Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2846181, 14 pages
https://doi.org/10.1155/2020/2846181

Research Article

Application of the Improved PSO-Based Extended Domain Method in Engineering

Bin Bai,[1,2] **Zhi-wei Guo**,[3] **Qi-liang Wu**,[4] **Junyi Zhang**,[1,2] **and Yan-chao Cui**[5]

[1]*State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300401, China*
[2]*School of Mechanical Engineering, Hebei University of Technology, Tianjin 300401, China*
[3]*Shenyang Engine Research Institute, Shenyang 110015, China*
[4]*School of Electrical Engineering and Automation, Tiangong University, Tianjin 300387, China*
[5]*AVIC Tianjin Aviation Electromechanical Co., Ltd., Tianjin 300308, China*

Correspondence should be addressed to Yan-chao Cui; duji2008@126.com

Received 21 May 2020; Revised 22 July 2020; Accepted 29 July 2020; Published 7 September 2020

Guest Editor: Jing Guo

The standard particle swarm optimization (PSO) algorithm is the boundary constraints of simple variables, which can hardly be directly applied in the constrained optimization. Furthermore, the standard PSO algorithm often fails to obtain the global optimal solution when the dimensionality is high for unconstrained optimization. Thus, an improved PSO-based extended domain method (IPSO-EDM) is proposed to solve engineering optimization problems. The core idea of this method is that the original feasible region is expanded in the constrained optimization which is transformed into the unconstrained optimization by combining the ergodicity of chaos optimization and the evolutionary variation to realize global search. In addition, to verify the effectiveness of the IPSO-EDM, an unconstrained optimization case study, four constrained optimization case studies, and one engineering example are investigated. The results indicate that the computational accuracy of the IPSO-EDM is comparable to that provided by the existing literature, and the computational efficiency of the IPSO-EDM is significantly improved. Meanwhile, this method has conspicuous global search ability and stability in engineering optimization.

1. Introduction

Optimization began in the 17th century, which originated from differential and integral calculus invented by Newton and Leibnitz. Then, optimization algorithms [1–5] were rapidly developed, such as artificial neural network, simulated annealing, genetic algorithm, ant colony optimization, and particle swarm optimization (PSO). All these methods were widely used in different fields [6–12], such as chemical engineering, biomedicine, navigation, robot, automobile, architecture, and aerospace.

Actually, the mechanical engineering optimization can be expressed as continuity interval constraint optimization. To investigate this problem, some traditional gradient methodologies [13–15] were investigated such as the penalty function and Lagrange multiplier. Although the theory of these methods is impeccable, the objective function and the constraint condition must be differentiable. However, the constraint function and objective function are nondifferentiable and discontinuous implicit functions in practical engineering. Consequently, a new optimization method was developed to study this problem, which plays an important role for the development of engineering optimization design.

Initially, the PSO was presented by Kennedy and Eberhart [16] to investigate the flight behaviour of birds, which was termed as the global PSO algorithm. This method has been extended to different kinds of fields. For instance, Xue et al. [17] used analytical method with modified PSO to establish the subdomain model and optimize cogging torque. Han et al. [18] adopted an adaptive gradient multiobjective PSO to improve the computational

performance for a mechanism. Yi et al. [19] presented a parallel chaotic local search algorithm to solve constrained engineering design problems. Park et al. [20] used chaotic sequences with conventional linearly decreasing inertia weights to increase the exploitation capability. However, the essence of these methods is not improved, thus some scholars renewed the equation of the global PSO. For instance, Phung et al. [21] proposed a discrete PSO algorithm to solve the extended travelling salesman problem for robotic inspection. Wang and Zhang [22] presented an optimization method to describe a planar parallel 3-DOF nano positioner in modelling. Nickabadi et al. [23] successfully regarded the adaptive inertia weight factor as the feedback parameter to ascertain the situation of the particles in the search space. Mojarrad and Nayeripour [24] used a fuzzy adaptive PSO to solve the non-convex economic dispatch problems. Khan et al. [25] proposed a modified PSO algorithm to avoid the premature convergence and strengthen its robustness. This algorithm can adaptively update parameters to keep the diversity of the swarm. However, the convergence speed of inertia weight method is not very fast, so another method is developed. Liang et al. [26] proposed a fuzzy multilevel algorithm-based PSO to optimize support vector regression machine and realize a fuzzy multilevel drilling leak risk evaluation system. Tian et al. [27] utilized sigmoid-based acceleration coefficients to avoid premature convergence and entrapment into local optima when they handled complex multimodal problems. Hsieh et al. [28] developed a discrete cooperative coevolving PSO algorithm to study the influence of detour distance constraints on the carpooling performance. Zahara et al. [29, 30] put forward the hybrid Nelder-Mead-PSO algorithm for unconstrained optimization, and then they presented embedded constraint handling methods for dealing with constraints. Liu et al. [31] adopted a bottleneck objective learning strategy for many-objective optimization to improve convergence on all objectives. Xia et al. [32] proposed a triple archives PSO to deal with the selecting proper exemplars and designing an efficient learning model. Wang et al. [33] investigated the evolutionary computation community for large-scale optimization.

The above investigations on the PSO algorithm indicate that some constrained points are not located in the feasible region but outside the feasible region; however, these excluded constrained points are closer to the boundary than the points located in the feasible region. Obviously, this is unreasonable. Based on the above research studies, a new methodology named improved particle swarm optimization-based extended domain method (IPSO-EDM) is proposed to investigate engineering optimization.

In the following, Section 2 describes the basic theory of the IPSO-EDM, including standard PSO, improved PSO, and the algorithm principle. Section3 gives an unconstrained optimization case study, four numerical constrained optimization case studies, and one engineering case study to verify the effectiveness of the IPSO-EDM. Section 4 gives the conclusions.

2. Improved Particle Swarm Optimization

2.1. Standard PSO. Assuming that the particle swarm is composed of M particles in an N-dimensional space, the position of the ith particle of the kth iteration is expressed as $\mathbf{X}_i(k) = (x_{i1}(k), x_{i2}(k), \ldots, x_{iN}(k))^T$, the flight speed is described as $\mathbf{V}_i(k) = (v_{i1}(k), v_{i2}(k), \ldots, v_{iN}(k))^T$, the local optimal position is $\mathbf{P}_i = (p_{i1}, p_{i2}, \ldots, p_{iN})^T$, and the global optimal location is $\mathbf{P}_g = (p_{g1}, p_{g2}, \ldots, p_{gN})^T$. Each particle updates its speed and position according to the following equation, which are expressed as

$$\begin{aligned} v_{ij}(k+1) &= v_{ij}(k) + c_1 r_{1j} \times \left[p_{ij}(k) - x_{ij}(k)\right] + c_2 r_{2j} \\ &\quad \times \left[p_{gj}(k) - x_{ij}(k)\right], \\ x_{ij}(k+1) &= x_{ij}(k) + v_{ij}(k+1), \end{aligned} \tag{1}$$

where j is the jth component of the ith particle, c_1 and c_2 are positive constants and they are called learning factor, c_1 adjusts the step length for particles to their optimal location, and c_2 adjusts the step length for particles to the global optimal position, and r_{1j} and r_{2j} are, respectively, random numbers that obey a uniform distribution, and their values are within $[0, 1]$.

To prevent particles from flying out of the search space in the optimization, the velocity and position are limited as $v_{ij} \in [v_{j\min}, v_{j\max}]$ and $x_{ij} \in [x_{j\min}, x_{j\max}]$. Meanwhile, the inertia weight w is involved. This method is termed as the standard PSO algorithm. The update equations are expressed as

$$\begin{aligned} v_{ij}(k+1) &= w(k) \times v_{ij}(k) + c_1 r_{1j} \times \left[p_{ij}(k) - x_{ij}(k)\right] \\ &\quad + c_2 r_{2j} \times \left[p_{gj}(k) - x_{ij}(k)\right], \\ x_{ij}(k+1) &= x_{ij}(k) + v_{ij}(k+1). \end{aligned} \tag{2}$$

The standard PSO does not have too many requirements on the objective function and constraint function, which can conduct constraint optimization, and these constraints limit the range of each variable interval value such as $x_j \in [x_{j\min}, x_{j\max}]$. This is different from the gradient method, but it becomes more complicated if there are equality or inequality constraints in the optimization. One obvious reason is that the feasible region has changed from a hypercube to a less regular region, and the variables are not independent on each other. For this reason, the standard PSO must be improved to deal with the constraints.

2.2. Improved PSO

2.2.1. Constraint Methods of PSO. At present, there are four typical methods to deal with constraints:

(1) Discriminant function method: inequality and equality constraints are used as discriminant functions to determine whether the search points are feasible points in the optimization process. It will be discarded or modified to be a feasible point during search if the search point is not a feasible point. Thus,

this method has strict restrictions to the search point, and it is very difficult to generate the initial feasible point when the feasible region composed by equality and inequality constraints is small.

(2) Penalty function method: the optimization and constraint functions are combined to form the penalty function. The original constrained optimization with equality and inequality constraints has become an unconstrained optimization with the penalty function. However, the disadvantage of this method is that the penalty factor must be chosen correctly; otherwise, it can hardly obtain the optimal solution.

(3) Multiobjective optimization method: optimization and constraint functions are, respectively, used as the new optimization targets. However, solving the multiobjective optimization is more difficult than solving a single-objective optimization in many cases.

(4) "Competitive selection" method: deals with the feasible particles (the design points represented by particles satisfy all constraint requirements) and infeasible particles (some or all design points represented by particles dissatisfy the optimization constraint requirements) in PSO. However, it is not appropriate to deem that feasible particles are superior to infeasible ones in competitive selection, and the infeasible region can be extended into a feasible region to find the optimal point.

2.2.2. Core Idea of the "Competitive Selection" Constraint. The "competitive selection" constraint can be summarized in three aspects:

(1) All feasible particles are superior to infeasible particles

(2) The particle with a better objective function is selected for two feasible particles

(3) The advantages and disadvantages of the particle are judged according to the degree of constraint violation for two infeasible particles; the lesser the degree of the constraint, the larger the violation

In Figure 1(a), unconstrained optimal point A is in the feasible region, and the constraint is useless to the whole optimization; essentially, the constrained optimal point is the unconstrained optimal point. However, the unconstrained optimal point is not located in the feasible region but at its boundary for some optimization. For example, point B is the constrained optimal point, point C is in the feasible region, and point D is outside the feasible region in Figure 1(b). According to the competitive selection, point C is superior to D. However, point D is closer to B than C; thus, the optimal information provided by D is superior to C. As a result, point D is labelled as suboptimal, which is not appropriate. So, the algorithm must be improved. Figure 1(c) shows an improved method, which expands the original feasible region to include points such as D which is unfeasible but close to the feasible region and can provide better function information and is taken as the feasible point. This methodology is termed as extended domain method (EDM).

In addition, the new speed depends on the current speed in the updated formula for standard PSO. However, the algorithm is random, and it is impossible to predict and control the size of particle speed. To solve this problem, a control variable ξ is introduced in the extended domain, and the result is shown in Figure 2,where $\mathbf{x}^*$ is the optimal location, $\mathbf{x}$ is the current particle position, $\mathbf{p}_g$ is the optimal historical position of the particle swarm, $\mathbf{p}_x$ is the optimal historical position of the current particle, $\widehat{\mathbf{x}}$ is the optimal location of the current particle swarm, and $\mathbf{v}_l$ and $\mathbf{v}_s$ are, respectively, the largest and smallest speeds.

Figure 2 shows that the particle with higher speed may miss a better position, while the particle with smaller speed improves the position, but it is not the optimal position. Thus, the current particle speed is expected to be controlled. One method is to use current location information of the particle swarm to determine its speed. The difference value between the particle swarm and the particle in the current optimal position is used to determine current particle velocity. The results indicate that the position $\mathbf{x}_{\widetilde{x}}$ determined using this method is better than the positions $\mathbf{x}_l$ and $\mathbf{x}_s$. In fact, the constrained optimization is transformed into an unconstrained optimization by the EDM, so a scientific and reasonable unconstrained optimization method needs to be developed.

2.2.3. Unconstrained PSO. Firstly, the initial particle swarm position and velocity are determined in a random way in standard PSO. Generally, the designers expect the particles in the particle swarm space can better reflect the information which is studied in the initial state. One of the direct ways is to generate many particles which fill in the whole search space. However, this will consume a lot of computational resources in subsequent iterations. The PSO algorithm will lose its characteristics of group cooperation if the number of particles in the particle swarm space is too small, which is meaningless. Usually, the particle swarm with dozens of particles can solve the complex optimization problem, but random arrangements of particles often result in a "cluster" in an area. To enable the particles to be relatively "evenly" distributed in the search space, the uniform design method based on statistical theory is used to initialize the particle swarm space. The random distribution and uniform distribution of the initial particle swarm space are shown in Figure 3.

Secondly, the standard PSO algorithm itself cannot obtain global information of the objective function, and it is easy to fall into the local optimal. Generally, the particle can jump out of the local optimal and give new global information by dynamically adjusting the inertia factor, but simply adjusting the inertia factor is not enough for the complex multipeak function. To handle this problem, the ergodicity of chaos optimization [22] is combined with evolutionary variation to realize global search. In this

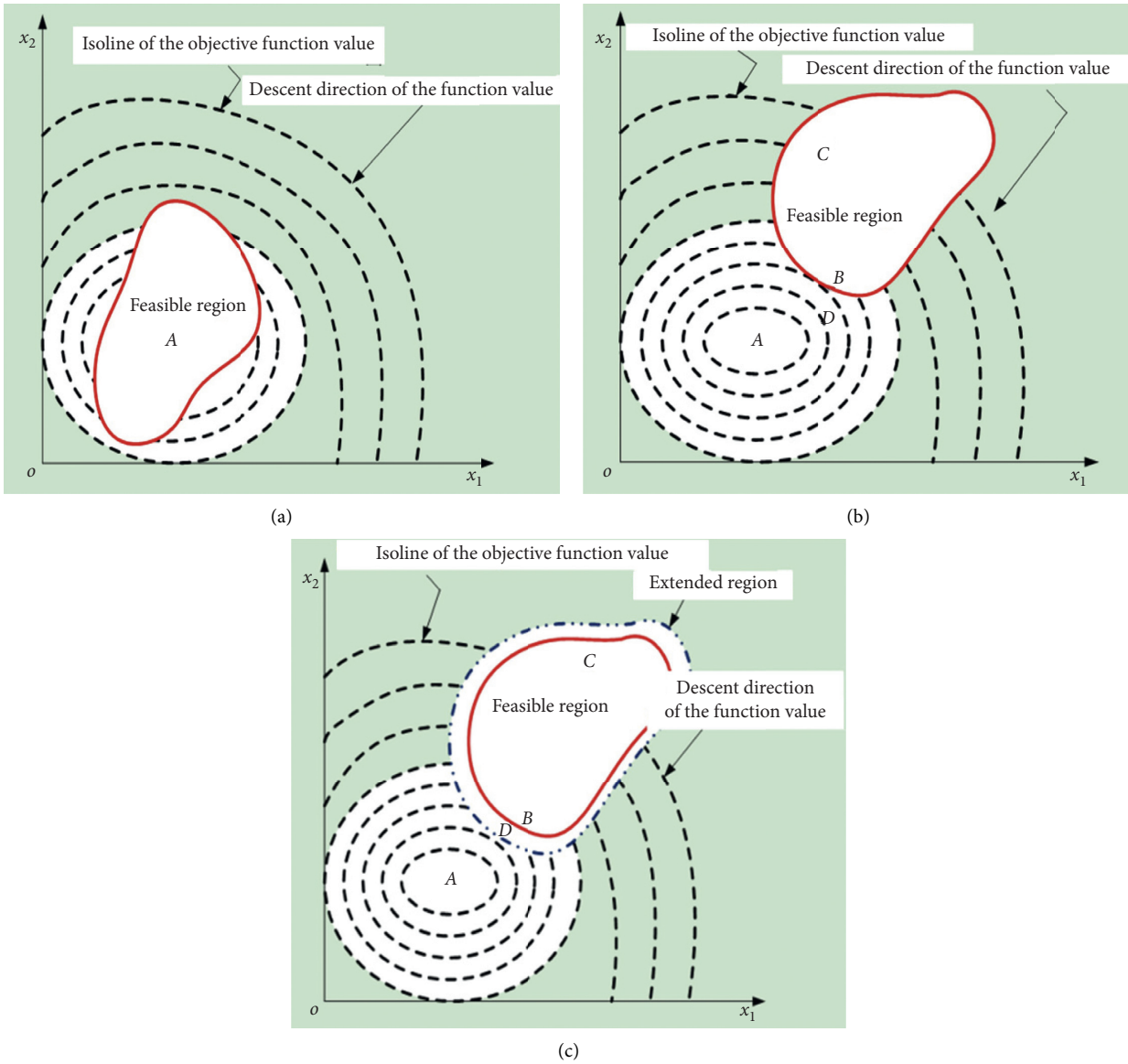

FIGURE 1: Relationship among the optimal point, feasible domain, and extended domain: (a) optimal point in the feasible region; (b) optimal point outside the feasible region; (c) optimal point in the extended domain.

method, logistic chaotic system equation is applied in the PSO algorithm and obeys Chebyshev distribution, as shown in Figure 4.

Figure 4 indicates that the middle value of the logistic chaotic sequence is relatively uniform, while the probability of both sides is relatively large. It means that the chance of finding the global optimal point will be reduced using the logistic chaotic sequence if the global optimal point is not at ends of the design variable. Therefore, it is very necessary to find a chaotic sequence, which can not only maintain ergodicity but also keep uniform statistical distribution. According to this problem, evolutionary variation strategy is developed. Generally, this strategy introduces a mutation operator to change the design variable. This method can help the particle swarm escape local optimal and maintain its overall vitality and prevent the particle swarm from falling into the condition of "precocity" at the earlier iteration. Based on the above research, the idea of uniform design, chaos optimization, and evolutionary variation is introduced into unconstrained PSO, and this method is improved to realize global search and local search.

Firstly, the uniform design of the Halton sequence [34] is adopted to initialize the particle swarm. Assume that the particle position scope of the jth design variable is $[x_{j\min}, x_{j\max}]$, whose component values of the ith particle are h_{ij}, and the position x_{ij} of the initialized particle swarm is expressed as

$$x_{ij}(0) = x_{j\min} + \left(x_{j\max} - x_{j\min}\right) \times h_{ij}. \tag{3}$$

Similarly, the velocity v_{ij} for the jth component values of the ith particle is expressed as

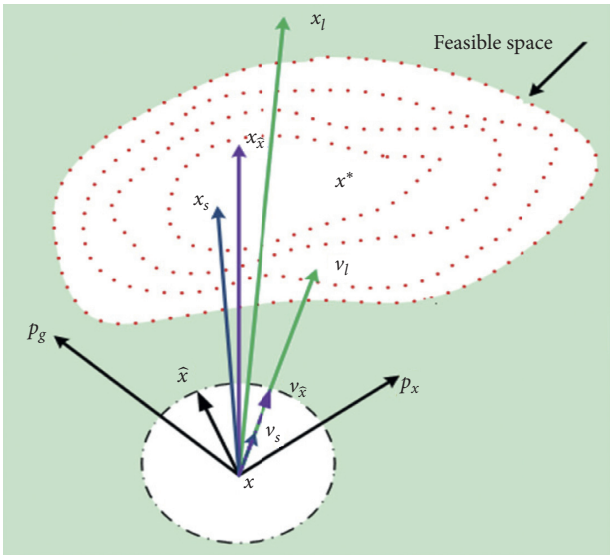

Figure 2: Renewal schematic diagram of the particle position.

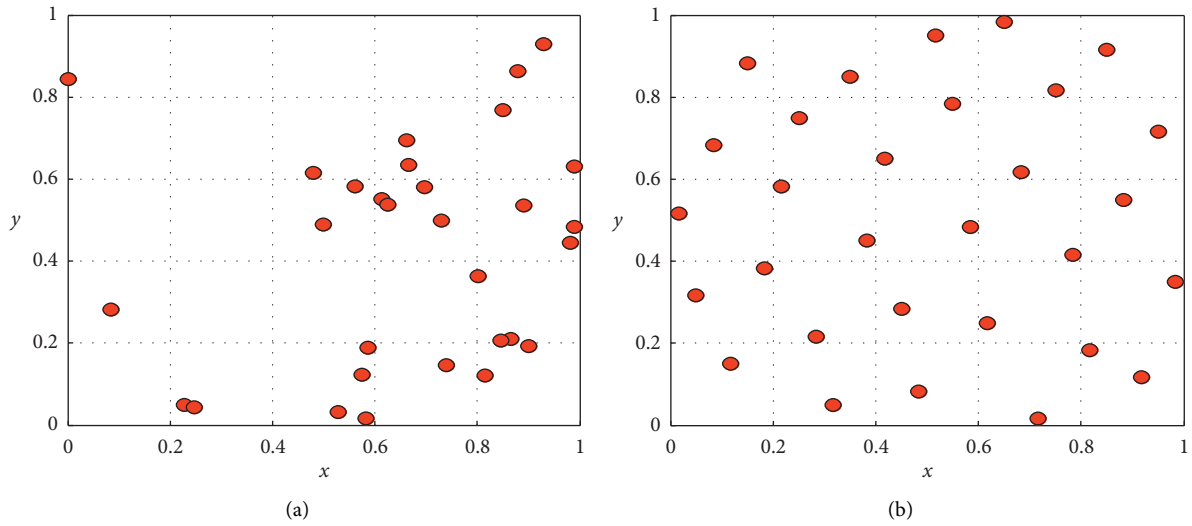

Figure 3: Distribution diagram of the initial particle swarm space: (a) random distribution; (b) uniform distribution.

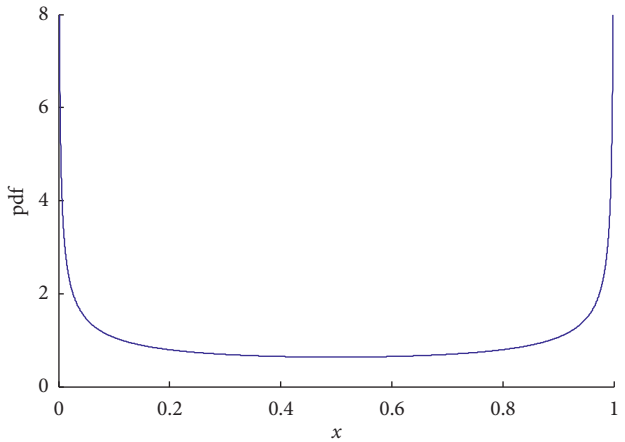

Figure 4: Probability density curve of the logistic chaotic sequence.

$$v_{ij}(0) = v_{j\min} + \left(v_{j\max} - v_{j\min}\right) \times h_{ij}. \tag{4}$$

In the process of particle swarm evolution, the points around the optimal position still need to be searched to get a better position when the optimal position of the particle swarm is found. This is called as the chaos optimization search method, which is expressed as

$$z(l+1) = 1 - \mu \times z(l)^2. \tag{5}$$

However, the chaotic sequence generated by equation (5) is not uniform in statistics; therefore, the chaotic sequence needs to be transformed as follows:

$$t(l) = \frac{\arccos[z(l)]}{\pi}. \tag{6}$$

Equation (6) not only satisfies the ergodicity of the chaotic sequence but also complies with the uniform distribution in [0, 1]. The uniform logistic chaotic sequence and original logistic chaotic sequence of the frequency curve and ergodic graph are shown in Figures 5 and 6. The point set of the chaotic sequence is 1*e*4.

The optimal position of the particle swarm is denoted as $\mathbf{P}_g(k)$ after the kth evolution, and the ith component is denoted as $\mathbf{p}_{gi}(k)$; then, the initial value of the chaotic sequence is expressed as

$$z_i(0) = \frac{p_{gi}(k) - x_{i\min}}{x_{i\max} - x_{i\min}}. \tag{7}$$

According to equations (5) and (6), the chaotic sequence $z_i(l)$, $l = 1, 2, \ldots, m$, is generated, and its coordinate value corresponding to the original space is obtained via reverse transformation, which is expressed as

$$\overline{p}_i(l) = x_{i\min} + \left(x_{i\max} - x_{i\min}\right) \times z_i(l), \tag{8}$$

where l is the number of iterations of the chaotic sequence and k is the number of iterations.

However, the chaos optimization can hardly make particle swarm get rid of local optimal, and the evolutionary variation is used to help it jump out local optimal. The mutation operator plays a key role in evolutionary variation. The Gaussian operator and Cauchy operator are two commonly used mutation operators. The global searching ability of the Cauchy mutation is stronger than that of the Gaussian mutation, but Cauchy mutation may produce large stride length in search, which means its local search ability is not as good as that of the Gaussian mutation. Therefore, according to the characteristics of Cauchy variation and Gaussian variation, the "coarse tune" of the particle swarm is obtained through the Cauchy mutation at first, and then its "fine tune" is obtained through the Gaussian mutation. The mutation formula is described as

$$z_k^* = z_k + \beta^{(g-1)} \times r, \tag{9}$$

where z_k and z_k^* represent the value of the kth design variable before and after mutation, respectively; r is a random number; β is the contraction coefficient and $\beta = 0.1$; and g is the number of mutations.

The design variable value can hardly exceed its interval after mutation because of its range limitation in optimization process. The design variable value will be mutated again when its value is not in the definition domain after mutation until its solution is convergence. The maximum number of mutations is stipulated as q when the mutation value still does not satisfy the value range in the procedure. Then, the mutation can be judged according to the absolute value of the difference between the mutation value and interval endpoint value, namely, the mutation value is the left endpoint value and vice versa if the variation value is close to the left endpoint.

2.2.4. Algorithm Principle. Generally, the optimization model is described as

$$\begin{aligned} \min \quad & f(x) \\ & g_i(x) \le 0, && i = 1, 2, \ldots, N \\ \text{s.t} \quad & h_j(x) = 0, && j = 1, 2, \ldots, M, \\ & x_k \in [x_{k\min}, x_{k\max}], && k = 1, 2, \ldots, n \end{aligned} \tag{10}$$

where $f(\cdot)$ is the objective function; $g(\cdot)$ is the inequality constraint function; and $h(\cdot)$ is the equality constraint function.

To divide the particle swarm space into the feasible domain and unfeasible domain, a constraint conflict function Vio (x) is constructed, which is defined as

$$\mathrm{Vio}(x) = \sum_{i=1}^{N} \max\left[g_i(x), 0\right] + \sum_{j=1}^{M} \left|h_j(x)\right|. \tag{11}$$

According to equation (11), an arbitrary feasible point satisfies Vio $(x) = 0$, and an arbitrary unfeasible point satisfies Vio $(x) > 0$; meanwhile, Vio (x) can also describe the degree of constraint violation of the unfeasible point.

To control the size of the extended domain and particle speed, the control variable ξ is defined in the following four situations for any two given search points x_1 and x_2:

(1) x_1 is superior to x_2 when x_1 and x_2 are both in the extended domain, i.e., $\mathrm{Vio}(\mathbf{x}_1) < \xi$, $\mathrm{Vio}(\mathbf{x}_2) < \xi$ and $f(\mathbf{x}_1) < f(\mathbf{x}_2)$

(2) x_1 is superior to x_2 when $\mathrm{Vio}(\mathbf{x}_1) = \mathrm{Vio}(\mathbf{x}_2)$ and $f(\mathbf{x}_1) < f(\mathbf{x}_2)$

(3) x_1 is superior to x_2 when x_1 and x_2 are both outside the extended domain, i.e., $\mathrm{Vio}(\mathbf{x}_1) > \xi$, $\mathrm{Vio}(\mathbf{x}_2) > \xi$ and $\mathrm{Vio}(\mathbf{x}_1) < \mathrm{Vio}(\mathbf{x}_2)$

(4) x_1 is superior to x_2 when x_1 is in the extended domain, $\mathrm{Vio}(\mathbf{x}_1) < \xi$, and x_2 is outside the extended domain, $\mathrm{Vio}(\mathbf{x}_2) > \xi$

The control variable ξ of the extended domain is gradually changing with particle swarm evolution, and the strategy is expressed as

$$\xi(k) = \xi(0) \times \exp(-\beta \times k), \tag{12}$$

where k is particle swarm evolution algebra and $\xi(0)$ is the initial control variable of the extended domain, which is defined as

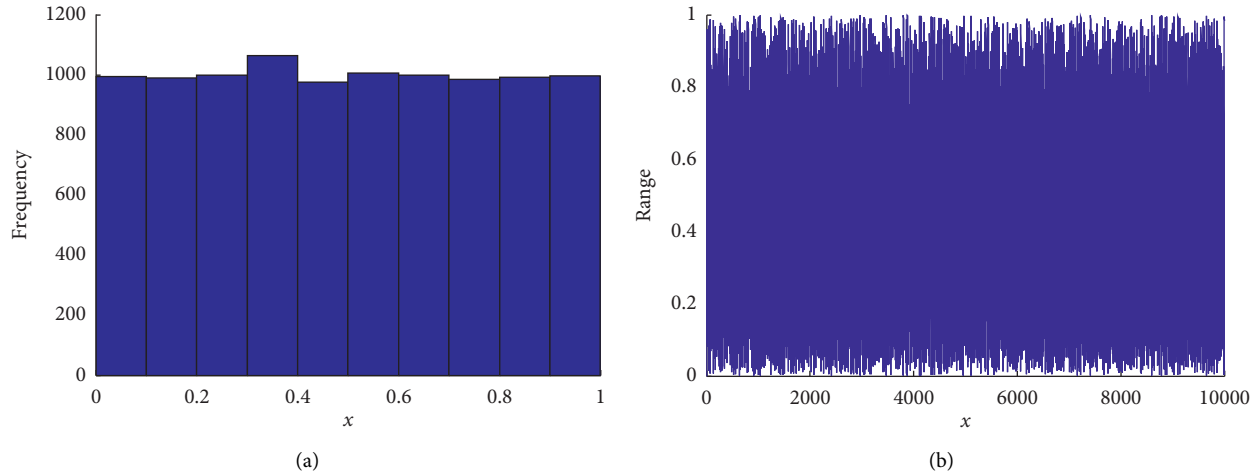

Figure 5: Uniform logistic chaotic sequence: (a) frequency curve; (b) ergodic graph.

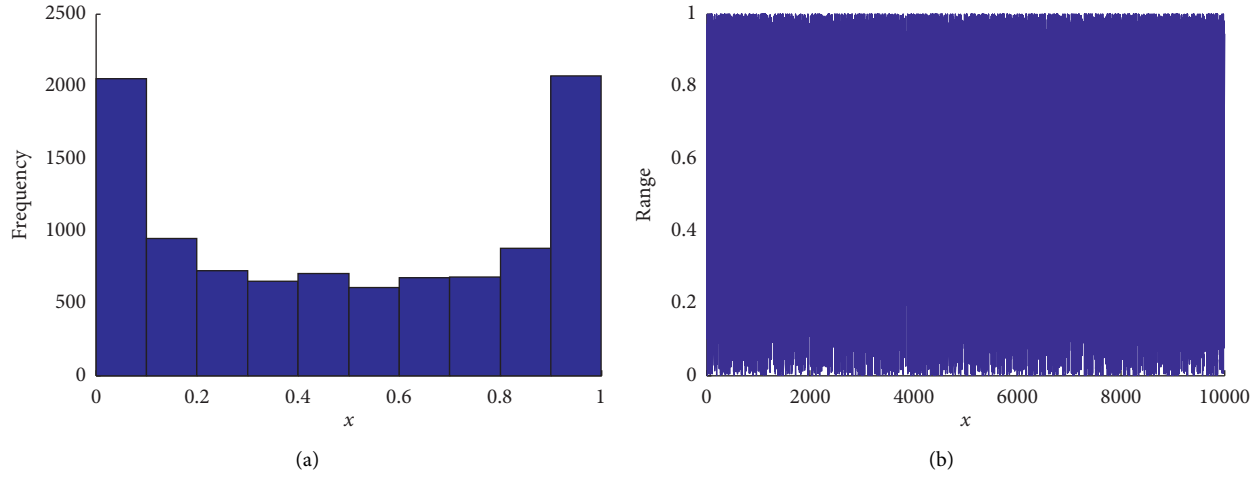

Figure 6: Original logistic chaotic sequence: (a) frequency curve; (b) ergodic graph.

$$\xi(0) = \frac{1}{2}\left(\frac{1}{N}\sum_{i=1}^{N}\mathrm{Vio}(x_i) + \min_{x\in\{x_0\}}\mathrm{Vio}(x)\right), \tag{13}$$

where $\{x_0\}$ is the set composed by the initial particle swarm.

When $\{x_0\}$ evolves to the kth generation, stipulating $\xi(K) = \xi_K$, β can be expressed as

$$\beta = \frac{\log(\xi_0/\xi_K)}{K}. \tag{14}$$

The control variable ξ can be expressed as

$$\xi(k) = \begin{cases} \xi(0) \times \exp(-\beta \times k), & 1 \le k \le K, \\ 0, & K \le k \le k_{\max}. \end{cases} \tag{15}$$

To make the particle swarm fast gather to the optimal point in the direction, its update strategy is written as

$$\begin{aligned} v_{ij}(k+1) &= w(k) \times \left|\hat{x}_j(k) - x_{ij}(k)\right| \times \mathrm{sgn}\left(v_{ij}\right) + r \\ &\quad \times \left[p_{ij}(k) - x_{ij}(k)\right] + (1-r) \\ &\quad \times \left[p_{gj}(k) - x_{ij}(k)\right], \\ x_{ij}(k+1) &= x_{ij}(k) + v_{ij}(k+1), \end{aligned} \tag{16}$$

where $\hat{x}$ is the optimal position of particles in the current generation; $|\hat{x}_j(k) - x_{ij}(k)|$ controls step length; and $\mathrm{sgn}(v_{ij})$ determines the motion direction of the particle.

The following limit policy is adopted when the particle swarm crosses the border in the process of updating, which is denoted as

$$\begin{aligned} x_{ij}(k) &= \overline{x}_j(k) + r \times \left[x_{j\min} - \overline{x}_j(k)\right], \quad x_{ij}(k) < x_{j\min}, \\ x_{ij}(k) &= \overline{x}_j(k) + r \times \left[x_{j\max} - \overline{x}_j(k)\right], \quad x_{ij}(k) > x_{j\max}, \end{aligned} \tag{17}$$

where r is the uniform-distributed random number and $\overline{x}$ is the locational average of the particle swarm.

To include points such as D in Figure 1(b), the EDM expands the original feasible region and provides preferable information, which is more reasonable.

3. Example

3.1. Unconstrained Optimization. Four test functions are investigated to verify the effectiveness of chaotic methodology. The comparison of PSO, CPSO, and IPSO-EDM is shown in Tables 1–4. Firstly, the mathematic models are established as follows.

(1) Sphere function:

$$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2, \quad x_i \in [-100, 100]. \tag{18}$$

(2) Rastrigin function:

$$f_2(\mathbf{x}) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right], \quad x_i \in [-5.12, 5.12]. \tag{19}$$

(3) Rosenbrock function:

$$f_3(\mathbf{x}) = \sum_{i=1}^{n} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2, \quad x_i \in [-30, 30]. \tag{20}$$

(4) Ackley function:

$$f_4(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)\right) + 20 + e, \quad x_i \in [-32.768, 32.768]. \tag{21}$$

It is seen from Tables 1–4 that four test functions are used to verify the effectiveness of the IPSO-EDM. As seen in Tables 1–4, the worst value, average value, optimal value, and standard variance in the same dimension are, respectively, calculated via PSO, CPSO, and IPSO-EDM, and the results calculated by the IPSO-EDM are the minimum values. Thus, the computational accuracy of the IPSO-EDM is the highest. In short, the IPSO-EDM has optimal computational accuracy compared with PSO and CPSO.

3.2. Constrained Optimization

3.2.1. Numerical Case Studies. To test the effectiveness of the IPSO-EDM, 3 test case studies are investigated, the scale of the particle swarm is 50, the number of iterations is 1000 times, and $\xi_K = 1e-15, K = 900$. The statistical results of the optimal function and constraint conflict function are listed in Tables 5 and 6. The comparison of different methods is shown in Tables 7–10, where "N/A" means not available. Firstly, the mathematic models are established as follows.

G1:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \\
\text{s.t.} \quad & g_1(\mathbf{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \\
& g_2(\mathbf{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \\
& g_3(\mathbf{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \\
& g_4(\mathbf{x}) = -8x_1 + x_{10} \le 0 \\
& g_5(\mathbf{x}) = -8x_2 + x_{11} \le 0 \\
& g_6(\mathbf{x}) = -8x_3 + x_{12} \le 0 \\
& g_7(\mathbf{x}) = -2x_4 - x_5 + x_{10} \le 0 \\
& g_8(\mathbf{x}) = -2x_6 - x_7 + x_{11} \le 0 \\
& g_9(\mathbf{x}) = -2x_8 - x_9 + x_{12} \le 0,
\end{aligned} \tag{22}$$

where $0 \le x_i \le 1\,(i = 1, 2, \ldots, 9, 13)$ and $0 \le x_i \le 100\,(i = 10, 11, 12)$.

G2:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0 \\
& g_2(\mathbf{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0 \\
& g_3(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0 \\
& g_4(\mathbf{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0 \\
& g_5(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0 \\
& g_6(\mathbf{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,
\end{aligned} \tag{23}$$

Table 1: Experimental result of the Sphere function.

f_1	PSO		CPSO		IPSO-EDM	
Dimensionality	$n = 10$	$n = 20$	$n = 10$	$n = 20$	$n = 10$	$n = 20$
Worst value	$4.4369e-49$	$7.4023e-20$	$3.0672e-48$	$6.0385e-22$	$3.3696e-70$	$9.4993e-39$
Mean value	$6.5256e-50$	$6.2447e-21$	$1.5934e-49$	$7.9355e-23$	$1.7190e-71$	$7.0559e-40$
Optimal value	$1.1728e-53$	$1.5012e-24$	$1.5102e-54$	$2.1277e-27$	0	0
Standard variance	$1.1743e-49$	$1.6707e-20$	$6.8481e-49$	$1.5866e-22$	$7.5283e-71$	$2.2945e-39$

Table 2: Experimental result of the Rastrigin function.

f_2	PSO		CPSO		IPSO-EDM	
Dimensionality	$n = 10$	$n = 20$	$n = 10$	$n = 20$	$n = 10$	$n = 20$
Worst value	5.9697	27.8588	0	$1.8847e-10$	0	0
Mean value	3.0346	16.2676	0	$9.4753e-12$	0	0
Optimal value	0	6.9647	0	0	0	0
Standard variance	1.5304	5.3168	0	$4.2132e-11$	0	0

Table 3: Experimental result of the Rosenbrock function.

f_3	PSO		CPSO		IPSO-EDM	
Dimensionality	$n = 10$	$n = 20$	$n = 10$	$n = 20$	$n = 10$	$n = 20$
Worst value	22.1917	3.0299e3	12.2003	76.5662	2.9913	4.9399
Mean value	4.3672	174.9661	2.4949	20.5855	0.2530	0.2822
Optimal value	0.0976	4.7483	0.0014	0.1926	$9.9854e-4$	$9.9068e-4$
Standard variance	4.6971	672.3508	2.7330	26.6168	0.6919	1.0986

where
$78 \le x_1 \le 102$, $33 \le x_2 \le 45$, and $27 \le x_i \le 45\,(i = 3, 4, 5)$.

G3:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + \left(\frac{0.000002}{3}\right)x_2^3 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = -x_4 + x_3 - 0.55 \le 0 \\
& g_2(\mathbf{x}) = -x_3 + x_4 - 0.55 \le 0 \\
& h_3(\mathbf{x}) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\
& h_4(\mathbf{x}) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\
& h_5(\mathbf{x}) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0,
\end{aligned} \tag{24}$$

where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$, and $-0.55 \le x_4 \le 0.55$.

Three numerical case studies are used to verify the effectiveness of the proposed IPSO-EDM. As seen in Tables 7–10, the worst value, average value, and the best value are, respectively, calculated via the IPSO-EDM and five famous methods, i.e., HM, ASCHEA, SR, EDPSO, and MPSO. The investigation indicates that the computational accuracy of the IPSO-EDM is comparable to HM, ASCHEA,

Table 4: Experimental result of the Ackley function.

f_4	PSO		CPSO		IPSO-EDM	
Dimensionality	$n=10$	$n=20$	$n=10$	$n=20$	$n=10$	$n=20$
Worst value	$2.6645e-15$	$1.0484e-10$	$2.6645e-15$	$1.0991e-11$	$6.2172e-15$	$1.3323e-14$
Mean value	$2.6645e-15$	$3.9470e-11$	$2.6645e-15$	$1.8865e-12$	$2.8422e-15$	$7.1054e-15$
Optimal value	$2.6645e-15$	$1.1324e-12$	$2.6645e-15$	$1.0925e-13$	$2.6645e-15$	$2.6645e-15$
Standard variance	0	$3.6101e-11$	0	$3.0165e-12$	$7.9441e-16$	$2.7938e-15$

Note. The scale of the particle swarm is 50, the change interval of inertia factor $\omega \in [0.4, 0.9]$, earning factors $c_1 = c_2 = 2$, the number of iterations is 2000 times, the number of chaotic iterations is 1000 times, and the maximum number of evolutionary variations allowed is 1000 times.

SR, EDPSO, and MPSO. Furthermore, this research manifests that the computational efficiency of the IPSO-EDM is significantly improved compared with the other five methods (HM, ASCHEA, SR, EDPSO, and MPSO) by measuring the product of the group size and algorithm cycle times. It can be seen from Table 10 that the computational efficiency of the IPSO-EDM is the largest compared with the other five methods, i.e., the product of population size and algorithm cycle times of the IPSO-EDM is the least among all methods.

To verify the effectiveness of the presented algorithm, the new parameters such as β and $\mathrm{Vio}(x)$ are investigated. The performance function is written as

$$\mathbf{g}(\mathbf{X}_1, \mathbf{X}_2) = \mathbf{X}_1^3 + \mathbf{X}_2^3 - 18, \tag{25}$$

where $\mathbf{X}_1 \sim N(10, 5^2)$ and $\mathbf{X}_2 \sim N(9.9, 5^2)$.

The iteration process of β and $\mathrm{Vio}(x)$ is calculated by HL, W-G, and IPSO-EDM, which are shown in Figure 7.

Figure 7 indicates that $\mathrm{Vio}(x)$ of the standard HL method is very large, and its convergence value is 674.2829, but convergence values of W-G and IPSO-EDM are, respectively, $2.42e-2$ and $5.14e-4$, which manifests feasible points of HL are very few, i.e., the accuracy is very low. Meanwhile, it can be seen that the curve of $\mathrm{Vio}(x)$ obtained is fairly close each other by W-G and IPSO-EDM, but the convergence value obtained via the IPSO-EDM is $5.14e-4$ which is smaller than $2.42e-2$ obtained by W-G, and this means the accuracy of IPSO-EDM is higher than that of W-G. Meanwhile, the convergence value of β via HL is 1.1657, and they are 2.22572 and 2.22599 by W-G and IPSO-EDM. Thus, the computational accuracy and efficiency of the IPSO-EDM are optimal and of HL are the worst.

3.2.2. Engineering Case Study. Many researchers use the finite element method to study engineering [38], but they do not optimize it. This method can also be used in practical engineering, for instance, this is a welded beam structure, which is shown in Figure 8.

The optimization goal is to seek four design variables, i.e., $x_1(h)$, $x_2(l)$, $x_3(t)$, and $x_4(b)$, which satisfy the constraints of shear stress τ, bending stress σ, bending load P_c of the welding rod, deviation δ, and the boundary condition, and the total manufacturing cost of the welding rod is the minimum. The mathematical model is described as

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 1.104712x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
\text{s.t.} \quad & g_1(\mathbf{x}) = \tau(\mathbf{x}) - 13600 \le 0 \\
& g_2(\mathbf{x}) = \sigma(\mathbf{x}) - 30000 \le 0 \\
& g_3(\mathbf{x}) = x_1 - x_4 \le 0 \\
& g_4(\mathbf{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \\
& g_5(\mathbf{x}) = 0.125 - x_1 \le 0 \\
& g_6(\mathbf{x}) = \delta(x) - 0.25 \le 0 \\
& g_7(\mathbf{x}) = P - P_c(\mathbf{x}) \le 0,
\end{aligned} \tag{26}$$

where

$$\begin{aligned}
\tau(\mathbf{x}) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \\
\tau'(\mathbf{x}) &= \frac{P}{\sqrt{2}x_1x_2}, \\
\tau'' &= \frac{QR}{J}, \\
Q(\mathbf{x}) &= P\left(L + \frac{x_2}{2}\right), \\
R(\mathbf{x}) &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_2}{2}\right)^2}, \\
J(\mathbf{x}) &= 2\left\{\sqrt{2}\,x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \\
\sigma(\mathbf{x}) &= \frac{6PL}{x_4x_3^2}, \\
\delta(\mathbf{x}) &= \frac{4PL^3}{Ex_3^3x_4}, \\
P_c(\mathbf{x}) &= \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right).
\end{aligned} \tag{27}$$

Note that $P = 6000$, $L = 14$, $E = 30 \times 10^6$, $G = 12 \times 10^6$, $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$, and $0.1 \le x_4 \le 2$.

The comparison results obtained by the IPSO-EDM and the other methods are shown in Table 11.

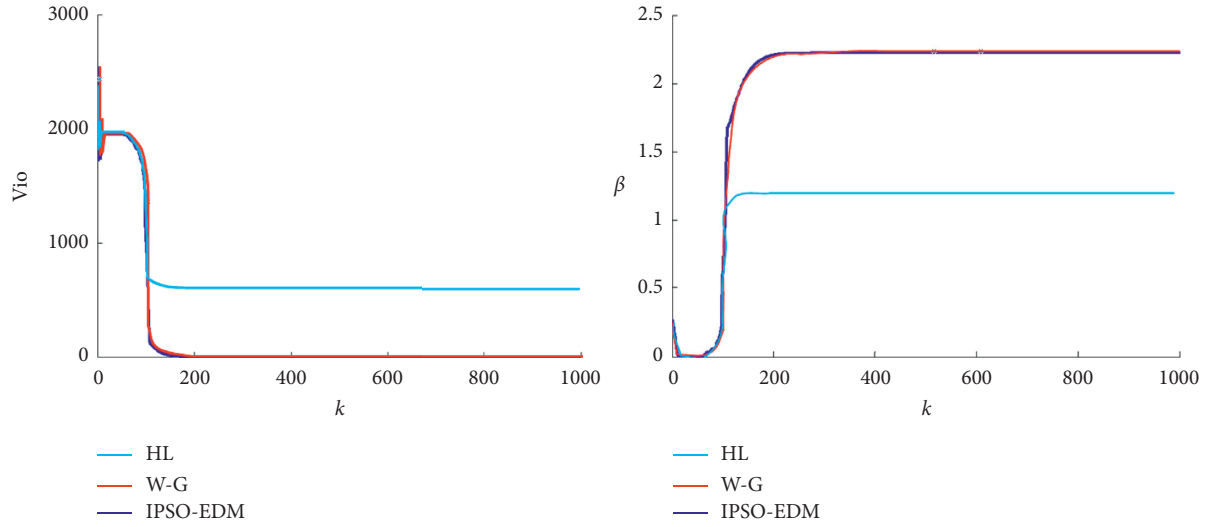

Figure 7: Iteration process of β and Vio(x).

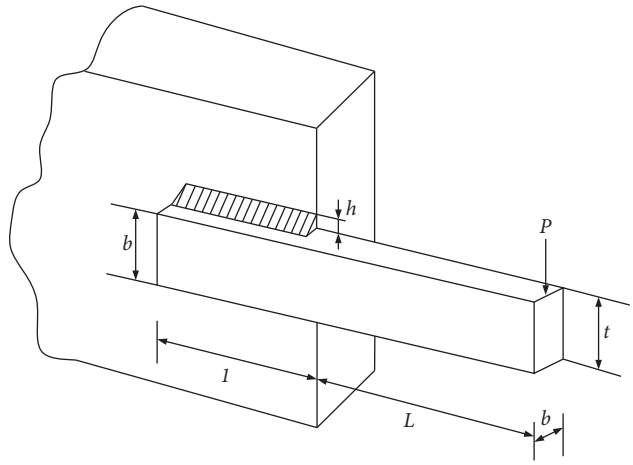

Figure 8: Welded beam schematic diagram.

Table 5: Statistics of the optimal value with the IPSO-EDM.

Test examples	Optimal value	Mean value	Worst value	Standard variance
G1	−15	−15	−15	$1.8724e-15$
G2	$-3.066e4$	$-3.066e4$	$-3.066e4$	0.0019
G3	$5.1265e3$	$5.1482e3$	$5.1871e3$	20.3178

Table 6: Statistics of the constraint conflict function with the IPSO-EDM.

Test examples	Optimal value	Mean value	Worst value	Standard variance
G1	0	$4.8850e-16$	$2.6645e-15$	$1.0352e-15$
G2	0	$1.7764e-15$	$1.4211e-14$	$4.5094e-15$
G3	0	$8.8510e-4$	0.0061	0.0020

Table 7: Comparison of the optimal results obtained by different methods.

Test examples	HM [35]	ASCHEA [36]	SR [37]	IPSO-EDM	EDPSO [21]	MPSO [25]
G1	−14.7886	−15.0000	−15.0000	−15.0000	−15.0000	−14.9863
G2	−30665.5	−30665.5	−30665.5	−30665.5	−30665.5	−30665.5
G3	N/A	5126.5	5126.4	5126.5	5126.5	5126.5

Table 8: Comparison of average results obtained by different methods.

Test examples	HM [35]	ASCHEA [36]	SR [37]	IPSO-EDM	EDPSO [21]	MPSO [25]
G1	−14.7082	−14.8400	−15.0000	−15.0000	−15.0000	−14.9986
G2	−30665.3	−30665.5	−30665.5	−30665.5	−30665.5	−30665.5
G3	N/A	5141.7	5128.9	5148.2	5148.1	5149.7

Table 9: Comparison of the worst results obtained by different methods.

Test examples	HM [35]	ASCHEA [36]	SR [37]	IPSO-EDM	EDPSO [21]	MPSO [25]
G1	−14.6154	N/A	−15.0000	−15.0000	−15.0000	−14.9992
G2	−30645.9	N/A	−30665.5	−30665.5	−30665.5	−30665.5
G3	N/A	N/A	5142.472	5187.134	5148.1	5146.5

Table 10: Comparison of computational efficiency of different methods.

Test examples	HM [35]	ASCHEA [36]	SR [37]	IPSO-EDM	EDPSO [21]	MPSO [25]	η_{IP_H} (%)	η_{IP_A} (%)	η_{IP_S} (%)	η_{IP_E} (%)	η_{IP_M} (%)
G1	1400000	1500000	350000	50000	60000	90000	96.42	96.67	85.71	16.67	44.44
G2	1400000	1500000	350000	50000	60000	90000	96.42	96.67	85.71	16.67	44.44
G3	1400000	1500000	350000	50000	60000	90000	96.42	96.67	85.71	16.67	44.44

Note. η_{IP_H} is the improved computational efficiency by the IPSO-EDM compared with that by HM; η_{IP_A} is the improved computational efficiency of the IPSO-EDM compared with that by ASCHEA; η_{IP_S} is the improved computational efficiency by the IPSO-EDM compared with that by SR; η_{IP_E} is the improved computational efficiency by the IPSO-EDM compared with that by EDPSO; and η_{IP_M} is the improved computational efficiency by the IPSO-EDM compared with that by MPSO.

Table 11: Comparison of the calculation results by different methods for the welding beam.

Methods	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	$f(x)$
Coello [39]	0.208800	3.420500	8.997500	0.210000	1.748309
Coello and Montes [40]	0.205986	3.471328	9.020224	0.206480	1.728226
Coello and Becerra [41]	0.205700	3.470500	9.036600	0.205700	1.724852
IPSO-EDM	0.205730	3.470489	9.036624	0.205730	1.724852
CPSO [27]	0.205731	3.470582	9.036839	0.205680	1.724237
Methods	Optimal value	Mean value	Worst value	Standard variance	
Coello [39]	1.748309	1.771973	1.785835	0.001122	
Coello and Montes [40]	1.728226	1.792654	1.993408	0.074713	
Coello and Becerra [41]	1.724852	1.971809	3.179709	0.443131	
IPSO-EDM	1.724852	1.738620	1.959606	0.048555	
CPSO [27]	1.724381	1.738810	1.958721	0.047891	

The optimal results obtained by the IPSO-EDM are equivalent to those provided by the existing literature, which verifies the accuracy of this method. However, the computational efficiency of the proposed IPSO-EDM is higher than the methods in the literature in Table 10.

4. Conclusions

The standard PSO algorithm is improved from the perspective of engineering application to solve engineering optimization problems.

(1) The original feasible region is expanded, and some points that are closer to the constrained optimal point in the feasible region are contained as feasible points, which provide preferable function information compared with the points in the original feasible region. This approach uses the current location information of the particles and particle swarm to determine the speed of the particles. In short, the difference value in the current optimal positions of the particle swarm and the particle is used to determine the current particle velocity. The optimal location of obtained points using the proposed method is better than that obtained using the standard PSO.

(2) The constrained optimization is transformed into the unconstrained optimization by combining the ergodicity of chaos optimization and the evolutionary variation to realize global search. The logistic chaotic system equation is applied in the PSO algorithm, and the mutation operator is introduced in the evolutionary variation strategy to escape local optimal and maintain its vitality of the particle swarm, which prevents the particle swarm from falling into the condition of "precocity" at the earlier iteration.

(3) An unconstrained optimization case study, four numerical case studies, and one engineering case study are used to verify the effectiveness of the

IPSO-EDM. The worst value, average value, and optimal value are, respectively, calculated via the IPSO-EDM and compared with other methods. The investigation indicates that the computational accuracy of the IPSO-EDM is comparable to that provided by the existing literature; however, the computational efficiency of the IPSO-EDM is significantly improved.

(4) Though the PSO method is improved and six case studies are investigated to prove the effectiveness of this method, yet the numerical case studies are relatively simple, and the design variables of the engineering case study are small. Moreover, only the deterministic optimization is studied. Thus, the nondeterministic optimization will be researched, and the random variable will be subject to normal distribution, exponential distribution, or Weibull distribution. Accordingly, the proposed method can be expanded to wide-spread engineering application fields. Further studies will focus on the IPSO-EDM considering random variables of different distributions including normal distribution, exponential distribution, and Weibull distribution to deal with actual operation of the machines.

Data Availability

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data 6/12 months after publication of this article will be considered by the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the financial support for this research from the National Key R&D Plan Project (Grant no. 2017YFB1301300), the National Natural Science Foundation of China (Grant nos. 11772011 and 11902220), and the National Natural Science Foundation of Hebei Province (Grant no. E2020202217).

References

[1] Y. M. Zhang, Z. X. Wen, H. Q. Pei, J. P. Wang, Z. W. Li, and Z. F. Yue, "Equivalent method of evaluating mechanical properties of perforated Ni-based single crystal plates using artificial neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, Article ID 112725, 2020.

[2] A. L. Soubhia and A. L. Serpa, "Discrete optimization for positioning of actuators and sensors in vibration control using the simulated annealing method," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 2, p. 101, 2020.

[3] H. Zhi and S. Y. Liu, "A hybrid GABC-GA Algorithm for mechanical design optimization problems," *Intelligent Automation and Soft Computing*, vol. 25, no. 4, pp. 815–825, 2019.

[4] J. Martínez-Morales, H. Quej-Cosgaya, J. Lagunas-Jiménez, E. Palacios-Hernández, and J. Morales-Saldaña, "Design optimization of multilayer perceptron neural network by ant colony optimization applied to engine emissions data," *Science China Technological Sciences*, vol. 62, no. 6, pp. 1055–1064, 2019.

[5] M. Huang and Z. Liu, "Research on mechanical fault prediction method based on multifeature fusion of vibration sensing data," *Sensors*, vol. 20, no. 1, p. 6, 2020.

[6] J. P. Janet, L. Chan, and H. J. Kulik, "Accelerating chemical discovery with machine learning: simulated evolution of spin crossover complexes with an artificial neural network," *The Journal of Physical Chemistry Letters*, vol. 9, no. 5, pp. 1064–1071, 2018.

[7] F. Zhao, S. Qin, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A two-stage differential biogeography-based optimization algorithm and its performance analysis," *Expert Systems with Applications*, vol. 115, pp. 329–345, 2019.

[8] E. Pena, S. M. Zhang, R. Patriat et al., "Multi-objective particle swarm optimization for postoperative deep brain stimulation targeting of subthalamic nucleus pathways," *Journal of Neural Engineering*, vol. 15, no. 6, Article ID 066020, 2018.

[9] E. Camci, D. R. Kripalani, L. L. Ma, E. Kayacan, and M. A. Khanesar, "An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm," *Swarm and Evolutionary Computation*, vol. 41, pp. 1–8, 2018.

[10] J.-Y. Jhang, C.-J. Lin, C.-T. Lin, and K.-Y. Young, "Navigation control of mobile robots using an interval type-2 fuzzy controller based on dynamic-group particle swarm optimization," *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2446–2457, 2018.

[11] W. Tao, Z. Liu, P. Zhu, C. Zhu, and W. Chen, "Multi-scale design of three dimensional woven composite automobile fender using modified particle swarm optimization algorithm," *Composite Structures*, vol. 181, pp. 73–83, 2017.

[12] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Applied Soft Computing*, vol. 73, pp. 969–991, 2018.

[13] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Improving the efficiency of stochastic vehicle routing: a partial Lagrange multiplier method," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3993–4005, 2016.

[14] J.-G. Ahn, H.-I. Yang, and J.-G. Kim, "Multipoint constraints with Lagrange multiplier for system dynamics and its reduced-order modeling," *AIAA Journal*, vol. 58, no. 1, pp. 385–401, 2020.

[15] J. Liang, L. Dai, S. Chen et al., "Generalized inverse matrix-exterior penalty function (GIM-EPF) algorithm for data processing of multi-wavelength pyrometer (MWP)," *Optics Express*, vol. 26, no. 20, pp. 25706–25720, 2018.

[16] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, 1995.

[17] Z. Xue, H. Li, Y. Zhou, N. Ren, and W. Wen, "Analytical prediction and optimization of cogging torque in surface-mounted permanent magnet machines with modified particle swarm optimization," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 12, pp. 9795–9805, 2017.

[18] H. Han, X. Wu, L. Zhang, Y. Tian, and J. Qiao, "Self-organizing RBF neural network using an adaptive gradient

multiobjective particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 69–82, 2019.

[19] J. Yi, X. Li, C.-H. Chu, and L. Gao, "Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 405–428, 2019.

[20] J. B. Park, Y. W. Jeong, J. R. Shin, and K. Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 156–166, 2010.

[21] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[22] R. Wang and X. Zhang, "Optimal design of a planar parallel 3-DOF nanopositioner with multi-objective," *Mechanism and Machine Theory*, vol. 112, pp. 61–83, 2017.

[23] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing*, vol. 11, no. 4, pp. 3658–3670, 2011.

[24] H. D. Mojarrad and M. Nayeripour, "A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch," *Energy*, vol. 35, no. 4, pp. 1764–1778, 2010.

[25] S. Khan, M. Kamran, and O. U. Rehman, "A modified PSO algorithm with dynamic parameters for solving complex engineering design problem," *International Journal of Computer Mathematics*, vol. 95, no. 11, pp. 2308–2329, 2018.

[26] H. B. Liang, D. L. Zou, Z. L. Li, M. J. Khan, and Y. J. Lu, "Dynamic evaluation of drilling leakage risk based on fuzzy theory and PSO-SVR algorithm," *Future Generation Computer Systems*, vol. 95, pp. 454–466, 2019.

[27] D. P. Tian, X. F. Zhao, and Z. Z. Shi, "Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization," *Swarm and Evolutionary Computation*, vol. 51, Article ID 100573, 2019.

[28] F. S. Hsieh, F. M. Zhan, and Y. H. Guo, "A solution methodology for carpooling systems based on double auctions and cooperative coevolutionary particle swarms," *Applied Intelligence*, vol. 49, no. 2, pp. 741–763, 2019.

[29] S. K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, vol. 181, no. 2, pp. 527–548, 2007.

[30] E. Zahara and C.-H. Hu, "Solving constrained optimization problems with hybrid particle swarm optimization," *Engineering Optimization*, vol. 40, no. 11, pp. 1031–1049, 2008.

[31] X. F. Liu, Z. H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 587–602, 2019.

[32] X. W. Xia, L. Gui, F. Yu et al., "Triple archives particle swarm optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.

[33] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin, and J. Zhang, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[34] A. Hinrichs, F. Pillichshammer, and S. Tezuka, "Tractability properties of the weighted star discrepancy of the Halton sequence," *Journal of Computational and Applied Mathematics*, vol. 350, pp. 46–54, 2019.

[35] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

[36] S. B. Hamida and M. Schoenauer, "ASCHEA: new results using adaptive segregational constraint handing," in *Proceedings of the 2002 Congress on Evolutionary Computation*, IEEE, Piscataway, NJ, USA, pp. 884–889, May 2002.

[37] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[38] B. Bai, H. Li, W. Zhang, and Y. C. Cui, "Application of extremum response surface method-based improved substructure component modal synthesis in mistuned turbine bladed disk," *Journal of Sound and Vibration*, vol. 472, Article ID 115210, 2020.

[39] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 1, pp. 113–127, 2000.

[40] C. A. C. Coello and E. M. Montes, "Constraint-handing in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, vol. 16, no. 1, pp. 193–203, 2002.

[41] C. A. C. Coello and R. L. Becerra, "Efficient evolutionary optimization through the use of a cultural algorithm," *Engineering Optimization*, vol. 36, no. 2, pp. 219–236, 2004.