

## СРЕДСТВА ДЕМОНСТРАЦИИ ЭВОЛЮЦИОННЫХ МЕТОДОВ НАСТРОЙКИ ВЕСОВЫХ КОЭФФИЦИЕНТОВ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Д. О. Петров, М. Ю. Стасюкевич

Брестский государственный технический университет, Брест, Беларусь

*The problems of developing neurocontrollers for controlling dynamic objects are described, including the complexity of generating training data sets. It is indicated that one of the known methods of training an artificial neural network to control an autonomous driving agent is the neuroevolutionary approach, which involves the use of a genetic algorithm to adjust the synaptic weighting coefficients of the artificial neural network. The idea of using a tool for demonstrating the evolutionary approach to setting the weighting coefficients of an artificial neural network for practical training of students in the basics of the neuroevolutionary approach is proposed. The general structure of the genetic algorithm as a multicriteria optimization method is described. Basic information about the structure, principle of operation and methods of training multilayer artificial neural networks with the direct direction of information propagation is provided. Software has been developed to demonstrate the neuroevolutionary approach using the example of the evolution of an artificial neural network of a given structure designed to control a simplified computer model of an autonomous vehicle. The problems of empirical selection of the selection operator and the crossover operator when using evolutionary methods for training artificial neural networks are described. The known negative consequences of rearranging the values of the weighting coefficients of interneuron connections in the interval between two successive layers of artificial neural networks when implementing the crossover operator for artificial neural networks are indicated. The situation of stagnation of the evolutionary process, characteristic of a genetic algorithm, when the function being optimized reaches a local extremum in the process of searching for an optimal solution to the problem is described. A method for resolving the problem of stagnation when using an evolutionary approach for training an artificial neural network is described. A comparison was made of the effectiveness of using proportional selection and tournament selection during the trial operation of the developed software to demonstrate the neuroevolutionary approach using the example of the evolution of an artificial neural network of a given structure designed to control a simplified computer model of an autonomous vehicle. Options for using the developed software when teaching students the basics of artificial intelligence technologies and evolutionary methods of multi-criteria optimization are proposed.*

### **Введение**

В настоящее время интерес к практическому применению и изучению технологий искусственного интеллекта находится на стабильно высоком

уровне. Одной из популярных областей применения технологий искусственного интеллекта является разработка нейроконтроллеров для управления динамическими объектами к которым можно отнести и автономные транспортные средства. Основной проблемой при обучении управляющей искусственной нейронной сети (ИНС) адаптивной коррекцией весовых коэффициентов синаптических связей на основе эталонных входных и выходных сигналов является сложность формирования адекватного обучающего набора данных [1]. Один из выходов для решения подобной проблемы – использование эволюционного метода для обучения ИНС. В предлагаемой статье рассматриваются особенности применения генетического алгоритма в качестве эволюционного метода обучения нейронной сети для управления упрощенной компьютерной моделью автономного транспортного средства пригодной для обучения студентов основам технологий искусственного интеллекта.

### **Генетический алгоритм как эволюционный метод оптимизации**

Генетический алгоритм (ГА) представляет собой адаптивный направленный стохастический метод поиска решения оптимизационных задач на основе использования аналогий механизма естественного отбора и генетического наследования в живой природе [2, 3]. ГА оперирует множеством возможных решений задачи оптимизации как некоторой популяцией особей, подвергаемой моделируемому процессу естественного отбора в процессе смены поколений с использованием механизмов селекции и скрещивания. Потенциальные решения оптимизационной задачи, по терминологии ГА называемые хромосомами, представляются последовательностями значений своих параметров-генов. С каждой хромосомой-решением связана величина приспособленности, соответствующая эвристической оценке близости потенциального решения к ожидаемому оптимуму.

Работу генетического алгоритма можно представить выполнением следующих шагов:

- Согласно сущности оптимизационной задачи представить потенциальные ее решения в виде фиксированного набора искомым значений параметров;

- Определить количество  $N$  потенциальных решений (размер популяции хромосом), поступательную эволюцию которых будет моделировать генетический алгоритм;

- Разработать эвристическую функцию, которая на основе значений параметров-генов входящих в состав решения-хромосомы будет вычислять величину близости потенциального решения к ожидаемому оптимуму (величину приспособленности хромосомы в популяции);

- Выбрать способ отбора пар хромосом-родителей на основе рассчитанной величины приспособленности (оператор селекции) для последующего создания на их основе пар хромосом-потомков;

- Выбрать способ порождения пары хромосом-потомков на основе обработки генов пары хромосом-родителей (оператор скрещивания);
- Определить вероятность и способ случайного изменения величин генов хромосом-потомков (оператор мутации);
- Сформировать начальную популяцию из  $N$  потенциальных решений-хромосом задавая значения искомым параметрам-генам при помощи генератора случайных чисел;
- Вычислить величину приспособленности каждой хромосомы в популяции согласно разработанной эвристической функции;
- На основе анализа величин приспособленности определить необходимость остановки процесса эволюционного поиска решения оптимизационной задачи и в случае положительного ответа завершить работу алгоритма, приняв в качестве результата хромосому с наивысшим значением значения приспособленности – в ином случае продолжить выполнение шагов алгоритма;
- Применить оператор селекции к популяции хромосом;
- Применить оператор скрещивания к парам хромосом-родителей для формирования хромосом-потомков, составляющих на следующей итерации алгоритма очередное поколение потенциальных решений задачи;
- В соответствии с установленной ранее вероятностью применить к хромосомам-потомкам оператор мутации;
- Продолжить выполнение алгоритма с шага 7.

### **Искусственная нейронная сеть как вычислительная система**

Искусственная нейронная сеть (ИНС) представляет собой вычислительную систему, действующую по аналогии с биологическим головным мозгом и состоящую из следующих конструктивных компонентов: вычислительных узлов (искусственных нейронов) и межузловых соединений (синаптических связей) с назначенными им весовыми (синаптическими) коэффициентами [4]. Множество вычислительных узлов вместе с топологией соединений между ними называют архитектурой ИНС. Архитектуру ИНС можно в дальнейшем разбить на три иерархических уровня [5]:

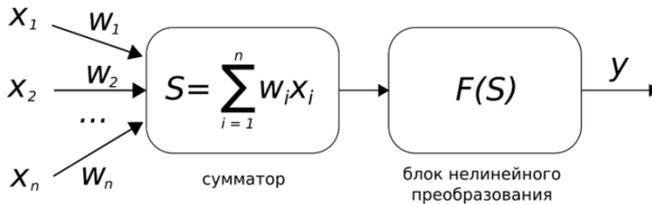
- микроструктурный – на этом уровне описываются характеристики отдельных вычислительных узлов сети;
- мезоструктурный – описывает топологическую организацию соединений между вычислительными узлами и направление распространения информации по сети;
- макроструктурный – способ соединения отдельных ИНС между собой при создании сети модульной структуры.

На микроструктурном уровне искусственный нейрон как вычислительное устройство (рисунок 1) функционирует следующим образом:

- по входным синаптическим связям искусственный нейрон получает сигналы  $x_i$  и вырабатывает один выходной сигнал  $u$ ;
- сигнал, поступающий по конкретной синаптической связи, модифицируется путем умножения на весовой коэффициент  $w_i$ ;

– модифицированные входные сигналы суммируются (агрегируются) и результирующая величина называется совокупным возбуждением искусственного нейрона;

– для получения выходного сигнала у величина совокупного возбуждения нейрона преобразуется нелинейной функцией F, которая называется функцией активации искусственного нейрона.

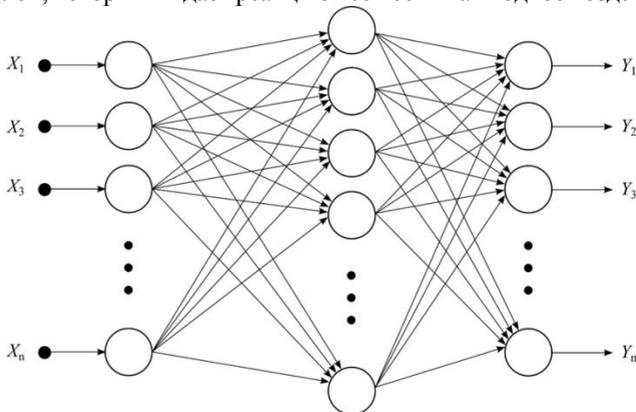


**Рисунок 1 – Искусственный нейрон на микроструктурном уровне**

В качестве функции активации часто используют логистическую, пороговую, гиперболического тангенса и реже – линейную.

На мезоструктурном уровне наиболее известной является многослойная архитектура ИНС с прямым направлением распространения информации (рисунок 2).

Многослойные ИНС с прямым направлением распространения информации состоят из последовательно соединенных синаптическими связями слоев искусственных нейронов: слой, принимающий сигналы из внешней среды, называется входным или распределительным, за ним следует произвольное количество так называемых скрытых слоев (на рисунке 2 изображен единственный скрытый слой) и самым последним в цепочке расположен выходной слой, который выдает реакцию всей сети на входное воздействие.



**Рисунок 2 – Многослойная архитектура искусственной нейронной сети**

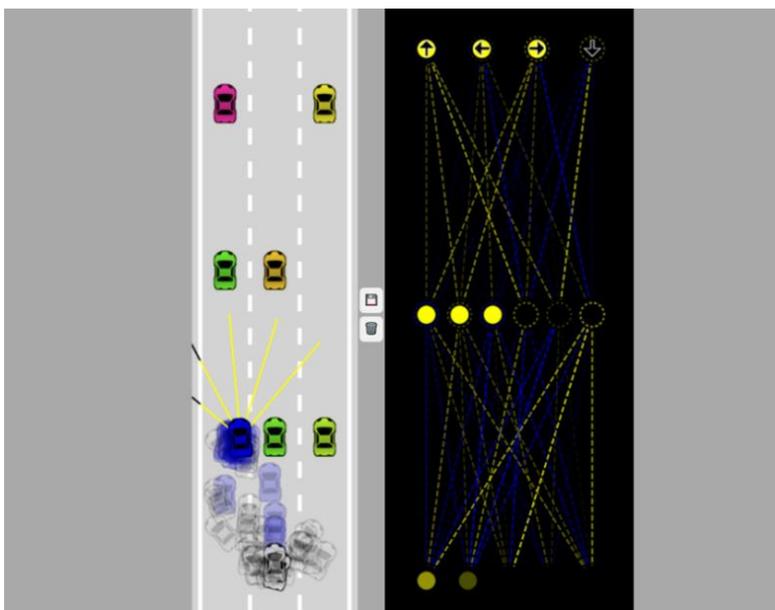
Функционирование ИНС зависит от величин весовых коэффициентов синаптических связей, поэтому при заданной структуре ИНС, отвечающей какой-либо задаче, необходимо найти оптимальные значения весовых коэффициентов [55].

Процесс нахождения оптимальных значений весовых коэффициентов синаптических связей называется обучением ИНС, в котором можно выделить два подхода: обучение с учителем и обучение без учителя [4]. При обучении с учителем необходимо наличие конечного набора значений входных сигналов и соответствующего набора ожидаемых значений выходных сигналов (обучающей выборки). Целью обучения становится подбор весовых коэффициентов таким образом, чтобы фактические выходные сигналы сети принимали значения наиболее близкие к ожидаемым значениям. При невозможности применения такого подхода необходимо использовать обучение без учителя. Подбор весовых коэффициентов синаптических связей сети проводится либо на основании конкуренции нейронов между собой, либо с учетом корреляции обучающих и выходных сигналов.

Для нахождения оптимальной структуры ИНС и значений весовых коэффициентов синаптических связей при затруднительности или невозможности формирования обучающей выборки возможно применять нейрорезолюционные методы, в арсенал которых входит генетический алгоритм [6, 7, 8]. В простейшем случае генетический алгоритм используется для эволюционного подбора значений весовых коэффициентов связей между нейронами ИНС фиксированной структуры при этом потенциальные решения оптимизационной задачи представлены хромосомами, генами которых являются искомые вещественные значения весов межнейронных связей [9].

### **Идея средства демонстрации эволюционного подхода к настройке весовых коэффициентов ИНС**

В качестве отправной точки для практического ознакомления учащихся с основами генетического эволюционного алгоритма оптимизации, был использован видеокурс «Self-Driving Car with JavaScript Course – Neural Networks and Machine Learning» ([https://www.youtube.com/watch?v=Rs\\_rAxEsAvI](https://www.youtube.com/watch?v=Rs_rAxEsAvI)), с сопутствующим исходным кодом приложения на языке JavaScript (<https://github.com/gniziemazity/Self-driving-car>), демонстрирующим применение эволюционного метода обучения нейронной сети для управления упрощенной компьютерной моделью автономного транспортного средства (рисунок 3), автором которого является Раду Мариеску-Истордо (<https://scholar.google.com/citations?user=rZQYWtcAAAAJ&hl=en>) из Университета Восточной Финляндии (<https://www.uef.fi/ru>).



**Рисунок 3 – Демонстрация эволюции ИНС, управляющей моделью автомобиля**

Модель транспортного средства (автомобиль) в форме прямоугольника способная под управлением ИНС к ускорению, торможению, поворотам направо и налево относительно своего геометрического центра движется по прямолинейному участку дороги фиксированной ширины. На участке дороги перед движущимся управляемым автомобилем расположены 7 перемещающихся с низкой скоростью неуправляемых автомобилей играющих роль препятствий.

Для возможности обнаружения препятствий при движении вперед модель обладает набором датчиков приближения в виде пяти отрезков прямых фиксированной длины, расходящихся веерообразно из геометрического центра прямоугольника. При пересечении отрезков прямых с границами дороги и прямоугольниками, представляющими собой иные автомобили, формируется вещественное значение находящееся в промежутке от нуля до единицы пропорционально положению расчетной точки пересечения на отрезке («0» – нет пересечения, «1» – точка пересечения совпадает с началом отрезка в центре прямоугольника модели автомобиля).

Автомобиль управляется искусственной нейронной сетью прямого распространения с одним скрытым слоем из шести нейронов – входной слой состоит из пяти нейронов, связанных с датчиками приближения, а выходной слой непосредственно связан с органами управления. В управляющей ИНС используется пороговая функция активации (функция Хевисайда) – величина порога активации каждого нейрона и величины весовых коэффициентов связей между нейронами подбирается генетическим алгоритмом.

Цель эволюции популяции, состоящей из 100 управляемых автомобилей – получение особи успешно обходящей все встречающиеся на пути препятствия при поступательном движении по дороге вперед без пересечения ее боковых границ.

Процесс эволюции нейроконтроллера интерактивно отображается в окне браузера, разделенном по горизонтали на две части: слева можно наблюдать движение всей популяции автомобилей, а справа изображается управляющая искусственная нейронная сеть наилучшей особи популяции (рисунок 3).

Отличительной чертой представленной реализации генетического алгоритма является то, что функция оператора селекции возложена исключительно на пользователя и состоит в своевременном сохранении управляющей ИНС особи, которая опережает в движении все остальные. На пользователя также возложена задача управления началом процесса формирования новой популяции и запуска очередной итерации алгоритма, причем новая популяция состоит из сохраненной на предыдущей итерации особи, а остальные 99 членов представлены ее случайными мутациями.

#### **Развитие средства демонстрации эволюционного подхода к настройке весовых коэффициентов ИНС**

Старая реализация генетического алгоритма была подвергнута значительному пересмотру в сторону наглядной демонстрации различных операторов селекции и применения одного из возможных операторов скрещивания. Так как структуру ИНС было решено оставить неизменной, то объектом нейроэволюции являются величины порога активации каждого нейрона и весовых коэффициентов межнейронных связей, представляющие собой в этом случае отдельные гены хромосомы. В качестве оператора скрещивания выбрана промежуточная рекомбинация (whole arithmetic recombination), подходящая к хромосомам состоящим из вещественных значений [10]. Промежуточная рекомбинация формирует гены хромосомы-потомка на основе генов хромосом-родителей следующим образом:

$$C_i = A_i + \alpha \cdot (B_i - A_i), \quad (1)$$

$$\alpha \in [d, 1 + d], \quad (2)$$

где  $A_i$ ,  $B_i$  – вещественные значения генов хромосом-родителей,  
 $C_i$  – значение гена хромосомы-потомка, полученное в результате вычислений по формуле (1),  
 $d$  рекомендуется принять равным 0,25 .

Операторы скрещивания, основанные на перекрестном обмене генами между парой хромосом-родителей были исключены из рассмотрения по следующей причине: исследованиями установлено, что перестановка значений весовых коэффициентов межнейронных связей в промежутке между двумя последовательными слоями и даже перестановка самих нейронов в пределах скрытых слоев могут не оказать значительного влияния на функционирование ИНС [11, 12]. Более того, исследования показывают,

что в процессе нейроэволюции скрещивание между особями со сравнимыми значениями приспособленности чаще всего ведет к ухудшению приспособленности потомков, на основании чего делается вывод о деструктивности оператора скрещивания и приемлемости лишь оператора мутации при использовании генетического алгоритма для эволюции ИНС [13].

Для возможности использования операторов селекции необходимо определить эвристическую функцию, результат вычисления которой будет определять приспособленность управляющей ИНС в популяции, поэтому, учитывая что целью нейроэволюции в данном случае является получение особи успешно обходящей все встречающиеся на пути препятствия, то мерой приспособленности нужно считать расстояние со знаком, показывающее положение относительно последнего движущегося неуправляемого автомобиля-препятствия. Признаком остановки итераций генетического алгоритма является опережение последнего препятствия лучшей особью текущего поколения на 800 пикселей экранного пространства.

Для наглядной оценки эффективности применяемых операторов селекции созданы две версии программного обеспечения (ПО): в одной из них используется селекция методом рулетки, а в другой – турнирная селекция. Обе упомянутые версии ПО расположены в общедоступном репозитории в сети Интернет:

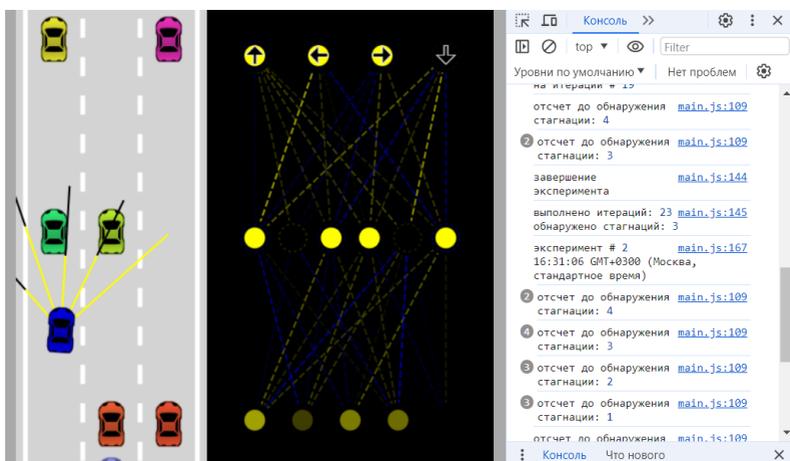
- селекция методом рулетки – <https://github.com/polegdo/JavaScript-Car-Neuroevolution/tree/main/roulette-wheel-selection>,
- турнирная селекция – <https://github.com/polegdo/JavaScript-Car-Neuroevolution/tree/main/tournament-selection>.

При отборе хромосом-родителей методом рулетки (пропорциональный отбор), каждому потенциальному родителю ставится сектор воображаемого колеса рулетки, площадь которого пропорциональна мере приспособленности особи – чем выше приспособленность, тем больше площадь соответствующего сектора и выше вероятность отбора особи в качестве родителя. Турнирная селекция состоит в случайном отборе из текущей популяции фиксированного количества особей (описываемом случае отбирается 80 особей из 100) и выборе в качестве родителя экземпляра с наивысшим значением функции приспособленности.

Как и любой метод многокритериальной оптимизации, генетический алгоритм при проведении нейроэволюции способен попасть в локальный экстремум оптимизируемой функции – индикатором такой ситуации является отсутствие повышения приспособленности особей в популяции при проведении серии очередных итераций алгоритма на пути к требуемому условиям оптимизационной задачи результату функционирования ИНС. Одним из выходов при стагнации генетического алгоритма является формирование нового поколения на основе переноса в неизменном виде наилучшей особи из предыдущего поколения и формирование остальных членов популяции на основе проведения над ней случайных мутаций [14, 15].

Такой подход и был реализован в разработанном программном обеспечении при обнаружении отсутствия улучшения приспособленности членов популяции на протяжении пяти последовательных итераций генетического алгоритма.

Работа программного обеспечения организована в виде проведения экспериментов – серии итераций генетического алгоритма, приводящих к подбору оптимальных значений весовых коэффициентов межнейронных связей управляющей ИНС и позволяющих модели автомобиля преодолеть все препятствия в процессе движения. В целях повышения информативности наблюдения за ходом нейроэволюции в консоли разработчика браузера отображается нумерация проходящих экспериментов, информация о обнаружении ситуации стагнации в эволюции ИНС, выводится итоговая статистика проведения эксперимента, а именно: количество выполненных итераций и обнаруженных стагнаций (рисунок 4).



**Рисунок 4 – Отображение дополнительной информации в консоли браузера (Google Chrome)**

### **Результаты и их обсуждение**

Разработанное средство демонстрации эволюционного подхода к настройке весовых коэффициентов ИНС размещенное в общедоступном репозитории в сети Интернет (<https://github.com/polegdo/JavaScript-Car-Neuroevolution>) возможно творчески применять при чтении лекций и проведении практических занятий по соответствующим предметам у студентов специальностей «Компьютерная инженерия», «Программная инженерия», «Искусственный интеллект», «Системы управления информацией».

При изучении основ генетического алгоритма как метода многокритериальной оптимизации представленное в статье программное обеспечение может наглядно служить примером необходимости эмпирического подбора оператора селекции для эффективного решения

поставленной задачи. В данном случае селекция особей на основе пропорционального отбора показала неспособность к решению задачи при сравнении с использованием турнирной селекции, которая показала многократно повторяющийся успешный результат: при проведении серии из 100 экспериментов генетический алгоритм эволюции управляющей ИНС добивался поставленной перед ним цели в среднем за 30 итераций причем среднее количество обнаруженных стагнаций достигало значения равного 4.

Предлагаемый практический подход к ознакомлению студентов с основами технологий искусственного интеллекта на базе изучения нейроэволюционного алгоритма является перспективным по следующим причинам:

- Наличие стабильного интереса к технологиям беспилотного управления наземными транспортными средствами;
- Возможность ознакомления с основами функционирования искусственных нейронных сетей;
- Изучение основ генетического алгоритма как метода многокритериальной оптимизации;
- Наглядность процесса эволюции искусственной нейронной сети;
- Отсутствие необходимости в использовании сложных средств разработки и тестирования программ на языке JavaScript – единственными минимальными требованиями являются наличие текстового редактора и современного веб-браузера;
- Возможность изучения основ языка программирования JavaScript для написания веб-приложений;
- Реализация ИНС на JavaScript без использования сторонних библиотек и сложного математического аппарата.

Исходный код разработанного автором программного обеспечения может послужить основой для управляемой самостоятельной работы студентов при изучении искусственных нейронных сетей и методов оптимизации на основе генетического алгоритма. В целях облегчения ориентации в исходном коде для заинтересованных лиц следует сказать, что реализация ИНС представляет собой класс `NeuralNetwork` размещенный в файле `network.js`, а функции реализующие операторы селекции расположены в файле `main.js` под именами `selectCarFromRouletteWheel()` и `getTournamentWinner()`. Для дальнейшего развития изложенной в статье идеи имеет смысл использовать платформу `Node.js` совместно библиотекой `TensorFlow.js`, что даст возможность сэкономить время и усилия при реализации более сложных структур ИНС.

#### **Список использованных источников**

1. Чернодуб, А. Н. Обзор методов нейроуправления / А. Н. Чернодуб, Д. А. Дзюба // Проблемы программирования, 2011, № 2, С. 79–94.

2. Katoch, S. A Review on Genetic Algorithm: Past, Present, and Future / S. Katoch, S. S. Chauhan, V. Kumar // *Multimed Tools Appl.* 2021. Vol. 80, P. 8091–8126.
3. McCall, J. Genetic Algorithms for Modelling and Optimisations / J. McCall // *Journal of Computational and Applied Mathematics.* 2020. Vol. 184, № 1. P. 205–222.
4. Васенков, Д. В. Методы обучения искусственных нейронных сетей / Д. В. Васенков // *Компьютерные инструменты в образовании.* 2007, № 1. С. 20–29.
5. Maren, A. J. A Logical Topology of Neural Networks / A. J. Maren // *Proceedings of the Second Workshop on Neural Networks, Auburn, USA, February 11–13, 1991.* – Auburn : Auburn University, 1991. P. 17–44.
6. Мищенко, В. А. Использование генетических алгоритмов в обучении нейронных сетей / В. А. Мищенко, А. А. Коробкин // *Современные проблемы науки и образования.* 2011, № 6.
7. Шумков, Е. А. Использование генетических алгоритмов для обучения нейронных сетей / Е. А. Шумков // *Научный журнал КубГАУ.* 2013, № 91.
8. Montana, D. J. Training Feedforward Neural Networks Using Genetic Algorithms / D. J. Montana, L. Davis // *IJCAI'89: Proceedings of the 11th International Joint Conference on Artificial Intelligence.* Vol. 1, August 20, 1989, P. 762–767.
9. Yao, X. Evolving Artificial Neural Networks / Xin Yao // *Proceedings of the IEEE.* 1999. Vol. 87, № 9. P. 1423–1447.
10. Picek, S. On the Recombination Operator in the Real-Coded Genetic Algorithms / S. Picek, D. Jakobovic, M. Golub // *2013 IEEE Congress On Evolutionary Computation, June 20–23, Cancun, Mexico, 2013.* P. 3103–3110.
11. Zankinski, I. Effects of the Neuron Permutation Problem on Training Artificial Neural Networks with Genetic Algorithms / I. Zankinski // *International Conference on Numerical Analysis and Its Applications, June 15–22, Lozenetz, Bulgaria, 2016.* P. 777–782.
12. Haflidason, S. On the Significance of the Permutation Problem in Neuroevolution / S. Haflidason, R. Neville // *GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, June 8, 2009.* P. 787–794.
13. Pretorius, K. Neural Network Crossover in Genetic Algorithms Using Genetic programming / K. Pretorius, N. Pillay // *Genetic Programming and Evolvable Machines.* 2024. Vol. 25, № 7.
14. Gomez, F. J. Active Guidance for a Finless Rocket Using Neuroevolution / F. J. Gomez, R. Miikkulainen // *Proceedings of the 2003 International Conference on Genetic and Evolutionary Computation: Part II, July 12, San Francisco, California, USA, 2003.* P. 2084–2095.
15. Gomez, F. J. Co-Evolving Recurrent Neurons Learn Deep Memory POMDPs // F. J. Gomez, J. Schmidhuber // *GECCO '05: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, June 25–29, Washington DC, USA, 2005.* P. 491–498.