

УДК 519.854.2

DOI: [10.26102/2310-6018/2026.53.2.001](https://doi.org/10.26102/2310-6018/2026.53.2.001)

## Анализ эффективности самоконфигурируемого бинарного генетического алгоритма с модифицированным методом динамической коррекции области поиска

И.П. Малашин<sup>1✉</sup>, Е.А. Сопов<sup>2</sup>

<sup>1</sup>Московский государственный технический университет имени Н.Э. Баумана, Москва, Российская Федерация

<sup>2</sup>Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнёва, Красноярск, Российская Федерация

**Резюме.** В работе представлена модификация самонастраивающегося генетического алгоритма (SelfCGA), направленная на повышение эффективности поиска в задачах глобальной оптимизации. Предложенный подход основан на сочетании динамической коррекции области поиска с кластеризацией фенотипов текущей популяции, что позволяет более точно выявлять и адаптивно исследовать перспективные области пространства решений. Использование кластеризации способствует поддержанию популяционного разнообразия, а также снижает вероятность преждевременной сходимости алгоритма к локальным экстремумам. Для оценки эффективности предложенной модификации были проведены вычислительные эксперименты на стандартном тестовом наборе CEC2017 при размерностях пространства поиска 10, 30 и 50. Каждый из сравниваемых алгоритмов запускался 50 независимых раз, что обеспечило статистическую достоверность результатов. В ходе экспериментов анализировались средние и наилучшие значения функций пригодности, а также динамика сходимости в процессе эволюционного поиска. Полученные результаты показывают, что модифицированный алгоритм SelfCGA с динамической коррекцией области поиска достигает состояния стабилизации, при котором улучшения решений практически прекращаются, за меньшее число поколений для большинства тестовых функций, причем данное преимущество сохраняется при увеличении размерности задачи. Отсутствие необходимости ручной настройки параметров и сохранение базовой структуры SelfCGA делают предложенную модификацию удобной и перспективной для практического применения.

**Ключевые слова:** глобальная оптимизация, самонастраивающиеся алгоритмы, адаптация пространства поиска, кластеризация популяции, динамическая коррекция области поиска.

**Для цитирования:** Малашин И.П., Сопов Е.А. Анализ эффективности самоконфигурируемого бинарного генетического алгоритма с модифицированным методом динамической коррекции области поиска. *Моделирование, оптимизация и информационные технологии*. 2026;14(2). URL: <https://moitvvt.ru/ru/journal/article?id=2150> DOI: 10.26102/2310-6018/2026.53.2.001

## Efficiency analysis of a self-configuring binary genetic algorithm with a modified method of dynamic correction of the search space

I.P. Malashin<sup>1✉</sup>, E.A. Sopov<sup>2</sup>

<sup>1</sup>Bauman Moscow State Technical University, Moscow, the Russian Federation

<sup>2</sup>Reshetnyov Siberian State University of Science and Technology, Krasnoyarsk, the Russian Federation

**Abstract.** This paper presents a modification of the self-configuring genetic algorithm (SelfCGA) aimed at improving search efficiency in global optimization problems. The proposed approach combines dynamic correction of the search domain with phenotype clustering of the population, which makes it

possible to identify promising regions of the solution space more effectively. The use of clustering helps maintain population diversity and reduces the risk of premature convergence to local optima. To evaluate the proposed modification, computational experiments were conducted using the CEC2017 benchmark suite with problem dimensions of 10, 30, and 50. Each algorithm was executed 50 independent times, ensuring statistical reliability of the results. The performance was assessed by comparing average and best fitness values, as well as by analyzing the convergence dynamics during the evolutionary process. The experimental results demonstrate that the modified SelfCGA with dynamic correction of the search domain reaches a stabilization state – where further improvements during the evolutionary search become negligible – in fewer generations for most benchmark functions. This advantage remains evident even as the dimensionality of the search space increases. The proposed modification does not require manual parameter tuning and does not increase the structural complexity of the base SelfCGA, which makes it well suited for practical applications.

**Keywords:** global optimization, self-configuring algorithms, search space adaptation, population clustering, dynamic correction of the search domain.

**For citation:** Malashin I.P., Sopov E.A. Efficiency analysis of a self-configuring binary genetic algorithm with a modified method of dynamic correction of the search space. *Modeling, Optimization and Information Technology*. 2026;14(2). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2150> DOI: 10.26102/2310-6018/2026.53.2.001

## Введение

Современные задачи глобальной оптимизации нередко обладают высокой размерностью, неоднородным ландшафтом целевой функции и большим количеством локальных экстремумов. В таких условиях классические методы, использующие градиентную информацию или опирающиеся на аналитическое описание функции, оказываются малоэффективными. Реальная структура многих задач неизвестна заранее, а вычисление даже одного значения функции может быть дорогостоящим. Поэтому все больше внимания уделяется эволюционным и популяционным алгоритмам.

Генетический алгоритм (ГА) относится к классу эволюционных методов оптимизации. Он оперирует множеством решений, называемых индивидами, каждый из которых содержит набор параметров задачи. На каждом шаге алгоритм выполняет оценку пригодности индивида, выбирает более успешные решения и создает новые с помощью операторов мутации и кроссинговера. Поиск осуществляется не точно, а через эволюцию популяции, что позволяет обходить локальные экстремумы и исследовать различные области пространства решений. ГА не требует знания градиентов или аналитического вида целевой функции

Одной из ключевых тенденций развития эволюционных алгоритмов стала идея самонастройки – автоматической адаптации операторов и параметров в процессе работы. Исторически большинство эволюционных методов требовали ручного подбора параметров: вероятностей мутации и кроссинговера, стратегии селекции, размера популяции и других гиперпараметров. Их неверный выбор мог резко снизить качество решения, а поиск оптимальной конфигурации занимал значительное время. На этом фоне возникла концепция self-adaptive / self-configuring эволюционных алгоритмов, где сам алгоритм постепенно подстраивает собственное поведение под структуру ландшафта целевой функции [1, 2]. Одним из представителей этого подхода является SelfCGA – самонастраивающийся ГА, в котором вероятности мутации и кроссинговера автоматически корректируются по статистике поколений. Такой подход снижает зависимость результата от пользовательских настроек и делает алгоритм более автономным [3]. Однако самонастройка операторов решает только одну сторону проблемы – управление динамикой эволюции.

Ранее в литературе был предложен метод динамической коррекции области поиска (ДКОП) [4], предполагающей целенаправленное изменение пространства поиска с течением поколений. Этот принцип реализован в алгоритме SelfCGA. Новая версия алгоритма анализирует фенотипы популяции с помощью кластеризации и выявляет области, которые не способствуют дальнейшему улучшению решения. Индивидов из таких «неперспективных» кластеров заменяют либо представителями элитного архива, либо случайно сгенерированными кандидатами [5].

Для оценки эффективности предложенный подход был протестирован на наборе CEC2017 – одном из наиболее признанных стандартов для оценки алгоритмов эволюционной оптимизации<sup>1</sup>. Рассматривались размерности 10, 30 и 50, а каждый алгоритм запускался 50 раз на каждой функции. Анализ включал сравнение средних и лучших значений, визуализацию динамики поиска по поколениям и статистическую проверку с использованием критерия Вилкоксона. Результаты показали, что интеграция модификации ДКОП повышает качество решений SelfCGA.

### Материалы и методы

Рассматриваемая задача формулируется как поиск глобального минимума функции  $f(x)$  в многомерном пространстве:

$$\min_{\{x \in \mathbb{R}^D\}} f(x), \quad (1)$$

где  $D$  – размерность пространства поиска. Структура функции неизвестна, аналитическое описание отсутствует, а вычисление одного значения может быть сопряжено с высокой стоимостью. В таких условиях требуется алгоритм, способный вести стохастический поиск без априорной информации и сохранять баланс между исследованием новых областей и уточнением текущих перспективных решений.

В работе рассматривается модификация самонастраивающегося ГА SelfCGA (Self-Configuring Genetic Algorithm). Базовый SelfCGA расширен за счет введения оператора равномерного кроссовера, который усиливает селекционное давление на этапе рекомбинации. Предложен подход к самонастройке ГА, основанный на вероятностных характеристиках операторов [6].

Предложенная модификация – ClusterSelfCGA (SelfCGA+ДКОП) – расширяет SelfCGA механизмом ДКОП, направленным на обработку фенотипов методом кластеризации. Пусть  $X \in \mathbb{R}^{n \times D}$  – матрица фенотипов популяции из  $n$  индивидов. На каждом поколении выполняется кластеризация методом  $k$ -средних [7]:

$$k = \min(n\_clusters, n), \quad (2)$$

после чего для каждого кластера оцениваются два показателя: количество индивидов  $N_k$  и средний фитнес  $F_k$ . Кластер считается «неперспективным», если одновременно выполняются условия:

$$N_k < \alpha_N \cdot \bar{N}, \quad F_k < \alpha_F \cdot \tilde{F}, \quad (3)$$

где  $\bar{N}$  – средний размер кластера, а  $\tilde{F}$  – медиана значений  $F_k$  по всем кластерам. В таких кластерах часть индивидов заменяется: в первую очередь используются особи из элитного архива, при нехватке – случайные генотипы. Замена выполняется с вероятностью *replace\_prob*, при этом размер популяции остается неизменным. Механизм

<sup>1</sup> Wu G., Mallipeddi R., Suganthan P.N. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization. Technical Report. ResearchGate. URL: <https://www.researchgate.net/publication/317228117> (дата обращения: 15.11.2025).

элитизма реализуется через выделение  $k$  лучших индивидов по функции пригодности, которые сохраняются и обновляются в каждом поколении.

Эксперименты проводились на наборе тестовых функций CEC2017 [8], включающем гибридные и композиционные задачи с множеством локальных экстремумов и переменной сложностью ландшафта. Рассматривались размерности  $D = 10, 30$  и  $50$ . Для каждой функции выполнялось по 50 независимых запусков. Параметры SelfCGA и его модификации подбирались, исходя из масштабируемости задачи: при  $D = 10$  использовалось 200 поколений и популяция из 2000 индивидов; при  $D = 30$  – 200 поколений и 4000 индивидов; при  $D = 50$  – 400 поколений и 8000 индивидов. Сравнение алгоритмов выполнялось по средним и лучшим значениям функции пригодности, по траекториям эволюции значений по поколениям, а также с использованием критерия Вилкоксона для статистической проверки различий.

Предложенный ClusterSelfCGA не изменяет структуру SelfCGA и может быть интегрирован в его цикл без модификации внутренней логики.

### Результаты

В данном разделе проводится сравнительный анализ производительности классического ГА, его модификации с ДКОП (ГА+ДКОП), а также самонастраивающегося алгоритма SelfCGA и его расширения с использованием ДКОП (SelfCGA+ДКОП). В качестве тестового набора использовался CEC-2017, включающая 30 функций различной сложности. Для каждого алгоритма выполнялось по 50 независимых запусков на каждой из функций.

Эксперименты проводились при различных размерностях пространства поиска. В Таблице 1 приведены параметры популяции и количество поколений, использованные для каждого алгоритма.

Таблица 1 – Параметры экспериментальных исследований  
Table 1 – Parameters of experimental studies

Алгоритм	Размерность $D$	Число поколений	Размер популяции
ГА / ГА+ДКОП	10	200	2000
SelfCGA / SelfCGA+ДКОП	10	200	2000
SelfCGA / SelfCGA+ДКОП	30	200	4000
SelfCGA / SelfCGA+ДКОП	50	400	8000

Результаты, представленные на Рисунке 1, демонстрируют поведение стандартного ГА (реализованного на основе библиотеки эволюционной оптимизации *baumeva*) и его модификации ГА+ДКОП при размерности  $D = 10$  на функциях набора CEC2017. В обеих реализациях использовались турнирная селекция [9] и механизм элитизма [10].



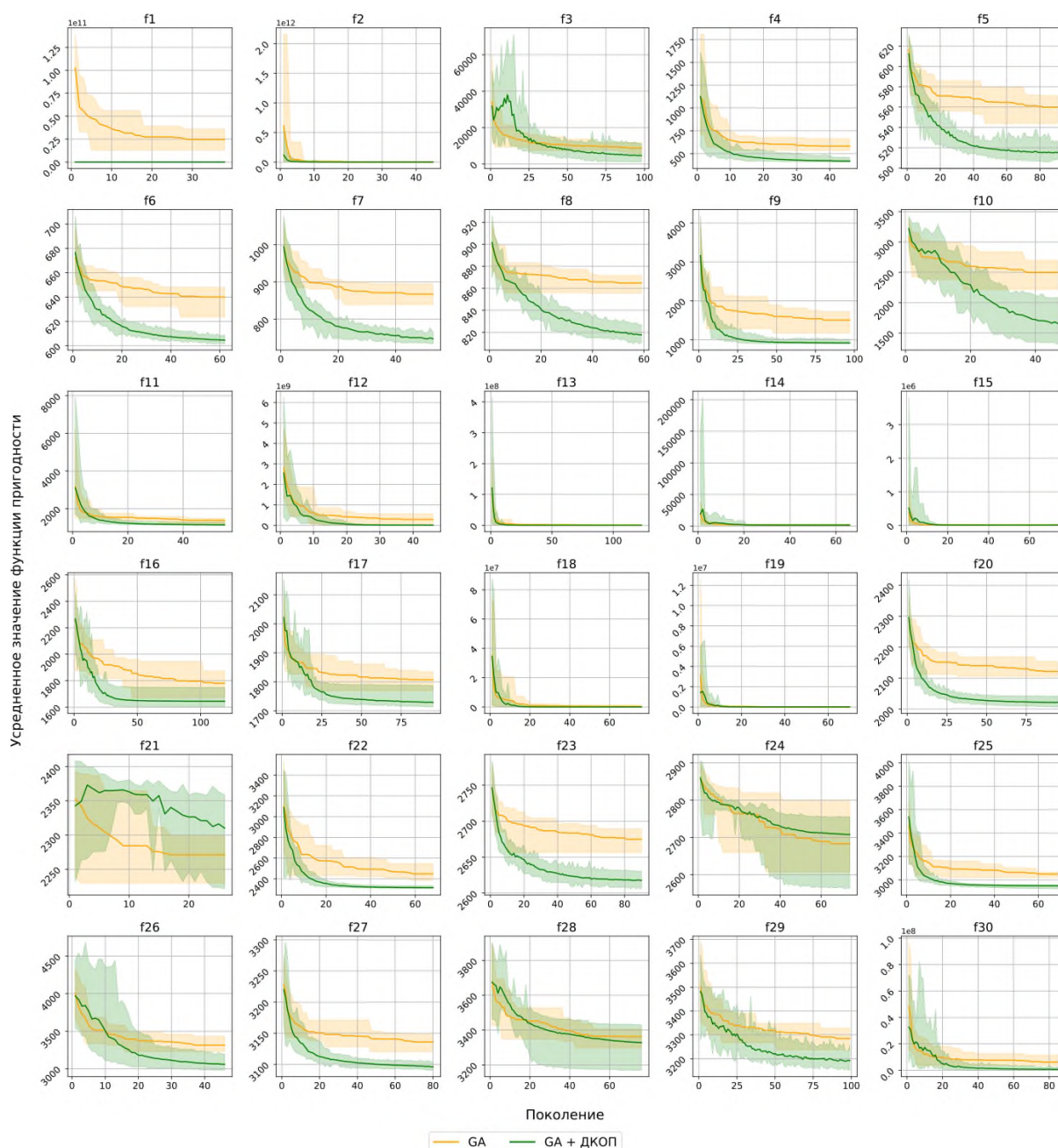


Рисунок 1 – Эволюция значения функции пригодности ГА и ГА+ДКОП при  $D = 10$   
Figure 1 – Evolution of the fitness function value of GA and GA+DCOP at  $D = 10$

ДКОП ускоряет сходимость в начальной фазе эволюции и позволяет достигать меньших значений функции пригодности. На графиках видно, что использование ДКОП в составе ГА приводит к более раннему снижению значения функции пригодности по сравнению с базовой версией алгоритма. В большинстве функций кривая ГА+ДКОП достигает области стабильности за меньшее число поколений. Также наблюдается более узкая доверительная полоса усредненного значения функции пригодности по поколениям для модификации ГА+ДКОП, что указывает на более стабильную динамику поиска по сравнению со стандартным ГА. Данный эффект можно наблюдать по характеру зеленых областей, которые в ряде случаев более узкие относительно оранжевых. Аналогичные наблюдения применимы также и к алгоритму SelfCGA при  $D = 10$ , что подтверждается Рисунок 2.

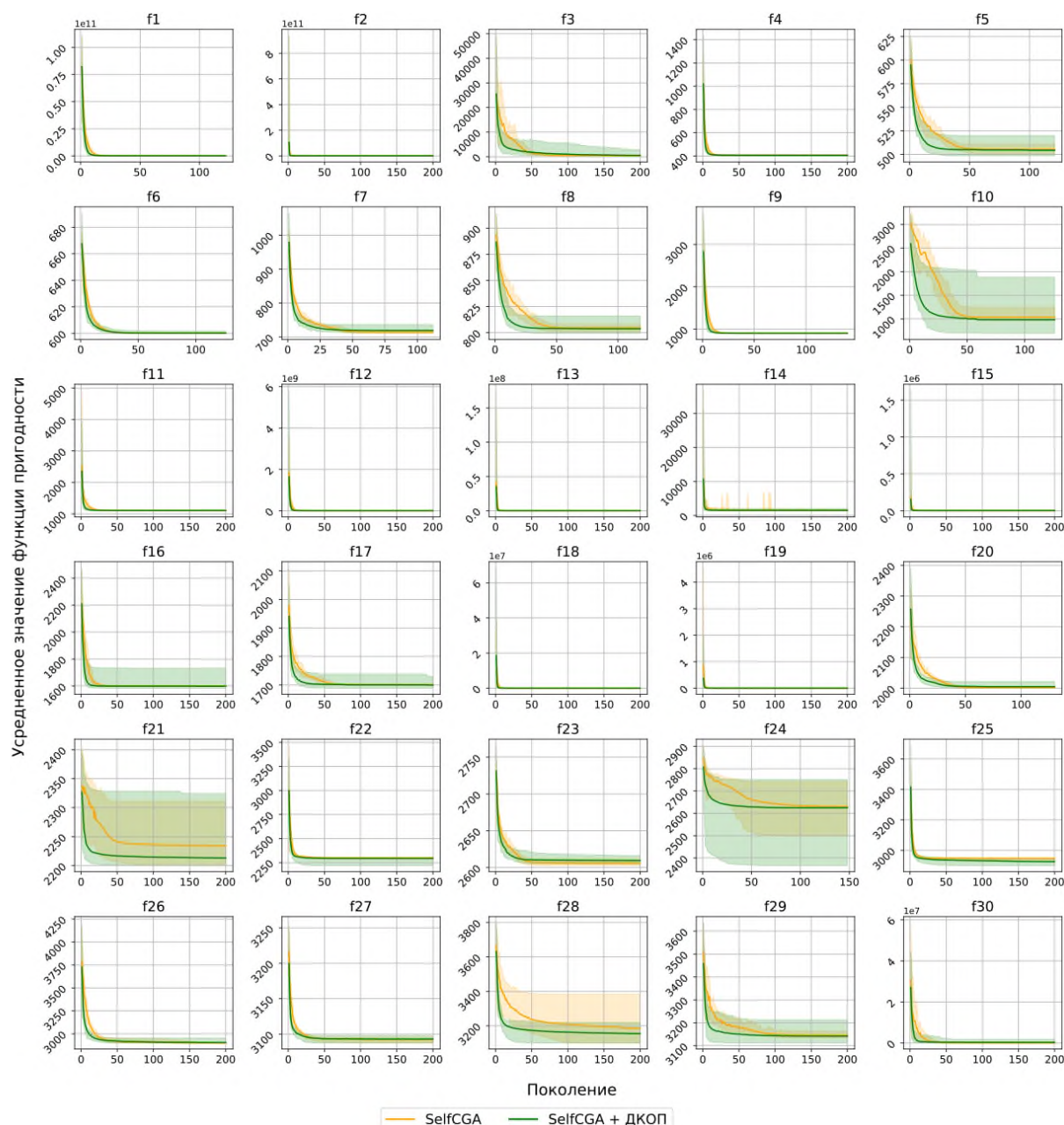


Рисунок 2 – Динамика SelfCGA и SelfCGA+ДКОП при  $D = 10$   
Figure 2 – Dynamics of SelfCGA and SelfCGA+DKOP at  $D = 10$

Анализ данных из Таблицы 2 показывает, что модификация ДКОП улучшила среднее значение функции пригодности для 28 из 30 задач. Почти во всех случаях также получены лучшие значения по отдельным запускам, что подтверждается низкими  $p$ -values большинства тестов Уилкоксона. Лишь для функций  $f_{14}$  и  $f_{19}$  наблюдается обратная ситуация или отсутствие статистически значимых различий.

Таблица 2 – Сравнение значений функций пригодности ГА и ГА+ДКОП с размерностью  $D = 10$  (лучшее значение выделено жирным шрифтом)

Table 2 – Comparison of the values of the fitness functions of GA and GA+DCOP with dimension  $D = 10$  (the best value is highlighted in bold)

Функ.	Средн. ГА	Средн. ГА+ДКОП	Лучш. ГА	Лучш. ГА+ДКОП	p-value
f1	18259680330,00	<b>2305,75</b>	12193893048,00	<b>606,88</b>	0,0022
f2	188425951,20	<b>3626767,79</b>	14101212,26	<b>978,34</b>	0,0037
f3	7560,04	<b>2834,02</b>	5381,84	<b>514,12</b>	0,0022
f4	514,58	<b>408,49</b>	461,67	<b>404,68</b>	0,0022
f5	555,48	<b>512,69</b>	543,77	<b>505,80</b>	0,0022
f6	634,01	<b>601,99</b>	623,38	<b>600,56</b>	0,0022
f7	837,93	<b>728,67</b>	810,91	<b>719,71</b>	0,0022
f8	851,65	<b>812,84</b>	840,83	<b>806,09</b>	0,0022
f9	1419,68	<b>913,89</b>	1169,00	<b>900,80</b>	0,0022
f10	2351,35	<b>1411,05</b>	2124,14	<b>1151,84</b>	0,0022
f11	1305,10	<b>1112,05</b>	1177,71	<b>1106,23</b>	0,0022
f12	104024999,50	<b>940203,40</b>	36976398,89	<b>5798,04</b>	0,0022
f13	269692,31	<b>10846,95</b>	31661,54	<b>1399,33</b>	0,0029
f14	1748,59	<b>1717,67</b>	<b>1460,61</b>	1575,84	0,8753
f15	6821,19	<b>4419,92</b>	1631,26	<b>1519,15</b>	0,0154
f16	1770,87	<b>1657,24</b>	1669,79	<b>1600,58</b>	0,0029
f17	1802,28	<b>1721,26</b>	1769,30	<b>1702,52</b>	0,0022
f18	110399,24	<b>9606,37</b>	12754,05	<b>1972,74</b>	0,0022
f19	6326,42	<b>5828,06</b>	<b>1914,63</b>	2059,07	0,6949
f20	2110,77	<b>2015,27</b>	2080,65	<b>2004,51</b>	0,0022
f21	2230,01	<b>2226,98</b>	2213,50	<b>2201,06</b>	0,0120
f22	2390,95	<b>2307,25</b>	2321,56	<b>2303,06</b>	0,0022
f23	2665,64	<b>2614,82</b>	2624,61	<b>2609,58</b>	0,0022
f24	<b>2649,72</b>	2702,46	2577,01	<b>2559,83</b>	0,0036
f25	3026,46	<b>2945,29</b>	2974,21	<b>2922,88</b>	0,0022
f26	3226,96	<b>3010,48</b>	3115,84	<b>2951,33</b>	0,0022
f27	3127,86	<b>3089,62</b>	3120,19	<b>3081,58</b>	0,0022
f28	3319,99	<b>3285,96</b>	3282,33	<b>3132,89</b>	0,0342
f29	3268,43	<b>3171,42</b>	3204,77	<b>3144,75</b>	0,0022
f30	4079767,52	<b>98792,64</b>	1812117,93	<b>4883,87</b>	0,0022

По данным Таблицы 3, модификация ДКОП дала более низкие средние значения функции пригодности для большинства задач, что отражается также на лучших отдельных запусках. Алгоритмы  $f_2$ ,  $f_{12}$  и  $f_{13}$  оказались исключением, где SelfCGA в

базовой форме сохранил преимущество, однако распределение значений для этих функций имеет вариативность, что также подтверждают значения  $p$ -value.

Таблица 3 – Сравнение значений функций пригодности SelfCGA и SelfCGA+ДКОП с размерностью  $D = 10$  (лучшие значения выделены жирным шрифтом)

Table 3 – Comparison of the values of the fitness functions of SelfCGA and SelfCGA+DCOP with dimension  $D = 10$  (the best values are highlighted in bold)

Функ,	Средн, SelfCGA	Средн, SelfCGA+ДКОП	Лучш, SelfCGA	Лучш, SelfCGA+ДКОП	p-value
f1	280,62	<b>183,47</b>	104,98	<b>104,97</b>	0,0293
f2	<b>4163,99</b>	4861,15	1050,91	<b>203,57</b>	0,0763
f3	487,56	<b>308,42</b>	303,83	<b>300,11</b>	0,0007
f4	406,66	<b>405,89</b>	405,90	<b>400,74</b>	0,0206
f5	510,40	<b>505,22</b>	503,98	<b>500,99</b>	0,0040
f6	600,16	<b>600,01</b>	600,00	<b>600,00</b>	0,0006
f7	719,33	<b>714,00</b>	712,13	<b>710,94</b>	0,0019
f8	809,70	<b>804,53</b>	804,97	<b>800,99</b>	0,0001
f9	900,00	<b>900,00</b>	900,00	<b>900,00</b>	0,0065
f10	1357,17	<b>1032,66</b>	1021,82	<b>1000,25</b>	0,0000
f11	1106,68	<b>1102,61</b>	1103,06	<b>1100,94</b>	0,0001
f12	<b>12435,56</b>	20640,04	<b>5373,09</b>	2437,83	0,0859
f13	<b>6651,76</b>	7932,13	<b>1331,77</b>	1361,03	0,8519
f14	1501,06	<b>1455,49</b>	1406,19	<b>1403,32</b>	0,2043
f15	2719,73	<b>2120,62</b>	1502,03	<b>1500,37</b>	0,0175
f16	1617,69	<b>1600,87</b>	1600,47	<b>1600,41</b>	0,0025
f17	1712,84	<b>1701,56</b>	1700,10	<b>1700,45</b>	0,0003
f18	10037,20	<b>4003,75</b>	1838,72	<b>2533,35</b>	0,0025
f19	2679,87	<b>2475,37</b>	<b>1902,72</b>	1903,54	0,0054
f20	2004,95	<b>2001,13</b>	2000,08	<b>2000,38</b>	0,0304
f21	2234,30	<b>2213,30</b>	2200,00	<b>2200,00</b>	0,0054
f22	2300,13	<b>2292,42</b>	2300,00	<b>2218,08</b>	0,0110
f23	2609,51	<b>2606,19</b>	2604,35	<b>2602,60</b>	0,0040
f24	2663,77	<b>2629,82</b>	2500,00	<b>2400,00</b>	0,0056
f25	2945,43	<b>2925,36</b>	2936,07	<b>2899,46</b>	0,0251
f26	2905,15	<b>2900,01</b>	2900,00	<b>2900,00</b>	0,0086
f27	3092,85	<b>3092,09</b>	<b>3087,33</b>	3087,36	0,0539
f28	3184,07	<b>3154,87</b>	3100,00	<b>3100,00</b>	0,0068
f29	3172,46	<b>3146,01</b>	3138,51	<b>3134,81</b>	0,0006
f30	333959,65	<b>39514,85</b>	10264,70	<b>5653,84</b>	0,0045



Для интерпретации полученных результатов необходимо учитывать структуру тестового набора СЕС2017. В него входят функции таких типов, как унимодальные (f1–f2), многоэкстремальные (f3–f10), гибридные (f11–f20) и композиционные функции (f21–f30), характеризующиеся сложным, неоднородным рельефом и большим числом локальных экстремумов. Анализ результатов вычислительных экспериментов показывает, что влияние ДКОП проявляется по-разному в зависимости от класса функций. На унимодальных функциях преимущество SelfCGA+ДКОП выражается главным образом в ускорении выхода на область стабильных значений, тогда как конечное качество решений сопоставимо с базовой версией алгоритма. На многоэкстремальных функциях и функциях со сложным ландшафтом (f3–f10, f11–f20) эффект ДКОП становится более существенным: наблюдается как более быстрое снижение значения функции пригодности, так и меньшая вариативность результатов между независимыми запусками, что указывает на снижение риска преждевременной сходимости. Наиболее заметное преимущество модифицированного алгоритма проявляется на композиционных функциях (f21–f25), где сочетание кластеризации фенотипов и выборочной замены индивидов позволяет эффективнее перераспределять вычислительные ресурсы между перспективными областями пространства поиска. При увеличении размерности с 10 до 50 различия между алгоритмами усиливаются, что свидетельствует о положительном влиянии ДКОП на масштабируемость SelfCGA.

### Заключение

Проведенное исследование показало, что включение ДКОП в структуру SelfCGA способствует улучшению результатов оптимизации на большинстве функций набора СЕС2017. Механизм кластеризации фенотипов и выборочной замены индивидов обеспечивает сохранение разнообразия и устойчивую сходимость даже при увеличении размерности пространства. При сравнении средних и лучших значений наблюдается преимущество модифицированного алгоритма. Метод не требует дополнительной настройки параметров и может быть интегрирован в другие эволюционные алгоритмы. Полученные результаты подтверждают перспективность предложенного подхода и обосновывают дальнейшее развитие адаптивных механизмов поиска.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Хритonenko Д.И., Семенкин Е.С. Самоконфигурируемый эволюционный алгоритм автоматического проектирования рекуррентных нейронных сетей. В сборнике: *Информационно-телекоммуникационные системы и технологии: Всероссийская научно-практическая конференция, 16–17 октября 2015 года, Кемерово, Россия*. Кемерово: Кузбасский государственный технический университет имени Т.Ф. Горбачева; 2015. С. 220.
2. Шерстнев П.А., Семенкин Е.С. SelfCSHAGA: самоконфигурируемый генетический алгоритм оптимизации с адаптацией на основе истории успеха. *Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение*. 2025;(2):122–139.  
Sherstnev P.A., Semenkina E.S. SelfCSHAGA: self-configuring genetic optimization algorithm with the success-history based adaptation. *Herald of the Bauman Moscow State Technical University. Series: Instrument Engineering*. 2025;(2):122–139. (In Russ.).
3. Semenkina E., Semenkina M. Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator. In: *Advances in Swarm Intelligence: Third International Conference, ICSI 2012: Proceedings: Part I, 17–20 June 2012, Shenzhen, China*. Berlin, Heidelberg: Springer; 2012. P. 414–421. [https://doi.org/10.1007/978-3-642-30976-2\\_50](https://doi.org/10.1007/978-3-642-30976-2_50)

4. Малашин И.П., Тынченко В.С. Применение методов кластерного анализа для динамической коррекции области поиска в генетическом алгоритме. *Сибирский аэрокосмический журнал*. 2025;26(3):318–333. <https://doi.org/10.31772/2712-8970-2025-26-3-318-333>  
Malashin I.P., Tynchenko V.S. Application of cluster analysis methods for dynamic search space adjustment in genetic algorithms. *Siberian Aerospace Journal*. 2025;26(3):318–333. <https://doi.org/10.31772/2712-8970-2025-26-3-318-333>
5. Малашин И.П. Процедура динамической модификации схемы бинарного кодирования индивидов в генетическом алгоритме. *Моделирование, оптимизация и информационные технологии*. 2025;13(3). <https://doi.org/10.26102/2310-6018/2025.50.3.040>  
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.). <https://doi.org/10.26102/2310-6018/2025.50.3.040>
6. Semenkin E., Semenkina M. Self-configuring genetic programming algorithm with modified uniform crossover. In: *2012 IEEE Congress on Evolutionary Computation, 10–15 June 2012, Brisbane, QLD, Australia*. IEEE; 2012. P. 1–6. <https://doi.org/10.1109/CEC.2012.6256587>
7. Осипова Ю.А., Лавров Д.Н. Применение кластерного анализа методом k-средних для классификации текстов научной направленности. *Математические структуры и моделирование*. 2017;(3):108–121. <https://doi.org/10.24147/2222-8772.2017.3.108-121>  
Osipova U.A., Lavrov D.N. Application of cluster analysis by the k-means method for the classification of scientific texts. *Mathematical Structures and Modeling*. 2017;(3):108–121. (In Russ.). <https://doi.org/10.24147/2222-8772.2017.3.108-121>
8. Stanovov V., Akhmedova Sh., Semenkin E. LSHADE Algorithm with Rank-Based Selective Pressure Strategy for Solving CEC 2017 Benchmark Problems. In: *2018 IEEE Congress on Evolutionary Computation (CEC), 08–13 July 2018, Rio de Janeiro, Brazil*. IEEE; 2018. P. 1–8. <https://doi.org/10.1109/CEC.2018.8477977>
9. Miller B.L., Goldberg D.E. Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*. 1995;9(3):193–212.
10. Ahn Ch.W., Ramakrishna R.S. Elitism-based compact genetic algorithms. *IEEE Transactions on Evolutionary Computation*. 2003;7(4):367–385. <https://doi.org/10.1109/TEVC.2003.814633>

#### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Малашин Иван Павлович**, инженер-программист, Научно-образовательный центр ФНС России и МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.  
e-mail: [ivan.p.malashin@gmail.com](mailto: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.

**Сопов Евгений Александрович**, доктор технических наук, доцент, профессор, Сибирский университет науки и технологий, Красноярск, Российская Федерация.  
e-mail: [evgenysopov@gmail.com](mailto:evgenysopov@gmail.com)  
ORCID: [0000-0003-4410-7996](https://orcid.org/0000-0003-4410-7996)

**Evgenii A. Sopov**, Doctor of Engineering Sciences, Docent, Professor, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, the Russian Federation.

*Статья поступила в редакцию 16.12.2025; одобрена после рецензирования 26.01.2026;  
принята к публикации 05.02.2026.*

*The article was submitted 16.12.2025; approved after reviewing 26.01.2026;  
accepted for publication 05.02.2026.*