

Математическое моделирование  
Mathematical modeling

УДК 004.89  
<https://doi.org/10.32362/2500-316X-2022-10-2-59-74>



НАУЧНАЯ СТАТЬЯ

# Применение биоинспирированных алгоритмов глобальной оптимизации для повышения точности прогнозов компактных машин экстремального обучения

Л.А. Демидова<sup>@</sup>,  
А.В. Горчаков

МИРЭА – Российский технологический университет, Москва, 119454 Россия  
<sup>@</sup> Автор для переписки, e-mail: demidova.liliya@gmail.com

## Резюме

**Цели.** В результате современных исследований в машинном обучении, направленных на повышение точности и снижение вычислительной сложности алгоритмов анализа данных, была предложена новая архитектура искусственной нейронной сети – машина экстремального обучения. Это нейронная сеть прямого пространства с единственным скрытым слоем. В этой сети веса соединений между входными нейронами и нейронами скрытого слоя инициализируются случайно, а веса соединений между нейронами скрытого слоя и выходными нейронами вычисляются с использованием операции псевдообращения Мура – Пенроуза. Замена итерационного процесса обучения, присущего многим архитектурам нейронных сетей, на случайную инициализацию одной части весов и вычисление другой части делает рассматриваемый инструмент существенно более производительным, с сохранением хорошей обобщающей способности. Однако случайная инициализация входных весов не гарантирует оптимальной точности прогнозов. Цель работы – разработка и исследование подходов к интеллектуальной настройке входных весов в машинах экстремального обучения биоинспирированными алгоритмами для повышения точности прогнозов этого инструмента анализа данных в задачах восстановления регрессии.

**Методы.** Используются методы теории оптимизации, теории эволюционных вычислений и роевого интеллекта, теории вероятностей и математической статистики, системного анализа.

**Результаты.** Разработаны и исследованы подходы к интеллектуальной настройке входных весов в машинах экстремального обучения, основанные на применении генетического алгоритма, алгоритма роя частиц, алгоритма поиска косяком рыб, алгоритма хаотического поиска косяком рыб с экспоненциальным убыванием шага, предложенного авторами настоящего исследования. Выявлено, что применение биоинспирированных алгоритмов способно улучшить точность прогнозов машин экстремального обучения в задачах восстановления регрессии, причем машине экстремального обучения с уточненными биоинспирированными алгоритмами весами требуется меньшее число нейронов на скрытом слое для достижения высокой точности прогнозов на тренировочных и тестовых наборах данных. С помощью хаотического алгоритма поиска косяком рыб с экспоненциальным убыванием шага могут быть получены наилучшие конфигурации машин экстремального обучения в рассмотренных задачах.

**Выводы.** Полученные результаты показывают, что точность прогнозов машин экстремального обучения может быть улучшена посредством применения биоинспирированных алгоритмов интеллектуальной настройки входных весов. Для выполнения настройки весов требуются дополнительные вычисления, поэтому

использование машин экстремального обучения в сочетании с биоинспирированными алгоритмами может быть целесообразно в тех областях, где необходимо получение наиболее точной и компактной конфигурации машины экстремального обучения.

**Ключевые слова:** нейронные сети, машины экстремального обучения, биоинспирированные алгоритмы, генетический алгоритм, алгоритм роя частиц, алгоритм поиска косяком рыб, машинное обучение, регрессионный анализ

• Поступила: 29.11.2021 • Доработана: 22.12.2021 • Принята к опубликованию: 01.03.2022

**Для цитирования:** Демидова Л.А., Горчаков А.В. Применение биоинспирированных алгоритмов глобальной оптимизации для повышения точности прогнозов компактных машин экстремального обучения. *Russ. Technol. J.* 2022;10(2):59–74. <https://doi.org/10.32362/2500-316X-2022-10-2-59-74>

**Прозрачность финансовой деятельности:** Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

## RESEARCH ARTICLE

# Application of bioinspired global optimization algorithms to the improvement of the prediction accuracy of compact extreme learning machines

Liliya A. Demidova<sup>@</sup>,  
Artyom V. Gorchakov

MIREA – Russian Technological University, Moscow, 119454 Russia  
<sup>@</sup> Corresponding author, e-mail: demidova.liliya@gmail.com

### Abstract

**Objectives.** Recent research in machine learning and artificial intelligence aimed at improving prediction accuracy and reducing computational complexity resulted in a novel neural network architecture referred to as an extreme learning machine (ELM). An ELM comprises a single-hidden-layer feedforward neural network in which the weights of connections among input-layer neurons and hidden-layer neurons are initialized randomly, while the weights of connections among hidden-layer neurons and output-layer neurons are computed using a generalized Moore–Penrose pseudoinverse operation. The replacement of the iterative learning process currently used in many neural network architectures with the random initialization of input weights and the explicit computation of output weights significantly increases the performance of this novel machine learning algorithm while preserving good generalization performance. However, since the random initialization of input weights does not necessarily guarantee optimal prediction accuracy, the purpose of the present work was to develop and study approaches to intelligent adjustment of input weights in ELMs using bioinspired algorithms in order to improve the prediction accuracy of this data analysis tool in regression problems.

**Methods.** Methods of optimization theory, theory of evolutionary computation and swarm intelligence, probability theory, mathematical statistics and systems analysis were used.

**Results.** Approaches to the intelligent adjustment of input weights in ELMs were developed and studied. These approaches are based on the genetic algorithm, the particle swarm algorithm, the fish school search algorithm, as well as the chaotic fish school search algorithm with exponential step decay proposed by the authors.

By adjusting input weights with bioinspired optimization algorithms, it was shown that the prediction accuracy of ELMs in regression problems can be improved to reduce the number of hidden-layer neurons to reach a high prediction accuracy on learning and test datasets. In the considered problems, the best ELM configurations can be obtained using the chaotic fish school search algorithm with exponential step decay.

**Conclusions.** The obtained results showed that the prediction accuracy of ELMs can be improved by using bioinspired algorithms for the intelligent adjustment of input weights. Additional calculations are required to adjust the weights; therefore, the use of ELMs in combination with bioinspired algorithms may be advisable where it is necessary to obtain the most accurate and most compact ELM configuration.

**Keywords:** neural networks, extreme learning machine, bioinspired algorithms, genetic algorithm, particle swarm optimization algorithm, fish school search algorithm, machine learning, regression analysis

• Submitted: 29.11.2021 • Revised: 22.12.2021 • Accepted: 01.03.2022

**For citation:** Demidova L.A., Gorchakov A.V. Application of bioinspired global optimization algorithms to the improvement of the prediction accuracy of compact extreme learning machines. *Russ. Technol. J.* 2022;10(2):59–74. <https://doi.org/10.32362/2500-316X-2022-10-2-59-74>

**Financial disclosure:** The authors have no a financial or property interest in any material or method mentioned.

The authors declare no conflicts of interest.

## ВВЕДЕНИЕ

Процесс цифровизации экономики приводит к тому, что все большее число предприятий интегрирует интеллектуальные модули в собственные продукты и корпоративные информационные системы с целью автоматизации и ускорения бизнес-процессов. Данные модули часто представляют собой системы поддержки принятия решений, экспертные и прогностические системы, использующие в процессе функционирования алгоритмы машинного обучения. Такие алгоритмы позволяют автоматизировать выявление скрытых зависимостей в наборах данных с целью выполнения предсказаний без участия человека. Современные требования к производительности алгоритмов машинного обучения повышаются с увеличением объемов данных, обрабатываемых интеллектуальными системами. При этом точность решений, принимаемых в автоматическом режиме, должна быть максимально высокой.

В ходе исследований в области классификации, регрессионного анализа и прогнозирования временных рядов были предложены такие эффективные инструменты машинного обучения, как метод  $k$  ближайших соседей [1], метод опорных векторов [2], их гибридные версии [3], ансамбли случайных лесов [4], искусственные нейронные сети [5], позволяющие достигать высокой точности принимаемых решений.

Искусственная нейронная сеть (ИНС) является широко используемым и эффективным инструментом машинного обучения с учителем, позволяющим автоматизировать принятие решений с помощью эволюции сложной нелинейной системы [6]. Разработано множество архитектур и способов обучения ИНС, показавших высокую эффективность в

решении задач классификации при использовании глубокого обучения, сверточных слоев и слоев проецирования [7], задач регрессии при использовании гибридов популяционных и градиентных алгоритмов обучения ИНС [8, 9]. С целью поиска оптимальных гиперпараметров ИНС часто применяют алгоритмы, вдохновленные живой природой. Так, в работе [10] алгоритм роя частиц был применен для оптимизации гиперпараметров рекуррентной сети с долгой краткосрочной памятью для прогнозирования временных рядов.

Задача обучения ИНС сводится к задаче минимизации некоторой функции оценки потерь ИНС на тренировочном наборе данных. Традиционно исследователи и практики применяют итерационный метод обратного распространения ошибки и алгоритмы, основанные на стохастическом градиентном спуске, для обучения ИНС [11, 12]. Однако с ростом объемов анализируемых данных растет время, необходимое для сходимости градиентных методов, особенно в случае использования глубоких архитектур ИНС. С целью ускорения процесса обучения ИНС исследователями из корпорации Google был предложен распределенный алгоритм стохастического градиентного спуска [13], предполагающий размещение подмножеств из обучающего набора данных на нескольких ведомых узлах вычислительной сети. При этом на каждой итерации ведомые вычислительные узлы вычисляют многомерные матрицы градиентов для собственного подмножества данных. Ведущий узел получает вычисленные многомерные матрицы градиентов от ведомых узлов, вычитает градиенты из матрицы весов ИНС, а затем рассылает обновленную многомерную матрицу весов ведомым узлам, после чего процесс повторяется до выполнения критерия останова. Однако даже в случае применения распределенных

методов обучения ИНС процесс обучения может занимать продолжительное время – от нескольких часов до нескольких суток.

С целью ускорения процесса обучения ИНС Гуан-Бин и соавторы предложили новую архитектуру ИНС [14], названную машиной экстремального обучения (англ. extreme learning machine, ELM). ELM – это ИНС с единственным скрытым слоем, в которой веса соединений между входными нейронами и нейронами скрытого слоя, а также вектор сдвигов скрытого слоя инициализируются случайно, а веса соединений между нейронами скрытого слоя и выходными нейронами вычисляются с использованием операции псевдообращения Мура – Пенроуза [15]. При этом функция активации скрытого слоя должна быть бесконечно дифференцируема [14]. ELM позволяет исключить итерационный процесс оптимизации функции оценки потерь ИНС из процесса подготовки модели ИНС, способной принимать точные решения, существенно снижая вычислительные затраты на обучение ИНС. Несмотря на отсутствие итерационного процесса обучения, ELM показывают хорошую обобщающую способность в ряде прикладных задач [16–18].

Однако случайная инициализация весов соединений между входными и скрытыми нейронами и вектора сдвигов скрытого слоя не гарантирует получения ELM с наилучшей конфигурацией из всех возможных. Известно о применении алгоритмов оптимизации, вдохновленных живой природой, в задачах интеллектуального уточнения входных весов и вектора сдвигов скрытого слоя ELM [16, 19]. Такие алгоритмы оптимизации, также называемые биоинспирированными, являются эвристическими методами глобальной оптимизации, не использующими информацию о производной оптимизируемой функции и обрабатывающими несколько решений на каждой итерации, что упрощает запуск данных алгоритмов в параллельном и распределенном режимах [20]. Широко применяемыми на практике биоинспирированными алгоритмами являются генетический алгоритм (англ. Genetic Algorithm, GA) [21], алгоритм роя частиц (англ. Particle Swarm Optimization, PSO) [22], алгоритм поиска косяком рыб (англ. Fish School Search, FSS) [23, 24] и др. В работе [16] описано применение алгоритма роя частиц для получения оптимальной конфигурации ELM для прогнозирования транспортных потоков. В работе [19] приведен сравнительный анализ классической реализации ELM и ELM, в которой проведен интеллектуальный выбор входных весов с помощью генетического алгоритма. Показана высокая эффективность конфигурации ELM, полученной с помощью GA.

В данной работе приведено исследование эффективности различных биоинспирированных алгоритмов, включая генетический алгоритм, алгоритм роя частиц, алгоритм поиска косяком рыб, алгоритм хаотического поиска косяком рыб с экспоненциальным убыванием шага, применяемых в задаче выбора оптимальных значений весов соединений между входными нейронами и нейронами скрытого слоя и вектора сдвигов скрытого слоя. Рассмотрены задачи восстановления регрессии на 3 наборах данных, приведено сравнение обобщающей способности классической ELM [14], и ELM, в которых произведена интеллектуальная настройка входных весов биоинспирированными алгоритмами. Дополнительно приведено исследование ландшафтов оптимизируемых биоинспирированными алгоритмами функций оценки потерь в процессе уточнения входных весов и сдвигов ELM.

### МАШИНА ЭКСТРЕМАЛЬНОГО ОБУЧЕНИЯ

В задаче машинного обучения с учителем даны набор объектов  $X = X_L \cup X_T$ , где  $X_L$  – набор данных для обучения (англ. learning),  $X_T$  – набор данных для тестирования (англ. testing); набор возможных ответов  $Y$  и неизвестная целевая функция  $f: X \rightarrow Y$ , отображающая множество объектов в множество возможных ответов. Значения  $f$  известны для каждого объекта из набора  $X$ . Набор  $X_L$  используется в процессе обучения модели, а оценка качества обученной модели производится на наборе  $X_T$ . Обучающий набор  $X_L$  имеет вид  $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_s\}$ , где  $s$  – число объектов в  $X_L$ , каждый  $i$ -й объект  $\vec{x}_i \in X_L$  представлен в виде набора признаков  $\vec{x}_i = (h_1, h_2, \dots, h_n)$ , характеризующих объект  $\vec{x}_i$ , значения  $\vec{y}_i$  известны для каждого объекта  $\vec{x}_i$  и равны  $f(\vec{x}_i)$ . В процессе обучения алгоритмом машинного обучения осуществляется построение функции  $a: X \rightarrow Y$ , которая бы достаточно хорошо аппроксимировала неизвестную целевую зависимость  $f$  на  $X = X_L \cup X_T$ .

Машина экстремального обучения является алгоритмом машинного обучения с учителем, позволяющим, аналогично ИНС, осуществлять принятие решений путем автоматического конфигурирования сложной нелинейной системы для конкретной задачи. ELM представляет собой ИНС прямого распространения с единственным скрытым слоем и бесконечно дифференцируемой функцией активации на скрытом слое [14], веса соединений между входными нейронами и нейронами скрытого слоя в ELM инициализируются случайно, как и сдвиги скрытого слоя, а веса между нейронами скрытого слоя и выходными нейронами – вычисляются. Структура ELM представлена на рис. 1.

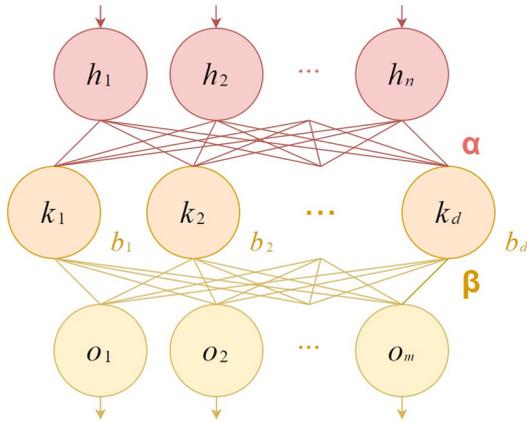


Рис. 1. Машина экстремального обучения

Элементы  $\alpha$  – матрицы весов соединений между входными нейронами  $h_1, h_2, \dots, h_n$  и нейронами скрытого слоя  $k_1, k_2, \dots, k_d$ , имеющей вид  $\mathbb{R}^{d \times n}$ , где  $n$  – число входных нейронов, а  $d$  – число скрытых нейронов, инициализируются случайно, сдвиги нейронов скрытого слоя  $b_1, b_2, \dots, b_d$ , также инициализируются случайно, вектор сдвигов имеет вид  $\mathbb{R}^d$ . Матрица весов соединений  $\beta$  между нейронами скрытого слоя  $k_1, k_2, \dots, k_d$  и выходными нейронами  $o_1, o_2, \dots, o_m$ , имеющая вид  $\mathbb{R}^{d \times m}$ , где  $m$  – число выходных нейронов, вычисляется следующим образом:

$$\beta = \mathbf{H}^\dagger \mathbf{Y}_L, \text{ где } \mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T, \mathbf{H} = g(\mathbf{X}_L \alpha^T + \mathbf{b}), \quad (1)$$

где  $\mathbf{X}_L$  обозначает матрицу вида  $\mathbb{R}^{s \times n}$ , в которой строки кодируют  $s$  объектов из обучающего набора  $X_L$ . При этом каждый объект представлен в виде набора  $n$  признаков  $\{h_1, h_2, \dots, h_n\}$ ,  $\alpha^T$  обозначает транспонированную матрицу весов соединений между нейронами вида  $\mathbb{R}^{n \times d}$ , где  $n$  – число входных нейронов,  $d$  – число выходных нейронов.  $\mathbf{b}$  – матрица вида  $\mathbb{R}^{s \times d}$ , полученная преобразованием вектора сдвигов скрытого слоя вида  $\mathbb{R}^d$  в матрицу  $\mathbb{R}^{1 \times d}$ , в которой 1-я строка затем дублируется  $s$  раз.  $g$  – бесконечно дифференцируемая функция активации, поэлементно применяемая к каждому элементу матрицы.  $\mathbf{H}$  – выходная матрица скрытого слоя вида  $\mathbb{R}^{s \times d}$ .  $\mathbf{H}^\dagger$  – псевдообращение Мура – Пенроуза для матрицы  $\mathbf{H}$  [15] вида  $\mathbb{R}^{d \times s}$ .  $\mathbf{Y}_L$  обозначает матрицу ответов вида  $\mathbb{R}^{s \times m}$ , в которой строки кодируют ответы, соответствующие объектам обучающего набора  $X_L$ , каждый ответ имеет вид  $\{o_1, o_2, \dots, o_m\}$ . Таким образом, матрица весов соединений между  $d$  нейронами скрытого слоя и  $m$  выходными нейронами  $\beta$  имеет вид  $\mathbb{R}^{d \times m}$ . В качестве функции активации скрытого слоя  $g$  в ELM часто используют сигмоидальную функцию активации [25]:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2)$$

где  $x$  – элемент выходной матрицы скрытого слоя, имеющей вид  $\mathbb{R}^{s \times d}$ , получаемой перемножением матриц  $\mathbf{X}_L$  и  $\alpha^T$ , и прибавлением к результату матрицы  $\mathbf{b}$ .

Число нейронов скрытого слоя  $d$  является гиперпараметром ELM и требует настройки в зависимости от решаемой задачи. Число признаков  $n$  каждого объекта из обучающего набора данных определяется в зависимости от специфики предметной области решаемой задачи. В задаче восстановления регрессии с помощью ELM, рассматриваемой в данной работе, число выходных нейронов  $m$  полагают равным 1.

### ИНТЕЛЛЕКТУАЛЬНЫЙ ВЫБОР ВХОДНЫХ ВЕСОВ БИОИНСПИРИРОВАННЫМИ АЛГОРИТМАМИ

Случайная инициализация весов соединений между входными и скрытыми нейронами  $\alpha$  и сдвигов скрытого слоя  $\mathbf{b}$  не гарантирует получения оптимальной конфигурации ELM [16, 19]. Известно о применении алгоритма роя частиц для получения оптимальной конфигурации ELM в задаче прогнозирования транспортных потоков [16], о применении генетического алгоритма в задачах восстановления регрессии [19]. В обозначенных работах авторы используют среднеквадратическую функцию ошибки (англ. root mean square error, RMSE) в качестве целевой функции для биоинспирированных алгоритмов, определяемую как:

$$\text{RMSE} = \sqrt{\frac{1}{s} \sum_{i=1}^s (a(\vec{x}_i) - y_i)^2}, \quad (3)$$

где  $\vec{x}_i$  –  $i$ -й объект из набора данных для обучения  $X_L$ , содержащего  $s$  объектов;  $y_i$  – ответ для  $i$ -го объекта;  $a(\vec{x}_i)$  – предсказание ELM для  $i$ -го объекта  $\vec{x}_i$ .

Для оценки обобщающей способности обученной модели ИНС часто используют функцию средней абсолютной ошибки, определяемую как:

$$\text{MAE} = \frac{1}{s} \sum_{i=1}^s |a(\vec{x}_i) - y_i|, \quad (4)$$

где  $\vec{x}_i$  –  $i$ -й объект из набора данных для обучения  $X_L$ , содержащего  $s$  объектов;  $y_i$  – ответ для  $i$ -го объекта;  $a(\vec{x}_i)$  – предсказание ELM для  $i$ -го объекта. Функция средней абсолютной ошибки была использована в [16] для оценки качества модели ELM.

При решении задачи выбора оптимальных значений входных весов, заданных матрицей  $\alpha$  вида  $\mathbb{R}^{d \times n}$ , и вектора сдвигов скрытого слоя  $\{b_1, b_2, \dots, b_d\}$  вида  $\mathbb{R}^d$  (рис. 1), каждый  $i$ -й агент популяции

биоинспирированных алгоритмов может быть представлен в виде вектора:

$$\{\alpha_{11}^i, \alpha_{12}^i, \dots, \alpha_{1n}^i, \alpha_{21}^i, \alpha_{22}^i, \dots, \alpha_{2n}^i, \dots, \alpha_{d1}^i, \alpha_{d2}^i, \dots, \alpha_{dn}^i, b_1^i, b_2^i, \dots, b_d^i\}.$$

Генетический алгоритм, применяемый в задаче интеллектуального выбора входных весов и сдвигов ELM в работе [19], является эвристическим популяционным алгоритмом оптимизации, вдохновленным эволюционными процессами живой природы. Алгоритм 1 определяет псевдокод генетического алгоритма. На каждой итерации к агентам популяции последовательно применяются операторы отбора, скрещивания и мутации, причем могут использоваться различные стратегии отбора, такие, как турнирный отбор, отбор усечением [21].

Алгоритм роя частиц, применяемый в задаче выбора оптимальных значений входных весов и

сдвигов скрытого слоя ELM в работе [16], является широко известным роевым алгоритмом глобальной оптимизации, вдохновленным поведением скоординированного полета стай птиц. Алгоритм был предложен в работе [22]. Алгоритм 2 определяет псевдокод алгоритма роя частиц.

Алгоритм поиска косяком рыб (англ. Fish School Search, FSS) является алгоритмом глобальной оптимизации, вдохновленным поведением косяков рыб, движущихся в поисках пищи. Этот алгоритм был предложен Bastos Filho и Lima Neto [23] и с тех пор нашел применение при решении множества задач, включая реконструкцию изображений [26] и расстановку весов в ИНС [27].

В работе [24] Л.А. Демидовой и А.В. Горчаковым была предложена модификация алгоритма поиска косяком рыб (англ. Tent map-based Fish School Search with Exponential step decay, ETFSS), ускоряющая сходимость FSS. Алгоритм 3 определяет псевдокод ETFSS.

---

#### Алгоритм 1. Генетический алгоритм

---

**Вход:**  $P_{\text{crossover}}$  – вероятность скрещивания,  $P_{\text{mutation}}$  – вероятность мутации

- 1: определить функцию приспособленности ( $f$ )
  - 2: установить номер поколения в 0 ( $t = 0$ )
  - 3: случайно сгенерировать агентов начальной популяции  $P_t$
  - 4: вычислить значение  $f$  для каждого агента из  $P_t$
  - 5: **пока** не выполнен критерий останова, **выполнять**
  - 6:      $t = t + 1$
  - 7:     отобрать агентов для популяции  $P_t$  из популяции  $P_{t-1}$
  - 8:     изменить агентов  $P_p$ , используя оператор *скрещивания* с вероятностью  $P_{\text{crossover}}$
  - 9:     изменить агентов  $P_p$ , используя оператор *мутации* с вероятностью  $P_{\text{mutation}}$
  - 10:    вычислить значение  $f$  для каждого агента в  $P_t$
  - 11: **завершить цикл**
  - 12: **возвратить** лучшее найденное в процессе эволюции решение
- 

#### Алгоритм 2. Алгоритм роя частиц\*

---

**Вход:**  $w, c_1, c_2, v_{\text{max}}$

- 1: определить функцию приспособленности ( $f$ )
  - 2: установить номер поколения в 0 ( $t = 0$ )
  - 3: случайно сгенерировать агентов начальной популяции  $P_t$
  - 4: случайно сгенерировать скорости агентов  $V_t$  на промежутке  $[-v_{\text{max}}, v_{\text{max}}]$
  - 5: вычислить значение  $f$  для каждого агента из  $P_t$
  - 6: выбрать агента  $\vec{p}_{\text{gbest}} \in P_t$  такого, что  $\forall \vec{p}_{i,t} \in P_t : f(\vec{p}_{\text{gbest},t}) \leq f(\vec{p}_{i,t})$
  - 7: для каждой частицы  $\vec{p}_{i,t} \in P_t$  установить лучшее положение  $\vec{p}_{\text{pbest},i}$  равным  $\vec{p}_{i,t}$
  - 8: **пока** не выполнен критерий останова, **выполнять**
-

- 9:  $t = t + 1$
- 10: **для каждого** агента  $\vec{p}_{i,t} \in P_t$  **выполнять**
- 11: случайно сгенерировать  $\vec{r}_1$  и  $\vec{r}_2$ , компоненты которых  $\in [0,1]$
- 12:  $\vec{v}_{i,t} = w\vec{v}_{i,t-1} + c_1\vec{r}_1(\vec{p}_{\text{gbest}} - \vec{p}_{i,t}) + c_2\vec{r}_2(\vec{p}_{\text{pbest},i} - \vec{p}_{i,t})$
- 13: ограничить  $\vec{v}_{i,t}$  промежутком  $[-v_{\text{max}}, v_{\text{max}}]$
- 14:  $\vec{p}_{i,t} = \vec{p}_{i,t-1} + \vec{v}_{i,t}$
- 15: вычислить значение  $f$  для агента  $\vec{p}_{i,t}$
- 16: **если**  $f(\vec{p}_{i,t}) < f(\vec{p}_{\text{pbest},i})$ , **то**  $\vec{p}_{\text{pbest},i} = \vec{p}_{i,t}$
- 17: **завершить цикл**
- 18: выбрать агента  $\vec{p}_{\text{gbest},t} \in P_t$  такого, что  $\forall \vec{p}_{i,t} \in P_t : f(\vec{p}_{\text{gbest},t}) \leq f(\vec{p}_{i,t})$
- 19: **если**  $f(\vec{p}_{\text{gbest},t}) < f(\vec{p}_{\text{gbest}})$ , **то**  $\vec{p}_{\text{gbest}} = \vec{p}_{\text{gbest},t}$
- 20: **завершить цикл**
- 21: **возвратить** лучшее найденное решение  $\vec{p}_{\text{gbest}}$

\*gbest – глобально лучшее решение на итерации  $t$  в терминологии алгоритма роя частиц (англ. global best); pbest – лучшее найденное решение конкретным агентом на итерации  $t$  в терминологии алгоритма роя частиц (англ. personal best)

### Алгоритм 3. Алгоритм хаотического поиска косяком рыб с экспоненциальным убыванием шага\*

**Вход:**  $step_{\text{ind,initial}}, step_{\text{vol,initial}}, \gamma = 5$

- 1: определить функцию приспособленности ( $f$ )
- 2: установить номер поколения в 0 ( $t = 0$ )
- 3: случайно сгенерировать агентов начальной популяции  $P_t$
- 4: вычислить значение  $f$  для каждого агента из  $P_t$
- 5: выбрать агента  $\vec{p}_{\text{gbest}} \in P_t$  такого, что  $\forall \vec{p}_{i,t} \in P_t : f(\vec{p}_{\text{gbest},t}) \leq f(\vec{p}_{i,t})$
- 6: **пока** не выполнен критерий останова, **выполнять**
- 7:  $t = t + 1$
- 8:  $step_{\text{ind},t} = step_{\text{ind,initial}} e^{\frac{-\gamma t}{iter_{\text{max}}}}$
- 9:  $step_{\text{vol},t} = step_{\text{vol,initial}} e^{\frac{-\gamma t}{iter_{\text{max}}}}$
- 10: **для каждого** агента  $\vec{p}_{i,t} \in P_t$  **выполнять**
- 11: случайно сгенерировать  $\vec{r}_1$  и  $\vec{r}_2$ , компоненты которых  $\in [0,1]$
- 12:  $\vec{p}_{i,t} = \vec{p}_{i,t-1} + step_{\text{ind},t}\vec{r}_1$
- 13: вычислить значение  $f$  для  $\vec{p}_{i,t}$
- 14: **если**  $f(\vec{p}_{i,t}) \geq f(\vec{p}_{i,t-1})$  **то**  $\vec{p}_{i,t} = \vec{p}_{i,t-1}$
- 15:  $w_{i,t} = w_{i,t-1} + \frac{\Delta f_{i,t}}{\max(\Delta f_{i,t+1})}$

16: завершить цикл

$$17: \vec{I} = \frac{\sum_{i=1}^n (\vec{p}_{i,t} - \vec{p}_{i,t-1}) \Delta f_{i,t}}{\sum_{i=1}^n \Delta f_{i,t}}$$

18: для каждого агента  $\vec{p}_{i,t} \in P_t$  выполнить  $\vec{p}_{i,t} = \vec{p}_{i,t} + \vec{I}_t$

$$19: \vec{B} = \frac{\sum_{i=1}^n \vec{p}_{i,t} w_{i,t}}{\sum_{i=1}^n w_{i,t}}$$

20: для каждого агента  $\vec{p}_{i,t} \in P_t$  выполнять

$$21: \vec{p}_{i,t} = \vec{p}_{i,t} \pm step_{vol,t} \vec{r}_2 \frac{\vec{p}_{i,t} - \vec{B}_{t+1}}{|\vec{p}_{i,t} - \vec{B}_{t+1}|}; \text{ знак «-», если } \sum_{i=1}^n w_{i,t} > \sum_{i=1}^n w_{i,t-1}, \text{ иначе «+»}$$

22: завершить цикл

23: выбрать агента  $\vec{p}_{gbest,t}$  такого, что  $\forall \vec{p}_{i,t} \in P_t : f(\vec{p}_{gbest,t}) \leq f(\vec{p}_{i,t})$

24: если  $f(\vec{p}_{gbest,t}) < f(\vec{p}_{gbest})$ , то  $\vec{p}_{gbest} = \vec{p}_{gbest,t}$

25: завершить цикл

26: вернуть лучшее найденное решение  $\vec{p}_{gbest}$

\* $step_{ind}$  – максимальный размер шага индивидуального перемещения в терминологии алгоритма поиска оптимума косяком рыб;  $step_{vol}$  – максимальный размер шага коллективно-волевого перемещения в терминологии алгоритма поиска оптимума косяком рыб;  $iter$  – число итераций;  $iter_{max}$  – максимально допустимое число итераций для алгоритма

В Алгоритме 3 генерация равномерно распределенных случайных чисел для векторов  $\vec{r}_1$  и  $\vec{r}_2$  осуществляется на основе отображения tent с помощью динамической системы, находящейся в хаотическом состоянии. Динамическая система описывается набором времен  $T$ , набором состояний  $S$  и отображением  $M: T \times S \rightarrow S$ , описывающим эволюцию динамической системы. Отображение tent задается как:

$$y_{n+1} = \mu \min(y_n, 1 - y_n), \text{ где } \mu = 1.9999, \quad (5)$$

где число  $\mu$  называют бифуркационным параметром;  $y_n$  определяет состояние динамической системы в момент времени  $t$ .

Диаграмма бифуркаций системы (5) представлена на рис. 2а.

Сравнение экспоненциального убывания шага, используемого в алгоритме ETFSS, и линейного

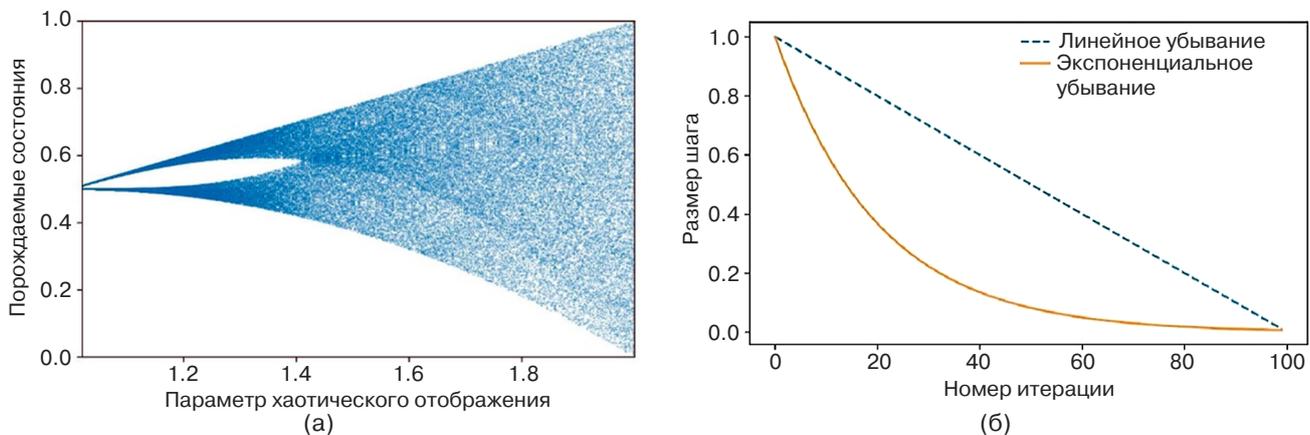


Рис. 2. (а) Диаграмма бифуркаций динамической системы отображения (5); (б) сравнение экспоненциального и линейного убывания шага в FSS и ETFSS

убывания шага, используемого в оригинальном алгоритме поиска косяком рыб, представлено на рис. 2б. Эффективность применения хаотического генератора псевдослучайных чисел (5) и экспоненциального убывания шага (рис. 2б) в алгоритме поиска косяком рыб показана в [24].

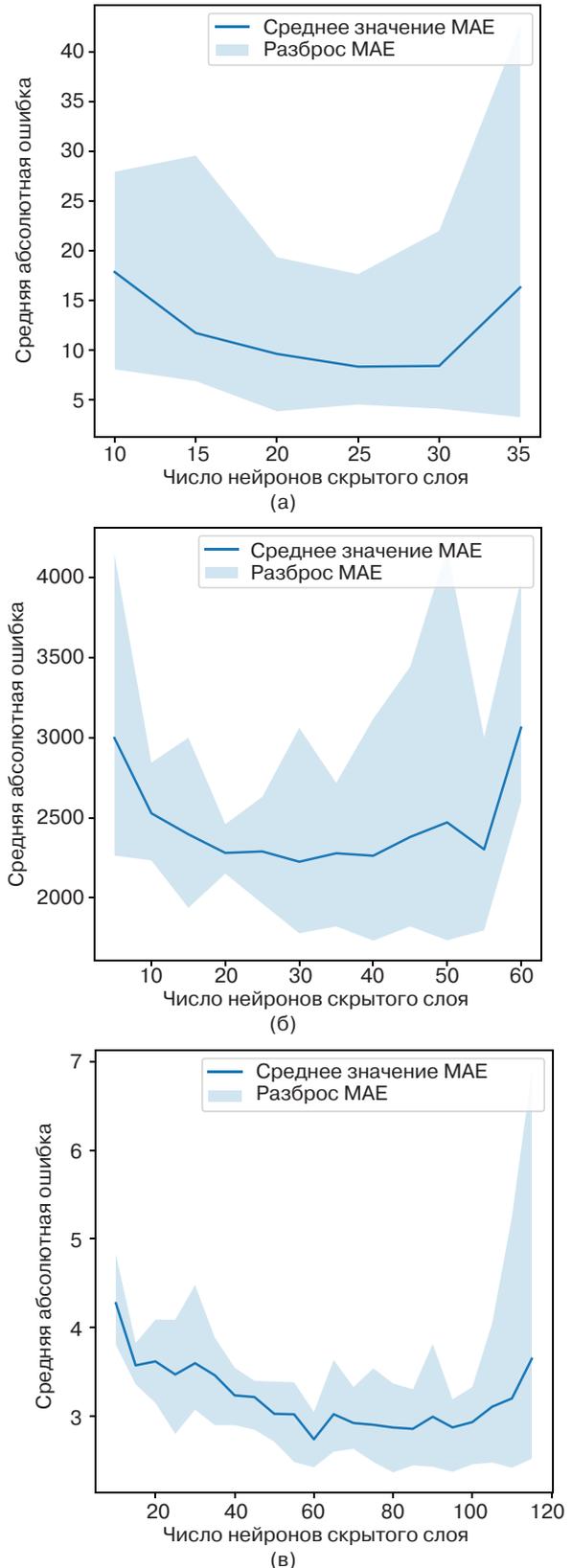
### ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Для исследования эффективности алгоритмов GA, PSO, FSS и ETFSS в задаче расстановки входных весов и сдвигов в ELM и для сравнения полученных конфигураций ELM с классической реализацией ELM, в которой входные веса и сдвиги инициализированы случайно, были рассмотрены 3 открытых набора данных, описанных в [28, 29]. Первый набор содержит сведения о производительности центральных процессоров (англ. Central Processing Unit (CPU) Performance), состоит из 209 строк и 10 столбцов. Каждая строка кодирует 9 признаков, которые потенциально могут влиять на 10-й признак, количественно характеризующий производительность процессора. Второй набор содержит сведения о ценах на автомобили (англ. Auto Imports), содержит 206 строк и 26 столбцов. Каждая строка кодирует 25 признаков объекта, которые могут влиять на 26-й признак – цену автомобиля. Третий набор данных содержит информацию о ценах домов в Бостоне (англ. Boston Housing), содержит 506 строк и 14 столбцов, каждая строка кодирует 13 признаков объектов, потенциально влияющих на 14-й признак.

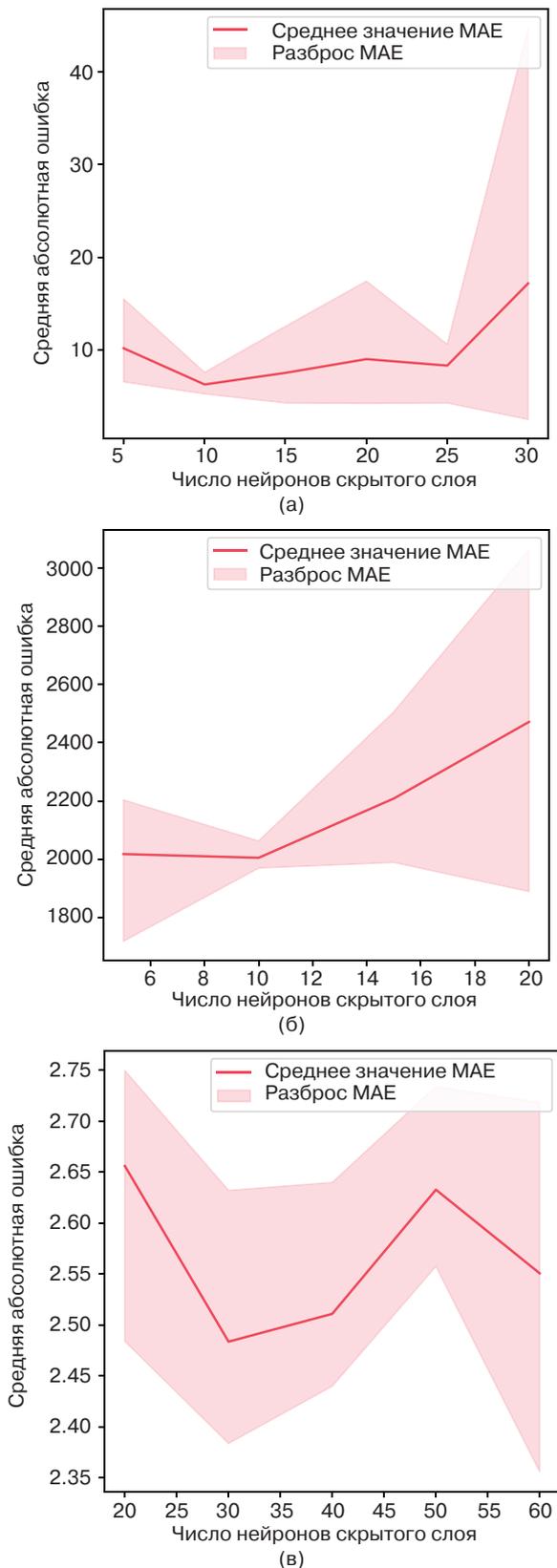
Для решения задачи восстановления регрессионной зависимости для указанных наборов данных с помощью ELM была использована сигмоидальная функция активации (2) на скрытом слое. Для классической реализации ELM каждый из наборов данных был 10 раз разделен на обучающий набор  $X_L$ , включающий 70% от общего числа объектов из набора  $X$ , и тестовый набор  $X_T$ , включающий 30% объектов набора  $X$ . Классическая реализация ELM была обучена 10 раз на наборе  $X_L$  по формуле (1) и 10 раз оценена на наборе  $X_T$  по формуле (4), с целью выбора оптимального числа нейронов на скрытом слое. Процесс выбора числа нейронов представлен на рис. 3, выбранные числа нейронов приведены в табл. 1.

**Таблица 1.** Выбранное число нейронов на скрытом слое для классической ELM

Набор данных	CPU Performance	Auto Imports	Boston Housing
Число нейронов	30	20	60



**Рис. 3.** Выбор оптимального числа нейронов ELM. Светло-синим цветом обозначен разброс значений (4) на  $X_T$  в процессе перекрестной валидации по 10 блокам, синим цветом обозначены усредненные значения (4) для наборов: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing



**Рис. 4.** Выбор числа нейронов ELM, в которой производится настройка входных весов и сдвигов алгоритмом ETFSS. Светло-красным цветом обозначен разброс значений (4) на  $X_T$ , красным цветом обозначены усредненные значения (4) для наборов: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing

Аналогичным образом был проведен выбор числа нейронов в ELM, обученной с помощью биоинспирированных алгоритмов. В работах [16, 19] показано, что компактные ELM (в которых число нейронов на скрытом слое сравнительно невелико) способны достигать лучшей обобщающей способности в случае применения биоинспирированных алгоритмов для выбора входных весов и сдвигов. Вследствие этого верхняя граница числа нейронов скрытого слоя была установлена согласно табл. 1. Предварительное исследование, проведенное методом перекрестной проверки по 10 блокам для различных чисел нейронов скрытого слоя в ELM, где входные веса и сдвиги скрытого слоя были расставлены биоинспирированными алгоритмами, установило, что с помощью ETFSS могут быть получены модели с наилучшей обобщающей способностью. Результаты выбора числа нейронов для ELM, улучшенной с помощью ETFSS (ETFSS-ELM), представлены в табл. 2 и на рис. 4.

**Таблица 2.** Выбранное число нейронов в ELM, настроенной алгоритмом ETFSS

Набор данных	CPU Performance	Auto Imports	Boston Housing
Число нейронов	10	10	30

Как показано на рис. 3 и рис. 4, в случае выбора слишком малого или слишком большого числа нейронов точность предсказаний ELM снижается вследствие недообучения или переобучения модели, соответственно. При этом ETFSS-ELM требуется меньшее число нейронов для достижения лучшей точности на тестовых данных по сравнению с традиционной ELM.

На рис. 5 представлены графики сходимости алгоритмов GA, PSO и ETFSS в процессе оптимизации среднеквадратической функции оценки потерь компактной ELM (3) на обучающих наборах данных при значениях параметров биоинспирированных алгоритмов, представленных в табл. 3. Числа нейронов на скрытом слое в компактных биоинспирированных ELM приведены в табл. 2.

**Таблица 3.** Значения параметров биоинспирированных алгоритмов

Алгоритм	Выбранные значения параметров
GA	$P_{\text{crossover}} = 0.9, P_{\text{mutation}} = 0.1$
PSO	$v_{\text{max}} = 5, c_1 = 0.8, c_2 = 0.5, w = 0.8$
FSS, ETFSS	$step_{\text{ind,initial}} = 0.7, step_{\text{vol,initial}} = 0.7$
Все алгоритмы	300 итераций, 100 агентов популяции

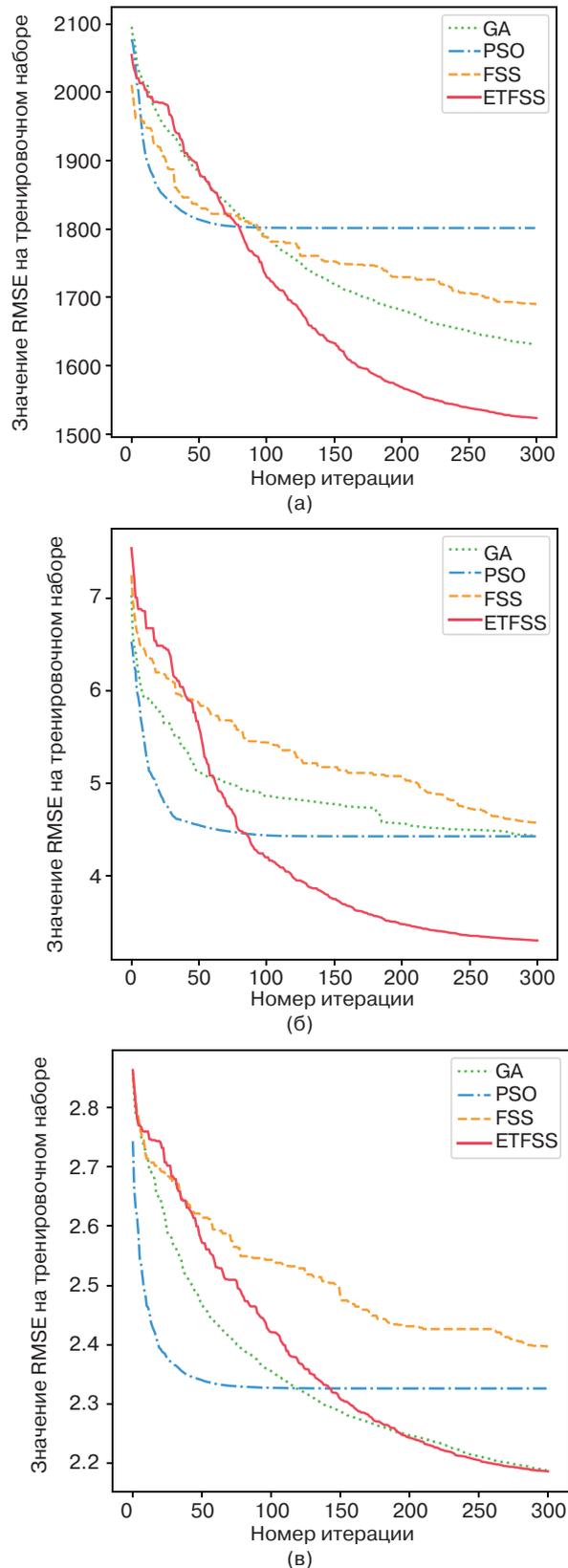
Из графиков, представленных на рис. 5, очевидно, что алгоритм ETFSS, являющийся улучшенной версией алгоритма FSS, способен находить наилучшие решения в задаче поиска оптимума функции оценки потерь (3) по сравнению с алгоритмами GA, PSO и FSS.

Для оценки обобщающей способности моделей GA-ELM, PSO-ELM, FSS-ELM и ETFSS-ELM, полученных в результате уточнения весов алгоритмами GA, PSO, FSS и ETFSS соответственно, были получены диаграммы размаха значений функции средней абсолютной ошибки (англ. mean average error, MAE) (4), представленные на рис. 6. Средние значения MAE на тестовых подмножествах рассмотренных наборов данных представлены в табл. 4. В сравнение также были включены классические реализации ELM со случайной инициализацией входных весов и сдвигов с числами нейронов, указанными в табл. 1 и табл. 2.

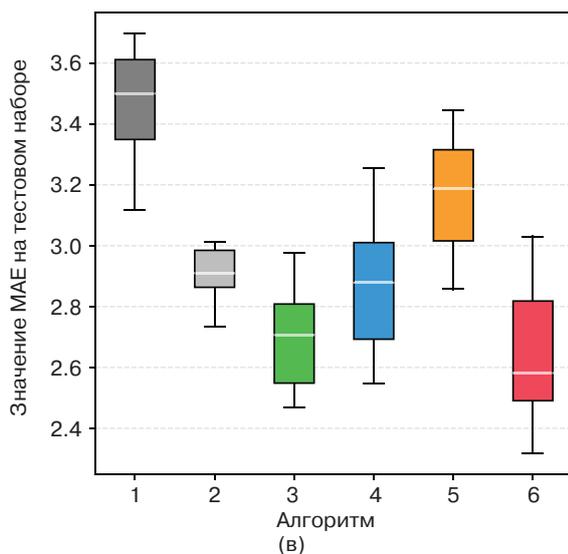
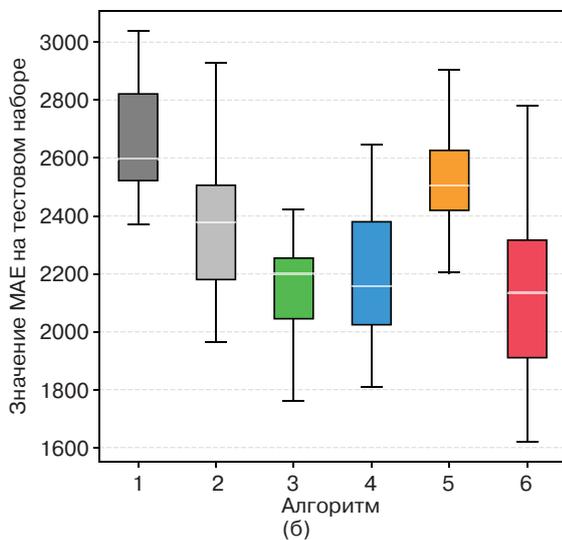
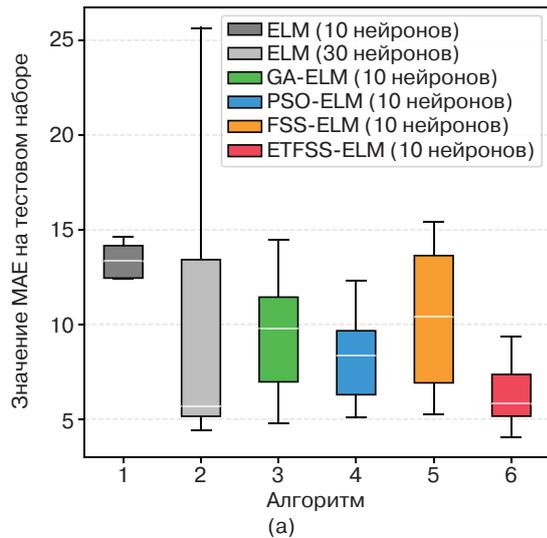
**Таблица 4.** Средние значения MAE (4) на тестовых подмножествах рассматриваемых наборов данных, полученные применением перекрестной проверки по 10 блокам к ELM (1) с числами скрытых нейронов (табл. 2), к ELM (2) с числами скрытых нейронов (табл. 1), к GA-ELM, PSO-ELM, FSS-ELM, ETFSS-ELM

Набор данных	ELM (1)	ELM (2)	GA-ELM	PSO-ELM	FSS-ELM	ETFSS-ELM
CPU Performance	13.2	10.4	9.4	8.8	10.3	<b>6.2</b>
Auto Imports	2657	2354	2155	2199	2522	<b>2124</b>
Boston Housing	3.47	2.89	2.69	2.86	3.11	<b>2.64</b>

Из рис. 6 и табл. 4 следует, что модель ETFSS-ELM показывает наилучшую обобщающую способность среди рассмотренных моделей. GA-ELM, PSO-ELM, FSS-ELM, ETFSS-ELM, в которых была проведена интеллектуальная настройка весов и сдвигов, для всех наборов данных показывают лучшую обобщающую способность по сравнению с классической ELM с аналогичным числом нейронов на скрытом слое. ELM с увеличенными числами нейронов на скрытом слое, указанными в табл. 1, показывает лучшие результаты по сравнению с компактной классической ELM, уступая компактным ELM, настроенным биоинспирированными алгоритмами, с числами скрытых нейронов, приведенными в табл. 2 и рис. 4.



**Рис. 5.** Сходимость алгоритмов, вдохновленных живой природой, в процессе оптимизации функции (3) с целью выбора входных весов и сдвигов ELM для наборов данных: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing



**Рис. 6.** Диаграммы размаха значений MAE (4) на тестовых данных для классических и биоинспирированных ELM с различным числом скрытых нейронов для наборов: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing

### ВИЗУАЛИЗАЦИЯ ЛАНДШАФТОВ ОПТИМИЗИРУЕМОЙ ФУНКЦИИ ОЦЕНКИ ПОТЕРЬ

Для демонстрации процесса поиска оптимальных значений входных весов  $\alpha$  размерности  $\mathbb{R}^{d \times n}$ , где  $d$  – число нейронов на скрытом слое ELM;  $n$  – число входных нейронов и сдвигов размерности  $\mathbb{R}^d$  была проведена визуализация ландшафтов целевой функции (3). Целевая функция (3) в рассматриваемой задаче интеллектуальной настройки входных весов и сдвигов ELM принимает на вход многомерный вектор  $\vec{w}$  вида  $\mathbb{R}^q$ , где  $q = d \times n + d$ . Значения  $q$  для каждого из рассматриваемых наборов данных представлены в табл. 5.

**Таблица 5.** Размерности вектора  $\vec{w}$ , содержащего веса и сдвиги ELM

Набор данных	CPU Performance	Auto Imports	Boston Housing
Размерность $\vec{w}$	100	260	420

Как показано в табл. 5, размерность целевой функции (3) достаточно высока в каждой из рассматриваемых задач, однако визуализация ландшафтов возможна только для двумерных функций. Существует ряд подходов, позволяющих визуализировать многомерные функции в трехмерной прямоугольной системе координат. В методе, описанном в работе [30], предлагается определить 2 ортогональных вектора  $\vec{a}$  и  $\vec{b}$ , размерность которых совпадает с размерностью  $\vec{w}$ , после чего определить визуализируемую функцию  $u$  как:

$$u(\alpha, \beta) = f(\vec{w} + \alpha \vec{a} + \beta \vec{b}), \quad (5)$$

где  $f$  – исходная функция;  $\vec{w}$  – многомерный вектор;  $\alpha$  и  $\beta$  – скалярные параметры;  $\vec{a}$  и  $\vec{b}$  – ортогональные единичные векторы, размерность которых совпадает с размерностью вектора  $\vec{w}$ .

В рассматриваемой задаче функция  $f$  определена по формуле (3); вектор  $\vec{w}$  содержит входные веса и скрытые сдвиги оптимальной конфигурации ELM;  $\vec{a} = \{0, 1, 0, 1, \dots\}$ ,  $\vec{b} = \{1, 0, 1, 0, \dots\}$ ; размерности векторов  $\vec{a}, \vec{b}, \vec{w}$  совпадают. Для каждого из рассмотренных наборов данных были получены визуализации ландшафтов функции (3) вблизи найденного оптимума  $\vec{w}$ , представленные на рис. 7 и рис. 8. Скалярные параметры  $\alpha$  и  $\beta$  изменяются в процессе визуализации на промежутке  $[-1, 1]$  с шагом 0.02.

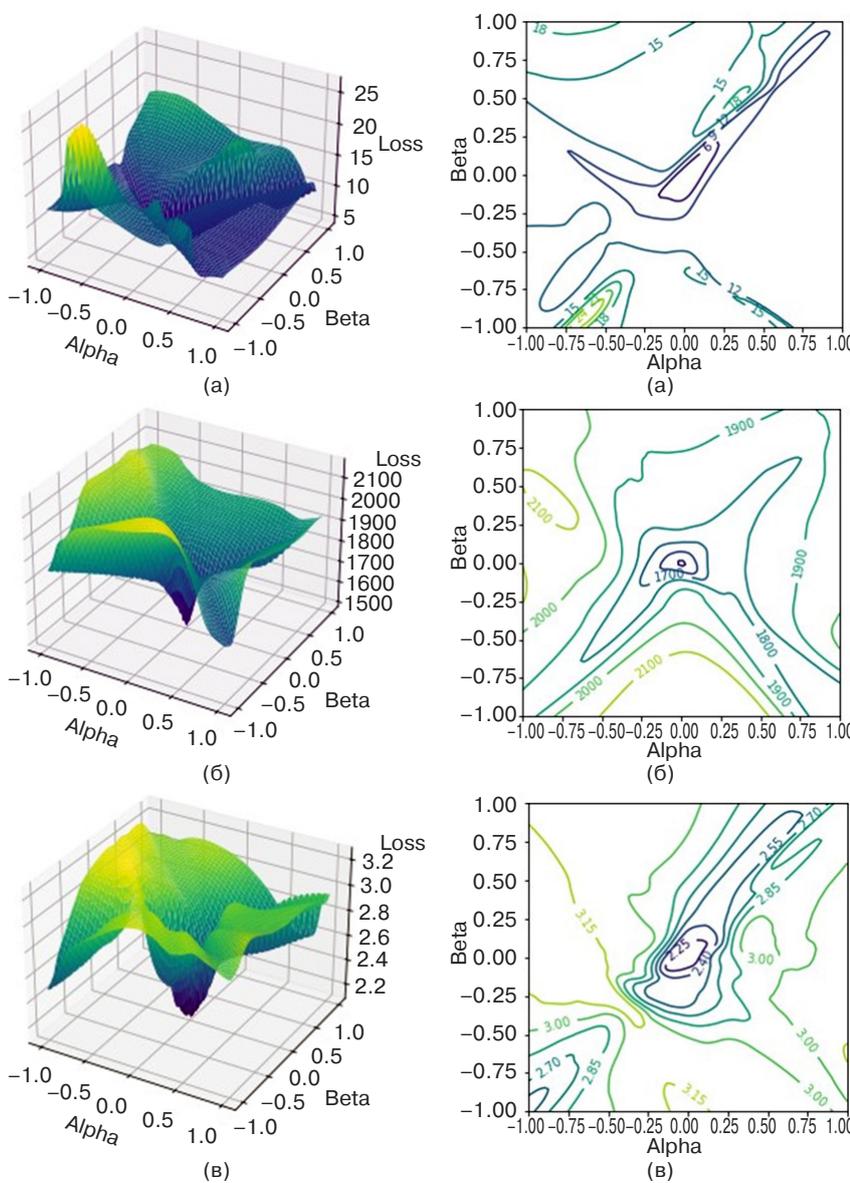
Для визуализации процесса сходимости алгоритма ETFSS в пространствах, содержащих множество локальных экстремумов, показанных на рис. 7 и

рис. 8, были получены изменения позиции случайно выбранного агента алгоритма ETFSS на каждой 25-й итерации. Для каждой позиции агента  $\vec{p}$  была выбрана ближайшая точка из построенной в процессе визуализации ландшафтов методом варьирования скаляров  $\alpha$  и  $\beta$  сетки, причем для определения близости вектора  $\vec{p}$  к точке  $\{\alpha_i, \beta_j\}$  было вычислено манхеттенское расстояние со смещением относительно вектора  $\vec{w}$ , вблизи которого выполняется визуализация:

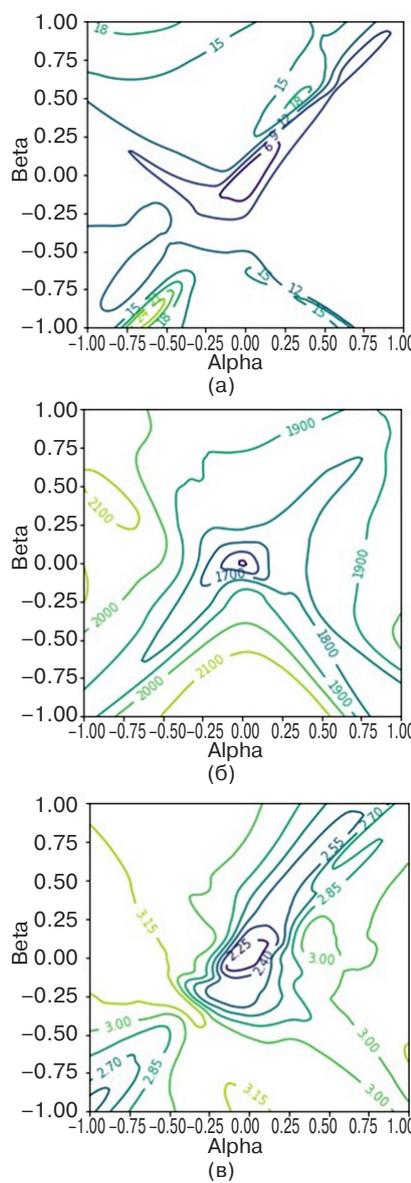
$$\begin{aligned} dist(\vec{p}, \alpha_i, \beta_j) &= |(\vec{p} - \vec{w}) - (\alpha_i \vec{a} + \beta_j \vec{b})| = \\ &= \sum_{k=1}^n |(p_k - w_k) - (\alpha_i a_k + \beta_j b_k)|, \end{aligned} \quad (6)$$

где  $n$  – размерность вектора позиции агента  $\vec{p}$ , совпадающая с размерностью вектора  $\vec{w}$ , кодирующего входные веса и сдвиги найденной оптимальной конфигурации ELM. Размерность вектора позиции агента  $\vec{p}$  также совпадает с размерностью взаимно ортогональных векторов  $\vec{a}$  и  $\vec{b}$ ,  $\vec{a} = \{0, 1, 0, 1, \dots\}$ ,  $\vec{b} = \{1, 0, 1, 0, \dots\}$ ,  $\alpha_i$  и  $\beta_j$  – координаты точки сетки визуализации.

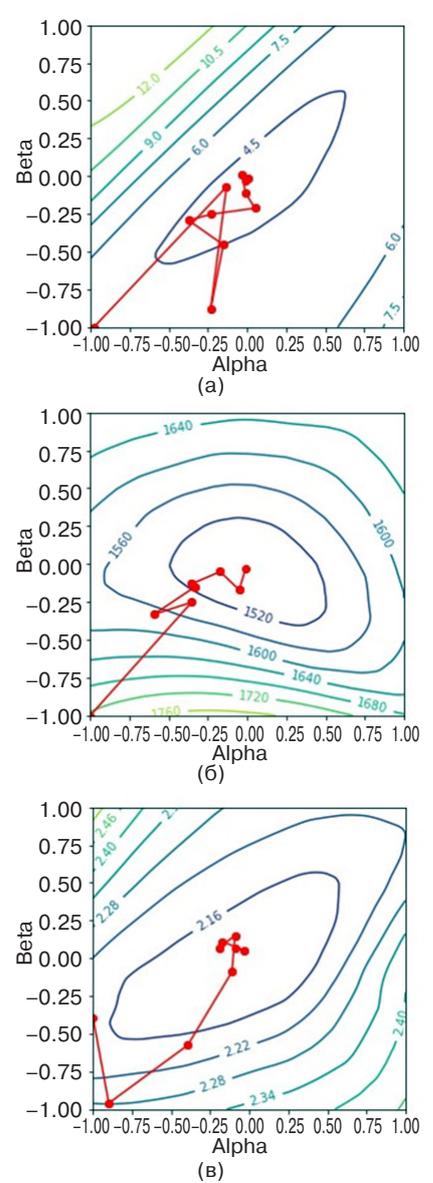
Визуализации изменения позиции случайно выбранного агента  $\vec{p}$  алгоритма ETFSS на каждой 25-й итерации в процессе поиска оптимальных значений входных весов и сдвигов ELM методом оптимизации функции оценки потерь (3) вблизи найденного решения  $\vec{w}$  представлены на рис. 9.



**Рис. 7.** Визуализация ландшафтов многомерных функций оценки потерь вблизи найденного оптимума для наборов данных: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing



**Рис. 8.** Визуализация линий уровня многомерных функций оценки потерь вблизи найденного оптимума для наборов: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing



**Рис. 9.** Изменения позиции случайно выбранного агента алгоритма ETFSS вблизи найденного решения для наборов: (а) CPU Performance; (б) Auto Imports; (в) Boston Housing

Как показано на рис. 7 и рис. 8, функция (3) в рассматриваемых задачах имеет множество экстремумов, успешно преодолеваемых биоинспирированными алгоритмами оптимизации в процессе поиска оптимальных значений входных весов и сдвигов скрытых нейронов ELM. Из траекторий, представленных на рис. 9, следует, что алгоритм ETFSS улучшает находимые популяцией агентов решения на каждой итерации вплоть до завершения выполнения алгоритма.

## ЗАКЛЮЧЕНИЕ

В представленном исследовании была изучена эффективность применения алгоритмов, вдохновленных живой природой, в задаче интеллектуального выбора весов соединений между входными нейронами и скрытыми нейронами, а также сдвигов скрытого слоя машины экстремального обучения в задачах восстановления регрессии.

Приведено сравнение обобщающей способности классической ELM, компактной классической ELM, компактных ELM, в которых проведена настройка входных весов и сдвигов генетическим алгоритмом (GA-ELM), алгоритмом роя частиц (PSO-ELM), алгоритмом поиска косяком рыб (FSS-ELM), алгоритмом хаотического поиска косяком рыб с экспоненциальным убыванием шага (ETFSS-ELM). Выявлено, что для достижения лучшей обобщающей способности ELM, в которых проведена настройка весов биоинспирированными алгоритмами, требуется меньшее число нейронов на скрытом слое. Подобные компактные ELM показывают лучшую обобщающую способность. С помощью хаотического алгоритма поиска косяком рыб с экспоненциальным убыванием шага ETFSS, предложенным в работе [24], могут быть получены наилучшие конфигурации машин экстремального обучения в рассмотренных задачах.

Приведенные визуализации ландшафтов многомерной функции оценки потерь в трехмерной прямоугольной системе координат вблизи найденного решения демонстрируют наличие множества экстремумов, из-за чего применение биоинспирированных алгоритмов поиска глобального оптимума является целесообразным. Полученные визуализации траектории движения случайно выбранного агента алгоритма ETFSS показывают, что алгоритм с каждой итерацией улучшает находимые популяцией решения.

Дальнейшие исследования могут быть направлены на изучение возможности применения ETFSS для уточнения весов онлайн-машины экстремального обучения, способной дообучаться по мере получения новых данных без итерационного процесса обучения [31]. Анализ зависимости результатов,

получаемых с помощью биоинспирированных алгоритмов, от их гиперпараметров также является перспективным направлением исследований.

### Вклад авторов

**Л.А. Демидова** – концептуализация, руководство, проверка, работа над текстом статьи.

**А.В. Горчаков** – программное обеспечение, визуализация, тестирование, работа над текстом статьи.

Все авторы прочитали и одобрили опубликованную версию рукописи.

### Authors' contribution

**L.A. Demidova**—conceptualization, guidance, supervision, validation, and original draft preparation.

**A.V. Gorchakov**—software, resources, visualization, testing, and original draft preparation.

All authors have read and agreed to the published version of the manuscript.

## СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Wu Y., Ianakiev K., Govindaraju V. Improved  $k$ -nearest neighbor classification. *Pattern Recognition*. 2002;35(10):2311–2318. [https://doi.org/10.1016/S0031-3203\(01\)00132-7](https://doi.org/10.1016/S0031-3203(01)00132-7)
2. Noble W.S. What is a support vector machine? *Nat. Biotechnol.* 2006;24(12):1565–1567. <https://doi.org/10.1038/nbt1206-1565>
3. Demidova L.A. Two-stage hybrid data classifiers based on SVM and kNN algorithms. *Symmetry*. 2021;13(4):615. <https://doi.org/10.3390/sym13040615>
4. Lin W., Wu Z., Lin L., Wen A., Li J. An ensemble random forest algorithm for insurance Big Data analysis. *IEEE Access*. 2017;5:16568–16575. <https://doi.org/10.1109/ACCESS.2017.2738069>
5. Deng L., Hinton G., Kingsbury B. New types of deep neural network learning for speech recognition and related applications: An overview. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013:8599–8603. <https://doi.org/10.1109/ICASSP.2013.6639344>
6. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*. 1958;65(6):386–408. <https://doi.org/10.1037/h0042519>
7. Affonso C., Debiaso Rossi A.L., Antunes Vieira F.H., Ponce de Leon Ferreira de Carvalho A.C. Deep learning for biological image classification. *Expert Systems with Applications*. 2017;85:114–122. <https://doi.org/10.1016/j.eswa.2017.05.039>
8. Chen N., Xiong C., Du W., Wang C., Lin X., Chen Z. An improved genetic algorithm coupling a back-propagation neural network model (IGA-BPNN) for water-level predictions. *Water*. 2019;11(9):1795. <https://doi.org/10.3390/w11091795>
9. Such F.P., Madhavan V., Conti E., Lehman J., Stanley K.O., Clune J. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv preprint arXiv:1712.06567. 2017. <https://arxiv.org/abs/1712.06567>

10. Shao B., Li M., Zhao Y., Bian G. Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm. *Mathematical Problems in Engineering*. 2019;2019(2):1934796. <https://doi.org/10.1155/2019/1934796>
11. Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747. 2016. <https://arxiv.org/abs/1609.04747>
12. Куликов А.А. Структура локального детектора модели репринта объекта на изображении. *Russ. Technol. J.* 2021;9(5):7–13. <https://doi.org/10.32362/2500-316X-2021-9-5-7-13>  
[Kulikov A.A. The structure of the local detector of the reprint model of the object in the image. *Russ. Technol. J.* 2021;9(5):7–13. <https://doi.org/10.32362/2500-316X-2021-9-5-7-13>]
13. Dean J., Corrado G.S., Monga R., Chen K., Devin M., Le Q.V., Mao M.Z., Ranzato M.A., Senior A., Tucker P., Yang K., Ng A.Y. Large scale distributed deep networks. *Advances in Neural Information Processing Systems*. 2012;25:1223–1231.
14. Huang G.B., Zhu Q.Y., Siew C.K. Extreme learning machine: theory and applications. *Neurocomputing*. 2006;70(1–3):489–501. <https://doi.org/10.1016/j.neucom.2005.12.126>
15. Rao C.R. Generalized inverse of a matrix and its applications. In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*. 1972. V. 1. *Theory of Statistics*. 1972:601–620. <https://doi.org/10.1525/9780520325883-032>
16. Cai W., Yang J., Yu Y., Song Y., Zhou T., Qin J. PSO-ELM: A hybrid learning model for short-term traffic flow forecasting. *IEEE Access*. 2020;8:6505–6514. <https://doi.org/10.1109/ACCESS.2019.2963784>
17. Liu Y., Loh H.T., Tor S.B. Comparison of extreme learning machine with support vector machine for text classification. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Innovations in Applied Artificial Intelligence*. 2005;3533:390–399. [http://doi.org/10.1007/11504894\\_55](http://doi.org/10.1007/11504894_55)
18. Li G.X. Application of extreme learning machine algorithm in the regression fitting. In: *2016 International Conference on Information System and Artificial Intelligence (ISAI)*. 2016:419–422. <https://doi.org/10.1109/ISAI.2016.0095>
19. Song S., Wang Y., Lin X., Huang Q. Study on GA-based training algorithm for extreme learning machine. In: *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE*. 2015;2:132–135. <https://doi.org/10.1109/IHMSC.2015.156>
20. Никонов В.В., Горчаков А.В. Тренировка моделей машинного обучения с использованием современных средств контейнеризации и облачной инфраструктуры. *Промышленные АСУ и контроллеры*. 2021;6:33–43. <https://doi.org/10.25791/asu.6.2021.1288>
- [Nikonov V.V., Gorchakov A.V. Train machine learning models using modern containerization and cloud Infrastructure. *Promyshlennye ASU i kontrolyery = Industrial Automated Control Systems and Controllers*. 2021;6:33–43 (in Russ.). <https://doi.org/10.25791/asu.6.2021.1288>]
21. Ereemeev A.V. A genetic algorithm with tournament selection as a local search method. *J. Appl. Ind. Math.* 2012;6(3):286–294. <https://doi.org/10.1134/S1990478912030039>
22. Kennedy J., Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. 1995;4:1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
23. Bastos Filho C.J.A., de Lima Neto F.B., Lins A.J.C.C., Nascimento A.I.S., Lima M.P. A novel search algorithm based on fish school behavior. In: *2008 IEEE Int. Conference on Systems, Man and Cybernetics*. 2008:2646–2651. <https://doi.org/10.1109/ICSMC.2008.4811695>
24. Demidova L.A., Gorchakov A.V. A study of chaotic maps producing symmetric distributions in the fish school search optimization algorithm with exponential step decay. *Symmetry*. 2020;12(5):784. <https://doi.org/10.3390/sym12050784>
25. Cao W., Gao J., Ming Zh., Cai Sh. Some tricks in parameter selection for extreme learning machine. *IOP Conf. Ser.: Mater. Sci. Eng.* 2017;261(1):012002. <https://doi.org/10.1088/1757-899X/261/1/012002>
26. Dos Santos W., Barbosa V., de Souza R., Ribeiro R., Feitosa A., Silva V., Ribeiro D., Covello de Freitas R., Lima M., Soares N. Image reconstruction of electrical impedance tomography using fish school search and differential evolution. In: *Critical Developments and Applications of Swarm Intelligence*. IGI Global; 2018. P. 301–338. <https://doi.org/10.4018/978-1-5225-5134-8.ch012>
27. Demidova L.A., Gorchakov A.V. Application of chaotic Fish School Search optimization algorithm with exponential step decay in neural network loss function optimization. *Procedia Computer Science*. 2021;186(6):352–359. <https://doi.org/10.1016/j.procs.2021.04.156>
28. Harrison D. Jr., Rubinfeld D.L. Hedonic housing prices and the demand for clean air. *J. Environ. Econ. Manag.* 1978;5(1):81–102. [https://doi.org/10.1016/0095-0696\(78\)90006-2](https://doi.org/10.1016/0095-0696(78)90006-2)
29. Kibler D., Aha D.W., Albert M.K. Instance-based prediction of real-valued attributes. *Comput. Intell.* 1989;5(2):51–57. <https://doi.org/10.1111/j.1467-8640.1989.tb00315.x>
30. Li H., Xu Z., Taylor G., Studer C., Goldstein T. Visualizing the loss landscape of neural nets. In: *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018:6391–6401. <https://arxiv.org/abs/1712.09913v3>
31. Dai B., Gu C., Zhao E., Zhu K., Cao W., Qin X. Improved online sequential extreme learning machine for identifying crack behavior in concrete dam. *Adv. Struct. Eng.* 2019;22(2):402–412. <https://doi.org/10.1177/1369433218788635>

### Об авторах

**Демидова Лилия Анатольевна**, д.т.н., профессор, профессор кафедры корпоративных информационных систем Института информационных технологий ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: demidova.liliya@gmail.com. Scopus Author ID 56406258800, ResearcherID R-6077-2016. <https://orcid.org/0000-0003-4516-3746>

**Горчаков Артём Владимирович**, аспирант кафедры корпоративных информационных систем Института информационных технологий ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: worldbeater-dev@yandex.ru. Scopus Author ID 57215001290, ResearcherID ABC-8911-2021. <https://orcid.org/0000-0003-1977-8165>

### About the authors

**Liliya A. Demidova**, Dr. Sci. (Eng.), Professor, Professor, ERP Systems Department, Institute of Information Technologies, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: demidova.liliya@gmail.com. Scopus Author ID 56406258800, ResearcherID R-6077-2016. <https://orcid.org/0000-0003-4516-3746>

**Artyom V. Gorchakov**, Postgraduate Student, ERP Systems Department, Institute of Information Technologies, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: worldbeater-dev@yandex.ru. Scopus Author ID 57215001290, ResearcherID ABC-8911-2021. <https://orcid.org/0000-0003-1977-8165>