Enhancing Genetic Algorithms Optimization Using Epigenetic-Inspired Mechanisms

Kornel Chromínski *, Małgorzata Przybyła-Kasperek, and Rafał Skinderowicz

Institute of Computer Science University of Silesia in Katowice Sosnowiec, Poland
Email: kornel.chrominski@us.edu.pl (K.C.); malgorzata.przybyla-kasperek@us.edu.pl (M.P.-K.); rafal.skinderowicz@us.edu.pl (R.S.)
*Corresponding author

***Abstract*—Genetic Algorithms (GA), developed several decades ago, are widely used to solve optimization problems. Since the invention of genetic algorithms, significant progress has been made in genetics, leading to the discovery of many new mechanisms of heredity that were not previously incorporated into genetic algorithms. One such group is epigenetic processes. The article presents two epigenetic-based operations used in genetic algorithms: cytosine methylation and allele inactivation (gene silencing). This article presents two original versions of genetic algorithms that use epigenetic processes. It presents the results of research on the impact of the introduced modifications on the genetic algorithm's performance. The study examined two optimization problems: the loading optimization problem and the outlier detection problem. The experiments examined the optimal probability of occurrence of the introduced modifications, the average number of iterations of the genetic algorithm with and without modifications required to obtain the optimal result, and compared execution times. The research confirmed that introducing new operations into genetic algorithms can improve their performance.**

***Keywords*—Genetic Algorithms (GA), epigenetic process, optimization**

I. Introduction

Nature-inspired algorithms have long been a paradigm in computer science and optimization [1, 2]. Based on observations of nature, algorithms have been developed that use operations mimicking natural processes such as evolution, swarm behavior, or physical phenomena [3, 4]. Nature-inspired algorithms have found particular application in optimization problems that cannot be fully solved by classical deterministic algorithms in a reasonable time or with limited resources, especially for Non-deterministic Polynomial (NP)-hard problems [5, 6]. Optimization problems are a class of problems for which we can define an objective function and a search space, and then iteratively select appropriate values for decision variables in order to obtain an (approximate) optimal value of the objective function [7]. Among the most commonly used metaheuristics for such problems are evolutionary algorithms, which maintain a population of potential solutions and apply biology-inspired operators such as selection, crossover, and mutation [8, 9]. The subject of this study is a specific type of nature-inspired algorithms—Genetic Algorithms (GAs). GAs are a class of evolutionary algorithms and are successfully used in engineering design, planning, data mining, and many other fields [10–12]. Recent research confirms that despite the emergence of many new metaheuristics, GAs remain a competitive foundation and an effective component of hybrid systems [13, 14]. The idea of genetic algorithms is based on the theory of evolution and the processes occurring in the cells of living organisms [8]. Classical GA algorithms were developed in the 1960s and 1970s, with significant contributions from Holland [8] and later Goldberg *et al.* [9]. Their canonical formulation models the basic mechanisms of heredity, such as recombination and mutation, but does not explicitly account for the recently discovered molecular mechanisms of gene regulation. This does not mean that there have been no attempts to modify the classical genetic algorithm and introduce new operations [15, 16]. Examples of such new approaches include Hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) [17], Modified Genetic Algorithm (MGA) [18], a fuzzy genetic algorithm [19], and approaches that implement operations from molecular biology [20]. Over the past two decades, there has been significant progress in biology and molecular genetics, with the discovery and detailed characterization of many new mechanisms of inheritance and adaptation, such as DNA methylation, histone modification, non-coding RNAs, and allele-specific expression [15, 16, 21]. These mechanisms are often grouped under the term “epigenetic processes” and understood as heritable changes in gene function that do not involve changes in the underlying DNA sequence [22]. Epigenetic processes have a significant impact on inheritance and influence the functioning and adaptation of living organisms [15, 22]. These processes can influence gene expression in response to environmental conditions, thereby affecting phenotypic variability, resistance, and evolutionary adaptation. This has led, in recent years, to the development of

Manuscript received February 6, 2026; revised February 26, 2026; accepted March 16, 2026; published June 10, 2026.

doi: 10.12720/jait.17.6.1057-1063

computational models and algorithms inspired by epigenetics, including epigenetic evolutionary algorithms, epigenetic learning mechanisms, and hybrid GA-epigenetic schemas [23–25]. Previous work suggests that incorporating additional layers of heritable state beyond the classical genotype can improve the balance between exploration and exploitation and resilience to premature convergence [23, 26]. However, the range of epigenetic mechanisms considered in current computational models remains limited, and there is no consensus on how best to map specific biological processes to algorithmic operators [24, 27]. This represents a research gap that motivates further investigation. In particular, classical genetic algorithms have not been able to account for all currently known processes of heredity and species variability, as many of these biological processes have only recently been characterized. It is therefore natural to ask whether some of these mechanisms can be incorporated into genetic algorithms to improve their performance on difficult optimization problems. In this work, we focus on two well-studied epigenetic processes: cytosine methylation and allele exclusion (allelic-specific silencing). In biological systems, both processes play an important role in regulating gene expression, controlling gene activity, and shaping the phenotypic response to environmental stimuli [16, 21]. The starting point for introducing new operations into the genetic algorithm (modeled on epigenetic processes) is the hypothesis that, since these processes are of great importance in nature for the adaptation of living organisms to environmental conditions, their appropriate incorporation into genetic algorithms can improve their effectiveness and robustness. This article therefore presents a modified genetic algorithm enhanced with two additional operators that mimic epigenetic processes: one inspired by cytosine methylation and the other by allelic exclusion. The proposed approach is part of a wider trend in research on epigenetic evolutionary computation, while filling a gap in the clear modeling of these two mechanisms at the level of GA variability operators [23, 25, 26]. The experiments presented later in this article evaluate the impact of the proposed epigenetic operators on the performance of genetic algorithms.

II. Problem Statement

In this article, we consider a general optimization problem with a single objective and box constraints.

$$\min_{x\in\Omega} f(x),\ \ \Omega = \{x \in \mathbb{R}^n | \ell_i \leq x_i \leq u_i, i = 1, \dots, n\}, \quad (1)$$

where x is the decision vector, ℓ_i, u_i are the lower and upper bounds, and $f : \Omega \longrightarrow \mathbb{R}$ is the objective (utility) function. Maximization problems can be treated analogously by transforming the objective or selection scheme.

We assume that $f(x)$ is treated as a black box, may be nonlinear, nonconvex, and multimodal, and gradient information is unavailable or unreliable. In many practical cases (including load-optimization and outlier-detection problems, which will be discussed later in this article), the decision vector may also contain discrete elements. In such cases, classical deterministic methods are difficult to apply or prone to premature convergence, which motivates the use of genetic algorithms. The proposed genetic algorithm with epigenetic operators is used as a solution for problems of the form Eq. (1), with definitions x, bounds (ℓ_i, u_i), and $f(x)$ specific to the problem at hand.

III. Genetic Algorithms

Genetic algorithms have been used to solve optimisation problems when deterministic approaches have not been entirely successful [8, 28, 29]. The idea of a genetic algorithm, as an algorithm inspired by the theory of evolution, was presented by John Holland in the 1960s and then developed by David E. Goldberg in the 1980s [8, 9]. In genetic algorithms, the terminology was borrowed from the biological sciences and adapted to their needs. The basic genetic algorithm consists of the following steps [7]:

- The first necessary step in genetic algorithms is to choose the appropriate way to encode an individual—this has a significant impact on whether the algorithm will produce the expected result.
- The next step in constructing a task for a genetic algorithm is to construct an appropriate individual evaluation function (fitness function). The fitness function is a value that shows how good a given solution is.
- In the next step, the selection of individuals is carried out based on the fitness function, and the selected individuals participate in the reproduction process, i.e., the creation of a new population.
- The final step is a mutation operation that introduces diversity into the population and prevents the genetic algorithm from becoming stuck in a local optimum.

The selection, reproduction, and mutation operations are performed until a satisfactory result is achieved or the specified number of iterations is reached.

IV. Epigenetic Operations

This section describes two proposed new operations in the genetic algorithm based on epigenetic processes. Epigenetics is the science that studies non-genetic inheritance processes and the influence of external factors on gene expression levels. Gene expression determines the phenotypic characteristics of an individual, i.e., its adaptation to the environment, behaviour, appearance, etc. Concepts related to epigenetics emerged when it was discovered that some changes in the genotype of living organisms are not directly related to DNA structure, its changes, or inheritance processes. This led to speculation about what could be causing these changes [21–23].

The research led to the discovery of numerous molecules that influence processes in living organisms and affect how the genetic code is read. In other words, how an individual’s phenotype changes without changing their DNA. The opera-tions proposed in the article are based on two such processes: cytosine methylation and allelic exclusion (also known as gene silencing) [22, 30].

A. Cytosine Methylation

Cytosine methylation is a process involving the attachment of methyl groups (-CH3) to the nitrogenous bases of nucleotides (the basic building blocks of DNA and RNA nucleic acids) [22, 23]. The attachment of a methyl group to nucleotides reduces the level of expression of genes encoded by a given DNA fragment. When a large fragment of a DNA strand is methylated, the fragment may be blocked, preventing the gene from being read. The cytosine methylation process may also affect the transmission of genetic information by preventing the inheritance of a specific gene.

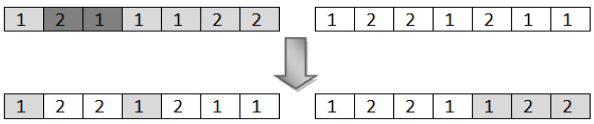

Fig. 1. Operation imitating the cytosine methylation process.

The proposed modification simulates the process of blocking a fragment of an individual's genotype sequence. The genetic algorithm simulating cytosine methylation (Epi_MC) modification has been implemented in the crossbreeding of individuals, modifying these operations. Offspring individuals contain fragments of each parent's genotype (see Fig. 1). If a process simulating cytosine methylation occurs during crossing, a specific fragment of the individual's genotype is blocked, preventing it from participating in the crossing. This means that the fragment of the genome undergoing cytosine methylation will not be passed on to offspring, i.e., it will not appear in the new population.

Algorithm 1: Modified crossover algorithm simulating cytosine methylation (Epi_MC)

Require: Population of individuals, Methylation probability (P_m), Fitness function (F_p)
Ensure: New population created via crossover with Epi_2 modification

```
Select pairs of individuals for crossover with Epi_2 modification with probability P_m
for all selected pair of individuals do
   Randomly determine the length and location of the fragment to be blocked
Block the genotype fragment in the selected individual
  if F_p(individual_1) > F_p (individual_2) then
     Block the genotype fragment in individual_2
  else
     Block the genotype fragment in individual_1
  end if
     Perform crossover operation excluding the blocked fragment
end for
```

B. Allelic Exclusion

Allelic silencing is a biological process in which a DNA fragment encoding a given gene is silenced. As a result of its silencing, another encoding fragment is activated, for example, a modified version of a given gene [21, 30]. Allelic exclusion results in changes in the proteins encoded by DNA. In eukaryotes (multicellular organisms), when two genes encoding the same protein are inherited from different parents, allelic exclusion can determine which version of the genetic code is activated (i.e., which form of the protein the organism produces).

The proposed genetic algorithm with allelic exclusion operation (Epi_GS) modification, which mimics allelic exclusion, is an additional operation in genetic algorithms. In the proposed modification, allelic exclusion is simulated as a process that modifies a specific fragment of an individual's genotype sequence (see Fig. 2). The fragment of the genotype is deactivated and replaced with a new fragment. The modification of a fragment of an individual's genotype in the process of allelic exclusion takes place in two ways (each with a probability of 50%), as two separate (independent) operations:

- Genome rearrangement—the deactivated fragment of the genotype undergoes a rearrangement process, i.e., its genes are randomly swapped in a given fragment of the individual's genotype;
- Sequence replacement—the deactivated fragment of the individual's genotype is replaced by another fragment corresponding in length to the deactivated fragment. In the proposed solution, the deactivated fragment is re-placed with a new one generated at random, in accordance with the method for coding individuals in the given genetic algorithm.

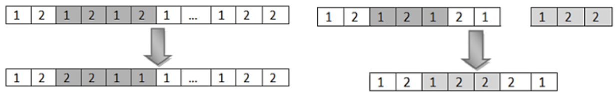

Fig. 2. Operation imitating the allelic exclusion process.

Algorithm 2: Algorithm for Allelic Exclusion Operation

Require: Population of individuals, Probability of modification occurrence (P_m)
Ensure: Population of individuals subjected to allelic exclusion

```
Create a subpopulation of the worst-performing individuals
  (negative selection)
for all selected individual in the subpopulation do
Randomly determine the type of allelic exclusion
Randomly determine the length of the genotype change
Randomly determine the location of the modification
if type = genome rearrangement then
Perform genome rearrangement at the selected location
  else
  Create a new genotype fragment
   Replace the existing genotype fragment with the new
     fragment
end if
end for
```

V. Experiments

The experiments were conducted to verify whether the proposed new operations in genetic algorithms improve the algorithm's performance, primarily by reducing running time, accelerating convergence, and preventing local optimization. The study used two problems and genetic algorithms developed for them. The first is the load optimization problem (i.e., the classic knapsack problem), and the second is the detection of outliers in data sets.

It should be emphasized that the aim of this study is not to propose another competitive evolutionary optimizer, but to investigate whether the introduction of operators inspired by epigenetic processes is beneficial within the standard GA structure. For this reason, we deliberately use a simple genetic algorithm and compare it with its epigenetic extension (EGA). A broader comparison with other evolutionary algorithms and GA variants (e.g., with different crossover schemes or hybrid GA-PSO approaches) is beyond the scope of this article and is planned as future research.

A. The Loading Optimisation Problem

The first problem presented, along with the genetic algorithm that solves it, is optimal container loading. The problem to be solved is to find the optimal container load to maximize profit from the transported goods, while not exceeding the container's permissible total weight. Formal definition of the problem (see Eqs. (2) and (3)):

There is a container with a maximum capacity of V_k and a set of N elements $\{x_1, \ldots, x_j, \ldots, x_N\}$, where each element of the set has a specific weight m_j and value c_j, $j \in \mathbb{N}$. Maximise:

$$\sum_{j=1}^{N} c_j x_j \tag{2}$$

Assumptions:

$$\sum_{j=1}^{N} m_j x_j \le V_k, x_j = 0 \text{ or } 1, j = 1, \ldots, N \tag{3}$$

In the load optimisation experiments, five data sets were used, differing in the possible number of goods to be transported and the maximum permissible weight per container. The data sets are artificial sets created for the experiments. A description of the data sets for the loading optimisation algorithm is presented in Table I.

TABLE I. DATA USED IN THE LOADING OPTIMISATION PROBLEM

Data sets	Number of goods to be transported	Maximum permissible load weight
dataset 1	10	20
dataset 2	20	40
dataset 3	40	80
dataset 4	80	160
dataset 5	160	320

B. Outlier Detection

Detecting outliers in data sets is a task in data analysis. The presence of outliers in a data set can distort data analysis results. Outliers may also be what one is looking for in data analysis, as they are anomalies that appear in the dataset; it is therefore important to detect their presence at an early stage. The formal definition of the problem of detecting outliers can be formulated as follows (Eq. (4)):

For a given set D consisting of N points $\{x_1, \ldots, x_j, \ldots, x_N\}$, where each point is a multidimensional vector m of attributes A, we search for a subset $O \subseteq D$ of size K in such a way as to minimise the entropy $E(D - -O)$ (i.e., the average amount of information):

$$\min_{O \subseteq D} E(D - -O) \tag{4}$$

where $|O| = k$.

In experiments concerning the applicability of operations mimicking epigenetic processes in an algorithm for detecting outliers, six data sets shown in Table II were used.

TABLE II. DATA SETS USED IN OUTLIER DETECTION EXPERIMENTS

Data sets	Name	Size	Columns	Outliers
set 1	Artificial set	2000	3	190
set 2	Thyroid Disease [31]	3772	6	93
set 3	Breast Cancer [32]	570	30	212
set 4	Diabetes [33]	768	8	268
set 5	Glass Identification [34]	214	7	9
Set 6	Pen-Based Recognition [34]	6870	16	156

C. Results for the Loading Optimisation Problem

The first group of experiments used a genetic algorithm to solve the loading optimisation problem. The first thing to empirically verify was the optimal probability of occurrence for the proposed epigenetic operations. The experiments yielded similar optimal probability values for both proposed operations. Table III presents the average number of operations required to obtain the optimal result using the basic genetic algorithm (without new operations) and with different probabilities of occurrence of new operations. The probability values for the operations were set to the same levels for both.

TABLE III. THE NUMBER OF ITERATIONS REQUIRED TO ACHIEVE THE OPTIMAL RESULT, FOR DIFFERENT PROBABILITIES OF OCCURRENCE OF THE PROPOSED MODIFICATIONS AND FOR THE BASIC GENETIC ALGORITHM

Probability	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5
basic GA	154	631	973	1355	1513
5%	189	623	885	1378	1523
10%	167	617	877	1369	1492
20%	144	436	760	1278	1533
30%	132	419	627	1202	1490
40%	124	***413***	***620***	1101	***1356***
50%	***119***	541	665	***1077***	1415
60%	134	602	736	1123	1489
70%	139	604	731	1169	1523
80%	150	640	816	1235	1519
90%	172	630	867	1294	1565
100%	187	666	906	1403	1598

Based on Table III, it can be seen that the optimal probability of occurrence of the proposed epigenetic operations is approximately 40–50%, at which point the algorithm finds the optimal solution in the fewest iterations.

Fig. 3 shows the change in the fitness function value for the base algorithm and the algorithm with epigenetic operations at optimal probability; results are presented for all data sets.

Based on Fig. 3, it can be observed that the introduction of epigenetic operations leads to faster changes in the fitness function and faster attainment of the optimal value.

Table IV compares the average execution times of a single generation in the basic genetic algorithm and with epigenetic operations. Although the average time per iteration is higher for the modified algorithm, reducing the number of iterations can speed up the entire algorithm.

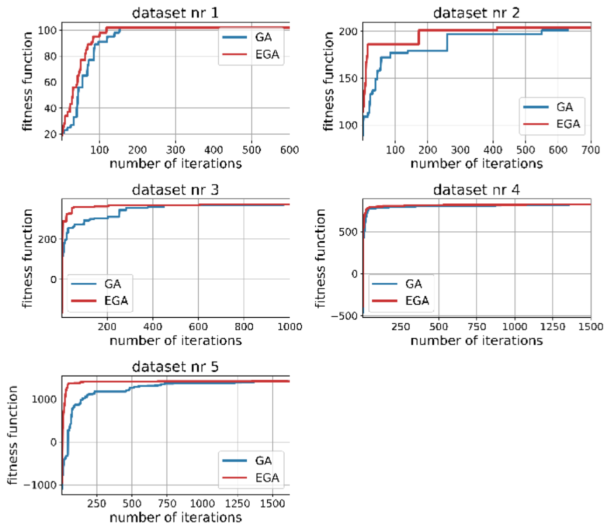

Fig. 3. Change in the value of the fitness function in subsequent generations, for individual data sets for the basic genetic algorithm and with epigenetic modifications introduced.

TABLE IV. Mean Time of one Iteration of the Basic Genetic Algorithm and with Epigenetic Operations

Data sets	Mean Epi	sd Epi	Mean GA	sd GA	% change
dataset 1	0.072	0.031	0.063	0.012	12.70%
dataset 2	0.081	0.005	0.072	0.010	11.35%
dataset 3	0.109	0.010	0.098	0.021	10.55%
dataset 4	0.182	0.017	0.169	0.022	7.05%
dataset 5	0.345	0.061	0.318	0.115	8.05%

For the loading optimisation problem, an overall improvement in the performance of the genetic algorithm with new operations based on epigenetics was achieved compared to the basic algorithm.

To further evaluate the stability of the proposed approach, both the basic GA and the GA with Epigenetic Operations (EGA) were run in multiple independent runs for each dataset. Table V shows the average fitness value obtained at the end of the optimization and the corresponding standard deviation.

In the loading optimization problem, the genetic algorithm with epigenetic operations achieved overall performance improvements compared to the basic algorithm. Furthermore, Table V shows that for all datasets, EGA achieves at least the same average efficiency as the baseline GA and, in most cases, a lower standard deviation, indicating more stable behavior in independent runs.

TABLE V. Stability of Solutions for the Loading Optimization Problem: Mean and Standard Deviation of the Final Fitness Value in Repeated Runs for the Basic GA and Epigenetic GA (EGA)

Datasets	EGA mean	EGA sd	GA mean	GA sde
dataset 1	102	0.0	102	0.0
dataset 2	180	0.0	165	12.3
dataset 3	356	5.4	323	18.7
dataset 4	802	7.1	795	15.9
dataset 5	1407	22.3	1231	64.5

D. *Results for the Outlier Detection Problem*

The second problem for which the impact of introducing new operations was examined was detecting outliers. As with the loading optimisation problem, the first step was to determine the optimal probability of occurrence for the proposed operations. The results with the average number of iterations needed to obtain the optimal result (detection of all outliers) for the basic algorithm and with the new operations are shown in Table VI.

TABLE VI. The Number of Iterations Required to Achieve the Optimal Result, for Different Probabilities of Occurrence of the Proposed Modifications and for the Basic Genetic Algorithm

Probability	ds_1	ds_2	ds_3	ds_4	ds_5	ds_6
basic GA	2675	2175	661	740	1809	1479
5%	1932	1568	415	735	1801	1481
10%	***1898***	1564	406	551	1655	1472
20%	1954	***1558***	397	***467***	***1510***	***1419***
30%	2354	1669	***390***	482	1582	1359
40%	2398	1857	409	563	1691	1305
50%	2425	1884	421	609	1709	1378
60%	2619	1928	478	869	1753	1441
70%	2756	1950	523	1052	1806	1522
80%	2887	2106	614	1104	1823	1541
90%	2954	2215	674	1129	1893	1603
100%	3012	2239	687	1206	1933	1622

Based on Table VI, it can be seen that, in the case of outlier detection, the optimal probability of new operations occurring is lower than in the case of the loading optimisation problem and amounts to approx. 20%. Fig. 4 shows how the fitness function value changes across successive generations for the base algorithm and the algorithm with epigenetic operations at optimal probability; results are shown for all data sets.

The fitness function converges more quickly to the optimal value when using the algorithm with epigenetic operations, as shown in Fig. 4.

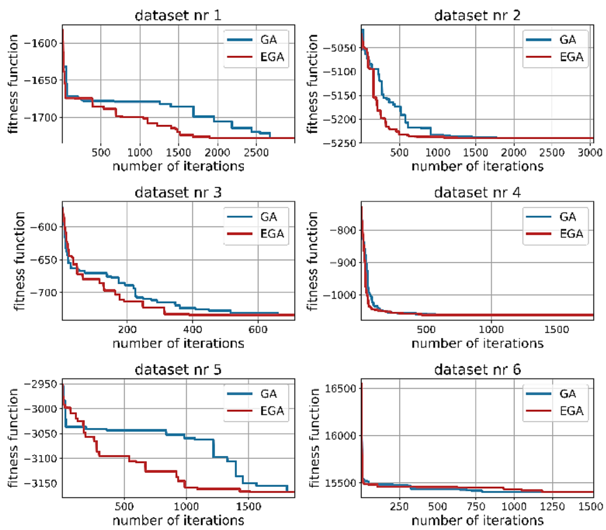

Fig. 4. Change in the value of the fitness function in subsequent generations, for individual data sets for the basic genetic algorithm and with epigenetic modifications introduced.

Following these observations, the last element tested was a comparison of the average execution time of a single generation in the basic genetic algorithm and in the epigenetic algorithm; the results are shown in Table VII.

As in previous experiments, an increase in the average time per iteration was observed, but with the reduced number of iterations, the overall running time of the algorithm may be shorter.

TABLE VII. PMEAN TIME OF ONE ITERATION OF THE BASIC GENETIC ALGORITHM AND WITH EPIGENETIC OPERATIONS

Data set	Mean Epi	sd Epi	Mean GA	sd GA	% change
dataset 1	0.592	0.149	0.567	0.136	4.18%
dataset 2	0.344	0.011	0.331	0.035	3.97%
dataset 3	0.236	0.016	0.225	0.027	4.68%
dataset 4	0.171	0.037	0.161	0.018	5.94%
dataset 5	1.144	0.039	1.135	0.077	0.80%
dataset 6	0.356	0.028	0.341	0.014	4.24%

Similarly, the stability of algorithms for outlier detection was evaluated across multiple runs. Table VIII summarizes the mean and standard deviation of the final fitness values for the basic GA and EGA.

TABLE VIII. STABILITY OF SOLUTIONS FOR THE OUTLIER DETECTION PROBLEM: MEAN AND STANDARD DEVIATION OF THE FINAL FITNESS VALUE IN REPEATED RUNS FOR THE BASIC GA AND EPIGENETIC GA (EGA)

Dataset	EGA Mean	EGA sd	GA Mean	GA sd
dataset 1	−1727	9.3	−1720	24.8
dataset 2	−5239	0.000	−5230	35.7
dataset 3	−732	4.1	−729	9.8
dataset 4	−1062	6.2	−1057	14.3
dataset 5	−3167	18.9	−3123	47.6
dataset 6	15402	203.5	14733	512.7

In the case of the genetic algorithm for outlier detection, overall performance improved thanks to new epigenetic-based operations compared to the basic algorithm. Furthermore, Table VIII shows that EGA provides better average fitness values and consistently lower standard deviations than the basic GA, which indicates greater stability of the obtained solutions despite the stochastic nature of the method.

E. Discussion and Limitations

The experimental results for two optimization tasks (load optimization and outlier detection) indicate that extending a simple GA with the proposed epigenetic operators reduces the number of iterations required to reach the optimum and can improve overall running time, despite a moderate increase in the cost per generation. Additional statistics in Tables V and VIII further show that the epigenetic GA achieves at least comparable and usually better mean fitness values with lower standard deviations than the basic GA. This demonstrates that the proposed approach not only improves performance but also increases solution stability, which is crucial given the stochastic nature of evolutionary algorithms.

At the same time, the study has clear limitations. We limit ourselves to one canonical GA configuration and do not compare it with other evolutionary algorithms or more advanced GA variants, such as single-point, two-point, or real-coded crossover schemes, diffusion-based or matrix-based crossover, PSO algorithms, or hybrid GA-PSO algorithms. This work should therefore be viewed as proof of concept demonstrating the usefulness of epigenetic operators in the simplest GA setting.

VI. CONCLUSION

This article proposes two new operators in the genetic algorithm, inspired by biological epigenetic processes: an operator mimicking cytosine methylation (Epi_MC) and an operator mimicking allelic exclusion (Epi_GS). Both operators have been introduced into the standard genetic algorithm so that they act as additional mechanisms that preserve the population's variability and diversity.

Experimental studies were conducted on two optimization problems: loading optimization and outlier detection. In both cases, introducing the proposed epigenetic operators into the basic genetic algorithm improved performance. The modified algorithms achieved better solution quality and more stable performance compared to the classic GA without epigenetic extensions. These results confirm the hypothesis that introducing mechanisms that mimic epigenetic processes into genetic algorithms can improve their performance.

At the same time, experiments have shown that the improvement depends on the problem and parameter settings. This indicates that the design and tuning of epigenetic operators should account for the problem's characteristics.

Future research will focus on several directions. First, we plan to investigate the impact of the proposed epigenetic operations in other variants of genetic algorithms, including multi-criteria and constrained algorithms, as well as hybrid algorithms. Second, the interactions between Epi_MC and Epi_GS with alternative crossover and mutation operators will be analyzed to identify combinations that yield the best performance of the genetic algorithm. Third, we will conduct extensive experiments on various test sets and real-world applications to evaluate the robustness and generality of the proposed operations. Finally, we intend to explore adaptive schemes in which the intensity or activation of epigenetic operators is automatically controlled during operation, potentially leading to self-tuning epigenetic evolutionary algorithms.

Overall, the results obtained in load optimization and outlier detection indicate that operators inspired by epigenetic processes are a promising direction for enriching genetic algorithms. Further work in this direction may contribute to a more sophisticated integration of biological knowledge with evolutionary computation and to the development of more effective and versatile optimization methods.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All authors contributed equally to the preparation of the article. Methodology: all authors; Experiment design: Kornel Chromiński; Experiment conduct: all authors;

Literature review: Małgorzata Przybyła-Kasperek and Rafał Skinderowicz; Analysis of results: Kornel Chromiński, Małgorzata Przybyła-Kasperek; all authors approved the final version.

Funding

The project was co-financed by the European Union within the program "The European Funds for Slask (Silesia) 2021-2027".

References

[1] T. Chen, S. Li, Y. Qiao, and X. Luo, "A robust and efficient ensemble of diversified evolutionary computing algorithms for accurate robot calibration," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–14, 7501814, 2024. doi: 10.1109/TIM.2024.3363783.

[2] S. C. Chu, Z. Y. Zhuang, J. S. Pan, S. Y. Li, and T. Y. Wu, "Recent evolutionary computing algorithms and industrial applications: A review," in *Proc. ICGEC 2023, International Conference on Genetic and Evolutionary Computing*, Singapore: Springer, 2024. doi: 10.1007/978-981-97-0068-4_46

[3] A. Kumar, M. Nadeem, and H. Banka, "Nature inspired optimization algorithms: A comprehensive overview," *Evolving Syst.*, vol. 14, pp. 141–156, 2023. https://doi.org/10.1007/s12530-022-09432-6

[4] C. Maraveas, P. G. Asteris, K. G. Arvanitis, D. Bartzanas, and D. Loukatos, "Application of bio and nature-inspired algorithms in agricultural engineering," *Arch. Comput. Methods Eng.*, vol. 30, pp. 1979–2012, 2023. https://doi.org/10.1007/s11831-022-09857-x

[5] B. Alhijawi and A. Awajan, "Genetic algorithms: Theory, genetic operators, solutions, and applications," *Evol. Intell.*, vol. 17, pp. 1245–1256, 2024. https://doi.org/10.1007/s12065-023-00822-6

[6] X. Chen, X. Li, T. Chen, H. Liang, and Z. Liu, "Symbolic discovery of optimization algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 49205–49233, 2023.

[7] S. S. Chandra and H. S. Anand, "Nature inspired meta heuristic algorithms for optimization problems," *Computing*, vol. 104, pp. 251–269, 2022. https://doi.org/10.1007/s00607-021-00955-5

[8] J. H. Holland, "Genetic algorithms and adaptation," in *Adaptive Control of Ill-Defined Systems*, Boston, MA, USA: Springer, 1984, pp. 317–333.

[9] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artif. Intell.*, vol. 40, no. 1–3, pp. 235–282, 1989.

[10] M. Gen and L. Lin, "Genetic algorithms and their applications," in *Springer Handbook of Engineering Statistics*, London, U.K.: Springer, 2023, pp. 635–674.

[11] Z. H. Zhan, L. Shi, K. C. Tan, and Y. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, vol. 55, pp. 59–110, 2022. https://doi.org/10.1007/s10462-021-10042-y

[12] D. Yadav, P. Ramu, and K. Deb, "Finding Robust Solutions for Many-Objective Optimization Using NSGA-III," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Chicago, IL, USA, 2023, pp. 1–8. doi: 10.1109/CEC53210.2023.10254160.

[13] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic algorithms for optimization: A brief review," *Eng. Proc.*, vol. 59, no. 1, 238, 2023. https://doi.org/10.3390/engproc2023059238

[14] T. H. W. Bäck, F. Hoffmeister, J. J. Merelo, and C. R. Stephens, "Evolutionary algorithms for parameter optimization—Thirty years later," *Evol. Comput.*, vol. 31, no. 2, pp. 81–122, Jun. 2023. doi: 10.1162/evco a 00325

[15] A. S. Akopov, "Parallel genetic algorithm with fading selection," *Int. J. Comput. Appl. Technol.*, vol. 49, no. 3–4, pp. 325–331, 2014. doi: 10.1504/IJCAT.2014.062368

[16] T. M. Hamdani, A. M. Alimi, and F. Karray, "Distributed genetic algorithm with bi-coded chromosomes and a new evaluation function for features selection," in *Proc. IEEE Int. Conf. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 581–588. doi: 10.1109/CEC.2006.1688362

[17] M. Younespour, M. Esmaelian, and K. Kianfar, "Optimizing the strategic and operational levels of demand-driven MRP using a hybrid GA-PSO algorithm," *Comput. Ind. Eng.*, vol. 193, 110306, 2024.

[18] M. Das, A. Roy, S. Maity, and S. Kar, "Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm," *Decis. Mak. Appl. Manag. Eng.*, vol. 5, no. 2, pp. 329–361, 2022.

[19] S. Papadimitriou, K. Chrysafiadi, and M. Virvou, "Adaptive quizzes using fuzzy genetic algorithm," in *Proc. 14th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Volos, Greece, 2023, pp. 1–8. doi: 10.1109/IISA59645.2023.10345881

[20] Z. Duan, S. Yang, Q. Shao, and M. Yang, "PEGA: Probabilistic environmental gradient-driven genetic algorithm considering epigenetic traits to balance global and local optimizations," *Front. Inf. Technol. Electron. Eng.*, vol. 25, no. 6, pp. 839–855, Jun. 2024. doi: 10.1631/FITEE.2300170.

[21] C. L. St. Pierre, M. S. S. De Almeida, M. C. L. B. de Oliveira, and L. F. S. Oliveira, "Genetic, epigenetic, and environmental mechanisms govern allele-specific gene expression," *Genome Res.*, vol. 32, no. 6, pp. 1042–1057, 2022.

[22] Z. D. Smith, S. Hetzel, and A. Meissner, "DNA methylation in mammalian development and disease," *Nat. Rev. Genet.*, vol. 26, pp. 7–30, 2025. https://doi.org/10.1038/s41576-024-00760-8

[23] S. Yuen, T. H. G. Ezard, and A. J. Sobey, "Epigenetic opportunities for evolutionary computation," *R. Soc. Open Sci.*, vol. 10, no. 5, 221256, May 2023. https://doi.org/10.1098/rsos.221256

[24] K. Chrominski, M. Tkacz, and M. Boryczka, "Epigenetic modification of genetic algorithm," in *Proc. International Conference on Computational Science (ICCS 2020)*, Switzerland: Springer, 2020. https://doi.org/10.1007/978-3-030-50417-5 20

[25] K. Chromiński and M. Boryczka, "Epigenetically inspired modification of genetic algorithm and his efficiency on biological sequence alignment," in *Proc. International Conference on Intelligent Decision Technologies 2016*, Switzerland: Springer, 2016. https://doi.org/10.1007/978-3-319-39627-9 9

[26] M. D. Dilmi, H. Azzag, and M. Lebbah, "Epigenetics algorithms: Self-reinforcement-attention mechanism to regulate chromosomes expression," arXiv preprint, arXiv:2303.10154, 2023.

[27] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*, Heidelberg, Germany: Springer, 2002.

[28] M. Angelova and T. Pencheva, "Tuning genetic algorithm parameters to improve convergence time," *Int. J. Chem. Eng.*, vol. 2011, pp. 1–7, 2011.

[29] D. Ashlock, *Evolutionary Computation for Modeling and Optimization*, New York, NY, USA: Springer, 2005.

[30] C. Dupont, D. Armant, and C. Brenner, "Epigenetics: Definition, mechanisms and clinical perspective," *Semin. Reprod. Med.*, vol. 27, no. 5, pp. 351–357, Aug. 2009.

[31] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus, "Inductive knowledge acquisition: A case study," in *Proc. 2nd Aust. Conf. Appl. Expert Syst.*, Sydney, Australia, 1986.

[32] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," in *Proc. the National Academy of Sciences*, 1990, vol. 87, pp. 9193–9196.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Min.*, Pisa, Italy, 2008, pp. 413–422.

[34] F. Keller, E. Müller, and K. Böhm, "HiCS: High-contrast subspaces for density-based outlier ranking," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Arlington, VA, USA, 2012, pp. 1037–1048.