Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

PEGA: probabilistic environmental gradient-driven genetic algorithm considering epigenetic traits to balance global and local optimizations*#

Zhiyu DUAN, Shunkun YANG†‡, Qi SHAO, Minghao YANG

School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China

†E-mail: ysk@buaa.edu.cn

Received Mar. 10, 2023; Revision accepted Aug. 6, 2023; Crosschecked Mar. 28, 2024

Abstract: Epigenetics' flexibility in terms of finer manipulation of genes renders unprecedented levels of refined and diverse evolutionary mechanisms possible. From the epigenetic perspective, the main limitations to improving the stability and accuracy of genetic algorithms are as follows: (1) the unchangeable nature of the external environment, which leads to excessive disorders in the changed phenotype after mutation and crossover; (2) the premature convergence due to the limited types of epigenetic operators. In this paper, a probabilistic environmental gradient-driven genetic algorithm (PEGA) considering epigenetic traits is proposed. To enhance the local convergence efficiency and acquire stable local search, a probabilistic environmental gradient (PEG) descent strategy together with a multi-dimensional heterogeneous exponential environmental vector tendentiously generates more offsprings along the gradient in the solution space. Moreover, to balance exploration and exploitation at different evolutionary stages, a variable nucleosome reorganization (VNR) operator is realized by dynamically adjusting the number of genes involved in mutation and crossover. Based on the above-mentioned operators, three epigenetic operators are further introduced to weaken the possible premature problem by enriching genetic diversity. The experimental results on the open Congress on Evolutionary Computation-2017 (CEC' 17) benchmark over 10-, 30-, 50-, and 100-dimensional tests indicate that the proposed method outperforms 10 state-of-the-art evolutionary and swarm algorithms in terms of accuracy and stability on comprehensive performance. The ablation analysis demonstrates that for accuracy and stability, the fusion strategy of PEG and VNR are effective on 96.55% of the test functions and can improve the indicators by up to four orders of magnitude. Furthermore, the performance of PEGA on the real-world spacecraft trajectory optimization problem is the best in terms of quality of the solution.

Key words: Evolutionary algorithm; Epigenetics; Epigenetic algorithm; Probabilistic environmental vector; Variable nucleosome reorganization

https://doi.org/10.1631/FITEE.2300170 **CLC number:** TP301

1 Introduction

Evolutionary algorithms (EAs) such as genetic algorithm (GA) (Katoch et al., 2021), genetic programming (GP) (Lin et al., 2020), differential evolution (DE) (Song et al., 2023), and evolutionary strategy (ES) (Coli et al., 2022) are well-known algorithms inspired by the biological evolution process (Slowik and Kwasnicka, 2020). The variants of GAs

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61672080)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2300170) contains supplementary materials, which are available to authorized users

ORCID: Shunkun YANG, https://orcid.org/0000-0002-2580-4529

are classified broadly into real and binary coded, chaotic, multi-objective, parallel, and hybrid ones (Katoch et al., 2021). Nevertheless, no single evolutionary algorithm can guarantee the optimal solution for all optimization problems. As a result, it remains imperative to enhance existing algorithms and devise novel evolutionary mechanisms to address this limitation.

Recently discovered epigenetic phenomena, such as paramutation (Gouil and Baulcombe, 2018), bookmarking (Owens et al., 2020), and imprinting (Thamban et al., 2020), have shed new light on gene regulation. By incorporating the effects of epigenetics on gene expression, epigenetic algorithms (epiGAs) (Makino et al., 2020) have the ability to control evolutionary processes through customizable mutation modes. Early epiGAs integrated degradation, autocatalysis, and epigenetic factors into GAs (Periyasamy et al., 2008; Birogul, 2016). Subsequent research incorporated histone–mediated gene regulation into GP to address predator–prey problems (Tanev and Yuta, 2003). epiGAs have also been applied for sequence comparison (Chromiński and Boryczka, 2016) and network intrusion detection (Ezzarii et al., 2020), leveraging epigenetic mechanisms associated with proteins and Internet protocol (IP) address sequences. In summary, epiGAs provide a means to guide evolutionary computation using principles derived from epigenetics.

The groundbreaking epiGA was introduced as a competitive binary coded approach for tackling combinatorial optimization problems (Stolfi and Alba, 2018). It establishes a cell-environment interaction framework, encompassing multiple vectors that encapsulate information pertaining to the environment, nucleosomes, and parental genes. Building upon this foundation, an epigenetic operator was devised and implemented to address the multi-dimensional knapsack problem. It demonstrated better performance than simulated annealing (Więckowski et al., 2020) and GA, but it faces the problem of insufficient accuracy in solving numerical problems. A significant hurdle to be surmounted prior to the application of epiGA to other large-scale problems is the issue of excessive disorder and premature convergence, which can result in degradation of precision and stability.

To address the challenges outlined herein, a probabilistic environmental gradient-driven genetic algorithm (PEGA) is proposed in this paper. The main contributions of this paper are as follows:

1. Considering the inconsistency of gene weights at different positions on the chromosome, a new operator named multi-dimensional exponential environmental vector (MEEV) is proposed to balance the uneven schema importance by adopting different mutation probabilities for each gene. This operator can make the local suboptimal solution stably converge to the local optimal solution from the perspective of searching scale.

2. A probabilistic environmental gradient (PEG) strategy is proposed based on the local gradient information, and a heterogeneous exponential probabilistic distribution is constructed for individual mutation to generate more offsprings along the gradient in the solution space. This gradient-oriented offspring generating mechanism can speed up the local optimal convergence from the perspective of searching direction.

3. We newly design a variable nucleosome reorganization (VNR) operator for finer-grained recombination control among gene segments in dynamic evolutionary stages. By gradually increasing the nucleosome radius, the global search is gradually transferred to the local search near the suboptimal solution so that the overall evolution process can evolve to the global optimal solution as much as possible to obtain a fast and stable convergence.

4. Based on the above-mentioned operators, three epigenetic operators are introduced, including gene imprinting, position effect, and paramutation, to further enrich the mechanism of mutation and crossover operations for more diverse population genes against the possible premature convergence. Together with MEEV, PEG, and VNR, these epigenetic operators can balance the exploitation and exploration.

A comprehensive experiment of 29 benchmark functions based on the Congress on Evolutionary Computation-2017 (CEC' 17) reveals that compared with the 10 most prominent algorithms, PEGA exhibits the best performance. Combining the test results in four dimensions, the ablation analysis demonstrates that for the accuracy and stability indicators mentioned in this paper, PEG and VNR are effective on the vast majority of the test functions and can improve the accuracy by up to four orders of magnitude.

2 Background and motivation

2.1 Framework of epiGA

The procedure used in the originally proposed epiGA is shown in Algorithm 1 (Stolfi and Alba, 2018). The meanings of the parameters are as follows: N_i is the number of individuals; N_c is the number of cells; P_e is the epigenetic probability; P_n is the nucleosome probability; R is the nucleosome radius; M_e is an epigenetic mechanism to be applied; $\boldsymbol{E}$ represents the environmental vector. First, the population is randomly initialized and the nucleosome vector is generated with a constant radius (lines 1–4). Then, for the cells in each individual $\boldsymbol{I}_j$ in every generation, nucleosome-based reproduction, epigenetic mechanisms, and replacement operations are carried out sequentially. The nucleosome-based reproduction between the randomly selected individuals $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ is shown in lines 5–9, where $\boldsymbol{I}^\mathrm{g}$ and $\boldsymbol{I}^\mathrm{n}$ represent the gene and nucleosome of an individual, respectively, and $\boldsymbol{U}$ is a constructed vector whose elements are 1. For ease of writing, all multiplicative relations in line 8 are Kronecker products. The operator of the epigenetic mechanisms is shown in lines 10–17, where every epigenetic operator is successively applied on epigenetic cells under the influence of a predetermined invariant environment vector. In addition to these operators, epiGAs provide more abundant epigenetic operators, such as gene silencing, paramutation, and gene bookmarking. These epigenetic operators are susceptible to interaction with the environment and each other. This framework applies a single operator called gene silencing to obtain complex mutation patterns. The replacement operator is shown in line 18, which uses an elitist strategy to guarantee the global convergence. The algorithm terminates if the criterion is met; otherwise, it jumps back to line 2 and starts a new iteration.

Algorithm 1 Original epigenetic algorithm

1: Randomly initialize the population P with N_i individuals each containing N_c cells
2: **for** each individual $\boldsymbol{I}_j \in P$ **do**
3: Generate a new nucleosome for its cells using constants P_n and R
4: **end for**
5: **while** size(Q) $< N_\mathrm{i}$ **do**
6: Select individuals $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ from P
7: $\left((\boldsymbol{I}_1^\mathrm{g})' \quad (\boldsymbol{I}_2^\mathrm{g})'\right) \leftarrow \left(\boldsymbol{I}_1^\mathrm{n}\boldsymbol{I}_2^\mathrm{n} \quad \boldsymbol{U} - \boldsymbol{I}_1^\mathrm{n}\boldsymbol{I}_2^\mathrm{n}\right) \begin{pmatrix} \boldsymbol{I}_1^\mathrm{g} & \boldsymbol{I}_2^\mathrm{g} \\ \boldsymbol{I}_2^\mathrm{g} & \boldsymbol{I}_1^\mathrm{g} \end{pmatrix}$
// Perform nucleosome-based reproduction between
// the best cells of $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$
8: Put the modified $\boldsymbol{I}_1'$ and $\boldsymbol{I}_2'$ into Q
9: **end while**
10: **for** each individual $\boldsymbol{I}_Q \in Q$ **do**
11: **for** $M_\mathrm{e} \in$ {Epigenetic mechanisms} **do**
12: Apply epigenetic mechanisms($\boldsymbol{I}_Q$, M_e, $\boldsymbol{E}$, P_e)
13: Evaluation($\boldsymbol{I}_Q$)
14: **end for**
15: **end for**
16: $P \leftarrow P + Q$
17: Make replacements to maintain the number of individuals in P
18: Terminate if it meets the termination criterion; otherwise, go back to line 2

2.2 Motivation of the proposed method

Since the proposal of GA, one of the classic evolutionary theorem based algorithms, by John H. Holland in the 1990s, the idea that mutations occur randomly has dominated the fundamentals of building evolutionary operators. The search process of epiGAs identically depends on these heuristic operators, and the calculation accuracy is improved by persistent iterations containing selection and replacement that remove individuals with lower fitness. Such methods may not impair the performance while dealing with combinatorial optimization (Stolfi and Alba, 2018) but may lead to catastrophic loss of precision for numerical problems, because the stochastic process makes it easier for individuals to move away from the optimal solution than to approach it. Fig. 1a depicts that during a small-scale search about an individual p_0, which locates in the valley near the optimal solution p^*, the evolutionary randomness leaves a disorganized distribution s_0 of individuals generated after a mutation or crossover, and the obtained next generation's optimal individual is inadequate. More specifically, the excessive disorder of the traditional evolutionary process is reflected in two aspects: (1) the aimlessness of individual mutation behavior and (2) the uneven schema importance problem. These individuals with declining competitiveness use more iterations to reach the optimal solution or are eliminated immediately. Even if elite strategies (Pereira et al., 2020) preserve the best individual, these random and unpredictable operations could still unavoidably reduce the search efficiency.

Numerous similar operators can be found in various evolutionary algorithms facing the same predicament. Fortuitously, the laws of nature always give inspiration to the design of algorithms.

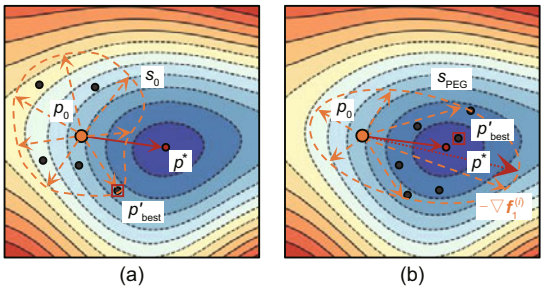

Fig. 1 Individual evolutionary trends under the influence of two types of environmental vectors: (a) randomly scattered offsprings generated by traditional genetic algorithm operators; (b) offsprings with a propensity distribution induced by environmental vector considering gradient information

From a biological perspective, Monroe et al. (2022) expounded how the environment guides mutation through epigenetics. Furthermore, coincidentally, Miikkulainen and Forrest (2021) illustrated plentiful internal phenomena, such as sexual recombination, epistasis modulated by epigenetic traits, and the interactions between cells and the environment, which are worth studying by applying extensive biological diversity to algorithms. Inspired by these factors, we propose a mutation-inducing mechanism that is effective in small-scale search, called the PEG descent method. Because the reversed gradient $-\nabla \boldsymbol{f}_1$ is quite close to the dominant descent direction in small-scale search, the gradient information is introduced into the environmental vector for generating more offsprings in the expected direction and weakening the negative effect of random mutations. Consequently, by comparing s_{PEG} in Fig. 1b with s_0 in Fig. 1a, we can see that under the directional induction of the environmental gradient vector, the algorithm will quickly tend to converge to the optimal solution with better improved accuracy. Then, by systematically combining multiple epigenetic operators involving the epigenetic factors mentioned above, the gradient information in the environmental vector is transmitted to the gene level. In the early stage of evolution, the global search ability is maintained, and when entering the final stage of evolution, the local search ability of each individual is strengthened.

3 The proposed method

In this section, we describe the improved epiGA induced by PEG. The iteration process involves a three-layer structure based on the evolutionary algorithm framework: environmental regulation layer, probabilistic transfer layer, and genetic regulation layer. The first layer imitates environmental impacts, through which the gradient information of the population is collected and the PEG vector is built. Subsequently, the PEG is passed to the genetic level to induce the optimal epigenetic mutations via the probabilistic transfer layer, which integrates multiple epigenetic operators, such as the position effect operator (PEO), gene imprinting operator (GIO), paramutation operator (PMO), and gene silencing operator (GSO). Finally, operators that directly manipulate population genomes, such as selection, VNR, nucleosome-based crossover, fitness evaluation considering epigenetic modifications, and replacement, are carried out sequentially in the third layer. The PEGA framework is shown in Fig. 2.

3.1 Generating environmental vectors with PEG descent method

When the environmental vector interacts with the gene through epigenetic operators such as gene silencing, it can independently affect the mutation tendency of each bit in the solution vector (Stolfi and Alba, 2018). Considering this characteristic, an MEEV generated by the PEG descent strategy is used to alleviate the uncertainty of evolution. As shown in Fig. 3, this strategy is completed in three steps. After the gradient is obtained, it is normalized and discretized into an exponential environmental vector (EEV) and finally synthesized into a multi-dimensional environment vector.

First, for an individual $\boldsymbol{I}$ in the m^{th}-generation population, it is assumed that the D-dimensional objective function $\boldsymbol{f}_1$ of its location is smooth. Then, $\boldsymbol{\tau}^{(i)}$ is the tangent line, which indicates the fastest descent direction, and the gradient $-\nabla \boldsymbol{f}_1$ calculated using Eq. (1) coincides with its projection. Degenerate gradients can be used instead when dealing with nonsmooth problems. The introduction of gradient information makes evolution more inclined and efficient, generating offsprings with high fitness and retaining dominant genes. This solves the first type of evolutionary uncertainty problem mentioned in Section 2.2.

$$-\nabla \boldsymbol{f}_1 = \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{f}_1}{\partial (x_1, x_2, ..., x_D)}. \tag{1}$$

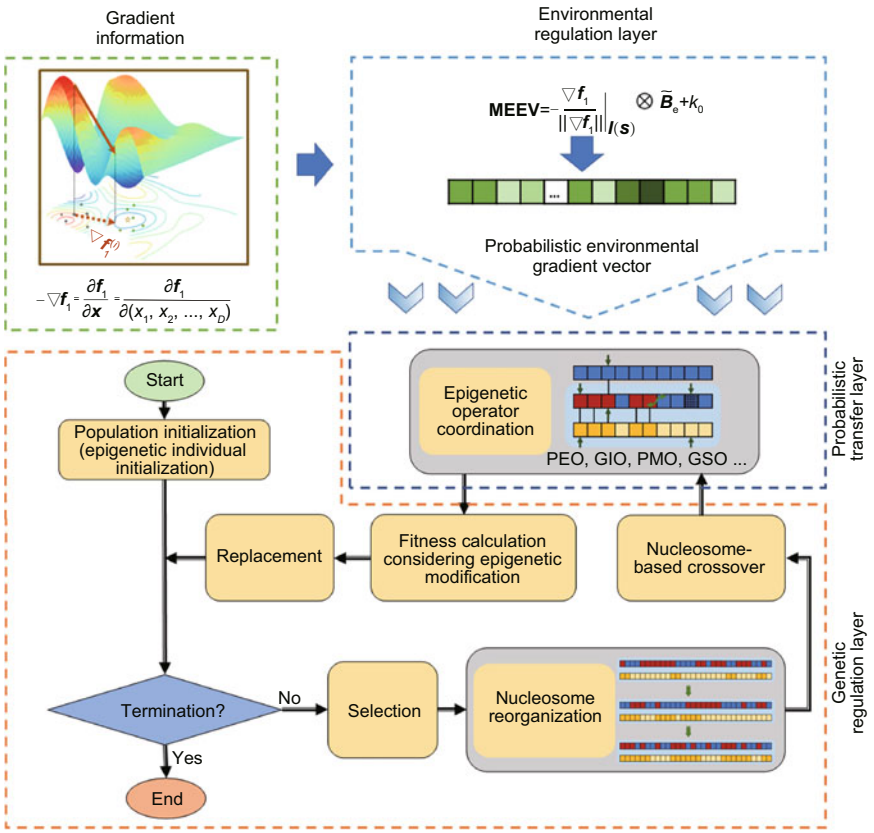

Fig. 2 The proposed probabilistic environmental gradient-driven epigenetic algorithm (PEGA) framework

Subsequently, an EEV modified by gradients is established as shown in Fig. 3a. We construct a positive n-dimensional basis vector $\tilde{\boldsymbol{B}}_{\mathrm{e}}$ in Eq. (2), in which $\eta > 1, 0 < \kappa_0 \leq 0.5, i = 0, 1, \cdots, D-1$. Here, η is a constant related to the encoding method, and κ_0 represents the boundary coefficient, which generally limits the strength of influence of the environment. It is originally set that $\kappa_0 = 0.5$ and $\eta = 2$ for using binary encoding in our later experiments. Let $\boldsymbol{A}$ be a matrix and b be a scalar. The operation $\boldsymbol{A}+b$ results in a new matrix where each element of $\boldsymbol{A}$ is incremented by b. Then, **EEV** can be obtained by modifying the exponential basis vector $\tilde{\boldsymbol{B}}_{\mathrm{e}}$ with the normalized gradient at the position $\boldsymbol{s}$ of individual $\boldsymbol{I}$ and κ_0, as shown in Eq. (3). The second type of evolutionary uncertainty is alleviated, which can be seen by calculating the expectation of the increment of each bit before and after mutation. EEV exerts an environmental pressure that is opposite to the exponential increasing trend of weight, balancing the impact of each bit's mutation in the solution vector as shown in Fig. 3b.

$$\tilde{\boldsymbol{B}}_{\mathrm{e}} = (B_i)_{1\times n},\ B_i = \frac{\kappa_0 \eta^{n-1}}{\eta^{n-1}-1}(1-\eta^i), \tag{2}$$

$$\mathbf{EEV}^{(i)} = -\left(\frac{\nabla \boldsymbol{f}_1}{\|\nabla \boldsymbol{f}_1\|}\bigg|_{\boldsymbol{I}(\boldsymbol{s})}\right)_i \tilde{\boldsymbol{B}}_{\mathrm{e}} + \kappa_0. \tag{3}$$

Finally, the aggregation of the MEEV is performed. For D-dimensional functions, each component x_i of $\boldsymbol{x}$ has an independent environmental vector, namely $\mathbf{EEV}^{(i)}$. These independent vectors can be combined to obtain multi-dimensional **EEV**. **MEEV** takes the form shown in Fig. 3c and Eq. (4). In summary, Eq. (5) derives a simpler calculation method for **MEEV**.

$$\mathbf{MEEV} = \mathbf{EEV}^{(0)} \oplus \mathbf{EEV}^{(1)} \oplus \cdots \oplus \mathbf{EEV}^{(D-1)}, \tag{4}$$

$$\mathbf{MEEV} = -\frac{\nabla \boldsymbol{f}_1}{\|\nabla \boldsymbol{f}_1\|}\bigg|_{\boldsymbol{I}(\boldsymbol{s})} \otimes \tilde{\boldsymbol{B}}_{\mathrm{e}} + \kappa_0, \tag{5}$$

where symbols "$\oplus$" and "$\otimes$" denote the element-wise addition and production, respectively.

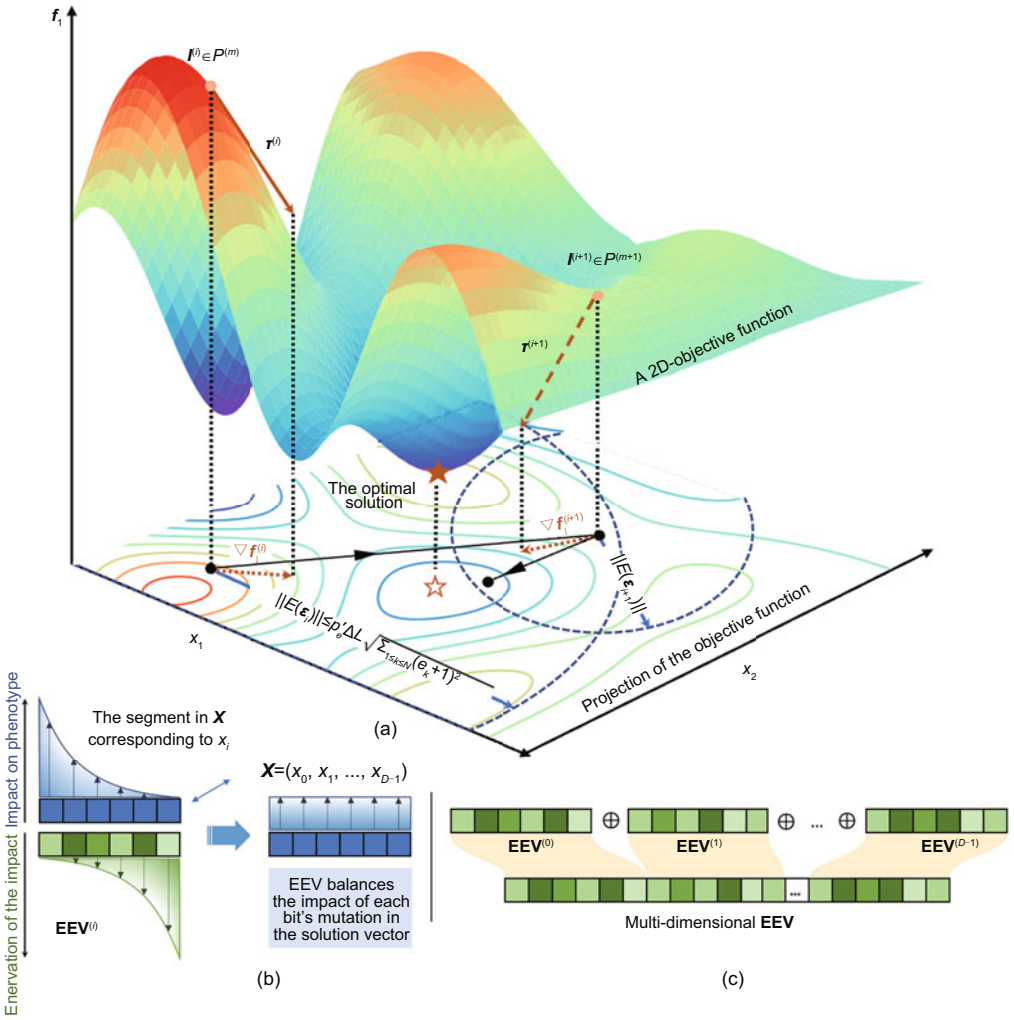

Fig. 3 Three steps for establishing a multi-dimensional environmental vector of probabilistic environmental gradient (PEG) to regulate evolution: (a) compution of gradients for searches based on PEG; (b) establishment of an exponential environmental vector (EEV) with the guidance of PEG; (c) aggregation of the multi-dimensional exponential environmental vector (MEEV) ($P^{(m)}$ denotes the population at the m^{th} generation)

3.2 An improved epigenetic individual based iteration process

Owing to the weak scalability of existing individual structures, newly added epigenetic operators are limited. To support rich epigenetic mechanisms, we design a new epigenetic structure recording more complicated epigenetic information, as shown in Fig. 4a. Each individual contains four vectors that record genetic information, epigenetic modifications, and nucleosomes. Genetic information refers to non-epigenetic information, including the solution vector $\boldsymbol{X}^n = (s^{(0)}, s^{(1)}, ..., s^{(n-1)})^{\mathrm{H}}$, which is an encoded gene list that contains the individual genotype. In addition, the sex-linked gene information of its parents $\boldsymbol{H}_{\sigma^{\prime},\varphi}^{n\times 2}$ can be divided into two parent vectors, each containing the historical paternal maternal gene information. The epigenetic modification matrix $\boldsymbol{M}_{\mathrm{g,p}}^{n\times 2}$ records the epigenetic modifications of each bit in the solution vector. $\boldsymbol{M}_{\mathrm{g}}^{n}$ represents epigenetic methylation and is closely related to epigenetic operators, such as gene silencing, while $\boldsymbol{M}_{\mathrm{p}}^{n}$ represents the molecular bookmark that regulates forced transcription from the parental genes. As shown in Fig. 4c, the nucleosome matrix $\boldsymbol{V}^{n\times n} = \mathrm{diag}\left(v^{(0)}, v^{(1)}, ..., v^{(n-1)}\right)$ consists of binary codes, wherein 1 and 0 correspond to different states of the nucleosome bits. Based on epigenetic individuals, the genetic information of the population can be generated using Algorithm 1.

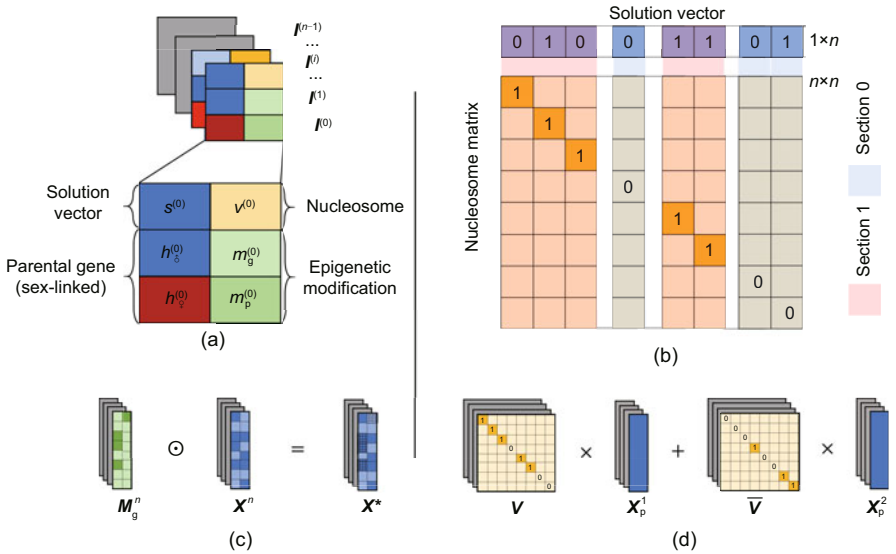

Fig. 4 Basic data structure and operation relationship in the proposed epigenetic algorithm: (a) epigenetic individual structure; (b) nucleosome matrix based solution vector segmentation; (c) epigenetic modifications; (d) nucleosome-based crossover

Most existing evolutionary algorithms ignore the inherent systemic nature of epigenetic operator regulation processes. During iteration, a new method of evaluation is devised, in which both genetic information and epigenetic modifications are systemically integrated by the Hadamard product between $\boldsymbol{M}_{\mathrm{g}}^{n}$ and $\boldsymbol{X}^{n}$, as shown in Eq. (6):

$$(m_{\mathrm{g}}^{(i)})^{\mathrm{H}} \odot (s^{(i)})^{\mathrm{H}} = (m_{\mathrm{g}}^{(i)} s^{(i)})^{\mathrm{H}}. \tag{6}$$

The evaluation consists of three steps. First, the modified genes $\boldsymbol{X}^{*} = \boldsymbol{M}_{\mathrm{g}}^{n} \odot \boldsymbol{X}^{n}$ are generated as shown in Fig. 4c. Second, for a D-dimensional optimization problem, namely, $\min f(\boldsymbol{x})$, $\boldsymbol{x} = (x_0, x_1, \ldots, x_{D-1})$, the mapping relationship between the epigenetic individual's genotype $\boldsymbol{X}$ and phenotype $\boldsymbol{x}$ is established, as shown in Eq. (7):

$$\mathscr{F} : \boldsymbol{x} \to \boldsymbol{X},\ \mathscr{F}^{-1} : \boldsymbol{X} \to \boldsymbol{x}. \tag{7}$$

Finally, the modified genes $\boldsymbol{X}^{*}$ are used to obtain the individual fitness $\mathcal{F}$, which is calculated by Eq. (8), depicting the maximum and minimum optimizations. Shift is a factor added in combination with the actual situation to prevent positive and negative changes in the objective function during calculation, and it does not affect the optimization results.

$$\mathcal{F} = [f(\mathscr{F}^{-1}(\boldsymbol{M}_{\mathrm{g}}^{n} \odot \boldsymbol{X}^{n})) + \mathrm{shift}]^{-1}. \tag{8}$$

3.3 Reproduction considering VNR

Nucleosome-based crossover exceeds traditional crossover process by using a nucleosome matrix. It originally refers to a gene exchange process bounded probabilistically by the nucleosome (Mayanagi et al., 2019). This mechanism occurs between two individual parents. First, we extract both the solution vectors and nucleosome vectors of two randomly selected individuals denoted $\boldsymbol{I}_{♂}$ as the male parent and $\boldsymbol{I}_{♀}$ as the female parent. Thereafter, the intermediate nucleosome vector is generated during the offspring generation process, as shown in Eq. (9), where $\boldsymbol{V}_{\mathrm{t}}$ represents the fused nucleosome vector, $\boldsymbol{V}_{♂}$ represents the male vector, and $\boldsymbol{V}_{♀}$ represents the female vector. Finally, as shown in Fig. 4d, two new solution vectors are calculated using Eq. (10) and then reinjected into two offspring individuals, where $\boldsymbol{U}$ is the identity matrix, and $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ refer to the two individuals selected during the VNR process.

$$\boldsymbol{V}_{\mathrm{t}} = \boldsymbol{V}_{♂} \boldsymbol{V}_{♀}, \tag{9}$$

$$\begin{cases} \boldsymbol{X}_{\mathrm{child},1} = \boldsymbol{V}_{\mathrm{t}} \boldsymbol{X}_1 + (\boldsymbol{U} - \boldsymbol{V}_{\mathrm{t}}) \boldsymbol{X}_2, \\ \boldsymbol{X}_{\mathrm{child},2} = (\boldsymbol{U} - \boldsymbol{V}_{\mathrm{t}}) \boldsymbol{X}_1 + \boldsymbol{V}_{\mathrm{t}} \boldsymbol{X}_2. \end{cases} \tag{10}$$

We design VNR as a semiadaptive nucleosome generation operator that dynamically adjusts the nucleosome radius according to the number of iterations. Instead of a constant nucleosome, it flexibly

constrains the population evolution, improves the efficiency in the early stage, and increases the accuracy in the later stage. VNR adjusts the radius according to Eq. (11), based on the fundamentals of nucleosome generation as the number of iterations increases, where $G_{\max}$ is the maximum number of iterations, $\mathcal{G}$ is the current iteration number, r is the shape parameter, and R is the initial nucleosome radius. When the radius of nucleosomes increases, the length of the collapsed nucleosomes decreases, limiting the exploration ability, and vice versa. A smaller radius is set in the early stage of evolution and gradually it increases over time, and the VNR balances the exploitation and exploration capabilities.

$$R(\mathcal{G}) = \left\lfloor \sqrt[r]{\frac{0.5G_{\max} - R}{G_{\max}^2 - 1}(\mathcal{G}^2 - 1) + R} \right\rfloor^r . \quad (11)$$

3.4 Other applied epigenetic operators

There are many phenomena in epigenetics that can be used to set different mutation operators. This study categorizes mutation operators into the following two types: position-sensitive epigenetic operators (PSEOs) and order-sensitive epigenetic operators (OSEOs). PSEOs do not directly affect the order of genes in the cell, but they do change the phenotype by slightly influencing gene transcription, and the mutations they cause are isolated. In contrast, mutations in OSEOs often occur continuously, causing dislocation and reversal of gene sequences. These operators are suitable for solving combinatorial optimization problems, such as the traveling salesman problem (TSP). Through the proper selection of operators based on many experiments, a general combination of epigenetic operators is found, and the operators included are shown in Fig. 5 (see the supplementary materials for the pseudocode of the operators).

3.4.1 Position effect

PEO is an OSEO established by the gene position effect. The gene position effect consists of gene expression changes in relation to shifts in gene location, often caused by translocation (Chen HC et al., 2017). The gene position effect greatly increases the randomness of variation by randomly changing the gene sequence, especially in solving combinatorial optimization problems. To realize PEO, first, the solution vector and nucleosome vector of the affected individual are extracted. The solution vector is split at section 0 or section 1 of the nucleosome vector according to the rule described in Fig. 4b.

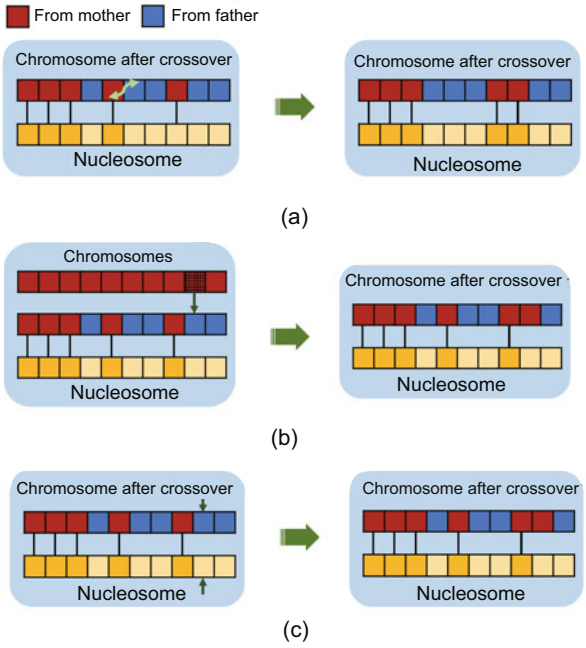

Fig. 5 Three newly introduced different epigenetic operators, i.e., an order-sensitive epigenetic operator (OSEO) and two position-sensitive epigenetic operators (PSEOs): (a) position effect; (b) gene imprinting; (c) paramutation

Subsequently, the different segments of the solution vector are randomly exchanged. In particular, when the objective function is multi-variate, the exchange of gene fragments may span the coding interval of different variables. Finally, we inject the new solution vector back to the individual.

3.4.2 Gene imprinting

GIO is a PSEO that refers to an epigenetic modification marker with a gene bookmark on the parental DNA strand. When this marker is detected, the gene is expressed in the parent-specific source, regardless of where it originally came from. An epigenetic marker is generated as an individual is created. If this gene is predominant in the population, the expression of its allele is always covered. If the location of the gene is critical, to avoid failure to converge, we choose to periodically clean up these epigenetic markers and regenerate them to replace the original gene sequence.

3.4.3 Paramutation

PMO refers to the transformation of the source of the expressed gene, meaning the gene that controls the expression of the site is transferred from the male to the female parent, and vice versa. This operator uses the genetic information of the parents recorded in $\boldsymbol{H}^{n\times 2}$, as shown in Fig. 4a. In particular, PMO has a lower priority than GIO, and the effect of PMO can be greatly reduced when the parental gene lists have long overlaping segments.

3.5 The proposed PEGA

Algorithm 2 shows the procedure used by the proposed PEGA. The main parameters involved include the population size, length of the solution vector, nucleosome generation probability, nucleosome radius, and probability of the environment exerting an impact. The evolution process starts after the population initialization operator generates epigenetic individuals and their nucleosome vectors. First, we obtain the dominant individuals in the population through tournament selection. Then, the nucleosomes are reorganized by VNR for these individuals. Subsequently, nucleosome-based crossover is performed to exchange long segments of genes between the selected individuals. Thereafter, each epigenetic operator is used for the mutation of each individual under the influence of PEG. While the mutation finishes, the individuals after elite replacement are preserved to maintain a constant population size. Finally, if the termination condition is not reached, the procedure returns to line 2 to repeat the iteration. After multiple iterations, the optimal solution can be obtained after decoding the best individual of the population.

Algorithm 2 The proposed PEGA

Input: Population size, N; gene length, n; possibility of nucleosome generation, P_{n}; nucleosome radius, R; environmental influence possibility, P_{e};
Output: A representative population with the individual with the best fitness.
1: $P \leftarrow$Population initialization(N, n);
// Population initialization
2: Nucleosome generator(P, P_{n}, R);
// Initialization of the variable nucleosome
3: Env $\leftarrow$Environment generator(n);
// Initialization of the environmental vector
4: **while** not termination **do**
5: $Q \leftarrow$Tournament selection(P);
// Use tournament selection; other kinds of selection
// can also be applied
6: $Q \leftarrow$Nucleosome reorganization(Q);
// Reorganization of the nucleosome vector
7: $Q \leftarrow$Nucleosome based crossover(Q);
8: Epigenetic operators(Q, Env, P_{e});
9: Evaluation(Q);
// Use the fitness function to calculate the individual's
// fitness
10: $P \leftarrow$Replacement(P, Q);
// Sort all individuals by fitness, and then let those
// better individuals enter the next generation
11: **end while**
12: **return** P;

4 Experiments and analysis

To demonstrate the overall performance of the newly proposed PEGA, a set of 29 widely used test suite functions were selected from CEC' 17 (Yue et al., 2017). First, the definitions of accuracy and stability were clarified, and the mathematical functions were used to demonstrate the performance improvement of the environmental gradient vector induced in PEGA compared to standard epiGA with linear environmental and nucleosome vectors. Then, the relevant indicators were calculated to analyze the characteristics of PEGA and to complete the comparison with other 10 state-of-the-art optimizers under 10, 30, 50, and 100 dimensions. PEGA shows comprehensive competitiveness based on the CEC' 17 criteria.

In the experiments, all algorithms were coded by Python 3.8 and executed on an Intel Core i7-7700 with four 3.6-GHz CPUs and 16-GB RAM.

4.1 Benchmark functions and compared algorithms

The benchmark functions, presented in Yue et al. (2017), are bound-constrained numerical optimization problems widely used in algorithm comparisons. All the functions are included in D-dimensional minimization problems formed from shifted and rotated basic functions and their combinations, defined as $\min f(\boldsymbol{x})$, $\boldsymbol{x}=(x_0, x_1, \cdots, x_{D-1})$, $x_i \in [-100, 100]$, where $f : \mathbb{R}^D \to \mathbb{R}$. These functions cover four main benchmark landscapes, specifically, unimodal, multimodal, hybrid, and composition functions, to evaluate both the algorithm performance and global ranking in the comparison between the proposed method and 10 state-of-the-art and most widely used optimization algorithms. The unimodal functions, F1 and F2, have only one global optimum, and thus they were used to assess

the exploitation capabilities of the competitive algorithms. The multimodal functions, F3–F9, have many local optima and can therefore be applied to evaluate the global search capability of the competitive algorithms. In real-world optimization problems, the hybrid functions (F10–F19) and composition functions (F20–F29) have different subcomponents consisting of basic functions with different properties.

To investigate the proposed algorithm, PEGA was compared with 10 state-of-the-art and most widely used optimization algorithms, specifically, GA (Mirjalili et al., 2020), social spider algorithm (SSA) (Nguyen, 2019), gaining–sharing knowledge-based algorithm (GSKA) (Mohamed et al., 2020), gray wolf optimizer (GWO) (Li et al., 2021), particle swarm optimization (PSO) (Chen HX et al., 2020; Khalid et al., 2021), artificial bee colony (ABC) (Więckowski et al., 2020), social spider optimization (SSO) (Baş and Ülker, 2020), moth search algorithm (MSA) (Feng et al., 2019), ant lion optimizer (ALO) (Abualigah et al., 2021), and atom search optimization (ASO) (Sun et al., 2021). The parameter configuration is listed in Table 1, and the other parameters of these algorithms are all based on their corresponding references.

The experimental design and comparison criteria are based on the standards established by CEC' 17. Each test was independently repeated 30 times for each problem per dimension, and all tests terminated before the number of evaluations reached the maximum number of function evaluations (MaxFE) required or the error value was lower than 10^{-13} (treated as zero). Factors considered in the comparison included the total error of the algorithm solving different problems and the error value based rank.

Table 1 Compared algorithms and parameter settings

Algorithm	Parameter	Value	Algorithm	Parameter	Value
PEGA	P_e	0.6	PSO	$c_1 = c_2$	1.2
	P_n	0.4		w	0.73
	R	0.02	ABC	SN	50
GA	p_m	0.025		Limit	100
	p_c	0.65	SSO	fp_{min}	0.65
SSA	r_a	1		fp_{max}	0.9
	p_m	0.1	MSA	n_{best}	5
	p_c	0.7		Partition	0.5
GSKA	p	0.1		Max step	1.0
	k_f	0.5	ALO	–	–
	k_r	0.9	ASO	α	50
	K	10		β	0.2
GWO	–	–			

The symbol "–" indicates the absence of algorithm parameters

4.2 Performance analysis of the proposed strategies and operators

4.2.1 Indicators of accuracy and stability

The stochastic operators of the evolutionary algorithm introduce the uncertainty of its convergence behavior. In this paper, two indicators, accuracy and stability, are used to reflect the ability of the algorithm to overcome evolutionary uncertainty. Accuracy is the sum of the average values of the convergence results for the problem in every test dimension, $\mathcal{D} = \{10, 30, 50, 100\}$, which reflects the algorithm's ability to overcome uncertainty and continuously exploit the optimal individual to make it close to the optimal solution. Stability is indicated by the standard deviation of the convergence results, which reflects the algorithm's ability to handle uncertain disturbances effectively. Accuracy and stability are measured using Eq. (12), where $x_{\text{best},i}$ represents the optimal solution of the i^{th} run.

$$\begin{cases} \text{accuracy}(f) = \sum\limits_{D\in\mathcal{D}} \dfrac{1}{N}\sum\limits_{i=1}^{N} f_D\left(x_{\text{best},i}\right), \\ \text{stability}(f) = \sum\limits_{D\in\mathcal{D}} \sqrt{\dfrac{1}{N}\sum\limits_{i=1}^{N}\left[f_D\left(x_{\text{best},i}\right) - \bar{f}_D\left(x_{\text{best}}\right)\right]^2}, \end{cases} \tag{12}$$

where $\bar{f}_D$ represents the average value of the best solutions obtained over N runs.

4.2.2 Ablation analysis of PEG and VNR

PEGA uses PEG along with VNR operators to accurately push the evolution toward an efficient convergence at the global optimum. Here, we discuss the experiments used to determine the PEG, VNR, and their combined effects on the accuracy and stability compared to linear environmental and nucleosome vectors. The experiments on the test problems were repeated under the condition provided in Section 4.1. The population size N was set to 500, n was set to $15 \times D$, the possibility of nucleosome generation P_n was set to 0.4, the nucleosome radius R was set to 25, and the probability of environmental influence was set to 0.15.

The accuracy and stability are listed in Tables S1 and S2 in the supplementary materials, respectively. If the PEGA exhibits a significant

improvement (>1%) in the accuracy or stability, the result is marked (−); if it is weakened, the result is marked (+); if there is no obvious effect, the result is marked (≈). The fusion strategy of PEG and VNR is effective in terms of accuracy and stability in 28 out of 29 test functions. Experimental results demonstrate that in the CEC' 17 benchmark, PEG is a highly generalizable means for boosting algorithm performance. Using PEG alone can boost the performance for 17 problems in terms of accuracy and 9 problems in terms of stability. Applying VNR alone can enhance the accuracy of solving 14 problems and the stability of 10 problems. Therefore, the fusion of these two strategies is far superior to using either alone. When acting together, the two operators yield complementary advantages. PEG and VNR together are effective on 96.55% of the test functions and can improve those indicators. Both the accuracy and stability are improved >90% in 11 functions by up to four orders of magnitude.

4.3 Analysis of the convergence behavior

We analyzed the convergence process of PEGA on some test functions by applying the following three aspects: average nucleosome radius, search route, and rate of evolution.

The experiments were repeated with two variables coded in 100 binary bits over 300 iterations, and Fig. 6a displays the objective functions. Fig. 6b indicates how the average nucleosome radius varies during evolution. In the optimization process, we maintained a low nucleosome radius in the early stage and increased the radius as the population evolved. This strategy balances the exploitation and exploration capabilities as well as the accuracy of PEGA.

The search history of the PEGA is illustrated in Fig. 6c. The arrow indicates how the best individual in the population moved during the continuous iteration process. For multimodal properties with relatively gentle gradient changes near the optimal solution, such as F6 and F25, the proposed algorithm can directly find the peak where the optima are located, owing to PEGA's powerful global search ability. For functions with more drastic gradient changes, such as F5 and F20, PEGA explores promising areas of the search space and accurately exploits the global optima to achieve rapid convergence.

Fig. 6d shows the rate of evolution per generation. The rate of evolution can be calculated using Eq. (13) to evaluate the robustness of the evolutionary population to random mutations, where N_{pre} represents the number of preserved individuals in the next generation and N_{c} represents the number of individuals in the current population. This ability ensures that a considerable proportion of the dominant individuals will not be eliminated or will not deviate from the optimal evolutionary direction due to untimely mutation, promoting the overall evolution of the population. As shown in the figure, the results of PEGA when dealing with different classes of problems are similar. In the early stage of evolution, epiGAs' global random search ability prevailed, and large-scale individual changes occurred; then, owing to the dynamic constraints of VNR, the population tended to be stable, which is conducive with high-precision search.

$$\text{Rate of evolution} = \frac{N_{\mathrm{pre}}}{N_{\mathrm{c}}}. \tag{13}$$

4.4 Performance evaluation of PEGA

Evolutionary algorithms are essentially stochastic optimization algorithms. To eliminate randomness from the results, the average and standard deviation were calculated using the experimental results to evaluate the performance of each algorithm. In addition, a statistical test is needed to identify a statistically significant difference between the results of two methods. The results reported in Tables S3–S5 (see the supplementary materials for details) were obtained from over 30 runs for each algorithm with 10, 30, 50, and 100 dimensions, solving the 29 benchmark functions on CEC' 17. PEGA provides better results than the other experimental algorithms for most of the functions, especially in high-dimensional problems.

Test functions F1–F9 are primarily separable unimodal and simple multimodal functions. By comparing the results of high dimensions in Table S3 (see the supplementary materials for details), it can be observed that when the complexity of the same problem becomes lower, the accuracy of PEGA is significantly improved. Moreover, between unimodal and multimodal problems, PEGA is better at handling the latter, and its accuracy is generally orders of magnitude ahead. For these benchmark functions, PEGA does not exhibit superior performance

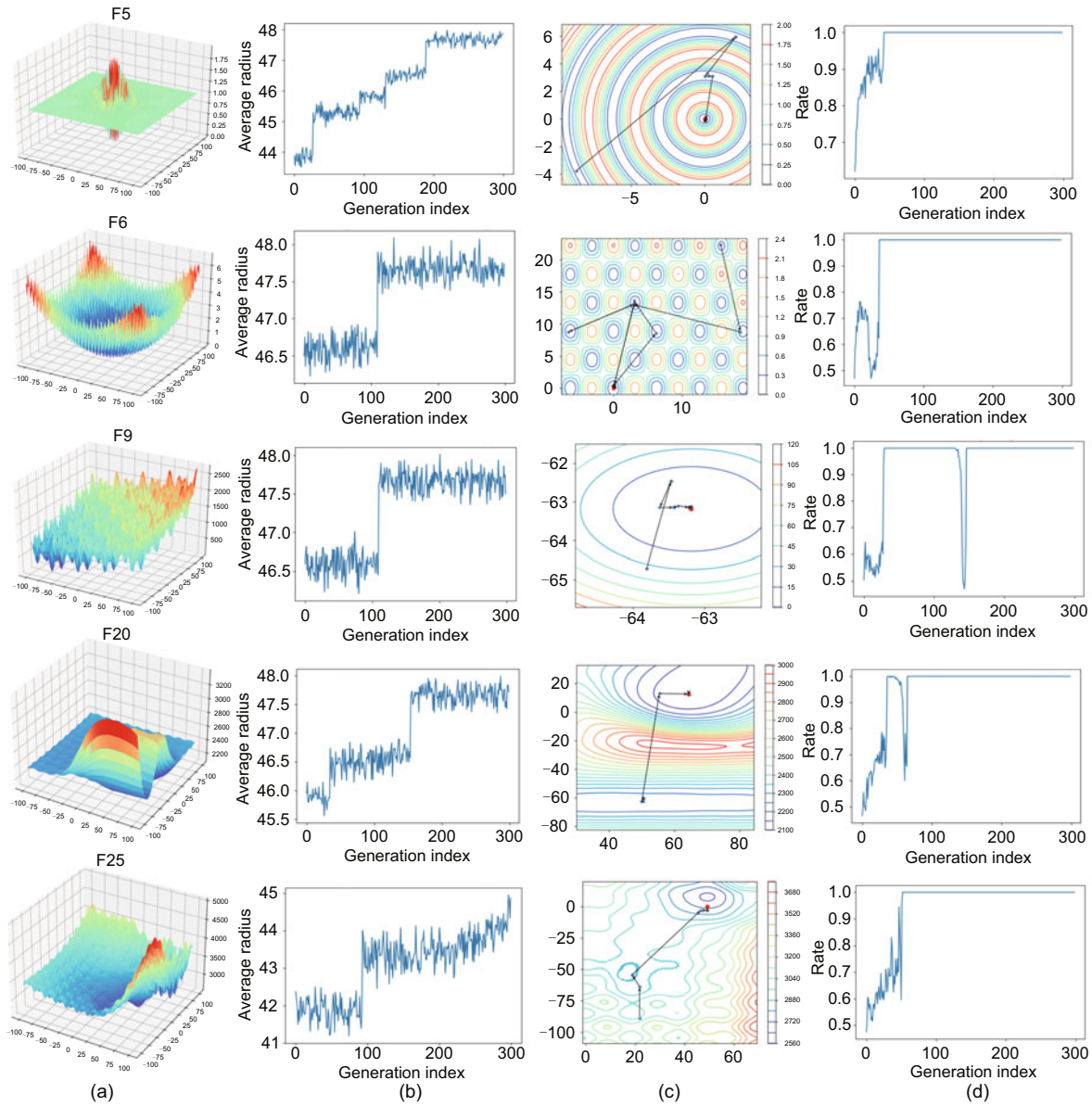

Fig. 6 Objective function (a), average nucleosome radius (b), search route (c), and rate of evolution (d)

at higher dimensions, but it still outperforms 60% of the contrasting methods in most cases. In these tests, in addition to GSKA and GWO, which have strong local search capability, PEGA still shows a huge performance advantage.

In the evaluation of hybrid problems, PEGA demonstrates superior accuracy compared to >70% of the comparison methods in most cases. Although the performance gap between GWO and GSKA is not pronounced, PEGA outperforms the other algorithms by a significant margin. Notably, PEGA exhibits exceptional proficiency in handling composition problems. This is evident from the analysis of Table S4 (see the supplementary materials for details); except for the case of 30 dimensions, the proposed algorithm consistently maintained a leading position in the benchmark function competition across the other dimensions, surpassing all the other algorithms by a substantial margin.

From the above preliminary analysis of the data, it can be seen that in the tests of four different benchmark functions, compared with the other 10 optimization algorithms, PEGA has different degrees of advantages. Among them, the test of composition

function shows the advantages of PEGA to the greatest extent. The success of PEGA in this field proves that under the combined action of the PEG strategy and the VNR operator, the global and local search abilities of the algorithm are well balanced and are significantly better than those of the other existing algorithms.

When independently calculating the weighted sum of error (SE) values (Yue et al., 2017) on all benchmark functions (F1–F29) of each dimension, as shown in Table 2 for instance, PEGA consistently outperforms most compared algorithms as the best results are underlined. Based on further analysis, it can be concluded that for composition functions, without considering the situation of $D = 30$, PEGA ranks first in most benchmark functions, especially in solving F22, F23, and F25. For unimodal, multimodal, and hybrid functions, the average errors obtained using the PEGA are close to those produced by GSKA and GWO, and are markedly less than those by the other eight metaheuristics, especially in solving F1, F4, F7, F17, and F18. For all the 30-dimensional problems, the proposed algorithm lags behind GSKA, GWO, and MSA in some aspects, but the gap is not significant. Moreover, for F7–F9 and F15–F18, the accuracy is significantly better than those of the other nine comparative algorithms, enabling PEGA to maintain its competitiveness. In summary, although GSKA, GWO, and MSA have taken the lead in some issues, their comprehensive performances in all dimensions are inferior to those of PEGA.

The performance metrics were established considering those metrics outlined by Yue et al. (2017). The algorithms were evaluated with a score composed of Score1 and Score2, which take both error values and ranks into consideration. Each score contributes half of the total score, with a maximum value of 100.

Through the comparison of sum of normalized error (SNE) values and sum of rank (SR) values on different benchmark functions shown in Tables S3–S5 (see the supplementary materials for details), it can be concluded that PEGA mostly ranks first, and the advantage of PEGA is more obvious with the increase of the dimension. The ranks of the proposed method are more stable than those of the other methods, especially at higher dimensions. Although GWO and GSKA have slight leads on some unimodal and multimodal benchmark functions, through a more detailed comparison, it can be found that although the accuracy ranking of PEGA for a single problem is not always elite, its huge advantage in composition functions renders PEGA the most accurate by combining the weights of all problems.

The lowest optimization errors for 100, 50, 30, and 10 dimensions are listed in Tables S3–S5 (see the supplementary materials for details). According to the highest ranks across all algorithms for each problem, the total score can then be calculated, as

Table 2 Results of comparison on 29 benchmark functions for PEGA and other algorithms (D=100)

Function	Mean absolute error										
	ALO	SSA	GSKA	GA	GWO	MSA	PSO	ASO	SSO	ABC	PEGA
F1	3.38E+06	3.58E+06	0	3.67E+09	0	1.78E+10	5.82E+09	2.08E+11	1.46E+08	1.97E+11	9.78E+06
F2	1.24E+05	1.00E+03	0	9.63E+03	1.68E+05	2.52E+05	9.45E+04	2.69E+05	1.53E+04	2.60E+05	2.28E+05
F3	1.03E+02	7.95E-01	0	7.29E+02	0	2.19E+03	3.35E+02	4.33E+04	3.33E+01	3.65E+04	1.90E+02
F4	1.70E+02	7.67E+00	0	6.37E+02	0	3.39E+02	6.48E+02	1.41E+03	4.66E+01	9.26E+02	5.72E+01
F5	2.61E+01	4.65E-01	0	1.38E+01	0	0	2.51E+01	1.14E+02	3.35E+00	7.27E+01	3.73E+00
F6	1.09E+03	6.34E+01	0	1.01E+03	0	1.45E+03	1.27E+03	9.19E+03	6.71E+02	8.76E+03	1.55E+02
F7	1.64E+02	7.02E+00	0	6.38E+02	0	3.35E+02	6.82E+02	1.40E+03	4.39E+01	9.11E+02	6.11E+01
F8	4.09E+03	1.06E+01	6.14E+00	1.24E+03	3.79E+00	8.42E+03	4.01E+03	6.67E+04	6.25E+01	1.65E+04	6.13E+01
F9	1.28E+04	1.55E+02	1.09E-10	1.71E+04	1.09E-10	7.08E+03	2.59E+04	3.20E+04	9.94E+02	1.97E+04	1.34E+02
F10	5.87E+03	1.36E+01	1.09E-10	1.69E+03	1.06E+04	3.91E+04	7.43E+03	6.09E+04	1.18E+03	5.80E+04	2.06E+04
F11	6.27E+07	6.89E+05	0	7.32E+08	0	8.77E+08	8.57E+08	6.51E+10	3.14E+07	5.58E+10	1.68E+05
F12	6.35E+03	1.36E+05	0	2.92E+08	0	8.79E+08	2.17E+08	3.48E+10	1.75E+07	3.02E+10	3.83E+02
F13	1.18E+06	7.42E+01	0	1.50E+06	0	2.01E+03	1.54E+06	1.33E+08	1.19E+05	8.33E+07	3.23E+01
F14	5.98E+03	4.69E+03	0	1.05E+08	9.88E-03	2.24E+05	5.13E+07	1.57E+10	6.73E+06	1.25E+10	1.43E+02
F15	5.48E+03	3.01E+01	4.54E-13	3.86E+03	7.45E-02	1.20E+03	5.19E+03	1.51E+04	2.56E+02	9.97E+03	5.75E+01
F16	3.30E+03	1.85E+01	1.13E-12	2.43E+03	1.72E-03	8.55E+02	3.28E+03	3.43E+04	2.49E+02	1.92E+04	4.77E+01
F17	2.08E+06	1.05E+02	2.95E-12	2.65E+06	0	9.79E+02	4.45E+06	2.39E+08	1.75E+05	1.60E+08	4.62E+01
F18	7.98E+04	4.74E+03	6.82E-13	1.08E+08	0	9.55E+05	3.57E+07	1.53E+10	7.02E+06	1.25E+10	2.79E+01
F19	2.59E+03	2.34E+01	6.82E-13	1.35E+03	5.30E-02	3.53E+02	2.78E+03	4.86E+03	1.27E+02	2.79E+03	3.41E+01
F20	7.28E+02	1.47E+03	1.19E+03	1.55E+03	1.42E+03	5.60E+02	1.45E+03	2.00E+03	1.36E+03	1.55E+03	4.62E+02
F21	1.49E+04	1.36E+04	2.08E+04	2.46E+04	7.28E+03	7.25E+03	2.86E+04	3.19E+04	1.32E+04	1.99E+04	8.13E+03
F22	1.46E+03	2.41E+03	2.17E+03	4.27E+03	2.75E+03	1.01E+03	2.68E+03	3.93E+03	2.43E+03	3.41E+03	7.82E+02
F23	1.65E+03	2.62E+03	2.31E+03	4.65E+03	3.18E+03	1.21E+03	3.03E+03	5.26E+03	2.96E+03	5.03E+03	9.58E+02
F24	1.22E+03	1.08E+04	1.24E+03	1.31E+03	5.20E+03	5.33E+03	4.72E+03	4.23E+04	8.74E+03	3.83E+04	1.34E+03
F25	1.18E+04	1.93E+04	1.40E+04	2.18E+04	1.66E+04	6.00E+03	1.55E+04	3.23E+04	2.00E+04	2.93E+04	3.96E+03
F26	1.88E+03	2.83E+03	5.00E+02	6.74E+03	5.00E+02	7.51E+02	3.20E+03	6.33E+03	2.57E+03	5.70E+03	6.68E+02
F27	1.26E+03	1.13E+04	5.00E+02	1.19E+04	5.00E+02	1.95E+03	8.27E+03	6.62E+04	9.63E+03	4.70E+04	2.05E+03
F28	7.59E+03	1.10E+04	9.09E+03	4.45E+04	1.01E+04	1.95E+03	8.27E+03	6.62E+04	9.63E+03	4.70E+04	2.05E+03
F29	1.67E+08	2.46E+09	4.88E+08	1.10E+10	1.46E+09	1.78E+07	2.77E+09	2.05E+10	1.26E+09	1.79E+10	1.33E+08
SNE_{100}	2.36E+08	2.47E+09	4.88E+08	1.59E+10	1.46E+09	1.87E+10	9.77E+09	3.60E+11	1.47E+09	3.26E+11	1.43E+08

Best results are underlined. SNE_{100} represents the total sum of normalized error (SNE) values for the benchmark functions in 100 dimensions

shown in Table 3. Based on the CEC' 17 scoring standard, PEGA has the best overall performance among all the compared algorithms on unimodal, multimodal, hybrid, and composition problems.

4.5 Performance evaluation on engineering application optimization problems

The optimization of spacecraft trajectories is a challenging problem faced in astronautical engineering. It involves finding the most optimal trajectory for an interplanetary probe, to travel from Earth to another planet or asteroid equipped with a chemical propulsion engine. The mathematical formulation of this problem was presented by Das and Suganthan (2010). The proposed algorithm has been tested on two instantiations issued by the European Space Agency: Messenger and Cassini-2. The parameters of competing metaheuristics and the proposed PEGA are not meticulously tuned to accommodate the selected real-world optimization problems. The parameter settings for other competing algorithms are completely kept the same as those listed in Table 1. The same population is used as the initial population to maintain fairness for all algorithms. Experimental results of the accuracy and standard deviation, as presented in Table 4, show that PEGA achieves the top rank in terms of accuracy in both instances after 70 000 function evaluations. Moreover, Figs. 7 and 8 demonstrate that PEGA has a higher convergence rate compared to other competitors on average.

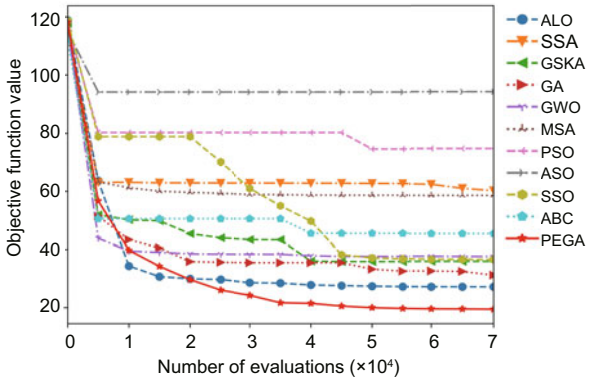

Fig. 7 Average convergence of all compared algorithms in optimizing the Messenger trajectory problem over 30 independent runs

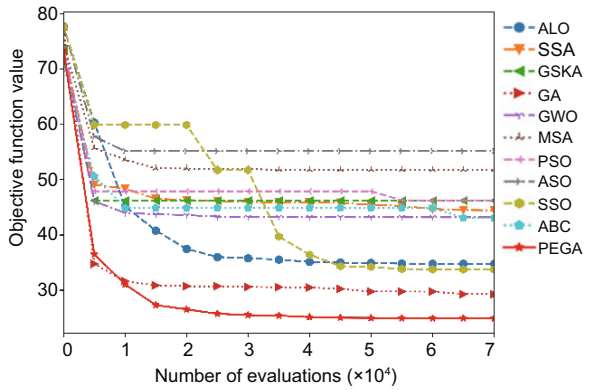

Fig. 8 Average convergence of all compared algorithms in optimizing the Cassini-2 trajectory problem over 30 independent runs

5 Conclusions

To promote the optimization accuracy and stability, a novel probabilistic environmental gradient-driven GA considering epigenetic traits called PEGA was proposed. To combat the problem of excessive disorder, from the perspective of searching scale, an MEEV operator can make the local suboptimal solution stably converge to the local optimal solution by adopting different mutation probabilities for each gene. Besides, from the perspective of searching direction, a PEG strategy was designed to speed up the local optimal convergence using a heterogeneous exponential probabilistic distribution. Furthermore, a VNR operator was used to ensure fast and stable convergence by balancing exploitation and exploration capabilities. Additionally, to overcome possible premature problem, multiple epigenetic operators were originally organized for generating more mutation patterns. Experiments on 29 CEC' 17 based

Table 3 Total scores

Metric	ALO	SSA	GSKA	GA	GWO	MSA	PSO	ASO	SSO	ABC	PEGA
SNE	1.08E+08	1.05E+09	2.29E+08	7.90E+09	6.06E+08	9.57E+09	4.93E+09	2.09E+11	6.56E+08	1.82E+11	6.1082E+07
Score1	28.09	2.89	13.35	0.39	5.04	0.32	0.62	0.01	4.65	0.02	50.00
SR	159.2	128.4	60.3	228	96.1	146.1	208.5	306.9	151.3	270.6	100
Score2	18.94	23.48	50.00	13.22	31.37	20.64	14.46	9.82	19.93	11.14	30.15
Total score	47.03	26.37	63.35	13.61	36.42	20.96	15.08	9.84	24.58	11.16	80.15
Rank	3	5	2	9	4	7	8	11	6	10	1

Table 4 Results of the Messenger and Cassini-2 trajectory optimization problem over 30 independent runs

Algorithm	Messenger			Cassini-2		
	Acc	Std	Rank	Acc	Std	Rank
ALO	36.698	4.594	8	38.257	9.103	9
SSA	32.308	2.031	7	36.787	2.587	8
GSKA	28.402	3.617	5	34.612	5.066	7
GA	25.103	6.914	3	24.183	7.025	2
GWO	30.310	5.641	6	27.065	7.873	3
MSA	20.231	1.748	2	32.574	9.912	4
PSO	46.162	2.729	10	50.006	3.171	10
ASO	49.995	3.484	11	53.136	4.744	11
SSO	26.805	1.137	4	34.336	1.593	5
ABC	38.161	1.858	9	34.466	2.037	6
PEGA	19.987	3.324	1	17.571	8.704	1

benchmark functions were conducted to evaluate the performance of PEGA. The capabilities of PEGA in terms of exploitation, exploration, and local optimal avoidance were investigated using unimodal, multimodal, hybrid, and composition problems. The experimental results indicated that the proposed method outperforms 10 state-of-the-art evolutionary algorithms in terms of accuracy and stability on comprehensive performance. Particularly, PEG+VNR was effective on 96.55% of the testing functions and can improve the accuracy by up to four orders of magnitude. PEGA was also successfully applied to solve two instantiations of spacecraft trajectory optimization. The performance of the PEGA algorithm on these real-world engineering optimization problems was the best in terms of solution accuracy.

In the future, PEGA should be further developed and applied to more practical large-scale optimization problems. Finally, this algorithm needs to be continuously compared with the most advanced algorithms to further improve its performance.

Contributors

Zhiyu DUAN and Shunkun YANG designed the research and drafted the paper. Qi SHAO and Minghao YANG helped organize the paper. Zhiyu DUAN and Qi SHAO revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abualigah L, Shehab M, Alshinwan M, et al., 2021. Ant Lion Optimizer: a comprehensive survey of its variants and applications. *Arch Comput Methods Eng*, 28(3):1397-1416. https://doi.org/10.1007/s11831-020-09420-6

Baş E, Ülker E, 2020. A binary social spider algorithm for continuous optimization task. *Soft Comput*, 24(17):12953-12979. https://doi.org/10.1007/s00500-020-04718-w

Birogul S, 2016. Epigenetic algorithm for optimization: application to mobile network frequency planning. *Arab J Sci Eng*, 41(3):883-896. https://doi.org/10.1007/s13369-015-1869-5

Chen HC, Martinez JP, Zorita E, et al., 2017. Position effects influence HIV latency reversal. *Nat Struct Mol Biol*, 24(1):47-54. https://doi.org/10.1038/nsmb.3328

Chen HX, Fan DL, Fang L, et al., 2020. Particle swarm optimization algorithm with mutation operator for particle filter noise reduction in mechanical fault diagnosis. *Int J Patt Recog Artif Intell*, 34(10):2058012. https://doi.org/10.1142/S0218001420580124

Chromiński K, Boryczka M, 2016. Epigenetically inspired modification of genetic algorithm and his efficiency on biological sequence alignment. In: Czarnowski I, Caballero A, Howlett R, et al. (Eds.), Intelligent Decision Technologies 2016. Springer, Cham, p.95-105. https://doi.org/10.1007/978-3-319-39627-9_9

Coli GM, Boattini E, Filion L, et al., 2022. Inverse design of soft materials via a deep learning-based evolutionary strategy. *Sci Adv*, 8(3):eabj6731. https://doi.org/10.1126/sciadv.abj6731

Das S, Suganthan PN, 2010. Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Technical Report, Jadavpur University, India, and Nanyang Technological University, Singapore, p.341-359.

Ezzarii M, El Ghazi H, El Ghazi H, et al., 2020. Epigenetic algorithm-based detection technique for network attacks. *IEEE Access*, 8:199482-199491. https://doi.org/10.1109/ACCESS.2020.3035250

Feng YH, Yi JH, Wang GG, 2019. Enhanced moth search algorithm for the set-union knapsack problems. *IEEE Access*, 7:173774-173785. https://doi.org/10.1109/ACCESS.2019.2956839

Gouil Q, Baulcombe DC, 2018. Paramutation-like features of multiple natural epialleles in tomato. *BMC Genomics*, 19(1):203. https://doi.org/10.1186/s12864-018-4590-4

Katoch S, Chauhan SS, Kumar V, 2021. A review on genetic algorithm: past, present, and future. *Multim Tools Appl*, 80(5):8091-8126. https://doi.org/10.1007/s11042-020-10139-6

Khalid QS, Azim S, Abas M, et al., 2021. Modified particle swarm algorithm for scheduling agricultural products. *Eng Sci Technol Int J*, 24(3):818-828. https://doi.org/10.1016/J.JESTCH.2020.12.019

Li Y, Lin XX, Liu JS, 2021. An improved gray wolf optimization algorithm to solve engineering problems. *Sustainability*, 13(6):3208. https://doi.org/10.3390/su13063208

Lin J, Zhu L, Gao KZ, 2020. A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Syst Appl*, 140:112915. https://doi.org/10.1016/j.eswa.2019.112915

Makino H, Feng XA, Kita E, 2020. Stochastic schemata exploiter-based optimization of convolutional neural network. IEEE Int Conf on Systems, Man, and Cybernetics, p.4365-4371. https://doi.org/10.1109/SMC42975.2020.9283473

Mayanagi K, Saikusa K, Miyazaki N, et al., 2019. Structural visualization of key steps in nucleosome reorganization by human fact. *Sci Rep*, 9(1):10183. https://doi.org/10.1038/s41598-019-46617-7

Miikkulainen R, Forrest S, 2021. A biological perspective on evolutionary computation. *Nat Mach Intell* , 3(1):9-15. https://doi.org/10.1038/s42256-020-00278-8

Mirjalili S, Dong JS, Sadiq AS, et al., 2020. Genetic Algorithm: Theory, Literature Review, and Application in Image Reconstruction. In: Mirjalili S, Dong JS, Lewis A, (Eds.), Nature-Inspired Optimizers. Studies in Computational Intelligence. Springer, Cham, p.69-85. https://doi.org/10.1007/978-3-030-12127-3_5

Mohamed AW, Hadi AA, Mohamed AK, 2020. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. *Int J Mach Learn Cybern*, 11(7):1501-1529. https://doi.org/10.1007/s13042-019-01053-x

Monroe JG, Srikant T, Carbonell-Bejerano P, et al., 2022. Mutation bias reflects natural selection in *Arabidopsis thaliana*. *Nature*, 602(7895):101-105. https://doi.org/10.1038/s41586-021-04269-6

Nguyen TT, 2019. A high performance social spider optimization algorithm for optimal power flow solution with single objective optimization. *Energy*, 171:218-240. https://doi.org/10.1016/j.energy.2019.01.021

Owens NDL, Gonzalez I, Artus J, et al., 2020. Mitotic bookmarking by transcription factors and the preservation of pluripotency. In: Meshorer E, Testa G (Eds.), Stem Cell Epigenetics. Academic Press, Amsterdam, the Netherlands, p.131-153. https://doi.org/10.1016/B978-0-12-814085-7.00006-4

Pereira AGC, Campos VSM, de Pinho ALS, et al., 2020. On the convergence rate of the elitist genetic algorithm based on mutation probability. *Commun Stat-Theory Methods*, 49(4):769-780. https://doi.org/10.1080/03610926.2018.1528361

Periyasamy S, Gray A, Kille P, 2008. The epigenetic algorithm. IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), p.3228-3236. https://doi.org/10.1109/CEC.2008.4631235

Slowik A, Kwasnicka H, 2020. Evolutionary algorithms and their applications to engineering problems. *Neur Comput Appl*, 32:12363-12379.

Song YJ, Cai X, Zhou XB, et al., 2023. Dynamic hybrid mechanism-based differential evolution algorithm and its application. *Expert Syst Appl*, 213:118834. https://doi.org/10.1016/j.eswa.2022.118834

Stolfi DH, Alba E, 2018. Epigenetic algorithms: a new way of building gas based on epigenetics. *Inform Sci*, 424:250-272. https://doi.org/10.1016/j.ins.2017.10.005

Sun P, Liu H, Zhang Y, et al., 2021. An intensify atom search optimization for engineering design problems. *Appl Math Model*, 89:837-859. https://doi.org/10.1016/j.apm.2020.07.052

Tanev I, Yuta K, 2003. Epigenetic programming: an approach of embedding epigenetic learning via modification of histones in genetic programming. The 2003 Congress on Evolutionary Computation, p.2580-2587. https://doi.org/10.1109/CEC.2003.1299413

Thamban T, Agarwaal V, Khosla S, 2020. Role of genomic imprinting in mammalian development. *J Biosci*, 45(1):20. https://doi.org/10.1007/s12038-019-9984-1

Więckowski J, Kizielewicz B, Kołodziejczyk J, 2020. Finding an approximate global optimum of characteristic objects preferences by using simulated annealing. Proc 12th KES Int Conf on Intelligent Decision Technologies, p.365-375. https://doi.org/10.1007/978-981-15-5925-9_31

Yue CT, Price KV, Suganthan PN, et al., 2017. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical Report, No. 201911 (2016).

List of supplementary materials

1 The pseudocode of three types of epigenetic operators: position effect operator (PEO), gene imprinting operator (GIO), and paramutation operator (PMO)
2 The ablation experimental results of the probabilistic environmental gradient (PEG) and variable nucleosome reorganization (VNR) strategies
3 The complete results of performance experiments of the probabilistic environmental gradient-driven epigenetic algorithm (PEGA) (D=100, 50, 30, and 10)
Table S1 PEG, VNR, and PEG+VNR strategy comparison on the accuracy
Table S2 PEG, VNR, and PEG+VNR strategy comparison on the stability
Table S3 Comparison results of two unimodal and seven multimodal benchmark functions for PEGA and other algorithms
Table S4 Comparison results of 10 hybrid functions for PEGA and other algorithms
Table S5 Comparison results of 10 composition benchmark functions for PEGA and other algorithms
Algorithm S1 Position effect
Algorithm S2 Gene imprinting
Algorithm S3 Paramutation