Grey Prediction Evolution Algorithm Based on Accelerated Even Grey Model

CONG GAO\textsuperscript{1}, ZHONGBO HU\textsuperscript{1}, ZENGGANG XIONG\textsuperscript{2}, AND QINGHUA SU\textsuperscript{1}\textsuperscript{1}\textsuperscript{2}\textsuperscript{1}

\textsuperscript{1}School of Information and Mathematics, Yangtze University, Jingzhou 434023, China
\textsuperscript{2}School of Computer and Information Sciences, Hubei Engineering University, Xiaogan 432000, China

Corresponding authors: Zhongbo Hu (hazbdl@126.com) and Zenggang Xiong (jkxxzg2003@163.com)

This work was supported in part by the State Key Laboratory of Biogeology and Environmental Geology, China University of Geosciences, under Grant GBL21801, in part by the National Nature Science Foundation of China under Grant 61972136, in part by the Education Bureau of Human Province of China under Grant 15B061, and in part by the Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation under Grant T201410.

\textbf{ABSTRACT} The grey prediction evolution algorithm based on the even grey model (GPEAe) is a pioneer of prediction-based evolutionary algorithms. Its offsprings are generated by a first-order inverse accumulating generation operation (1-IAGO) depending on every prediction of the even grey model. For the fact that the original values are already known for the first few predicted values in 1-IAGO, this paper firstly develops an accelerated 1-IAGO (1-AIAGO) which replaces a particular prediction with the corresponding original value. An accelerated even grey model (AEGM(1,1)) based on the 1-AIAGO is then proposed. Finally, this paper proposes a new grey prediction evolution algorithm (GPEAae) which uses the AEGM(1,1) as the reproduction operator of evolutionary algorithm to forecast the offspring. The performance of GPEAae is verified on the CEC2014 benchmark functions and the set containing nine engineering constrained design problems. The experimental results show that the GPEAae has superiority and highly competitive effectiveness when compared with the GPEAe and other state-of-the-art algorithms. Matlab Codes of this article can be found in https://github.com/Zhongbo-Hu/Prediction-Evolutionary-Algorithm-HOMEPAGE.

\textbf{INDEX TERMS} Accelerated even grey model, engineering constrained design problem, grey prediction evolution algorithm.

\section{I. INTRODUCTION}

The grey prediction evolution algorithm based on even grey model (GPEAe) [1], first proposed by Zhongbo Hu \textit{et al.} in 2019, is constructed by introducing the grey prediction theory [2]–[6] into the optimization field. The GPEAe is a population-based stochastic search method, which predicts the evolutionary direction at the macroscopic level to lead the population gradually move toward the global optimum. The main difference between GPEAe and other evolutionary algorithms (EAs) [7]–[11] is that GPEAe adopts the prediction theory to obtain the offsprings. Unlike traditional optimization techniques, GPEAe does not rely on gradient information and has the ability in jumping out of local optima. In addition, GPEAe has broad application prospects due to its advantages of simple structure, easy coding, few parameters, and satisfactory effect in solving complex numerical optimization problems.

The grey system theory, as a theoretical cornerstone of the grey prediction evolution algorithm (GPEA), has been widely applied in many different fields so far [12]–[15]. Some advanced grey models have been created in the past few years, i.e., the new multivariable grey model with structure compatibility [16], the nonlinear grey bernoulli model [17], the new-structure grey Verhulst model [18], etc. Since the GPEA was proposed, some relevant researches have been developed in the following three aspects. (I) Introduce new grey models to design new GPEA: a multivariable grey model [19] was used to design a new GPEA in [20]. (II) Improve the GPEA according to evolutionary approaches: a topological opposition-based learning strategy was used to improve the GPEA in [21]. (III) Apply to the GPEA to solve specific optimization problems: the GPEA is applied to solve environmental/economic dispatch problem in [22].
In the evolution process, the offspring of GPEAe is generated by a first-order inverse accumulating generation operation (1-IAGO). The 1-IAGO is used to calculate the final expression that depends on every prediction of the even grey model (EGM(1,1)). Concretely, the EGM(1,1) firstly converts the unordered original data to data sequence with specific law, and then calculates the predicted values of the current and next data sequence. Finally, the 1-IAGO operates on the two predicted values to generate the data of next trial population. The 1-IAGO is a recursive expression and it will directly affect the prediction effect of EGM(1,1). The 1-IAGO can help GPEAe to reach good optimization results, but there is still room for further improvement of its prediction effect.

When analyzing the two predicted values in the calculation expression of 1-IAGO, we find that the first predicted value has already been known. This paper firstly develops an accelerated 1-IAGO (1-AIAGO) which replaces the first predicted value with the corresponding original value. Based on the 1-AIAGO, a new grey model called the accelerated EGM(1,1) (AEGM(1,1)) is then proposed. The operation on original value helps the 1-AIAGO to make better use of existing information of data sequence. At the same time, the 1-AIAGO can reduce the calculation of the predicted value once, so that AEGM(1,1) has a smaller computational complexity compared with EGM(1,1).

On the basic of the AEGM(1,1), this paper proposes a new grey prediction evolution algorithm (GPEAae). The GPEAae considers the population sequence of the evolution algorithm as time series, and then uses an AEGM(1,1) to predict the reproductive individuals in the evolution process for generating the trial populations. Intuitively speaking, the motivation of the GPEAae is to amplify the weak effect in the grey model by using the iterative frame of evolution algorithms. Concretely, the GPEAae can superimpose and amplify the weak effect generated by the proposed 1-AIAGO which replaces the approximate (predicted) value with the corresponding accurate (original) value. The main contributions of this paper are as follows.

- **Propose a new grey prediction evolution algorithm.** As a new grey prediction evolution algorithm, the GPEAae has better overall performance and less computational complexity than the GPEAe.
- **Verify the advantages of the GPEAae.** The performance of GPEAae is verified on the CEC2014 benchmark functions and the set containing nine engineering constrained design problems. When comparing GPEAae with other state-of-the-art algorithms, the experimental results show excellent performance of GPEAae in terms of solution accuracy.
- **Propose a new grey prediction model.** The AEGM(1,1) and GPEAae are introduced with detailed explanations in Section 3. Several numerical experiments are carried out and discussed in Section 4. Finally, Section 5 gives the concluding remarks and future work.

II. PRELIMINARIES: THE GREY PREDICTION THEORY
The grey prediction theory has been widely used and achieved remarkable results over the past few decades, the prediction process of it can be generally divided into three stages to eventually acquire the prediction sequence of raw data. At first, two operations (first-order accumulating generation operation and background value generation operation) are implemented to transform the discrete raw data into data sequence with the law of approximate exponential growth. Whereafter, the raw data and transformed data sequence are brought into the grey prediction model (such as even grey model) to establish exponential function for generating the next elements. Finally, an inverse operation (first-order inverse accumulated generating operation) will be performed on just generated elements to obtain the forecasting elements of the raw data.

A. GREY DATA TRANSFORMING
The purpose of grey data transforming is to convert unordered non-negative columns into near-exponential increasing sequences, providing data basis for construction of grey prediction model. The accumulating generation operation (AGO) is the most fundamental data transforming operator in the grey prediction theory.

**Definition 1 (1-AGO):** Assume the sequence $Y^{(0)} = (y^{(0)}(1), y^{(0)}(2), \ldots, y^{(0)}(n))$ be an original sequence. Denote the first-order accumulating generation operation (1-AGO) of $Y^{(0)}$ by $Y^{(1)} = (y^{(1)}(1), y^{(1)}(2), \ldots, y^{(1)}(n))$, where

$$y^{(1)}(k) = \sum_{i=1}^{k} y^{(0)}(i), \quad k = 1, 2, \ldots, n. \quad (1)$$

The randomness in the raw data is weakened when the raw data is accumulated and generated as $Y^{(1)}$. Then, the two adjacent data in sequence $Y^{(1)}$ are averaged to obtain the generating sequence.

**Definition 2 (Background Value):** Given $y^{(1)}(k)$ as shown above, consider that $Y^{(1)} = (z^{(1)}(2), z^{(1)}(3), \ldots, z^{(1)}(n))$ denotes the background value generating operation of $Y^{(1)}$, where

$$z^{(1)}(k) = \frac{1}{2}(y^{(1)}(k) + y^{(1)}(k - 1)), \quad k = 2, 3, \ldots, n. \quad (2)$$

B. DATA PREDICTION FOR TRANSFORMED DATA SEQUENCE

**Definition 3 (Even Grey Model):** Given $y^{(0)}(k)$ and $z^{(1)}(k)$ as shown above, an even grey model (EGM(1,1)) is described as

$$y^{(0)}(k) + \alpha z^{(1)}(k) = \beta, \quad k = 1, 2, \ldots, n. \quad (3)$$

Development coefficient $\alpha$ evinces the development tendency of the time response sequence and raw data. The grey
action quantity $\beta$ evinces the changing relationship between data. To acquire the time response function, difference equation Eq. (3) of EGM(1,1) is converted into the corresponding whitization differential equation as

$$\frac{dy^{(1)}}{dt} + \alpha y^{(1)} = \beta.$$  \hspace{1cm} (4)

The solution of Eq. (4), which is known as time response function, is obtained as follows:

$$y^{(1)}(t) = (y^{(1)}(1) - \frac{\beta}{\alpha})e^{-\alpha(t-1)} + \frac{\beta}{\alpha}. \hspace{1cm} (5)$$

When $t = 1$, take $y^{(1)}(1) = y^{(0)}(1)$, then the time response sequence (after discretization) of EGM(1,1) is given by

$$\hat{y}^{(1)}(k+1) = (y^{(0)}(1) - \frac{\beta}{\alpha})e^{-\alpha k} + \frac{\beta}{\alpha}, \hspace{1cm} k = 1, 2, \ldots, n. \hspace{1cm} (6)$$

The $\hat{y}^{(1)}(k)$ makes up the predicted sequence $\hat{y}^{(1)}$.

C. PREDICTIVE GENERATION FOR RAW DATA

Prediction generating is an especially important step in the EGM(1,1), and the prediction sequence of raw data is ultimately figured out by the introduced inverse operator of 1-AGO.

**Definition 4 (1-IAGO):** Consider that $\hat{y}^{(0)} = (\hat{y}^{(0)}(1), \hat{y}^{(0)}(2), \ldots)$ is the finally obtained prediction sequence. Through a first-order inverse accumulated generating operation (1-IAGO), the elements in $\hat{y}^{(0)}$ can be acquired from the calculation of $\hat{y}^{(1)}$ as follows:

$$\begin{align*}
\hat{y}^{(0)}(1) &= y^{(0)}(1) \\
\hat{y}^{(0)}(k+1) &= \hat{y}^{(1)}(k+1) - \hat{y}^{(1)}(k) \\
&= (1 - e^{\alpha})y^{(0)}(1) - \frac{\beta}{\alpha}e^{-\alpha k}, \hspace{0.5cm} k = 1, 2, \ldots, n.
\end{align*} \hspace{1cm} (7)$$

III. THE GREY PREDICTION EVOLUTION ALGORITHM BASED ON AEGM(1,1)

Based on the grey prediction theory introduced above, this section first introduce the accelerated even grey model, then outline the specific calculation procedures of GPEAae in detail.

**Special Statement:** In the algorithm design concept, it is an eternal topic to realize the tradeoff between solution accuracy and complexity of an algorithm. In this paper, the accelerated even grey model is constructed under the circumstance of employing three data items to predict the fourth data (the reasons for constructing prediction models with three data items have been explained in [1]).

A. THE ACCELERATED EVEN GREY MODEL

As an improvement of EGM(1,1), the proposed accelerated even grey model (AEGM(1,1)) is identified by the new operation in generating the predictive data for raw data. The calculation steps of AEGM(1,1) will be introduced as follows.

(i) Data generating and transforming.

Assume $Y^{(0)} = (y^{(0)}(1), y^{(0)}(2), y^{(0)}(3))$ be an original sequence. According to the Eq. (1), we can get a 1-AGO sequence $Y^{(1)} = (y^{(1)}(1), y^{(1)}(2), y^{(1)}(3))$ as follows

$$Y^{(1)} = (y^{(0)}(1), y^{(0)}(1)+y^{(0)}(2), y^{(0)}(1)+y^{(0)}(2)+y^{(0)}(3)). \hspace{1cm} (8)$$

Then, from the Eq. (2), a MEANO sequence $Z^{(1)} = (z^{(1)}(2), z^{(1)}(3))$ is obtained as

$$Z^{(1)} = (y^{(0)}(1) + \frac{1}{2}y^{(0)}(2), y^{(0)}(1) + y^{(0)}(2) + \frac{1}{2}y^{(0)}(3)). \hspace{1cm} (9)$$

(ii) Establishing a grey model and solving the predictive expression.

Plug the above $Y^{(1)}, Z^{(1)}$ into the Eq. (3) and Eq. (4), we can get the $\alpha, \beta$, and $\hat{y}^{(1)}(4)$ as follows

$$\hat{y}^{(1)}(4) = (y^{(0)}(1) - \frac{\beta}{\alpha}e^{-3\alpha} + \frac{\beta}{\alpha}, \hspace{1cm} (10)$$

where

$$\begin{align*}
\alpha &= \frac{2(y^{(0)}(2) - y^{(0)}(3))}{y^{(0)}(2) + y^{(0)}(3)} \\
\beta &= \frac{2((y^{(0)}(2))^2 + y^{(0)}(1) \cdot y^{(0)}(2) - y^{(0)}(1) \cdot y^{(0)}(3))}{y^{(0)}(2) + y^{(0)}(3)}. \hspace{1cm} (11)
\end{align*}$$

(iii) Accelerated data forcasting for raw data.

In this stage, the accelerated 1-IAGO is employed to obtain the next predictive data of raw sequence.

**Definition 5 (1-AIAGO):** The accelerated 1-IAGO (1-AIAGO) for predicting the next data item of $y^{(0)}$ can be calculated as

$$\begin{align*}
\hat{y}^{(0)}(4) &= \hat{y}^{(1)}(4) - y^{(1)}(3) \\
&= (y^{(0)}(1) - \frac{\beta}{\alpha}e^{-3\alpha} + \frac{\beta}{\alpha} - y^{(0)}(1) - y^{(0)}(2) - y^{(0)}(3). \hspace{1cm} (12)
\end{align*}$$

1) ACCELERATION MECHANISM OF AEGM(1,1)

The 1-AIAGO is the source of AEGM(1,1)’s accelerating effect compared with EGM(1,1), and it directly influence the results of data forcasting for raw data. Then we will explain the acceleration mechanism of AEGM(1,1) under the circumstance of employing three data items $(y^{(0)}(1), y^{(0)}(2), y^{(0)}(3))$ to predict a fourth data item $(\hat{y}^{(0)}(4))$. In the 1-AIAGO of EGM(1,1), the $\hat{y}^{(0)}(4)$ is obtained by subtracting $\hat{y}^{(1)}(3)$ from $\hat{y}^{(1)}(4)$ (the $\hat{y}^{(1)}(3)$ and $\hat{y}^{(1)}(4)$ are the third and fourth prediction values of 1-AGO sequence $Y^{(1)}$, respectively). This could be explained by exponential function constructed in prediction process of EGM(1,1), which is the subtraction of the function values of the fourth and third points on the curve. When employing the 1-AIAGO of AEGM(1,1) to calculate the $\hat{y}^{(0)}(4)$, the $\hat{y}^{(1)}(3)$ in the 1-AIAGO of EGM(1,1) is replaced by $y^{(1)}(3)$ (the $y^{(1)}(3)$ is the third data in 1-AGO sequence $Y^{(1)}$). In other words, the 1-AIAGO adopts the original data $y^{(1)}(3)$ rather than the predicted data $\hat{y}^{(1)}(3)$ to generate the $\hat{y}^{(0)}(4)$. Compared with EGM(1,1), although AEGM(1,1) only has a different operation in the final stage of the prediction processes, this difference directly affects the generation of final prediction.
2) TIME COMPLEXITY ANALYSIS

While the 1-IAGO needs to calculate on two predicted values $\hat{y}^{(1)}(3)$ and $\hat{y}^{(1)}(4)$ for prediction, the 1-AIAGO calculates on the known original value $y^{(1)}(3)$ and the predicted value $\hat{y}^{(1)}(4)$. It can be seen that the 1-AIAGO makes full use of the original data information without calculating the predicted value $\hat{y}^{(1)}(3)$. This reduces the time complexity of the grey model to some extent. Replacing the approximate value ($\hat{y}^{(1)}(3)$) with the corresponding accurate value ($y^{(1)}(3)$) seems to save only a small amount of computational overhead. But under the iterative frame of evolutionary algorithms, this weak effect of saving computational overhead can be superimposed and amplified to ultimately form great influence on the time complexity of GPEAae.

B. THE GPEAae

The GPEAae is constructed based on the AEGM(1,1) above, and its evolution process will be introduced. At the beginning of GPEAae, the population is randomly initialized to generate some potential solutions in the feasible region. During the iterations, the parent populations reproduce its offspring by employing a reproduction operator inspired by AEGM(1,1) model. Finally, after looping a selection operator, GPEAae finds out the function-optimized solutions to reach the termination conditions.

1) INITIALIZATION

Similar to traditional population-based EAs, each generation population of GPEAae can be expressed as $X^g = (x_1^g, x_2^g, \ldots, x_N^g)$, $g = 0, 1, \ldots, g_{\text{max}}$. Where $g$ represents the current generation and $g_{\text{max}}$ means the maximum number of generations, $N$ is the number of individuals in the population. The individual in population $X^g$ is symbolized by $x_i^g = (x_{i,1}^g, x_{i,2}^g, \ldots, x_{i,D}^g)$, $i = 1, 2, \ldots, N$. Where $D$ denotes the dimensions of variables. Unlike traditional EAs, the proposed GPEAae needs to initialize three generation populations. So $3N$ individuals are randomly generated in feasible region, then they are sorted according to fitness values. Concretely, $N$ smallest individuals make up the first generation $X^0$ and the middle $N$ individuals make up the second generation $X^1$ ($g = 1$), at last, the remaining largest $N$ individuals make up the third generation $X^2$ ($g = 2$).

In the process of population initialization, the potential individuals are generated by randomly distributed numbers within the feasible region. As for the $i$th individual in first generation population $X^0$, the $j$th dimension element value of it is randomly generated by

$$x_{i,j}^0 = L_j + \text{rand}(0, 1) \cdot (U_j - L_j)$$  \hspace{1cm} (13)

where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, D$. The $\text{rand}(0, 1)$ represents a random number in the range $[0, 1]$. The $L_j$ and $U_j$ are the low and up boundaries of $j$th dimension element.

Discussion: Three initialized populations are essential for GPEAae, and there is an alternative and effective way to realize the initialization by means of other metaheuristic.

2) REPRODUCTION OPERATOR

A novel reproduction operator, named $\text{ae reproduction}$ operator, is designed based on AEGM(1,1). Broadly speaking, $\text{ae reproduction}$ operator treats the population sequence as a time series, and bring three neighbouring populations into the AEGM(1,1) to construct exponential functions for predicting offspring solutions. Specifically speaking, all calculations of $\text{reproduction}$ operator are performed at the element level of the individual in population.

Consider $X^g$ as a target population, then three neighbouring generations $X^{g-2}, X^{g-1}, X^g$ ($g \geq 2$) compose a population series to generate the trial population (denoted as $U^g$). The individual series is symbolized by $X_1^g, X_2^g, X_3^g$, which are randomly chosen from $X^{g-2}, X^{g-1}, X^g$, respectively. Let the trial vector $u^g_i = (u_{i,1}^g, u_{i,2}^g, \ldots, u_{i,D}^g)$ be the $i$th individual in trial population $U^g$. Then the formulation of

$$u_{i,j}^g = (x_{r_1,j} - \frac{\beta_{i,j}}{\alpha_{i,j}})e^{-3\alpha_{i,j}} + \frac{\beta_{i,j}}{\alpha_{i,j}} - x_{r_1,j} - x_{r_2,j} - x_{r_3,j},$$  \hspace{1cm} (14)

where

$$\begin{align*}
\alpha_{i,j} &= \frac{2(x_{r_2,j} - x_{r_3,j})}{x_{r_2,j} + x_{r_3,j}} \\
\beta_{i,j} &= \frac{2((x_{r_2,j})^2 + x_{r_1,j} \cdot x_{r_2,j} - x_{r_1,j} \cdot x_{r_3,j})}{x_{r_2,j} + x_{r_3,j}}.
\end{align*}$$  \hspace{1cm} (15)

As defined in section 2 above, the $\alpha_{i,j}$, $\beta_{i,j}$ are the grey developmental coefficient and grey control parameter, respectively.

![FIGURE 1. Geometric explanation of $\text{ae reproduction}$.](image)

The geometric explanation of the $\text{ae reproduction}$ operator is illustrated in Fig. 1. The symbols $x_{r_1}, x_{r_2},$ and $x_{r_3}$ represent three original data in the $\text{ae reproduction}$ operator. The green dots denote the transformed data after the 1-AGO operation. Based on the transformed data, an exponential function is then constructed. The yellow dot denotes the forth predicted data in the exponential function. After a 1-AIAGO operation, the predicted data $u_{i,j}^g$ is finally generated.
Note: (1) When the three elements of the individual series are close enough, the exponential prediction can’t be constructed to predict the next element, then a local searching strategy based on random disturbance will be run as a replacement. (2) When there are two of the three elements are close enough, the exponential prediction conducted by AEGM(1,1) will degenerate into a linear prediction. (3) When the elements of new individuals overstep the feasible region, they will be replaced by some random numbers generated within the feasible region. Please refer to the literature [1] for the specific expressions.

3) SELECTION OPERATOR
To ensure the convergence rate of GPEAae, a greedy selection operator is adopted for reserving some potential individuals into next generation. The most potential individuals are obtained from the comparing of objective function values between the target \( \vec{x}_i \) and trial \( \vec{u}_i \) individuals. When solving the minimization problems, if the \( \vec{u}_i \) acquires a smaller function value, it will be passed into next generation. Otherwise, the \( \vec{x}_i \) will be reserved in the population. This selection operator can be presented as follows:

\[
\vec{x}_{i+1}^{g} = \begin{cases} 
\vec{u}_i^g, & \text{if } f(\vec{u}_i^g) < f(\vec{x}_i^g) \\
\vec{x}_i^g, & \text{otherwise}.
\end{cases}
\]

(16)

IV. NUMERICAL EXPERIMENTS
In the numerical experiments, 30 CEC2014 benchmark functions and nine engineering constrained design optimization will be presented for comparing and evaluating the search performance of GPEAae and GPEAe. As completely new evolutionary algorithms, GPEAae and GPEAe are compared with DE and PSO on the CEC2014. During the recent years, DE and PSO are two of the most widely propagated and applied metaheuristics in solving many kinds of optimization problems. The comparison experiments with DE and PSO can well reveal the effectiveness and application prospect of GPEAae and GPEAe. In the subsequent nine engineering constrained design optimization experiments, GPEAae and GPEAe are compared with some original and improved well-known algorithms. The comparison results reflect the capability of GPEAae and GPEAe when resolving practical optimization problems. Both programs of GPEAae and GPEAe are coded and executed in MATLAB 2012a (Windows7; Intel(R) Core(TM) i5-4590 CPU @ 3.30GHZ; 4GB RAM). All the experimental results will be analyzed and discussed in detail.

A. CEC2014 BENCHMARK FUNCTIONS EXPERIMENTS
The CEC2014 benchmark functions first appeared in the CEC’2014 special section on real parameter optimization, including 30 functions with different characteristics. These 30 benchmark functions can be broadly categorized into four groups: unimodal functions (F1-F3), multimodal functions (F4-F16), hybrid functions (F17-F22), and composition functions (F23-F30). The detailed information of the 30 benchmark functions is provided in [23].

1) PARAMETER SETTINGS
To solve the benchmark functions, the same population size of 50 are used by all the compared algorithms (GPEAae, GPEAe, PSO and DE), and two different dimension sizes 10D and 30D (D is the dimension variable) are separately simulated. As the termination criterion for the four algorithms, the maximum function evaluations (FEs) of 10D and 30D simulations are set to 100000 and 500000, respectively. All experimental results are obtained through 30 independent runs, and the performance indices such as the best, mean and standard deviation of the error values \( (F(x) - F(x^*)) \) are recorded and assessed. Where \( F(x) \) is the minimum function value obtained by the best solution and \( F(x^*) \) is the global optimal function value. The remaining parameters of DE and PSO are set as follows.

- As for DE (rand/1): \( F = 0.5, CR = 0.5 \).
- As for PSO: \( w = 1.0, c1 = c2 = 2.0 \).

2) COMPARISONS ON SOLUTION ACCURACY AND CONVERGENCE RATE OF 10D CEC2014 BENCHMARK FUNCTIONS
In these experiments, GPEAae is compared with GPEAe, PSO and DE. The statistical results that can reveal the detail performance on the solution accuracy of four algorithms are recorded in Table 1 and Table 2, including the Best, Mean, and Std values for 30 10D benchmark functions in 30 independent runs. The best results of each index among four compared algorithms are prominently represented in boldface, and the average ranks are listed at the end of Table 2. Furthermore, the convergence characteristics of nine representative 10D functions F3, F7, F15, F16, F17, F19, F24, F28, and F30 are illustrated in Fig. 1. The curves are given to show the performance on convergence rate of four compared algorithms more clearly. From the subfigure (c) in Fig. 2, it can be seen that GPEAae can achieve higher solution accuracy with faster convergence rate when compared with GPEAe, PSO, and DE. As the subfigure (c), (e), (f), (g) and (i) in Fig. 2 show, the GPEAae has competitive convergence rate and can achieve the highest solution accuracy in the end. It is necessary to emphasize that the smaller the Best value obtained by the algorithm, the higher the solution accuracy of the algorithm is to approach the theoretical optimal value. With respect to the Best index, the function values obtained by GPEAae are the smallest among four compared algorithms. In the overall ranking of Best values, GPEAae ranks first on 30 functions with an average ranking of 1.87, while DE, GPEAe, and PSO rank second, third, and last with the average ranking of 2.00, 2.40, and 3.57, respectively. This indicates that GPEAae performs higher solution accuracy than DE, GPEAe and PSO. Unfortunately, GPEAae ranks only second in terms of the Mean and Std values, worse than DE and better than GPEAe and PSO.
## TABLE 1. Comparison results of GPE Ae, GPE Ae, PSO and DE on 10D CEC2014 functions.

| Functions | \( p_{\text{Rank}} \) | Statistics | GPE Ae | GPE Ae | PSO | DE |
|-----------|------------------------|-----------|--------|--------|-----|----|
|           | D=10, T=2000           |           |        |        |     |    |
| **Unimodal Functions** | | | | | | |
| F1 | Best | 0.025933 | 0.029295 | 2931.855599 | 777.515326 |
| Mean | 433.594493 | 3277.199837 | 7.08E+04 | 2300.796262 |
| Std | 552.849105 | 1.16E+04 | 9.76E+04 | 950.115851 |
| F2 | Best | 0.018274 | 0.031761 | 26.696887 | 0.000000 |
| Mean | 32.155242 | 2.771291 | 1461.895232 | 0.000000 |
| Std | 119.150947 | 6.278440 | 1793.153897 | 0.000000 |
| F3 | Mean | 3.69E-05 | 0.000256 | 0.065710 | 0.000000 |
| Std | 108.495171 | 299.118880 | 0.538886 | 0.000000 |
| Std | 306.386131 | 994.567078 | 0.353980 | 0.000000 |
| F4 | Mean | 3.07E-05 | 1.43E-07 | 0.021118 | 0.032029 |
| Std | 17.243921 | 8.710095 | 26.151973 | 9.708502 |
| F5 | Mean | 20.312436 | 20.713010 | 20.112123 | 20.166646 |
| Std | 0.177163 | 0.197647 | 0.092448 | 0.107414 |
| F6 | Mean | 2.255009 | 5.241859 | 6.028942 | 0.000000 |
| Std | 1.736785 | 2.419513 | 1.307821 | 0.000000 |
| F7 | Mean | 0.000000 | 0.027106 | 0.022739 | 0.031951 |
| Std | 0.123764 | 0.140434 | 0.318711 | 0.133425 |
| F8 | Mean | 3.979836 | 4.959017 | 12.934610 | 0.000000 |
| Std | 13.896244 | 13.306101 | 24.114290 | 0.000000 |
| F9 | Mean | 12.403810 | 13.000784 | 24.641875 | 13.512764 |
| Std | 5.403894 | 5.192566 | 11.050620 | 2.664316 |
| F10 | Mean | 3.477415 | 21.949498 | 244.550068 | 0.000000 |
| Std | 479.741824 | 577.022099 | 648.172508 | 0.154191 |
| F11 | Mean | 686.260024 | 562.865295 | 908.427792 | 685.810720 |
| Std | 416.112657 | 302.330240 | 287.249282 | 99.245497 |
| F12 | Mean | 5.99E-08 | 1.25E-11 | 0.027556 | 0.313613 |
| Std | 0.839970 | 0.182745 | 0.305449 | 0.509366 |
| F13 | Mean | 0.012036 | 0.048127 | 0.182547 | 0.000000 |
| Std | 0.336651 | 0.296485 | 0.437354 | 0.150387 |
| F14 | Mean | 0.152409 | 0.137979 | 0.130793 | 0.028184 |
| Std | 0.154676 | 0.168529 | 0.095903 | 0.067633 |
| F15 | Mean | 0.372412 | 0.380030 | 0.351995 | 0.156070 |
| Std | 0.157999 | 0.200994 | 0.116866 | 0.040588 |
| F16 | Mean | 0.557230 | 0.578475 | 1.632539 | 0.951759 |
| Std | 1.749102 | 1.758816 | 4.112686 | 1.644372 |
| F17 | Mean | 0.104575 | 0.821985 | 1.732869 | 0.299365 |
| Std | 1.507740 | 2.367155 | 2.346952 | 1.420956 |
| F18 | Mean | 2.817211 | 3.072093 | 3.116819 | 2.236517 |
| Std | 0.574334 | 0.357057 | 0.374839 | 0.327290 |
| F19 | Mean | 0.275645 | 1.239997 | 557.363857 | 20.482630 |
| Std | 97.895887 | 177.534702 | 2245.544563 | 43.500885 |
| F20 | Mean | 88.368311 | 149.323202 | 2185.318884 | 12.525592 |
| Std | 5.409471 | 7.394273 | 9249.306785 | 1.981074 |
| F21 | Mean | 3.571822 | 6.971810 | 1.04E+04 | 0.616036 |
| Std | 0.066267 | 0.835359 | 1.799366 | 0.096852 |
| F22 | Mean | 1.147871 | 3.426292 | 3.511221 | 0.228408 |
| Std | 1.174244 | 1.527802 | 1.183965 | 0.102373 |
| F23 | Mean | 0.839049 | 1.842776 | 28.370020 | 0.000106 |
| Std | 5.657133 | 5.470634 | 544.583525 | 0.161005 |
| F24 | Mean | 4.606217 | 5.470634 | 1067.144265 | 0.112467 |
| Std | 0.009159 | 0.523336 | 310.873147 | 0.002659 |
| F25 | Mean | 19.955105 | 99.511085 | 2895.539852 | 0.228906 |
| Std | 29.873937 | 114.921080 | 2027.029790 | 0.192326 |
| F26 | Mean | 0.199008 | 0.460761 | 22.951059 | 2.272465 |
| Std | 36.828648 | 85.120985 | 101.965613 | 0.066016 |
| F27 | Mean | 47.608359 | 84.610434 | 63.972864 | 0.126095 |
However, the statistical results in Table 1 and Table 2 show that GPEAae performs better solution accuracy than GPEAe on 22 out of the 30 functions. When comparing the Mean and Std values, GPEAae also outperforms GPEAe on 22 and 19 functions, respectively. Based on the above observation, it can be concluded that although GPEAae did not consistently perform best in terms of all performance indices, its comprehensive performance is better than GPEAe, PSO and DE in 10D CEC2014 benchmark function experiments.

3) COMPARISONS ON SOLUTION ACCURACY AND CONVERGENCE RATE OF 30D CEC2014 BENCHMARK FUNCTIONS

To explore the performance of GPEAae in solving high-dimensional functions, the dimensions of 30 benchmark functions are set to 30 in these experiments. Table 3 and Table 4 shows the statistical results obtained by GPEAae, GPEAe, PSO and DE in 30 independent runs. Similar to the 10D CEC2014 experimental results above, the performance of GPEAae on solution accuracy is still the best. The Best values acquired from GPEAae are the smallest among the four algorithms on functions F1, F4, F9, F10, F11, F15, F16, F18, F21, F23, F29, and F30. All the subfigures in Fig. 3 illustrate that the GPEAae has highest solution accuracy among all compared algorithms. Especially, the subfigure (f) show that the fast convergence rate in the middle iterations help GPEAae finally achieves high solution accuracy. The curves demonstrate that GPEAae behaves both competitive solution accuracy and convergence rate compared with the other three competitors. When comparing the performance between GPEAae and GPEAe, the solution accuracy obtained by GPEAae is better than that of GPEAe on eighteen functions, and GPEAae can achieve smaller Mean and Std values than GPEAe on more than half of the all 30 functions. The comparisons show that GPEAae performs much better than GPEAe for these 30 functions. Even when the dimension of CEC2014 benchmark functions change from 10 to 30, GPEAae still has stable performance. All in all, the experimental results show that although the robustness of GPEAae is not the best, it can achieve better solution accuracy than GPEAe, PSO, and DE in 30D CEC2014 benchmark function experiments.

4) SIGN TEST

As a widely used test method, the sign test [24] is employed to judge whether the difference of two selected algorithms is significant enough. The Best values obtained by the GPEAae and other compared algorithms are selected as the evaluation indicator of the sign test. All the Best values of the 60 functions (composed of 30 10D CEC2014 and 30 30D functions F4, F9, F10, F11, F15, F16, F18, F21, and F29. All the subfigures in Fig. 3 illustrate that the GPEAae has highest solution accuracy among all compared algorithms. Especially, the subfigure (f) show that the fast convergence rate in the middle iterations help GPEAae finally achieves high solution accuracy. The curves demonstrate that GPEAae behaves both competitive solution accuracy and convergence rate compared with the other three competitors. When comparing the performance between GPEAae and GPEAe, the solution accuracy obtained by GPEAae is better than that of GPEAe on eighteen functions, and GPEAae can achieve smaller Mean and Std values than GPEAe on more than half of the all 30 functions. The comparisons show that GPEAae performs much better than GPEAe for these 30 functions. Even when the dimension of CEC2014 benchmark functions change from 10 to 30, GPEAae still has stable performance. All in all, the experimental results show that although the robustness of GPEAae is not the best, it can achieve better solution accuracy than GPEAe, PSO, and DE in 30D CEC2014 benchmark function experiments.
CEC2014 functions) are obtained through 30 independent runs. The signs ‘+’, ‘≈’ and ‘−’ in Table 5 denote that the GPEAae has better, nearly the same and worse performance than the compared algorithms, respectively. In the sign test, set a null hypothesis that there is no significant difference between the GPEAae and the compared algorithms. When the P value is less than 0.05, the null hypothesis is rejected. As shown in the last column of Table 5, the p value (GPEAae vs. GPEAe, GPEAae vs. PSO, GPEAae vs. DE) is 0.015543, 2.21E-08 and 0.051510, respectively. In other words, the performance of GPEAae is significantly better than that of GPEAe and PSO. But compared with DE, the advantage of GPEAae is slight and not significant enough.

B. ENGINEERING CONSTRAINED DESIGN EXPERIMENTS

In this investigation, GPEAae and GPEAe are compared with some original and improved well-known algorithms on nine practical engineering design problems. The experimental results reflect the capability of GPEAae and GPEAe when resolving practical optimization problems. The nine engineering cases are as follows: three-bar truss, pressure vessel, tension/compression spring, welded beam, speed reducer, gear train, cantilever beam, heat exchanger and tubular column.

1) PARAMETER SETTINGS

In the past few years, many effective constraint processing methods have been developed, such as penalty functions and $\varepsilon$-constrained methods [25]. The GPEAae and GPEAe employ the $\varepsilon$-constrained methods to handle the constraits with $\varepsilon = 1E - 6$. The parameter setting of nine engineering design problems is presented in Table 6. The ‘N’, ‘T’ and ‘D’ denote population size, maximal iteration times and number of optimization parameters, respectively. All experimental results are obtained through 50 independent runs, and some performance indices are recorded such as the Best, Mean, Worst, Standard deviation (Std), and the Function estimation times (FEs) needed for getting current optimum value. The objective functions, constraint conditions and descriptions of nine engineering design problems are provided in [26] and [27].
### TABLE 3. Comparison results of GPEAe, GPEAe, PSO and DE on 30D CEC2014 functions.

| Functions | \( \rho^{\text{Rank}} \) | Statistics | GPEAe | GPEAe | PSO | DE |
|-----------|-----------------|-----------|-------|-------|-----|----|
|           |                 | \( D=30, T=10000 \) |       |       |     |    |
| Best      |                 |           |       |       |     |    |
| Mean      |                 |           |       |       |     |    |
| Std       |                 |           |       |       |     |    |
|            | Unimodal Functions |            |       |       |     |    |
| F1        |                 |           |       |       |     |    |
| Mean      | 2.27E+05        | 2.90E+05  | 1.08E+06 | 2.66E+07 | 7.23E+06 |
| Std       | 0.01715         | 0.15701   | 932.066896 | 0.000000 | 0.000000 |
| F2        |                 |           |       |       |     |    |
| Mean      | 0.187974        | 0.280692  | 3.74E+07  | 5.68E-15  | 1.16E-14  |
| Std       | 0.379784        | 0.417485  | 2.05E+08  | 0.000000  | 0.000000  |
| F3        |                 |           |       |       |     |    |
| Mean      | 749.413360      | 727.754626 | 7.308568  | 2.65E-14  | 2.88E-14  |
| Std       | 1501.989926     | 2314.821924 | 2.651993  | 0.000000  | 0.000000  |
| F4        |                 |           |       |       |     |    |
| Mean      | 0.046339        | 0.832998  | 4.428207  | 0.359501  | 0.000000  |
| Std       | 0.169566        | 0.14522   | 0.016292  | 4.38E-14  | 0.000000  |
| F5        |                 |           |       |       |     |    |
| Mean      | 47.74867        | 46.763015 | 87.556321 | 54.317767 | 12.129267 |
| Std       | 73.792598       | 78.170408 | 112.26268 | 72.466523 | 24.106646 |
| F6        |                 |           |       |       |     |    |
| Mean      | 16.939344       | 17.678351 | 16.276757 | 21.269441 | 24.106646 |
| Std       | 52.732764       | 61.678356 | 94.521091 | 129.52245 | 557.68283 |
| F7        |                 |           |       |       |     |    |
| Mean      | 89.546017       | 93.857466 | 138.995486 | 158.344750 | 254.956480 |
| Std       | 23.092538       | 20.489500 | 29.895520 | 12.129267 | 24.106646 |
| F8        |                 |           |       |       |     |    |
| Mean      | 1381.883702     | 1317.122131 | 678.424598 | 244.023626 | 557.68283 |
| Std       | 2247.835656     | 2556.977151 | 3126.953683 | 557.68283 | 557.68283 |
| F9        |                 |           |       |       |     |    |
| Mean      | 7081.092950     | 4786.593338 | 4253.678106 | 6227.136319 | 254.956480 |
| Std       | 1164.308597     | 1570.072741 | 574.189205 | 254.956480 | 254.956480 |
| F10       |                 |           |       |       |     |    |
| Mean      | 764.912168      | 6256.997501 | 3.12E-05  | 4.91E-05  | 1.28E+04  |
| Std       | 5183.373530     | 4539.674280 | 3.04E+05  | 1.81E+05  | 1.28E+04  |
| F11       |                 |           |       |       |     |    |
| Mean      | 19.161494       | 37.371797  | 253.237495 | 152.71412 | 152.71412 |
| Std       | 1344.789987     | 1303.590175 | 1808.273960 | 222.394395 | 222.394395 |
| F12       |                 |           |       |       |     |    |
| Mean      | 2035.176058     | 2152.170097 | 1870.704462 | 58.812302 | 58.812302 |
| Std       | 6.713694        | 4.645185  | 10.860500 | 4.666766 | 4.666766 |
| F13       |                 |           |       |       |     |    |
| Mean      | 23.163547       | 17.548954  | 31.042612 | 5.468001 | 5.468001 |
| Std       | 25.031344       | 20.442593  | 29.930315 | 0.443983 | 0.443983 |
| F14       |                 |           |       |       |     |    |
| Mean      | 32.427499       | 22.952701  | 133.195754 | 49.787746 | 49.787746 |
| Std       | 120.562831      | 142.394711 | 257.713946 | 60.793589 | 60.793589 |
| F15       |                 |           |       |       |     |    |
| Mean      | 82.850706       | 143.422535 | 78.568890 | 5.605741 | 5.605741 |
| Std       | 213.535122      | 647.906984 | 3207.121836 | 3305.969660 | 3305.969660 |
| F16       |                 |           |       |       |     |    |
| Mean      | 2928.042537     | 4228.215664 | 2.75E+08  | 2.94E+08  | 2.94E+08  |
| Std       | 3682.477577     | 4604.645388 | 7.49E+04  | 1.542.362031 | 1.542.362031 |
| F17       |                 |           |       |       |     |    |
| Mean      | 183.821348      | 408.013504 | 410.863059 | 60.165038 | 60.165038 |
| Std       | 594.999585      | 686.624769 | 838.602953 | 118.802465 | 118.802465 |
| F18       |                 |           |       |       |     |    |
| Mean      | 235.798036      | 173.875027 | 196.533849 | 58.995920 | 58.995920 |
| Std       |                  |           |       |       |     |    |
TABLE 4. Continued Tab 3: Comparison results of GPEAae, GPEAe, PSO and DE on 30D CEC2014 functions.

| Functions | $\hat{P}^{rank}$ | D=30, T=10000 |
|-----------|----------------|----------------|
|           | Statistics     | GPEAae | GPEAe | PSO | DE   |
| F23       | Best           | 315.244102 | 315.244102 | 315.244102 | 315.244102 |
|           | Mean           | 315.244104 | 315.244132 | 315.252882 | 315.244102 |
|           | Std             | 4.98E-06 | 0.000128 | 0.019556 | 5.78E-14 |
| F24       | Best           | 225.671236 | 224.093293 | 200.848266 | 221.334564 |
|           | Mean           | 241.680509 | 238.850806 | 227.679795 | 222.301319 |
|           | Std             | 8.86E-12 | 8.634252 | 7.683223 | 6.742009 |
| F25       | Best           | 204.697709 | 203.920366 | 210.951637 | 209.225672 |
|           | Mean           | 215.438251 | 215.161187 | 219.777449 | 212.381248 |
|           | Std             | 3.948386 | 3.998933 | 5.894419 | 1.985969 |

TABLE 5. Sign tests between GPEAae and other algorithms.

| GPEAae-algorithms | + | ≈ | − | p value |
|-------------------|---|---|---|---------|
| GPEA              | 38 | 2 | 20 | 0.015543 |
| PSO               | 52 | 0 | 8  | 2.21E-08 |
| DE                | 31 | 2 | 27 | 0.051510 |

TABLE 6. Parameter setting of GPEAae and GPEAe on nine engineering design problems.

| Problem                  | N  | T  | D  |
|--------------------------|----|----|----|
| Three-bar truss          | 20 | 500| 2  |
| Pressure vessel          | 20 | 2000| 4  |
| Tension/Compression spring | 20 | 1000| 3  |
| Welded beam              | 20 | 2000| 4  |
| Speed reducer            | 20 | 1000| 7  |
| Gear train               | 20 | 500 | 4  |
| Cantilever beam          | 20 | 1000| 5  |
| Heat exchanger           | 20 | 6000| 8  |
| Tubular column           | 20 | 2000| 2  |

2) EXPERIMENTS ON NINE ENGINEERING DESIGN PROBLEMS

Case 1 (Three-Bar Truss Design Problem): The components of this problem is illustrated in Fig. 4. Table 7 represents the statistical results acquired from GPEAae, GPEAe and other seven algorithms, including the ant lion optimizer (ALO) [28], dynamic stochastic selection (DEDS) [29], hybrid particle swarm optimization with differential evolution (PSO-DE) [30], mine blast algorithm (MBA) [31], differential evolution with level comparison (DELC) [32], hybrid evolutionary algorithm (HEAA) [33], and crow search algorithm (CSA) [26]. The optimal function value for this problem is acquired from GPEAae with $f(X) = 263.895712$ at $X = \{0.788695, 0.408192\}$, which means that GPEAae can attain better solution accuracy when compared with GPEAe and other six algorithms. The index FEs in Table 7 are recorded to reflect the computational overhead. The GPEAae has the smallest FEs equals to 7700, this indicates that GPEAae reaches the currently optimal solution with the lowest computational overhead. So it can be concluded that GPEAae has better solution accuracy and convergence rate than the compared algorithms on this problem.

Case 2 (Pressure Vessel Design Problem): This problem is a well-known problem from Kannan and Kramer [34]. Fig. 5 shows the pressure vessel and its parameters. Table 8 shows the statistical results acquired from GPEAae, GPEAe and other seven algorithms, namely, grey wolf optimizer (GWO) [35], GA based co-evolution model (CGA) [36], co-evolutionary PSO (CPSO) [37], GA through the use of dominance-based tour tournament selection (DGA) [38], hybrid PSO (HPSO) [39], co-evolutionary differential evolution (CDE) [40], and CSA. For this problem, both GPEAae and GPEAe can reach the same optimal function value $f(X) = 605.708025$ which is the smallest among all
the algorithms except GWO. In other words, GPEAae and GPEAe perform equally well in terms of solution accuracy. But for computational overhead, the FEs of GPEAae is equal to 20280 which is slightly better than 21180 FEs of GPEAe. And the both FEs of GPEAae and GPEAe are far superior to other compared algorithms. Although the Mean and Std of GPEAae on this problem perform generally, GPEAae can attain competitive solution accuracy through the smallest computational overhead.

Case 3 (Tension/Compression Spring Design Problem): This problem was taken from Belegundu and Arora [41]. The schematic diagram of this problem are illustrated in Fig. 6. In this problem, the statistical results of GPEAae and GPEAe are compared with nine algorithms include GWO, modified

| Algorithms |
|------------|
| X1         |
| X2         |
| Worst      |
| Mean       |
| Std        |
| Best       |
| FEs        |
|------------|
| ALO        | 0.788663 |
| DEDS       | 0.785675 |
| HEAA       | 0.786680 |
| PSO-DE     | 0.788675 |
| DELC       | 0.788675 |
| MBA        | 0.785665 |
| CSA        | 0.786675 |
| GPEAe      | 0.788719 |
| GPEAae     | 0.788695 |

Note: ‘N.A’ refers to unavailable data.
TABLE 8. Comparison results on the pressure vessel design problem.

| Algorithms | X1    | X2    | X3    | X4    | Worst | Mean   | Std    | Best | FE |
|------------|-------|-------|-------|-------|-------|--------|--------|------|----|
| GWO        | 0.81250 | 0.43750 | 42.089181 | 176.758731 | N.A   | N.A    | N.A    | 6051.563900 | N.A |
| CGA        | 0.81250 | 0.43750 | 40.323900 | 200.000000 | 6308.497000 | 6293.843200 | 7.413300 | 6288.744500 | 900000 |
| DGA        | 0.81250 | 0.43750 | 42.0897400 | 176.654000 | 6469.322000 | 6177.253300 | 130.929700 | 6059.946300 | 80000 |
| CPSO       | 0.81250 | 0.43750 | 42.091300 | 176.746500 | 6363.804100 | 6147.133200 | 86.450000 | 6061.077700 | 240000 |
| HPSO       | 0.81250 | 0.43750 | 42.098400 | 176.636600 | 6288.677000 | 6099.932300 | 86.200000 | 6059.714300 | 81000 |
| CDE        | 0.81250 | 0.43750 | 42.098411 | 176.637690 | 6371.045500 | 6085.230300 | 43.013000 | 6059.734000 | 204800 |
| CSA        | 0.81250 | 0.43750 | 42.098445 | 176.636599 | 7332.841621 | 6342.499106 | 384.954316 | 6059.714363 | 250000 |
| GPEAe      | 0.81250 | 0.43750 | 42.098497 | 176.635954 | 7322.749386 | 6215.489177 | 7332.858319 | 6059.708025 | 21180 |
| GPEAae     | 0.81250 | 0.43750 | 42.098497 | 176.635954 | 7273.006261 | 6156.755311 | 7332.838526 | 6059.708025 | 20280 |

which is the smallest. This means that GPEAae and GPEAe perform equally well in terms of solution accuracy. With regard to the computational overhead, GPEAae reaches the global optimum with a ranking second FE of 35860, which is the best among all compared algorithms except MDE with 24000 FE. That is to say, GPEAae only needs to sacrifice just a little computational overhead to obtain better solution accuracy.

Case 5 (Speed Reducer Design Problem): The components of this problem are presented in Fig. 8. As the Table 11 shows, the performance of GPEAae, GPEAe is compared with other six algorithms, including MDE, DEDS, HEAA, DELC, POS-DE, and MBA. The optimal function value for this problem is acquired from GPEAae with $f(X) = 2994.468240$ at $X = \{3.499997, 0.700000, 17.000000, 7.300001, 7.715311, 3.350214, 5.286653\}$, which means that GPEAae can attain better solution accuracy when compared with GPEAe and other six algorithms. It is obvious that GPEAae uses lower computational overhead (with 19980 FE) than all the compared algorithms except MBA (with 6300 FE). This indicates that GPEAae can reach the optimal function value with competitive computational overhead.

Case 6 (Gear Train Design Problem): The components of this problem are illustrated in Fig. 9. The algorithms applied to this problem include GPEAae, ALO, ABC, MBA, GPEAe, and CSA. Table 12 shows the comparison results acquired from them. It can be observed, all six algorithms can reach the same optimal function value which equal to 2.7E-12. With respect to FE and Std, the GPEAae performs mediocre, better than MBA, CSA and worse than GPEAe, ABC, ALO. That is to say, although the computational overhead of GPEAae is
TABLE 9. Comparison results on the tension/compression spring design problem.

| Algorithms | X1    | X2    | X3    | Worst | Mean  | Std   | Best   | FEs   |
|------------|-------|-------|-------|-------|-------|-------|--------|-------|
| GWO        | 0.051690 | 0.356737 | 11.288850 | N.A   | N.A   | 0.00002 | 0.012665 | 24000 |
| MDE        | 0.051688 | 0.356692 | 11.290483 | 0.012674 | 0.012666 | 0.000002 | 0.012665 | 240000 |
| CPSO       | 0.051728 | 0.356744 | 11.244543 | 0.012924 | 0.012730 | 0.000052 | 0.012675 | 240000 |
| HPSO       | 0.051706 | 0.357126 | 11.265083 | 0.012719 | 0.012707 | 0.000016 | 0.012665 | 81000 |
| DEES       | 0.051689 | 0.356717 | 11.288965 | 0.012738 | 0.012669 | 0.000013 | 0.012665 | 240000 |
| HEAA       | 0.051689 | 0.356729 | 11.288229 | 0.012665 | 0.012665 | 1.40E-09 | 0.012665 | 240000 |
| DELC       | 0.051689 | 0.356717 | 11.288965 | 0.012666 | 0.012665 | 1.308-07 | 0.012665 | 200000 |
| MBA        | 0.051656 | 0.355940 | 11.344665 | 0.012900 | 0.012713 | 0.000063 | 0.012665 | 7650  |
| CSA        | 0.051894 | 0.356717 | 11.289012 | 0.012670 | 0.012666 | 0.000001 | 0.012665 | 50000 |
| GPEAe      | 0.051750 | 0.358190 | 11.203073 | 0.013378 | 0.012759 | 0.000145 | 0.012665 | 19980 |
| GPEAae     | 0.051685 | 0.356631 | 11.294000 | 0.013453 | 0.012720 | 0.000116 | 0.012665 | 19980 |

TABLE 10. Comparison results on the welded beam design problem.

| Algorithms | X1    | X2    | X3    | X4    | Worst | Mean  | Std   | Best   | FEs   |
|------------|-------|-------|-------|-------|-------|-------|-------|--------|-------|
| GWO        | 0.205676 | 3.478377 | 9.036810 | 0.205778 | N.A   | N.A   | N.A   | 1.726240 | 24000 |
| CGA        | 0.205986 | 3.471328 | 9.020224 | 0.206480 | 1.993408 | 1.792654 | 0.074900 | 1.728226 | 80000 |
| MDE        | 0.205730 | 3.470489 | 9.036624 | 0.205730 | 1.724854 | 1.724853 | N.A   | 1.724852 | 240000 |
| CPSO       | 0.202369 | 3.544214 | 9.042210 | 0.205723 | 1.782143 | 1.748313 | 0.012900 | 1.728024 | 240000 |
| HPSO       | 0.205730 | 3.470489 | 9.036624 | 0.205730 | 1.814295 | 1.749040 | 0.040100 | 1.728452 | 81000 |
| MDA        | 0.205729 | 3.470493 | 9.036626 | 0.205729 | 1.724853 | 1.724853 | 6.94E-19 | 1.724853 | 47340 |
| CSA        | 0.205730 | 3.470493 | 9.036624 | 0.205730 | 1.724853 | 1.724853 | 1.19E-15 | 1.724853 | 100000 |
| GPEAe      | 0.205731 | 3.470467 | 9.036624 | 0.205730 | 1.732281 | 1.725037 | 0.001100 | 1.724851 | 36100 |
| GPEAae     | 0.205731 | 3.470467 | 9.036624 | 0.205730 | 1.727564 | 1.724914 | 0.000440 | 1.724851 | 35860 |

TABLE 11. Comparison results on the speed reducer design problem.

| Algorithms | X1    | X2    | X3    | X4    | X5    | X6    | X7    | Worst     | Mean      | Std       | Best      | FEs     |
|------------|-------|-------|-------|-------|-------|-------|-------|-----------|-----------|-----------|-----------|---------|
| GPEAe      | 1.90E-12 | 0.000008 | 0.000006 | 0.000006 | 0.000006 | 0.000006 | 0.000006 | 1.90E-12 | 1.90E-12 | 1.90E-12 | 1.90E-12 | 477430 |
| GPEAe      | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.471066 |
| FEs        | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 | 240000 |

FIGURE 7. Welded beam design problem.

not so excellent when solving this problem, but GPEAae can obtain a good solution accuracy.

Case 7 (Cantilever Beam Design Problem): The overall schematic of this problem is illustrated in Fig. 10. Table 13 represents the statistical results obtained by GPEAae, GPEAe and other five algorithms, including moth-flame optimization algorithm (MFO) [43], method of moving asymptotes (MMA) [44], multi-verse optimizer (MVO) [45], symbiotic organisms search (SOS) [46], generalized convex approximation (GCA) [44] and CSA. The optimal function value for this problem is acquired from both GPEAae and GPEAe with the same \( f(X) = 1.339957 \), which means that GPEAae and GPEAe can attain better solution accuracy when compared with other six algorithms. At the same time, GPEAae and GPEAe reach the same FEs equal to 19920 (smaller than 125000 FEs of CS and larger than 15000 FEs of SOS). This indicates that GPEAae and GPEAe reach the same optimal
TABLE 12. Comparison results on the gear train design problem.

| Algorithms | X1 | X2 | X3 | X4 | Worst | Mean | Std | Best | FEs |
|------------|----|----|----|----|-------|------|-----|------|-----|
| ALO        | 49 | 19 | 16 | 43 | N.A   | N.A  | N.A | 2.70E-12 | 120 |
| ABC        | 49 | 19 | 16 | 43 | N.A   | 3.60E-10 | 5.52E-10 | 2.70E-12 | 60  |
| MBA        | 43 | 19 | 16 | 49 | 2.10E-08 | 2.50E-09 | 3.94E-09 | 2.70E-12 | 1120 |
| CSA        | 49 | 19 | 16 | 43 | 3.20E-08 | 2.06E-09 | 5.10E-09 | 2.70E-12 | 100000 |
| GPEAe      | 49 | 19 | 16 | 43 | 3.45E-09 | 6.88E-10 | 8.76E-10 | 2.70E-12 | 900  |
| GPEAae     | 49 | 19 | 16 | 43 | 1.12E-08 | 1.17E-09 | 1.77E-09 | 2.70E-12 | 1060 |

Solution when spending the same computational overhead. So it can be concluded that both GPEAae and GPEAe have satisfactory performance of solution accuracy and convergence rate on this problem.

Case 8 (Heat Exchanger Design Problem): As another constrained engineering minimization test problem, the heat exchanger design problem was taken from Deb [47]. As Table 14 shows, the performance of GPEAae and GPEAe is compared with other five algorithms from the literature, namely, harmony search (HS) [48], genetic algorithms (GAs) [47], changing range genetic algorithm (CRGA) [49], proposed harmony search (PHS) [27], and improving proposed harmony search (IPHS) [27]. The optimal function value obtained by GPEAae is 

$$f(X) = 7049.254884 \text{ with } X = \{576.280035, 1355.550094, 5117.424755, 181.764537, 295.302805, 218.235863, 286.462132, 395.302905\}.$$ 

It is obvious that the GPEAae can reach the smallest optimal solution. Moreover, GPEAae requires fewer FEs than other compared algorithms except GPEAe (with 100300 FEs). Overall, GGPGEAae has achieved competitive solution accuracy in heat exchanger design problem by sacrificing very little computational overhead.

Case 9 (Tubular Column Design Problem): The overall schematic of this problem is illustrated in Fig. 11. The statistical results of GPEAae and GPEAe are compared with CSA, cuckoo search algorithm with particle swarm optimization (CS-PSO) [50] and cuckoo search algorithm with genetic algorithm (CS-GA) [50]. As shown in Table 15, both GPEAae and GPEAe can reach the same optimal function value 

$$f(X) = 1.724851$$

which is the smallest. This means that GPEAae and GPEAe perform equally well in terms of solution accuracy. As for computational overhead, the FEs of GPEAae is equal to 18300 which is only worse than 15000 FEs of CS. This indicates that GPEAae has competitive performance in both computational overhead and solution accuracy on this problem.

3) STATISTICAL ANALYSES OF ENGINEERING RESULTS

The optimal feasible solutions and FEs of all the above nine engineering design problems are recorded in Table 16. As the Table 16 shows, the GPEAae can obtain the optimal solutions of eight engineering design problems. Especially, GPEAae has the lowest FEs in terms of three-bar truss and pressure vessel design problem, which means that GPEAae can achieve the competitive solution accuracy with the minimum computational overhead. When solving the rest seven problems, the FEs of GPEAae are also competitive among that of other compared algorithms. As for tension/compression spring and gear train design problems, the feasible solutions acquired from GPEAae and GPEAe are the same as most other compared algorithms. Although GPEAe holds nearly the same excellent performance with GPEAae on six out of the nine engineering design problems, its feasible solutions...
are worse than that of GPEAae on speed reducer, three-bar
truss and heat exchanger design problems. All in all, consid-
ering the overall performance on solution accuracy and com-
putational overhead, GPEAae behaves an extremely compet-
tive performance in solving these nine practical optimization
problems.

C. DISCUSSION ON EXPERIMENTAL RESULTS

Take the above statistical results on CEC2014 benchmark
functions and nine practical engineering problems into con-
sideration at the same time. The performance of GPEAae’s
search behavior is discussed from three aspects as follows.

• Discussion on the solution accuracy. As shown above,
the Best values can reveal the solution accuracy an
algorithm obtained when solving the objective func-
tions (on CEC2014 benchmark functions and the prac-
tical engineering problems). The smaller Best value
an algorithm obtains, the better solution accuracy
an algorithm reaches to near the theoretical opti-
mal solution. From Table 1 to Table 4, when com-
pared with that of GPEAe, PSO, and DE, the aver-
age ranking of GPEAae’s Best values rank first on all
10 and 30 dimensions CEC2014 benchmark functions.

Furthermore, the GPEAae also acquires the current opti-
mal solutions of eight engineering design problems.

• Discussion on the convergence rate. The FEs values
are recorded to reflect the computational overhead of
the compared algorithms, and the small computational
overhead also means fast convergence rate. In solving
the nine engineering design problems, two FEs values of
GPEAae rank first, six FEs values of GPEAae rank sec-
ond, and the rest ranks third. Moreover, the convergence
curves show the variation of the error values with the
increase of iterations, and the trend of curves also reflect
the convergence rate. As illustrated in Fig. 2 and 3,
GPEAae has competitive performance in convergence
rate compared with GPEAe, PSO, and DE.

• Discussion on the algorithmic robustness. The robust-
ness of an algorithm can be reflected by the Worst,
Mean and Std values, and the smaller these values also
mean higher robustness of the algorithm. As shown
in Table 1 to Table 4, when compared with GPEAe,
PSO, and DE, the average ranking of GPEAae’s Mean
and Std values rank second on all 10 and 30 dimen-
sions CEC2014 benchmark functions. Unfortunately,
the Worst, Mean and Std values acquired from GPEAae

TABLE 13. Comparison results on the cantilever beam design problem.

| Algorithms | MFO  | MMA  | GCA  | MOV  | CSA  | SOS  | GPEAe | GPEAae |
|------------|------|------|------|------|------|------|-------|--------|
| X1         | 5.984872 | 6.001000 | 6.010000 | 6.023940 | 6.008900 | 6.018780 | 6.021409 | 6.014808 |
| X2         | 5.316727 | 5.300000 | 5.300000 | 5.306011 | 5.304900 | 5.303440 | 5.309098 | 5.306724 |
| X3         | 4.497333 | 4.490000 | 4.490000 | 4.495011 | 4.502300 | 4.495870 | 4.492566 | 4.492323 |
| X4         | 3.513613 | 3.490000 | 3.490000 | 3.496022 | 3.507700 | 3.498960 | 3.496025 | 3.505168 |
| X5         | 2.161620 | 2.150000 | 2.150000 | 2.152726 | 2.153040 | 2.153640 | 2.154593 | 2.153781 |

Worst N.A N.A N.A N.A N.A N.A 1.339994 1.339982
Mean N.A N.A N.A N.A N.A 1.339970 1.339963 1.339961
Std N.A N.A N.A N.A N.A 0.000011 0.000007 0.000006
Best 1.339988 1.340000 1.340000 1.339960 1.339990 1.339960 1.339957 1.339957
FEs N.A N.A N.A N.A N.A 125000 15000 19920 19920

TABLE 14. Comparison results on the heat exchanger design problem.

| Algorithms | HIS  | PHIS | PIHS | GAs  | CRGA | GPEAe | GPEAae |
|------------|------|------|------|------|------|-------|--------|
| X1         | 500.000380 | 565.647600 | 598.195600 | N.A  | N.A  | 595.807017 | 576.280035 |
| X2         | 1359.310000 | 1000.000000 | 1285.399300 | N.A  | N.A  | 1345.948152 | 1355.50094 |
| X3         | 5197.959500 | 5543.542400 | 5167.706100 | N.A  | N.A  | 5107.752849 | 5117.424755 |
| X4         | 174.726300 | 180.865000 | 183.573200 | N.A  | N.A  | 183.380055 | 181.764537 |
| X5         | 292.081700 | 278.258300 | 293.294500 | N.A  | N.A  | 295.693338 | 295.302805 |
| X6         | 224.705400 | 219.134700 | 216.426000 | N.A  | N.A  | 216.620345 | 218.258663 |
| X7         | 282.646600 | 302.606600 | 290.274900 | N.A  | N.A  | 237.687117 | 286.462132 |
| X8         | 392.081700 | 378.258300 | 393.294500 | N.A  | N.A  | 395.693315 | 395.302905 |

Worst N.A N.A N.A N.A 8835.655000 | 8262.299821 | 7498.596216 |
Mean N.A N.A N.A N.A 7372.613000 | 7134.302667 | 7128.011367 |
Std N.A N.A N.A N.A 1000 105.704138 176.199479 |
Best 7057.274414 7109.190100 | 7051.301200 | 7060.221000 | 7054.316000 | 7049.308108 | 7049.253854 |
FEs N.A 200000 200000 320080 128000 100300 109340 |

TABLE 15. Comparison results on the tubular column design problem.

| Algorithms | X1  | X2  | Worst | Mean | Std | Best | FEs |
|------------|-----|-----|-------|------|-----|------|-----|
| CSA        | 5.451390 | 0.291960 | 26.539720 | 26.530400 | 0.001930 | 26.532170 | 15000 |
| CS-GA      | 5.451157 | 0.291966 | N.A   | N.A   | N.A   | 26.531330 | N.A |
| CS-PSO     | 5.451132 | 0.291974 | N.A   | N.A   | N.A   | 26.531690 | N.A |
| GPEAe      | 5.451156 | 0.291965 | 26.531312 | 26.531312 | 3.59K-15 | 26.531312 | 20400 |
| GPEAae     | 5.451156 | 0.291965 | 26.532191 | 26.531330 | 0.00124 | 26.531312 | 18300 |
TABLE 16. Total comparisons of best values and FEs on nine engineering design problems.

| F              | Smallest | Best of GPEAe | Best of GPEAae | Smallest FEs | FEs of GPEAe (Rank) | FEs of GPEAae |
|---------------|----------|---------------|---------------|--------------|---------------------|---------------|
| Three-bar truss | 263.895843 | 263.895712    | 263.895713    | 10000        | 7700                | 10000         |
| Pressure vessel | 6051.563900 | 6059.708025  | 6059.708025  | 80000       | 20280              | 21180         |
| Tension/Compression spring | 0.012665  | 0.012665     | 0.012665     | 7650        | 19480 (2)           | 19980         |
| Welded beam    | 1.724852  | 1.724851     | 1.724851     | 24000        | 35860 (2)          | 36100         |
| Speed reducer  | 2994.471066 | 2994.468240  | 2994.468307  | 6300        | 19480 (2)           | 20000         |
| Gear train     | 2.7E-12   | 2.7E-12       | 2.7E-12       | 60          | 1060 (4)           | 900           |
| Cantilever beam | 1.339960  | 1.339957     | 1.339957     | 15000       | 19920 (2)          | 19920         |
| Heat exchanger | 7051.301200 | 7049.254884  | 7049.508108  | 128000       | 109340 (2)         | 100300        |
| Tubular column | 26.531330 | 26.531312     | 26.531312     | 15000       | 18300 (2)          | 20400         |

behave mediocre when solving the nine engineering design problems. Compared with GPEAe, the GPEAae has better robustness, but it needs to be further improved.

The experimental results demonstrate that GPEAae has superior performances in both benchmark functions and practical applications, and shows great advantages compared with GPEAe.

V. CONCLUDING REMARKS

A new grey prediction evolution algorithm, which is abbreviated as GPEAae, is proposed in this paper. This paper first develops an accelerated first-order inverse accumulating generation operation (1-AIAGO), then based on the 1-AIAGO to create an accelerated even grey model (AEGM(1,1)). Finally, the GPEAae regards the population sequence as time series, superimposes and amplifies the weak effect produced by the proposed 1-AIAGO which replaces the predicted value with the corresponding original value. Since the 1-AIAGO calculates the predicted value one less time in each iteration, the computational complexity of the GPEAae is lower than that of the GPEAe. In CEC2014 benchmark functions experiments, the GPEAe is tested and compared with GPEAe, PSO, and DE. The experimental results indicate that the comprehensive performance of GPEAae is significantly better than GPEAe, PSO and DE. Specifically, the GPEAae outperforms the GPEAe on both solution accuracy and convergence rate. In nine engineering constrained design experiments, the GPEAae can reach all the current best feasible solutions, and has competitive computational overhead compared with GPEAe and the state-of-the-art algorithms.

Along the improvement ideas of GPEAae, some other effective grey models such as grey Bernoulli model [17], fractional grey model [51], etc, could also be utilized to construct more GPEA in the future. In addition, some effective evolutionary approaches such as parameter control, operator hybridization, adaptive strategy, etc, could be employed to improve the search behavior of GPEAae. In future research, it is also promising to explore GPEAae’s wider application scenarios and suitability for various problems. With the created novel AEGM(1,1) as a bridge, the GPEAae can be further studied by researchers in both optimization field and grey system field.

REFERENCES

[1] Z. Hu, X. Xu, Q. Su, H. Zhu, and J. Guo, “Grey prediction evolution algorithm for global optimization,” Appl. Math. Model., vol. 79, pp. 145–160, Mar. 2020.
[2] S. Liu, J. Forrest, and Y. Yang, “A brief introduction to grey systems theory,” Grey Syst., Theory Appl., vol. 2, no. 2, pp. 89–104, Aug. 2012.
[3] J. Cui, S.-F. Liu, B. Zeng, and N.-M. Xie, “A novel grey forecasting model and its optimization,” Appl. Math. Model., vol. 37, no. 6, pp. 4399–4406, Mar. 2013.
[4] S. Liu and J. Y. L. Forrest, “Advances in grey systems research,” J. Grey Syst., vol. 25, no. 2, pp. 1–18, 2013.
[5] L. Wu, S. Liu, L. Yao, S. Yan, and D. Liu, “Grey system model with the fractional order accumulation,” Commun. Nonlinear Sci. Numer. Simul., vol. 18, no. 7, pp. 1775–1785, Jul. 2013.
[6] N.-M. Xie, S.-F. Liu, Y.-J. Yang, and C.-Q. Yuan, “On novel grey forecasting model based on non-homogeneous index sequence,” Appl. Math. Model., vol. 37, no. 7, pp. 5059–5068, Apr. 2013.
[7] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.
[8] R. Storn and K. Price, “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,” J. Global Optim., vol. 11, no. 4, pp. 341–359, 1997.
[9] I. Boussaïd, J. Lepagnot, and P. Siarry, “A survey on optimization metaheuristics,” Inf. Sci., vol. 237, pp. 82–117, Jul. 2013.
[10] H. Wang, Z. Hu, Y. Sun, Q. Su, and X. Xia, “A novel modified BSA inspired by species evolution rule and simulated annealing principle for constrained engineering optimization problems,” Neural Comput. Appl., vol. 31, no. 8, pp. 4157–4184, Aug. 2019.
[11] Z. Li, Z. Hu, Y. Miao, Z. Xiong, X. Xu, and C. Dai, “Deep-mining backtracking search optimization algorithm guided by collective wisdom,” Math. Problems Eng., vol. 2019, pp. 1–30, Dec. 2019.
[12] X. Yang, Z. Fang, Y. Deng, D. Mba, and X. Li, “A novel multi-information fusion grey model and its application in wear trend prediction of wind turbines,” Appl. Math. Model., vol. 71, pp. 543–557, Jul. 2019.
[13] C.-C. Hsu and C.-Y. Chen, “Applications of improved grey prediction model for power demand forecasting,” Energy Convers. Manage., vol. 44, no. 14, pp. 2241–2249, Aug. 2003.
[14] L. Zeng, “A fractional order opposite-direction accumulative grey prediction model with time-power,” J. Grey Syst., vol. 31, no. 3, pp. 90–104, 2019.
[15] N. Xie and R. Wang, “A historic review of grey forecasting models,” J. Grey Syst., vol. 29, no. 4, p. 29, 2017.
[16] B. Zeng, H. Duan, and Y. Zhou, “A new multivariable grey prediction model with structure compatibility,” Appl. Math. Model., vol. 75, pp. 385–397, Nov. 2019.
[17] L. Kong and X. Ma, “Comparison study on the nonlinear parameter optimization of nonlinear grey Bernoulli model (NGBM (1, 1)) between intelligent optimizers,” Grey Syst., Theory Appl., vol. 8, no. 2, pp. 210–226, Apr. 2018.
[18] B. Zeng, M. Tong, and X. Ma, “A new-structure grey Verhulst model: Development and performance comparison,” Appl. Math. Model., vol. 81, pp. 522–537, May 2020.
[19] Z. He, Y. Shen, J. Li, and Y. Wang, “Regularized multivariable grey model for stable grey coefficients estimation,” Expert Syst. Appl., vol. 42, no. 4, pp. 1806–1815, Mar. 2015.
[20] X. Xu, Z. Hu, Q. Su, Y. Li, and J. Dai, “Multivariable grey prediction evolution algorithm: A new Metaheuristic,” Appl. Soft Comput., vol. 89, Apr. 2020, Art. no. 106086.
[21] J. Liang, Z. Hu, Z. Li, Z. Xiong, and Q. Su, “An improved grey prediction evolution algorithm based on topological opposition-based learning,” IEEE Access, vol. 8, pp. 30745–30762, 2020.

[22] Y. Wang, Z. Cai, Y. Zhou, and Z. Fan, “Constrained optimization based method for mixed integer discrete continuous optimization and its applications to mechanical design,” J. Mech. Design, vol. 116, no. 2, pp. 193–203, Jul. 2004.

[23] S. Mao, M. Gao, X. Xiao, and M. Zhu, “A novel fractional grey system model and its application,” Appl. Math. Model., vol. 40, nos. 7–8, pp. 5063–5076, Apr. 2016.

[24] H. Chickermane and H. C. Gea, “Structural optimization using a new local approximation method,” Int. J. Numer. Methods Eng., vol. 39, no. 5, pp. 829–846, Mar. 1996.

[25] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: A nature-inspired algorithm for global optimization,” Neural Comput. Appl., vol. 27, no. 2, pp. 495–513, Feb. 2016.

[26] M.-Y. Cheng and D. Prayogo, “Symbiotic organisms search: A new Meta-heuristic optimization algorithm,” Comput. Struct., vol. 139, pp. 98–112, Jul. 2014.

[27] Z. Hu, Z. Li, C. Dai, X. Xu, Z. Xiong, and Q. Su, “Multiobjective grey prediction evolution algorithm for environmental/economic dispatch problem,” IEEE Access, vol. 8, pp. 84162–84176, 2020.

[28] S. Mirjalili, “The ant lion optimizer,” Adv. Eng. Softw., vol. 83, pp. 80–98, May 2015.

[29] C. Gao, “A novel Metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm,” Comput. Struct., vol. 169, pp. 1–12, Jun. 2016.

[30] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on evolutionary and swarm intelligence algorithms,” Swarm Evol. Comput., vol. 1, no. 1, pp. 3–18, Mar. 2011.

[31] Q. He and L. Wang, “A hybrid particle swarm optimization with a stochastic selection for constrained optimization,” Struct. Multidisciplinary Optim., vol. 41, no. 6, pp. 947–963, Jun. 2010.

[32] H. Liu, Z. Cai, and Y. Wang, “Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization,” Appl. Soft Comput., vol. 10, no. 2, pp. 629–640, Mar. 2010.

[33] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, “Mine blast engineering optimization problems,” Appl. Soft Comput., vol. 13, no. 5, pp. 2592–2612, May 2013.

[34] L. Wang and L.-P. Li, “An effective differential evolution with level comparison for constrained engineering design,” Struct. Multidisciplinary Optim., vol. 41, no. 6, pp. 497–513, Jun. 2010.

[35] Y. Wang, Z. Cai, Y. Zhou, and Z. Fan, “Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique,” Struct. Multidisciplinary Optim., vol. 37, no. 4, pp. 395–413, Jun. 2008.

[36] B. K. Kannan and S. N. Kramer, “An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design,” J. Mech. Design, vol. 116, no. 2, pp. 405–411, Jun. 1994.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Swarm, Evolutionary, and Memetic Computing. Berlin, Germany: Springer, 2015, pp. 260–273.

[38] A. Amirjanov, “The development of a changing range genetic algorithm,” Comput. Methods Appl. Mech. Eng., vol. 195, nos. 19–22, pp. 2495–2508, Apr. 2006.

[39] G. Kanagaraj, S. Ponnambalam, and A. H. Gandomi, “Hybridizing cuckoo search with bio-inspired algorithms for constrained optimization problems,” in Swarm, Evolutionary, and Memetic Computing. Berlin, Germany: Springer, 2015, pp. 260–273.

[40] A. Askarzadeh, “A novel Metaheuristic method for solving constrained optimization problems,” Comput. Struct., vol. 169, pp. 1–12, Jun. 2016.

[41] A. S. Lee and Z. W. Geem, “A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice,” Comput. Methods Appl. Mech. Eng., vol. 194, nos. 36–38, pp. 3902–3933, Sep. 2005.

[42] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: A nature-inspired algorithm for global optimization,” Neural Comput. Appl., vol. 27, no. 2, pp. 495–513, Feb. 2016.

[43] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” Knowl.-Based Syst., vol. 89, pp. 228–249, Nov. 2015.