

УДК 004.4'242

СОВМЕСТНОЕ ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ, КОНЕЧНЫХ АВТОМАТОВ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ БЕСПИЛОТНЫМ ЛЕТАТЕЛЬНЫМ АППАРАТОМ

Ф.Н. Царев

(Санкт-Петербургский государственный университет информационных технологий, механики и оптики)

В работе рассматривается применение генетических алгоритмов для построения системы управления беспилотным летательным аппаратом. Система управления строится как совокупность искусственной нейронной сети и конечного автомата. Нейронная сеть преобразует входные вещественные переменные в логические, которые подаются на вход конечному автомату. Он, в свою очередь, вырабатывает выходные воздействия. Для оптимизации этой модели используется генетическое программирование.

Ключевые слова: генетическое программирование, нейронные сети, конечный автомат, беспилотный летательный аппарат, автоматное программирование

Введение

В последнее время все чаще применяется автоматное программирование [1], в рамках которого поведение программ описывается с помощью конечных детерминированных автоматов. В ряде задач автомат удается построить эвристическими методами, однако часто такое построение требует больших затрат времени или вообще невозможно. Примером такой задачи является управление командой беспилотных летательных аппаратов [2, 3] в соревнованиях с другой командой. Полный перебор крайне трудоемок, а эвристическое построение не всегда дает приемлемые результаты. Поэтому для построения автоматов в задачах такого рода целесообразно применять генетические алгоритмы и генетическое программирование [4–10].

С другой стороны, большая часть известных автору методов построения автоматов с помощью генетических алгоритмов [11–13] пригодна только для случая логических входных переменных. 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 автомату, необходимо определить, 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 настоящей работе применяется второй подход. В качестве классификатора используется нейронная сеть. Ее настройка и построение автомата производится совместно с помощью генетического программирования.

Целью настоящей работы является построение с помощью генетического программирования системы управления беспилотным летательным аппаратом, состоящей из конечного автомата и нейронной сети.

Постановка задачи

Проводится соревнование [2, 3] 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 задана начальная скорость и начальное направление движения. В простейшем случае начальные скорости всех аппаратов одинаковы, а направления – строго вперед. 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*.

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).

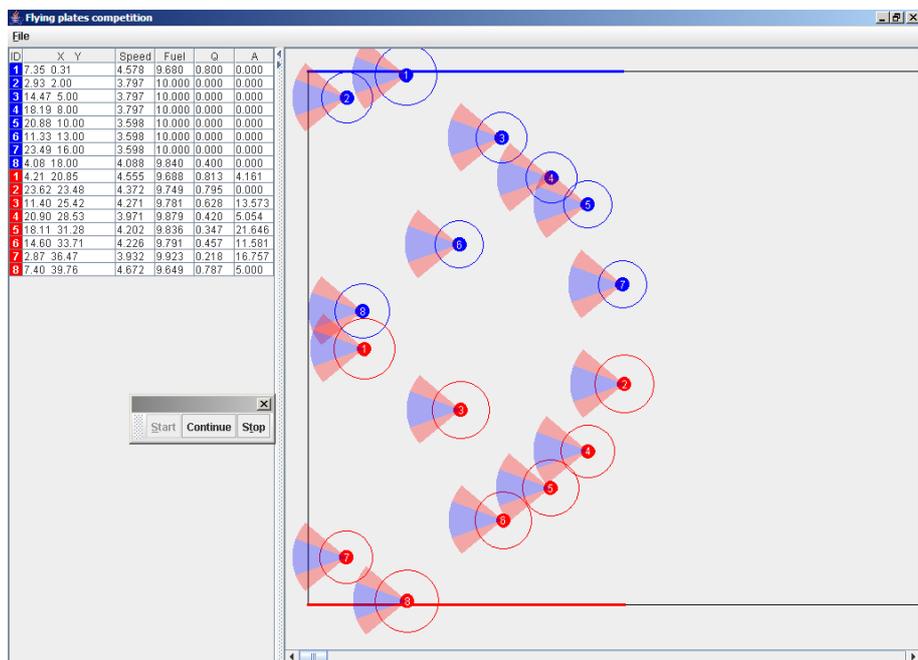


Рис. 1. Беспилотные летательные аппараты на старте

Жизненный цикл беспилотного летательного аппарата в рассматриваемой задаче может быть описан графом переходов автомата с тремя состояниями: «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

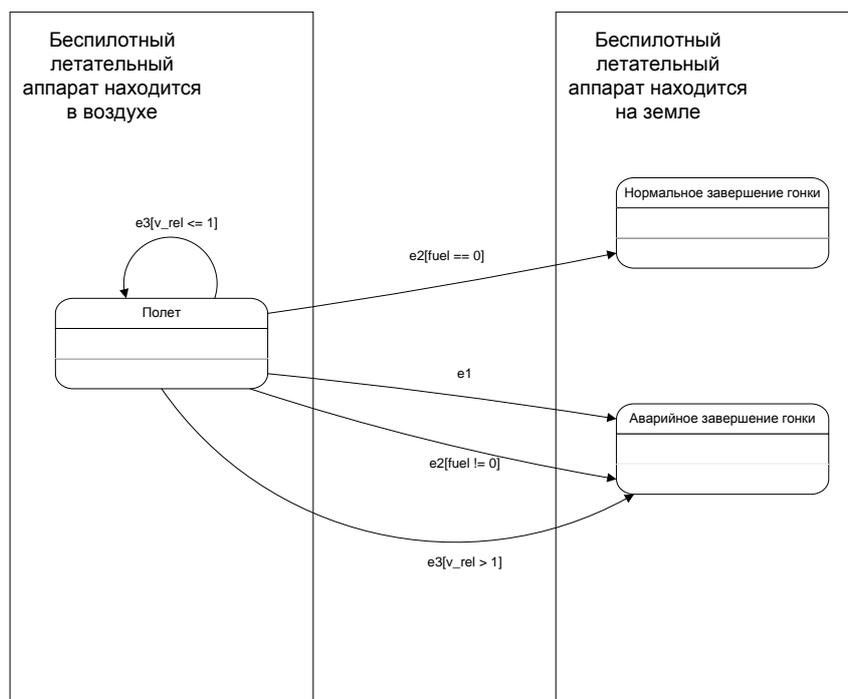


Рис. 2. Возможные состояния беспилотного летательного аппарата и переходы между ними

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 1 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 1 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 1 м/с (условие $v_{rel} > 1$), он аварийно завершает гонку. При относительной скорости столкновения, не превышающей 1 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.

Динамика беспилотного летательного аппарата

Беспилотный летательный аппарат представляет собой дискообразное «летающее крыло» радиусом 1 м. На рис. 3 представлен вид сверху описываемого летательного аппарата.

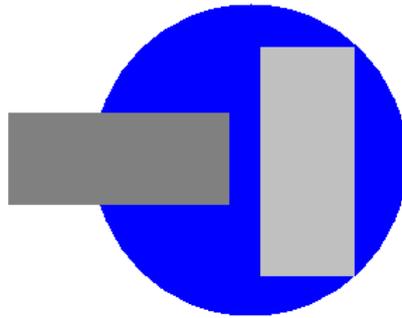


Рис. 3. Беспилотный летательный аппарат

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 1 м/с. Его максимальная скорость зависит от запаса топлива и сопротивления воздуха. Ограничение в 1 м/с 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 летательный 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 [см³/с]. 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°.

Аэродинамическое взаимодействие между летательными аппаратами

При полете беспилотного летательного аппарата от траектории его полета в направлениях назад и в стороны под углом около 30° распространяются конические вихревые потоки воздуха. Если другой аппарат попадет в этот вихрь, то сопротивление воздуха его полету резко *снизится* (рис. 4). Отметим, что летательный аппарат, находящийся за хвостом

(два сектора по 20°) другого беспилотного летательного аппарата, испытывает *дополнительное сопротивление* движению, обусловленное реактивной струей.

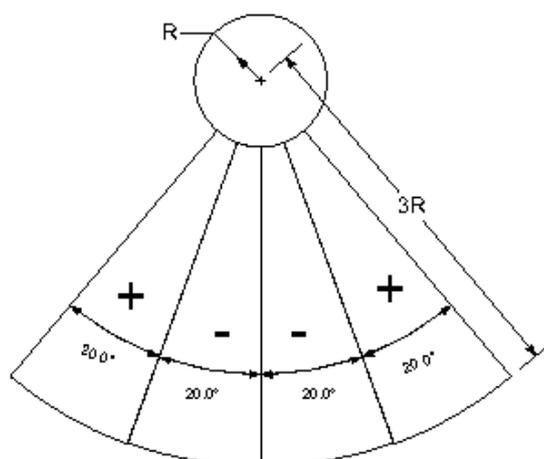


Рис. 4. Зоны повышенного и пониженного сопротивления воздуха

Поясним, как учитывается изменение сопротивления воздуха. Если центр второго летательного аппарата находится в областях, отмеченных на рис. 4 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.

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 ΔN отрицательно, то сопротивление в этой области повышается на $50|\Delta N|$ процентов.

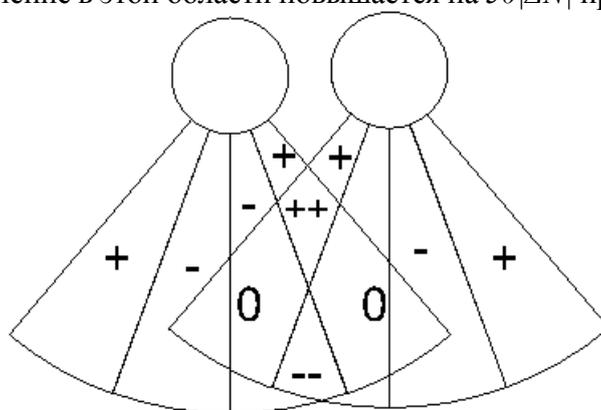


Рис. 5. Наложение областей аэродинамического взаимодействия двух летательных аппаратов

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 1 м/с, то оба участвовавшие в столкновении беспилотных летательных аппарата повреждаются и аварийно завершают гонку. В рамках рассматриваемой модели поврежденные летательные 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.

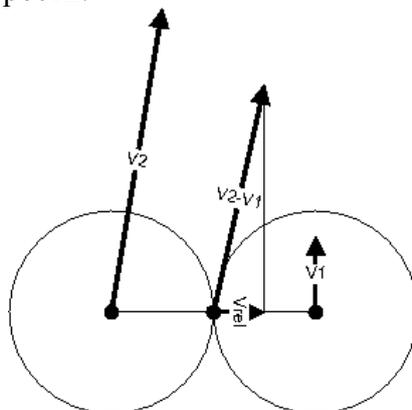


Рис. 6. Относительная скорость столкновения двух летательных аппаратов

Моделирование гонки

Моделирование гонки происходит по ходам, каждый из которых занимает t миллисекунд. 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 1 м/с. 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 1 м/с. 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, т.е. расстояние между центрами каких-либо двух аппаратов становится меньше 2 м, то их скорости и координаты изменяются в соответствии с законами сохранения импульса и энергии. При этом летательные аппараты, относительная скорость столкновения которых превосходила 1 м/с, выбывают из гонки.

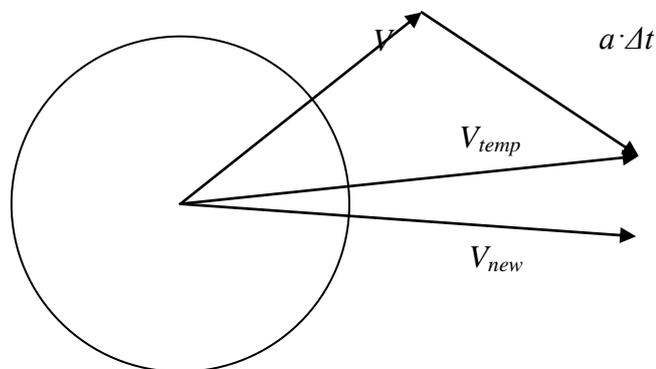


Рис. 7. Пересчет скорости летательного аппарата на шаге моделирования

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.

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* [4] – 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. 8).

Структура нейронной сети и способ ее взаимодействия с конечным автоматом показаны на рис. 9. Символами *S* на рис. 9 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.

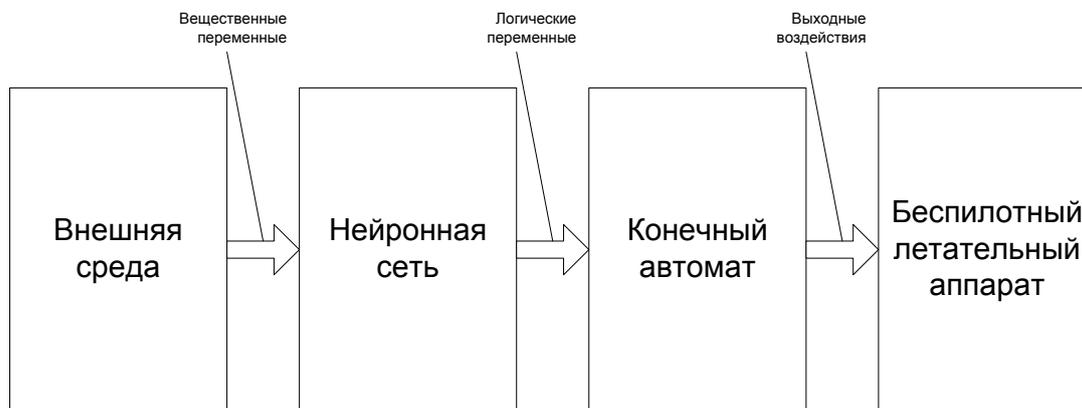


Рис. 8. Структурная схема системы управления беспилотным летательным аппаратом

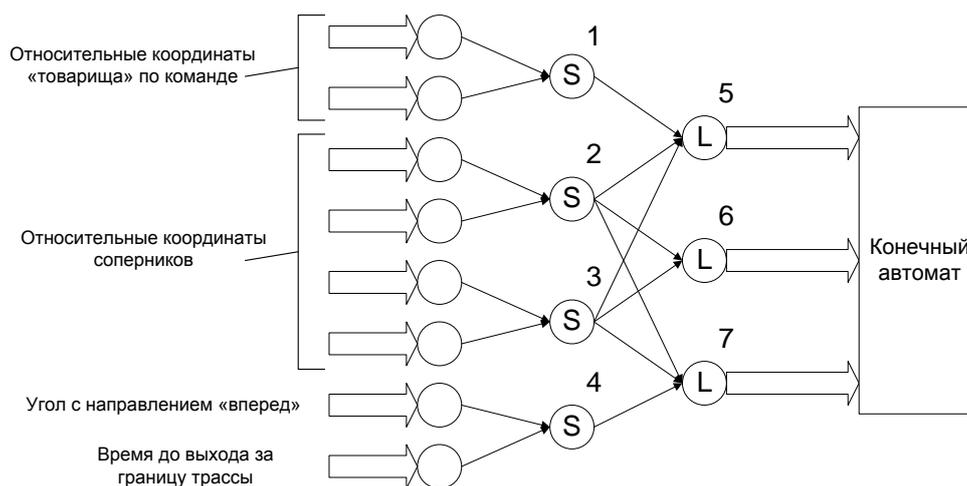


Рис. 9. Нейронная сеть и ее взаимодействие с конечным автоматом

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).

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, ука-

занная 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.

Таким образом, особь в описываемом алгоритме генетического программирования состоит из двух систем управления беспилотным летательным аппаратом (рис. 10).

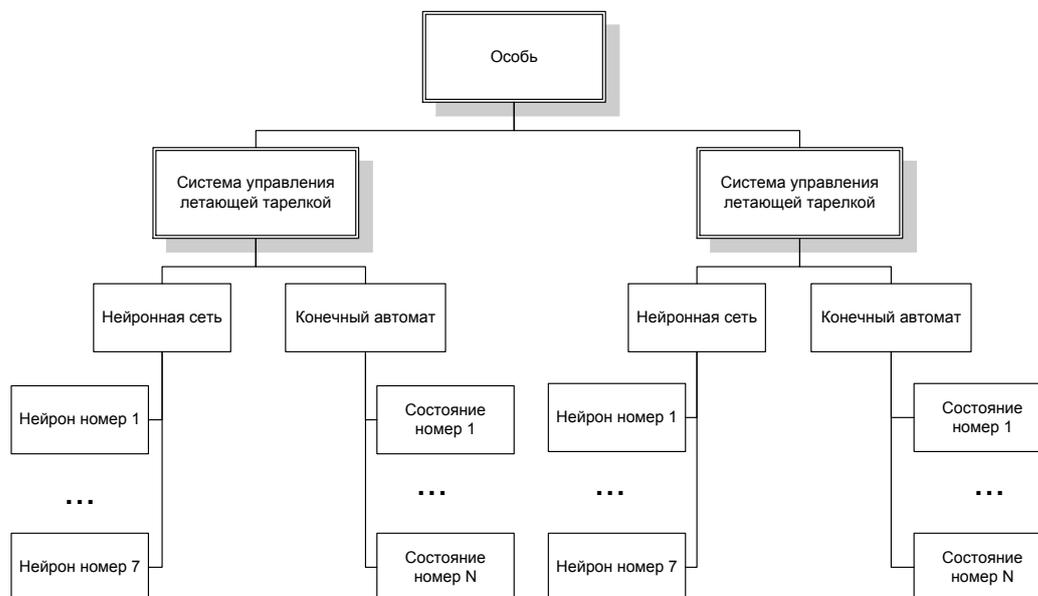


Рис. 10. Структура особи

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;
}
  
```

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 -1 до 1.

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

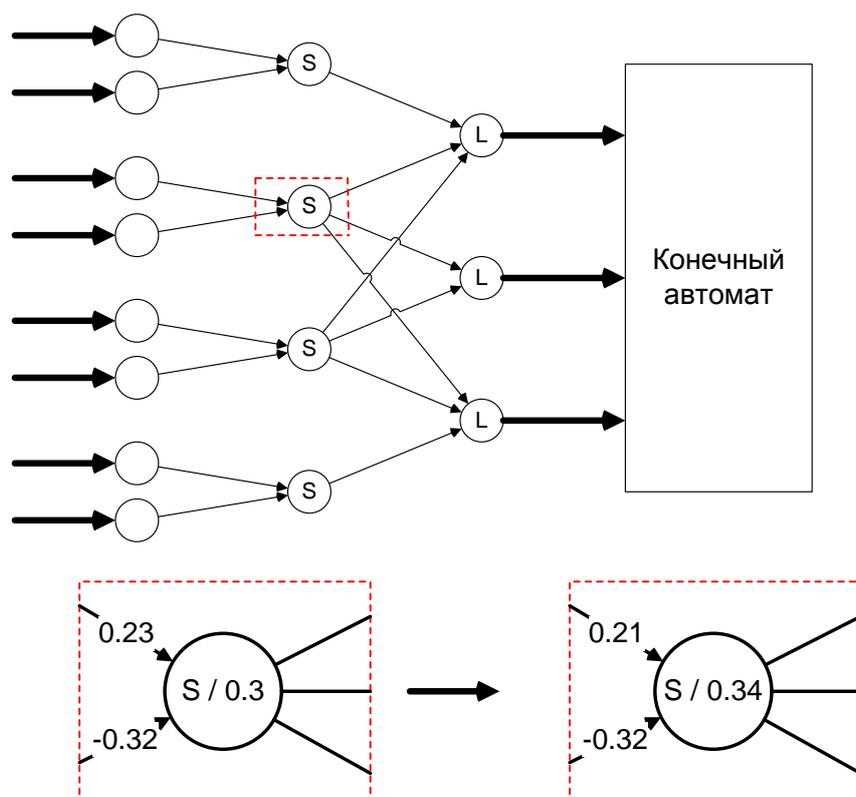


Рис. 11. Мутация нейронной сети

Мутация нейронной сети. При мутации нейронной сети случайно и равновероятно выбирается один элемент (искусственный нейрон) сети и мутирует. Мутация нейронной сети проиллюстрирована на рис. 11.

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 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°).

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: $S1.s1 = s11, S1.s2 = s12, S2.s1 = s21, S2.s2 = s22$.

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. 12. 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 резуль-

тате 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.

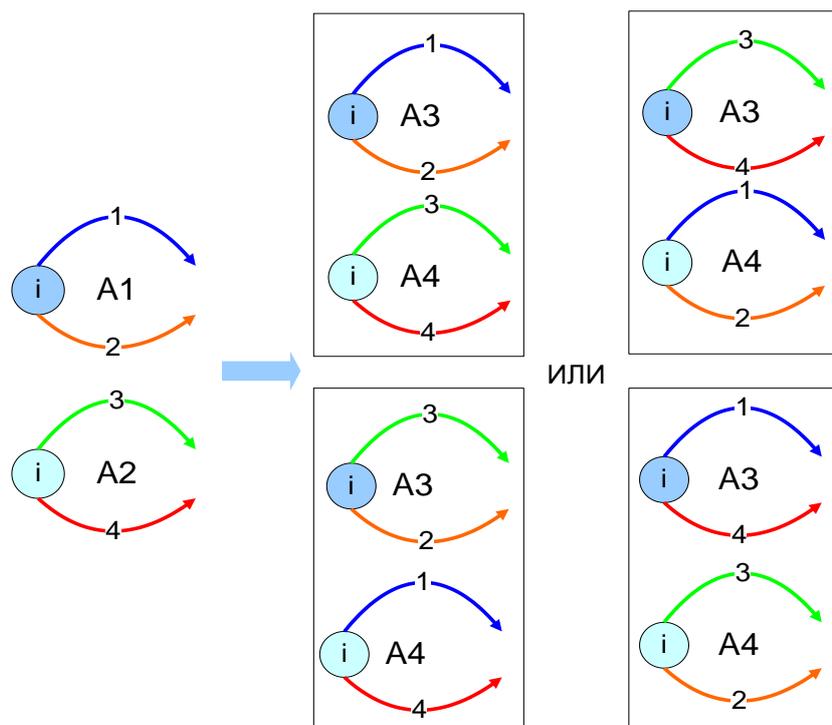


Рис. 12. Варианты переходов при скрещивании

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. 9). 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. 13.

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.

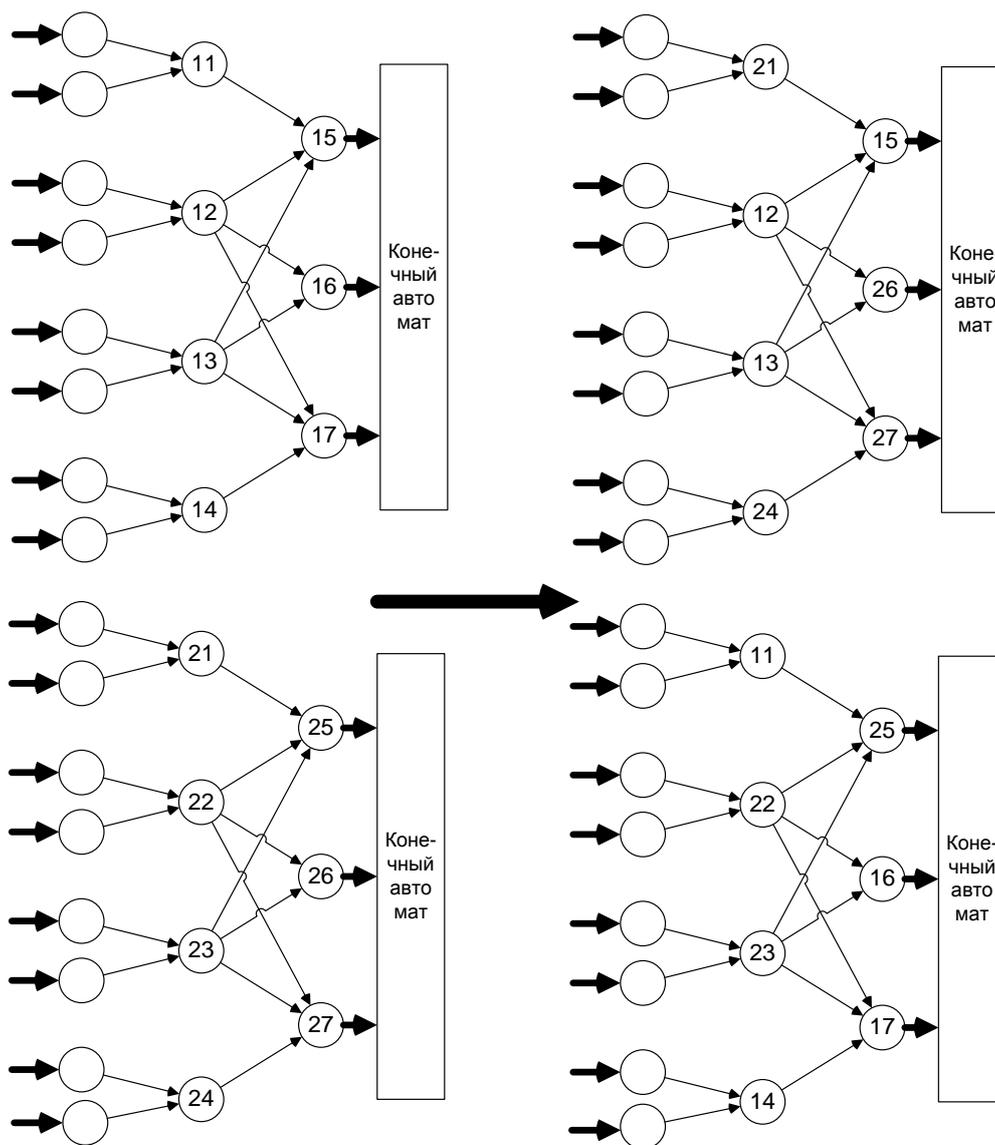


Рис. 13. Скрещивание нейронных сетей

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 [3]. Проводилось по пять соревнований с каждой из стратегий при следующих параметрах летательных аппаратов:

$$c1 = 0.625; \tag{5}$$

$$c2 = 0.025; \tag{6}$$

$$c4 = 3.125; \tag{7}$$

$$\Delta t = 0.3; \tag{8}$$

$$L = 7. \tag{9}$$

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.

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 [3]. Тестирование проводилось с помощью соревнования построенной системы с командой, аппараты которой управляются системой, также описанной в работе [3]. Соревнования проводились при числе летательных аппаратов в каждой команде, равном 8. Чтобы построенные с помощью генетического программирования системы управления беспилотным летательным аппаратом могли работать в этом случае, на первые два входа нейронной сети (рис. 9) 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*.

Таблица 2. 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 0,4	-10 0,8	10 0,8	-15 0,4	-5 0,8
5	2	1	2	3	1	4	2	2
	-25 0,7 8	-15 0,4	-20 0,4	10 0,4	-20 0,8	-5 0,8	0 0,8	-20 0,4

Функция переходов и действий автомата, входящего в систему управления первым беспилотным летательным аппаратом, приведена в табл. 2. 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. Ячейки имеют формат, показанный в табл. 3: в верхней строке указано новое состояние, а в нижней – выполняемое действие. В табл. 4 приведены веса нейронной сети, входящей в построенную систему управления первым беспилотным летательным аппаратом.

Таблица 3. Формат ячеек табл. 2

Новое состояние	
Изменение угла поворота	Новый расход топлива

Таблица 4. 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	<i>0,7820135561918911</i>	<i>0,0000823910892</i>
	Вход-2	<i>0,9843639098567366</i>	
2	Вход-3	<i>0,8621015893426324</i>	<i>0,0000342189215</i>
	Вход-4	<i>0,49004984673483654</i>	
3	Вход-5	<i>0,31478285017906643</i>	<i>0,0000238490014</i>
	Вход-6	<i>0,9111794712756234</i>	
4	Вход-7	<i>0,9383238994033571</i>	<i>0,0000341289013</i>
	Вход-8	<i>0,5244096391670159</i>	
5	1	<i>-0,9357005374202394</i>	<i>0,4971125943935544</i>
	2	<i>-0,4004385439577506</i>	
	3	<i>0,97490945048913</i>	
6	2	<i>0,9562669142389417</i>	<i>-0,568426590601512</i>
	3	<i>-0,2903272439699007</i>	
7	2	<i>0,1400255181142766</i>	<i>-0,859430022111886</i>
	3	<i>-0,315932752555524</i>	
	4	<i>-0,5112115605370626</i>	

Таблица 5. 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 0,3 8	-15 0,4	0 0,8	-15 0,3 6
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 0,3 5	0 0,4	0 0,8

Таблица 6. 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	

В табл. 5 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. 2. В табл. 6 приведены веса связей нейронной сети, входящей в построенную систему управления вторым беспилотным летательным аппаратом.

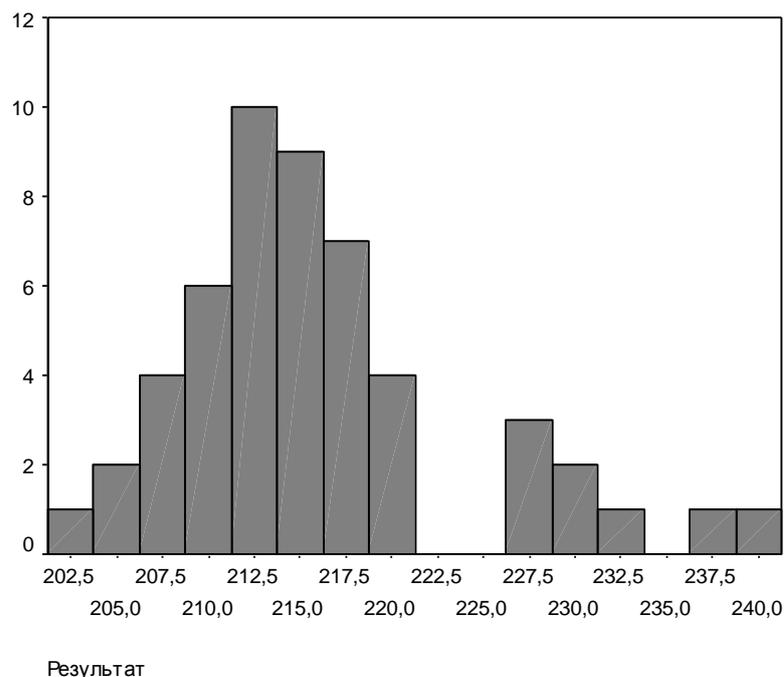


Рис. 14. Распределение результатов, обеспеченных системой, построенной при помощи генетического программирования

Приведем результат сравнения построенной с помощью генетического алгоритма системы управления с системой, построенной вручную в работе [3]. На рис. 14 показано распределение результатов, обеспеченных системой управления, построенной в настоящей работе с помощью генетического программирования. На рис. 15 показано распределение результатов, обеспеченных системой управления, построенной в работе [3].

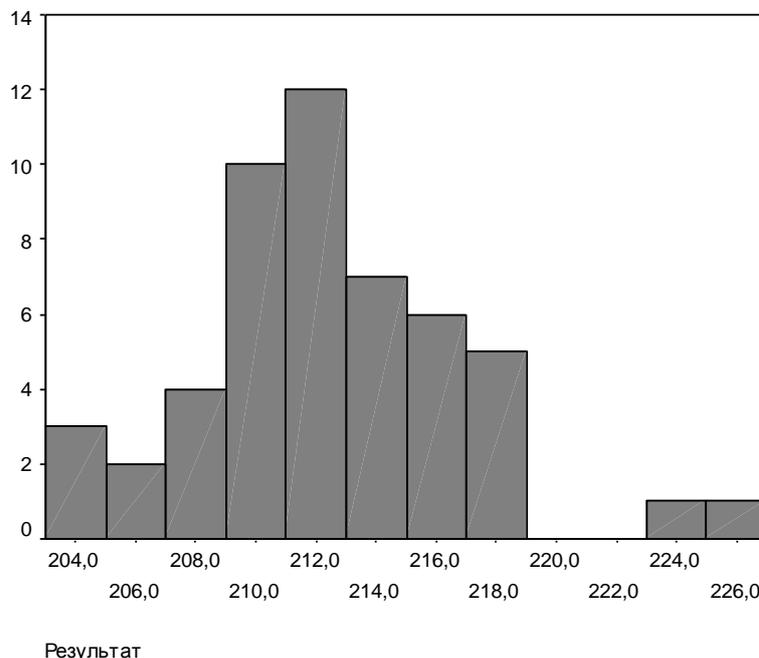


Рис. 15. Распределение результатов, обеспеченных системой, построенной в работе [3]

Выводы

В работе предложен алгоритм генетического программирования, осуществляющий построение системы управления беспилотным летательным аппаратом. Система управления строится как совокупность нейронной сети и конечного автомата. Нейронная сеть служит для преобразования вещественных входных переменных от внешней среды в логические переменные. Эти переменные подаются на вход конечного автомата, который вырабатывает выходные воздействия.

Проведено сравнение системы управления с системой автоматов, построенной в работе [3]. Результаты этого сравнения позволяют сделать вывод о том, что с помощью генетического программирования была построена система управления, превосходящая по результатам соревнований построенную вручную в работе [3]. Это доказывает эффективность используемой модели и метода ее оптимизации.

Литература

1. Шалыто А.А. Технология автоматного программирования // Труды первой Всероссийской научной конференции «Методы и средства обработки информации». – М.: МГУ. – Режим доступа: 2003. http://is.ifmo.ru/works/tech_aut_prog/
2. Заочный тур всесибирской олимпиады 2005 по информатике. – Режим доступа: <http://olimpic.nsu.ru/widesiberia/archive/wso6/2005/rus/1tour/problem/problem.html>

3. Парашенко 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. – Режим доступа: <http://is.ifmo.ru/unimod-projects/plates/>
4. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. – М.: Вильямс, 2006.
5. Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A. The Genesys System. 1992. – Режим доступа: www.cs.ucla.edu/~dyer/Papers/AlifeTracker/Alife91Jefferson.html
6. Angeline P. J., Pollack J. Evolutionary Module Acquisition // Proceedings of the Second Annual Conference on Evolutionary Programming. 1993. – Режим доступа: <http://www.demon.cs.brandeis.edu/papers/ep93.pdf>
7. Chambers L. Practical Handbook of Genetic Algorithms. Complex Coding Systems. Volume III. CRC Press, 1999.
8. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2006.
9. Koza J. R. Genetic programming: on the programming of computers by means of natural selection. – MIT Press, 1992.
10. Яминов Б. Генетические алгоритмы – Режим доступа: <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
11. Царев Ф. Н., Шалыто А. А. О построении автоматов с минимальным числом состояний для задачи об "умном муравье" // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПбГЭТУ "ЛЭТИ". – Т.2. – 2007. – С. 88–91. http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf
12. Данилов В.Р. Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Бакалаврская работа / СПбГУ ИТМО. – 2007.. – Режим доступа: http://is.ifmo.ru/papers/danilov_bachelor/
13. Поликарпова Н.И., Точилин В.Н. Применение генетического программирования для реализации систем со сложным поведением // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 39. – С.276–293.