

УДК 519.854.2

DOI: [10.26102/2310-6018/2025.50.3.040](https://doi.org/10.26102/2310-6018/2025.50.3.040)

Процедура динамической модификации схемы бинарного кодирования индивидов в генетическом алгоритме

И.П. Малашин✉

Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация

Резюме. В статье представлена процедура динамической модификации схемы бинарного кодирования в генетическом алгоритме (ГА), обеспечивающая адаптивную коррекцию области поиска в процессе работы алгоритма. В предложенной процедуре шаг дискретизации каждой координаты изменяется от поколения к поколению в зависимости от текущих границ областей с качественными решениями и плотности распределения индивидов в них. Для каждого такого диапазона рассчитывается число бит бинарной строки, представляющей решения в алгоритме, на основе количества кодируемых точек, после чего пересчитывается шаг дискретизации области поиска. При этом схема кодирования перестраивается таким образом, чтобы обеспечить корректность выполнения генетических операторов при наличии разрывов в области поиска, сохранить постоянную мощность множества перебираемых решений на каждом поколении, а также повысить точность найденных решений за счет динамически изменяющегося шага дискретизации. Экспериментальные результаты на многомодальных тестовых функциях, таких как Растригина и Стыблински-Танга, показали, что предложенная модификация ГА последовательно корректирует область поиска в ходе эволюции, концентрируя найденные решения вокруг глобальных экстремумов. В случае функции Растригина изначально разрозненные диапазоны постепенно фокусируются на области с максимальными значениями. Для функции Стыблински-Танга из заведомо неверной начальной области алгоритм смещает поиск к одному из глобальных оптимумов.

Ключевые слова: адаптивное кодирование, генетический алгоритм, дискретизация, многомодальная оптимизация, область поиска.

Для цитирования: Малашин И.П. Процедура динамической модификации схемы бинарного кодирования индивидов в генетическом алгоритме. *Моделирование, оптимизация и информационные технологии.* 2025;13(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=2000> DOI: 10.26102/2310-6018/2025.50.3.040

A procedure for dynamic modification of binary encoding scheme in genetic algorithms

I.P. Malashin✉

Bauman Moscow State Technical University, Moscow, the Russian Federation

Abstract. This paper presents a procedure for dynamically modifying the binary encoding scheme in a genetic algorithm (GA), enabling adaptive adjustment of the search space during the algorithm's execution. In the proposed approach, the discretization step for each coordinate is updated from generation to generation based on the current boundaries of regions containing high-quality solutions and the density of individuals within them. For each such region, the number of bits in the binary string representing solutions is determined according to the number of encoded points, after which the discretization step is recalculated. The encoding scheme is restructured in a way that ensures the correctness of genetic operators in the presence of discontinuities in the search space, preserves the fixed cardinality of the solution set at each generation, and increases the precision of the solutions due to the dynamic adjustment of the discretization step. Experimental results on multimodal test functions such

as Rastrigin and Styblinski–Tang demonstrate that the proposed GA modification progressively refines the search area during evolution, concentrating solutions around the global extrema. For the Rastrigin function, initially fragmented regions gradually focus around the global maximum. In the Styblinski–Tang case, the algorithm shifts the search from an intentionally incorrect initial area toward one of the global optima.

Keywords: adaptive encoding, genetic algorithm, discretization, multimodal optimization, search space.

For citation: Malashin I.P. A Procedure for Dynamic Modification of Binary Encoding Scheme in Genetic Algorithms. *Modeling, Optimization and Information Technology*. 2025;13(3). (In Russ.). URL: <https://moitvvt.ru/ru/journal/pdf?id=2000> DOI: 10.26102/2310-6018/2025.50.3.040

Введение

Генетический алгоритм (ГА) – это стохастический метод оптимизации, основанный на принципах естественного отбора и эволюции [1, 2]. Он работает с популяцией потенциальных решений, представленных в виде хромосом (обычно бинарных строк), которые эволюционируют от поколения к поколению с помощью операторов отбора, скрещивания и мутации [3]. За счёт механизма «выживания сильнейших» ГА способен эффективно искать глобальные экстремумы сложных, многомерных и многомодальных функций, не требуя знания градиентов или других аналитических свойств.

В стандартных реализациях ГА для работы с вещественными переменными применяется фиксированная схема дискретизации [4], когда шаг кодирования каждой координаты и границы области поиска остаются постоянными для всех поколений. При этом сама область поиска предполагается непрерывной по всем координатам пространства решений, каждый индивид представляется фиксированной битовой строкой определённой длины, а границы области поиска для вещественных переменных задаются заранее и остаются неизменными на всём протяжении работы алгоритма. Для каждой координаты отводится фиксированное число бит, и при декодировании битовой подстроки в вещественное значение используется линейное отображение из диапазона $[0, 2^{L_i} - 1]$ в заранее заданный интервал $[a_i, b_i]$. Благодаря этому шаг дискретизации жёстко фиксирован и определяется лишь длиной битовой подстроки и границами переменных. Инициализация популяции выполняется случайным образом по всем возможным битовым строкам [5], что обеспечивает равномерное покрытие исходного пространства решений, но не предполагает адаптации области поиска в процессе эволюции. На каждой итерации полагаться можно на стандартные операторы: после декодирования индивидов в наборы вещественных координат вычисляется значение функции пригодности, затем с помощью турнирной [6] или ранговой [7] селекции выбираются родители, к их битовым строкам применяются операции скрещивания (одноточечное, двухточечное, равномерное и т. д.) и побитовая мутация с низкой вероятностью флипа, после чего формируется новое поколение с сохранением длины всех строк [8]. Поскольку длина строк и границы остаются теми же, структура кодирования не меняется, и все операторы продолжают работать без дополнительных преобразований. На ранних поколениях такой подход обеспечивает широкое исследование пространства за счёт разнообразной инициализации; по мере появления более приспособленных индивидов оператор выбора концентрирует ресурсы на областях с более высокими значениями функции пригодности. Тем не менее фиксированная дискретизация не позволяет уточнять поиск в перспективных областях и одновременно избегать избыточного рассмотрения бесперспективных зон, особенно если первоначально заданные границы слишком широки либо слишком узки в отношении оптимума. Фиксированный шаг кодирования может оказаться слишком

грубым для высокой точности или избыточно мелким, расходуя лишние вычисления без заметного выигрыша, и не учитывает накопленную информацию о поведении популяции.

Однако при использовании модифицированных версий ГА, в которых область поиска динамически корректируется – например, сдвигаются её границы или формируются интервалы разрывов – возникает необходимость в гибкой, адаптивной схеме кодирования. Такая схема должна обеспечивать корректное выполнение всех генетических операторов, включая скрещивание и мутацию, без введения дополнительных ограничений в формулировку задачи. При этом важно сохранить фиксированную длину битовой строки, что необходимо для унификации бинарного представления и совместимости с классическими механизмами генетической обработки.

В настоящей работе предложена процедура динамической модификации схемы бинарного кодирования индивидов, в которой шаг дискретизации для каждой переменной x_i на поколении g становится функцией состояния популяции и текущих границ области поиска. В отличие от классического ГА, где оператор бинарного кодирования остаётся неизменным на всём протяжении эволюции, в предложенном подходе схема Bin^g пересоздаётся при каждом расширении или сужении диапазона, что позволяет:

- локально увеличивать точность представления в тех областях, где сосредоточены перспективные решения;
- сохранять компактность генотипа и управлять размером популяции при перераспределении вычислительных ресурсов;
- автоматически реагировать на изменения структуры поискового пространства без вмешательства пользователя.

Материалы и методы

Предлагаемая в настоящем исследовании процедура динамической модификации схемы бинарного кодирования индивидов строится на следующих принципах:

- вне зависимости от наличия разрывов в области поиска ГА бинарное представление переменных не должно кодировать решения, попадающие в разрывы, для обеспечения корректности выполнения генетических операторов;
- размер бинарной строки должен оставаться постоянным на протяжении всех поколений генетического алгоритма для сохранения мощности множества перебираемых решений на каждом поколении;
- шаг дискретизации по каждой переменной может динамически изменяться на различных поколениях ГА для повышения точности найденных решений.

В случае применения методов кластеризации [9, 10] для поиска областей поискового пространства, представляющих повышенный интерес для алгоритма оптимизации, и исключения неперспективных областей, формируются разнесенные по координатам пространства области поиска [11], схема кодирования индивидов при работе с которыми представлена на Рисунке 1.

Если в стандартном ГА шаг дискретизации по переменной x_i является постоянным ($\text{step}_{x_i} = \text{const}$), то в предложенной модификации ГА шаг на каждом поколении g может динамически меняться – $\text{step}_{x_i}^g$. Начальный шаг $\text{step}_{x_i}^1$ и область поиска $D_{x_i}^1 = \{x_i^l, x_i^h\}$ (где x_i^l и x_i^h – левые и правые границы области поиска соответственно) в обоих случаях задаются пользователем.



Рисунок 1 – Блок-схема процедуры динамической модификации схемы бинарного кодирования индивидов

Figure 1 – Block diagram of the procedure for dynamic modification of the binary coding scheme for individuals

Введем следующие обозначения:

– $\text{Bin}^1(\cdot) = (\text{Bin}_{x_1}^1(\cdot), \dots, \text{Bin}_{x_n}^1(\cdot))$ – классическая схема бинарного кодирования, которая постоянна в течение всех поколений работы стандартного ГА, представляющая собой оператор над множеством фенотипов решений. В предлагаемой процедуре динамической модификации схемы бинарного кодирования индивидов Bin^1 будет использоваться на первом поколении;

– $N\text{Bit}_{x_i}$ – количество бит, необходимое для кодирования множества решений по каждой переменной x_i с заданным шагом $\text{step}_{x_i}^1$:

$$N\text{Bit}_{x_i} = \left\lceil \log_2 \left(\frac{x_i^h - x_i^l}{\text{step}_{x_i}^1} \right) \right\rceil; \quad (1)$$

– $\text{step}_{x_i}^1$ – коррекция шага дискретизации значений по переменной x_i с учетом изменяющихся границ области поиска и фиксированного количества кодируемых точек:

$$\text{step}C_{x_i}^1 = \frac{\text{step}x_i^1}{2^{N\text{Bit}x_i}} = \frac{x_i^h - x_i^l}{2^{N\text{Bit}x_i} \cdot \frac{x_i^h - x_i^l}{\text{step}x_i^1}} \quad (2)$$

- Ph_{m,x_i}^g – фенотип индивида (значение решения) по координате x_i ;
- Ind_{m,x_i}^g – компонент индивида по координате x_i :

$$\text{Ind}_{m,x_i}^g = \left(\frac{\text{Ph}_{m,x_i}^g - x_i^l}{\text{step}C_{x_i}^1} \right)_{10 \rightarrow 2}, \quad (3)$$

где $(\cdot)_{10 \rightarrow 2}$ – операция перевода десятичного числа в двоичное представление с $N\text{Bit}x_i$ разрядами, причем первые незначимые разряды обязательно указываем со значением 0.

Для поколений, в которых не происходит коррекции границ области поиска, схема бинарного кодирования не претерпевает изменений, т. е. $\text{Bin}^g(\cdot) = \text{Bin}^{g-1}(\cdot)$, в противном случае $\text{Bin}^g(\cdot)$ определяется как процедура, состоящая из 4 основных шагов:

Шаг 1. Упорядочивание кластеров по возрастанию левых границ $x_i^{g,k,l} + \Delta_{x_i^{g,k,l}}$.

Шаг 2. Для каждого кластера k найти показатель удаленности границы минимального значения по x_i относительно начала $D_{x_i}^{g,l}$ пространства поиска $D_{x_i}^g$. Определим его как сдвигу $S_{x_i}^{g,k,l}$ (4):

$$S_{x_i}^{g,k,l} = \left(x_i^{g,1,h} + \Delta_{x_i^{g,1,h}} \right) - \left(x_i^{g,1,l} + \Delta_{x_i^{g,1,l}} \right) +$$

$$+ \sum_{\substack{e=\overline{2,k-1} \\ x_i^{g,e,h} > x_i^{g,d,h}, \forall d=\overline{1,e-1} \\ x_i^{g,e,l} \leq x_i^{g,d,h}, \forall d=\overline{1,e-1}}} \left(\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \underbrace{\max_{d=\overline{1,e-1}} \left(x_i^{g,d,h} + \Delta_{x_i^{g,d,i}} \right)}_{\text{отрезки с частичными перекрытиями}} \right) + \quad (4)$$

$$+ \sum_{\substack{e=\overline{2,k-1} \\ x_i^{g,e,h} > x_i^{g,d,h}, \forall d=\overline{1,e-1} \\ x_i^{g,e,l} > x_i^{g,d,h}, \forall d=\overline{1,e-1}}} \left(\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \underbrace{\left(x_i^{g,e,l} + \Delta_{x_i^{g,e,i}} \right)}_{\text{неперекрывающиеся отрезки}} \right).$$

Если $x_i^{g,e,l} > x_i^{g,e-1,h}$ – то в расчет берется вся длина отрезка.

Если $x_i^{g,e,h} > x_i^{g,d,h}, \forall d = \overline{2..e-1}$ – тогда 0, т. к. вложенные отрезки не суммируются.

Если $x_i^{g,e,l} \leq x_i^{g,e-1,h}, \forall d = \overline{2..e-1}$ – тогда берем часть длины $\left(x_i^{g,e,h} + \Delta_{x_i^{g,e,h}} \right) - \max_{d=\overline{1,e-1}} \left(x_i^{g,d,h} + \Delta_{x_i^{g,d,i}} \right)$.

Шаг 3. Коррекция шага по каждой переменной x_i :

$$\text{step}C_{x_i}^1 = \frac{\left(x_i^{g,k,h} + \Delta_{x_i^{g,k,h}} \right) - \left(x_i^{g,k,l} + \Delta_{x_i^{g,k,l}} \right)}{2^{N\text{Bit}x_i}}. \quad (5)$$

Шаг 4. Для всех решений, оставшихся после удаления индивидов, и не попадающих в новое пространство поиска, выполняем следующую процедуру:

$$\text{Ind}_{m,x_i}^{g+1,k} = \begin{cases} \frac{\text{Ph}_{m,x_i}^{g,k} - \left(x_i^{g,1,l} + \Delta_{x_i^{g,1,l}}\right)}{\text{step}C_{x_i}^1}, & \text{если } k = 1, \\ S_{x_i}^{g,k,l} + \frac{\text{Ph}_{m,x_i}^{g,k} - \left(x_i^{g,k,l} + \Delta_{x_i^{g,1,l}}\right)}{\text{step}C_{x_i}^1}, & \text{если } k = \overline{2..K}. \end{cases} \quad (6)$$

Вышеописанная динамическая схема кодирования позволяет адаптировать разрешение области поиска к локальным особенностям распределения решений. На Рисунке 2 схематично представлена процедура упорядочивания отрезков, где 1 – полностью неперекрывающийся отрезок, 3 и 4 – полностью перекрывающиеся отрезки, 5 – частично перекрывающийся отрезок.

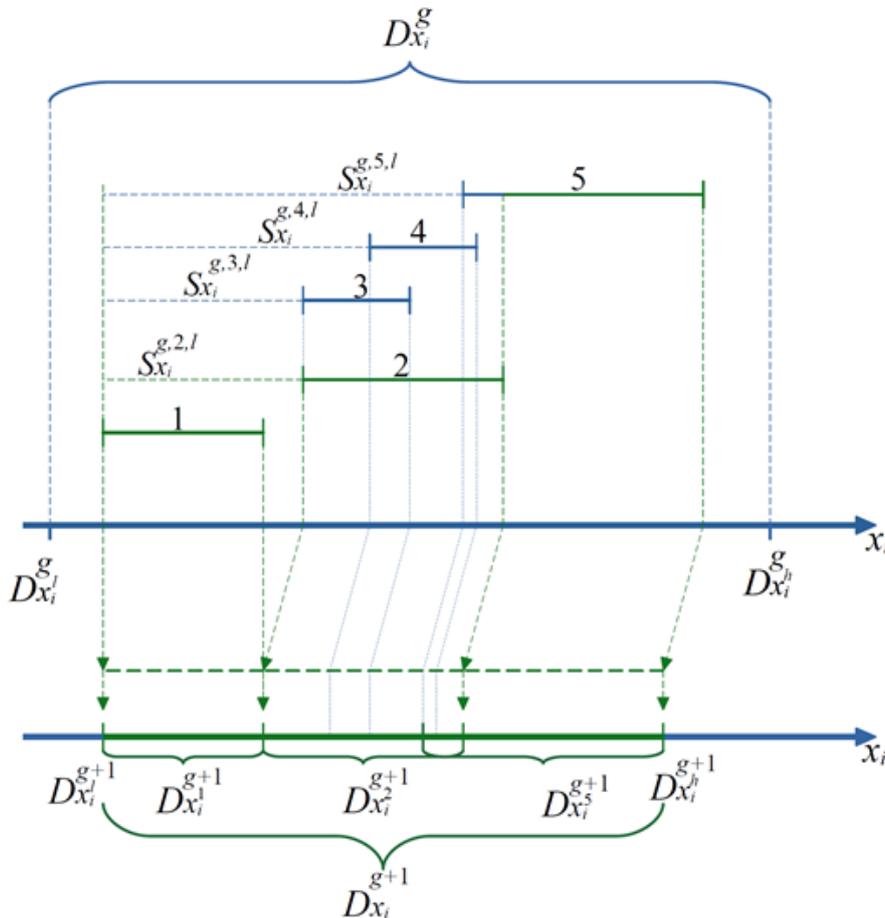


Рисунок 2 – Иллюстрация к упорядочиванию отрезков
 Figure 2 – Illustration for sorting segments

Результаты

На Рисунке 3 представлена последовательность визуализаций процесса поиска максимума функции Растргина [12] в течение 50 поколений. Оптимизация проводилась в течение 50 поколений, изображения соответствуют 0, 6, 12, 18, 24, 30, 36, 42 и 49 поколениям. Исходное поисковое пространство охватывало диапазоны $x \in [-12,5] \cup [8,10]$ и $u \in [12,10] \cup [10,12]$ с начальным шагом 0,1. В ходе оптимизации границы области поиска постепенно сужались, в итоге сосредоточившись вблизи глобального максимума функции. Начальное пространство поиска включало разрозненные области, но по мере работы алгоритма оно постепенно сужалось, концентрируясь вокруг глобального

максимума. Это подтверждает, что адаптивное обновление границ эффективно выделяет наиболее значимые регионы, ускоряя сходимость ГА.

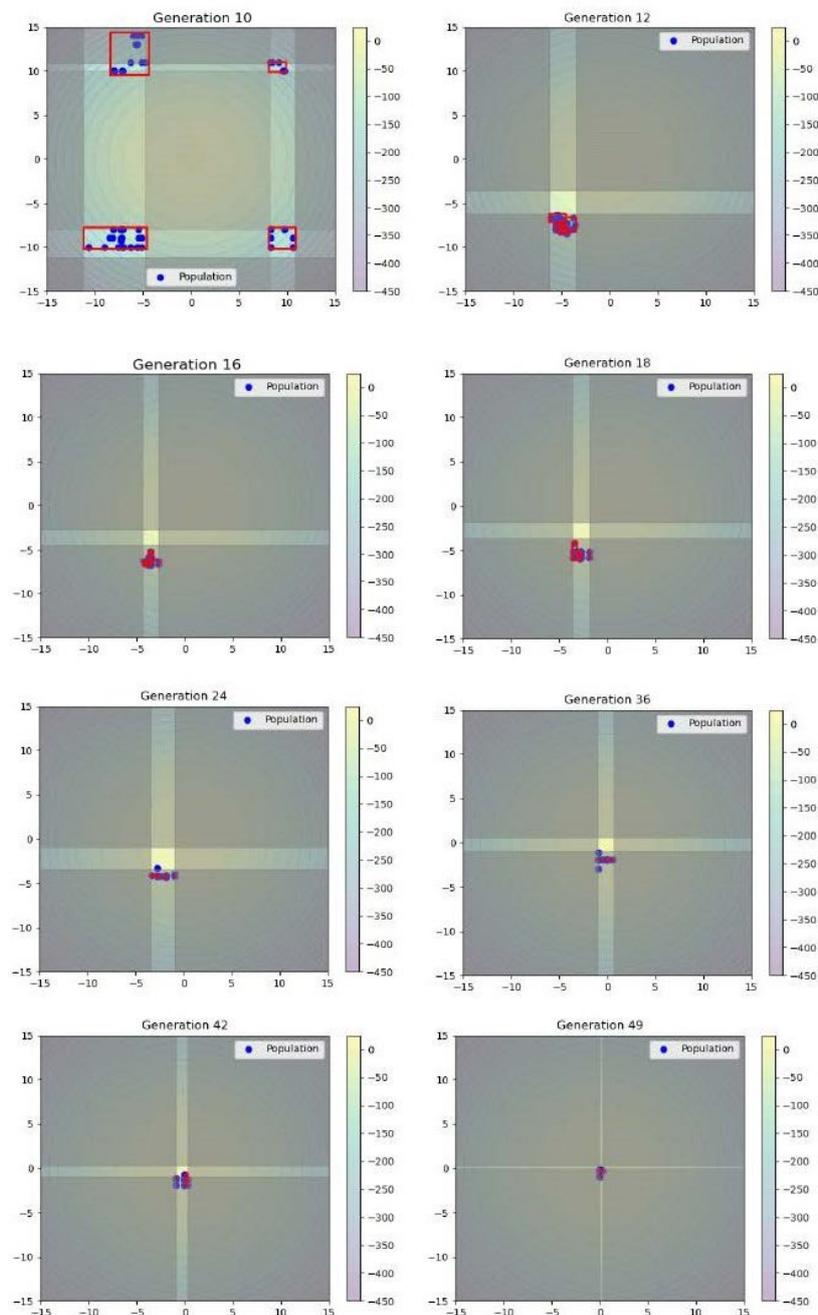


Рисунок 3 – Последовательность визуализаций работы предложенного ГА при поиске перспективной поисковой области для функции Растригина
 Figure 3 – Sequence of visualizations of the proposed GA's work when searching for a promising search area for the Rastrigin function

На Рисунке 4 представлена последовательность визуализаций работы адаптивного ГА для поиска перспективного поискового пространства функции Стыблински-Танга [13]. Оптимизация проводилась в течение 50 поколений, и изображения показывают состояния на 0, 7, 14, 21, 28 и 43 поколениях. Исходно границы поискового пространства были заданы за пределами оптимальных значений, в квадрате, ограниченном прямыми от -1 до 1 по обеим переменным, с начальным шагом

0,1. В процессе работы алгоритм постепенно перемещал область поиска, в итоге сосредоточив внимание на одном из глобальных оптимумов функции.

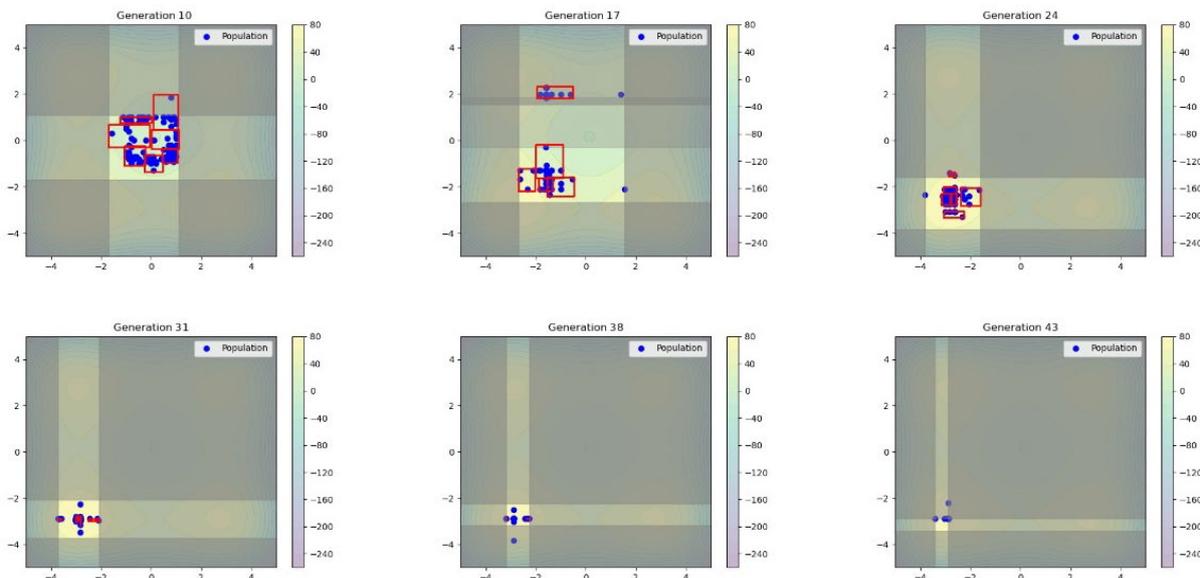


Рисунок 4 – Последовательность визуализаций работы предложенного ГА при поиске перспективной поисковой области для функции Стыблински-Танга
 Figure 4 – Sequence of visualizations of the proposed GA's work when searching for a promising search area for the Stybinski-Tanga function

На основании полученных визуализаций видно, что предложенная модификация ГА последовательно перемещает область поиска в ходе эволюции, концентрируясь вокруг глобальных экстремумов. В случае функции Растригина (Рисунок 3) изначально разрозненные диапазоны постепенно фокусируются на области с максимальными значениями. Для функции Стыблински–Танга (Рисунок 4) из заведомо неверной начальной области алгоритм смещает поиск к одному из глобальных оптимумов.

Заключение

В данной работе представлена методика динамической смены схемы бинарного кодирования в ГА, позволяющая адаптивно изменять шаг дискретизации каждой координаты в зависимости от текущей структуры популяции и границ области поиска. В отличие от классического подхода с фиксированным шагом кодирования, предложенный метод пересчитывает необходимое число бит для представления диапазона $D_{x_i}^g$ и обновляет схему Bin^g при каждом изменении поискового пространства. Это обеспечивает повышение разрешающей способности в перспективных областях за счёт локального увеличения детализации представления, сохранение компактности генотипа при изменении размеров областей поиска и автоматическую адаптацию к динамически меняющемуся ландшафту задачи без ручной перенастройки параметров. Результаты экспериментального сравнения на стандартных тестовых функциях демонстрируют, что описанная методика повышает точность найденных решений и ускоряет сходимость алгоритма по сравнению с традиционной схемой фиксированного шага. Перспективными направлениями дальнейших исследований являются интеграция данной схемы с адаптивными операторами мутации и кроссовера, а также оценка её эффективности в условиях многокритериальных и стохастических задач оптимизации.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Mirjalili S. Genetic Algorithm. In: *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham: Springer; 2018. P. 43–55. https://doi.org/10.1007/978-3-319-93025-1_4
2. Katoch S., Chauhan S.S., Kumar V. A Review on Genetic Algorithm: Past, Present, and Future. *Multimedia Tools and Applications*. 2021;80:8091–8126. <https://doi.org/10.1007/s11042-020-10139-6>
3. Мясников А.С. Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов. *Наука и образование: научное издание МГТУ им. Н.Э. Баумана*. 2010;(1). URL: https://www.elibrary.ru/download/elibrary_13062781_75581185.pdf
4. Звонков В.Б., Попов А.М. Сравнительное исследование классических методов оптимизации и генетических алгоритмов. *Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева*. 2013;(4):23–27. Zvonkov V.B., Popov A.M. Comparative Investigation of Classical Optimization Methods and Genetic Algorithms. *Vestnik of SibGAU*. 2013;(4):23–27. (In Russ.).
5. Maaranen H., Miettinen K., Penttinen A. On Initial Populations of a Genetic Algorithm for Continuous Optimization Problems. *Journal of Global Optimization*. 2007;37(3):405–436. <https://doi.org/10.1007/s10898-006-9056-6>
6. Yang J., Soh Ch.K. Structural Optimization by Genetic Algorithms with Tournament Selection. *Journal of Computing in Civil Engineering*. 1997;11(3):195–200. [https://doi.org/10.1061/\(ASCE\)0887-3801\(1997\)11:3\(195\)](https://doi.org/10.1061/(ASCE)0887-3801(1997)11:3(195))
7. Shukla A., Pandey H.M., Mehrotra D. Comparative Review of Selection Techniques in Genetic Algorithm. In: *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE 2015), 25–27 February 2015, Greater Noida, India*. IEEE; 2015. P. 515–519. <https://doi.org/10.1109/ABLAZE.2015.7154916>
8. Kumar A. Encoding Schemes in Genetic Algorithm. *International Journal of Advanced Research in IT and Engineering*. 2013;2(3):1–7.
9. Нейский И.М. Классификация и сравнение методов кластеризации. В сборнике: *Интеллектуальные технологии и системы: сборник учебно-методических работ и статей аспирантов и студентов: Выпуск 8*. Москва: НОК «CLAIM»; 2006. С. 130–142.
10. Дугушкина Н.В. Обзор популярных методов кластеризации в машинном обучении. *Наукосфера*. 2020;(7):112–118. Dugushkina N.V. Overview of Popular Clustering Methods in Machine Learning. *Naukosfera*. 2020;(7):112–118. (In Russ.).
11. Bacha S.Z.A., Benatchba K., Tayeb F.B.-S. Adaptive Search Space to Generate a Per-Instance Genetic Algorithm for the Permutation Flow Shop Problem. *Applied Soft Computing*. 2022;124. <https://doi.org/10.1016/j.asoc.2022.109079>
12. Omeradzic A., Beyer H.-G. Self-Adaptation of Multi-Recombinant Evolution Strategies on the Highly Multimodal Rastrigin Function. *IEEE Transactions on Evolutionary Computation*. 2024. <https://doi.org/10.1109/TEVC.2024.3400857>
13. Ustun D., Erkan U., Toktas A., Lai Q., Yang L. 2D Hyperchaotic Styblinski-Tang Map for Image Encryption and Its Hardware Implementation. *Multimedia Tools and Applications*. 2024;83:34759–34772. <https://doi.org/10.1007/s11042-023-17054-6>

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Малашин Иван Павлович, инженер-программист, НОЦ ФНС России и МГТУ им. Баумана, Москва, Российская Федерация.
e-mail: ivan.p.malashin@gmail.com
ORCID: [0009-0008-8986-402X](https://orcid.org/0009-0008-8986-402X)

Ivan P. Malashin, software engineer, Scientific and Educational Center of the Federal Tax Service of Russia and Bauman Moscow State Technical University, Moscow, the Russian Federation.

Статья поступила в редакцию 18.06.2025; одобрена после рецензирования 16.07.2025; принята к публикации 22.08.2025.

The article was submitted 18.06.2025; approved after reviewing 16.07.2025; accepted for publication 22.08.2025.