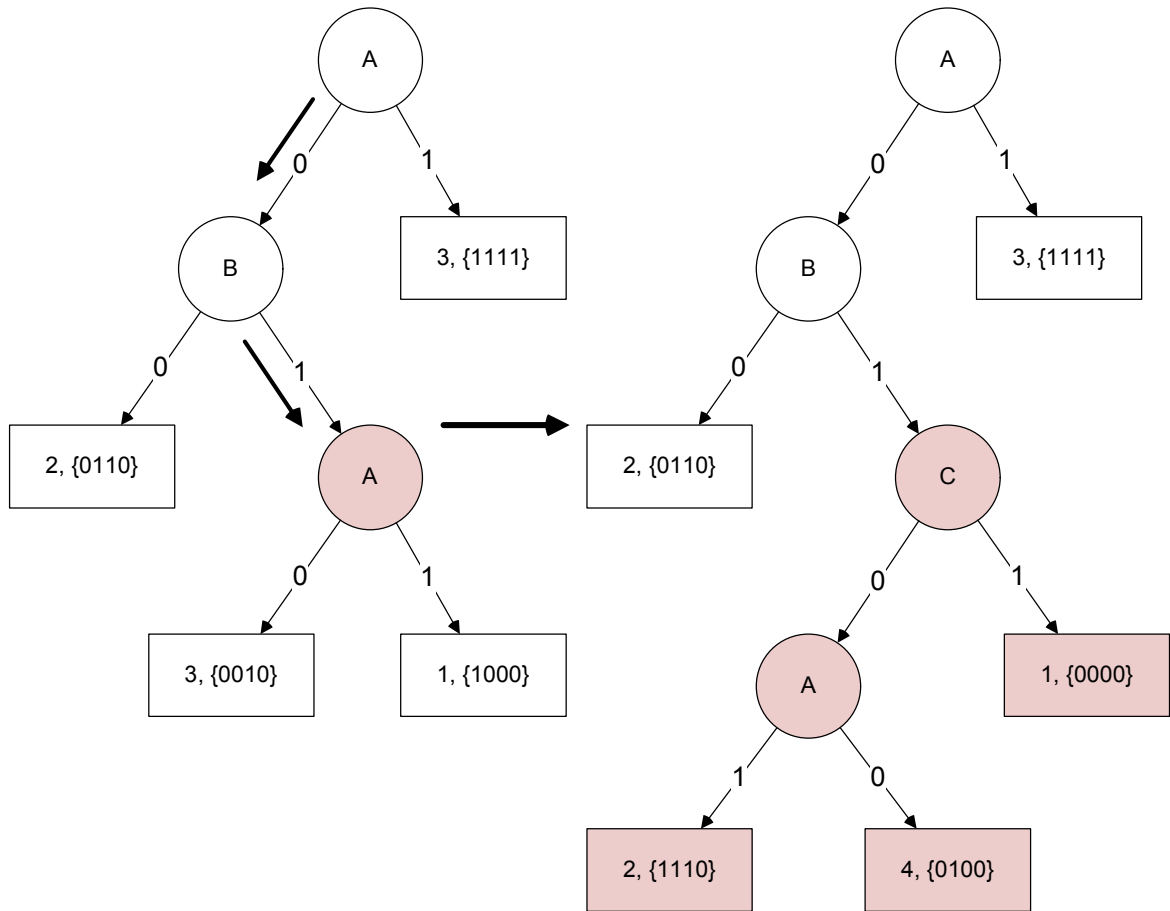


Рис. 35. Мутация деревьев решений



### 3.3.5. Оператор скрещивания

Оператор скрещивания особей представляется следующим образом.

Обозначим родительские особи –  $P1$  и  $P2$ , а детей –  $S1$  и  $S2$ . Обозначим  $k$ -ое состояние автомата  $A$ , как  $A.a[k]$ . Обозначим  $c1[k]$  и  $c2[k]$  результат скрещивания состояний  $P1.a[k]$  и  $P2.a[k]$ . Тогда для любого  $k$  будет верно утверждение:

- $S1.a[k] = c1[k], S2.a[k] = c2[k]$ .

Оператор скрещивания состояний фактически выбирает два поддерева: одно из первого дерева решений, второе из второго, а затем меняет их местами. Этот алгоритм также как операция мутации и алгоритм обрезки недостижимых ветвей представляет собой модификацию алгоритма поиска в глубину и имеет рекурсивную структуру. При каждом рекурсивном вызове этого алгоритма выполняются следующие действия:

- если текущий узел является листом, то обязательно, а для внутренних узлов с вероятностью  $P$ , вернуть данное поддерево;
- иначе равновероятно пойти в одно из поддеревьев текущего узла.

На рис. 36 приведен пример, в котором необходимо, как и в случае с оператором мутации, провести обрезку недостижимых ветвей. При этом для всех деревьев, соответствующих состояниям автомата, запускается алгоритм обрезки недостижимых ветвей.

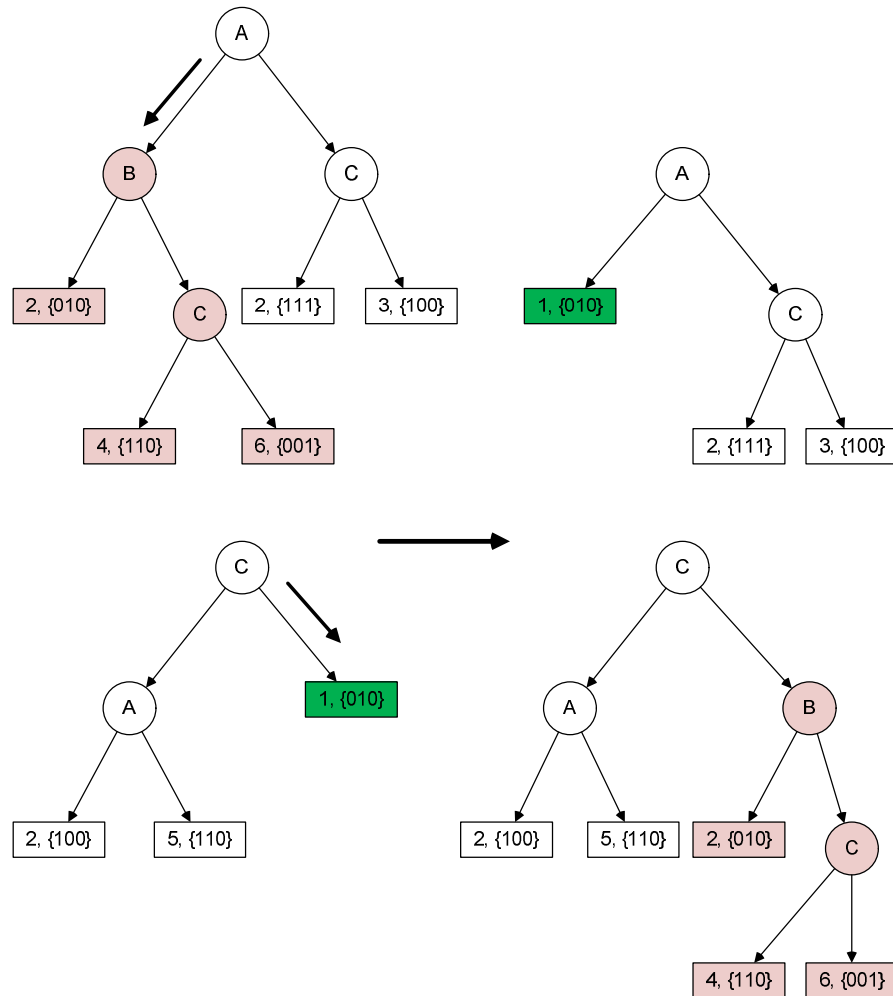


Рис. 36. Пример скрещивание деревьев решений

### 3.3.6. Особенности представления при помощи деревьев решений

Дерево решений имеет те же достоинства, что и сокращенные таблицы:

- так же, как и при использовании сокращенных таблиц, могут использоваться не все входные переменные. Это позволяет с большей вероятностью, чем у представления с помощью битовых строк и полными таблицами переходов, исключить из рассмотрения переходы с несовместными переменными;
- еще большее сокращение памяти, по сравнению с сокращенными таблицами. Даже если в конкретном дереве используется  $k$  входных переменных, то его размер может быть меньше чем  $2^k$ , вплоть до линейного от числа переменных.

Специфика деревьев решений указывает на то, что в каждом состоянии автомата задается свой приоритет входных переменных. Это связано с тем, что чем выше находится лист дерева, в котором происходит расщепление по данной переменной, тем больше вероятность, что в него можно попасть в зависимости от значений других входных переменных.

Еще раз отметим, что не существует жестких ограничений на высоту данного дерева (за исключением случая генерации случайных деревьев). Борьба с излишним «разрастанием» деревьев происходит за счет изменения функции приспособленности, что описано ниже.

### 3.3.7. Генетический алгоритм

Островной генетический алгоритм, используемый для построения управляющих автоматов с использованием метода представления автоматов с помощью деревьев решений, аналогичен описанному в разд. 3.2.7. Отличия состоят в создании начального поколения и в вычислении функции приспособленности.

#### 3.3.7.1. Создание начального поколения

Все острова заполняются случайно сгенерированными особями. Все особи имеют заранее заданное число состояний. Кроме числа состояний задается «рекомендуемая» высота дерева, также одинаковая для всех. Эта высота используется далее при вычислении значения функции приспособленности.

#### 3.3.7.2. Вычисление функции приспособленности

Вычисление функции приспособленности особи осуществляется так же, как описано в разд. 3.2.7.4, но к слагаемому, отвечающему за результаты соревнований, добавляется второе слагаемое, отвечающее за высоту деревьев решений в автомате, описываемом данной особью. Итоговая формула имеет вид (17):

$$F = \frac{\sum_{i=0}^k r_k}{k} - C * Z(h_{max}, height), \quad (17)$$

где  $C$  – некоторая константа,  $h_{max}$  – максимальная из высота деревьев решений, соответствующих состояниям данного автомата,  $height$  – «рекомендуемая» высота дерева решений, а  $Z$  – функция определяемая следующим образом:

$$Z(a, b) = \begin{cases} a - b, & a \geq b; \\ 0, & a < b. \end{cases} \quad (18)$$

Таким образом, данная функция отвечает за то чтобы дерево решений не сильно «разрасталось». При этом если дерево решений имеет высоту больше, чем *height*, функция приспособленности такой особи может быть ниже, даже если средний результат ее соревнований выше.

### 3.3.8. Результат применения метода представления автоматов с помощью деревьев решений

При помощи алгоритма генетического программирования с использованием метода представления автоматов с помощью деревьев решений построен конечный автомат управления беспилотным летательным аппаратом. Его граф переходов приведен на рис. 37. Построение этого автомата заняло 32 поколения при следующих значениях параметров алгоритма:

- число островов – 10;
- размер острова – 100 особей;
- процент «элиты» – 10%;
- вероятность мутации – 10%;
- процент особей, участвующих в миграции между островами, – 3%;
- число поколений, проходящее между миграциями особей между островами – 10;
- число поколений, проходящее между «большими» мутациями, – 41;
- рекомендуемая высота дерева решений – 4;
- вероятность генерации внутреннего узла дерева решений – 75%;
- процент островов, на которых все особи заменяются случайно сгенерированными в процессе «большой» мутации, – 85%.

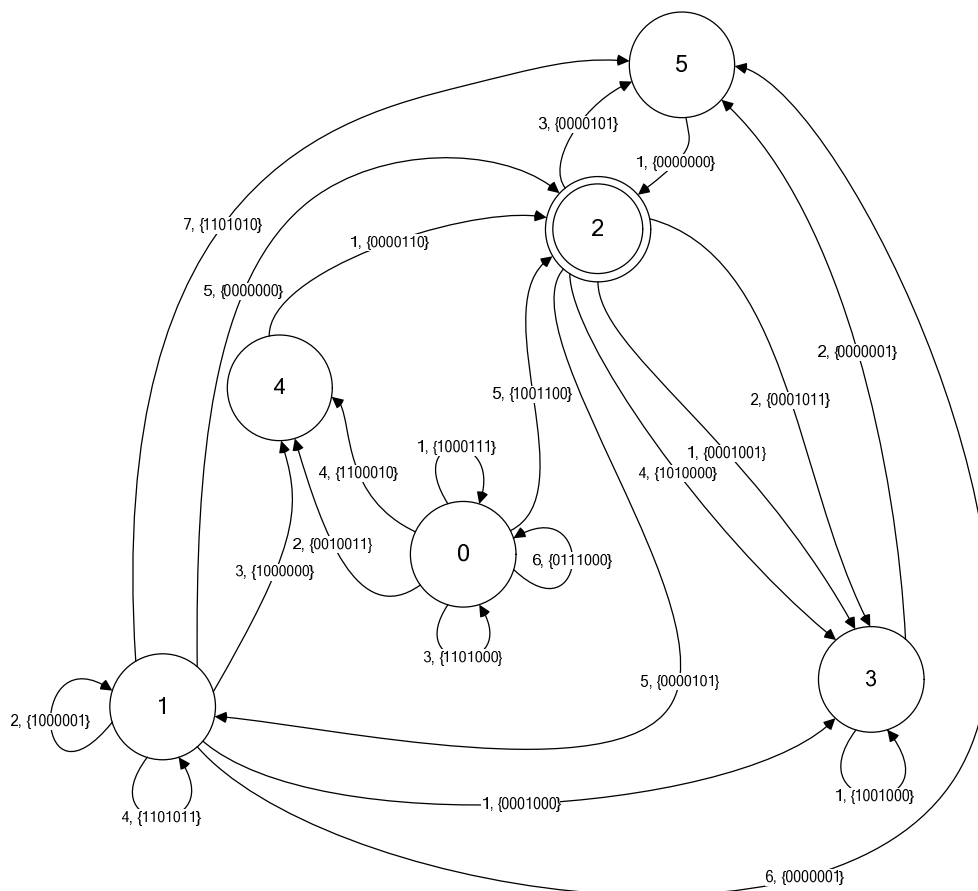


Рис. 37. Граф переходов конечного автомата, построенного с применением метода

## представления автоматов с помощью деревьев решений

На этом рисунке пометки на переходах имеют следующий формат: *номер перехода, {вектор действий}*. Опишем семантику каждой компоненты вектора действий:

1. Установить нормальный расход топлива.
2. Изменить направление аэродинамического руля на фиксированный угол налево.
3. Изменить направление аэродинамического руля на фиксированный угол направо.
4. Лететь прямо.
5. Сделать расход топлива максимально возможным.
6. Увеличить расход топлива на фиксированную величину.
7. Уменьшить расход топлива на фиксированную величину.

Переходы из каждого состояния пронумерованы натуральными числами, начиная с единицы. Номера переходов соответствуют номерам листов деревьев решений, показанных на рисунках ниже. Заметим, что в этом автомате состояние с номером ноль недостижимо и может быть удалено.

На рис. 38 показано дерево решений, соответствующее состоянию с номером ноль построенного автомата.

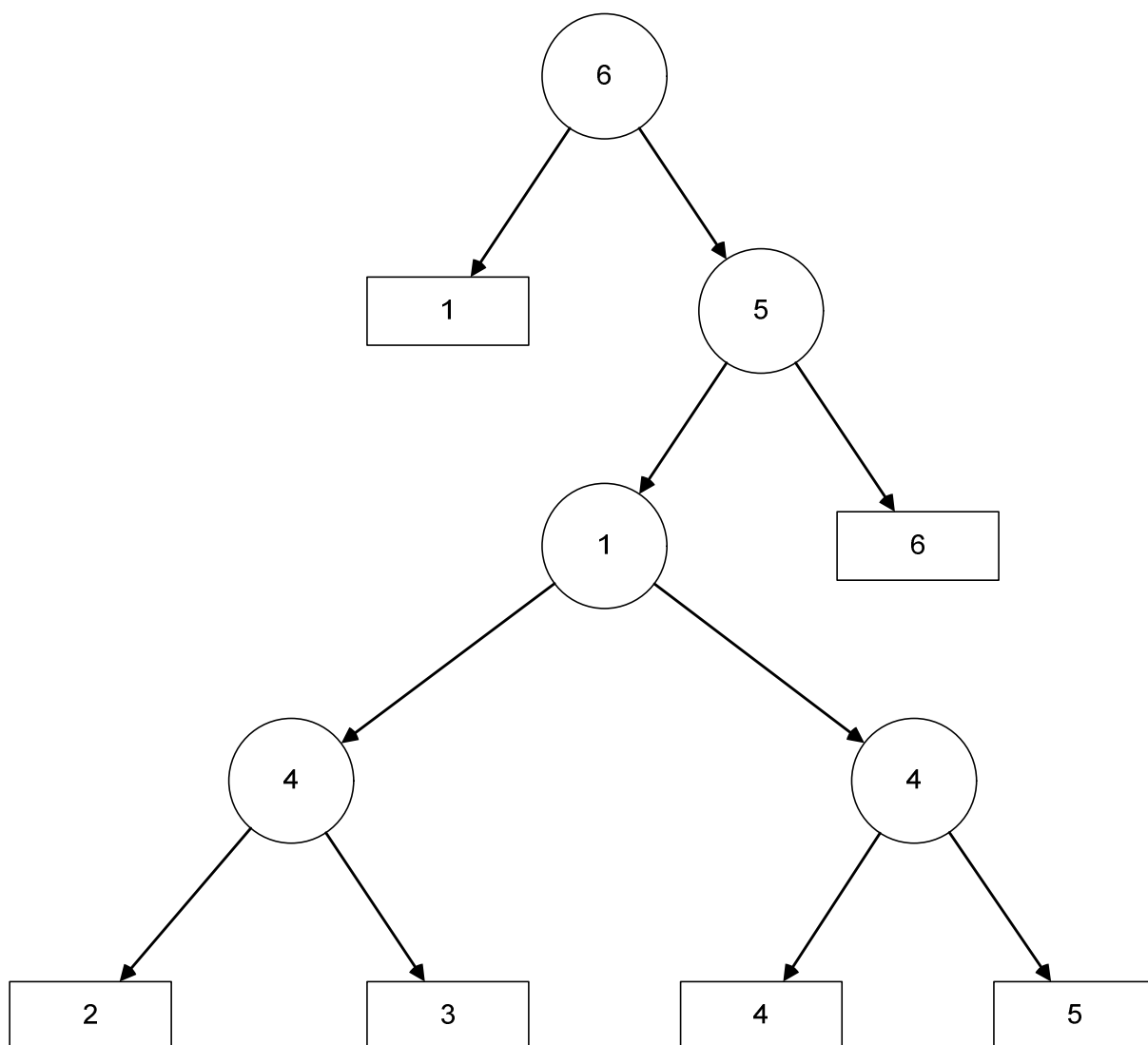


Рис. 38. Дерево решений, соответствующее нулевому состоянию автомата

На этом рисунке внутренние узлы дерева решений показаны кругами, а листья – прямоугольниками. Внутренние узлы помечены номером переменной, по которой осуществляется ветвление. Левое поддерево соответствует значению «ложь» переменной, которой помечен узел, а правое – значению «истина». Листья помечены номером перехода, который выполняется при достижении этого листа. Эти номера переходов соответствуют номерам переходов на рис. 37. Приведем перечень переменных, используемых на приведенных в настоящем разделе деревьях:

1. Граница трассы справа.
2. Граница трассы слева.
3. Другой аппарат слева.
4. Другой аппарат справа.
5. Другой аппарат спереди.
6. Другой аппарат сзади.

На рис. 39 показано дерево решений, соответствующее состоянию с номером один построенного автомата.

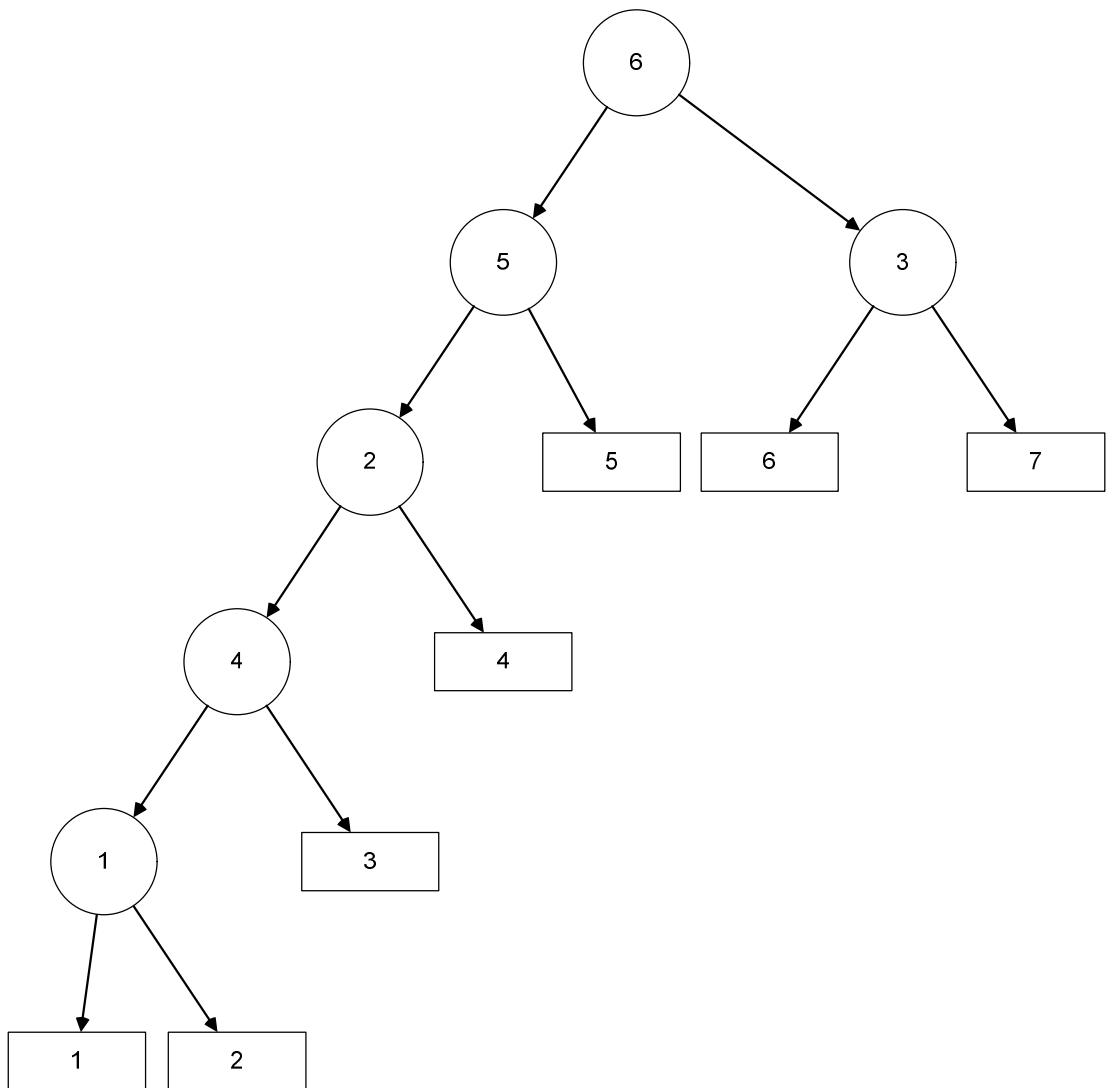


Рис. 39. Дерево решений, соответствующее первому состоянию автомата

На рис. 40 показано дерево решений, соответствующее второму состоянию автомата.

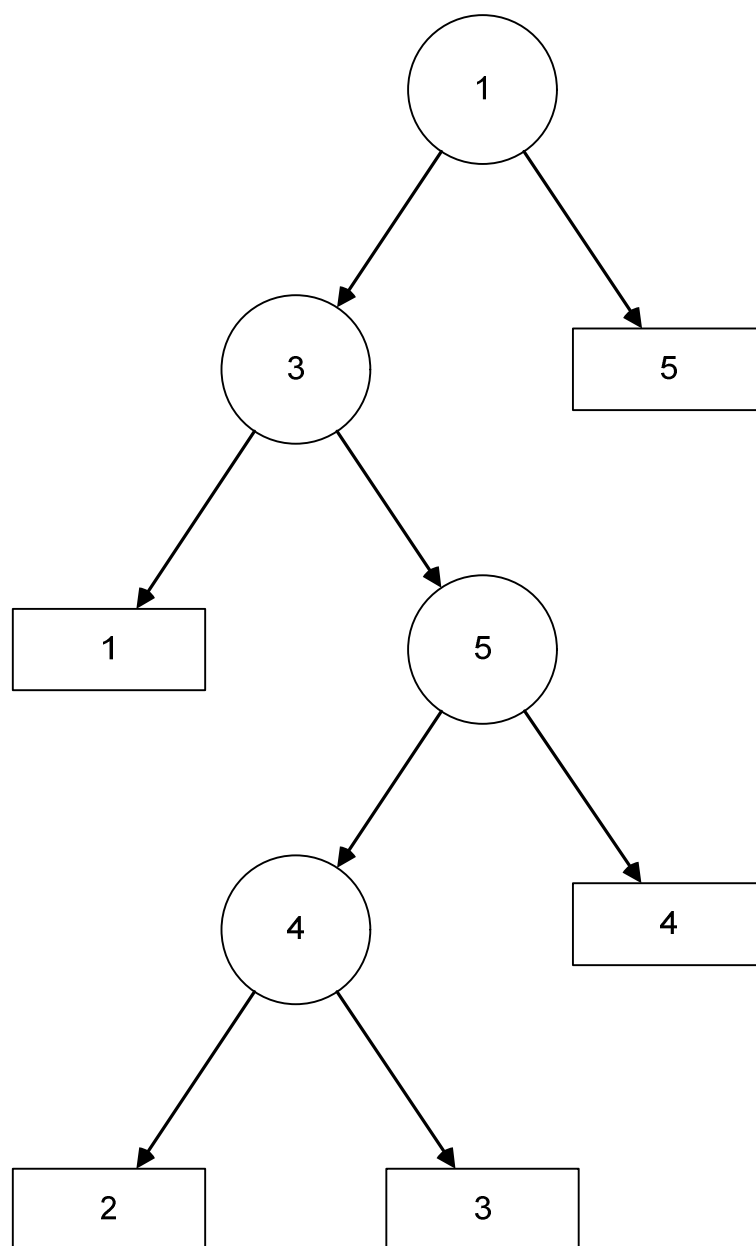


Рис. 40. Дерево решений, соответствующее второму состоянию автомата

На рис. 41 показано дерево решений, соответствующее второму состоянию автомата.

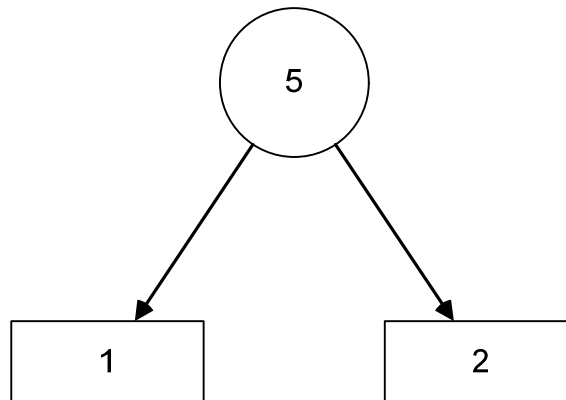


Рис. 41. Дерево решений, соответствующее третьему состоянию автомата

На рис. 42 показано дерево решений, соответствующее второму состоянию автомата.



Рис. 42. Дерево решений, соответствующее четвертому и пятому состояниям автомата

Наличие в этом дереве только одного узла, который является одновременно и корнем и листом, означает, что из этих состояний всегда выполняется один и тот же безусловный переход.

### 3.4. Выводы

На основе методов генерации автоматов управления системами со сложным поведением были автоматически построены три системы управления моделью беспилотного летящего аппарата:

1. На основе совместного применения генетического программирования и нейронных сетей.
2. На основе метода сокращенных таблиц переходов и алгоритма восстановления связей между состояниями.
3. На основе метода представления автоматов деревьями решений.

Разработанные системы в следующем разделе будут сравнены между собой и ручной реализацией, что позволит оценить соответствующие методы генерации автоматов управления системами со сложным поведением.



## 4. СРАВНЕНИЕ СИСТЕМ УПРАВЛЕНИЯ

В настоящем разделе описаны результаты сравнения систем управления, построенных с помощью методов, описанных в предыдущем разделе. Системы управления моделью беспилотного летательного аппарата тестировались в разработанной в работе [27] среде моделирования соревнований моделей летательных аппаратов. При этом проводилось соревнование каждой построенной системы с каждой, а также с системой, построенной без применения методов автоматизации и с системой, реализующей «агрессивную» стратегию.

Каждое соревнование проводилось 50 раз. При этом учитывалось число моделей беспилотных летательных аппаратов, успешно завершивших соревнование, и наибольшее расстояние, на которое удалилась от линии старта одна из моделей беспилотного летательного аппарата. Далее, полученные результаты усреднялись, и строилось их распределение.

### 4.1. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ С СИСТЕМОЙ УПРАВЛЕНИЯ, ПОСТРОЕННОЙ БЕЗ ПРИМЕНЕНИЯ МЕТОДОВ АВТОМАТИЗАЦИИ

В настоящем разделе описаны результаты команды беспилотных летательных аппаратов с системой управления, построенной без применения методов автоматизации.

#### 4.1.1. Соревнование с командой беспилотных летательных аппаратов, реализующей «агрессивную» стратегию

На рис. 43 показано распределение результатов, которые показала команда беспилотных летательных аппаратов с системой управления, построенной без применения методов автоматизации, в соревновании с командой, реализующей «агрессивную» стратегию.

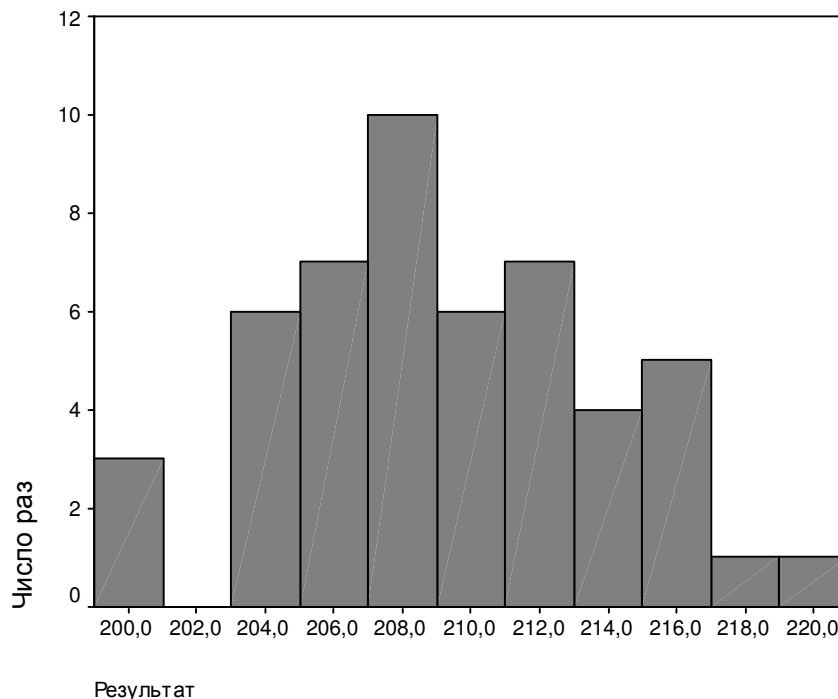
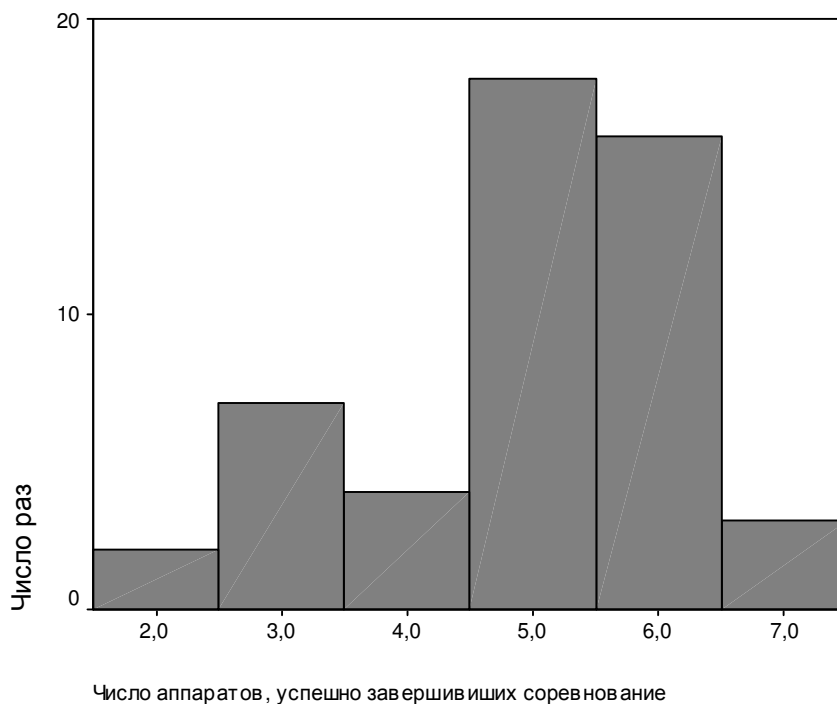


Рис. 43. Распределение результатов, показанных системой, построенной без применения методов автоматизации

На рис. 44 показано распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, которое было получено в результате соревнований команды беспилотных летательных аппаратов с системой, построенной без применения методов автоматизации, с командой, реализующей «агрессивную» стратегию.



**Рис. 44. Распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации**

На рис. 45 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, с «агрессивной» стратегией, в соревнованиях с командой с системой, построенной без применения методов автоматизации.

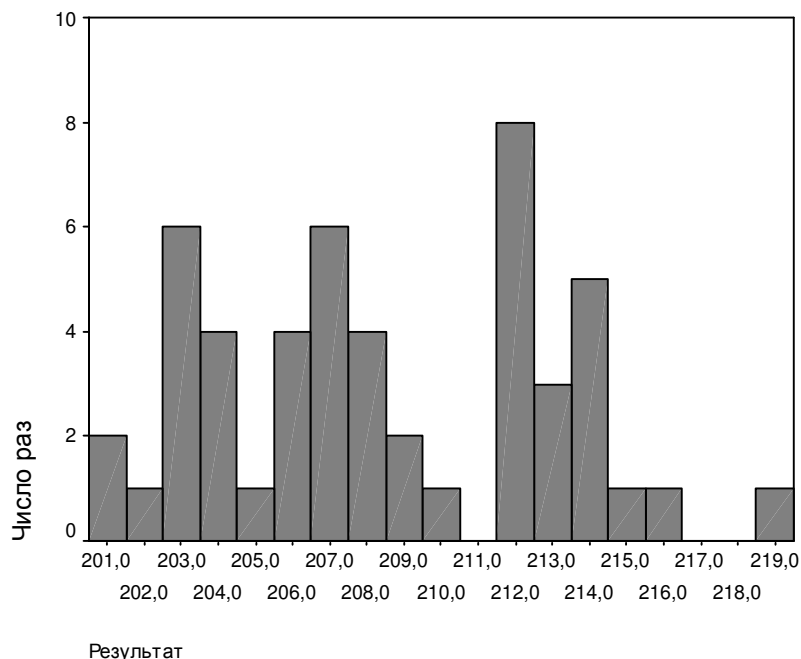


Рис. 45. Распределение результатов, показанных командой, реализующей «агрессивную» стратегию

Гистограмма распределения числа успешно завершивших соревнования моделей беспилотных летательных аппаратов системы, реализующей «агрессивную» стратегию не приводится, так как при такой стратегии соревнования всегда успешно завершала ровно одна модель беспилотного летательного аппарата.

В табл. 11 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 11. Результаты соревнований

	Система управления, построенная без применения методов автоматизации		Система управления, реализующая «агрессивную» стратегию	
	Результат	Число аппаратов, успешно завершивших соревнования	Результат	Число аппаратов, успешно завершивших соревнования
Минимум	199,54	2	201,38	1
Максимум	220,20	7,00	219,47	1,00
Среднее	209,08	4,90	208,28	1,00

Из анализа приведенных результатов следует, что система управления моделью беспилотного летательного аппарата, построенная без применения методов автоматизации, и система, реализующая «агрессивную» стратегию показывают примерно одинаковые результаты.

#### 4.1.2. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

На рис. 46 показано распределение результатов, которые показала команда беспилотных летательных аппаратов с системой, построенной без применения методов автоматизации, в соревновании с командой с системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.

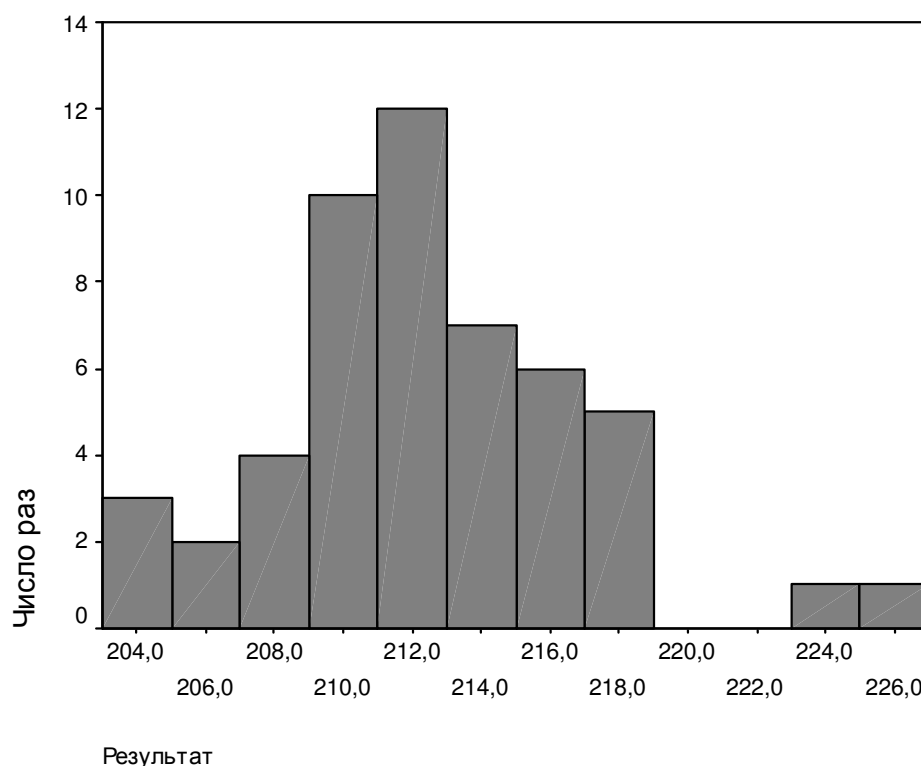


Рис. 46. Распределение результатов, обеспеченных системой управления, построенной без применения методов автоматизации

На рис. 47 показано распределение числа успешно завершивших соревнование моделей беспилотных летательных аппаратов, которое было получено в результате соревнований системы управления моделью беспилотного летательного аппарата, построенной без применения методов автоматизации, с системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.

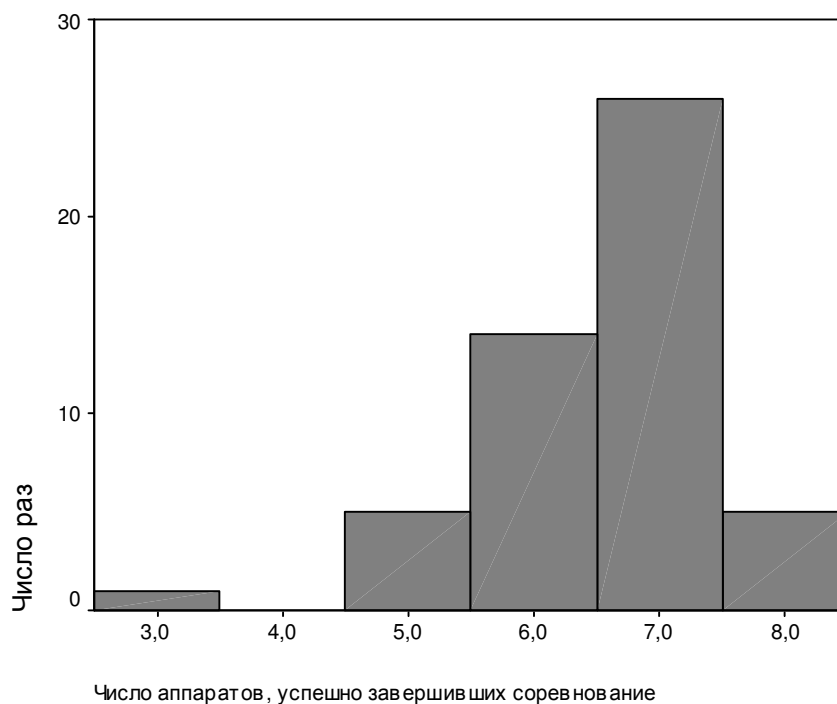


Рис. 47. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации

В табл. 12 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 12. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, конечных автоматов и нейронных сетей		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	203,05	1	203,44	3
Максимум	241,11	8,00	225,09	8,00
Среднее	215,57	4,79	212,31	6,52

Из анализа приведенных результатов следует, что команда с системой, построенной без применения методов автоматизации, показывает несколько худший результат, чем команда с системой, построенной с помощью метода совместного применения генетического

программирования, автоматов и нейронных сетей по дальности полета, однако в среднем результаты похожи.

### 4.1.3. Соревнование командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода сокращенных таблиц переходов

На рис. 48 показано распределение результатов, которые показала команда беспилотных летательных аппаратов с системой, построенной с помощью метода сокращенных таблиц переходов, в соревновании с командой с системой, построенной без применения методов автоматизации.

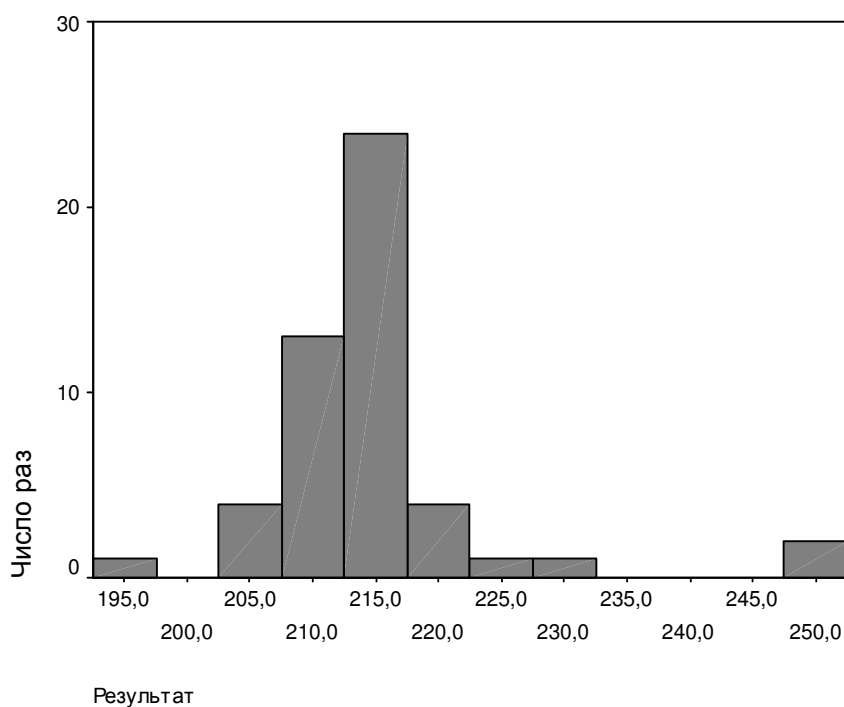


Рис. 48. Распределение результатов, обеспеченных системой, построенной без применения методов автоматизации

На рис. 49 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды с системы управления моделью беспилотного летательного аппарата, построенной с помощью метода сокращенных таблиц, с командой с системой, построенной без применения методов автоматизации.

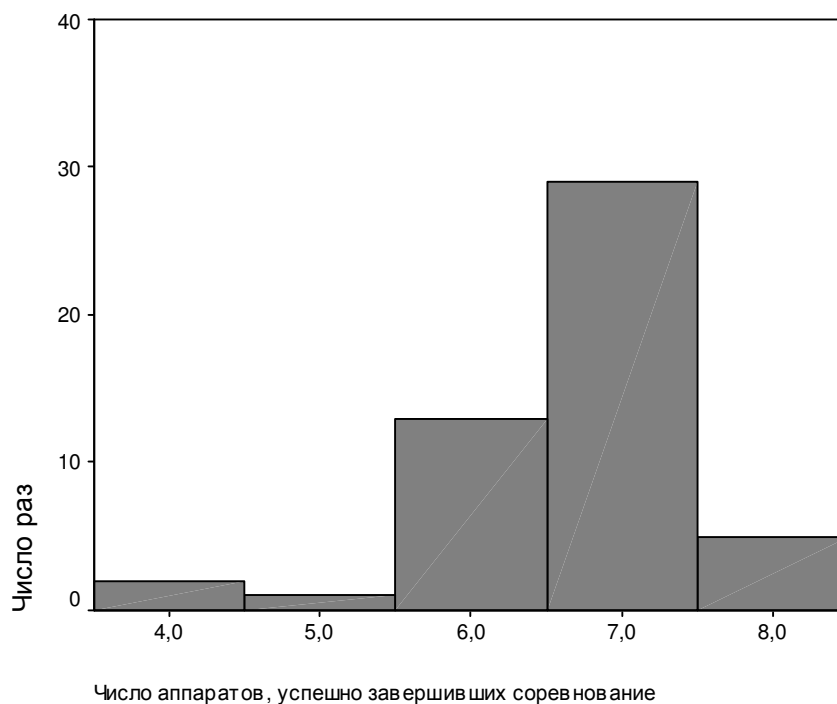


Рис. 49. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации

В табл. 13 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 13. Результаты соревнований

	Система управления, построенная с помощью метода сокращенных таблиц переходов		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	212,34	4	197,16	4
Максимум	309,34	7,00	251,20	8,00
Среднее	268,69	6,69	214,52	6,73

Из анализа приведенных результатов следует, что команда с системой, построенной без применения методов автоматизации, проигрывает команде с системой, построенной с помощью метода сокращенных таблиц переходов, по дальности полета.

#### 4.1.4. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода представления автоматов деревьями решений

На рис. 50 показано распределение результатов, которые показала команда беспилотных летательных аппаратов с системой, построенной без применения методов автоматизации, в соревновании с командой с системой, построенной с помощью метода представления автоматов деревьями решений.

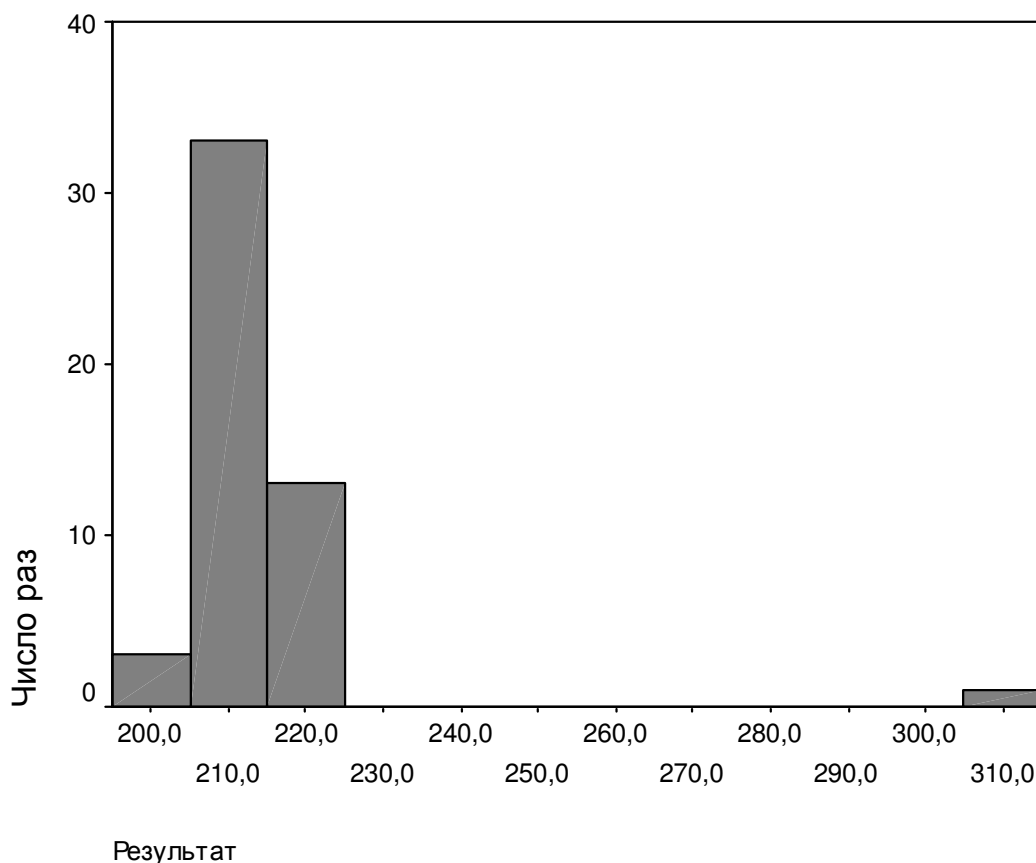


Рис. 50. Распределение результатов, обеспеченных системой, построенной без применения методов автоматизации

На рис. 51 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды с системой управления, построенной без применения методов автоматизации, с командой с системой, построенной с помощью метода представления автоматов деревьями решений.



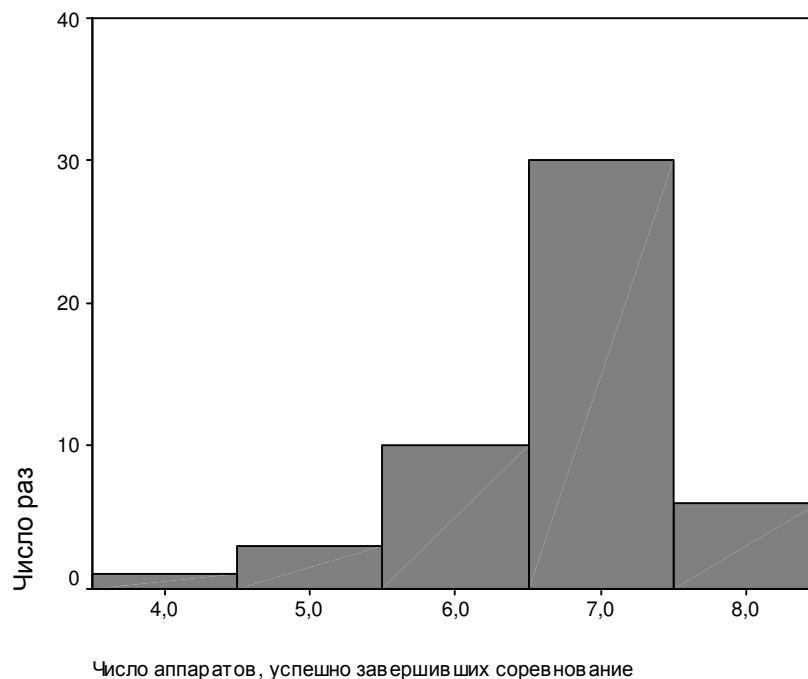


Рис. 51. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации

В табл. 14 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 14. Результаты соревнований

	Система управления, построенная с помощью метода представления автоматов деревьями решений		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	210,45	2	196,68	4
Максимум	303,43	7,00	310,80	8,00
Среднее	251,61	6,53	214,66	6,72

Из анализа приведенных результатов следует, что команда с системой, построенной без применения методов автоматизации, в среднем проигрывает более 20% команде с системой, построенной с помощью метода представления автоматов деревьями решений. При этом число моделей беспилотных летательных аппаратов, успешно завершивших соревнование у построенной без применения методов автоматизации системы больше.

## 4.2. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМЫХ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА СОВМЕСТНОГО ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ, АВТОМАТОВ И НЕЙРОННЫХ СЕТЕЙ

В настоящем разделе описаны результаты, показанные командой, управляемой построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей системой управления моделью беспилотного летательного аппарата. Этот метод описан в разд. 3.1, а построенная с его помощью система управления представлена в разд. 3.1.4.7.

### 4.2.1. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной без применения методов автоматизации

На рис. 52 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, в соревновании с командой с системой, построенной без применения методов автоматизации.

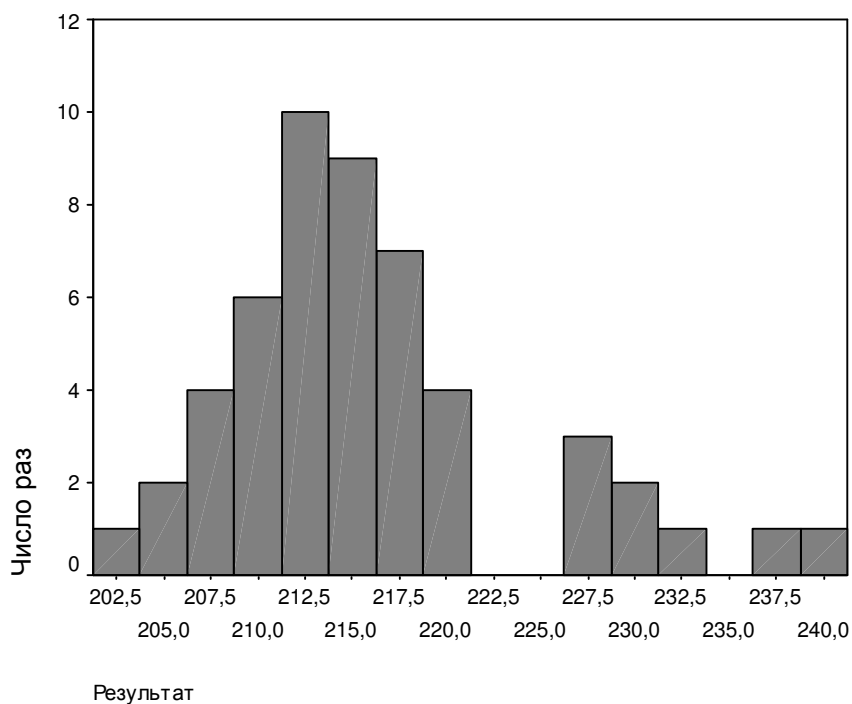


Рис. 52. Распределение результатов, обеспеченных системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

На рис. 53 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды с системой управления, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, с командой с системой, построенной без применения методов автоматизации.

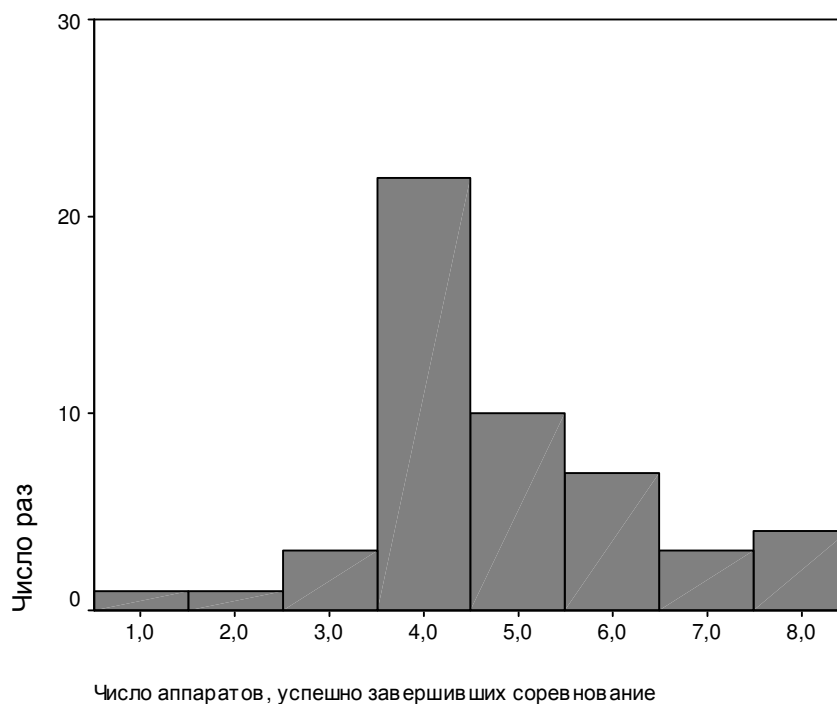


Рис. 53. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

В табл. 15 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 15. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, конечных автоматов и нейронных сетей		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	203,05	1	203,44	3
Максимум	241,11	8,00	225,09	8,00
Среднее	215,57	4,79	212,31	6,52

Из анализа приведенных результатов следует, что система, построенная без применения методов автоматизации, проигрывает системе, построенной с помощью метода сокращенных таблиц

переходов по дальности полета. При этом среднее число беспилотных летательных аппаратов, успешно завершивших соревнование, меньше у системы управления построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей без применения методов автоматизации.

#### 4.2.2. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, реализующей «агрессивную» стратегию

На рис. 54 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, в соревновании с командой с системой, реализующей «агрессивную» стратегию.

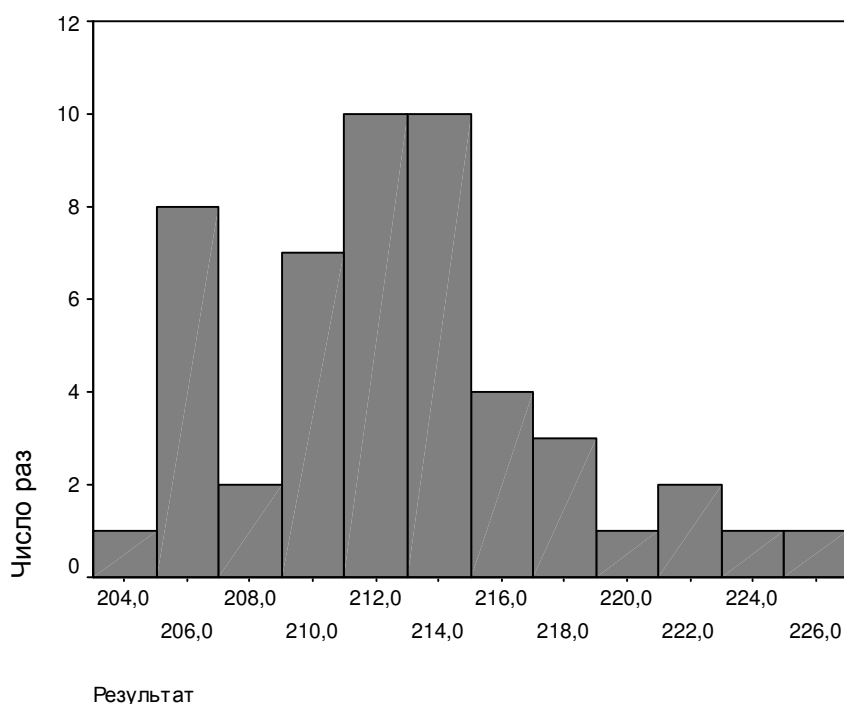
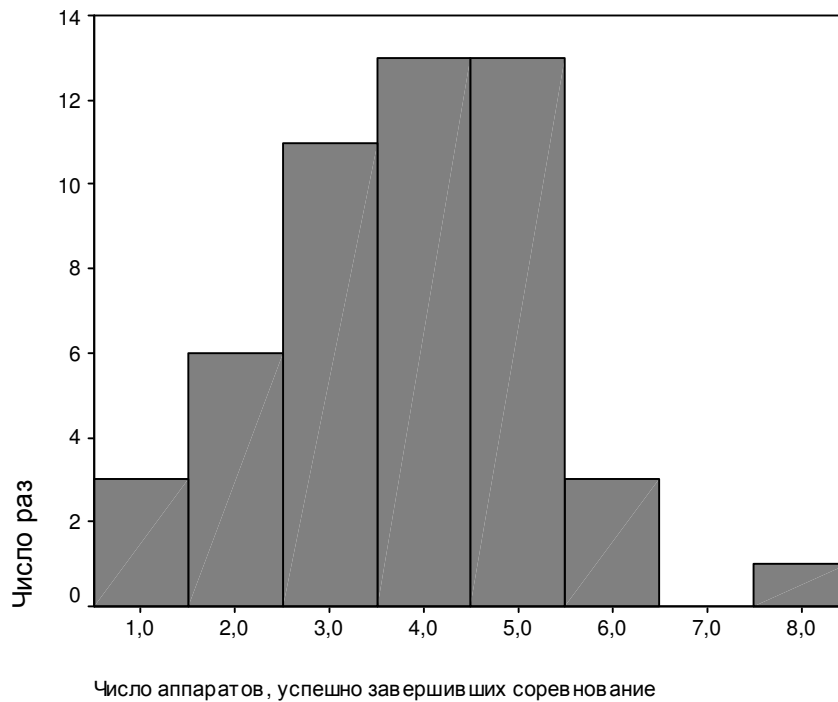


Рис. 54. Распределение результатов, обеспеченных системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

На рис. 55 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды с системой, построенной с помощью метода совместного применения генетических алгоритмов, автоматов и нейронных сетей, с командой с системой, реализующей «агрессивную» стратегию.



**Рис. 55. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей**

На рис. 56 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, реализующая «агрессивную» стратегию, в соревнованиях с командой, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.

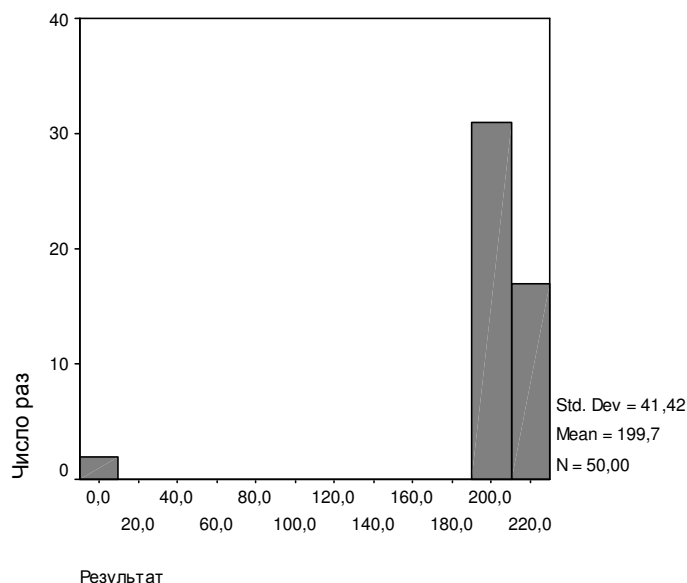


Рис. 56. Распределение результатов, показанных командой с системой, реализующей агрессивную стратегию

Гистограмма распределения числа успешно завершивших соревнование моделей беспилотных летательных аппаратов системы, реализующей «агрессивную» стратегию не приводится, так как при такой стратегии и ненулевом результате соревнование всегда успешно завершала ровно одна модель беспилотного летательного аппарата. Если же результат данной системы нулевой, то ни один беспилотный летательный аппарат, управляемый этой системой, не смог успешно завершить соревнование.

В табл. 16 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 16. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей		Система управления, реализующая «агрессивную» стратегию	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	203,12	1	0,00	0
Максимум	225,83	8,00	215,06	1,00
Среднее	212,42	3,77	199,05	0,96

Из анализа приведенных результатов следует, что команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, вполне успешно противостоит команде, реализующей агрессивную стратегию, побеждая ее примерно в четырех соревнованиях из пяти.

#### 4.2.3. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода сокращенных таблиц переходов

На рис. 57 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, в соревновании с командой, управляемой системой, построенной с помощью метода сокращенных таблиц.

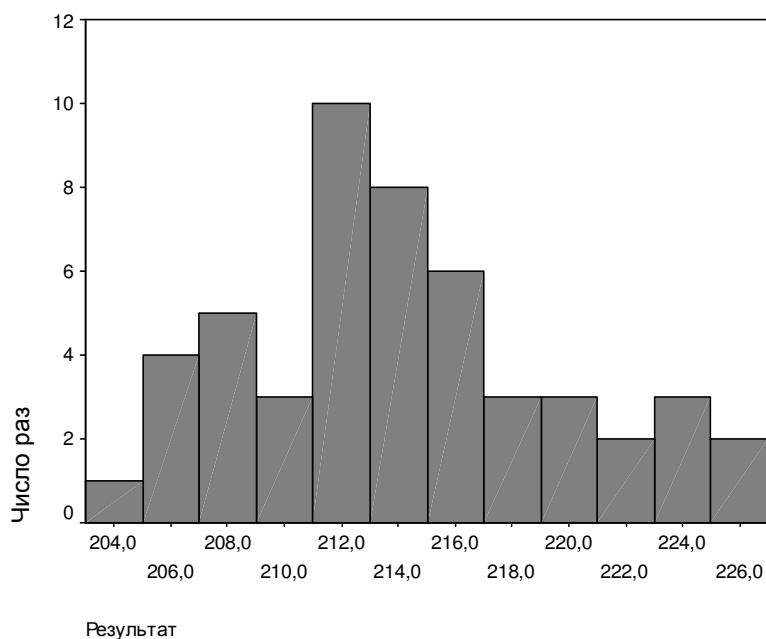


Рис. 57. Распределение результатов, обеспеченных системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

На рис. 58 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, с командой, управляемой системой, построенной с помощью метода сокращенных таблиц переходов.

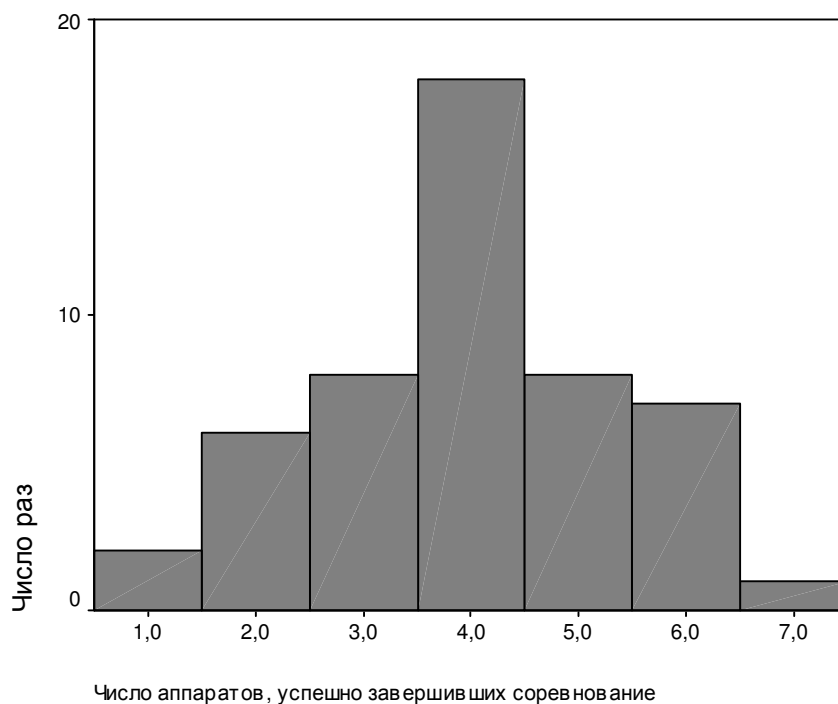


Рис. 58. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

В табл. 17 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 17. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей		Система управления, построенная с помощью метода сокращенных таблиц переходов	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	204,04	1	201,32	1
Максимум	226,43	6,00	312,89	7,00
Среднее	214,17	3,98	248,86	5,13

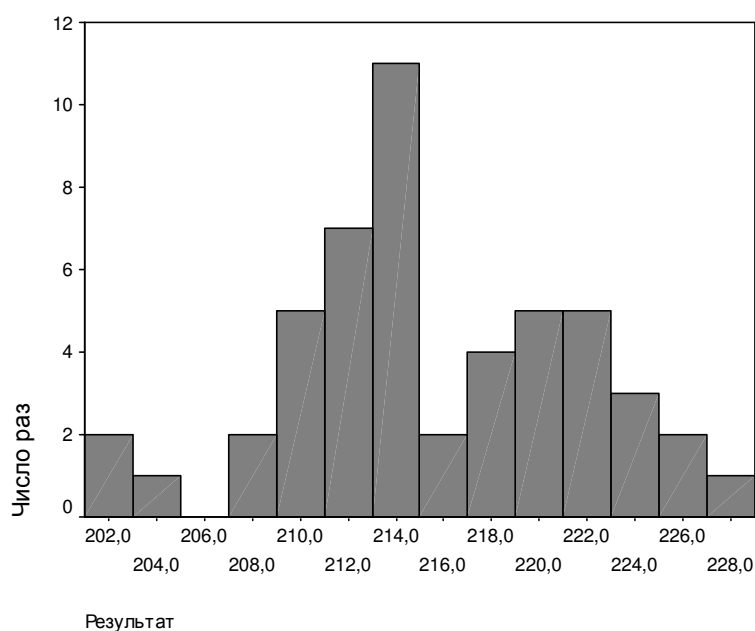
Из анализа приведенных результатов следует, что система, построенная с помощью совместного применения генетического программирования, автоматов и нейронных сетей,



проигрывает системе, построенной с помощью метода сокращенных таблиц переходов, как по дальности полета, так и по числу моделей беспилотных летательных аппаратов, успешно завершающих соревнование.

#### **4.2.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода представления автоматов деревьями решений**

На рис. 59 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, в соревновании с командой, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.



**Рис. 59. Распределение результатов, обеспеченных системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей**

На рис. 60 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей, с командой, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.

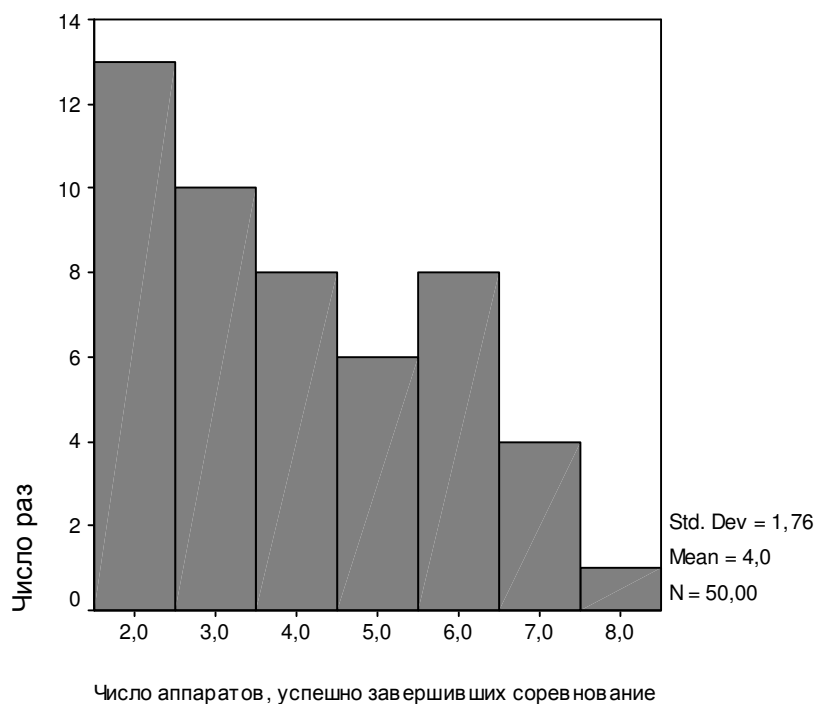


Рис. 60. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей

В табл. 18 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 18. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей		Система управление, построенная с помощью метода представления автоматов деревьями решений	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	201,36	2	212,17	3
Максимум	228,14	8,00	303,85	7,00
Среднее	215,46	4,00	248,90	5,92

Из анализа приведенных результатов следует, что система, построенная с помощью совместного применения генетического программирования, автоматов и нейронных сетей,

проигрывает системе, построенной с помощью метода представления автоматов деревьями решений, как по дальности полета, так и по числу моделей беспилотных летательных аппаратов, успешно завершающих соревнование.

Анализ приведенных результатов позволяет сделать вывод, что построенная с помощью генетического программирования система в среднем показывает лучший результат, чем система, построенная в работах [26, 27] без применения методов автоматизации. Наиболее вероятной причиной этого является высокая эффективность системы, построенной с помощью генетического программирования.

Выигрывая по результату, команда с системой управления, построенной с помощью генетического программирования, однако, проигрывает по числу беспилотных летательных аппаратов, успешно завершивших гонку. При этом во многих случаях это связано с тем, что летательные аппараты, управляемой системой, построенной с помощью генетического программирования, сталкиваются друг с другом или выходят за границы трассы. Однако, в ряде случаев они демонстрируют достаточно «разумное» поведение, двигаясь прямо при отсутствии вблизи препятствий и уклоняясь от других аппаратов и границ коридора в случае их наличия по близости.

Такое положение дел, скорее всего, связано с недостаточно эффективной функцией приспособленности, учитывающей только общий результат команды и не учитывающей поведение агентов во время соревнования. Поэтому одним из направлений улучшений приведенного в настоящей работе алгоритма является применение функций приспособленности, обеспечивающих большую эффективность решения задачи за счет учета большего числа параметров.

#### **4.3. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМОЙ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА СОКРАЩЕННЫХ ТАБЛИЦ ПЕРЕХОДОВ**

В настоящем разделе описаны результаты, показанные командой беспилотных летательных аппаратов, управляемой построенной с помощью метода сокращенных таблиц переходов системой управления. Этот метод описан в разд. 3.2, а построенная система управления представлена в разд. 3.2.8.

##### **4.3.1. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, реализующей «агрессивную» стратегию**

На рис. 61 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода сокращенных таблиц, в соревновании с командой, управляемой системой, реализующей «агрессивную» стратегию.

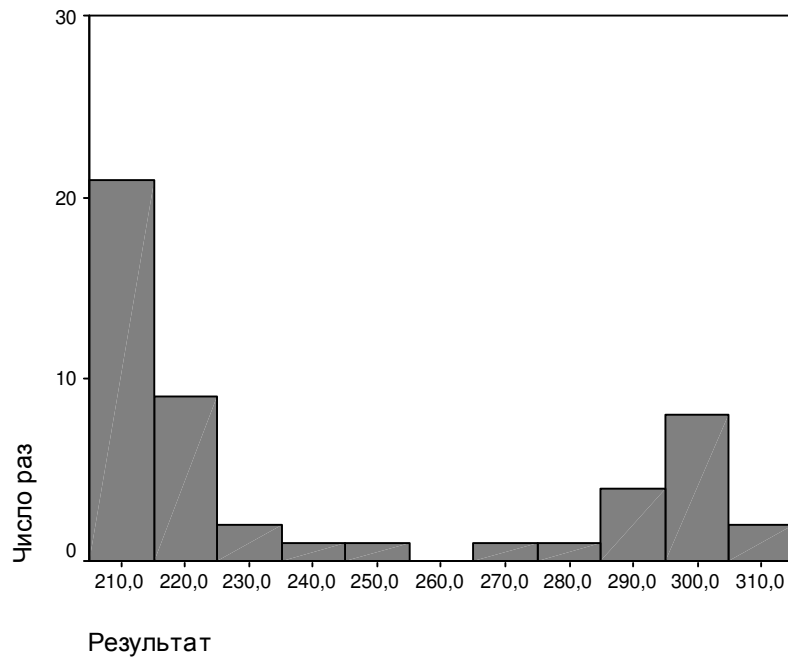


Рис. 61. Распределение результатов, обеспеченных системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 62 показано распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода сокращенных таблиц, с командой, управляемой системой, реализующей «агрессивную» стратегию.

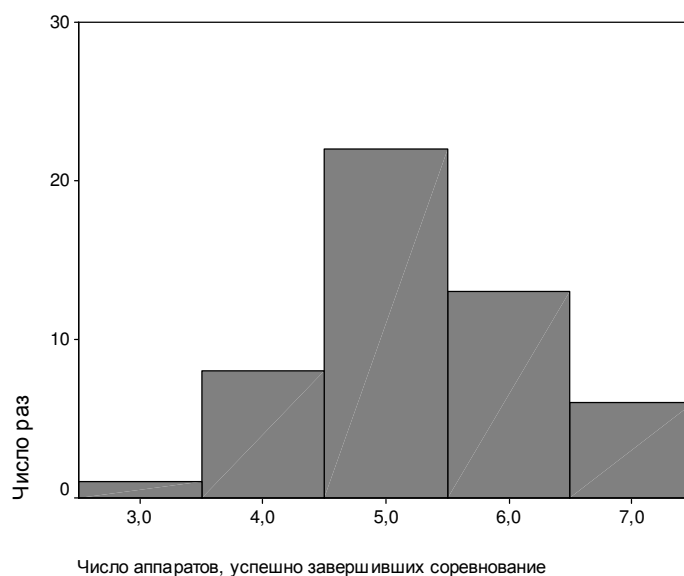


Рис. 62. Распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 63 показано распределение результатов, которые показала система управления моделью беспилотного летательного аппарата, построенная с помощью метода сокращенных таблиц, в соревновании с системой, реализующей «агрессивную» стратегию.

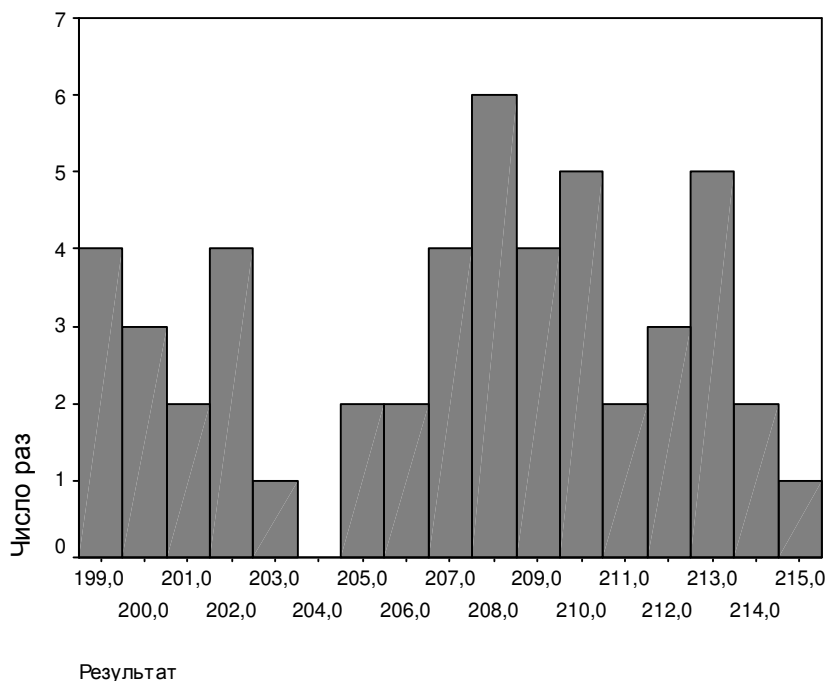


Рис. 63. Распределение результатов, показанных системой, реализующей агрессивную стратегию

Гистограмма распределения числа успешно завершивших соревнование беспилотных летательных аппаратов команды, реализующей «агрессивную» стратегию не приводится, так как при такой стратегии соревнование всегда успешно завершала ровно одна модель беспилотного летательного аппарата.

В табл. 19 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 19. Результаты соревнований

	Система управления, построенная с помощью метода сокращенных таблиц переходов		Система управления, реализующая «агрессивную» стратегию	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	207,41	3	198,66	1
Максимум	308,65	7,00	214,75	1
Среднее	241,87	5,29	207,09	1

Из анализа приведенных результатов следует, что система, построенная при помощи метода сокращенных таблиц, успешно противостоит системе, реализующей «агрессивную» стратегию. При этом преимущество в среднем составляет 20%, а также в среднем более пяти моделей беспилотных летательных аппаратов успешно завершают соревнование.

#### 4.3.2. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной без применения методов автоматизации

На рис. 64 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода сокращенных таблиц, в соревновании с командой, управляемой системой, реализующей «агрессивную» стратегию.

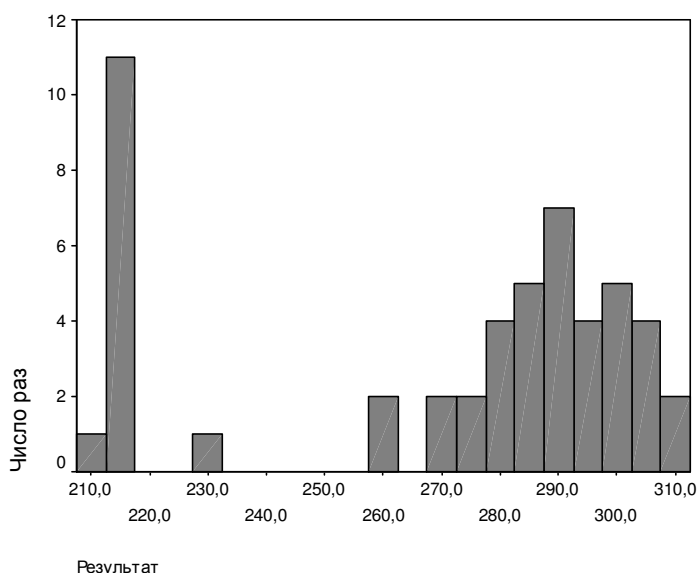


Рис. 64. Распределение результатов, обеспеченных системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 65 показано распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода сокращенных таблиц, с командой, управляемой системой, реализующей «агрессивную» стратегию.

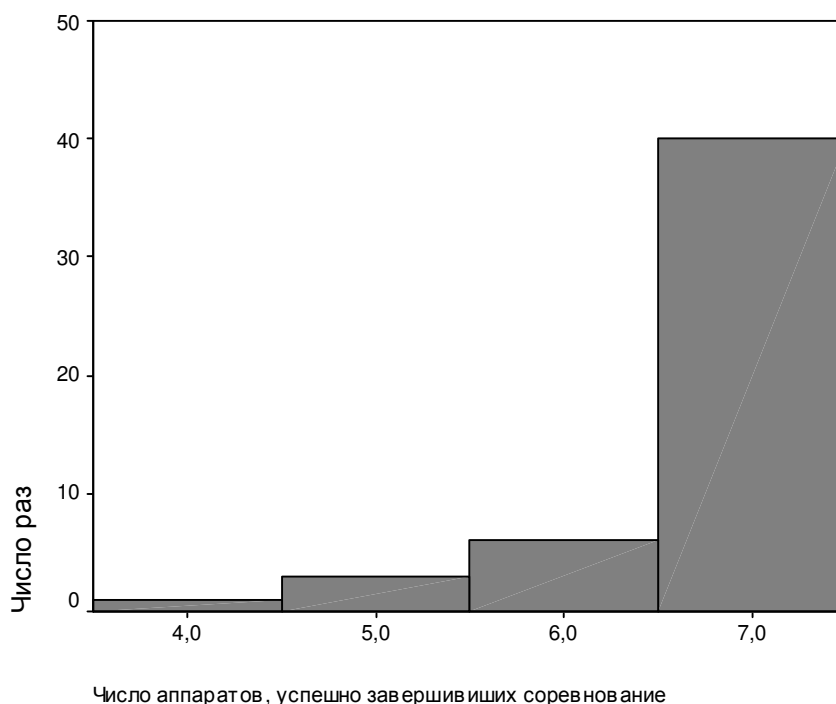


Рис. 65. Распределение числа успешно завершивших соревнования беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода сокращенных таблиц переходов

В табл. 20 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 20. Результаты соревнований

	Система управления, построенная с помощью метода сокращенных таблиц переходов		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнования	Результат	Число аппаратов, успешно завершивших соревнования
Минимум	212,34	4	197,16	4
Максимум	309,34	7,00	251,20	8,00
Среднее	268,69	6,69	214,52	6,73

Из анализа приведенных результатов следует, что система, построенная при помощи метода сокращенных таблиц, построенная при помощи метода сокращенных таблиц, превосходит систему, построенную без применения методов автоматизации как по дальности полета, так и по числу моделей беспилотных летательных аппаратов, успешно завершающих соревнование.

#### **4.3.3. Соревнование с командой, беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей**

На рис. 66 показано распределение результатов, которые показала команда, управляемая системой, построенной с помощью метода сокращенных таблиц, в соревновании с командой, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.

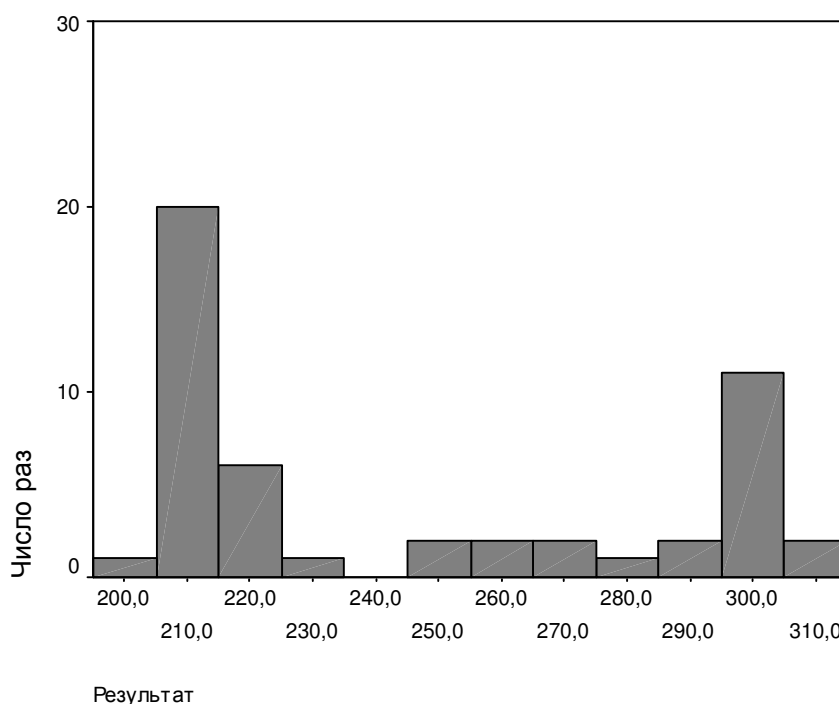


Рис. 66. Распределение результатов, обеспеченных системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 67 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода сокращенных таблиц, с командой, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.



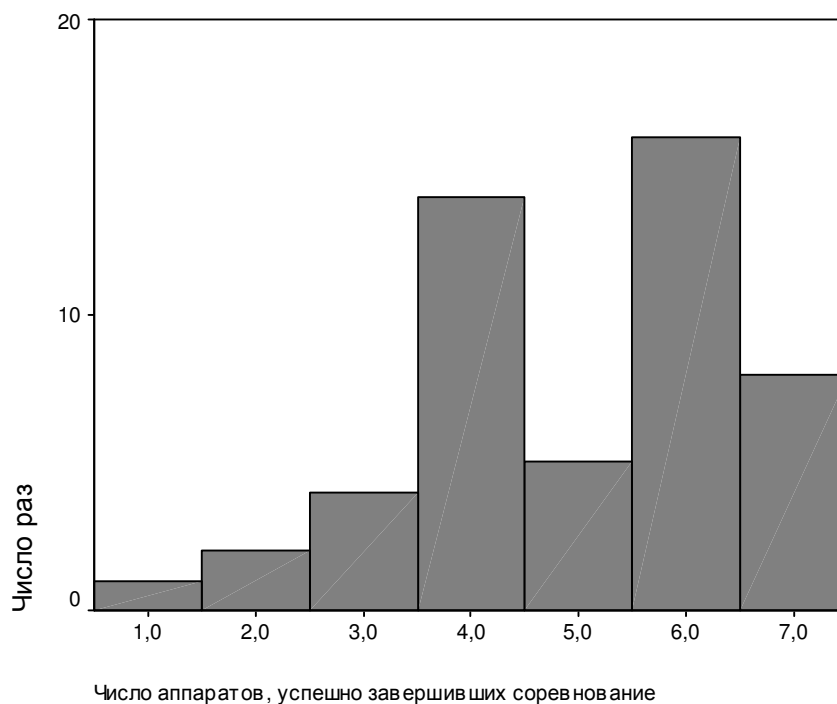


Рис. 67. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода сокращенных таблиц переходов

В табл. 21 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 21. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей		Система управления, построенная с помощью метода сокращенных таблиц переходов	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	204,04	1	201,32	1
Максимум	226,43	6,00	312,89	7,00
Среднее	214,17	3,98	248,86	5,13

Из анализа приведенных результатов следует, что система, построенная с помощью совместного применения генетического программирования, автоматов и нейронных сетей, проигрывает системе, построенной с помощью метода сокращенных таблиц переходов, как по

дальности полета, так и по числу моделей беспилотных летательных аппаратов, успешно завершающих соревнование.

#### 4.3.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода представления автоматов деревьями решений

На рис. 68 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода сокращенных таблиц, в соревновании с командой, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.

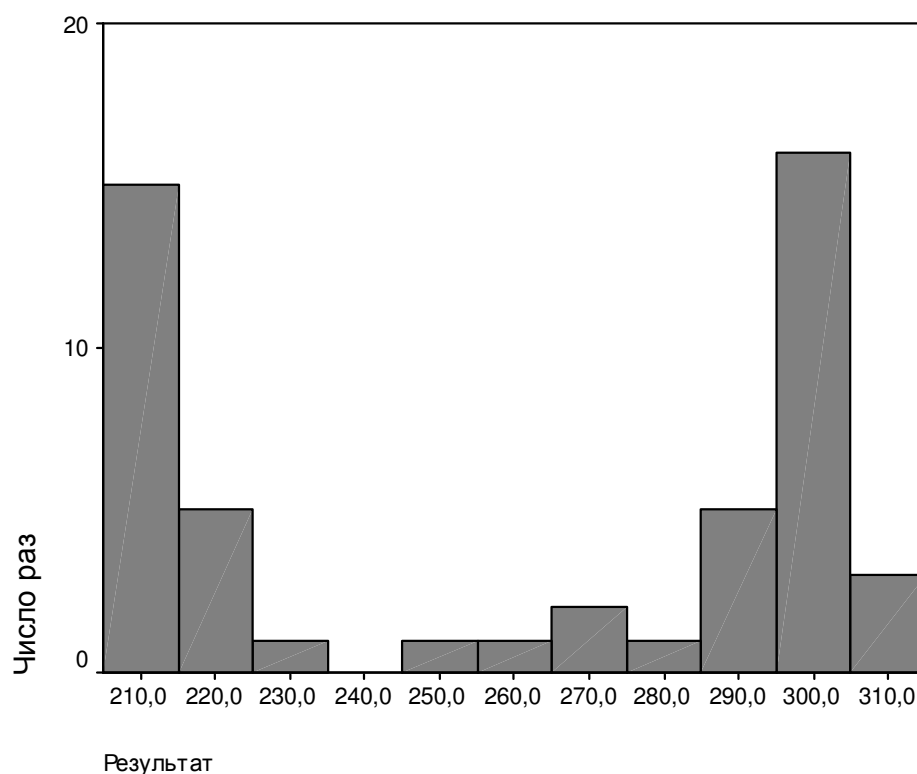


Рис. 68. Распределение результатов, обеспеченных системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 69 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода сокращенных таблиц, с командой, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.

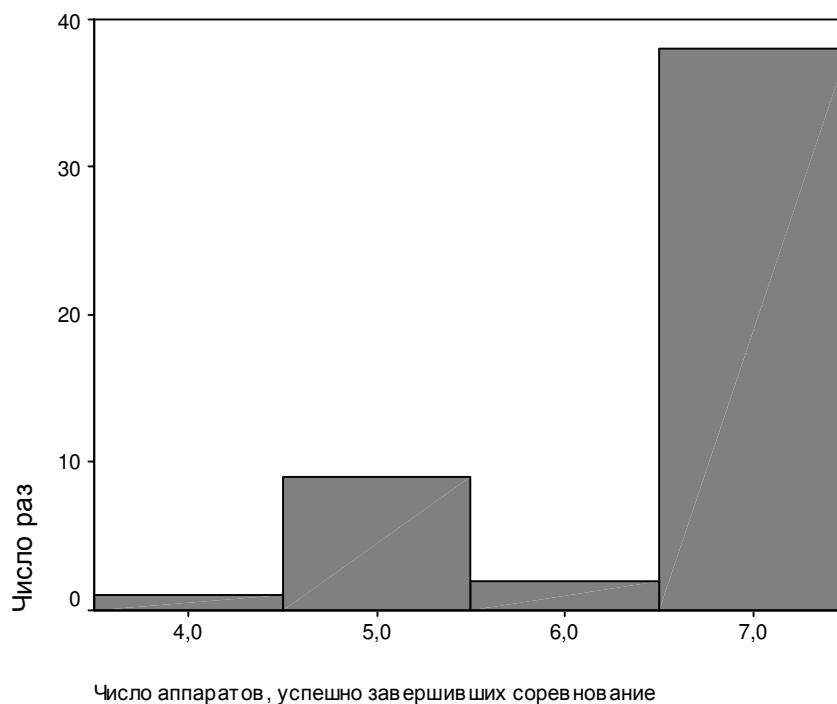


Рис. 69. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода сокращенных таблиц переходов

В табл. 22 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 22. Результаты соревнований

	Система управления, построенная с помощью метода сокращенных таблиц переходов		Система управления, построенная с помощью метода представления автоматов деревьями решений	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	212,23	4	206,02	4
Максимум	312,86	7,00	302,54	7,00
Среднее	261,48	6,53	253,90	6,61

Из анализа приведенных результатов следует, что система, построенная при помощи метода сокращенных таблиц, показывает несколько лучший результат, чем система, построенная с помощью метода представления автоматов деревьями решений. При этом, среднее число моделей аппаратов,

успешно завершивших соревнование, больше у системы управления, построенной с помощью метода представления автоматов деревьями решений.

#### **4.4. РЕЗУЛЬТАТЫ КОМАНДЫ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ, УПРАВЛЯЕМОЙ СИСТЕМОЙ, ПОСТРОЕННОЙ С ПОМОЩЬЮ МЕТОДА ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ДЕРЕВЬЯМИ РЕШЕНИЙ**

В настоящем разделе описаны результаты, показанные командой беспилотных летательных аппаратов, управляемых построенной с помощью метода представления автоматов с помощью деревьев решений системой. Этот метод описан в разд. 3.3, а построенная система управления представлена в разд. 3.3.8.

##### **4.4.1. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной без применения методов автоматизации**

На рис. 70 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода представления автоматов деревьями решений, в соревновании с командой, управляемой системой, построенной без применения методов автоматизации.

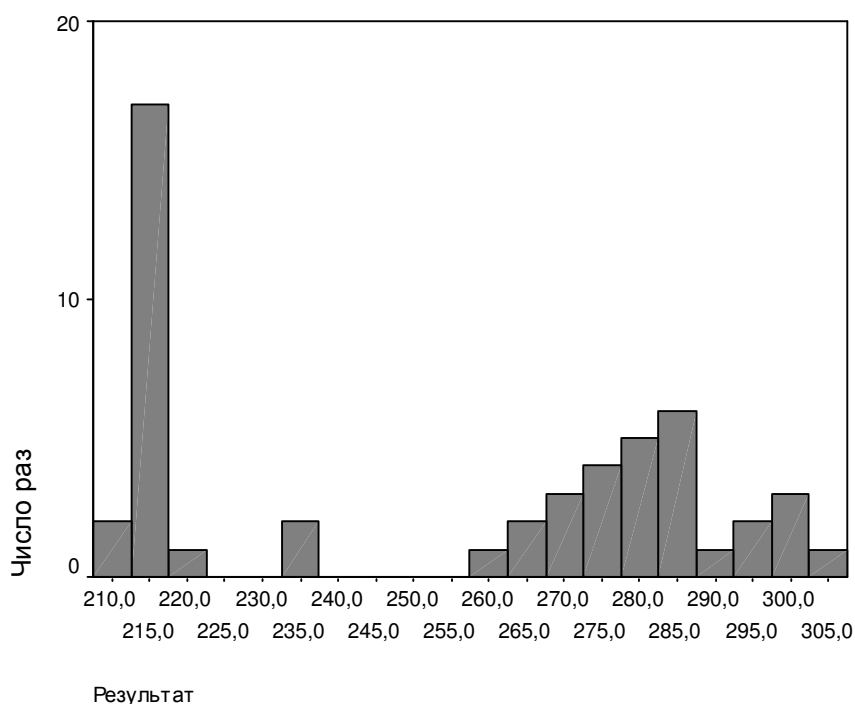


Рис. 70. Распределение результатов, обеспеченных системой, построенной при помощи метода представления автоматов деревьями решений

На рис. 71 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода представления автоматов деревьями решений, с командой, управляемой системой, построенной без применения методов автоматизации.

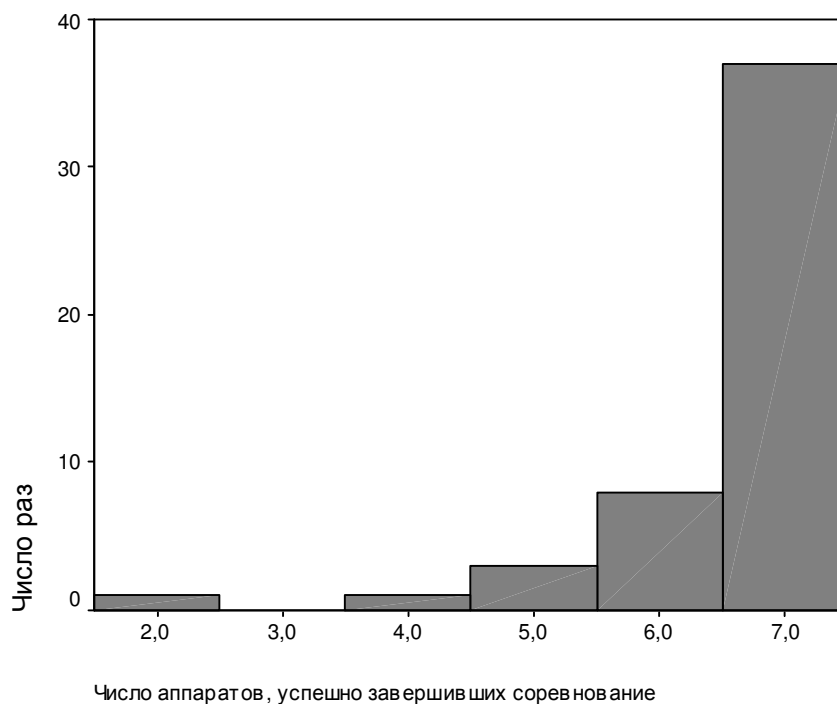


Рис. 71. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода представления автоматов деревьями решений

В табл. 23 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 23. Результаты соревнований

	Система управления, построенная с помощью метода представления автоматов деревьями решений		Система управления, построенная без применения методов автоматизации	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	210,45	2	196,68	4
Максимум	303,43	7,00	310,80	8,00
Среднее	251,61	6,53	214,66	6,72

Из анализа приведенных результатов следует, что система, построенная при помощи метода представления автоматов деревьями решений в среднем более, чем на 20% превосходит систему управления, построенную без применения методов автоматизации.

#### 4.4.2. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, реализующей «агрессивную» стратегию

На рис. 72 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода представления автоматов деревьями решений, в соревновании с командой, управляемой системой, реализующей «агрессивную» стратегию.

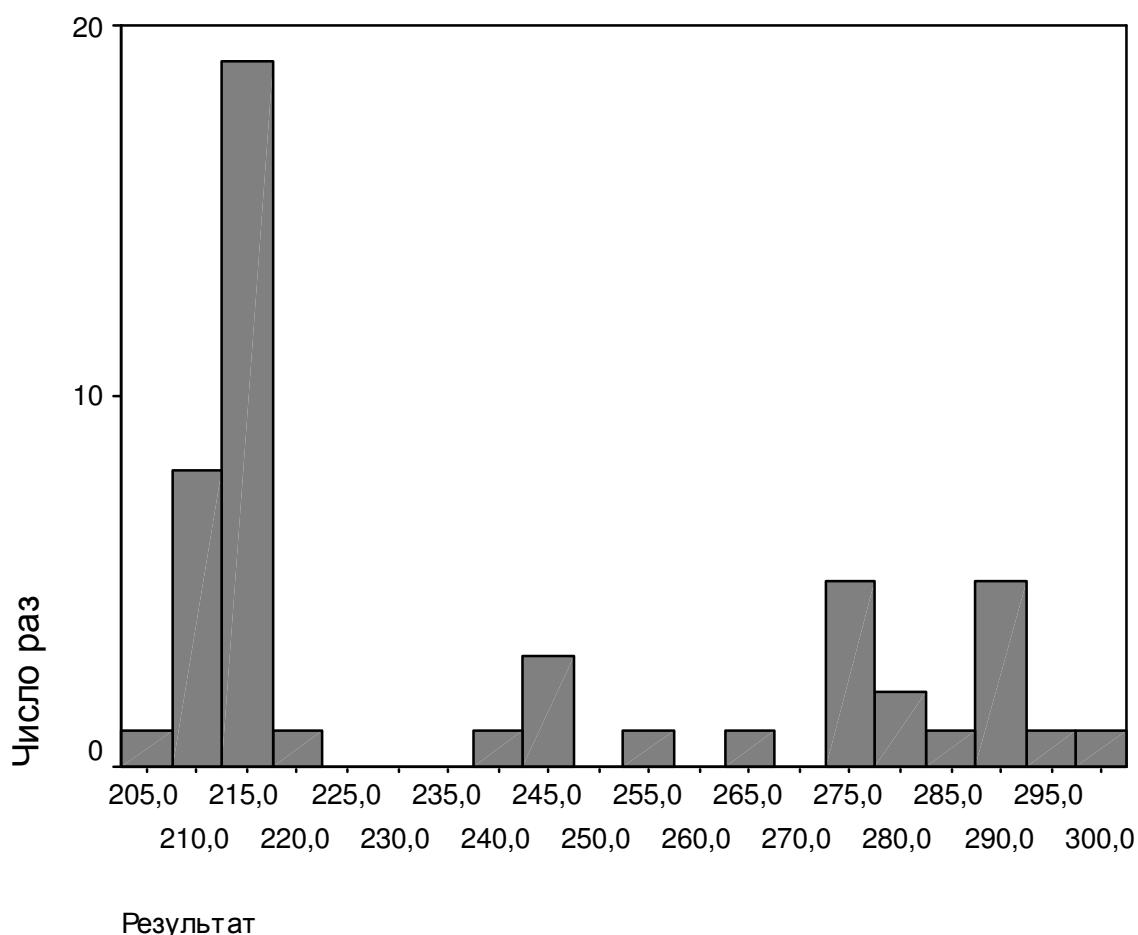


Рис. 72. Распределение результатов, обеспеченных системой, построенной при помощи метода представления автоматов деревьями решений

На рис. 73 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода деревьев решений, с командой, управляемой системой, реализующей «агрессивную» стратегию.

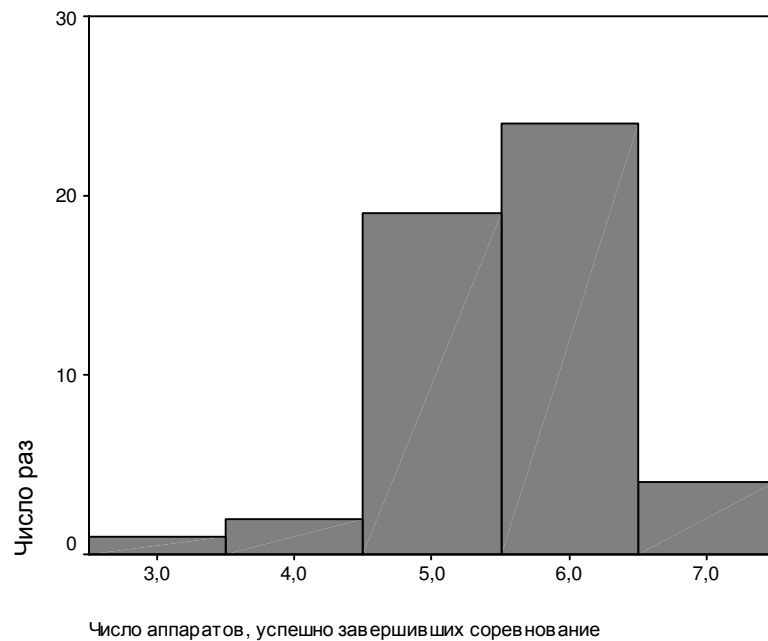


Рис. 73. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода представления автоматов деревьями решений

На рис. 74 показано распределение результатов, которые показала команда, реализующая «агрессивную» стратегию.

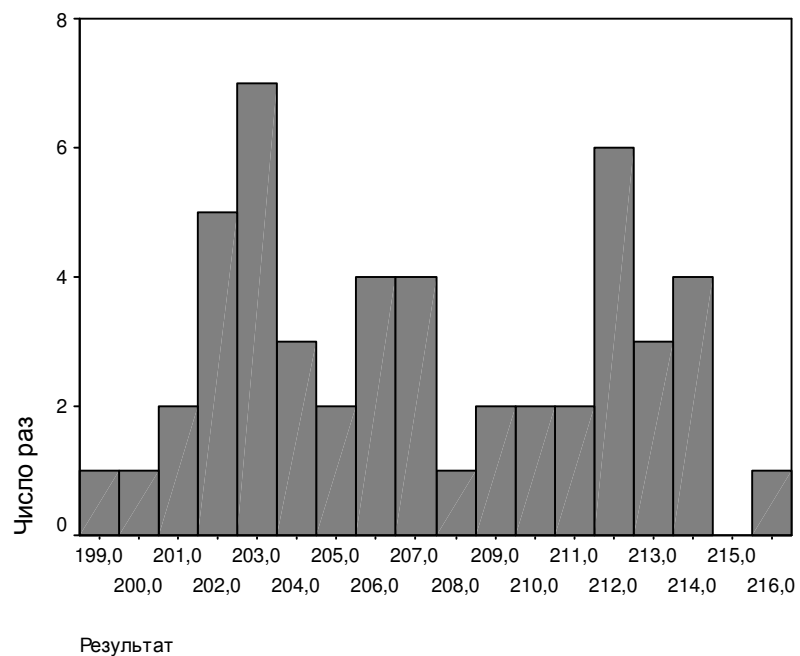


Рис. 74. Распределение результатов, показанных системой, реализующей агрессивную стратегию

Гистограмма распределения числа успешно завершивших соревнование беспилотных летательных аппаратов системы, реализующей «агрессивную» стратегию не приводится, так как при такой стратегии соревнование всегда успешно завершал только один беспилотный летательный аппарат.

В табл. 24 приведены максимальные, минимальные и средние значения результатов, показанных системами управления в проведенных соревнованиях.

Таблица 24. Результаты соревнований

	Система управления, построенная с помощью метода представления автоматов деревьями решений		Система управления, реализующая «агрессивную» стратегию	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	205,14	3	199,33	1
Максимум	302,43	7,00	215,97	1,00
Среднее	238,38	5,56	207,44	1,00

Из анализа приведенных результатов следует, что система, построенная при помощи метода представления автоматов деревьями решений, успешно противостоит системе, реализующей «агрессивную» стратегию. При этом преимущество в среднем составляет 20%, кроме этого, в среднем более пяти моделей беспилотных летательных аппаратов успешно завершают соревнование.

#### **4.4.3. Соревнование с командой беспилотных летательных аппаратов, управляемых системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей**

На рис. 75 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода представления автоматов деревьями решений, в соревновании с командой, управляемой системой, построенной с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей.



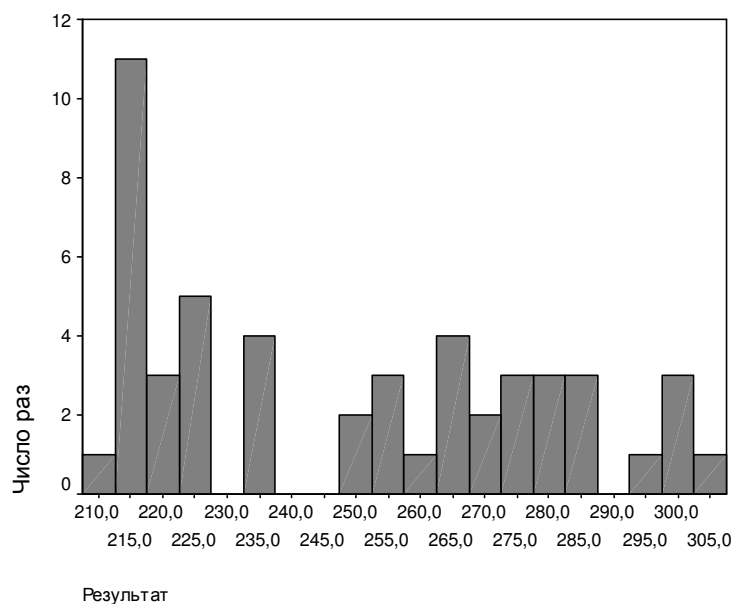


Рис. 75. Распределение результатов, обеспеченных системой, построенной при помощи метода сокращенных таблиц переходов

На рис. 76 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода представления автоматов деревьями решений, с командой, управляемой системой, построенной с помощью метода представления автоматов деревьями решений.

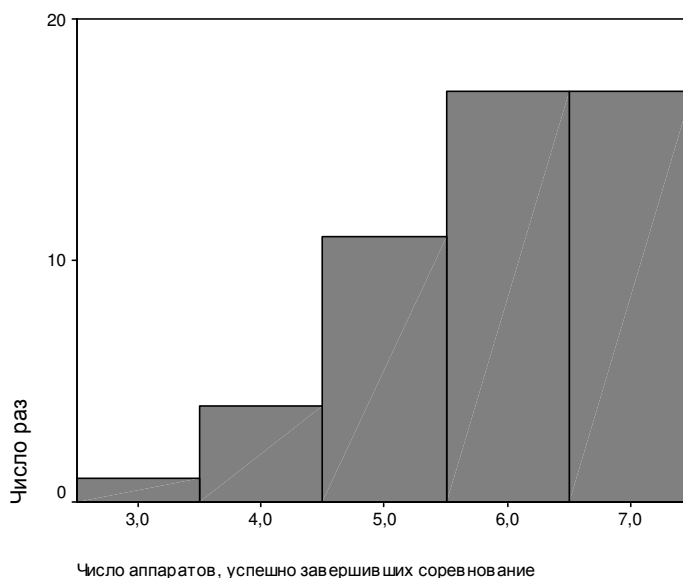


Рис. 76. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода сокращенных таблиц переходов

В табл. 25 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 25. Результаты соревнований

	Система управления, построенная с помощью метода совместного применения генетического программирования, автоматов и нейронных сетей		Система управления, построенная с помощью метода представления автоматов деревьями решений	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	201,36	2	212,17	3
Максимум	228,14	8,00	303,85	7,00
Среднее	215,46	4,00	248,90	5,92

Из анализа приведенных результатов следует, что система, построенная с помощью совместного применения генетического программирования, автоматов и нейронных сетей, проигрывает системе, построенной с помощью метода представления автоматов деревьями решений, как по дальности полета, так и по числу моделей беспилотных летательных аппаратов, успешно завершающих соревнование.

#### **4.4.4. Соревнование с командой беспилотных летательных аппаратов, управляемой системой, построенной с помощью метода сокращенных таблиц переходов**

На рис. 77 показано распределение результатов, которые показала команда беспилотных летательных аппаратов, управляемая системой, построенной с помощью метода представления автоматов деревьями решений, в соревновании с командой, управляемой системой, построенной с помощью метода сокращенных таблиц.

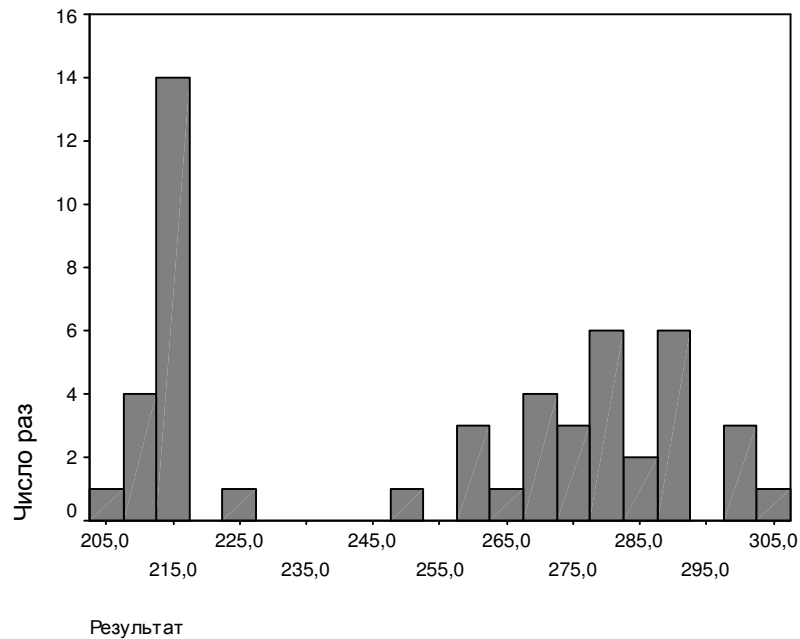


Рис. 77. Распределение результатов, обеспеченных системой, построенной при помощи представления автоматов деревьями решений

На рис. 78 показано распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, которое было получено в результате соревнований команды, управляемой системой, построенной с помощью метода представления автоматов деревьями решений, с командой, управляемой системой, построенной с помощью метода сокращенных таблиц переходов.

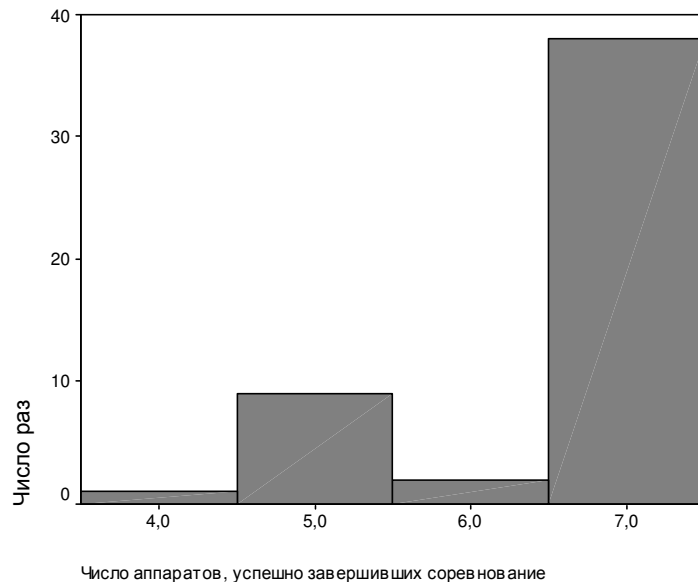


Рис. 78. Распределение числа успешно завершивших соревнование беспилотных летательных аппаратов, управляемых системой, построенной при помощи метода представления автоматов деревьями решений

В табл. 26 приведены максимальные, минимальные и средние значения результатов, показанных командами в проведенных соревнованиях.

Таблица 26. Результаты соревнований

	Система управления, построенная с помощью метода сокращенных таблиц переходов		Система управления, построенная с помощью метода представления автоматов деревьями решений	
	Результат	Число аппаратов, успешно завершивших соревнование	Результат	Число аппаратов, успешно завершивших соревнование
Минимум	212,23	4	206,02	4
Максимум	312,86	7,00	302,54	7,00
Среднее	261,48	6,53	253,90	6,61

Из анализа приведенных результатов следует, что система, построенная при помощи метода сокращенных таблиц, показывает несколько лучший результат, чем система, построенная с помощью метода представления автоматов деревьями решений. При этом, среднее число моделей аппаратов, успешно завершивших соревнование, больше у системы управления, построенной с помощью метода представления автоматов деревьями решений.

#### **4.5. ОБЩАЯ ОЦЕНКА СИСТЕМ УПРАВЛЕНИЯ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ**

В табл. 27 приведены общие результаты оценки в соответствии с критериями, описанными в разд. 1.6, представленных в настоящем отчете систем управления моделью беспилотного летательного аппарата, построенных с помощью описанных методов.

Таблица 27. Оценка построенных систем управления

	Результат	Число аппаратов, успешно завершивших соревнование	Степень автоматизации построения	Временные затраты на построение	Сложность системы управления
Система, построенная без применения методов автоматизации	От 200 до 225, в среднем – 215	От одного до восьми, в среднем – четыре	Низкая – построение системы производится без применения методов автоматизации	Несколько недель	Высокая – семь автоматов, содержащих суммарно более тридцати состояний
Система, построенная с помощью метода совместного применения генетического программирования, нейронных сетей и конечных автоматов	От 200 до 250, в среднем – 220	От одного до восьми, в среднем – четыре	Высокая – автоматизировано построение и автоматов, и входных воздействий	Сутки	Средняя – два конечных автомата из шести состояний и две нейронных сети
Система, построенная с помощью метода представления автоматов деревьями решений	От 200 до 305, в среднем – 235	От трех до семи, в среднем – шесть	Средняя – построение входных воздействий осуществляется вручную	Несколько часов	Низкая – один конечный автомат из шести состояний
Система, построенная с помощью метода сокращенных таблиц переходов	От 210 до 310, в среднем – 240	От трех до семи, в среднем – шесть	Средняя – построение входных воздействий осуществляется вручную	Несколько часов	Низкая – один конечный автомат из шести состояний

#### 4.6. Выводы

Проведена оценка автоматов, построенных на основе методов генерации автоматов управления системами со сложным поведением и их сравнение с системой автоматов, построенных без применения методов автоматизации.

По результатам сравнения, можно сказать, что результаты, полученные предложенными методами, превосходят результаты системы управления, построенной без применения методов автоматизации. При этом наиболее эффективным оказался метод сокращенных таблиц переходов, а метод совместного применения автоматов и нейронных сетей и метод представления автоматов при помощи деревьев решений оказались несколько менее эффективными.

## **ЗАКЛЮЧЕНИЕ**

В результате исследований, выполненных на третьем этапе работ по контракту, была проведена апробация и экспериментальное исследование методов генетического программирования для генерации автоматов управления системами со сложным поведением. Была доказана применимость этих методов на практике и их эффективность при генерации систем управления объектами со сложным поведением.

В первой главе была разработана тестовая модель объекта со сложным поведением — модель беспилотного летательного аппарата. Также был разработан набор критериев для оценки систем управления.

Во второй главе приведена система управления, построенная вручную и применяемая в качестве точки отсчета при оценке других решений.

В третьей главе описываются системы управления, построенные на основе методов, разработанных на втором этапе работ.

В четвертой главе приведены результаты оценки методов генерации автоматов управления системами со сложным поведением. При этом наиболее эффективным оказался метод сокращенных таблиц переходов.

Таким образом, были получены решения всех задач, поставленных в техническом задании на проведение третьего этапа работы.

Результаты выполненных работ, а также патентных исследований, позволяют утверждать, что научно-технический уровень исследований соответствует уровню исследований в рассматриваемой области, проводимых в лучших исследовательских центрах мира.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Angeline P., Pollack J.* Evolutionary Module Acquisition / Proceedings of the Second Annual Conference on Evolutionary Programming. Cambridge: MIT Press. 1993, pp.154–163. <http://www.demo.cs.brandeis.edu/papers/ep93.pdf>
2. *Belz A., Eskikaya B.* A Genetic Algorithm for Finite State Automata Induction with Application to Phonotactics / Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing. Saarbruecken. 1998, pp. 9–17. [http://www.itri.brighton.ac.uk/~Anja.Belz/Publications/A\\_GA\\_for\\_FSA\\_induction\\_with\\_an\\_applicati\\_on\\_to\\_phonotactics.ps](http://www.itri.brighton.ac.uk/~Anja.Belz/Publications/A_GA_for_FSA_induction_with_an_applicati_on_to_phonotactics.ps)
3. *Chambers L.* Practical Handbook of Genetic Algorithms. Complex Coding Systems. Volumes I, II, III. CRC Press, 1999.
4. *Fogel L.* Autonomous Automata // Industrial Research. 1962. V.4, pp. 14–19.
5. *Fogel L., Owens A., Walsh M.* Artificial Intelligence through Simulated Evolution. NY: Wiley. 1966.
6. *Frey C., Leugering G.* Evolving Strategies for Global Optimization – A Finite State Machine Approach / Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001). Morgan Kaufmann. 2001, pp. 27–33. <http://citeseer.ist.psu.edu/456373.html>
7. *Holland J.* Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
8. <http://unimod.sourceforge.net>
9. *Koza J.* Genetic Evolution and Co-Evolution of Computer Programs / Proceedings of Second Conference on Artificial Life. Redwood City, CA: Addison-Wesley. 1992. pp. 603–629. <http://citeseer.ist.psu.edu/177879.html>
10. *Koza J.* Genetic programming. On the Programming of Computers by Means of Natural Selection. MA: The MIT Press, 1998.
11. *Lucas M., Reynolds T.J.* Learning DFA: Evolution versus Evidence Driven State Merging.
12. *MacLennan B.* Synthetic Ethology: An Approach to the Study of Communication / Proceedings of Artificial Life II: The Second Workshop on the Synthesis and Simulation of Living. CA: Addison Wesley. 1992. V. X, pp. 631–658. <http://www.cs.utk.edu/~mclennan/papers/SEASC.pdf>
13. *McCulloch W. S., Pitts W.* A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics, 1943, 5, pp. 115–137.
14. *Mitchell M.* An Introduction to Genetic Algorithms. MA: The MIT Press, 1996.
15. *Naidoo A., Pillay N.* The Induction of Finite Transducers Using Genetic Programming / Proceedings of Euro GP. Springer. 2007. <http://saturn.cs.unp.ac.za/~nelishiap/papers/eurogp07.pdf>
16. *Nedjah N., Mourelle L.* Mealy Finite State Machines: An Evolutionary Approach // International Journal of Innovative Computing, Information and Control. 2006. V.2, I. 4.
17. *Воронин О., Дьюдни А.* Дарвинизм в программировании // Мой компьютер. 2004. № 35. <http://www.mycomp.kiev.ua/text/7458>
18. *Гладков Л. А., Курейчик В. В., Курейчик В. М.* Генетические алгоритмы. М.: Физматлит. 2006.
19. *Гуров В. С., Мазин М. А., Нарвский А. С., Шальто А. А.* UML. SWITCH-технология. Eclipse // Информационно-управляющие системы, 2004, № 6, с.12–17. <http://is.ifmo.ru/works/uml-switch-eclipse/>
20. *Данилов В. Р.* Технология генетического программирования для генерации автоматов управления системами со сложным поведением. СПбГУ ИТМО. 2007. Бакалаврская работа. [http://is.ifmo.ru/papers/danilov\\_bachelor/](http://is.ifmo.ru/papers/danilov_bachelor/)
21. Заочный тур всесибирской олимпиады 2005 по информатике. <http://olimpic.nsu.ru/widesiberia/archive/wso6/2005/rus/1tour/problem/problem.html>
22. *Кизимов А. Т., Фролова Л. Е.* Стабилизация малоразмерного беспилотного летательного аппарата в турбулентной атмосфере // Информационно-измерительные и управляющие системы. № 12, 2007 г.

23. Лобанов П. Г., Шалыто А. А. Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о “Флибах” / Сборник докладов 4-й Всероссийской научной конференции «Управление и информационные технологии» (УИТ-2006). СПбГЭТУ «ЛЭТИ». 2006, с.144–149. <http://is.ifmo.ru/works/flib>
24. Непейвода Н. Н. Стили и методы программирования. М.: Интернет-Университет Информационных технологий, 2005.
25. Николенко С. И. Лекции по генетическим алгоритмам. <http://logic.pdmi.ras.ru/~sergey/teaching/ml/>
26. Паращенко Д. А., Царев Ф. Н., Шалыто А. А. Применение автоматного программирования при моделировании одного класса мультиагентных систем / Материалы девятой международной конференции «Интеллектуальные системы и компьютерные науки». М.: МГУ. 2006. Т.2, с. 352–355.
27. Паращенко Д.А., Царев Ф.Н., Шалыто А.А. Технология моделирования одного класса мультиагентных систем на основе автоматного программирования на примере игры «Соревнование летающих тарелок». Проектная документация. СПбГУ ИТМО. 2006. <http://is.ifmo.ru/unimod-projects/plates/>
28. Поликарпова Н. И., Точилин В. Н. Применение генетического программирования для реализации систем со сложным поведением // Научно-технический вестник СПбГУ ИТМО. Выпуск 39, с.276–293. [http://vestnik.ifmo.ru/ntv/39/ntv\\_39.3.3.pdf](http://vestnik.ifmo.ru/ntv/39/ntv_39.3.3.pdf)
29. Промежуточный отчет по этапу II «Теоретические исследования поставленных перед НИР задач». СПбГУ ИТМО. 2007. [http://is.ifmo.ru/genalg/2007\\_02\\_report-genetic.pdf](http://is.ifmo.ru/genalg/2007_02_report-genetic.pdf)
30. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. М.: Вильямс. 2006.
31. Шалыто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления СПб.: Наука, 1998. <http://is.ifmo.ru/books/switch/1>