

21. Fedosov V.P., Al'-Musavi Visam Mokhammedtaki M. Dzhavad, Kucheryavenko S.V. Prostranstvenno-vremennoy adaptivnyy algoritm s ispol'zovaniem koda KHamminga na osnove modeli besprovodnogo kanala 3D-MIMO [Spatio-temporal adaptive algorithm using the Hamming code based on the 3D-MIMO wireless channel model], *Radiotekhnika* [Journal Radioengineering], 2024, Vol. 88, No. 2, pp. 113-123.
22. Fedosov V., Jameel J., Kucheryavenko S. Medical Image Transmission in 3D WiMAX Channel Using Adaptive Algorithm Based on MIMO-OFDM Principles, *Conference Proceedings - 2023 Radiation and Scattering of Electromagnetic Waves, RSEMW 2023*, 2023, pp. 236-239.
23. Fedosov V., Lomakina A., Legin A., Voronin V. Modeling of systems wireless data transmission based on antenna arrays in underwater acoustic channels, *Book Modeling of systems wireless data transmission based on antenna arrays in underwater acoustic channels. International Society for Optics and Photonics*, 2016, pp. 98720G.
24. Kucheryavenko A., Fedosov V. Model of multicomponent micro-Doppler signal in environment MATLAB, *MATEC Web of Conferences*, 2017, pp. 05008.
25. Fedosov V., Legin A. Wireless Data Transmission in Underwater Hydroacoustic Environment Based on MIMO-OFDM System and Application Adaptive Algorithm at the Receiver Side, *Serbian journal of electrical engineering*, February 2019, Vol. 16, No. 1, pp. 71-83.

Федосов Валентин Петрович – Южный федеральный университет; e-mail: vpfedosov@sfedu.ru; г. Таганрог, Россия; кафедра теоретических основ радиотехники; профессор.

Аль-Мусави Висам Мохаммедтаки М. Джавад – Южный федеральный университет; e-mail: almusavi@sfedu.ru; г. Таганрог, Россия; тел.: +78634371632; кафедра теоретических основ радиотехники; аспирант.

Кучерявенко Светлана Валентиновна – Южный федеральный университет; e-mail: svkucheryavenko@sfedu.ru; г. Таганрог, Россия; кафедра теоретических основ радиотехники; доцент.

Fedosov Valentin Petrovich – Southern Federal University; e-mail: vpfedosov@sfedu.ru; Taganrog, Russia; the Department of Theoretical Foundations of Radio Engineering; professor.

Al-Musawi Wisam Mohammedtaqi M. Jawad – Southern Federal University; e-mail: almusavi@sfedu.ru; Taganrog, Russia; phone: +78634371632; the Department of Theoretical Foundations of Radio Engineering; postgraduate student.

Kucheryavenko Svetlana Valentinovna – Southern Federal University; e-mail: svkucheryavenko@sfedu.ru; Taganrog, Russia; the Department of Theoretical Foundations of Radio Engineering; associate professor.

УДК 004.832.23

DOI 10.18522/2311-3103-2025-1-45-56

С.И. Родзин, А.И. Дерменжи

ИНИЦИАЛИЗАЦИЯ РЕШЕНИЙ В ПОПУЛЯЦИОННЫХ АЛГОРИТМАХ НА ОСНОВЕ МЕТОДА МЕТРОПОЛИСА–ГАСТИНГСА

Наиболее важными задачами принятия оптимальных решений с использованием эвристических алгоритмов считаются повышение точности и предотвращение преждевременной сходимости. Большинство исследований в этом направлении сосредоточено на разработке новых операторов, настройке параметров популяционной метаэвристики и гибридизации нескольких стратегий поиска решений. Гораздо меньше внимания уделяется инициализации – важной операции в популяционных алгоритмах, которая связана с созданием исходной популяции решений. Предлагается новый подход к инициализации популяции для эвристических алгоритмов. При формировании множества начальных решений предлагается использовать метод Метрополиса–Гастингса. В соответствии с этим методом исходные решения в популяции принимают значения, близкие к глобальному или локальным оптимумам целевой функции. Это позволяет повысить точность получаемых решений. Чтобы продемонстрировать возможности предлагаемого подхода к инициализации, он был встроен в базовый алгоритм дифференциальной эволюции. Для оценки эффективности стратегии проведена экспериментальная проверка путем сравнения с такими известными методами как случайная инициализация, обучение на основе методов оппозиции и хаоса, а также метода диагонального равномерного распределения. Сравнение проводилось на репрезен-

тативном наборе мультимодальных, унимодальных и гибридных функций, включая функцию Расстригина, Квинга, Розенброка, Швевеля, квинтовую, ступенчатую, сферическую. Анализировались скорость сходимости алгоритмов и точность получаемых решений. В качестве показателей сравнения использовались среднее значение по лучшим решениям, медианное лучшее решение, стандартное отклонение от лучшего решения, количество вызовов функций, коэффициент успешности, коэффициент ускорения. Значения показателей усреднялись по результатам 30 отдельных запусков каждого алгоритма. Предлагаемый алгоритм работает быстрее, показывает лучшую сходимость и точность. Алгоритм дает лучшие результаты, поскольку стратегия инициализации позволяет выбирать перспективные решения, близкие к локальным или глобальным оптимумам. Статистическая проверка результатов работы алгоритмов по критерию Фридмана подтвердила, что предлагаемый подход к инициализации популяции решений обеспечивает лучший баланс скорости сходимости/точность решений.

Метаэвристика; глобальная оптимизация; дифференциальная эволюция; инициализация популяции; агент; многомерная функция; метод Метрополиса–Гастингса; критерий Фридмана.

S.I. Rodzin, A.I. Dermenzhi

INITIALIZATION OF SOLUTIONS IN POPULATION METAHEURISTICS BASED ON THE METROPOLIS–HASTINGS METHOD

The most important tasks of making optimal decisions using heuristic algorithms are considered to be improving accuracy and preventing premature convergence. Most of the research in this area focuses on the development of new operators, tuning the parameters of population metaheuristics, and hybridization of several solution search strategies. Much less attention is paid to initialization, an important operation in population algorithms that involves creating an initial population of solutions. A new approach to population initialization for heuristic algorithms is proposed. When forming a set of initial solutions, it is proposed to use the Metropolis–Hastings method. According to this method, the initial solutions in the population take values close to the global or local optima of the objective function. This makes it possible to increase the accuracy of the solutions obtained. To demonstrate the possibilities of the proposed initialization approach, it was integrated into the basic differential evolution algorithm. To assess the effectiveness of the strategy, an experimental test was carried out by comparing it with such well-known methods as random initialization, training based on opposition and chaos methods, as well as the method of diagonal uniform distribution. The comparison was carried out on a representative set of multimodal, unimodal, and hybrid functions, including Rastrigin, Quing, Rosenbrock, Schwefel, quintal, step, and spherical functions. The convergence rate of the algorithms and the accuracy of the obtained solutions were analyzed. The average value for the best solutions, the median best solution, the standard deviation from the best solution, the number of function calls, the success rate, and the acceleration coefficient were used as comparison indicators. The values of the indicators were averaged based on the results of 30 separate runs of each algorithm. The proposed algorithm works faster, shows better convergence and accuracy. The algorithm gives the best results because the initialization strategy allows you to choose promising solutions that are close to local or global optima. Statistical verification of the results of the algorithms using the Friedman criterion confirmed that the proposed approach to initializing a population of solutions provides a better balance of convergence rate/accuracy of solutions.

Metaheuristics; global optimization; differential evolution; population initialization; agent; multi-dimensional function; Metropolis–Hastings method; Friedman criterion.

Введение. Главное преимущество классических методов оптимизации заключается в том, что они гарантируют нахождение наилучшего из возможных решений задачи оптимизации. Однако их использование ограничено такими требованиями к оптимизируемой функции как дифференцируемость и унимодальность.

Альтернативой классическим методам оптимизации являются алгоритмы, основанные на стохастических принципах с использованием популяции поисковых агентов, которые выступают в качестве возможных решений. Целевая функция оптимизации оценивает пригодность каждого возможного решения. Алгоритмы, использующие эвристические правила для генерации шаблонов поиска, дают лучшие результаты, чем классические подходы, с точки зрения скорости, надежности и точности. Эвристические правила, обычно, соответствуют моделям, инспирированным природными механизмами и системами [1].

На вопрос о существовании эвристического алгоритма, который дает всегда лучшие результаты для задач принятия оптимальных решений давно дан отрицательный ответ – такого лучшего алгоритма не существует. Согласно *NFL*-теореме не существует эвристического алгоритма (поиска или оптимизации), который "работает" лучше других на всем множестве задач [2]. Теорема утверждает, что любые два эвристических алгоритма оптимизации эквивалентны, если их производительность усредняется по всем возможным задачам. Если некоторый алгоритм работает лучше (быстрее, точнее) на одних задачах, значит, на других задачах он будет хуже. Из *NFL*-теоремы следует, что алгоритмы имеют различную эффективность при решении разных задач.

В последнее время в литературе было предложено много эвристических алгоритмов. Среди них наиболее популярными и часто цитируемыми алгоритмами являются дифференциальная эволюция (*DE*), генетический алгоритм (*GA*), искусственная иммунная система (*AIS*), гравитационный поиск (*GSA*), имитация отжига (*SA*), колония социальных пауков (*SSO*), роя частиц (*PSO*), саранчи (*LSA*), колония муравьев (*ACO*), пчел (*ABC*), пауков (*SCA*) поиск кукушки (*CS*) [3].

Основной особенностью этих популяционных алгоритмов является относительно невысокая вычислительная сложность, поскольку поиск может быть реализован с помощью нескольких операций в условиях нелинейности и мультимодальности целевых функций. Это мотивировало использование алгоритма *DE* для решения многих практических задач, включая машинное обучение, инженеррию и финансы [4]. В частности, при проектировании аэродинамических форм самолетов алгоритм *DE* эффективно оптимизирует параметры профиля крыла; при обучении нейронных сетей оптимизирует веса и смещения сети, чтобы модель точно отражала закономерности в данных; при разработке моделей поведения неигровых персонажей *DE* совершенствует принятие решений для создания сложных и динамичных противников.

За последние два десятилетия предложено несколько модификаций алгоритма *DE* с целью улучшения его поисковых возможностей [5]. Хотя модификации *DE* обладают лучшими характеристиками, чем оригинальный *DE*, в данном исследовании он был выбран в качестве базового алгоритма для совместимости с другими публикациями.

Основными недостатками эвристических алгоритмов являются преждевременная сходимость и длительное время вычислений, особенно когда область поиска трудно исследовать. Поэтому двумя наиболее важными задачами эвристической оптимизации считаются повышение точности и предотвращение преждевременной сходимости. Причем большая часть исследований в этом направлении сосредоточена на разработке новых операторов, настройке параметров алгоритмов и сочетании нескольких стратегий поиска [1]. Гораздо меньше внимания уделяется инициализации – важной операции, которая связана с созданием исходной популяции решений. При отсутствии информации о целевой функции наиболее распространенным способом инициализации является генерация случайных решений в пределах допустимого диапазона при том, что соответствующая инициализация возможных решений могла бы повысить точность, увеличить скорость сходимости, избегая застревания в локальных оптимумах. Тем не менее в этом направлении было предпринято несколько попыток.

В [6] исходная популяция решений создается с использованием последовательности равномерно распределенных квазислучайных точек, а не псевдослучайных чисел. В [7] представлен подход, основанный на создании двух оппозиционных популяций. В [8] предложена концепция адаптивной случайности, когда новое решение добавляется к исходной популяции только в том случае, если оно не слишком близко к остальным, уже содержащимся в исходной популяции. В [9] представлен способ инициализации, основанный на хаотических последовательностях.

Целью инициализации является выбор перспективных решений. Однако большинство методов инициализации, генерирующих решения с использованием случайных и хаотических принципов, тем не менее, зачастую приводят к получению решений, близких друг к другу, оставляя в стороне значительные неисследованные участки пространства поиска. С другой стороны, оппозиционные подходы, которые приводят к получению равноудаленных решений, как правило, образуют разрозненную популяцию вдали от перспективных регионов, что увеличивает число итераций алгоритма и снижает скорость сходимости.

Необходим подход к инициализации, позволяющий генерировать перспективные решения, тем самым повышая производительность эвристического алгоритма. Воспользуемся для этих целей методом Метрополиса–Гастингса (*МН*) [10]. Он позволяет генерировать популяцию решений таким образом, чтобы перспективное значение целевой функции $f(x)$ соответствовало высокой вероятности выбора решения x . Основная идея метода *МН* заключается в том, чтобы производить частую выборку решений x с перспективным значением $f(x)$, избегая выборки из неперспективных областей.

В статье предлагается новый подход к инициализации популяции с использованием метода *МН*. Подход позволяет формировать исходную популяцию из наиболее перспективных областей пространства поиска глобального оптимума, что существенно сокращает число требуемых итераций и ускоряет сходимость алгоритма. Возможности подхода к инициализации с использованием метода *МН* иллюстрируются на примере базового алгоритма дифференциальной эволюции и репрезентативного множества тестовых функций. Экспериментальные результаты показывают, что предложенный подход обеспечивает лучшую скорость сходимости и повышает точность решения по сравнению с другими методами инициализации в популяционных алгоритмах.

Метод Метрополиса–Гастингса. Метод *МН* соответствует методу Монте-Карло и основан на создании цепи Маркова, в которой на каждом шаге генерация случайного значения x^{k+1} из функции $f(x)$ зависит только от предыдущего значения x^k . С помощью вероятностного механизма выбранное значение принимается или отклоняется. Принятие или отклонение выбранного значения зависит от следующих обстоятельств.

Предположим, что нам нужно выполнить выборку переменных из функции $f(x)$, где x соответствует некоторому значению $f(x)$ в d -мерном пространстве. Затем в два этапа осуществляется переход цепи Маркова из состояния x^k в новое состояние x^{k+1} . Вначале выбирается значение-кандидат y путем агрегирования текущего x^k со случайным значением, полученным по закону нормального распределения $N(0, 1)$. После формирования выборки кандидатов используется вероятностный механизм для решения о том, принимается y или нет в качестве нового значения x^{k+1} .

Вероятность принятия выбранного значения-кандидата y зависит от следующего соотношения:

$$\alpha(x^k, y) = \min \left\{ \frac{f(y)}{f(x^k)}, 1 \right\}. \quad (1)$$

В соответствии с этим вероятностным правилом генерируется случайное равномерно распределенное на интервале $(0, 1)$ число r . Если $r < \alpha$, то выборка y принимается в качестве нового состояния x^{k+1} , в противном случае, y отбрасывается и x^k остается в качестве нового состояния x^{k+1} .

В методе *МН* точка x с большим значением $f(x)$, имеет более высокую вероятность быть выбранной. Основная идея *МН* заключается в том, чтобы производить частую выборку перспективных точек x , избегая выборки областей с низким значением $f(x)$. Теоретическое объяснение и доказательство этого эффекта можно найти в [11].

Проиллюстрируем результаты выборки методом *МН* на примере следующей функции:

$$f(x) = \frac{1}{0,4\sqrt{2\pi}} e^{-\frac{x^2}{0,32}} + \frac{1}{0,2\sqrt{2\pi}} e^{-\frac{(x+2,5)^2}{0,08}} + \frac{1}{0,2\sqrt{2\pi}} e^{-\frac{(x-2,5)^2}{0,32}}. \quad (2)$$

Здесь $-4 \leq x \leq 4$, функция $f(x)$ имеет два глобальных максимума при $x = -2.5$ и $x = 2.5$ и один локальный максимум при $x = 0$. На рис. 1 представлены гистограммы выборок, полученных методом *МН*, для функции (2).

В этих результатах рассматриваются гистограммы для объема выборки 10 (рис. 1,а), для объема выборки 100 (рис. 1,б), для объема выборки 1000 (рис. 1,в) 1000 и для объема выборки 10000 (рис. 1,г). Из рис. 1,а видно, что, несмотря на небольшой объем выборки, равный 10, некоторые значения x связаны с максимальными значениями $f(x)$. По мере увеличения объема выборок, полученных с помощью *МН*, значения x , связанные с максимальными значениями $f(x)$, выбираются с более высокой частотой, согласно рис. 1,б-г.

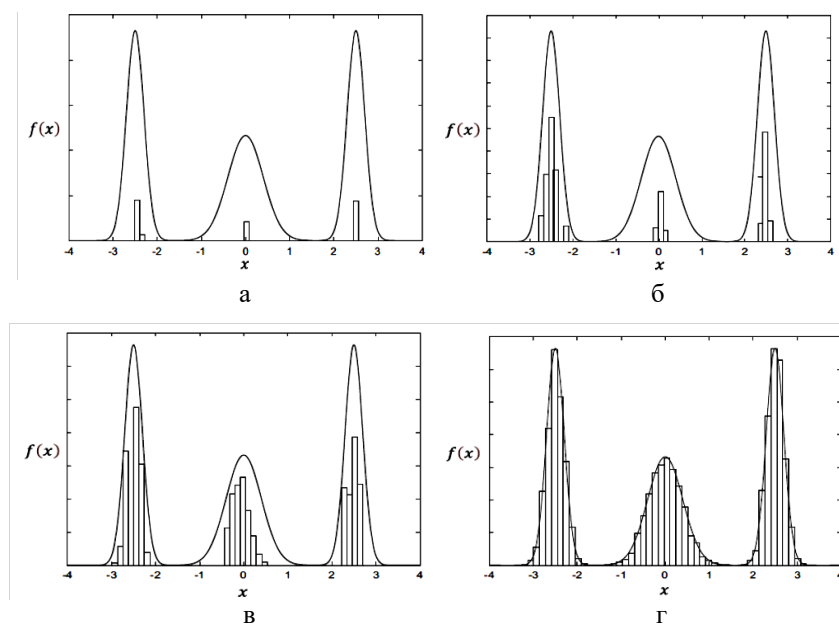


Рис. 1. Гистограммы выборок, полученных методом МН для функции $f(x)$ из уравнения (2):
 а – для объема выборки 10; б – для объема выборки 100; в – для объема выборки 1000;
 г – для объема выборки 10000

Базовый алгоритм дифференциальной эволюции. Алгоритм *DE*, являясь одним из наиболее цитируемых эвристических алгоритмов, имеет простую структуру, низкую вычислительную трудоемкость и неплохую сходимость [4]. Он был разработан для определения глобального решения задачи нелинейной оптимизации следующего вида:

Найти минимум функции

$$f(x), x \in D, \quad (3)$$

где D – область допустимых решений.

Популяционный алгоритм *DE* включает четыре операции: инициализацию, мутацию, кроссинговер и отбор.

На этапе инициализации случайно генерируется начальная совокупность из n агентов. На каждой итерации алгоритма операторами *DE* генерируется новое поколение агентов, причем число агентов в каждом поколении одно и то же, что является одним из параметров алгоритма.

Новое поколение агентов генерируется следующим образом. Для каждого агента v_i из предыдущего поколения случайно выбираются несколько агентов, за исключением самого v_i , и генерируется так называемый мутантный вектор. Наиболее популярными формулами для образования мутантного вектора являются следующие:

$$v = v_1 + F \cdot (v_2 - v_3), \quad (4)$$

$$v = v_1 + F \cdot (v_2 - v_3) + F \cdot (v_4 - v_5), \quad (5)$$

$$v = v_{best} + F \cdot (v_1 - v_2), \quad (6)$$

где v_1, v_2, v_3, v_4, v_5 – это различные решения, случайно выбранные из текущей популяции, v_{best} – наилучшее решение в популяции, F – коэффициент масштабирования мутации (положительное вещественное число в диапазоне от 0 до 1).

Далее, над мутантным вектором v выполняется операция «скрещивания» (кроссинговер), состоящая в том, что некоторые его координаты замещаются соответствующими координатами из v_i (каждая координата замещается с некоторой вероятностью, которая также является еще одним параметром алгоритма *DE*). Полученный после скрещивания

вектор называется пробным вектором u_i . Если u_i оказывается лучше v_i (т.е. значение целевой функции стало меньше), то в новом поколении пробный вектор заменяет v_i , а в противном случае v_i остаётся без изменений.

Наконец, в *DE* используется механизм селекции для отбора лучших решений и формирования следующего поколения.

Этот процесс повторяется до тех пор, пока не будет достигнут критерий останова.

Ниже представлен предлагаемый алгоритм инициализации популяции решений на основе метода *MH*.

Алгоритм инициализации популяции на основе метода *MH*. В большинстве задач оптимизации априорная информация о местоположении глобального оптимума в виде областей в пространстве решений, внутри которых – точно или приближённо – должно находиться оптимальное решение, отсутствует. Поэтому она не может быть использована для формирования исходной популяции решений близких к оптимальным значениям, что влияет на скорость сходимости алгоритма.

Псевдокод алгоритма инициализации популяции начальных решений на основе метода *MH* имеет следующий вид:

1. Ввод n, X
2. $I = \{\emptyset\}, i = 1$
3. Случайно генерируется $x_0 \in X$
4. **While** ($i \leq n$)
5. $y = x_{i-1} + N(0, 1)$
6. $\alpha = \min \left\{ \frac{f(y)}{f(x_{i-1})}, 1 \right\}$
7. Генерируется случайное равномерно распределенное на интервале $(0, 1)$ число r
8. **If** ($r < \alpha$) **then**
9. $x_i = y$
10. $I = \{I \cup x_i\}$
11. $i = i + 1$
12. **End if**
13. **End while**
14. Вывод I .

Входными данными здесь являются размер популяции n и ограничения D из области поиска X . Вначале запускается установка множества исходных решений I как пустого множества и индексной переменной $i = 1$ (строка 2). Затем случайно генерируется первое начальное значение x_0 в пространстве X (строка 3). Далее, начинается циклический процесс получения всех n исходных решений-кандидатов в популяции алгоритма (строка 4). Во-первых, создается новое решение-кандидат y путем модификации предыдущего решения x_{i-1} с помощью чисел, распределенных по нормальному закону $N(0, 1)$ (строка 5). Затем вычисляется вероятность α принятия созданного решения-кандидата y путем сравнения $f(y)$ и $f(x_{i-1})$ (строка 6). После этого генерируется случайное равномерно распределенное на интервале $(0, 1)$ число r (строка 7). Если $r < \alpha$, то в качестве нового решения x_i принимается y (строка 9), а x_i добавляется в формируемую на этапе инициализации исходную популяцию решений I , счетчик i увеличивается на единицу (строки 10-11). Процесс повторяется до тех пор, пока не будет сгенерирована вся исходная популяция из n начальных решений.

С целью демонстрации возможностей предлагаемого алгоритма инициализации популяции решений он был встроен в базовый алгоритм *DE* и проведены эксперименты.

Результаты экспериментов. В ходе экспериментов сопоставлялись результаты, полученные с помощью предлагаемого алгоритма инициализации популяции, с результатами, полученными при использовании конкурирующих подходов к инициализации, таких как случайная инициализация (*RI*) [12], обучение на основе оппозиции (*OP*) [13], метод, основанный на хаосе (*CM*) [14] и метод диагонального равномерного распределения (*DU*) [15].

Эксперименты проводились в программной среде на языке программирования C# с использованием среды программирования приложений *Microsoft Visual Studio*. Это позволило воспользоваться возможностями, предоставляемыми объектно-ориентированным подходом в разработке программного обеспечения, а также максимально использовать фреймворк *Windows Forms* (зарегистрирована программа для ЭВМ, запись № 2024690925 от 18.12.2024 г.). При отладке и тестировании использован компьютер *IBM PC* с процессором *Core i7* и ОЗУ 8 Гб.

Все конкурирующие алгоритмы инициализации также были встроены в базовый алгоритм *DE* с формулой (6) для образования мутантного вектора, с размером популяции $n = 50$ и рекомендуемыми для них наилучшими настройками таких параметров как вероятность кроссинговера $CR = 0,9$ и весовой параметр $F = 0,8$.

План экспериментов предусматривал следующие оценки:

- ♦ точность получаемых решений;
- ♦ скорость и надежность сходимости алгоритма;
- ♦ статистический анализ результатов;
- ♦ влияние размеров популяции на сходимость алгоритмов.

Существует около трех десятков различных тестовых функций оптимизации. Для экспериментов были выбраны восемь функций, включая функции Растригина, Квинга, Розенброка, Швепеля, квинтовую, ступенчатую, сферическую. Из них пять функций являются мультимодальными ($f_1(x) - f_5(x)$), две унимодальные функции ($f_6(x) - f_7(x)$) и одна гибридная функция ($f_8(x)$) [16]:

- ♦ $f_1(x) = \sum_{i=1}^d (x_i^2 - i)^2, x_i \in [-1,28; 1,28]^d, f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_2(x) = \sum_{i=1}^d |x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4|, x_i \in [-5,12; 5,12]^d;$
 $f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_3(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)], x_i \in [-5; 10]^d;$
 $f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_4(x) = \sum_{i=1}^d [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], x_i \in [-100; 100]^d;$
 $f_{min}(x^*) = 0, x^* = (0,5, \dots, 0,5);$
- ♦ $f_5(x) = \sum_{i=1}^d |x_i^2|, x_i \in [-100; 100]^d, f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_6(x) = \sum_{i=1}^d i x_i^2, x_i \in [-10; 10]^d, f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_7(x) = \sum_{i=1}^d |x_i|^{i+1}, x_i \in [-1; 1]^d, f_{min}(x^*) = 0, x^* = (0, \dots, 0);$
- ♦ $f_8(x) = [10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]] + [\sum_{i=1}^d |x_i| + \sum_{i=1}^d |x_i|] + + [\sum_{i=1}^d x_i^2];$
 $x_i \in [-100; 100]^d, f_{min}(x^*) = 0, x^* = (0, \dots, 0).$

Выбор тестовых функций объясняется не только их разнообразием, но также тем, что по ним известны результаты, полученные конкурирующими алгоритмами инициализации популяций.

В этом эксперименте предполагается, что каждая функция является многомерной ($d = 30$). Показателями, по которым проводилось сравнение, являлись среднее значение по лучшим решениям (av), медианное лучшее решение (med) и стандартное отклонение от лучшего решения (sd). Усредненные результаты по 30 отдельным запускам каждого *DE* алгоритма, приведены в табл. 1.

Показатели av и med соответствуют точности решений, а sd – их дисперсии. Согласно табл. 1, алгоритм *DE-MH* дает лучшие результаты для av в 5 функциях из 8 и для med в 6 функциях из 8. Это означает, что предлагаемая стратегия инициализации обеспечивает лучшую точность примерно для 70% функций $f_i(x) - f_8(x)$.

При оценке влияния методов инициализации на скорость сходимости метаэвристического алгоритма учитывались следующие показатели: количество вызовов функций (cf), коэффициент успешности (sr) и коэффициент ускорения (ac).

Таблица 1

Результаты сравнения подхода к инициализации на основе *MH* с методами инициализации *RI, OP, CM, DU* в алгоритме *DE*

Функция		<i>DE-RI</i>	<i>DE-OP</i>	<i>DE-CM</i>	<i>DE-DU</i>	<i>DE-MH</i>
<i>f₁</i>	<i>av</i>	8,7512	8,6613	8,5658	8,8777	8,5263
	<i>med</i>	$2,34 \cdot 10^{-04}$	$8,61 \cdot 10^{-05}$	$1,72 \cdot 10^{-04}$	$2,08 \cdot 10^{-04}$	$1,19 \cdot 10^{-09}$
	<i>sd</i>	2,64	0,81	0,14	0,21	0,0036
<i>f₂</i>	<i>av</i>	1,6039	1,5732	1,5322	1,5935	1,5282
	<i>med</i>	$2,29 \cdot 10^{-04}$	$4,08 \cdot 10^{-04}$	$4,73 \cdot 10^{-04}$	$2,30 \cdot 10^{-04}$	$7,77 \cdot 10^{-05}$
	<i>sd</i>	0,0007	0,0346	0,0003	0,0007	$4,58 \cdot 10^{-05}$
<i>f₃</i>	<i>av</i>	31,667	30,360	31,594	31,091	30,293
	<i>med</i>	155,13	150,04	150,06	154,22	149,58
	<i>sd</i>	26,675	28,360	37,090	26,0004	18,237
<i>f₄</i>	<i>av</i>	10,545	10,059	11,024	7,8339	0,874
	<i>med</i>	24,793	23,807	24,400	24,004	22,888
	<i>sd</i>	14,169	15,366	14,463	13,001	10,574
<i>f₅</i>	<i>av</i>	-115,3	-115,8	-110,8	-114,1	-114,8
	<i>med</i>	0	0	0	0	0
	<i>sd</i>	0,2061	0,6621	0,2467	0,2011	0,8653
<i>f₆</i>	<i>av</i>	$5,89 \cdot 10^{-11}$	$4,23 \cdot 10^{-10}$	$9,69 \cdot 10^{-11}$	$5,57 \cdot 10^{-11}$	$7,34 \cdot 10^{-11}$
	<i>med</i>	$2,86 \cdot 10^{-14}$	$2,43 \cdot 10^{-14}$	$2,58 \cdot 10^{-14}$	$2,65 \cdot 10^{-14}$	$4,87 \cdot 10^{-15}$
	<i>sd</i>	$5,21 \cdot 10^{-14}$	$4,99 \cdot 10^{-14}$	$6,33 \cdot 10^{-14}$	$4,99 \cdot 10^{-14}$	$2,35 \cdot 10^{-14}$
<i>f₇</i>	<i>av</i>	$5,59 \cdot 10^{-07}$	$4,86 \cdot 10^{-07}$	$4,87 \cdot 10^{-07}$	$4,77 \cdot 10^{-07}$	$9,83 \cdot 10^{-06}$
	<i>med</i>	$5,79 \cdot 10^{-16}$	$5,77 \cdot 10^{-15}$	$3,28 \cdot 10^{-15}$	$5,55 \cdot 10^{-16}$	$3,32 \cdot 10^{-15}$
	<i>sd</i>	$3,16 \cdot 10^{-10}$	$8,94 \cdot 10^{-10}$	$4,57 \cdot 10^{-10}$	$3,83 \cdot 10^{-10}$	$3,96 \cdot 10^{-09}$
<i>f₈</i>	<i>av</i>	35,824	36,357	33,624	33,252	26,989
	<i>med</i>	$5,12 \cdot 10^{-07}$	$4,03 \cdot 10^{-09}$	$4,61 \cdot 10^{-07}$	$4,79 \cdot 10^{-07}$	$8,17 \cdot 10^{-09}$
	<i>sd</i>	$3,62 \cdot 10^{-07}$	$1,18 \cdot 10^{-09}$	$1,68 \cdot 10^{-07}$	$2,79 \cdot 10^{-07}$	$6,87 \cdot 10^{-09}$

Количество вызовов функций *cf* представляет собой количество обращений к целевой функции во время работы алгоритма. Значения *cf* усреднялись по результатам 30 отдельных запусков каждого *DE* алгоритма. Меньшее значение *cf* означает, что оцениваемый метод имеет лучшую скорость сходимости. Коэффициент успешности *sr* представляет собой частоту успешных выполнений алгоритма и определяется следующим образом:

$$sr = \frac{\text{Число успешных выполнений алгоритма}}{\text{Общее число запусков алгоритма}}. \quad (7)$$

Коэффициент ускорения (*ac*) сравнивает количество вызовов функций *cf* алгоритмов *A₁* и *A₂* следующим образом:

$$cr_{A_1/A_2} = \frac{cf_{A_1}}{cf_{A_2}}. \quad (8)$$

Если $cr_{A_1/A_2} > 1$, то это означает, что алгоритм *A₂* работает быстрее, чем *A₁*.

В табл. 2 представлены численные результаты показателей cf и sr для функций $f_1(x)$, $f_2(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$.

Таблица 2

Результаты показателей cf и sr для функций $f_1(x)$, $f_2(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$

	<i>DE-RI</i>		<i>DE-OP</i>		<i>DE-CM</i>		<i>DE-DU</i>		<i>DE-MH</i>	
	cf	sr	cf	sr	cf	sr	cf	sr	cf	sr
f_1	13,34	0,43	13,72	0,27	13,15	0,50	17,0	0,35	13,61	0,77
f_2	-	-	763,4	-	-	-	-	0,41	263,7	0,68
f_5	10,36	0,72	9,78	0,70	11,47	0,72	11,11	0,54	7,533	0,91
f_6	20,92	1	21,42	1	21,584	1	20,438	1	18,512	1
f_7	688,34	1	600,7	1	688,32	1	663,38	1	524,36	1
f_8	40,94	1	40,50	1	40,936	1	43,552	1	31,825	1

Анализ табл. 2 показывает, что на множестве тестовых функций $f_1(x)$, $f_2(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$ *DE-MH* превосходит методы инициализации *DE-RI*, *DE-OP*, *DE-CM*, *DE-DU* по показателю коэффициент успешности и на множестве функций $f_2(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$ – по показателю количество вызовов функций. Это означает, что предлагаемый метод инициализации позволяет быстрее получить результат, нежели конкурирующие методы. Несмотря на то, что показатель cf метода *DE-CM* для функции $f_1(x)$ лучше, чем *DE-MH*, однако он уступает *DE-MH* по показателю sr . Символ дефиса «-» означает, что алгоритм не смог успешно достичь решения, равного или меньшего 10^{-6} .

В табл. 3 представлены численные результаты для коэффициента ускорения cr при поиске оптимума функций $f_1(x)$, $f_2(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$.

Таблица 3

Результаты показателя cr для функций $f_1(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$ при сравнении количества вызовов функций алгоритмов *DE-RI*, *DE-OP*, *DE-CM* и *DE-DU* с алгоритмом *DE-MH*

Функция	cr			
	$cr_{DE-RI/DE-MH}$	$cr_{DE-OP/DE-MH}$	$cr_{DE-CM/DE-MH}$	$cr_{DE-DU/DE-MH}$
f_1	0,988	1,007	0,988	1,191
f_5	2,378	2,261	2,602	2,531
f_6	1,083	1,091	1,129	1,145
f_7	1,115	1,117	1,115	1,147
f_8	1,261	1,254	1,266	1,331

Из табл. 3 видно, что алгоритм *DE-MH* работает быстрее, нежели алгоритмы *DE-RI*, *DE-OP*, *DE-CM* и *DE-DU* и показывает лучшую сходимость на функциях $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$. Некоторым исключением является функция $f_1(x)$, в которой алгоритмы *DE-RI* и *DE-CM* дают несколько лучшие результаты.

Была проведена статистическая проверка результатов работы алгоритмов *DE-RI*, *DE-OP*, *DE-CM*, *DE-DU* и *DE-MH* по критерию Фридмана [17] по усредненным наилучшим значениям (av) из табл. 1. Согласно тесту Фридмана, методы ранжируются следующим образом: *DE-MH*, *DE-OP*, *DE-RI*, *DE-CM* и *DE-DU*. В соответствии с этим порядком *DE-MH* является наилучшим алгоритмом, тогда как *DE-DU* считается наихудшим подходом к инициализации. Механизмы инициализации *DE-CM*, *DE-DU*, *DE-RI* и *DE-OP* не могут одновременно увеличивать скорость сходимости и точность решений, в отличие от *DE-MH*, который обеспечивает лучший баланс скорость сходимости/точность решений.

В [18] отмечалось, что на производительность алгоритма *DE* влияет размер популяции, увеличивая количество итераций, необходимых для достижения глобального оптимума. С другой стороны, небольшое количество элементов в популяции провоцирует преждевременную сходимость [19, 20].

Были проведены эксперименты по сравнению различных подходов к инициализации с точки зрения различных размеров популяции с целью оценки точности, получаемых решений. При сравнении рассматривались тестовые функции $f_2(x)$ и $f_6(x)$. Для оценки результатов использовались показатели *av*, *med* и *sd*. Результаты усреднялись по 30 запускам алгоритмов. В табл. 4 и 5 представлены результаты для размеров популяции $n = 100$ и $n = 200$ соответственно.

Таблица 4

Результаты сравнения алгоритма инициализации на основе МН с методами инициализации RI, OP, CM, DU для функций $f_2(x)$ и $f_6(x)$ при размере популяции $n = 100$

Функция		DE-RI	DE-OP	DE-CM	DE-DU	DE-MH
f_2	<i>av</i>	2,233	3,732	2,176	2,875	2,149
	<i>med</i>	1,465	2,018	2,294	3,140	0,007
	<i>sd</i>	2,022	3,907	1,634	1,875	0,175
f_6	<i>av</i>	0,001	$2,23 \cdot 10^{-09}$	0,0010	0,0012	$5,02 \cdot 10^{-07}$
	<i>med</i>	0,002	0,0009	0,0031	0,0026	$6,21 \cdot 10^{-08}$
	<i>sd</i>	0,001	0,0004	0,0007	0,0009 ³	$3,21 \cdot 10^{-06}$

Таблица 5

Результаты сравнения алгоритма инициализации на основе МН с методами инициализации RI, OP, CM, DU для функций $f_2(x)$ и $f_6(x)$ при размере популяции $n = 200$

Функция		DE-RI	DE-OP	DE-CM	DE-DU	DE-MH
f_2	<i>av</i>	2,4682	5,5055	3,4987	3,4000	3,021
	<i>med</i>	2,145	2,232	2,897	3,191	1,617
	<i>sd</i>	2,451	4,196	2,505	2,445	0,922
f_6	<i>av</i>	1,293	1,425	0,0010	1,277	1,428
	<i>med</i>	1,647	1,801	1,600	1,600	1,366
	<i>sd</i>	1,109	1,353	1,004	1,004	1,023

Согласно табл. 4 и 5 все алгоритмы приводят к ухудшению результатов из-за увеличения размеров популяции. Тем не менее, в сравнении с другими подходами к инициализации, предлагаемый алгоритм DE-MH дает лучшие результаты, поскольку стратегия инициализации позволяет выбирать перспективные решения, близкие к локальным или глобальным оптимумам.

Заключение. Предложен новый алгоритм инициализации популяции для эвристических алгоритмов на основе метода Метрополиса–Гастингса. Стратегия инициализации позволяет выбирать перспективные решения, близкие к локальным или глобальным оптимумам. Это ускоряет сходимость популяционного алгоритма и повышает точность решений.

Для демонстрации возможностей предлагаемой стратегии инициализации она была встроена в популярный алгоритм дифференциальной эволюции. Эффективность подхода сравнивалась с другими известными методами инициализации, встроенными в алгоритм дифференциальной эволюции, такими как случайная инициализация, обучение на основе оппозиции, метод, основанный на хаосе и метод диагонального равномерного распределения. В качестве тестовых функций оптимизации были выбраны известные мультимодальные и унимодальные функции.

Экспериментальные результаты показывают, что предлагаемая стратегия обеспечивает лучшую скорость сходимости и качество решений в сравнении с конкурирующими методами инициализации. Статистическая проверка результатов работы алгоритмов по критерию Фридмана подтвердила, что предлагаемый подход к инициализации популяции решений обеспечивает лучший баланс скорость сходимости/точность решений.

Работа выполнена при поддержке Министерства науки и ВО РФ в рамках проекта “Разработка и исследование методики интеллектуального геоинформационного моделирования транспортных процессов в условиях неполной информации”.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Родзин С.И. Современное состояние биоэвристик: классификация, бенчмаркинг, области применения // Известия ЮФУ. Технические науки. – 2023. – № 2. – С. 280-298.
2. Wolpert D.H., Macready W.G. No free lunch theorems for optimization // IEEE Trans. Evol. Comput. – 1997. – No. 1. – P. 67-82.
3. Dragoi E.N., Dafinescu V. Review of Metaheuristics Inspired from the Animal Kingdom // Mathematics. – 2021. – No. 9. – P. 2335.
4. Родзин С.И., Родзина О.Н. Сравнение программных реализаций эволюционных вычислений для задач многомерной оптимизации // Программная инженерия. – 2019. – Т. 10, № 11-12. – С. 451-456.
5. Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution // IEEE Cong. on Evol. Comp. – 2013. – P. 71-78.
6. Wen J., Ma H., Zhang X. Optimization of the occlusion strategy in visual tracking // Tsinghua Sci. Technol. – 2016. – Vol. 21 (2). – P. 221-230.
7. Rahnamayan S., et.al. A novel population initialization method for accelerating evolutionary algorithms // Comput. Math. Appl. – 2007. – Vol. 53 (10). – P. 1605-1614.
8. Pan W., et. al. Adaptive randomness: a new population initialization method // Math. problems in engineering. – 2014. – Vol. 2014. – 14 p.
9. Ahmad M., Isa N., Limb W., Ang K. Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy // Jour. Al. Engin. – 2022. – Vol. 61 (12). – P. 11835-11858.
10. Chib S, Greenberg E. Understanding the Metropolis-Hastings Algorithm // Am Stat. – 1995. – Vol. 49 (4). – P. 327-335.
11. Chauveau D., Vandekerckhove P. Improving convergence of the Hastings-Metropolis algorithm with an adaptive proposal // Scand. Jour. Stat. – 2002. – Vol. 29 (1). – P. 13-29.
12. Rodzin S., Rodzina O. New computational models for big data and optimization // Proc. 9th Int. Conf. on Application of Information and Communication Technologies (AICT). – 2015. – P. 3-7.
13. Rahnamayan S., Tizhoosh H., Salama M. A novel population initialization method for accelerating evolutionary algorithms // Comput. Math. Appl. – 2007. – Vol. 53 (10). – P. 1605-1614.
14. Ahmad M., Isa N., Limb W., Ang K. Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy // Alexan. Engin. Jour. – 2022. – Vol. 61 (12). – P. 11835-11858.
15. Li Q., Bai Y., Gao W. Improved Initialization Method for Metaheuristic Algorithms: A Novel Search Space View // IEEE Access. – 2021. – Vol. 9. – P. 158508-158539.
16. Cuevas E., Escobar H., Sarkar R., Eid H. A new population initialization approach method // Applied Intelligence. – 2023. – Vol. 53. – P. 16575-16593.
17. Wang H., Wu Z., Rahnamayan S. Enhanced opposition based differential evolution for solving high-dimensional continuous optimization problems // Soft. Comput. – 2011. – Vol. 15 (11). – P. 2127-2140.
18. Rodzin S., Rodzina L. Theory of bionic optimization and its application to evolutionary synthesis of digital devices // Proc. of IEEE East-West Design and Test Symposium (EWDTS). – 2014. – P. 7027058.
19. Курейчик В.В., Родзин С.И. Биоэвристики, инспирированные фауной (обзор) // Информационные технологии. – 2023. – Т. 29, № 11. – С. 559-573.
20. Родзин С.И., Скобцов Ю.А., Эль-Хатиб С.А. Биоэвристики: теория алгоритмы и приложения: монография. – Чебоксары: ИД "Среда", 2019. – 224 с.

REFERENCES

1. Rodzin S.I. Sovremennoe sostoyanie bioevristik: klassifikatsiya, benchmarking, oblasti primeneniya [Current state of bioheuristics: classification, benchmarking, application areas], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2023, No. 2, pp. 280-298.
2. Wolpert D.H., Macready W.G. No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, 1997, No. 1, pp. 67-82.
3. Dragoi E.N., Dafinescu V. Review of Metaheuristics Inspired from the Animal Kingdom, *Mathematics*, 2021, No. 9, pp. 2335.
4. Rodzin S.I., Rodzina O.N. Sravnenie programmnykh realizatsiy evolyutsionnykh vychisleniy dlya zadach mnogomernoy optimizatsii [Comparison of software implementations of evolutionary computing for multidimensional optimization problems], *Programmnaya inzheneriya* [Software engineering], 2019, Vol. 10, No. 11-12, pp. 451-456.
5. Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution, *IEEE Cong. on Evol. Comp.*, 2013, pp. 71-78.
6. Wen J., Ma H., Zhang X. Optimization of the occlusion strategy in visual tracking, *Tsinghua Sci. Technol.*, 2016, Vol. 21 (2), pp. 221-230.
7. Rahnamayan S., et.al. A novel population initialization method for accelerating evolutionary algorithms, *Comput. Math. Appl.*, 2007, Vol. 53 (10), pp. 1605-1614.

8. Pan W., et. al. Adaptive randomness: a new population initialization method, *Math. problems in engineering*, 2014, Vol. 2014, 14 p.
9. Ahmad M., Isa N., Limb W., Ang K. Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy, *Jour. Al. Engin.*, 2022, Vol. 61 (12), pp. 11835-11858.
10. Chib S., Greenberg E. Understanding the Metropolis–Hastings Algorithm, *Am Stat.*, 1995, Vol. 49 (4), pp. 327-335.
11. Chauveau D., Vandekerckhove P. Improving convergence of the Hastings-Metropolis algorithm with an adaptive proposal, *Scand. Jour. Stat.*, 2002, Vol. 29 (1), pp. 13-29.
12. Rodzin S., Rodzina O. New computational models for big data and optimization, *Proc. 9th Int. Conf. on Application of Information and Communication Technologies (AICT)*, 2015, pp. 3-7.
13. Rahnamayan S., Tizhoosh H., Salama M. A novel population initialization method for accelerating evolutionary algorithms, *Comput. Math. Appl.*, 2007, Vol. 53 (10), pp. 1605-1614.
14. Ahmad M., Isa N., Limb W., Ang K. Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy, *Alexan. Engin. Jour.*, 2022, Vol. 61 (12), pp. 11835-11858.
15. Li Q., Bai Y., Gao W. Improved Initialization Method for Metaheuristic Algorithms: A Novel Search Space View, *IEEE Access.*, 2021, Vol. 9, pp. 158508-158539.
16. Cuevas E., Escobar H., Sarkar R., Eid H. A new population initialization approach method, *Applied Intelligence*, 2023, Vol. 53, pp. 16575-16593.
17. Wang H., Wu Z., Rahnamayan S. Enhanced opposition based differential evolution for solving high-dimensional continuous optimization problems, *Soft. Comput.*, 2011, Vol. 15 (11), pp. 2127-2140.
18. Rodzin S., Rodzina L. Theory of bionic optimization and its application to evolutionary synthesis of digital devices, *Proc. of IEEE East-West Design and Test Symposium (EWDTS)*, 2014, pp. 7027058.
19. Kureychik V.V., Rodzin S.I. Bioevristiki, inspirirovannye faunoy (obzor) [Bioheuristics inspired by fauna (review)], *Informatsionnye tekhnologii* [Information technologies], 2023, Vol. 29, No. 11, pp. 559-573.
20. Rodzin S.I., Skobtsov Yu.A., El'-Khatib S.A. Bioevristiki: teoriya algoritmy i prilozheniya: monografiya [Bioheuristics: theory, algorithms and applications: monograph]. Cheboksary: ID "Sreda", 2019, 224 p.

Родзин Сергей Иванович – Южный федеральный университет; e-mail: srodzin@sfedu.ru; г. Таганрог, Россия; кафедра математического обеспечения и применения ЭВМ; профессор.

Дерменжи Анастасия Игоревна – ООО «Видучи»; e-mail: dermenzhi-00@mail.ru; г. Москва, Россия; генеральный директор.

Rodzin Sergey Ivanovich – Southern Federal University; e-mail: srodzin@sfedu.ru; Taganrog, Russia; phone: +78634371673; the Department of Software Engineering; professor.

Dermenzhi Anastasia Igorevna – Viduchi LLC, e-mail: dermenzhi-00@mail.ru; Moscow, Russia; general manager.

УДК 004.932

DOI 10.18522/2311-3103-2025-1-56-65

А.В. Полтавский, Д.С. Поляниченко, Е.Р. Коломенская, М.А. Бугакова

АЛГОРИТМЫ ГЕНЕРАЦИИ И ОБРАБОТКИ SEM-ИЗОБРАЖЕНИЙ ДЛЯ ИДЕНТИФИКАЦИИ СВОЙСТВ БИОНЕОРГАНИЧЕСКИХ МАТРИЦ И МЕТОДЫ ИХ ВЕРИФИКАЦИИ

Одним из распространенных методов анализа характеристик материалов, получаемых в результате химического синтеза, является сканирующая электронная микроскопия (SEM), применение которой позволяет получить изображения с высоким разрешением и увеличением. В статье исследуются алгоритмы анализа изображений материалов, обладающих определенными свойствами (например, пористостью) – бионеорганических матриц. Пористые структуры представляют собой обширный класс материалов с широкой областью применения, включая сельское хозяйство, медицину, катализ и многое другое. Одна из важных сфер использования подобных структур – тканевая инженерия, в которой подобные каркасы необходимы для обеспечения восстановительных процессов тканей организма. И для каждого организма матрицы должны быть персонализированными, что требует выполнения трудоемкого процесса по подбору характеристик каркаса, применимого в конкретном случае. Эта задача в настоящее время частично решается применением технологий искусственного интеллекта для повышения точности или поддержки при-