Research Article

A Framework for Constrained Optimization Problems Based on a Modified Particle Swarm Optimization

Biwei Tang,^{1,2} Zhanxia Zhu,^{1,2} and Jianjun Luo^{1,2}

¹National Key Laboratory of Aerospace Flight Dynamics, Xi'an, Shaanxi 710072, China

²School of Astronautics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Zhanxia Zhu; zhuzhanxia@nwpu.edu.cn

Received 1 March 2016; Revised 19 May 2016; Accepted 15 June 2016

Academic Editor: László T. Kóczy

Copyright © 2016 Biwei Tang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper develops a particle swarm optimization (PSO) based framework for constrained optimization problems (COPs). Aiming at enhancing the performance of PSO, a modified PSO algorithm, named SASPSO 2011, is proposed by adding a newly developed self-adaptive strategy to the standard particle swarm optimization 2011 (SPSO 2011) algorithm. Since the convergence of PSO is of great importance and significantly influences the performance of PSO, this paper first theoretically investigates the convergence of SASPSO 2011. Then, a parameter selection principle guaranteeing the convergence of SASPSO 2011 is provided. Subsequently, a SASPSO 2011-based framework is established to solve COPs. Attempting to increase the diversity of solutions and decrease optimization difficulties, the adaptive relaxation method, which is combined with the feasibility-based rule, is applied to handle constraints of COPs and evaluate candidate solutions in the developed framework. Finally, the proposed method is verified through 4 benchmark test functions and 2 real-world engineering problems against six PSO variants and some well-known methods proposed in the literature. Simulation results confirm that the proposed method is highly competitive in terms of the solution quality and can be considered as a vital alternative to solve COPs.

1. Introduction

Over the last few decades, constrained optimization problems (COPs) have rapidly gained increasing research interests, since they are frequently encountered in different areas such as path planning [1], resource allocation [2], and economic environmental scheduling [3] to name but a few. Generally, solving a constrained optimization problem is to optimize a predefined objective function under some equality and/or inequality constraints [4, 5]. Nevertheless, owing to the nonlinearity in either the objective or constraints, or both, efficiently solving COPs remains a big challenge [4, 5]. Therefore, far more effective optimization algorithms are always needed.

Due to their population-based nature and promising search ability to produce high-quality solutions, even for complex optimization problems [6], evolutionary algorithms (EAs), such as genetic algorithm (GA) [2], simulated annealing (SA) [3], and differential evolution (DE) [7], have been proposed for solving different COPs. As one of the most

powerful EAs, thanks to its simplicity and high convergence speed, particle swarm optimization (PSO) has been widely and successfully applied to solve different COPs in recent years [8–12].

Yet, since the basic PSO algorithm suffers from some drawbacks such as stagnation and poor ability in balancing exploration and exploitation, its optimization efficiency may be restricted [13, 14]. In order to improve the performance of the PSO, these weaknesses must be overcome. Moreover, when designing a PSO algorithm, the convergence of PSO is paramount because this property significantly influences the performance of PSO [15, 16]. To date, despite some studies investigating the convergence to an equilibrium point of PSO [16–19], the optimality of this point is not clearly established. Actually, it is still difficult to theoretically analyze the global or local convergence (i.e., the global or local optimality of this equilibrium point) of PSO due to its stochastic nature [15, 16].

So far, many researchers have committed themselves to developing different PSO algorithms in order to enhance the performance of PSO. Liu et al., proposed a hybrid PSO

algorithm which hybridizes PSO algorithm with differential evolution (DE) algorithm in [20]. To tackle the stagnation issue, they proposed a new DE algorithm to evolve the personal best experience of particles in their hybrid PSO [20]. For sufficiently balancing the exploration and exploitation capabilities of PSO, Taherkhani and Safabakhsh [21] proposed a novel stability-based PSO algorithm, in which an adaptive approach is developed to determine the inertia weight of each particle in different dimensions. Furthermore, by considering the stability condition and the adaptive inertial weight, the cognitive and social acceleration parameters of their proposed PSO algorithm are adaptively determined [21]. Through extensive simulations on different benchmark test functions and a real-world application, the effectiveness and superiority of their proposed PSO have been validated in [21].

Additionally, among the currently existing PSO variants, based on the best knowledge of the authors, the standard particle swarm optimization (SPSO 2011) algorithm [22, 23] may be one of the most recent standard versions for PSO. By randomly drawing a point in a hypersphere which is centered on three points, the current position of the particle, a point a little “beyond” the personal best position of the particle, and a point a little “beyond” the global best position of the swarm, the nonstagnation property can be achieved in SPSO 2011 [22, 23]. However, since its three control parameters (i.e., the inertial weight and the cognitive and social acceleration parameters) are constant and there is no distinction between the cognitive acceleration parameter and the social acceleration parameter, SPSO 2011 cannot dynamically adjust the exploration and exploitation abilities. Besides, the convergence and the stability of SPSO 2011 have not been investigated in [22, 23].

Considering the advantage and the disadvantage of SPSO 2011, we propose a modified PSO algorithm, called SASPSO 2011, which is developed based on SPSO 2011. The main consideration of the development of SASPSO 2011 is to exploit the advantage (i.e., nonstagnation property) and overcome the shortcoming (i.e., poor ability in balancing exploration and exploitation) of SPSO 2011, so that the performance of SASPSO 2011 can be enhanced. To this end, particles in SASPSO 2011 first follow the same moving rules defined in SPSO 2011 to update their velocities and positions to prevent stagnation in SASPSO 2011. Then, a new self-adaptive strategy is developed for fine-tuning the three control parameters of particles in SASPSO 2011 to well balance the exploration and exploitation abilities of SASPSO 2011.

Although SASPSO 2011 is developed based on SPSO 2011, there are significant differences between these two algorithms, since

- (1) a novel self-adaptive strategy is proposed for fine-tuning the three control parameters of particles in SASPSO 2011,
- (2) the stability and the local convergence of SASPSO 2011 are investigated,
- (3) the convergence behavior of particles in SASPSO 2011 is investigated,

- (4) a parameter selection principle that can guarantee the local convergence of SASPSO 2011 is provided.

After analytical investigation on SASPSO 2011, this paper designs a SASPSO 2011-based framework for solving COPs. In order to easily handle constraints of COPs and release the burden of implementing the optimization algorithm, the adaptive relaxation method [4, 5] is combined with the feasibility-based rule [24, 25] to handle constraints of COPs and evaluate candidate solutions in the framework established. To verify the proposed method, it is compared to six state-of-the-art PSO variants and some methods proposed in the literature by solving 4 benchmark test functions and 2 real-world engineering problems. The simulation results show that the proposed method is highly competitive in finding high-quality solutions. Furthermore, the search stability of the proposed method is comparable with that of SAIWPSO [21] and outperforms those of the other compared methods. Thus, the proposed method can be considered as an effective optimization tool to solve COPs.

The remainder of this paper is organized as follows. After briefly reviewing SPSO 2011, SASPSO 2011 is presented in Section 2. Section 3 theoretically investigates some properties such as the stability, the local convergence, the convergence behavior of particles, and parameter selection principle pertaining to SASPSO 2011. The SASPSO 2011-based framework for COPs is described in Section 4. Simulations and comparisons are performed in Section 5. Section 6 summarizes this paper by way of a conclusion and options of future work.

2. Particle Swarm Optimization (PSO)

2.1. Review of SPSO 2011. Inspired by birds flocking and fish schooling, Eberhart and Kennedy [26] first proposed PSO in 1995. The aim of original PSO is to reproduce the social interactions among agents to solve different optimization problems [27]. Each agent in PSO is called a particle and is associated with a velocity that is dynamically adjusted accordingly to its own flight experience, as well as those of its companions. Since the first introduction of PSO in 1995, many different PSO algorithms have been proposed, among which SPSO 2011 [22, 23] may be one of the most recently proposed PSO algorithms. Because our proposed PSO algorithm is developed on the basis of SPSO 2011, this subsection will briefly describe SPSO 2011.

Let $X_i(t) = [x_{i1}(t), \dots, x_{in}(t)]$ and $V_i(t) = [v_{i1}(t), \dots, v_{in}(t)]$ denote the position and velocity of particle i in a swarm with NP particles in a search space $S \in \mathbb{R}^n$ for $n \in \mathbb{N}$ at iteration t . $X_{pi}(t) = [x_{pi1}(t), \dots, x_{pin}(t)]$ stands for the personal best position of the i th particle at iteration t . $X_g(t) = [x_{g1}(t), \dots, x_{gn}(t)]$ represents the global best position of the swarm at iteration t . The position of particle i at iteration $t + 1$ is obtained from three components: the current velocity $V_i(t)$, the personal best position $X_{pi}(t)$, and the global best position $X_g(t)$. Let $G_i(t)$ denote the isobarycenter of the i th particles $X_i(t)$, $\phi_1 X_{pi}(t)$, and $\phi_2 X_g(t)$, where ϕ_1 and ϕ_2 are two positive real coefficients denoting the cognitive and social

acceleration parameters. The coordinates of the barycenter $G_i(t)$ can be obtained as follows [22, 23]:

$$G_i(t) = X_i(t) + \frac{\phi_1 [X_{pi}(t) - X_i(t)] + \phi_2 [X_g(t) - X_i(t)]}{3}. \quad (1)$$

Then, a point $X'_i(t)$ is randomly drawn in a hypersphere $H(G_i(t), \|G_i(t) - X_i(t)\|)$ that is centered on $G_i(t)$ with a radius $\|G_i(t) - X_i(t)\|$. After randomly obtaining a point $X'_i(t)$ in the hypersphere, each particle updates its velocity and position as follows [22, 23]:

$$V_i(t+1) = wV_i(t) + X'_i(t) - X_i(t), \quad (2)$$

$$X_i(t+1) = V_i(t+1) + X_i(t), \quad (3)$$

where w is a real coefficient representing the inertia weight. In [22], it is suggested that if $X_{pi}(t) = X_g(t)$, $G_i(t) = (X_i(t) + X_i(t) + \phi_2(X_g(t) - X_i(t)))/2$.

In SPSO 2011, the particle can always explore the surroundings of the explored region with a nonnull velocity, since a random point $X'_i(t)$ is added to the particle's velocity as shown in (2). Hence, the nonstagnation property can be achieved in SPSO 2011 [22, 23]. The authors in [22] also propose to set the three control parameters of SPSO 2011 as follows:

$$\begin{aligned} w &= \frac{1}{2 \ln 2}, \\ \phi_1 &= 0.5 + \ln 2, \\ \phi_2 &= 0.5 + \ln 2. \end{aligned} \quad (4)$$

2.2. Description of SASPSO 2011. When applying PSO to solve an optimization problem, it is necessary to properly control the exploration and exploitation abilities of PSO in order to find optimal solutions efficiently [13, 14, 28]. Ideally, on one hand, in the early stage of the evolution, the exploration ability of PSO must be promoted so that particles can wander through the entire search space rather than clustering around the current population-best solution [13, 14, 28]. On the other hand, in the later stage of the evolution, the exploitation ability of PSO needs to be strengthened so that particles can search carefully in a local region to find optimal solutions efficiently [13, 14, 28].

Proverbially, the exploration and exploitation capabilities of PSO heavily depend on its three control parameters. The basic philosophies concerning how the three control parameters influence such abilities of PSO can be summarized as follows: (1) a large inertia weight enhances exploration, while a small inertia weight facilitates exploitation [13, 14, 28]; (2) a large cognitive component, compared to the social component, results in the wandering of particles through the entire search space, which strengthens exploration [14, 27]; (3) a large social component, compared with the cognitive component, leads particles to a local search, which strengthens exploitation [14, 27].

According to the basic philosophies noted above, although SPSO 2011 is a nonstagnation algorithm, it cannot strike a good balance between exploration and exploitation, since its three control parameters remain unchanged and there is no difference between ϕ_1 and ϕ_2 . Considering the weakness of SPSO 2011, we propose a modified PSO algorithm, which is developed based on SPSO 2011 and is named SASPSO 2011. The main purpose of the development of this PSO is to adaptively adjust the exploration and exploitation abilities of SASPSO 2011. To achieve this goal, a novel self-adaptive strategy that is used to update the three control parameters of particles in SASPSO 2011 is proposed as follows:

$$w_i(t+1) = (w_s - w_f) \exp\left(-\frac{\delta_w t}{\beta_i}\right) + w_f, \quad (5)$$

$$\phi_{i1}(t+1) = (\phi_{1s} - \phi_{1f}) \exp\left(-\frac{\delta_{\phi_1} t}{\beta_i}\right) + \phi_{1f}, \quad (6)$$

$$\phi_{i2}(t+1) = (\phi_{2s} - \phi_{2f}) \exp\left(\frac{\delta_{\phi_2} t}{\beta_i}\right) + \phi_{2f}, \quad (7)$$

where

$$\delta_w = \frac{w_s - w_f}{t_{\max}}, \quad (8)$$

$$\delta_{\phi_1} = \frac{\phi_{1s} - \phi_{1f}}{t_{\max}}, \quad (9)$$

$$\delta_{\phi_2} = \frac{\phi_{2s} - \phi_{2f}}{t_{\max}}, \quad (10)$$

$$\beta_i = \|X_g(t) - X_{pi}(t)\|, \quad (11)$$

where w_s and w_f denote the initial and final values of the inertia weight, respectively; ϕ_{1s} and ϕ_{1f} represent the initial and final values of the cognitive acceleration parameter, respectively; ϕ_{2s} and ϕ_{2f} stand for the initial and final values of the social acceleration parameter, respectively; $t_{\max}$ is the maximum iteration number; $\|X_g(t) - X_{pi}(t)\|$ denotes the Euclidean distance between the personal best position of particle i and the global best position of the swarm at iteration t . Note that $w_s > w_f$, $\phi_{1s} > \phi_{1f}$, and $\phi_{2f} > \phi_{2s}$ in SASPSO 2011. Also, note that particles in SASPSO 2011 update their velocities and positions based on (1)–(3) in order to avoid them falling into stagnation.

2.2.1. Parametric Analysis for SASPSO 2011. From (5) to (7), with increasing iteration number t , it is clear that w_i and ϕ_{i1} decrease, while ϕ_{i2} increases in SASPSO 2011. Therefore, according to the aforementioned basic philosophies, SASPSO 2011 may start with high exploration, which will be reduced over time, so that exploitation may be favored in the later phase of the evolution. Note that, following the update rule of a fixed β_i , the balance between exploration and exploitation varies only with respect to the iteration number t .

We also adapt the balance of the search in SASPSO 2011 using an additional parameter β_i . From (5) and (6), it is trivial

that w_i and ϕ_{i1} decrease as β_i increases. On the other hand, the variation in ϕ_{i2} becomes larger as β_i increases according to (7). This implies that, for large β_i , the exploration capability of SASPSO 2011 tends to be retained. According to (11), a large β_i indicates that the personal best position of the particle is far away from the global best position of the swarm. Therefore, in the case where β_i is large, it is natural to facilitate the exploration capability of the algorithm, so that the particle is promoted to a global search and can quickly move closer to the global best position.

In contrast, in the case where β_i is small, the exploitation ability of the algorithm takes over the exploration ability more rapidly as β_i decreases. It is also natural to strengthen the exploitation ability of the algorithm in the case where β_i is small, because, according to (11), small β_i implies that the personal best position of the particle is close to the global best position of the swarm. In this case, through strengthening the exploitation ability of the algorithm, particles tend to a local search around the global best position, so that the possibility of improving the quality of the global best solution can be increased.

Briefly, by utilizing the proposed self-adaptive strategy, the three control parameters of the algorithm can be adaptively adjusted, complying with the basic philosophies of PSO development. Hence, SASPSO 2011 is expected to improve the ability in finding high-quality solutions. Figure 1 demonstrates the tendency of these changes in the three control parameters with respect to different values of β_i . Note that $w_s = 0.9$, $w_f = 0.1$, $\phi_{1s} = \phi_{2f} = 2.5$, $\phi_{1f} = \phi_{2s} = 0.5$, and $t_{\max} = 100$ in Figure 1.

3. Analytical Investigations on SASPSO 2011

3.1. Mathematical Theory of Convergence and Some Basic Concepts. When designing a PSO algorithm, the convergence of the algorithm remains a key issue [15]. However, owing to the stochastic nature of PSO, it results in difficulties in theoretically investigating the global or local convergence of the algorithm [15, 17]. Fortunately, Solis and Wets [29] have studied under which conditions the stochastic algorithms such as PSO can be considered as either globally or locally convergent search algorithms. In [29], it has been proven that only if a stochastic algorithm satisfies the *algorithm condition* and the *convergence condition* can the algorithm be considered as a local convergence algorithm. Since we will use these two conditions to analyze the local convergence of SASPSO 2011, these two conditions and some relevant notations presented in [15, 29] are first reproduced below for convenience.

Optimality Region. Let D be the mapping that integrates the position of a particle $X(t)$ with the current global best solution X_g . In a minimization optimization problem, the optimality region can be mathematically described as follows [15]:

$$R_\xi = \{x \in S \mid f(x) < X^* + \xi\}, \quad (12)$$

where X^* denotes the optimal solution of f on S ; f denotes the objective function; S denotes the search space; ξ is a positive coefficient.

Algorithm Condition. In a minimization optimization problem, mapping $D : S \times \mathbb{R}^n$ must satisfy $f(D(X_g, X(t+1))) \leq f(X_g)$ and if $X(t+1) \in S$, then $f(D(X_g, X(t+1))) \leq f(X(t+1))$, where t denotes the iteration number (monotonic condition).

The algorithm condition on mapping D stipulates that the solution obtained at current iteration is not worse than the previous ones. In other terms, at each iteration, if the newly obtained fitness dominates the previously best one, then the new one replaces the previously best fitness. Otherwise, the best fitness remains unchanged.

Convergence Condition. Based on the definition of the optimal region R_ξ given by (12), the convergence to a local optimum is obtained by the sufficient condition: for all $X_t \in S$, there exists $\delta \in \mathbb{R} > 0$ and $0 < \eta \in \mathbb{R} < 1$, such that [15]

$$\mu_t(\text{dist}(X(t+1), R_\xi) \leq \text{dist}(X(t), R_\xi) - \delta, \quad (13)$$

or $X(t) \in R_\xi \geq \eta$,

where μ_t represents a probability measure; X_t denotes the solution obtained by the optimization problem at iteration t ; $\text{dist}(x, Z)$ is the distance between a point x and a set Z . For more details about $\text{dist}(x, Z)$, the reader is referred to [15].

The convergence condition implies that a stochastic algorithm can be considered as an optimization problem if the particle has a nonzero probability to move closer to the optimality region R_ξ by a minimum distance δ at each iteration or if the particle is already located at the optimality region R_ξ with a probability no less than η .

Theorem 1. Suppose f is a measurable function, S represents a measure subset of $\mathbb{R}^n$, and a stochastic algorithm satisfies both the algorithm and convergence conditions. Then, considering the sequence $\{x_t\}_{t=0}^\infty$ searched by the algorithm, the following condition holds [29]:

$$\lim_{t \rightarrow \infty} \mu_t(x_t \in R_\xi) = 1, \quad (14)$$

where $\mu_t(x_t \in R_\xi)$ denotes the probability that the algorithm converges to the optimal region R_ξ .

Theorem 1 indicates that only if a stochastic algorithm satisfies the algorithm condition and convergence condition stated above, can it be considered as a local convergence algorithm and the local optimality of the algorithm can be at least guaranteed. The proof of this theorem can be found in [29].

3.2. Local Convergence Proof of SASPSO 2011

3.2.1. Verification of the Algorithm Condition. The algorithm condition can be validated through applying mapping D used in SASPSO 2011. Let $\{X_g(t)\}_{t=0}^\infty$ represent the global best position set searched by particles in SASPSO 2011. In

FIGURE 1: Changes of the three control parameters under different β_i in SASPSO 2011.

a minimization optimization problem, after each iteration, mapping D in SASPSO 2011 is updated as follows:

$$D(X_g(t), X_i(t)) = \begin{cases} X_g(t), & \text{if } f(X_g(t)) \leq f(X_i(t)), \\ X_i(t), & \text{if } f(X_g(t)) > f(X_i(t)), \end{cases} \quad (15)$$

where $X_i(t)$ denotes the obtained position of particle i at iteration t . According to (15), if the newly obtained solution $X_i(t)$ is better than $X_g(t)$, $X_i(t)$ replaces $X_g(t)$ in SASPSO 2011. Otherwise, $X_g(t)$ is kept in D . Therefore, the obtained

sequence $\{f(X_g(t))\}_{t=0}^{\infty}$ searched by particles in SASPSO 2011 is monotonic. Thus, SASPSO 2011 satisfies the algorithm condition.

3.2.2. Stability Analysis for SASPSO 2011. Prior to verifying that SASPSO 2011 satisfies the convergence condition, we first investigate the stability of SASPSO 2011. The aim of the stability analysis is to find boundaries of the three control parameters to guarantee the trajectory convergence of particles in SASPSO 2011. Note that this study focuses on the deterministic model stability analysis [17] for SASPSO 2011 in the case where $X'_i = G_i(t)$. Without loss of generality,

by omitting the subscript i of each variable in (1)–(3) for simplicity, the update rules of particles in SASPSO 2011 can be rewritten into a matrix form as follows:

$$\begin{bmatrix} X(t+1) \\ V(t+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} X(t) \\ V(t) \end{bmatrix} + \mathbf{B}P, \quad (16)$$

where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1-\phi & w \\ -\phi & w \end{bmatrix}, \\ \mathbf{B} &= [\phi, \phi]^T, \\ \phi &= \frac{\phi_1 + \phi_2}{3}, \\ P &= \frac{\phi_1 \cdot X_p + \phi_2 \cdot X_g}{\phi_1 + \phi_2}. \end{aligned} \quad (17)$$

Solving $|\lambda \mathbf{E} - \mathbf{A}| = 0$, where $\mathbf{E}$ is the identity matrix with the same size of $\mathbf{A}$, the characteristic equation of the dynamic system (16) is derived as follows:

$$\lambda^2 - (w+1-\phi)\lambda + w = 0, \quad (18)$$

where two roots, denoted by $\lambda_{1,2}$, are obtained as follows:

$$\lambda_{1,2} = \frac{1+w-\phi \pm \sqrt{(1+w-\phi)^2 - 4w}}{2}. \quad (19)$$

In the context of the dynamic system theory, the necessary and sufficient condition for the convergence of system (16) is that magnitudes of λ_1 and λ_2 are less than 1 [15, 30]. Therefore, system (16) converges, if and only if

$$\max\{|\lambda_1|, |\lambda_2|\} < 1. \quad (20)$$

As it appears from (19) that λ_1 and λ_2 are two real or complex numbers, we will discuss two cases where λ_1 and λ_2 are two real and complex numbers separately to analyze the convergence of system (16).

(1) The first case is where $\lambda_{1,2}$ are two complex numbers, denoted as $\lambda_{1,2} \in \mathbb{C}$.

Lemma 2. For system (16), $\lambda_{1,2} \in \mathbb{C}$, if and only if

$$\begin{aligned} w - 2\sqrt{w} + 1 &< \phi < w + 2\sqrt{w} + 1, \\ w &\geq 0. \end{aligned} \quad (21)$$

Proof. For system (16), it is obvious that

$$\begin{aligned} \lambda_{1,2} \in \mathbb{C} &\iff \\ (1+w-\phi)^2 - 4w &< 0. \end{aligned} \quad (22)$$

Solving the right-hand side of (22) by classical approach, Lemma 2 can be easily proven. $\square$

FIGURE 2: Convergent region of SASPSO 2011 when $\lambda_{1,2} \in \mathbb{C}$.

Now, let us find conditions on ϕ and w guaranteeing the convergence of system (16) in the case where $\lambda_{1,2} \in \mathbb{C}$. It is trivial that the system converges if and only if $\max\{|\lambda_1|, |\lambda_2|\} < 1$.

Lemma 3. System (16) converges in the case where $\lambda_{1,2} \in \mathbb{C}$, if and only if

$$\begin{aligned} w - 2\sqrt{w} + 1 &< \phi < w + 2\sqrt{w} + 1, \\ 0 &\leq w < 1. \end{aligned} \quad (23)$$

Proof. Note that the magnitude of a complex number Z can be calculated as $|Z| = \sqrt{Z_r^2 + Z_c^2}$, where Z_r and Z_c denote the real and imaginary parts of Z . Hence, for $\lambda_{1,2} \in \mathbb{C}$, it is clear that

$$\max\{|\lambda_1|, |\lambda_2|\} = |\lambda_1| = |\lambda_2| = \sqrt{w}. \quad (24)$$

Therefore

$$\begin{aligned} \max\{|\lambda_1|, |\lambda_2|\} &< 1 \iff \\ \sqrt{w} &< 1. \end{aligned} \quad (25)$$

For $\lambda_{1,2} \in \mathbb{C}$, according to Lemma 2, (21) must be held. Hence, when considering conditions that $\lambda_{1,2} \in \mathbb{C}$ and $\max\{|\lambda_1|, |\lambda_2|\} < 1$, for $\lambda_{1,2} \in \mathbb{C}$, system (16), that is, SASPSO 2011, converges, if and only if

$$\begin{aligned} w - 2\sqrt{w} + 1 &< \phi < w + 2\sqrt{w} + 1, \\ 0 &\leq w < 1. \end{aligned} \quad (26)$$

$\square$

In the case where $\lambda_{1,2} \in \mathbb{C}$, the convergent region of SASPSO 2011 is shown in Figure 2.

(2) The second case is where λ_1 and λ_2 are two real numbers, denoted as $\lambda_{1,2} \in \mathbb{R}$.

Lemma 4. For system (16), $\lambda_{1,2} \in \mathbb{R}$, if and only if

$$\begin{aligned} \phi &\in \mathbb{R}, \quad w < 0, \\ \phi &\leq w - 2\sqrt{w} + 1, \\ \text{or } \phi &\geq w + 2\sqrt{w} + 1, \\ w &\geq 0. \end{aligned} \quad (27)$$

Proof. For system (16), it is clear that

$$\begin{aligned} \lambda_{1,2} \in \mathbb{R} &\iff \\ (1 + w - \phi)^2 - 4w &\geq 0. \end{aligned} \quad (28)$$

Solving the right-hand side of (28) by classical approach, Lemma 4 can be easily proven. $\square$

Then, let us find conditions on ϕ and w guaranteeing the convergence of system (16) in the case where $\lambda_{1,2} \in \mathbb{R}$. According to (19) and (20), for $\lambda_{1,2} \in \mathbb{R}$, $\max\{|\lambda_1|, |\lambda_2|\} < 1$ holds, if and only if

$$-1 < \frac{1 + w - \phi \pm \sqrt{(1 + w - \phi)^2 - 4w}}{2} < 1. \quad (29)$$

Thus

$$-3 - w + \phi < \pm \sqrt{(1 + w - \phi)^2 - 4w} < 1 - w + \phi. \quad (30)$$

As $\lambda_{1,2} \in \mathbb{R}$, it is clear that

$$\begin{aligned} (29) \\ \iff \\ -3 - w + \phi &< -\sqrt{(1 + w - \phi)^2 - 4w}, \\ \sqrt{(1 + w - \phi)^2 - 4w} &< 1 - w + \phi. \end{aligned} \quad (31)$$

Solving the right-hand inequalities in (31) yields

$$\begin{aligned} (29) \\ \iff \\ 2w + 2 - \phi &> 0, \\ \phi &> 0. \end{aligned} \quad (32)$$

According to Lemma 4, (27) must hold in the case where $\lambda_{1,2} \in \mathbb{R}$. Considering both conditions that $\lambda_{1,2} \in \mathbb{R}$ and $\max\{|\lambda_1|, |\lambda_2|\} < 1$, for $\lambda_{1,2} \in \mathbb{R}$, system (16) converges, if and only if

$$\begin{aligned} 0 < \phi < 2w + 2, \quad -1 < w < 0, \\ 0 < \phi \leq w - 2\sqrt{w} + 1, \\ \text{or } w + 2\sqrt{w} + 1 \leq \phi < 2w + 2, \\ 0 \leq w < 1. \end{aligned} \quad (33)$$

FIGURE 3: Convergent region of SASPSO 2011.

Then, considering both cases where $\lambda_{1,2} \in \mathbb{C}$ and $\lambda_{1,2} \in \mathbb{R}$ together, system (16), that is, SASPSO 2011, converges, if and only if

$$\begin{aligned} 0 < \phi < 2w + 2, \\ -1 < w < 1. \end{aligned} \quad (34)$$

Figure 3 illustrates the convergent region of SASPSO 2011 in both cases where $\lambda_{1,2} \in \mathbb{R}$ and $\lambda_{1,2} \in \mathbb{C}$. Only if any parameter selection of w and ϕ locates in this triangle area as shown in Figure 3 does SASPSO 2011 guarantee the trajectory convergence of particles. Figure 4 shows a 3-dimensional representation of the value of $\max\{|\lambda_1|, |\lambda_2|\}$.

Now, let us find the equilibrium point of SASPSO 2011. Calculating limits on both sides of (16) yields

$$\begin{aligned} \lim_{t \rightarrow \infty} X(t + 1) &= w \lim_{t \rightarrow \infty} V(t) + \phi \lim_{t \rightarrow \infty} (P - X(t)), \\ \lim_{t \rightarrow \infty} V(t + 1) &= \lim_{t \rightarrow \infty} X(t) + \lim_{k \rightarrow \infty} V(t). \end{aligned} \quad (35)$$

When SASPSO 2011 converges, $\lim_{t \rightarrow \infty} X(t + 1) = \lim_{t \rightarrow \infty} X(t)$ and $\lim_{t \rightarrow \infty} V(t + 1) = \lim_{t \rightarrow \infty} V(t)$. Therefore, substituting these two equations into (35) yields

$$\lim_{t \rightarrow \infty} X(t) = P = \frac{\phi_1 X_p + \phi_2 X_g}{\phi_1 + \phi_2}, \quad (36)$$

$$\lim_{t \rightarrow \infty} V(t) = 0, \quad (37)$$

where X_p and X_g denote the personal best position of the particle and the global best position of the swarm, respectively.

Through the stability analysis, it can be concluded that if and only if the convergence condition given by (34) is satisfied, the trajectories of particles can converge to the equilibrium point given by (36). Here, note that only the trajectory convergence of the particle can be assessed through the stability analysis. The global or local optimality of the equilibrium point cannot be assessed through the stability analysis.

(a) The case where $0 \leq w < 1$ (b) The case where $-1 < w < 1$ FIGURE 4: 3D representation of the value of $\max\{|\lambda_1|, |\lambda_2|\}$.

3.2.3. Verification of the Convergence Condition. As stated in Section 3.1, only if a stochastic algorithm satisfies the algorithm and convergence conditions can it be considered as local convergence algorithm; namely, the local optimality of the algorithm can be at least guaranteed. In Section 3.2.1, it is proven that SASPSO 2011 satisfies the algorithm condition. Therefore, we still need to assess that SASPSO 2011 satisfies the convergence condition in order to investigate the local convergence of the algorithm.

Before proving that SASPSO 2011 satisfies the convergence condition described in Section 3.1, we first reproduce some notations presented in [15, 29] for convenience. Let X_0 denote the worst particle in a swarm with NP particles. In a minimization optimization problem, the worst particle is the one with the largest cost function f in the swarm. Hence, X_0 can be defined as $X_0 = \arg\max\{f(X_i)\}$, where $1 \leq i \in \mathbb{N} \in$

NP . From the worst particle, the convex compact set L_0 can be defined as the convex compact set in which all points have a fitness value smaller than or equal to $f(X_0)$ [15, 29]:

$$L_0 = \{X \in S : f(X) \leq f(X_0)\}. \quad (38)$$

By the definition of L_0 , it can be assessed that X_p and X_g in (36) are in L_0 . Since L_0 is convex and X_p and X_g are in L_0 , any point along the line connecting X_p and X_g is thus in L_0 . Since the equilibrium point given by (36) is on the line connecting X_p and X_g , we can conclude that the equilibrium point given by (36) is in L_0 . Moreover, in Section 3.2.2, it is proven that the trajectories of particles in SASPSO 2011 converge to the equilibrium point given by (36), if and only if the convergent condition given by (34) is satisfied. Although the optimality of this equilibrium point cannot be stated from

this proof, it allows us at least to conclude that, regardless of the initial position of the particle, the particle can converge to the equilibrium point given by (36), which is in L_0 . In other terms, SASPSO 2011 can always generate a new point that can be sampled arbitrarily close to the point given by (36), which is in L_0 . Thus, we can conclude that there always exists a nondegenerated sampling hypersphere which guarantees that a new point can be sampled arbitrarily close to the equilibrium point given by (36), which is in L_0 .

As stated in Section 2.1, the stagnation is prevented in SPSO 2011 by adding an arbitrary point $X'_i(t)$ to the particle's velocity as given in (2) [22, 23]. Since the same moving rules defined in SPSO 2011 are used in SASPSO 2011, SASPSO 2011 is a nonstagnant algorithm. Considering the nonstagnation property of SASPSO 2011, the algorithm can always improve the fitness of the global best position X_g with a nonzero probability. Thus, we can conclude that, given any starting point in L_0 , SASPSO 2011 guarantees a nondegenerate sampling volume with a nonzero probability of sampling a point closer to the optimality region R_ξ , as described by the convergence condition in Section 3.1. Hence, it is sufficient to conclude that SASPSO 2011 satisfies the convergence condition. Note that the authors in [15] used the same method to prove that their PSO algorithm satisfies the convergence condition. Since SASPSO 2011 satisfies both the algorithm and convergence conditions, it is a locally convergent algorithm.

3.3. Convergence Behavior of Particles in SASPSO 2011. Before particles converge to the equilibrium point given in (36), they may oscillate in different ways around the equilibrium point as a result of different values of w and ϕ . Since different convergence oscillations may influence the quality of the final solution searched by particles [17, 18], it is necessary to investigate the oscillation behavior of particles. Four typical oscillations of particles in SASPSO 2011 are shown in Figure 5.

Nonoscillatory convergence behavior, as shown in Figure 5(a), leads particles to only search on one side of the equilibrium point, which may be useful when the search space is bounded. Particles exhibit nonoscillatory convergence when λ_1 and λ_2 are two real roots and at least one of them is positive, which is equivalent to $0 \leq (1+w-\phi)^2 - 4w$ and $0 < 1+w-\phi$. Harmonic oscillation, as demonstrated in Figure 5(b), may be beneficial in the exploitation stage, since particles smoothly oscillate around the equilibrium point. Harmonic oscillation behavior occurs when λ_1 and λ_2 are complex; that is, $(1+w-\phi)^2 - 4w < 0$. Zigzagging convergence, as illustrated in Figure 5(c), may also facilitate exploitation as particles zigzag around the equilibrium point, which may be suitable for solving optimization problems with rugged search spaces. Particles display zigzagging convergence behavior when at least one of λ_1 and λ_2 has a negative real part; that is, $w < 0$ or $1+w-\phi < 0$. The combined harmonic behavior with zigzagging behavior, as visualized in Figure 5(d), may be beneficial for the transition from exploration to exploitation due to its mixed nature. The combined harmonic behavior with zigzagging behavior emerges when at least one of the two complex roots λ_1 and λ_2 has a negative real part; that

is, $(1+w-\phi)^2 - 4w < 0 \cap w < 0$ or $(1+w-\phi)^2 - 4w < 0 \cap 1+w-\phi < 0$.

If the boundaries of the coefficients associated with these oscillations are known beforehand, one may easily design an adaptive method to change values of these coefficients, so that the convergence of PSO can be guaranteed and the quality of the final solution found by particles could be improved.

3.4. Parameter Selection Principle for SASPSO 2011. In Section 3.2.2, it is proven that if and only if the convergent condition given by (34) is satisfied, SASPSO 2011 can converge to the equilibrium point given by (36). Now, we need to answer how to set the initial and final values of w , ϕ_1 , and ϕ_2 to guarantee the convergence of SASPSO 2011. Please note that, without loss of generality, the subscript i is omitted from w , ϕ_1 , and ϕ_2 for simplicity.

Lemma 5. *SASPSO 2011 converges, only if the initial and final values of w , ϕ_1 , and ϕ_2 satisfy the following:*

$$\begin{aligned} 0 < \phi_{1s} + \phi_{1f} &< 6w_f + 6, \\ -1 < w_f &< w_s < 1, \end{aligned} \quad (39)$$

$$\phi_{1s} = \phi_{2f} > \phi_{1f} = \phi_{2s} > 0.$$

Proof. Since $\phi = (\phi_1 + \phi_2)/3$, the necessary and sufficient condition given by (34) can be rewritten as

$$\begin{aligned} 0 < \phi_1 + \phi_2 &< 6w + 6, \\ -1 < w &< 1. \end{aligned} \quad (40)$$

When $\phi_{1s} = \phi_{2f} > \phi_{1f} = \phi_{2s} > 0$, it appears from (6), (7), (9), and (10) that $\phi_1 + \phi_2 = \phi_{1s} + \phi_{1f} > 0$ for any particle at any iteration in SASPSO 2011. From (6) to (8), it is clear that $w_f \leq w \leq w_s$, $\phi_{1f} \leq \phi_1 \leq \phi_{1s}$, and $\phi_{2s} \leq \phi_2 \leq \phi_{2f}$ for any particle at any iteration in SASPSO 2011. Therefore

$$\begin{aligned} 0 < \phi_{1s} + \phi_{1f} &< 6w_f + 6, \\ -1 < w_f &< w_s < 1, \\ \phi_{1s} = \phi_{2f} &> \phi_{1f} = \phi_{2s} > 0, \\ \Downarrow \\ 0 < \phi_1 + \phi_2 &< 6w + 6, \\ -1 < w &< 1. \end{aligned} \quad (41)$$

Since the right-hand side of (41) is the necessary and sufficient condition for the convergence of SASPSO 2011, Lemma 5 can be easily proven based on the relationship given by (41). $\square$

Since w_s , w_f , ϕ_{1s} , ϕ_{1f} , ϕ_{2s} , and ϕ_{2f} are predefined parameters, the convergent condition given by (39) can be easily satisfied by setting proper values of these parameters. Figure 6 shows the convergent position and velocity trajectories of the particle in SASPSO 2011 under the suggested parameter selection: $w_s = 0.9$, $w_f = 0.1$, $\phi_{1s} = \phi_{2f} = 2.5$, and $\phi_{1f} = \phi_{2s} = 0.1$.

FIGURE 5: Convergence behavior of particles in SASPSO 2011.

4. Applying SASPSO 2011 for Solving COPs

4.1. Modeling of COPs. Generally, a constrained optimization problem is to optimize an objective function under some equality/inequality constraints, which can be mathematically presented as

$$\text{minimize: } f(x), \quad (42)$$

$$\text{subject to } h_{k_1}(x) = 0, \quad k_1 = 1, 2, \dots, m_1, \quad (43)$$

$$g_{k_2}(x) \leq 0, \quad k_2 = 1, 2, \dots, m_2, \quad (44)$$

$$x_k^l \leq x_k \leq x_k^u, \quad k = 1, 2, \dots, n, \quad (45)$$

where f_x is the objective function; $x = [x_1, x_2, \dots, x_n]$ is the n -dimensional vector of decision variables; $h_{k_1}(x)$ and $g_{k_2}(x)$ denote the k_1 th equality and k_2 th inequality constraints,

where $k_1 = 1, 2, \dots, m_1$ and $k_2 = 1, 2, \dots, m_2$; m_1 and m_2 are the number of equality and inequality constraints, respectively; x_k^l and x_k^u represent the lower and upper bounds of x_k .

4.2. Constraint Handling Technique. As shown in (43)–(45), COPs are usually constrained by some conditions. To easily and efficiently solve COPs, the task of how to handle constraints of COPs must be addressed. Among the existing handling constraint techniques, the penalty function method may be the most common approach [9]. By adding penalty terms to the objective function, the penalty function method can transform a constrained optimization problem into an unconstrained one. The drawback of the method is that determining proper values for the penalty factors is time-consuming and problem-dependent, which requires the previous experience of the users [31].

FIGURE 6: Convergent position and velocity trajectories of the particle in SASPSO 2011 under the suggested parameter selection.

To overcome the shortcoming of the penalty function method and increase the diversity of solutions, the adaptive relaxation method [4, 5], which is integrated with the feasibility-based rule [24, 25], is applied to handle constraints of COPs and evaluate candidate solutions in this paper.

In the adaptive relaxation method, in order to handle equality and inequality constraints of COPs, the total constraint violation value of particle i is first calculated as follows [4, 5]:

$$\text{viol}_i = \sum_{k_1=1}^{m_1} |h_{k_1}(x)| + \sum_{k_2=1}^{m_2} \max(0, g_{k_2}(x)), \quad (46)$$

where h_{k_1} , g_{k_2} , k_1 , k_2 , m_1 , and m_2 have the same definitions as those in (43) and (44).

After calculating the total constraint violation value of each particle, the median of total constraint violation values of all particles is assigned to the initial constraint relaxation value σ . If the total constraint violation value of a particle is less than σ , the particle is temporarily considered as a feasible solution; otherwise, it is temporarily considered as a nonfeasible solution. During the evolution, σ is gradually reduced according to the fraction of feasible particles (F_F) with respect to the relaxed constraints. Given a swarm with NP particles, from iteration t to iteration $t + 1$, σ is updated as follows [4]:

$$\sigma(t+1) = \sigma(t) \left(1 - \frac{F_F}{NP}\right). \quad (47)$$

From (47), it is clear that more nonfeasible particles are allowed into the next iteration in the early phase of the evolution, which can consequently add some diversification to the swarm [4]. With the evolution continuing, the relaxation value σ adaptively decreases when more and more feasible solutions are found. Hence, the feasible region gradually reduces until it converges to the real feasible region, which may increase the possibility of finding optimal solutions [5].

Since σ is adaptively updated, no additional penalty factor is needed in the adaptive relaxation method, which can thus reduce the optimization difficulties [5].

After calculating the fitness and constraint violation values of each particle, the feasibility-based rule [24, 25] is applied to evaluate and select the elite solution between any two candidate solutions. The feasibility-based rule can be described as follows: (1) for any two solutions with same constraint violation value, the solution with better fitness value is preferred; (2) for any two solutions with different constraint violation values, the solution with smaller constraint violation value is preferred.

As the fitness and constraint violation information is considered separately in the feasibility-based rule, no additional parameter is needed when using this rule, which can decrease the optimization difficulties [24, 25]. Moreover, although the nonfeasible solutions violate some constraints, they may also contain some information that is useful for finding good solutions [5]. When this useful information is considered, the likelihood of finding high-quality solutions may be increased. This is one main reason why the nonfeasible solutions are considered in the feasibility-based rule.

Except the equality and inequality constraints, each design variable must satisfy its boundary constraints, as shown in (45). When any variable x_k ($k = 1, 2, \dots, n$) cannot satisfy its boundary constraints, the saturation strategy is applied to modify x_k as follows [9]:

$$x_k = \begin{cases} x_k^l, & \text{if } x_k < x_k^l, \\ x_k^u, & \text{if } x_k > x_k^u, \\ x_k, & \text{otherwise,} \end{cases} \quad (48)$$

where x_k , x_k^l , and x_k^u have the same definitions as those in (45).

TABLE 1: The simulation parameters for different methods.

Method	Simulation parameters
SASPSO 2011	$w_s = 0.9, w_f = 0.1, \phi_{1s} = \phi_{2f} = 2.5, \phi_{1f} = \phi_{2s} = 0.1$
TVPSO [9]	$w_s = 0.7, w_f = 0.4, c_{1s} = 5c_{1f} = 2.5, c_{2f} = 5c_{2s} = 2.5$
SPSO 2011 [22]	$w = 1/2 \ln(2), \phi_1 = 1/2 + \ln(2), \phi_2 = 1/2 + \ln(2)$
SAIWPSO [21]	$w_0 = 0.729, w_{\max} = 1, w_{\min} = 0.1, \xi = 0.005$
FGIWPSO [28]	$w_s = 0.8, c_1 = 2.8, c_2 = 1.3$
RPSO [32]	$w = (1 + \text{rand}(1))/2, c_1 = 1.494, c_2 = 1.494$
CPSO [10]	$\phi = 4.1$

TABLE 2: Mathematical characteristics of 4 benchmark test functions.

Function	NV	LI	NI	LE	NE	Optimal f^*
G_{07}	10	3	5	0	0	24.30620
G_{09}	7	0	4	0	0	680.6301
G_{10}	8	3	3	0	0	7049.248
E_1	5	0	0	0	3	0.053950

4.3. *The SASPSO 2011-Based Framework for Solving COPs.* Let NP denote the size of the swarm and let $t_{\max}$ denote the maximum iteration number. The algorithmic scheme of applying SASPSO 2011 to solve COPs is illustrated in Algorithm 1.

5. Numerical Simulations and Analysis

To verify the proposed method, it is tested through 4 benchmark test functions and 2 real-world engineering problems against time-varying PSO (TVPSO) [9], constrained PSO (CPSO) [10], fine grained inertia weight PSO (FGIWPSO) [28], stability-based adaptive inertia weight PSO (SAIWPSO) [21], SPSO 2011 [22, 23], random PSO (RPSO) [32], and some well-established methods proposed in the literature. Note that the 4 benchmark test functions and 2 real-world engineering problems are listed in the Appendix.

In order to reduce the random discrepancy, a Monte Carlo experiment with 50 independent runs is conducted for each studied problem. After 50 independent runs, the best, mean, and worst results, as well as the standard deviation of each method, are examined and compared. In each run, the final solution of each method is obtained after 4000 iterations of 100 particles on MATLAB 2012B software on a Windows 8 personal computer with i3-2350 @ 2.30-GHz and 2 GB RAM. The simulation parameters for the different methods are shown in Table 1.

5.1. *Simulations on 4 Benchmark Test Functions.* This subsection shows the simulations results of all methods for the 4 benchmark test functions. The mathematical characteristics of the 4 benchmarks are shown in Table 2, where NV denotes the number of variables, LI and NI are, respectively, the numbers of linear and nonlinear inequality constraints, and LE and NE are, respectively, the numbers of linear and nonlinear equality constraints.

TABLE 3: Statistical results of different methods for test function G_{07} ($f^* = 24.3062$).

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	24.4549	25.3041	25.8132	0.4415
SAIWPSO	24.6763	25.5011	25.6714	0.3761
FGIWPSO	24.8848	25.7688	26.8060	0.5510
TVPSO	24.9948	26.1300	27.2804	0.6741
SPSO 2011	25.4362	26.7785	28.2169	0.7726
CPSO	26.0603	27.3082	28.8190	0.7756
RPSO	26.4620	27.7619	29.3009	0.7804
Krohling and Coelho [33]	24.4810	25.7780	27.5970	0.7767

TABLE 4: Statistical results of different methods for test function G_{09} ($f^* = 680.6301$).

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	680.6403	680.6961	681.6312	0.0169
SAIWPSO	680.7037	680.7978	681.6101	0.0157
FGIWPSO	680.7731	680.9701	681.7842	0.0184
TVPSO	680.7796	681.2985	682.0132	0.0208
SPSO 2011	680.7941	681.3764	682.6756	0.0194
CPSO	680.8308	681.7771	682.7671	0.0291
RPSO	680.9414	682.1002	683.2115	0.0286
Krohling and Coelho [33]	680.7390	680.9290	681.5900	0.0196

After performing 50 independent runs, the best, mean, worst, and standard deviation results of all methods for the four test functions are summarized in Tables 3–6, respectively. Note that, as shown in Tables 3–6, the statistical results of Krohling and Coelho [33] and Mezura-Montes and Coello Coello [34] are extracted from their corresponding references. In all cases, the best results with respect to the best, mean, and worst values and standard deviation are highlighted in boldface in these tables. Figure 7 displays the fitness curves of the best results of all PSO methods for the four test functions.

From Tables 3–6, it is evident that SASPSO 2011 and SAIWPSO are the first and second most efficient algorithms in terms of the average solution, which implies that SASPSO

```

(1) Set simulation parameters and randomly generate an initial swarm
(2) Obtain  $X_g, X_{pi}$  and  $\sigma$  at the initial iteration
(3) while  $t \leq t_{\max}$  do
(4)    $F_F = 0$  % set the number of feasible solutions to be 0
(5)   for  $i = 1 : NP$  do
(6)      $G_i(t) \leftarrow X_i(t) + (\phi_{i1}[X_{pi}(t) - X_i(t)] + \phi_{i2}[X_g(t) - X_i(t)])/3$  % calculate center of  $G_i(t)$ 
(7)     Randomly generate  $X'_i(t)$  within the hypersphere  $H(G_i(t), \|G_i(t) - X_i(t)\|)$ 
(8)      $V_i(t) \leftarrow w_i V_i(t) + X'_i(t) - X_i(t)$  % update velocity of particle  $i$ 
(9)      $X_i(t) \leftarrow V_i(t) + X_i(t)$  % update position of particle  $i$ 
(10)    Modify each dimension of  $X_i(t)$  by the saturation strategy given by (48)
(11)     $\text{viol}_i \leftarrow \sum_{k_1=1}^{m_1} |h_{k_1}(x)| + \sum_{k_2=1}^{m_2} \max(0, g_{k_2}(x))$  % calculate constraint violation
(12)    if  $\text{viol}_i \leq \sigma(t)$  do
(13)      Particle  $i$  is feasible and  $F_F = F_F + 1$ 
(14)    else
(15)      Particle  $i$  is non-feasible and  $F_F = F_F$ 
(16)    end if
(17)    Calculate fitness value  $f(X_i(t))$  of particle  $i$ 
(18)    Update the personal best position  $X_{pi}(t)$  of particle  $i$  by the feasibility-based rule
(19)  end for
(20)  Update the global best position  $X_g(t)$  of the swarm by the feasibility-based rule
(21)  for  $i = 1 : NP$  do
(22)     $w_i(t) \leftarrow (w_s - w_f) \exp(-\delta_w t / \beta_i) + w_f$  % update  $w_i$ 
(23)     $\phi_{i1}(t) \leftarrow (\phi_{1s} - \phi_{1f}) \exp(-\delta_{\phi_1} t / \beta_i) + \phi_{1f}$  % update  $\phi_{i1}$ 
(24)     $\phi_{i2}(t) \leftarrow (\phi_{2s} - \phi_{2f}) \exp(\delta_{\phi_2} t / \beta_i) + \phi_{2f}$  % update  $\phi_{i2}$ 
(25)  end for
(26)   $\sigma(t) \leftarrow \sigma(t)(1 - F_F/NP)$  % update the relaxation value
(27)   $t = t + 1$ 
(28) end while
(29) Output  $X_g(t)$ 

```

ALGORITHM 1: The framework of applying SASPSO 2011 for solving COPs.

TABLE 5: Statistical results of different methods for test function G_{10} ($f^* = 7049.248$).

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	7051.4559	7057.7333	7065.1077	7.9605
SAIWPSO	7053.3778	7059.3314	7064.2103	7.2147
FGIWPSO	7054.4426	7062.0209	7068.5618	8.8144
TVPSO	7056.0332	7064.9206	7072.7052	9.7440
SPSO 2011	7101.1337	7106.8628	7114.5401	8.7865
CPSO	7105.8172	7113.2698	7119.1772	8.0916
RPSO	7132.5572	7137.0463	7142.4531	8.1709
Krohling and Coelho [33]	7095.5000	7350.4000	7831.8000	182.13

TABLE 6: Statistical results of different methods for test function E_1 ($f^* = 0.053950$).

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	0.053972	0.059571	0.061217	$1.01E - 02$
SAIWPSO	0.057135	0.060125	0.061013	$9.72E - 03$
FGIWPSO	0.061032	0.060633	0.062784	$1.32E - 02$
TVPSO	0.062526	0.062876	0.064874	$1.33E - 02$
SPSO 2011	0.062526	0.065209	0.067223	$1.44E - 02$
CPSO	0.066071	0.068993	0.071229	$1.48E - 02$
RPSO	0.076139	0.078258	0.080519	$1.25E - 02$
Mezura-Montes and Coello Coello [34]	0.053986	0.166385	0.468294	$1.77E - 01$

2011 and SAIWPSO generally outperform the other algorithms in solving these 4 test functions. It can be also seen from these tables that the proposed algorithm can find the best solutions for the 4 benchmark test functions, and SAIWPSO can find the best worst solutions for G_{07} , G_{10} , and E_1 .

Moreover, it is clear from Tables 3–6 that SAIWPSO and SASPSO 2011 are the most stable and second most stable algorithms, since they can provide the least and second least standard deviations in solving the 4 benchmark test

functions. The reason why our proposed algorithm and SAIWPSO are more robust than the other algorithms may be that these two algorithms are stability-based algorithms. By setting proper values of the three control parameters, the stability and convergence of these two algorithms can be guaranteed. It is notable from Tables 3–6 that the difference of the standard deviation between our proposed method and SAIWPSO is not significant for each test function.

FIGURE 7: The best fitness curves of different PSO algorithms for the four benchmark test functions.

Therefore, summarizing the simulation results shown in Tables 3–6, it can be concluded that the proposed method is highly competitive in solving the 4 benchmarks in terms of the solution quality. Moreover, the search robustness of the proposed method is comparable with that of SAIWPSO and outperforms those of the other methods.

5.2. Application on Two Real-World Engineering Problems. To further verify the proposed method, it is applied to solve two practical engineering problems: the tension compression spring design problem and the three-bar truss design

problem. Note that the performance of the proposed algorithm is still compared with those of the aforementioned PSO algorithms and some other well-known methods proposed in the literature.

5.2.1. Tension Compression Spring Design Problem. The statistical results of different methods for the tension compression spring design problem are shown in Table 7. Note that, in this table, the statistical results of PSO-DE [20] and CDE [35] are extracted from their corresponding literature. Table 8 exhibits the details of the best solution searched by each

FIGURE 8: The best fitness curves of different PSO algorithms for tension compression spring design problem.

TABLE 7: Statistical results of different methods for the tension compression spring design problem.

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	0.01266523	0.01266524	0.01266527	$1.13E-08$
SAIWPSO	0.01266530	0.01270133	0.01275011	$1.03E-08$
FGIWPSO	0.01272058	0.01273121	0.01278633	$1.83E-07$
TVPSO	0.01272058	0.01274328	0.01282146	$2.81E-07$
SPSO 2011	0.01305451	0.01327612	0.01377532	$4.73E-07$
CPSO	0.01306812	0.01347413	0.01383239	$5.15E-07$
RPSO	0.01314291	0.01362154	0.01402336	$5.49E-07$
PSO-DE [20]	0.01266523	0.01266524	0.01266530	$1.20E-08$
CDE [35]	0.01267020	0.01270300	0.01279000	$2.70E-05$

method for this problem. The best fitness curves of different PSO algorithms are visualized in Figure 8.

From Table 7, it can be seen that SASPSO 2011 and PSO-DE [20] provide the best performance in terms of the average solution, and SAIWPSO ranks the second in terms of the average solution. From this table, it is also interesting to find that SASPSO 2011 is the best one in terms of the best and worst solutions. Therefore, it can be concluded that our proposed method performs similar to PSO-DE [20] and slightly outperforms the other methods in terms of the solution quality. Moreover, according to the standard deviation of each method, as shown in Table 7, it is apparent that SAIWPSO and SASPSO 2011 are the most robust and second most robust algorithms in solving this problem.

From Table 8, it is worth noting that the best solutions searched by all methods are feasible, since the design variables x_1 , x_2 , and x_3 and constraints g_1 , g_2 , and g_3 of these best solutions satisfy the corresponding constraints of the tension compression spring design problem. This, to some extent,

FIGURE 9: The best fitness curves of different PSO algorithms for the three-bar truss design problem.

reflects the feasibility of these methods in solving the tension compression spring design problem.

5.2.2. Three-Bar Truss Design Problem. The statistical results of different methods for the three-bar truss design problem are shown in Table 9, in which the statistical results of Ray and Liew [36] and CSA [37] are obtained from their corresponding literature. Table 10 shows the information of the best solution of each method for this design problem. The fitness curves of the best results searched by all PSO algorithms are exhibited in Figure 9.

It is trivial from Table 9 that SASPSO 2011 and SAIWPSO are the most sufficient and second most sufficient algorithms in finding the best, mean, and worst solutions. Therefore, we can conclude that our proposed method is highly competitive in solving this problem in terms of the solution quality. Additionally, it is notable from Table 9 that SAIWPSO and SASPSO 2011 provide the least and second least standard deviations in solving this problem, which indicates that these two algorithms are more stable than the other algorithms in solving this problem. Besides, it can be found from Table 10 that the best solution searched by each method satisfies the constraints of the three-bar truss design problem, which, to a certain degree, reflects the feasibility of each method in solving this problem.

6. Conclusion and Future Work

In this study, a modified PSO algorithm, called SASPSO 2011, is proposed based on SPSO 2011. In order to enhance the performance of SASPSO 2011, a novel self-adaptive strategy is developed for updating the three control parameters of particles in the algorithm. By fine-tuning the three control

TABLE 8: The best results of different methods for tension compression spring design problem.

Methods	Variables and constraints							Cost function
	x_1	x_2	x_3	g_1	g_2	g_3	g_4	
SASPSO 2011	0.05169793	0.35693118	11.27646560	$-2.3227E-07$	$-8.3900E-09$	-4.0542	-0.7276	0.01266523
SAIWPSO	0.05170333	0.35819358	11.20296603	$-2.4978E-07$	$-3.5545E-11$	-4.0567	-0.7267	0.01266530
FGIWPSO	0.05000000	0.31740420	14.03076460	$-1.2951E-05$	$-5.2611E-05$	-3.9680	-0.7551	0.01272058
TVPSO	0.05000000	0.31735633	14.04028984	$-2.3923E-04$	$-1.7130E-04$	-3.9662	-0.7551	0.01272621
SPSO 2011	0.05168856	0.35421966	11.79428645	$-2.3006E-02$	$-5.6059E-03$	-3.9057	-0.7294	0.01305451
CPSO	0.05615078	0.47379809	6.747988161	$-5.7669E-03$	$-1.0669E-04$	-4.2061	-0.6467	0.01306812
RPSO	0.05682401	0.49270160	6.261210792	$-5.7310E-04$	$-1.0206E-03$	-4.2508	-0.6336	0.01314291
PSO-DE [20]	0.05168881	0.35671170	11.28931993	$-1.9923E-09$	$-3.9338E-09$	-4.0538	-0.7277	0.01266523
CDE [35]	0.05160900	0.35471400	11.41083100	$-3.8642E-05$	$-1.8289E-04$	-4.0486	-0.7291	0.01267020

TABLE 9: Statistical results of different methods for the three-bar truss design problem ("NA" means unknown).

Methods	Best	Mean	Worst	Std. dev.
SASPSO 2011	263.89584	263.89613	263.89653	$3.8E-05$
SAIWPSO	263.89610	263.89932	263.95441	$9.4E-04$
FGIWPSO	263.89612	263.90113	263.97956	$4.9E-03$
TVPSO	263.89671	264.00121	264.76125	$7.2E-03$
SPSO 2011	263.89686	264.21552	264.80231	$3.5E-02$
CPSO	263.89724	264.55263	264.34219	$5.6E-02$
RPSO	263.89926	264.78712	265.35781	$8.9E-02$
Ray and Liew [36]	263.89584	263.90335	263.96976	$1.3E-02$
CSA [37]	263.97156	264.06690	NA	$9.0E-05$

parameters, the exploration and exploitation capabilities of SASPSO 2011 can be well balanced. Consequently, SASPSO 2011 can promote particles to search high-quality solutions. Since the convergence of PSO remains an important issue in the context of PSO development, this study first theoretically investigates the local convergence of SASPSO 2011. Afterwards, a parameter selection principle that can guarantee the local convergence of SASPSO 2011 is provided. Finally, based on the SASPSO 2011 algorithm developed, this paper completes the design of a framework for solving COPs. For adding some diversification to the swarm and decreasing optimization difficulties, the adaptive relaxation method [4, 5] is integrated with the feasibility-based rule [24, 25] to handle constraints of COPs and evaluate candidate solutions in our developed framework.

The proposed method is verified by 4 benchmark test functions and 2 real-world applications against six state-of-the-art PSO variants and several well-known methods presented in the literature. The simulation results show that the proposed method is highly competitive in finding high-quality solutions. Furthermore, the search robustness of the proposed method is comparable with that of SAIWPSO [21] and outperforms those of the other compared methods. Hence, the proposed method can be considered as an efficient optimization approach to solve COPs.

There are some issues that deserve some future study. Since the three control parameters of PSO also determine the convergence speed of PSO, the first issue is how to

theoretically analyze the sensitivity of the convergence speed of SASPSO 2011 to its three control parameters. Another issue that deserves some future study is to modify SASPSO 2011, so that the global convergence of the algorithm can be achieved. Moreover, more advanced mechanisms can be developed for better updating the three control parameters of SASPSO 2011, so that the search robustness of the algorithm can be further enhanced. One potential way is to follow the idea proposed by Taherkhani and Safabakhsh [21] to update each control parameter in different dimensions. Last but not least, we are also considering the possibilities of developing some new constraint handling techniques, since how to handle constraints of COPs influences the performance of the obtained solution.

Appendix

Benchmark Test Functions

(1) The test function G_{07} consists of minimizing

$$\begin{aligned}
 f(x) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\
 &\quad + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\
 &\quad + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\
 &\quad + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\
 &\quad + (x_{10} - 7)^2 + 45,
 \end{aligned}$$

$$\begin{aligned}
 \text{subject to } g_1(x) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \\
 &\leq 0,
 \end{aligned}$$

$$\begin{aligned}
 g_2(x) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \\
 &\quad - 120 \leq 0,
 \end{aligned}$$

$$g_3(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0,$$

TABLE 10: The best results of different methods for the three-bar truss design problem.

Methods	Variables and constraints					Cost function
	x_1	x_2	g_1	g_2	g_3	
SASPSO 2011	0.78860050	0.40845943	$-1.2103E-08$	-1.4639	-0.5361	263.89584
SAIWPSO	0.78815418	0.40972433	$-4.4177E-07$	-1.4624	-0.5376	263.89610
FGIWPSO	0.78835591	0.40915400	$-1.5623E-06$	-1.4631	-0.5369	263.89612
TVPSO	0.78772057	0.41095691	$-1.5002E-06$	-1.4610	-0.5390	263.89671
SPSO 2011	0.78911058	0.40702683	$-6.6720E-06$	-1.4655	-0.5345	263.89686
CPSO	0.79005265	0.40436606	$-8.9503E-08$	-1.4685	-0.5315	263.89724
RPSO	0.78656794	0.41424254	$-1.0159E-06$	-1.4573	-0.5427	263.89926
Ray and Liew [36]	0.78862103	0.40840133	$-9.7611E-09$	-1.4639	-0.5361	263.89584
CSA [37]	0.78867000	0.40902000	$-2.9000E-04$	-0.2685	-0.7317	263.97156

$$\begin{aligned}
g_4(x) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0, \\
g_5(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0, \\
g_6(x) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0, \\
g_7(x) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0, \\
g_8(x) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4) + 3x_5^2 - x_6 - 30 \leq 0,
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
g_2(x) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0, \\
g_3(x) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0, \\
g_4(x) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0,
\end{aligned} \tag{A.2}$$

where $-10 \leq x_i \leq 10$ ($i = 1, 2, \dots, 7$) and the global optimum is $x^* = (2.3304, 1.9513, -0.4775, 4.3657, -0.6244, 1.0381, 1.5942)$ with $f(x^*) = 680.6301$ [33].

(3) The test function G_{10} is to minimize

$$f(x) = x_1 + x_2 + x_3,$$

$$\text{subject to } g_1(x) = -1 + 0.0025(x_4 + x_6) \leq 0,$$

$$g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0,$$

$$g_3(x) = -1 + 0.01(x_8 - x_5) \leq 0,$$

$$\begin{aligned}
g_4(x) &= -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0,
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
g_5(x) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0,
\end{aligned}$$

$$\begin{aligned}
g_6(x) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0,
\end{aligned}$$

where $-10 \leq x_i \leq 10$ ($i = 1, 2, \dots, 10$) and the global minimum is $x^* = (2.1719, 2.3636, 8.7739, 5.0959, 0.9906, 1.4305, 1.3216, 9.8287, 8.2800, 8.3759)$ with $f(x^*) = 24.30620$ [33].

(2) The test function G_{09} is to minimize

$$\begin{aligned}
f(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \\
&\quad + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 \\
&\quad - 4x_6x_7 - 10x_6 - 8x_7,
\end{aligned}$$

$$\begin{aligned}
\text{subject to } g_1(x) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \\
&\leq 0,
\end{aligned}$$

where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$ ($i = 2, 3$), $10 \leq x_i \leq 1000$ ($i = 4, \dots, 8$), and the global minimum is $x^* = (579.3167, 1359.943, 5110.71, 182.0714, 295.5985, 217.979, 286.4162, 395.5979)$ with $f(x^*) = 7049.248$ [33].

(4) The test function E_1 is to minimize

$$\begin{aligned} f(x) &= \exp(x_1 x_2 x_3 x_4 x_5), \\ \text{subject to } h_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5 - 10 \\ &= 0, \\ h_2(x) &= x_2 x_3 - 5 x_4 x_5 = 0, \\ h_3(x) &= x_1^3 + x_2^3 + 1 = 0, \end{aligned} \quad (\text{A.4})$$

where $-2.3 \leq x_i \leq 2.3$ ($i = 1, 2$), $-3.2 \leq x_i \leq 3.2$ ($i = 3, 4, 5$), and the global optimum is $x^* = (-1.171431, 1.595709, 1.827247, -0.7636413, -0.763645)$ with $f(x^*) = 0.05395$ [34].

(5) The tension compression design problem can be mathematically presented as follows [20, 35]:

$$\begin{aligned} \text{minimize: } f(x) &= (x_3 + 2) x_2 x_1^2, \\ \text{subject to } g_1(x) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0, \\ g_2(x) &= \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \\ &\leq 0, \\ g_3(x) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0, \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0, \end{aligned} \quad (\text{A.5})$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, and $2 \leq x_3 \leq 15$.

(6) The three-bar truss design problem can be mathematically presented as follows [37]:

$$\begin{aligned} \text{minimize: } f(x) &= (2\sqrt{2}x_1 + x_2) \times l, \\ \text{subject to } g_1(x) &= \frac{\sqrt{x_1} + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0, \\ g_2(x) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0, \\ g_3(x) &= \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0, \end{aligned} \quad (\text{A.6})$$

where $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $l = 100$ cm, and $P = \sigma = 2$ KN/cm².

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is financially supported by the Chinese National Science Foundation under Grant no. 11472213.

References

- [1] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.
- [2] Y.-K. Lin and C. S. Chong, "Fast GA-based project scheduling for computing resources allocation in a cloud manufacturing system," *Journal of Intelligent Manufacturing*, 2015.
- [3] D. N. Pattanayak, R. N. Chakrabarti, and M. Basu, "Economic environmental scheduling of hydrothermal power system," *Journal of The Institution of Engineers: Series B*, vol. 95, no. 4, pp. 329–336, 2014.
- [4] H. Zhang and G. P. Rangaiah, "An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization," *Computers and Chemical Engineering*, vol. 37, pp. 74–88, 2012.
- [5] Z. Wang, S. Li, and Z. Sang, "A new constraint handling method based on the modified Alopex-based evolutionary algorithm," *Computers & Industrial Engineering*, vol. 73, no. 1, pp. 41–50, 2014.
- [6] P. Yang, K. Tang, J. A. Lozano, and X. Cao, "Path planning for single unmanned aerial vehicle by separately evolving waypoints," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1130–1146, 2015.
- [7] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066–2081, 2013.
- [8] Y. Zhang, L. Wu, and S. Wang, "UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.
- [9] K. Khalili-Damghani, A.-R. Abtahi, and M. Tavana, "A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems," *Reliability Engineering & System Safety*, vol. 111, pp. 58–75, 2013.
- [10] E. Masehian and D. Sedighizadeh, "An improved particle swarm optimization method for motion planning of multiple robots," *Springer Tracts in Advanced Robotics*, vol. 83, pp. 175–188, 2012.
- [11] J. Sun, V. Palade, X.-J. Wu, W. Fang, and Z. Wang, "Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 222–232, 2014.
- [12] L. Xu, J. Wang, Y.-P. Li, Q. Li, and X. Zhang, "Resource allocation algorithm based on hybrid particle swarm optimization for multiuser cognitive OFDM network," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7186–7194, 2015.
- [13] R. Akbari and K. Ziarati, "A rank based particle swarm optimization algorithm with dynamic adaptation," *Journal of Computational and Applied Mathematics*, vol. 235, no. 8, pp. 2694–2714, 2011.
- [14] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

- [15] F. van den Bergh and A. P. Engelbrecht, "A convergence proof for the particle swarm optimiser," *Fundamenta Informaticae*, vol. 105, no. 4, pp. 341–374, 2010.
- [16] M. R. Bonyadi and Z. Michalewicz, "A locally convergent rotationally invariant particle swarm optimization algorithm," *Swarm Intelligence*, vol. 8, no. 3, pp. 159–198, 2014.
- [17] M. Bonyadi and Z. Michalewicz, "Analysis of stability, local convergence, and transformation sensitivity of a variant of particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 370–385, 2015.
- [18] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [19] M. Jiang, Y. P. Luo, and S. Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, no. 1, pp. 8–16, 2007.
- [20] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 629–640, 2010.
- [21] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Applied Soft Computing Journal*, vol. 38, pp. 281–295, 2016.
- [22] M. Clerc, Standard Particle Swarm Optimisation, <https://hal.archives-ouvertes.fr/hal-00764996>.
- [23] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: a baseline for future PSO improvements," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 2337–2344, Cancun, Mexico, June 2013.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [25] C.-L. Sun, J.-C. Zeng, and J.-S. Pan, "A modified particle swarm optimization with feasibility-based rules for mixed-variable optimization problems," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 6, pp. 3081–3096, 2011.
- [26] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, Nagoya, Japan, October 1995.
- [27] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, 1998.
- [28] P. Chauhan, K. Deep, and M. Pant, "Novel inertia weight strategies for particle swarm optimization," *Memetic Computing*, vol. 5, no. 3, pp. 229–251, 2013.
- [29] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 19–30, 1981.
- [30] Y. Zhang, X. Xiong, and Q. Zhang, "An improved self-adaptive PSO algorithm with detection function for multimodal function optimization problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 716952, 8 pages, 2013.
- [31] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1407–1422, 2007.
- [32] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the Congress on Evolutionary Computation*, pp. 94–100, May 2001.
- [33] R. A. Krohling and L. D. S. Coelho, "Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1407–1416, 2006.
- [34] E. Mezura-Montes and C. A. Coello Coello, "A simple multi-membered evolution strategy to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, 2005.
- [35] F.-Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [36] T. Ray and K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 386–396, 2003.
- [37] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.