An optimizer using swarm of chaotic dynamical particles

Yoshikazu Yamanaka\textsuperscript{1a}) and Tadashi Tsubone\textsuperscript{1b})

\textsuperscript{1} Department of Electronics, Electrical and Information Engineering, Nagaoka University of Technology, 1603-1 Kamitomioka, Niigata 940-2188, Japan

\textsuperscript{a}) y_yama@stn.nagaokaut.ac.jp
\textsuperscript{b}) tsubone@vos.nagaokaut.ac.jp

Received May 23, 2014; Revised September 8, 2014; Published January 1, 2015

Abstract: A novel optimization method called, “optimizer using swarm of chaotic dynamical particles,” (OSCDP) is proposed. The proposed method searches for an optimal solution on the basis of particles governed by chaotic dynamics. The chaotic dynamics make the particles behave in a complex way, by stretching and holding mechanisms, even though the dynamics do not contain any stochastic terms. The particles share information of their best position that is a candidate of the optimal solution, as with particle swarm optimization (PSO). The complex behavior and the sharing information mechanism enable the proposed method to search for the optimal solution. This method consists of only three system parameters, making it suitable for implementing the algorithm. We compared the proposed method with other deterministic PSOs and the standard PSO, and found that the proposed method exhibited a better searching performance.

Key Words: nonlinear dynamical systems, chaos, optimization, particle swarm optimization

1. Introduction

Particle swarm optimization, (PSO) is one of the most effective heuristic optimization methods. PSO is a non gradient-based method to search for an optimal solution that is characterized by the maximum or minimum value of fitness. It was developed by Kennedy and Eberhart in 1995 \cite{1, 2}. In 1998, it was modified to include an inertia weight by Shi and Eberhart \cite{3}. PSO algorithm successfully applied in many application areas \cite{4–8}. Even though PSO has been shown to perform well, one basic problem is that it is difficult to adequately understand how it works \cite{9}. A fully comprehensive mathematical PSO model is still not available for several reasons \cite{10}, one of which, as reported in \cite{10}, is that the stochastic factors prevent the use of standard mathematical tools. Some researchers have proposed deterministic PSO (DPSO) models that simplify PSO by removing the stochastic factors, in order to theoretically analyze the behavior of the particles \cite{9, 11–18}. These theoretical analysis derived parameter ranges which guarantee convergence of particles under some assumptions. These works demonstrate the importance of analyzing PSO dynamics in order to improve performance. Although deterministic PSO models are suitable to analyze the behavior of the particles, their searching performance is more limited than that of the standard PSO \cite{3}. Focusing on improving searching
performance, other deterministic optimizations are studied in earlier studies [19–23].

In this paper, we propose a novel optimization method based on chaotic dynamics following the same concept of our earlier studies [22, 23]. The proposed method is also a population-based method in which the particles have social relationships as with PSO. We utilize the chaotic dynamics for particles because the feature seems suitable for heuristic optimizers. The behavior is complex because the dynamics contains stretching and holding mechanisms, in other words, these are chaotic dynamics. This behavior helps the particles escape a local best position. The benefit of the proposed method is that it is deterministic system and performs better than PSO and DPSO. The results are shown in Sec. 6. Therefore, the proposed method can be more suitable than PSO and DPSO for uncovering the relationship between the dynamics of particles and its performance.

In previous works, some researchers embedded chaos into PSO instead of the system parameters and the stochastic elements, in order to improve performance [24–29]. Our proposed method is different from these approaches. In these methods, the parameters and the stochastic terms are simply replaced with chaotic elements, the dynamics of their particles are the same as the standard PSO. In our proposed method, particles are governed by the chaotic dynamics, and the particles therefore exhibit complex motion without any updating system parameters, stochastic terms or additional time-series data.

There are some researches that improved PSO performance with hybrid approaches, by combining PSO with an additional mechanism. In [30–35], chaos was inserted into the procedures of PSO algorithms. Shindo and Jin’no proposed a deterministic PSO with a resetting mechanism (RD-PSO) that forcibly pushes particles far away from the best searched position and confirmed that the mechanism improved the searching ability of the deterministic PSO [21]. Our proposed method is also different from these hybrid methods. These hybrid methods apply another mechanism to the standard PSO. Even though hybrid approaches improve searching performance, it makes the system complex and increases the number of parameters. Our proposed method does not contain any additional mechanisms and therefore has fewer system parameters than these hybrid methods. Essentially, the proposed method is more simple.

In general, the dynamics of PSO is designed such that particles converge to the weighted average of previous searched best positions because the convergence is what seems to provide the searching ability. In contrast, our proposed dynamics causes behavior that makes the particles hover around the weighted average of previous best positions with chaotic motion. This delivers a better searching ability than the standard PSO. Our proposed method has only three system parameters and no stochastic terms. This feature makes it easy to design an algorithm. Some implementations of PSO in electronic circuits have been studied [36, 37]. The proposed method would be suitable for hardware implementation because of the low number of system parameters and the lack of random values.

It can be said that the proposed method is close to the standard PSO and DPSO, because the proposed chaotic dynamics is based on rotation dynamics same as PSO and DPSO. The standard PSO has been improved by some approaches, such as updating some of the system parameters for each iteration and changing population topology [2, 3, 38–42]. These improving approaches can be applied to the proposed method. In addition, the approaches that improve the standard PSO by using chaos [21, 24–35] can also be applied to the proposed method. However, these approaches are beyond the scope of this paper. In this paper, we compare the basic performance of our proposed method with the standard PSO and DPSO. These comparisons clearly express the performance of the proposed dynamics. We also compare the proposed method with RD-PSO, because RD-PSO is also based on the rotation dynamics.

In Sec. 2 and 3, we explain the standard PSO and DPSO, respectively. Related works about algorithms combined PSO with chaos are introduced in Sec. 4. The proposed method is described in Sec. 5. In Sec. 6, we present the experimental results for four well-known benchmark functions and 27 benchmark functions of CEC’13 test suite [43]. The proposed method shows better searching ability than compared methods. Finally we conclude this paper in Sec. 7.
2. Standard PSO

This section describes the standard PSO algorithm that is the global best model, with a star neighborhood topology and an inertia weight [3]. The PSO algorithm searches for a solution using the population called particles. The position, velocity and best positions are described in Sec. 2.1. The dynamics is described in Sec. 2.2.

2.1 Particle information

Each particle has its own position, which is a candidate for the optimal solution of a given problem, and keeps its own previous best position showing the maximum or minimum fitness value in its searching history. To improve searching ability, the particles make a swarm in which they share information about the best position that exhibits the best fitness value in the set of their previous best positions.

In \(N\)-dimensional space, the position and velocity are denoted by \(x\) and \(v\), respectively.

\[
x_i(t) = \{x_{i1}(t), x_{i2}(t), \ldots, x_{iN}(t)\},
\]

\[
v_i(t) = \{v_{i1}(t), v_{i2}(t), \ldots, v_{iN}(t)\}.
\]

The \(i\)-th particle is evaluated at each time-step by an objective function \(f\). Let a position of the \(i\)-th particle be \(p_{bi}\), if \(f(x_i(t))\) is the best evaluated value in its searching history, and let \(gb\) if \(f(p_{bi})\) is the best value in all evaluated values, \(f(p_{b1}), f(p_{b2}), \ldots, f(p_{bn})\), where \(n\) is the number of particles. \(p_{bi}\) and \(gb\) are described as

\[
p_{bi} = \{p_{b1i}, p_{b2i}, \ldots, p_{bNi}\},
\]

\[
{gb} = \{gb_1, gb_2, \ldots, gb_N\}.
\]

2.2 Dynamics of PSO

The particles change their position following stochastic dynamics that includes their own best position and the best position of the swarm. In other words, the travel distance of the particle from a previous point to the next point is affected by the stochastic factors. The dynamics is

\[
v_{ij}(t+1) = \omega v_{ij}(t) + c_1 rand_{1ij} (p_{b_ij} - x_{ij}(t)) + c_2 rand_{2ij}(gb_j - x_{ij}(t)),
\]

\[
x_{ij}(t+1) = v_{ij}(t+1) + x_{ij}(t),
\]

where \(j\) indicates a specific dimension, \(\omega\), \(c_1\) and \(c_2\) are positive constants and \(rand_{1ij}\) and \(rand_{2ij}\) are stochastic terms given by uniformed distribution \([0,1]\). There exist many variations of PSO algorithms. In this paper, the use of Eqs. (5 and 6) is referred to as the standard PSO.

One of the distinctive features of PSO is that the particles converge to weighted average of the \(p_{bi}\) and \(gb\) with appropriate parameters. This behavior is designed by supposing that there is an optimal solution near the best position. Another feature is that the dynamics of the particles contain stochastic factors. These factors produce the complex behavior of the particles, which improves their ability to escape a local best position. These features make the PSO algorithm successfully applied in many application areas [4–8].

Even though PSO has been shown to perform well, one basic problem is that it is difficult to adequately understand how it works [9]. A fully comprehensive mathematical PSO model is still not available for several reasons [10], one of which, as reported in [10], is that the stochastic factors prevent the use of standard mathematical tools. Some researchers have proposed deterministic PSO models that simplify PSO by removing the stochastic factors in order to theoretically analyze the behavior of the particles [9,11–18].

3. Deterministic PSO

Deterministic version of PSO (DPSO) with an inertia weight is described in Sec. 3.1. The DPSO was reported in [13–18]. Sec. 3.2 introduces some researches about theoretically analysis.
3.1 Dynamics of DPSO

The DPSO algorithm searches for a solution using particles. The particles are updated by:

\[ v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 (pb_{ij} - x_{ij}(t)) + c_2 r_2 (gb_j - x_{ij}(t)), \quad (7) \]

\[ x_{ij}(t+1) = v_{ij}(t+1) + x_{ij}(t), \quad (8) \]

where \( v_i(t), x_i(t), pb_i, \) and \( gb \) are described in Sec. 2.1 and \( \omega, c_1, c_2, r_1 \) and \( r_2 \) are positive constants.

Following [21], Eqs. (7 and 8) can be mentioned as:

\[ \begin{bmatrix} v_{ij}(t+1) \\ y_{ij}(t+1) \end{bmatrix} = r \begin{bmatrix} \cos \Theta & \sin \Theta \\ -\sin \Theta & \cos \Theta \end{bmatrix} \begin{bmatrix} v_{ij}(t) \\ y_{ij}(t) \end{bmatrix}, \quad (9) \]

where

\[ y_{ij}(t) = x_{ij}(t) - p_{ij}, \quad (10) \]

\[ p_{ij} = \frac{c_1 r_1 pb_{ij} + c_2 r_2 gb_j}{c_1 r_1 + c_2 r_2}, \quad (11) \]

and \( \Theta \) and \( r \) are given by \( \omega, c_1, c_2, r_1 \) and \( r_2 \).

3.2 Theoretical analysis on DPSO

Ozcan and Mohand studied DPSO, excluding the inertia weight, and showed that the particle trajectory exhibits convergence, divergence and sinusoid wave depending on system parameters [11, 12]. Clerc and Kennedy also studied the DPSO excluding the inertia weight and derived which parameter settings would guarantee convergence of the particles [9]. In [13–17], DPSO including the inertia weight were analyzed. Bergh et al. showed particles are attracted towards weighted average of \( pb_{ij} \) and \( gb_j \) [13, 14]. Yasuda et al. found a qualitative relation between the classes of the behavior and parameter settings [15]. Trelea provided qualitative guidelines for parameter selection [16]. Zheng studied multitude possible cases of convergence [17]. These analysis of the DPSO models were based on the fact that \( pb_{ij} \) and \( gb_j \) are assumed to be non-changings. Recently, Clenghorn and Engelbrecht derived the proof of the particles convergence without \( pb_{ij} \) and \( gb_j \) are fixed [18].

These works demonstrate the importance of analyzing PSO dynamics in order to improve performance.

4. PSO algorithms combined with chaos

This section introduces some algorithms combined PSO and chaos. In [33, 44], these combined algorithms were classified into two types. The first type is that chaos is embedded into the velocity updating equation of PSO. Another type is that chaotic search is inserted into the procedures of PSO. These two kinds of algorithm are described in Sec. 4.1 and 4.2, respectively.

4.1 Chaos is embedded into the velocity updating equation

Some researchers embedded chaos into PSO instead of the system parameters and the stochastic elements. Jiang and Bompard updated an inertia weight for each iteration by Logistic map [24]. Gandomi et. al also updated system parameter of accelerated PSO by 12 different chaotic maps [25]. In [25, 26], Jiang et al. and Cheng-Hong Yang et al. replaced the stochastic terms as time-series data produced by Logistic map. Leandro dos Santos Coelho et al. also replaced the stochastic terms by time-series of Hénon map [28]. Alatas et al. substituted time-series of eight chaotic maps for the stochastic terms, an inertia weight and both of these elements [29].

While chaos is embedded into the parameters and the stochastic terms, the dynamics of their particles are the same as the standard PSO.

4.2 PSOs with chaotic search

There are some researches that insert chaotic search into the procedures of PSO. Bo Liu et al. and Jiejin et al. combined PSO and chaotic local search based on Logistic map [30, 31]. The chaotic local
search are implemented by Tent map [31, 32] and piecewise liner map [33]. Ying Wang et al. imported chaotic mutation by Logistic map into PSO [34]. Hefny and Azab extend the chaotic local search for multi swarm optimizer [35]. They separated population into two swarms namely PSO population and chaotic population. PSO population and chaotic population are updated by the standard PSO and Logistic map, respectively.

Even though the optimizers combined chaotic search and PSO improve searching performance, it makes the system complex and increases the number of parameters.

5. Optimizer using swarm of chaotic dynamical particles

This section describes our proposed method. Our proposed method searches for an optimal solution using a swarm of particles. The particle information is denoted in Sec. 5.1. The position and velocity of the particles are updated at each time-step, \( t \), by following the dynamics, which we will describe in Sec. 5.2. Each particle has its own center point for searching. The center point is also updated at each time-step by following the dynamics we will describe in Sec. 5.3. The particles search for an optimal solution hovering around the center point with chaotic motion, as we will discuss in Sec. 5.2 and 5.3. The basic particle’s behavior is described in Sec. 5.4. Section 5.5 shows algorithm of proposed method.

5.1 Particle information

The position, velocity and best position of \( i \)-th particle is mentioned Eqs. (1, 2 and 3). The best position of a swarm is mentioned Eq. (4). The center point for searching is denoted by \( N \)-dimensional vector \( f_{p_i}(t) \).

\[
f_{p_i}(t) = \{ f_{p_{i1}}(t), f_{p_{i2}}(t), \ldots, f_{p_{iN}}(t) \}.
\]  

5.2 Dynamics of chaotic particles

Particles search for an optimal solution by following chaotic dynamics. We focused on rotation matrix to design chaotic dynamic following same concept of [45]. The rotation matrix form of DPSO dynamics was shown in Sec. 3.1. Therefore, proposed method is essentially the same as DPSO. Here, the position of a particle that is transformed linearly from \( x_i(t) \) by \( f_{p_i}(t) \) is denoted by \( y_i(t) \).

\[
y_i(t) = x_i(t) - f_{p_i}(t) = \{ y_{i1}(t), y_{i2}(t), \ldots, y_{iN}(t) \}.
\]

The \( j \)-th elements of \( y_i(t) \) and \( v_i(t) \) are updated by

\[
\begin{bmatrix}
y_{ij}(t+1) \\
v_{ij}(t+1)
\end{bmatrix} = \begin{cases}
    \begin{bmatrix}
        2y_{th1ij} - y_{ij}(t) \\
        0
    \end{bmatrix} & \text{for } (y_{ij}(t), v_{ij}(t)) \in \Pi_1, \\
    \begin{bmatrix}
        2y_{th2ij} - y_{ij}(t) \\
        0
    \end{bmatrix} & \text{for } (y_{ij}(t), v_{ij}(t)) \in \Pi_2 \text{ and } (y_{ij}(t-1), v_{ij}(t-1)) \in \Pi_1, \\
    \begin{bmatrix}
        \cos \theta & \sin \theta \\
        -\sin \theta & \cos \theta
    \end{bmatrix} & \text{otherwise},
\end{cases}
\]

\[
\begin{align}
\Pi_1 &= \{ (y_{ij}(t), v_{ij}(t)) \mid y_{ij}(t) < y_{th1ij}, v_{ij}(t) \geq 0 \}, \\
\Pi_2 &= \{ (y_{ij}(t), v_{ij}(t)) \mid y_{ij}(t) > y_{th2ij}, v_{ij}(t) = 0 \}, \\
y_{th1ij} &= -\frac{1}{2} |gb_{ij} - pb_{ij}|, \\
y_{th2ij} &= y_{th1ij}R(k-1)\cos(k\theta) = \cos(\theta), \\
k &= \left\lfloor \frac{\pi}{\theta} \right\rfloor \in \{1, 2, \ldots\},
\end{align}
\]
where $R$ and $\theta$ refer to the expanding and the degree parameter, respectively, $k$ is the number of mapping times during a half around the origin by Eq. (14c) and $\lceil \cdot \rceil$ is a ceiling function such as $\lceil a \rceil = \min\{Z \mid Z \geq a\}$, where $a$ is a real number and $Z$ is an integer.

Here, we consider the stability of the particle trajectory on $y_{ij}(t)$-$v_{ij}(t)$ space. Jacobi matrix $J(y_{ij}(t),v_{ij}(t))$ of Eq. (14) are

$$J(y_{ij}(t),v_{ij}(t)) = \begin{cases} [-1 \ 0] \ & \text{for } \{(y_{ij}(t),v_{ij}(t)) \in \Pi_1\} \text{ or } \{(y_{ij}(t),v_{ij}(t)) \in \Pi_2 \text{ and } (y_{ij}(t-1),v_{ij}(t-1)) \in \Pi_1\}, \\ P^{-1} \begin{bmatrix} R e^{i\theta} & 0 \\ 0 & R e^{-i\theta} \end{bmatrix} P \ & \text{otherwise,} \end{cases} \tag{20a}$$

where $i$ is an imaginary unit and $P$ is a diagonal transformation matrix. Stability of the particles depends on the maximum absolute eigenvalues of these Jacobi matrixes. The maximum absolute eigenvalues of Eqs. (20a and 20b) are 1 and $R$, respectively. Therefore, the parameter range

$$R > 1 \tag{21}$$

guarantees the expansion and oscillation of the particles. If Eq. (21) is satisfied, the parameter range

$$0 < \theta < \frac{\pi}{2} \tag{22}$$

guarantees that the particles reach to $\Pi_1$. Hereafter, we consider these parameter ranges.

The typical behavior of a particle on $y_{ij}(t)$-$v_{ij}(t)$ space is shown in Figs. 1, 2, and 3. Here, $p(t) = [y_{ij}(t) \ v_{ij}(t)]$, and $y_{th1ij}$, which is a border of $\Pi_1$, is assumed to be a constant value for simplicity. In Fig. 1, $p(0)$ is given between $y_{th1ij}$ and $y_{th2ij}$ on $v_{ij}(t) = 0$. The trajectory started from $p(0)$ rotates divergently around the origin and then reaches $p(5)$ by Eq. (14c). When the trajectory enters $\Pi_1$ like $p(5)$, the state is mapped to $p(6)$ depending on Eq. (14a). Figure 2 shows $y_{th2ij}$, which is a border of $\Pi_2$. In order to give $y_{th2ij}$, we consider the maximum trajectory that reaches the maximum $y_{ij}(t)$ leaving from $y_{th1ij}$. In Fig. 2, let $p(1)$ be a point on $y_{th1ij}$ such that the particle started from $p(1)$ can be calculated by

$$p(1) = \begin{bmatrix} y_{th1ij} \\ -y_{th1ij} \tan \theta \end{bmatrix} \tag{23}$$

Let $p(0)$ be a virtual point that is a previous point of $p(1)$ and is calculated by
Fig. 3. Typical behavior and searching range of a particle on $y_{ij}(t)$-$v_{ij}(t)$ space whose initial position is on $y_{th1ij}$ and $v_{ij}(t) = 0$ with $R = 1.2$, $\theta = 37[\text{deg}]$, $y_{th1ij} = -3$, and $y_{th2ij} = 7.76$.

Fig. 4. A chaotic attractor on $y_{ij}(t)$-$v_{ij}(t)$ space with $R = 1.2$, $\theta = 37[\text{deg}]$, $y_{th1ij} = -3$, $y_{th2ij} = 7.76$, and 29000 to 30000 iterations.

$$p(0) = R^{-1} \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} p(1) = \begin{bmatrix} \frac{y_{th1ij}}{R \cos \theta} \\ 0 \end{bmatrix}. \quad (24)$$

As shown in Fig. 2, the trajectory started from $p(1)$ reaches $p(k)$ after it makes a half round around the origin, where $k$ is given by Eq. (19). $p(k)$ is

$$p(k) = R^k \begin{bmatrix} \cos(k\theta) & \sin(k\theta) \\ -\sin(k\theta) & \cos(k\theta) \end{bmatrix} p(0). \quad (25)$$

We define $y_{th2ij}$ by the $y_{ij}(t)$-coordinate of $p(k)$ as denoted by Eq. (18). Figure 3 describes the folding dynamics by Eqs. (14a and 14b). In Fig. 3, let $p(0)$ be a point on $y_{th2ij}$ and on $v_{ij}(t) = 0$. $p(0)$ is

$$p(0) = \begin{bmatrix} y_{th2ij} \\ 0 \end{bmatrix}, \quad (26)$$

where $y_{th2ij}$ is given by Eq. (18). As shown in Fig. 3, the particles which leaves from $p(0)$ reaches $p(k)$ after it makes a half round around the origin. Because $p(k)$ is in $\Pi_1$, the state is mapped to $p(k+1)$ by Eq. (14a). Since $p(k+1)$ is in $\Pi_2$, the state is mapped to $p(k+2)$ by Eq. (14b). The folded particle starts again rotating divergently around the origin from $p(k+2)$. These folding mechanisms must make particles hitting the range $[y_{th1ij}, y_{th2ij}]$ on $v_{ij}(t) = 0$. In other words, both $\Pi_1$ and $\Pi_2$ are needed to prevent divergence of the trajectory with any initial states. Because particles are stretching by Eq. (14c) and are folding by Eqs. (14a and 14b), particles exhibit chaotic attractor on $y_{ij}(t)$-$v_{ij}(t)$.

Particles search for an optimal solution using the $y_{ij}(t)$-coordinate of their trajectory. Here, as shown in Fig. 3, we consider the maximum searching range of $y_{ij}(t)$, which a particle leaving from $v_{ij}(t) = 0$ denotes. The trajectory started from $p(0)$ gives the range, shown in Fig. 3. The minimum and maximum $y_{ij}(t)$ of the range are denoted as $y_{ij\min}$ and $y_{ij\max}$, respectively. $y_{ij\min}$ is given by

$$y_{ij\min} = \min\{y_{ij}(k), y_{ij}(k-1)\}. \quad (27)$$

$y_{ij}(k)$ and $y_{ij}(k-1)$ can be calculated by Eqs. (25 and 26) as

$$y_{ij}(k) = y_{th2ij} R^k \cos(k\theta), \quad (28)$$
$$y_{ij}(k-1) = y_{th2ij} R^{(k-1)} \cos((k-1)\theta), \quad (29)$$

$y_{ij\max}$ is given by

$$y_{ij\max} = \max\{y_{th2ij}, 2y_{th1ij} - y_{ij\min}\}. \quad (30)$$
Fig. 5. A stable center point of the searching.

Fig. 6. Typical particle behavior where $f_{p_i}(t)$, $p_{b_i}$ and $g_p$ are assumed constant.

Following the dynamics described by Eq. (14), a particle exhibits a chaotic attractor, as shown in Fig. 4. This attractor exists on $(y_{ij_{\min}}, y_{ij_{\max}})$. Figure 4 is plotted with $R = 1.2$, $\theta = 37[^\circ]$ and 29000 to 30000 iterations without transient. The maximum Lyapunov exponent of the attractor was 0.1504, which was calculated with 30000 iterations by [46]. The 30000 iterations are considered as enough iteration to neglect transient. We also calculated the maximum Lyapunov exponents between $1 < R \leq 1.6$ with 0.01 increments and between $1 \leq \theta[^\circ] \leq 89$ with 1[^\circ] increments in a case of constant $y_{th_{1ij}}$. The calculated maximum Lyapunov exponents showed positive values.
When \( f_p \) Here, we look at the two-dimensional case of \( N = 5 \). Basic particle’s behavior

\[ 5.3 \] Dynamics of searching center points

Algorithm 1 Optimizer using chaotic dynamical particles

| STEP1: Initialization |
|-----------------------|
| particle positions \( x_i(0) \in R^N, i = 1, 2, ..., n \) |
| particle velocities \( v_i(0) = \{0, 0, ..., 0\}, i = 1, 2, ..., n \) |
| searching center points \( f_p i = x_i(0), i = 1, 2, ..., n \) |
| \( pb_i = x_i(0) \) |
| \( gb = pb_i \) |
| where \( i_y = \arg \min f(pb_i), i = 1, 2, ..., n \) |

STEP2: Updating searching center point

\( f_p i(t + 1) = (1 - 2c)f_p i(t) + c(pb_i + gb) \)

STEP3: Updating particle positions and velocities

\[ y_i(t) = x_i(t) - f_p i(t + 1) \]
\[ y_i(t + 1) \text{ and } v_i(t + 1) \text{ are given by Eq. (14)} \]
\[ x_i(t + 1) = y_i(t + 1) + f_p i(t + 1) \]

STEP4: Evaluating particles positions

\[ \text{for } i = 1 \text{ to } i = n \text{ do} \]
\[ \text{if } f(p_{bi}) < f(x_i(t + 1)) \text{ then} \]
\[ pb_i = x_i(t + 1) \]
\[ \text{if } f(gb) < f(x_i(t + 1)) \text{ then} \]
\[ gb = x_i(t + 1) \]
\[ \text{end if} \]
\[ \text{end if} \]
\[ \text{end for} \]

STEP5: Checking termination criterion

\[ \text{if } t = t_{\text{max}} \text{ then} \]
\[ \text{terminate} \]
\[ \text{else} \]
\[ t = t + 1 \]
\[ \text{return to STEP2} \]
\[ \text{end if} \]

5.3 Dynamics of searching center points

An \( i \)-th particle searches for an optimal solution by exploring around \( f_p i(t) \), which is a center point of their searching. At \( t = 0 \), all elements of \( f_p i(0) \) are initially set to \( x_i(0) \). For \( t \geq 0 \), \( f_p i(t) \) is updated by

\[
 f_p i(t + 1) = g(f_p i(t)) \\
 = f_p i(t) + c \{pb_i - f_p i(t) + gb - f_p i(t)\} \\
= (1 - 2c)f_p i(t) + c(pb_i + gb),
\]

where \( c \) is a positive constant. In order to be guaranteed stable \( f_p i(t) \), \( c \) should be set to \((0, 1)\).

Figure 5 shows the typical behavior of \( f_{p_{ij}}(t) \), which is the \( j \)-th element of \( f_p i(t) \) by following Eq. (31). For simplicity, we assume that \( pb_{ij} \) and \( gb \) are constant. In Fig. 5, an intersection of \( f_{p_{ij}}(t + 1) = f_{p_{ij}}(t) \) and \( f_{p_{ij}}(t + 1) = (1 - 2c)f_{p_{ij}}(t) + c(pb_{ij} + gb) \) is denoted by \( p \). \( p \) is equal to \( \frac{1}{2}(pb_{ij} + gb) \). Assuming that \( c \) is set to \((0, 1)\), \( f_{p_{ij}}(t) \) starting from any initial point \( f_{p_{ij}}(0) \) must converge to \( p \), as shown in Fig. 5, that is, \( p \) is a stable fixed point. In the standard PSO and DPSO, the particles converge to \( \frac{1}{2}(pb_{ij} + gb) \) if \( c_1 = c_2 \). Therefore, the converged point of the center point is equal to these methods in the situations.

5.4 Basic particle’s behavior

Here, we look at the two-dimensional case of \( N = 2 \). First, we consider single particle trajectory. When \( f_p i(t) \), \( pb_i \) and \( gb \) are constants the particles must exhibit chaos on \( x \) space. Figure 6 shows
### Table I. Benchmark functions.

| Type        | $f$ Function                                                                 | Optimal fitness | Search range                  |
|-------------|------------------------------------------------------------------------------|-----------------|-------------------------------|
| **Basic**   | $f_1$ Sphere, $f_1(x) = \sum_{i=1}^{N} x_i^2$                                | 0               | $[-20, 20]^N$                |
|             | $f_2$ Rastrigin, $f_2(x) = \sum_{i=1}^{N} (x_i^2 - 10 \cos(2\pi x_i) + 10)$ | -1400           | $[-5.12, 5.12]^N$            |
|             | $f_3$ Rosenbrock, $f_3(x) = \sum_{i=1}^{N-1} 100 \left( (x_{i+1}^2 - x_i^2)^2 + (1 - x_i)^2 \right)$ | -1100           | $[-10, 10]^N$                |
|             | $f_4$ Griewank, $f_4(x) = \frac{1}{4000} \sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1$ | -1000           | $[-600, 600]^N$              |
| **Multimodal** | $f_5$ Sphere Function                                                           | -1300           |                               |
|             | $f_6$ Rotated High Conditioned Elliptic Function                              | -1200           |                               |
|             | $f_7$ Rotated Bent Cigar Function                                              | -1100           |                               |
|             | $f_8$ Rotated Discus Function                                                 | -1000           |                               |
|             | $f_9$ Different Powers Function                                                |                 |                               |
| **CEC 2013 test suite [43]** | $f_{10}$ Rotated Rosenbrock’s Function                                     | -900            |                               |
|             | $f_{11}$ Rotated Schaffers F7 Function                                         | -800            |                               |
|             | $f_{12}$ Rotated Ackley’s Function                                             | -700            |                               |
|             | $f_{13}$ Rotated Weierstrass Function                                          | -600            |                               |
|             | $f_{14}$ Rotated Griewank’s Function                                           | -500            |                               |
|             | $f_{15}$ Rastrigin’s Function                                                  | -400            |                               |
|             | $f_{16}$ Rotated Rastrigin’s Function                                          | -300            |                               |
|             | $f_{17}$ Non-Continuous Rotated Rastrigin’s Function                           | -200            |                               |
|             | $f_{18}$ Schwefel’s Function                                                   | -100            |                               |
|             | $f_{19}$ Rotated Schwefel’s Function                                           | 100             |                               |
|             | $f_{20}$ Lunacek Bi-Rastrigin Function                                         | 300             |                               |
|             | $f_{21}$ Rotated Lunacek Bi-Rastrigin Function                                 | 400             |                               |
|             | $f_{22}$ Expanded Griewank’s plus Rosenbrock’s Function                       | 500             |                               |
|             | $f_{23}$ Expanded Schaffer’s F6 Function                                       | 600             |                               |
| **Composition** | $f_{24}$ Composition Function 1 (n = 5, Rotated)                            | 700             |                               |
|             | $f_{25}$ Composition Function 2 (n = 3, Unrotated)                            | 800             |                               |
|             | $f_{26}$ Composition Function 3 (n = 3, Rotated)                              | 900             |                               |
|             | $f_{27}$ Composition Function 4 (n = 3, Rotated)                              | 1000            |                               |
|             | $f_{28}$ Composition Function 5 (n = 3, Rotated)                              | 1100            |                               |
|             | $f_{29}$ Composition Function 6 (n = 5, Rotated)                              | 1200            |                               |
|             | $f_{30}$ Composition Function 7 (n = 5, Rotated)                              | 1300            |                               |
|             | $f_{31}$ Composition Function 8 (n = 5, Rotated)                              | 1400            |                               |

### Table II. Experimental conditions.

| Conditions | Experiments parameters | comparison for 4 methods |
|------------|------------------------|--------------------------|
| Number of particles, $n$ | 30 |             |
| Number of dimensions, $N$ | 30 |             |
| $x_i(0)$ | randomly generated by the uniform distribution in the range of Table I | 100 sets of initial positions* |
| $v_i(0)$ | $\{0, 0, ..., 0\}$ |             |
| Maximum iteration $t_{max}$ | 1000 | 1000, 2000, 3000 |
| Number of independent runs | 12 | 100 |

* The positions were randomly generated following the uniform distribution in the range of Table I.
* All method used same sets for the comparison.

A typical trajectory. As shown in Fig. 6, the particle trajectory does not diverge and exhibits chaos under the conditions.

Typical behavior of particles is shown in Fig. 7. We depict only three particles. The best positions of their searching history are represented by $pb_1$, $pb_2$, and $pb_3$, respectively, and their center points of searching are also given by $fp_1$, $fp_2$, and $fp_3$, respectively. The best position of the swarm’s searching history is denoted by $gb$. Each particle has its own searching area represented by a gray rectangle where the particle prows around the center point with chaotic motion. The searching area of the first particle of $x_{11}(t)$-axis is denoted by a range $[A, B]$. By Eq. (13), $A$ and $B$ are given as
Table III. Parameters search ranges.

| Method     | Parameters     | Range         |
|------------|----------------|---------------|
| PSO        | $\omega$       | $c_1 = c_2$   | (0, 2)       |
|            | $c_1 = c_2$    | (0, 3)        |
| DPSO       | $\omega$       | $c_1 = c_2$   | (0, 2)       |
|            | (0, 3)         |               |
| RDPSO      | $\Delta$       | (0, 1)        |
|            | $\theta_{[\text{deg}]}$ | $(0, 1)$       |
|            | $v_{th}$       | $(10 \times 10^{-6}, 10 \times 10^{-4})$ |
|            | $\alpha$       | (0, 3)        |
| OSDCP (proposed) | $R$                | (1, 3)        |
|            | $\theta_{[\text{deg}]}$ | $(0, 90)$  |
|            | $c$             | (0, 1)        |

Table IV. Adjusted system parameters.

| $f$ | PSO $\omega$ | DPSO $\omega$ | $\Delta$ | RL-FPSO $\theta$ | $\psi$ | OSDCP (proposed) $\theta$ |
|-----|--------------|---------------|----------|------------------|-------|--------------------------|
| 0.516 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.989 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.464 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.340 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.295 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.910 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.506 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.570 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.812 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.688 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.709 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.807 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.530 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.725 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.666 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.682 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.851 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.450 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.882 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.960 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.917 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.811 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |
| 0.978 | 0.82 | 0.994 | 0.180 | 0.843 | 95.2 | 2.65e-05 | 1.31 |

\[ A = y_{11\text{min}} + f_{p11}, \]  
\[ B = y_{11\text{max}} + f_{p11}, \]  

where $y_{11\text{min}}$ and $y_{11\text{max}}$ are given by Eqs. (27 and 30), respectively. Equations (32 and 33) show that the searching ranges of particles depend on the $y_{ithij}$ described in Sec. 5.2. $y_{ithij}$ is a dependent variable determined by $p_{bi}$ and $gb_i$, as described in Eq. (17). Therefore, the proposed method can search for an optimal solution with only three system parameters. As shown in Fig. 7, the searching area of the second particle is smaller than that of the first because $p_{b2}$ is closer to $gb$ than $p_{b1}$. In other words, a particle that has a $p_{b1}$ far from $gb$ searches for the solution roughly (like the first particle) and a particle that has a near $p_{b1}$ searches with high density (like the second particle).

5.5 Algorithm of proposed method

Proposed method is summarized in Algorithm 1.
Table V. Experimental results for object functions with $t_{\text{max}} = 1000$. Each mean evaluated value is calculated with 100 independent trials. The values with * denote the best value for each function.

| $f$ | PSO | DPSO | RD-PSO | OSCDP (proposed) |
|-----|-----|------|--------|-------------------|
| $f_1$ | 3.309e-04 | 17.81 | 2.851 | * 1.402e-09 |
|      | 2.651e-03 | 11.20 | 10.70 | * 7.517e-09 |
| $f_2$ | 52.34 | 80.40 | 2.851 | * 31.24 |
|      | 13.43 | 22.03 | 17.29 | * 8.775 |
| $f_3$ | 44.36 | 512.3 | 535.0 | * 28.44 |
|      | 35.01 | 340.2 | 1556 | * 12.79 |
| $f_4$ | * 1.431e-02 | 4.521 | 1.665 | * 1.681e-02 |
|      | * 1.451e-02 | 2.352 | 2.363 | 2.100e-02 |
| $f_5$ | * -1400 | 9037 | -1172 | -1400 |
|      | * 2.665e-13 | 4910 | 803.9 | 2.613e-09 |
| $f_6$ | * 3.742e+06 | 3.189e+07 | 1.918e+07 | 1.249e+07 | 3.825e+06 |
|      | * 2.308e+06 | * 1.734e+10 | 7.086e+09 | 8.875e+09 | 9.659e+06 |
| $f_7$ | * 5.134e+09 | 6.654e+04 | * 4.823e+04 | 5.595e+04 |
|      | * 1.266e+04 | 1.438e+04 | 1.438e+04 | 1.656e+04 |
| $f_8$ | * -999.8 | -219.1 | -861.4 | * -1000.0 |
|      | * -1400 | 9037 | -1172 | -1400 |
| $f_9$ | * 3.742e+06 | 3.189e+07 | 1.918e+07 | 1.249e+07 | 3.825e+06 |
|      | * 2.308e+06 | * 1.734e+10 | 7.086e+09 | 8.875e+09 | 9.659e+06 |
| $f_{10}$ | * -999.8 | -219.1 | -861.4 | * -1000.0 |
|      | * -1400 | 9037 | -1172 | -1400 |
| $f_{11}$ | * -648.1 | 2.086e+04 | -594.3 | * -691.1 |
|      | * -737.4 | -137.8 | 377.9 | * 4.619e-06 |
| $f_{12}$ | * -679.0 | 2.086e+04 | -594.3 | * -691.1 |
|      | * -737.4 | -137.8 | 377.9 | * 4.619e-06 |

123
Table VI. Experimental results for object functions with $t_{max} = 2000$. Each mean evaluated value is calculated with 100 independent trials. The values with * denote the best value for each function.

| $f$ | PSO | DPSO | RD-PSO | OSCDP (proposed) |
|-----|-----|------|--------|------------------|
| $f_1$ Mean | 2.952e-04 | 16.63 | 2.834 | * 4.138e-21 |
| Std. Dev. | 2.925e-03 | 11.24 | 10.69 | * 3.500e-20 |
| $f_2$ Mean | 54.25 | 19.00 | 51.07 | * 2.854 |
| Std. Dev. | 14.51 | 22.06 | 15.91 | * 7.459 |
| $f_3$ Mean | 27.85 | 497.2 | 517.9 | 23.75 |
| Std. Dev. | 25.84 | 337.3 | 1555 | * 10.78 |
| $f_4$ Mean | * 1.470e-02 | 4.405 | 1.487 | * 1.668e+02 |
| Std. Dev. | 2.929e-02 | 2.289 | 2.362 | * 2.080e-02 |
| $f_5$ Mean | -1400 | 7229 | -1188 | * -1400 |
| Std. Dev. | 1.036e-03 | 4803 | 801.0 | * 1.771e-13 |
| $f_6$ Mean | * 3.246e+06 | 3.163e+07 | 2.834 | * 3.935e+06 |
| Std. Dev. | 2.510e+06 | 1.919e+07 | 9.659e+06 | * 2.245e+06 |
| $f_7$ Mean | * 5.864e+10 | 1.551e+10 | 1.541e+10 | 3.740e+09 |
| Std. Dev. | 6.449e+10 | 2.360e+10 | 5.618e+09 | * 4.389e+04 |
| $f_8$ Mean | 2.711e+09 | 5.864e+10 | 1.551e+10 | 4.173e+04 |
| Std. Dev. | * 4.517e+09 | 6.449e+10 | 2.360e+10 | * 4.389e+02 |
| $f_9$ Mean | * 2.785e+04 | 6.629e+04 | * 2.593e+04 | 4.389e+02 |
| Std. Dev. | 1.225e+04 | 1.444e+04 | 1.162e+04 | * 1.050e-01 |
| $f_{10}$ Mean | -999.9 | -284.3 | -867.8 | * -1000.0 |
| Std. Dev. | 6.036e-01 | 409.1 | 379.6 | * 4.304e-13 |
| $f_{11}$ Mean | * -868.7 | -249.6 | -744.4 | * -866.2 |
| Std. Dev. | 18.99 | 541.5 | 218.2 | * 26.46 |
| $f_{12}$ Mean | -648.8 | 2.084e+04 | -608.4 | * -692.9 |
| Std. Dev. | 50.30 | 1.066e+05 | 6.14 | * 53.70 |
| $f_{13}$ Mean | * -349.4 | 27.03 | -295.9 | * -354.8 |
| Std. Dev. | 14.00 | 88.74 | 40.73 | * 15.46 |
| $f_{14}$ Mean | -147.0 | 136.6 | 151.0 | * -185.0 |
| Std. Dev. | 51.94 | 103.2 | 98.05 | * 35.42 |
| $f_{15}$ Mean | 583.3 | 895.1 | 738.9 | * 579.3 |
| Std. Dev. | 47.14 | 105.0 | 66.13 | * 26.46 |
| $f_{16}$ Mean | 506.8 | 2979 | 618.4 | * 505.5 |
| Std. Dev. | 2.224 | 3157 | 489.0 | * 1.680 |
| $f_{17}$ Mean | 613.3 | 614.9 | 614.9 | * 613.2 |
| Std. Dev. | 7.764e+01 | 3.000e-01 | 3.316e-01 | * 1.003 |
| $f_{18}$ Mean | 1018 | 2160 | 1122 | * 973.1 |
| Std. Dev. | 83.07 | 316.1 | 210.6 | * 97.76 |
| $f_{19}$ Mean | 4457 | 3615 | 4951 | * 4314 |
| Std. Dev. | 740.8 | 918.0 | 928.8 | * 752.5 |
| $f_{20}$ Mean | 401.5 | 795.9 | 468.4 | * 429.7 |
| Std. Dev. | 22.78 | 107.6 | 64.49 | * 32.33 |
| $f_{21}$ Mean | 583.3 | 895.1 | 738.9 | * 579.3 |
| Std. Dev. | 47.14 | 105.0 | 66.13 | * 54.90 |
| $f_{22}$ Mean | 506.8 | 2979 | 618.4 | * 505.5 |
| Std. Dev. | 2.224 | 3157 | 489.0 | * 1.680 |
| $f_{23}$ Mean | 613.3 | 614.9 | 614.9 | * 613.2 |
| Std. Dev. | 7.764e+01 | 3.000e-01 | 3.316e-01 | * 1.003 |
| $f_{24}$ Mean | 1018 | 2160 | 1122 | * 973.1 |
| Std. Dev. | 83.07 | 316.1 | 210.6 | * 97.76 |
| $f_{25}$ Mean | 4457 | 3615 | 4951 | * 4314 |
| Std. Dev. | 740.8 | 918.0 | 928.8 | * 752.5 |
| $f_{26}$ Mean | 506.8 | 2979 | 618.4 | * 505.5 |
| Std. Dev. | 2.224 | 3157 | 489.0 | * 1.680 |
| $f_{27}$ Mean | 613.3 | 614.9 | 614.9 | * 613.2 |
| Std. Dev. | 7.764e+01 | 3.000e-01 | 3.316e-01 | * 1.003 |

124
Table VII. Experimental results for object functions with $t_{\text{max}} = 3000$. Each mean evaluated value is calculated with 100 independent trials. The values with * denote the best value for each function.

| $f$  | PSO         | DPSO         | RD-PSO       | OSCDP (proposed) |
|------|-------------|--------------|--------------|------------------|
| $f_1$  | Mean: 4.172e-07 | 16.63 | 2.832 | * 6.372e-32 |
|       | Std. Dev.: 2.857e-06 | 11.24 | 10.69 | * 4.540e-31 |
| $f_2$  | Mean: 51.72 | 18.99 | 50.23 | 28.30 |
|       | Std. Dev.: 12.82 | 22.06 | 15.67 | * 7.453 |
| $f_3$  | Mean: 20.80 | 497.2 | 512.0 | * 20.06 |
|       | Std. Dev.: 23.53 | 337.3 | 1555 | * 10.77 |
| $f_4$  | Mean: * 1.425e-02 | 4.403 | 1.451 | 1.668e-02 |
|       | Std. Dev.: * 1.602e-02 | 2.288 | 2.360 | 2.080e-02 |
| $f_5$  | Mean: -1400 | 6899 | 6809 | * 1.371e-13 |
|       | Std. Dev.: 1.211e-08 | 4766 | 4760 | * 1.371e-13 |
| $f_6$  | Mean: 2.588e+06 | 3.163e+07 | 6.372e-03 | 2.872e+06 |
|       | Std. Dev.: 1.991e-06 | 1.919e+07 | 8.879e+06 | * 1.630e+06 |
| $f_7$  | Mean: 50.23 | 8.234 | 2.346 | 2.098e+02 |
|       | Std. Dev.: 1.058e+04 | 1.444e+04 | * 1.462e+04 | 3.948e+04 |
| $f_8$  | Mean: -1400 | 6899 | 6809 | * 1.371e-13 |
|       | Std. Dev.: 1.211e-08 | 4766 | 4760 | * 1.371e-13 |
| $f_9$  | Mean: * 2.588e+06 | 3.163e+07 | 6.372e-03 | 2.872e+06 |
|       | Std. Dev.: 1.991e-06 | 1.919e+07 | 8.879e+06 | * 1.630e+06 |
| $f_{10}$ | Mean: 50.23 | 8.234 | 2.346 | 2.098e+02 |
|       | Std. Dev.: 1.058e+04 | 1.444e+04 | * 1.462e+04 | 3.948e+04 |

6. Experimental results

We compared the performance of searching for an optimal solution by the proposed method with three other methods, the standard PSO, DPSO and RD-PSO. We used the 30 dimensional objective...
Table VIII. t-test between OSCDP and PSO with $t_{\text{max}} = 3000$.

| $f$ | test statistic, $t$ | probability of $t$ | significantly better method |
|-----|---------------------|---------------------|----------------------------|
| $f_1$ | -1.460             | 7.37e-02            |                          |
| $f_2$ | -15.799            | 1.02e-34            | OSCDP (proposed)         |
| $f_3$ | -0.286             | 0.388               |                          |
| $f_4$ | 0.925              | 0.822               |                          |
| $f_5$ | -1.000             | 0.160               |                          |
| $f_6$ | 1.102              | 0.864               |                          |
| $f_7$ | 0.760              | 0.776               |                          |
| $f_8$ | 4.570              | 1.000               |                          |
| $f_9$ | -1.000             | 0.160               |                          |
| $f_{10}$ | 0.028           | 0.511               |                          |
| $f_{11}$ | -7.250          | 7.87e-12            | OSCDP (proposed)         |
| $f_{12}$ | 3.172             | 0.999               |                          |
| $f_{13}$ | -4.893           | 1.10e-06            | OSCDP (proposed)         |
| $f_{14}$ | -1.231            | 0.111               |                          |
| $f_{15}$ | -2.998            | 1.54e-03            | OSCDP (proposed)         |
| $f_{16}$ | -6.433            | 6.42e-10            | OSCDP (proposed)         |
| $f_{17}$ | -3.049            | 1.31e-03            | OSCDP (proposed)         |
| $f_{18}$ | 6.641             | 1.000               |                          |
| $f_{19}$ | -1.578            | 5.81e-02            |                          |
| $f_{20}$ | 6.005             | 1.000               | PSO                       |
| $f_{21}$ | -0.313            | 0.377               |                          |
| $f_{22}$ | -2.817            | 2.68e-03            | OSCDP (proposed)         |
| $f_{23}$ | -0.635            | 0.263               |                          |
| $f_{24}$ | -2.912            | 2.00e-03            | OSCDP (proposed)         |
| $f_{25}$ | 9.921             | 1.000               |                          |
| $f_{26}$ | -2.959            | 1.76e-03            | OSCDP (proposed)         |
| $f_{27}$ | -0.298            | 0.383               |                          |
| $f_{28}$ | -2.475            | 7.12e-03            | OSCDP (proposed)         |
| $f_{29}$ | -12.936           | 4.23e-28            | OSCDP (proposed)         |
| $f_{30}$ | -1.000            | 0.160               |                          |
| $f_{31}$ | 0.028             | 0.511               |                          |

Table IX. t-test between OSCDP and DPSO with $t_{\text{max}} = 3000$.

| $f$ | test statistic, $t$ | probability of $t$ | significantly better method |
|-----|---------------------|---------------------|----------------------------|
| $f_{1}$ | -14.795           | 3.99e-27            | OSCDP (proposed)           |
| $f_{2}$ | -21.775           | 4.97e-44            | OSCDP (proposed)           |
| $f_{3}$ | -14.139           | 8.19e-26            | OSCDP (proposed)           |
| $f_{4}$ | -19.166           | 2.16e-35            | OSCDP (proposed)           |
| $f_{5}$ | -17.411           | 3.38e-32            | OSCDP (proposed)           |
| $f_{6}$ | -14.933           | 1.49e-27            | OSCDP (proposed)           |
| $f_{7}$ | -8.645            | 4.57e-14            | OSCDP (proposed)           |
| $f_{8}$ | -13.221           | 2.58e-29            | OSCDP (proposed)           |
| $f_{9}$ | -17.509           | 2.22e-32            | OSCDP (proposed)           |
| $f_{10}$ | -11.388           | 5.11e-20            | OSCDP (proposed)           |
| $f_{11}$ | -2.020            | 2.30e-02            | OSCDP (proposed)           |
| $f_{12}$ | -3.068            | 1.25e-03            | OSCDP (proposed)           |
| $f_{13}$ | -18.699           | 1.32e-44            | OSCDP (proposed)           |
| $f_{14}$ | -20.668           | 5.43e-38            | OSCDP (proposed)           |
| $f_{15}$ | -42.442           | 3.37e-68            | OSCDP (proposed)           |
| $f_{16}$ | -29.484           | 1.20e-57            | OSCDP (proposed)           |
| $f_{17}$ | -26.942           | 6.96e-02            | OSCDP (proposed)           |
| $f_{18}$ | -34.878           | 1.53e-42            | OSCDP (proposed)           |
| $f_{19}$ | -11.547           | 6.09e-24            | OSCDP (proposed)           |
| $f_{20}$ | -32.664           | 7.33e-61            | OSCDP (proposed)           |
| $f_{21}$ | -26.738           | 4.49e-59            | OSCDP (proposed)           |
| $f_{22}$ | -7.827            | 2.81e-12            | OSCDP (proposed)           |
| $f_{23}$ | -17.493           | 9.02e-35            | OSCDP (proposed)           |
| $f_{24}$ | -35.871           | 1.57e-65            | OSCDP (proposed)           |
| $f_{25}$ | -45.500           | 2.14e-92            | OSCDP (proposed)           |
| $f_{26}$ | -17.659           | 1.66e-41            | OSCDP (proposed)           |
| $f_{27}$ | -24.008           | 9.98e-59            | OSCDP (proposed)           |
| $f_{28}$ | -18.215           | 7.79e-43            | OSCDP (proposed)           |
| $f_{29}$ | -15.621           | 2.36e-36            | OSCDP (proposed)           |
| $f_{30}$ | -16.993           | 2.89e-40            | OSCDP (proposed)           |
| $f_{31}$ | -42.301           | 1.91e-88            | OSCDP (proposed)           |

functions listed in Table I as the benchmark problems. Experimental conditions are described in Table II. Adjusted system parameters for each function are denoted in Sec. 6.1. Numerical results are described in Sec. 6.2.
Table X. t-test between OSCDP and RD-PSO with $t_{\text{max}} = 3000$.

| $f$ | test statistic, $t$ | probability of $t$ | significantly better method |
|-----|---------------------|--------------------|----------------------------|
| $f_1$ | -2.648 | 4.71e-03 | OSCDP (proposed) |
| $f_2$ | -12.636 | 2.79e-05 | OSCDP (proposed) |
| $f_3$ | -3.163 | 1.04e-03 | OSCDP (proposed) |
| $f_4$ | -6.077 | 1.14e-08 | OSCDP (proposed) |
| $f_5$ | -2.625 | 5.02e-03 | OSCDP (proposed) |
| $f_6$ | -10.820 | 4.11e-19 | OSCDP (proposed) |
| $f_7$ | -5.183 | 5.23e-07 | OSCDP (proposed) |
| $f_8$ | 15.327 | 1.000 | RD-PSO |
| $f_9$ | -3.471 | 3.85e-04 | OSCDP (proposed) |
| $f_{10}$ | -5.663 | 7.07e-08 | OSCDP (proposed) |
| $f_{11}$ | -12.530 | 1.27e-24 | OSCDP (proposed) |
| $f_{12}$ | 0.226 | 0.589 | |
| $f_{13}$ | 15.327 | 1.000 | RD-PSO |
| $f_{14}$ | -3.471 | 3.85e-04 | OSCDP (proposed) |
| $f_{15}$ | -5.663 | 7.07e-08 | OSCDP (proposed) |
| $f_{16}$ | -12.530 | 1.27e-24 | OSCDP (proposed) |
| $f_{17}$ | 0.226 | 0.589 | |
| $f_{18}$ | -15.043 | 2.84e-34 | OSCDP (proposed) |
| $f_{19}$ | -3.163 | 1.04e-03 | OSCDP (proposed) |
| $f_{20}$ | -6.077 | 1.14e-08 | OSCDP (proposed) |
| $f_{21}$ | -2.625 | 5.02e-03 | OSCDP (proposed) |
| $f_{22}$ | -10.820 | 4.11e-19 | OSCDP (proposed) |
| $f_{23}$ | -5.183 | 5.23e-07 | OSCDP (proposed) |
| $f_{24}$ | 15.327 | 1.000 | RD-PSO |
| $f_{25}$ | -3.471 | 3.85e-04 | OSCDP (proposed) |
| $f_{26}$ | -5.663 | 7.07e-08 | OSCDP (proposed) |
| $f_{27}$ | -12.530 | 1.27e-24 | OSCDP (proposed) |
| $f_{28}$ | 0.226 | 0.589 | |
| $f_{29}$ | -15.043 | 2.84e-34 | OSCDP (proposed) |
| $f_{30}$ | -3.163 | 1.04e-03 | OSCDP (proposed) |
| $f_{31}$ | -6.077 | 1.14e-08 | OSCDP (proposed) |

6.1 Parameters adjusting
It is important for fair comparison to adjust system parameters of each optimizer. In this paper, the parameters were adjusted by PSO algorithm described in Sec. 2. It took almost same computational cost for each method to decide the best set of parameters. For the adjusting, each parameter were denoted a vector $X$. Depending on the optimizers, $X$ is

$$
X_i(t) = \begin{cases} 
\{\omega, c\} & \text{for PSO and DPSO}, \\
\{\Delta, R, v_{th}, \alpha\} & \text{for RD-PSO}, \\
\{R, \theta, c\} & \text{for OSCDP}
\end{cases}
$$

where $i$ is an index number of particles. 20 particles were used for adjusting the parameters. Initial positions of the particles were randomly generated following the uniform distribution in the ranges in Table III. The velocity of $i$-th particle was set to $V_i(0) = \{0, 0, \ldots, 0\}$. $X_i(t)$ is evaluated by the benchmark results that were given by the optimizer constructed with $X_i(t)$. The results were given by a condition described in Table II. The final fitness values of each independent run were averaged and $X_i(t)$ was evaluated by the mean value. As following Sec. 2, $X_i(t)$ and $V_i(t)$ were updated with $\omega = 0.729, c_1 = c_2 = 1.49445$ [47]. The parameter adjusting was executed 5 times. Table IV shows the best adjusted parameter set.

6.2 Results
Tables V, VI and VII shows the mean evaluated values and standard deviations of the searched best positions at the end of iteration by 100 independent trials.

We compared the numerical results between proposed method and the others by Welch’s t-test [48] with significance level 0.05. Table VIII shows t-test results between PSO and proposed method with $t_{\text{max}} = 3000$. While PSO performed better than proposed method for five functions, proposed method denoted better performance for 13 functions. Difference between PSO and proposed could not observed with the significance level for 13 functions. Tables IX and X shows t-test results between deterministic PSOs and proposed method with $t_{\text{max}} = 3000$. Our proposed method performed better
than these deterministic PSOs for almost all functions.

7. Conclusion
In this paper, we proposed a novel optimization procedure called the “optimizer using swarm of chaotic dynamical particles” procedure. The proposed method does not contain any stochastic factors and has only three system parameters. The performance was compared with the standard PSO and deterministic PSOs, and results showed that the proposed method exhibited better performance compared with deterministic PSOs. The proposed method also had a better performance than the standard PSO for 13 benchmark functions.

In our future work, we will analyze the behavior of the particles and examine its chaotic property. We will also study the relationship between the system parameters of the proposed method and its performance, and the dependence of the parameters on the problems for the future work. The improving methods for the standard PSO, such as updating some of the system parameters and changing population topology, can be applied to the proposed method. It is also the future work to apply these methods to the proposed method and to compare the modified proposed method with other improved PSOs.

Acknowledgments
This work was supported by JSPS KAKENHI Grant Number 26630172.

References
[1] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proc. IEEE Int. Conf. neural networks, vol. 4, pp. 1942–1948, 1995.
[2] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” Proc. Int. Symp. Micro Mach. Hum. Sci., pp. 39–43, October 1995.
[3] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” Proc. 1998 IEEE Int. Conf. Evol. Comput., pp. 69–73, May 1998.
[4] R. Eberhart and X. Hu, “Human tremor analysis using particle swarm optimization,” Proc. 1999 Congr. Evol. Comput., pp. 1927–1930, July 1999.
[5] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage security assessment,” IEEE Trans. Power Syst., vol. 15, no. 4, pp. 1232–1239, 2000.
[6] M. Wachowiak, R. Smolikova, Y. Zheng, J. Zurada, and A. Elmaghraby, “An approach to multimodal biomedical image registration utilizing particle swarm optimization,” IEEE Trans. Evol. Comput., vol. 8, pp. 289–301, June 2004.
[7] M. AlRashidi and M. El-Hawary, “A survey of particle swarm optimization applications in electric power systems,” IEEE Trans. Evol. Comput., vol. 13, pp. 913–918, August 2009.
[8] M. Miyatake, M. Veerachary, F. Toriumi, N. Fujii, and H. Ko, “Maximum power point tracking of multiple photovoltaic arrays: A PSO approach,” IEEE Trans. Aerosp. Electron. Syst., vol. 47, pp. 367–380, January 2011.
[9] M. Clerc and J. Kennedy, “The particle swarm - explosion, stability, and convergence in a multidimensional complex space,” IEEE Trans. Evol. Comput., vol. 6, no. 1, pp. 58–73, 2002.
[10] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” Swarm Intell., vol. 1, pp. 33–57, August 2007.
[11] E. Ozcan and C. Mohan, “Analysis of a simple particle swarm optimization system,” Proc. 1998 Artif. Neural Networks Eng. Conf., vol. 8, pp. 253–258, 1998.
[12] E. Ozcan and C. Mohan, “Particle swarm optimization: surfing the waves,” Proc. 1999 Congr. Evol. Comput., pp. 1939–1944, 1999.
[13] F.V.D. Bergh, An analysis of particle swarm optimizers, PhD thesis, University of Pretoria etd, 2002.
[14] F.V.D. Bergh and A.P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” Inf. Sci. (Ny), vol. 176, pp. 937–971, April 2006.
15. K. Yasuda, A. Ide, and N. Iwasaki, “Adaptive particle swarm optimization,” *SMC’03 Conf. Proceedings. 2003 IEEE Int. Conf. Syst. Man Cybern. Conf. Theme - Syst. Secur. Assur. (Cat. No.03CH37483)*, vol. 2, pp. 1554–1559, October 2003.

16. I.C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection,” *Inf. Process. Lett.*, vol. 85, pp. 317–325, March 2003.

17. Y.-L. Zheng, L.-H. Ma, L.-Y. Zhang, and J.-X. Qian, “On the convergence analysis and parameter selection in particle swarm optimization,” *Proc. 2003 Int. Conf. Mach. Learn. Cybern.*, vol. 3, pp. 1802–1807, November 2003.

18. C.W. Cleghorn and A.P. Engelbrecht, “A generalized theoretical deterministic particle swarm model,” *Swarm Intell.*, vol. 8, pp. 35–59, January 2014.

19. M. Kubota and T. Saito, “A discrete particle swarm optimizer for multi-solution problems,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E95-A, pp. 406–409, 2012.

20. K. Tamura and K. Yasuda, “Spiral optimization - A new multipoint search method,” *Proc. 2011 IEEE Int. Conf. Syst. Man, Cybern.*, pp. 1759–1764, October 2011.

21. T. Shindo and K. Jin’no, “On a reacceleration mechanism for particle swarm optimizer,” *J. Signal Process.*, vol. 15, pp. 407–416, November 2011.

22. Y. Yamanaka and T. Tsubone, “A basic study on particle swarm optimization based on chaotic spike oscillator dynamics,” *Proc. 2012 Int. Conf. Neural Inf. Process.*, pp. 199–207, November 2012.

23. Y. Yamanaka and T. Tadashi, “An idea of novel optimizer using swarm of chaotic dynamical particles,” *Proc. 2013 Int. Symp. Nonlinear Theory its Appl.*, pp. 240–243, September 2013.

24. J. Chuanwen and E. Bompard, “A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation,” *Math. Comput. Simul.*, vol. 68, pp. 57–65, February 2005.

25. A.H. Gandomi, G.J. Yun, X.-S. Yang, and S. Talatahari, “Chaos-enhanced accelerated particle swarm optimization,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, pp. 327–340, February 2013.

26. J. Chuanwen and E. Bompard, “A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment,” *Energy Convers. Manag.*, vol. 46, pp. 2689–2696, October 2005.

27. L.-Y. Chuang, C.-J. Hsiao, and C.-H. Yang, “Chaotic particle swarm optimization for data clustering,” *Expert Syst. Appl.*, vol. 38, pp. 14555–14563, November 2011.

28. L.D.S. Coelho and V.C. Mariani, “A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch,” *Chaos, Solitons & Fractals*, vol. 39, pp. 510–518, January 2009.

29. B. Alatas, E. Akin, and A.B. Ozer, “Chaos embedded particle swarm optimization algorithms,” *Chaos, Solitons & Fractals*, vol. 40, pp. 1715–1734, May 2009.

30. B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, “Improved particle swarm optimization combined with chaos,” *Chaos, Solitons & Fractals*, vol. 25, pp. 1261–1271, September 2005.

31. J. Cai, X. Ma, L. Li, and P. Haipeng, “Chaotic particle swarm optimization for economic dispatch considering the generator constraints,” *Energy Convers. Manag.*, vol. 48, pp. 645–653, February 2007.

32. Y. Song, Z. Chen, and Z. Yuan, “New chaotic PSO-based neural network predictive control for nonlinear process,” *IEEE Trans. neural networks*, vol. 18, pp. 595–600, March 2007.

33. T. Xiang, X. Liao, and K.-W. Wong, “An improved particle swarm optimization algorithm combined with piecewise linear chaotic map,” *Appl. Math. Comput.*, vol. 190, pp. 1637–1645, July 2007.

34. Y. Wang, J. Zhou, H. Qin, and Y. Lu, “Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects,” *Energy Convers. Manag.*, vol. 51, pp. 2893–2900, December 2010.

35. H. Hefny and S. Azab, “Chaotic particle swarm optimization,” *Proc. 2010 Int. Conf. Informatics Syst.*, pp. 1–8, March 2010.
A. Farmahini-Farahani, S.M. Fakhraie, and S. Safari, “SOPC-Based architecture for discrete particle swarm optimization,” Proc. 2007 IEEE Int. Conf. Electron. Circuits Syst., pp. 1003–1006, December 2007.

C. Johnson, G.K. Venayagamoorthy, and P. Palangpour, “Hardware implementations of Swarmng Intelligence - a survey,” Proc. 2008 IEEE Swarm Intell. Symp., vol. 7, pp. 1–9, September 2008.

S. Yuliu and E. Russell C., “Parameter selection in particle swarm optimization,” Evol. Program. VII (V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben, eds.), vol. 1447 of Lecture Notes in Computer Science, pp. 591–600, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

X.-S. Yang, Nature-Inspired Metaheuristic Algorithms: Second Edition, Luniver Press, 2010.

A. Ratnaweera, S. Halgamuge, and H. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” IEEE Trans. Evol. Comput., vol. 8, pp. 240–255, June 2004.

R. Mendes, J. Kennedy, and J. Neves, “The fully informed particle swarm: Simpler, maybe better,” IEEE Trans. Evol. Comput., vol. 8, pp. 204–210, June 2004.

E. Miyagawa and T. Saito, “Particle swarm optimizers with growing tree topology,” IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E92-A, pp. 2275–2282, November 2009.

J.J. Liang, B.Y. Qu, P.N. Suganthan, and A.G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou, China Nanyang Technol. Univ. Singapore, Tech. Rep., vol. 201212, 2013.

B. Alatas and E. Akin, “Chaotically encoded particle swarm optimization algorithm and its applications,” Chaos, Solitons & Fractals, vol. 41, pp. 939–950, July 2009.

T. Tsubone and W. Yasuhiro, “Stabilizing unknown periodic orbits of a chaotic spiking oscillator,” IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E92-A, pp. 1316–1321, May 2009.

I. Shimada and T. Nagashima, “A numerical approach to ergodic problem of dissipative dynamical systems,” Prog. Theor. Phys., vol. 61, pp. 1605–1616, June 1979.

R. Eberhart and Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” Proc. 2000 Congr. Evol. Comput., vol. 1, pp. 84–88, July 2000.

B. Welch, “The generalization of ‘student’s’ problem when several different population variances are involved,” Biometrika, vol. 34, pp. 28–35, 1947.