

Создание web-приложения, реализующего алгоритм решения задачи символьной регрессии на основе искусственной иммунной системы

Ю.В. Хицкова

Аннотация— Исходными данными для задачи символьной регрессии служат множество свободных параметров и соответствующее ему множество значений функции от этих параметров. Кроме этого задается множество всех допустимых для суперпозиции функций и ограничений для них, если они требуются. В качестве функций стоит выбирать непрерывно дифференцируемые функции.

Каждое полученное решение необходимо оценить. Для этого определяется функция аффинности, которая определяет степень приближения полученного решения к ожидаемым результатам. Так как при оценке используются заданные множества - точки, то в качестве целевой функции можно использовать различные метрики и погрешности. Суперпозиция выбранных функций является хромосомой и потенциальным решением, и представляется в виде двоичного дерева. Для оценки выживаемости каждой хромосомы необходимо задать целевую функцию (функцию выживаемости). Так же её называют мерой аффинности. В качестве неё выбирается одна из предложенных функций.

Один из операторов искусственной иммунной системы (ИИС) – селекция, процесс отбора лучших кандидатов для «получения» нового поколения лимфоцитов. Лучших особей старой популяции можно включить в следующую популяцию.

Скрещивание осуществляется по принципу кроссинговера. В биологии кроссинговер - процесс обмена участками гомологичных хромосом. В нашем случае, когда хромосомы – бинарные деревья, необходимо получить наследника, т.е. комбинацию этих деревьев.

Еще один оператор – мутация хромосомы. Этот этап необходим для внесения разнообразия в популяцию и предотвращения «проваливания» решения в локальный максимум или минимум. Приводится весь алгоритм работы иммунной системы.

Для успешного создания web-приложения необходимо перечислить технологии. Был выбран язык программирования Java. В качестве СУБД выбор пал на PostgreSQL. Для создания серверной части использовался Spring Framework, для клиентской – VueJs.

Для написания кода была выбрана среда разработки IntelliJ IDEA из-за ее удобного интерфейса. Приводится

модель данных и структура пакета. Проведено тестирование.

Ключевые слова— искусственные иммунные системы, оператор кроссовера, оператор мутации, функция аффинности, веб приложение, тестирование.

I. ВВЕДЕНИЕ

Для описания алгоритма часто применяют биологические термины, как отголоски происхождения данной теории. Так, одно из множества решений называют особью или лимфоцитом, а все множество – популяцией. Так как естественный отбор говорит о том, что «выживает наиболее приспособленный» необходимо каждую из особей в популяции оценить на «выживаемость» с помощью целевой функции (так же ее называют fitness function, функция аффинности), которая дает оценку пригодности особи для требуемого условия. Чем выше оценка (либо минимально отклонение от требуемых результатов), тем вероятнее то, что особь будет участвовать в дальнейшей селекции. Селекция – процесс искусственного отбора особей для создания потомков, которые в дальнейшем попадут в следующую популяцию. После выбора происходит процедура скрещивания – размножение популяции, которая должна поддерживать разнообразие в популяции. Так же для увеличения разнообразия используются мутации конкретных особей, которые происходят для небольших групп внутри популяции. Под мутацией понимается изменение нескольких параметров особи. Формируется новая популяция из полученных особей и ей присваивается следующий номер, номер поколения. Происходит оценивание всех особей вновь и алгоритм повторяется, пока не будет достигнута необходимая точность, либо номер популяции не достигнет максимального значения [1-13].

Исходными данными для задачи символьной регрессии служат множество свободных параметров и соответствующее ему множество значений функции от этих параметров. Кроме этого задается множество всех допустимых для суперпозиции функций и ограничения для них, если они требуются. В качестве функций стоит выбирать непрерывно дифференцируемые функции.

Каждое полученное решение необходимо оценить. Для этого определяется целевая функция(функция аффинности), которая определяет степень приближения

Статья получена 27 июля 2024.

Ю.В. Хицкова – доцент кафедры систем управления и информационных технологий в строительстве факультета информационных технологий и компьютерной безопасности, кандидат экономических наук, доцент, Воронежский государственный технический университет (email: prosvetovau@list.ru) ORCID: <https://orcid.org/0000-0001-6322-8633>

полученного решения к ожидаемым результатам. Так как при оценке используются заданные множества - точки, то в качестве целевой функции можно использовать различные метрики и погрешности.

В качестве решения представляется оптимальная функция (суперпозиция заданных функций), которую надо закодировать для генетического программирования. Для этого используется представление решения в виде дерева. Все дальнейшие манипуляции происходят именно над деревом, а после, при получении ответа уже происходит преобразование в строковое выражение.

II. СТРУКТУРА АЛГОРИТМА

В качестве входных данных служат два множества $\{x_1, x_2, \dots, x_n\} x_i \in R^m \forall i$ и $\{y_1, y_2, \dots, y_n\}$, которые представляют собой точки из множества исходных данных D . Все точки должны принадлежать одному пространству точек R^m . Вторым параметром является набор используемых в суперпозиции функций. В качестве них можно использовать следующие: $\cos(x), \sin(x), +, -, *, \ln(x + 0.0001)$ и другие. Все функции должны быть непрерывны на решаемом промежутке [1-2,4].

Суперпозиция выбранных функций является лимфоцитом и потенциальным решением, и представляется в виде двоичного дерева. На рисунке 1 представлена запись решения $(\sqrt{x} * 3) + x$. В листовых узлах дерева содержатся переменные или константы, а в нелистовых расположены операции над зависимыми узлами.

Высоту дерева из-за соображений ограниченности ресурсов необходимо ограничивать. Так же это способствует оптимизации решения, чтобы не было переизбытка ветвей дерева.

Для оценки выживаемости каждого лимфоцита необходимо задать целевую функцию (функцию выживания). Так же её называют мерой аффинности. В качестве неё можно выбрать одну из следующих формул:

$$\sqrt{\sum_1^n (f(x_i) - y_i)^2};$$

$$\sqrt{\frac{\sum_1^n (f(x_i) - y_i)^2}{n}} \text{ или } \sqrt{\frac{\sum_1^n (f(x_i) - y_i)^2}{n * (n-1)}};$$

$$\sqrt{\sum_1^n \left(\frac{f(x_i) - y_i}{y_i} \right)^2},$$

где x_i и y_i – это пара значений, точка из входных множеств, размерности n . Функция $f(x)$ – полученное стохастическое решение (суперпозиция функций) в точке x . Каждая из полученных оценок показывает, как «далеко» от нужного решения расположено найденное. Следовательно, для улучшения решения, требуется минимизировать данную оценку.

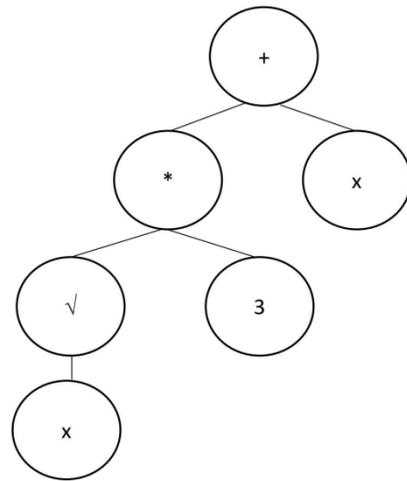


Рисунок 1. Представление решения в виде двоичного дерева.

Высоту дерева из-за соображений ограниченности ресурсов необходимо ограничивать. Так же это способствует оптимизации решения, чтобы не было переизбытка ветвей дерева.

III. МОДЕЛЬ КЛОНАЛЬНОГО АЛГОРИТМА ИИС

Для начала работы алгоритма требуется первичный набор лимфоцитов. Случайным образом генерируется популяция заданного размера. В нашем случае будут генерироваться случайные лимфоциты, которые в дальнейшем будут оптимизироваться.

Следующий этап – селекция, процесс отбора лучших кандидатов для «получения» нового поколения лимфоцит. Для этого необходимо число раз из всей популяции отбирается пара родителей, которая даёт потомство, одну особь. Выбираются два родителя, и происходит процесс скрещивания. Так же лучших особей старой популяции можно включить в следующую популяцию.

Скрещивание осуществляется по принципу кроссинговера. В биологии кроссинговер – процесс обмена участками гомологичных хромосом. В нашем случае, когда лимфоциты – бинарные деревья, необходимо получить наследника, т.е. комбинацию этих деревьев. Для удобства будем выбирать первый лимфоцит как основной, а от второго брать лишь часть и присоединять к первому в случайном удобном месте. Пример оператор «скрещивания» двух бинарных деревьев представлен на рисунке 2. В итоге получили новый лимфоцит, который попадет в популяцию следующего поколения [1-4,5-6].

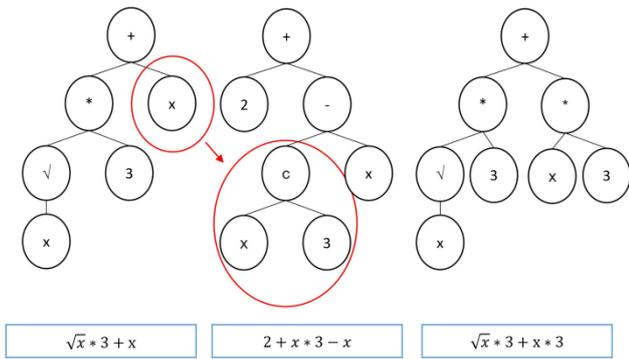


Рисунок 2. Пример работы «скрещивания» двух решений.

Следующий этап – мутация лимфоцита. Этот этап необходим для внесения разнообразия в популяцию и предотвращения «проваливания» решения в локальный максимум или минимум. Мутация происходит следующим образом: выбирается небольшое число лимфоцитов из популяции и проводится их изменение по наперед определенным правилам. В качестве мутации можно выбрать следующие изменения:

- замена одной простой функции в нелистовом узле на другую;
- удаление поддерева;
- создание нового поддерева;
- удаление функции, связанной с одним листовым узлом;
- перемещение двух значений в листовых узлах.

Пример проведения мутации представлен на рисунке 3.

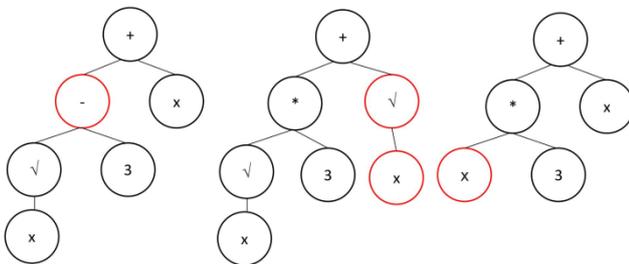


Рисунок 3. Замена функции в дереве, добавление поддерева, удаление функции.

Следующей процедурой является поддержание постоянного числа особей в популяции. Добавляются недостающие лимфоциты - «свежее пополнение» которые так же обеспечивают разнообразие. Либо удаляются «лишние» лимфоциты, наиболее неудачные. И повторяется процесс отбора лучших вариантов с последующим скрещиванием.

Цикл прекращается, если достигнут результат вычислений, либо достигнуто максимальное число поколений, ведь не всегда можно найти удовлетворительный результат за приемлемое число итераций. Лучший результат является ответом. Его требуется перевести в строчное представление и оптимизировать по возможности.

Весь алгоритм представлен на рисунке 4.



Рисунок 4. Схема работы алгоритма иммунной Системы

Для успешного создания web-приложения необходимо было определиться с набором технологий. Прежде всего был выбран язык программирования Java т.к. он очень удобен для написания web- приложений. Для создания серверной части использовался Spring Framework, для клиентской – VueJs.

Для написания кода была выбрана среда разработки IntelliJ IDEA из-за возможности получения удобного интерфейса, множества встроенных плагинов и возможностей, поддержки языка Java и Spring Framework.

Запуск web-приложения можно произвести на любой ЭВМ с установленной средой разработки IntelliJ IDEA, установленной JVM и базой данных PostgreSQL.

IV. МОДЕЛЬ ДАННЫХ

На рисунке 5 представлена схема база данных web-приложения, которая состоит из следующих сущностей [14-15]:

- Auth_user – таблица в которой хранятся данные о зарегистрированных пользователях приложения;
- User_role – таблица в которой хранятся роли всех пользователей приложения;
- Report – таблица предназначенная для хранения различных жалоб от пользователей приложения;

- `Algorithm_parameters` – таблица для хранения набора для конкретной задачи;
- `Solutions` – таблица необходимая для хранения полученного решения сохраненные решения.

На рисунке 6 представлена диаграмма пакетов, которые используются в программе.

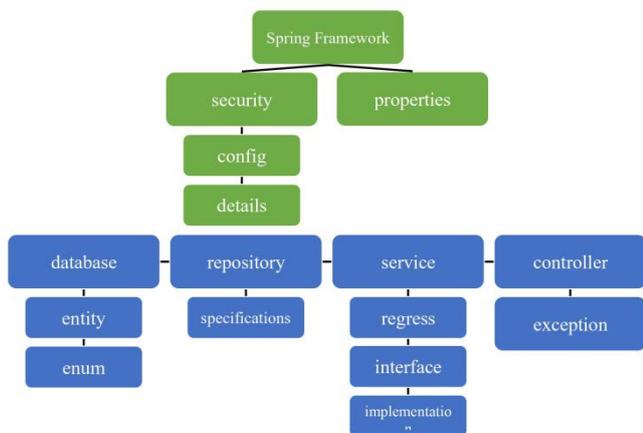


Рис. 6. Диаграмма пакетов

В пакете `security` расположены пакет `config`, содержащий классы с параметрами настройки `Basic Spring Security`, которые позволяют настроить защиту приложения на уровне методов и не описывать все паттерны доступа к разделам web-приложения. Так же в пакете `security` расположен пакет `details`, в котором описан класс для предоставления данных о существующих пользователях `Spring security`.

В пакете `properties` расположен файл с параметрами подключения к СУБД `PostgreSQL`.

Пакет `database` содержит классы описывающие объекты, помеченные как `@Entity`, о которых позже, и объекты типа `enum` (перечисления).

В пакете `repository` расположены интерфейсы для взаимодействия с базой данных с помощью встроенных средств – `Repository`. Так же данный пакет содержит пакет `specification`, в котором расположены классы для настройки критериев – объектов для поиска, и спецификации, о которых более подробно будет сказано позже.

В пакете `service` расположены классы, отвечающие за основную логику приложения. Так в пакете `regress` расположены классы для решения задачи символьной регрессии. А в пакетах `interface` и `implementation` расположены интерфейсы сервисов и их реализации соответственно.

Пакет `controller` содержит классы, описывающие `rest`-контроллеры, и пакет `exception`, в котором содержатся классы для создания исключений, которые могут возникнуть при работе приложения.

Для взаимодействия с таблицами в базе данных созданы `Java` классы, помеченные аннотацией `@Entity`. Общий класс для всех сущностей и класс пользователя.

`@MappedSuperclass` – аннотация обозначающая, что данный класс не является сущностью, но содержит поля, которые может унаследовать класс-сущность. Необходима для создания обобщенного класса, который

содержит только поле идентификатора. Создан, чтобы каждый раз не описывать данное поле, а так же, чтобы можно было создать обобщенный класс репозитория.

При создании классов, аннотированных как сущности, был создан абстрактный класс (см. Приложение А), содержащий только идентификаторы и имплементирующий `Serializable`. Все остальные сущности его наследуют.

`Hibernate` при запуске web-приложения, проверяет все сущности и таблицы в базе данных на соответствие имен таблиц и наименований полей классов и таблиц, если возникают решаемые проблемы – решает их (добавляет и обновляет таблицы, колонки).

Для обеспечения взаимодействий с базой данных были созданы `Repositories`, которые позволяют обеспечить выполнение `CRUD` операций.

Фрэймворк `Spring Data` предоставляет удобный для использования интерфейс `JpaRepository`, который позволяет настроить доступ к определенной таблице через интерфейс репозитория. Реализовывать методы в явном виде не требуется, а описание происходит с помощью заранее заданных правил и с помощью ключевых слов. Кроме репозитория используется его расширение – `PagingAndSortingRepository`, которое позволяет быстро настроить пагинацию и сортировку путем добавления в аргументы метода параметр `Pagination`.

`Service` – `Java` класс, в котором реализуется основная логика web-приложения, является связующим звеном между контроллером и базой данных.

Для удобства использования был создан абстрактный сервис, который реализует обобщенные методы, схожие для всех сущностей (`CRUD` операции).

Для обработки `HTTP`-запросов с клиентской части приложения служат контроллеры. В приложении используется `REST` – стиль архитектуры приложения, при реализации которого не передаются состояния объектов. `Rest`-контроллеры передают объекты в формате `Json` или `xml`.

V. ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

При реализации интерфейса приложения был выбран ролевой подход к предоставлению доступа к возможностям сайта. Таким образом неавторизованный пользователь web-приложения имеет доступ к урезанным возможностям приложения. На главной странице в окне ввода параметров можно ввести набор точек, соблюдая шаблон: аргументы перечисляются через запятую, ставится запятая и записывается значение функции в этой точке, ставится точка с запятой и записывается следующая точка. После нажатия кнопки рассчитать происходят вычисления на стороне сервера и выдается строковое представление решения. Данный процесс изображен на рисунке 7.

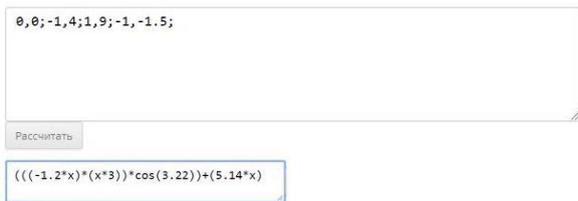


Рисунок 7. Интерфейс неавторизованного пользователя

Каждый неавторизованный пользователь может перейти на страницу авторизации и войти под своим профилем, либо создать новый профиль, если у него его нет.

После успешного прохождения авторизации, авторизованный пользователь, так же, как и неавторизованный пользователь, получает возможность ввести точки в окне ввода точек.

Входные параметры	Высота дерева	Решение	Искомая функция	Полученная точность
1,1;2,8;3,27;-1,-1	3	$(\sin((x0 - 7.9)) + ((x0 \wedge 3.0) + \cos(x0)))$	x^3	0.0695
1,1;2,4;3,9;4,16;-1,1;	3	$((\cos(2.2) + (x0 * x0)) + \sqrt{\cos(1.9)})$	x^2	0.0445
-1,1;-2,4;-3,9;-4,16;-5,25;4,16;	3	$(\ln(\sqrt{7.4}) - (\sin(7.6) - (x0 * x0)))$	x^2	0.0804
1,1;-1,1;-2,16;2,16;3,81;-3,81;	3	$\sqrt{(\ln(x0) - (x0 \wedge 8))}$	x^4	0.0321
0,0.54;1,-1.41;2,-2.99;3,-3.65;4,-3.71;	3	$(\sin((x0 - 8.9)) - \sqrt{(x0 \wedge x0)})$	$\cos(x+1)-x$	0.0789
1,5;2,15;3,39;4,83;	3	$((0.9 \wedge x0) * \sqrt{x0}) + ((x0 + x0) * (0.9 + x0))$	x^3+x^2+3	0.7243

Рисунок 8. Результаты работы приложения

VI. ЗАКЛЮЧЕНИЕ

В ходе работы были спроектировано и создано web-приложение, реализующее решение задачи символьной регрессии. Основные технологии при разработке приложения:

- СУБД PostgreSQL;
- Spring Framework;
- Hibernate;
- Концепции REST и AJAX.

Были выполнены следующие задачи:

- проанализированы иммунные алгоритмы;
- реализован алгоритм на основе иммунной системы, решающий задачу символьной регрессии;
- создана структура базы данных, отвечающая требованиям приложения;
- реализовано взаимодействие с базой данных;
- реализована архитектура REST приложения;
- реализована возможность регистрации и авторизации в приложении;
- создан интерфейсы для авторизованного и неавторизованного пользователей;
- создан интерфейс редактирования профиля пользователя.

БИБЛИОГРАФИЯ

- [1] Astachova I., Ushakov S., Selemenev A., Hitskova J. The application of an artificial immune system for solving the identification problem // ITM Web of Conferences. – 2017. – Vol. 9. – Article number: 02003. <https://doi.org/10.1051/itmconf/20170902003>
- [2] Astachova I. F., Ushakov S. A., Shashkin A. I., Belyaeva N. V. The application of artificial immune system for parallel process of calculation and their comparison with existing methods // Journal of Physics: Conference Series. – 2019. – Vol. 1202, No. 1. – Article number: 012003. <https://doi.org/10.1088/1742-6596/1202/1/012003>
- [3] Astachova I. F., Makoviy K. A., Khitskova Y. V. Possibilities for predicting the state of usability web resources // Journal of Physics: Conference Series. –2021. – Vol. 1902, No. 1. – Article number: 012029. <https://doi.org/10.1088/1742-6596/1742-6596/1902/1/012029>
- [4] Astachova I., Kiseleva E. The application of the artificial immune system for design, development and using of the hybrid system in education. In: Sukhomlin V., Zubareva E. (eds) Modern Information Technology and IT Education. SITTO 2017. Communications in Computer and Information Science, vol. 1204. – Cham: Springer, 2017. – P. 67-75. https://doi.org/10.1007/978-3-030-78273-3_7
- [5] Dasgupta D. (Ed.). Artificial immune systems and their applications. – Springer Science & Business Media, 2012.
- [6] Dasgupta D., Yu S., Nino F. Recent advances in artificial immune systems: models and applications // Applied Soft Computing. – 2012. – Vol. 11, issue 2. – P. 1574-1587. <https://doi.org/10.1016/j.asoc.2010.08.024>
- [7] Hart E., Timmis J. Application Areas of AIS: The Past, The Present and The Future. In: Jacob C., Pilat M.L., Bentley P.J., Timmis J.I. (eds) Artificial Immune Systems. ICARIS 2005. Lecture Notes in Computer

- Science, vol. 3627. Berlin, Heidelberg: Springer, 2005. P. 483-497.
https://doi.org/10.1007/11536444_37
- [8] Hart E., Timmis J. Application areas of AIS: The past, the present and the future // *Applied soft computing*. – 2008. – Vol. 8, issue 1. – P. 191-201. <https://doi.org/10.1016/j.asoc.2006.12.004>
- [9] Mamady D., Tan G., Toure M. L., Alfawaer Z. M. An Artificial Immune System Based Multi-Agent Robotic Cooperation. In: Sobh T., Elleithy K., Mahmood A., Karim M.A. (eds) *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*. – Dordrecht: Springer, 2008. – P. 60-67. https://doi.org/10.1007/978-1-4020-8737-0_12
- [10] Uy N. Q., Hoai N. X., O'Neill M., McKay R. I., Galván-López E. Semantically-based crossover in genetic programming: application to real-valued symbolic regression // *Genetic Programming and Evolvable Machines*. – 2011. – Vol. 12, No. 2. – P. 91-119. <https://doi.org/10.1007/s10710-010-9121-2>
- [11] Watkins A., Timmis J., Boggess L. Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm // *Genetic Programming and Evolvable Machines*. – 2004. – Vol. 5, No. 3. – P. 291-317. <https://doi.org/10.1023/B:GENP.0000030197.83685.94>
- [12] Joshi S., Borse M. Detection and Prediction of Diabetes Mellitus Using Back-Propagation Neural Network // 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE). – Ghaziabad, India: IEEE Computer Society, 2016. – P. 110-113. <https://doi.org/10.1109/ICMETE.2016.11>
- [13] Jayashree J., Kumar S. A. Linear Discriminant Analysis Based Genetic Algorithm with Generalized Regression Neural Network – A Hybrid Expert System for Diagnosis of Diabetes // *Programming and Computer Software*. – 2018. – Vol. 44. – P. 417-427. <https://doi.org/10.1134/S0361768818060063>

Creation of a web application that implements an algorithm for solving a symbolic regression problem based on an artificial immune system

Yuliya V. Khitskova

Abstract – The initial data for the symbolic regression problem is the set of free parameters and the corresponding set of function values of these parameters. In addition, the set of all functions allowed for superposition and restrictions for them, if they are required, are specified. It is worth choosing continuously differentiable functions as functions.

Each solution obtained must be evaluated. To do this, an objective function is determined, which determines the degree of approximation of the resulting solution to the expected results. Since the assessment uses given sets - points, various metrics, and errors can be used as the objective function. The superposition of the selected features is a chromosome and a potential solution and is represented as a binary tree. To estimate the survival rate of each chromosome, it is necessary to specify an objective function (survival function). It is also called a measure of affinity. One of the proposed functions has been selected.

One of the operators of the artificial immune system (AIS) is selection, the process of selecting the best candidates to “receive” a new generation of chromosomes. Also, the best individuals of the old population can be included in the next population.

The crossing is carried out according to the principle of crossing over. In biology, crossing over is the process of exchanging sections of homologous chromosomes. In our case, when the chromosomes are binary trees, it is necessary to obtain an heir, i.e. a combination of these trees.

Another operator is chromosome mutation. This stage is necessary to introduce diversity into the population and prevent the solution from falling into a local maximum or minimum. The entire algorithm of the immune system is given.

To successfully create a web application, you need to list the technologies. The Java programming language was chosen. The choice fell on PostgreSQL as the DBMS. Spring Framework was used to create the server part, and VueJS was used for the client part.

The data model and package structure are given. Testing has been carried out.

Keywords – artificial immune systems, crossover operator, mutation operator, affinity function, web application, testing.

REFERENCES

- [1] I. Astachova, S. Ushakov, A. Selemenev and J. Hitskova, "The application of an artificial immune system for solving the identification problem", *ITM Web of Conferences*, vol. 9, article number: 02003, 2017. <https://doi.org/10.1051/itmconf/20170902003>
- [2] I. F. Astachova, S. A. Ushakov, A. I. Shashkin and N. V. Belyaeva, "The application of artificial immune system for parallel process of calculation and their comparison with existing methods", *Journal of Physics: Conference Series*, vol. 1202, no. 1, article number: 012003, 2019. <https://doi.org/10.1088/1742-6596/1202/1/012003>
- [3] I. F. Astachova, K. A. Makoviy and Y. V. Khitskova, "Possibilities for predicting the state of usability web resources", *Journal of Physics: Conference Series*, vol. 1902, no. 1, Article number: 012029, 2021. <https://doi.org/10.1088/1742-6596/1742-6596/1902/1/012029>
- [4] I. Astachova and E. Kiseleva, "The application of the artificial immune system for design, development and using of the hybrid system in education". In: Sukhomlin, V., Zubareva, E. (eds) *Modern Information Technology and IT Education*. SITTITO 2017. *Communications in Computer and Information Science*, vol. 1204. Cham: Springer; pp. 67-75, 2017. https://doi.org/10.1007/978-3-030-78273-3_7
- [5] D. Dasgupta (Ed.), "Artificial immune systems and their applications". Springer Science & Business Media, 2012.
- [6] D. Dasgupta, S. Yu and F. Nino, "Recent advances in artificial immune systems: models and applications", *Applied Soft Computing*, vol. 11, issue 2, pp. 1574-1587, 2012. <https://doi.org/10.1016/j.asoc.2010.08.024>
- [7] E. Hart and J. Timmis, "Application Areas of AIS: The Past, The Present and The Future". In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds) *Artificial Immune Systems*. ICARIS 2005. *Lecture Notes in Computer Science*, vol. 3627. Berlin, Heidelberg: Springer, pp. 483-497, 2005. https://doi.org/10.1007/11536444_37
- [8] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future", *Applied soft computing*, vol. 8, issue 1, pp. 191-201, 2008. <https://doi.org/10.1016/j.asoc.2006.12.004>
- [9] D. Mamady, G. Tan, M. L. Toure and Z. M. Alfawaer, "An Artificial Immune System Based Multi-Agent Robotic Cooperation". In: Sobh, T., Elleithy, K., Mahmood, A., Karim, M.A. (eds) *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*. Dordrecht: Springer, pp. 60-67, 2008. https://doi.org/10.1007/978-1-4020-8737-0_12
- [10] N. Q. Uy, N. X. Hoai, M. O'Neill, R. L. McKay and E. Galván-López, "Semantically-based crossover in genetic programming: application to real-valued symbolic regression", *Genetic Programming and Evolvable Machines*, vol. 12, no. 2, pp. 91-119, 2011. <https://doi.org/10.1007/s10710-010-9121-2>
- [11] A. Watkins, J. Timmis and L. Boggess, "Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm", *Genetic Programming and Evolvable Machines*, vol. 5, no. 3, pp. 291-317, 2004. <https://doi.org/10.1023/B:GENP.0000030197.83685.94>
- [12] S. Joshi and M. Borse, "Detection and Prediction of Diabetes Mellitus Using Back-Propagation Neural Network", *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*. Ghaziabad, India: IEEE Computer Society, pp. 110-113, 2016. <https://doi.org/10.1109/ICMETE.2016.11>
- [13] J. Jayashree and S. A. Kumar, "Linear Discriminant Analysis Based Genetic Algorithm with Generalized Regression Neural Network – A Hybrid Expert System for Diagnosis of Diabetes", *Programming and Computer Software*, vol. 44, pp. 417-427, 2018. <https://doi.org/10.1134/S0361768818060063>

About the authors:

Yuliya V. Khitskova, Associate Professor of the Chair of Control Systems and Information Technologies in Construction, Faculty of Information Technology and Computer Safety, Cand. Sci. (Econ.), Associate Professor, Voronezh State Technical University, (email: prosvetovau@list.ru) ORCID: <https://orcid.org/0000-0001-6322-8633>