Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.Doi Number

Easy Particle Swarm Optimization for Nonlinear Constrained Optimization Problems

Hsuan-Yu Tseng[1], Pao-Hsien Chu[2], Hao-Chun Lu[2-3,*], and Ming-Jyh Tsai[4]

[1]Graduate Institute of Business Administration, Fu Jen Catholic University, New Taipei 242, Taiwan.
[2]Department of Cardiology, Chang Gung Memorial Hospital, Taoyuan 333, Taiwan.
[3]Graduate Institute of Business and Management, Chang Gung University, Taoyuan 333, Taiwan.
[4]Department of Information Management, Fu Jen Catholic University, New Taipei 242, Taiwan.

Hsuan-Yu Tseng and Pao-Hsien Chu made equal contributions and serve as co-first authors for this publication.

*Corresponding author: Hao-Chun Lu. (e-mail: bach0809@mail.cgu.edu.tw).

The work was partially supported by the Ministry of Science and Technology of Taiwan under the grant MOST 108-2410-H-030-078-MY2 and MOST 110-2410-H-182-008-MY3.

ABSTRACT Particle swarm optimization (PSO) is a popular stochastic approach for solving practical optimal problems from industries due to its effective performance and few hyperparameters. Nonlinear constrained optimization (NCO) problems frequently cause multiple optimal regions and can cause many infeasible regions in the search space. The state-of-the-art approaches for handling the infeasible regions generated by problems' constraints either block particles' paths or penalize NCO problems' objective values based on the standard updating velocity formula. The standard updating velocity formula introduces difficulties for particles in searching the undiscovered optimal solutions separated by infeasible regions and being mutually restrained on directions by social and cognitive factors. Afterward, the particles cause premature convergence and difficulty searching the undiscovered optimal regions to improve their solutions. Observing the biological ant colony and inspired by lazy ant behavior, this study proposes an easy particle that simulates the lazy ant to diversify the moving direction. Finally, this study integrates the proposed easy particles with referenced PSO-based approaches for solving NCO problems. The experiment results show that the proposed easy particles can effectively reinforce exploration abilities and improve the performances of all referenced PSO-based algorithms to reduce the status of premature convergence in solving NCO problems.

INDEX TERMS Particle swarm optimization; Easy Particles; Exploration; Premature convergence; Nonlinear constrained optimization problems.

I. INTRODUCTION

In nature, the biological swarm establishes a social system for food searching, migration, and defense. Such biological behavior inspires many scientists to develop related theoretical bases. Particle swarm optimization (PSO), proposed by Kennedy and Eberhart [1], is a popular approach in swarm intelligence. PSO has since been applied in many industries from the real world for solving nonlinear optimization problems because of its effective performance and few hyperparameters.

Most optimal problems in practical industries can be expressed the nonlinear constrained optimization (NCO) problems as follows.

$$\text{Min} \quad f(\mathbf{x}) \tag{1}$$

$$\text{s.t.} \quad c_j(\mathbf{x}) \le 0,\ j = 1,\ldots,m\ , \tag{2}$$

$$e_j(\mathbf{x}) = 0,\ j = 1,\ldots,m'\ . \tag{3}$$

where $c_j(\mathbf{x})$ in (2) denote the problem-specific inequality constraints and $e_j(\mathbf{x})$ in (3) denote the problem-specific equation constraints. The problem-specific constraints (2)–(3) are derived from the limitations of resources in practical industries, such as in electromagnetics [2], real-time UAV path planning [3], prediction of seismic slope stability [4], energy development [5][6], incomplete data clustering [7], and production inventory [8]. [9] presented a performance comparison of harmony search (HS), differential evolution (DE), and PSO for the standard benchmark functions. [10] have used PSO, DE, and genetic algorithm (GA) altogether to estimate low atmospheric refractivity profiles from radar sea clutter. Such studies essentially recognized PSO's superiority compared with other algorithms for NO problems[9][11].

The problem-specific constraints in (2)–(3) frequently cause multiple optimal regions and can cause many infeasible regions in the search space. Given that primitive PSO does not discuss any approach for handling the infeasible regions in the search space, many improvements have been developed in the literature. However, the state-of-the-art approaches for handling the infeasible regions in the search space either block particles' paths or penalize NCO problems' objective values on the basis of the standard updating velocity formula. Given that the traditional updating velocity formula does not consider the infeasible regions, those approaches cause the particles (i) difficulty in searching the undiscovered optimal solutions separated by infeasible regions and (ii) to be mutually restrained on directions by social and cognitive factors. Afterward, the particles encounter difficulty in improving their solutions, resulting in premature convergence. In particular, the optimal solutions enclosed by the infeasible regions are difficult to obtain. The randomized velocity is a simple way to avoid the stagnation situation; however, [12][13][14]indicated that randomized velocity always obtained terrible results. Developing a novel approach for PSO is necessary to solve the issue of constraints in NCO problems.

This study solves this issue by observing ant colony behavior. Most ants harvest food with effort in the usual sense, but some ants roam around everywhere and do nothing. Such ants are referred to by biologists as lazy ants. When food shortages occur, most ants cannot do anything but the lazy ants come forward to guide the typical ants to find a new region for harvesting food. Therefore, lazy ants are not lazy; they are looking for additional food sources everywhere. The lazy ant effect is a popular theory in organization management in recent years [15][16][17][18][19]. This study proposes an easy particle that simulates the lazy ant to improve the ability to search the optimal regions separated by infeasible regions and solve the dilemma of mutually restrained directions by social and cognitive factors. This idea is never proposed for PSO in current literature. The advantages and contributions of the proposed easy particle are listed as follows.

(i) Increasing the probability of exploring the search space across the infeasible regions for PSO-based approaches solving the NCO problems.
(ii) Effectively reducing the status of premature convergence for PSO-based approaches solving the NCO problems.
(iii) Current PSO-based approaches can conveniently embed the proposed easy particles to significantly improve the performances of solving the NCO problems.

The remainder of this study is as follows. Section II investigates the literature of current PSO-based approaches. Section III introduces the proposed easy particle. Section IV presents some numerical experiments to demonstrate the advantages of the proposed easy particle. Section V provides some concluding remarks.

II. LITERATURE REVIEW

PSO was first proposed by Kennedy and Eberhart [1] to solve optimization problems, and Shi and Eberhart [12] introduced a new parameter called inertia weight in PSO. On this basis [20], the main processes of PSO include initialization, fitness, updating velocity, and moving particle. In this study, the notations are defined as follows:

Notation	Meaning
T	The total number of iterations.
t	The index of iterations where $t \in \{1,2,...,T\}$.
G	The size of swarm.
i	The index of particles in swarm where $i \in \{1,2,...,G\}$.
d	The dimension of the optimization problem.
ω	The inertia weight factor.
c_1	The cognitive factor.
c_2	The social factor.
$\mathbf{r}_1$, $\mathbf{r}_2$	The random vector where $\mathbf{r}_i = (r_{i,1},\ldots,r_{i,d})$, $r_{i,j} \in [0,1]$, $i = 1,2$, and $j = 1,...,d$.
$\mathbf{x}_i^{(t)}$	The position vector of particle i at iteration t where $\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)},\ldots,x_{i,d}^{(t)})$.
$\mathbf{v}_i^{(t)}$	The velocity vector of particle i at iteration t where $\mathbf{v}_i^{(t)} = (v_{i,1}^{(t)},\ldots,v_{i,d}^{(t)})$.
$S^{(t)}$	The swarm at iteration t where $S^{(t)} = \{\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)},...,\mathbf{x}_G^{(t)}\}$.
$\mathbf{p}_i$	The best position vector of $\{\mathbf{x}_i^{(1)},\ldots,\mathbf{x}_i^{(t)}\}$ for particle i from 1st to tth iterations where $\mathbf{p}_i = (p_{i,1},\ldots,p_{i,d})$.
$\mathbf{p}_g$	The best position vector of $\{\mathbf{p}_1,\ldots,\mathbf{p}_G\}$ for entire swarm from 1st to tth iterations where $\mathbf{p}_g = (p_{g,1},\ldots,p_{g,d})$.

On the basis of Shi and Eberhart [20], the formulas of updating velocity and moving are shown in Equations (4) and (5).

$$\mathbf{v}_i^{(t+1)} = \omega\mathbf{v}_i^{(t)} + c_1\mathbf{r}_1(\mathbf{p}_i - \mathbf{x}_i^{(t)}) + c_2\mathbf{r}_2(\mathbf{p}_g - \mathbf{x}_i^{(t)}) , \tag{4}$$

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)} . \tag{5}$$

In Equation (4), $c_1\mathbf{r}_1(\mathbf{p}_i - \mathbf{x}_i^{(t)})$ and $c_2\mathbf{r}_2(\mathbf{p}_g - \mathbf{x}_i^{(t)})$ present the cognitive movement and social movement, respectively. The social and cognitive movements are based on the individual best position   and group best position   where and   are only storing feasible solutions. The primitive PSO is designed for nonlinear optimization problems; however, it is inappropriate to solve the NCO problems. Given that the NCO problems are solved through traditionally updating velocity formulas in Equations (4)–(5), particles will frequently be moved to the infeasible regions constructed by the problem-specific constraints in (2)–(3). Given that no schema exists in primitive PSO for handling the particles entering the infeasible regions, many schemas are proposed in the literature to solve constraints in NCO problems [21][22]. The following contexts describe the favored schemas of handling the constraints in (2)–(3) for PSO.

A. Penalty schema

The most popular idea is the penalty schema because it is straightforward in addressing the constraints issue. The penalty schema for PSO is calculating the penalty value of constraint violations and adding this penalty value to the objective function in (1) [23][24][25][26]. Based on Koziel and Michalewicz [24] and O. Yeniay [26], the NCO problems with the penalty schema can be expressed as follows:

Min $\quad f(\mathbf{x})+P(\mathbf{x}) \qquad (6)$

s.t. $\quad$ (2), (3),

$$P(\mathbf{x}) = p_1 N + p_2\left(\sum_{j=1}^{m} v_j(\mathbf{x}) + \sum_{j=1}^{m'} v'_j(\mathbf{x})\right), \tag{7}$$

$$v_j(\mathbf{x}) = \max\{0, c_j(\mathbf{x})\},\ j=1,\ldots,m\,, \tag{8}$$

$$v'_j(\mathbf{x}) = |\,e_j(\mathbf{x})\,|,\ j=1,\ldots,m'\,. \tag{9}$$

where $P(\mathbf{x})$ in Equation (7) is the penalty value that will be added to the objective function, p_1 and p_2 are the penalty coefficients that the user can define, N is the number of violated constraints, $v_j(\mathbf{x})$ in Equation (8) is the penalty value of the jth violated constraints in (2), and $v'_j(\mathbf{x})$ in Equation (9) is the penalty value of the jth violated constraints in (3).

The penalty schema is very convenient because it does not involve modifications of the used PSO or specialized operators' development for the constraints. However, $f(\mathbf{x})+P(\mathbf{x})$ in (6) results in an infeasible solution for NCO problems. For example, supporting the objective function $f(\mathbf{x})=-10$ with $P(\mathbf{x})=0$ in (6) is a global optimum and $f(\mathbf{x})=-100$ with $P(\mathbf{x})=1$ in (6) is an infeasible solution. Then, the final objective value will be -100 with a slight penalty factor of $P(\mathbf{x})=1$. However, $f(\mathbf{x})=-100$ is an infeasible solution. [27] proposed a mutation function to solve the specific constraint of bin-packing problems when particles encounter the boundaries of constraints. However, the concept of mutation is similar to that of GA, that is, the fitness function typically drives the computational load.

B. Boundary schema

Some studies treat the infeasible regions as particles' boundaries. Thus, they will let the particles likely explore feasible regions and avoids the particles entering infeasible regions. Sanaz et al. [28] and Li-Yeh et al. [29] proposed a boundary schema to handle constraints in NCO problems. When a particle moves to an infeasible region, [28] and [29] drag the particle back to a closer feasible position against the infeasible region's boundary, the particle moves to a position, $\mathbf{x}_i^{(temp)}$, in the infeasible region, and the particle will be dragged to the position $\mathbf{x}_i^{(t+1)}$. He et al. [30] proposed a fly-back schema to handle constraints in NCO problems. This schema is straightforward, and it is effortless to implement in PSO. In this schema, the particles are dragged back into the original feasible solution.

However, the boundary schema proposed by [28] and [29] does not offer a precise formula to calculate the closer feasible position, and the distance between $\mathbf{x}_i^{(temp)}$ and $\mathbf{x}_i^{(t+1)}$ is difficult to calculate if the objective function or constraints are nonlinear. The fly-back schema proposed by [30] will generate many dummy moves in evolution processes and reduce PSO efficiency. [31] indicated that taking the bounds as the corresponding positions of new particles in [28] and [29] and keeping the positions of particles unchanged in [30] will reduce the diversity of the particles in the search process. Moreover, if the infeasible regions completely enclose the best optimal solution, the particles cannot achieve the best optimal solution until the evolution processes are terminated.

In addition to directly solving the constraints issue, many schemas are available for PSO to increase its exploration ability. The following contexts describe the most popular schemas.

C. Reinforcement best position schema

In literature, two main variants of PSO-based approaches on the number of best position, $\mathbf{p}_g$, are used, namely global and local search PSOs. The global search PSO is the primitive PSO proposed by Shi and Eberhart [20], and only one $\mathbf{p}_g$ exists in Equation (4) for all particles where the $\mathbf{p}_g$'s neighborhoods are the entire swarm $S^{(t)}$. For local search PSO, the swarm will be divided into multiple groups, and each group has the best position. $\mathbf{p}_{ng}$ is denoted as the best position for group n in the swarm. The formula of the local search PSO can be changed to Equations (10)–(11).

$$\mathbf{v}_{Local,i}^{(t+1)} = \omega\mathbf{v}_i^{(t)} + c_1\mathbf{r}_1(\mathbf{p}_i - \mathbf{x}_i^{(t)}) + c_2\mathbf{r}_2(\mathbf{p}_{ng} - \mathbf{x}_i^{(t)})\,, \tag{10}$$

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_{Local,i}^{(t+1)}\,. \tag{11}$$

The global search PSO in Equations (4)–(5) promotes exploitation because all particles are attracted by one group best position $\mathbf{p}_g$, which will converge rapidly toward the same point. In contrast to the local search PSO, the updating velocity formula in Equations (10)–(11) has better exploration effects because many best positions $\mathbf{p}_{ng}$ exist for the related groups, and the related groups' best positions will attract the particles. This process is a tradeoff between the global and local search PSO. Parsopoulos and Vrahatis [32] proposed a unified PSO (UPSO) that combines the exploitation feature of global search PSO and the exploration feature of local search PSO. The UPSO's updating velocity formulas can be expressed by Equations (4), (10), (12), and (13).

$$\mathbf{v}_{UPSO,i}^{(t+1)} = (1-u)\mathbf{v}_{Local,i}^{(t+1)} + u\mathbf{v}_i^{(t+1)}\,, \tag{12}$$

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_{UPSO,i}^{(t+1)}\,. \tag{13}$$

where $\mathbf{v}_i^{(t+1)}$ in Equation (4) and $\mathbf{v}_{Local,i}^{(t+1)}$ in Equation (10) are global and local velocities. Afterward, Parsopoulos and Vrahatis [33] indicated that the UPSO with $u=0.5$ and UPSO with mutation (UPSOm) proved the most promising scheme on their optimization problems examined.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3110708, IEEE Access

[34] propose a differential evolution PSO (DEPSO) that uses the operators (crossover, recombination, and mutation) of the genetic algorithm to create diversified best group positions $\mathbf{p}_g$ for improving the exploitation ability of PSO. The particles of DEPSO converge faster than those in traditional PSOs in our experiments. However, DEPSO results in premature convergence for solving the NCO problems. This reinforcement best position schema may improve the exploitation ability of PSO, but it still does not solve the mutual restraint on directions by social and cognitive factors.

D. Reinforcement updating velocity formula schema

[35] proposed a new updating velocity updating method, named the foothold concept, to solve constraints in the NCO problems. If the particle is moved into an infeasible region, the repairing process will be started until a new feasible particle is found. The new feasible particle is calculated using a linear combination between the infeasible particle and the randomly selected feasible particle. [31] adjusted the original velocity obtained by Equation (4) according to the number of particles moving outside the feasible region, and the adjustment original updating velocity formula can be expressed as follows:

$$\mathbf{v}_i^{(t+1)} = (1+\beta)^{\alpha}\,\mathbf{v}_i^{(t)} \ \text{ if } \ N_{out,i} < 1\,,$$

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} \,/\left(1 + N_{out,i} / N_T\right)^{\gamma} \ \text{ if } \ N_{out,i} \geq 1\,.$$

where α, β, and γ are positive constants; $N_{out,i}$ is the number of moves outside the feasible region since the last velocity adjustment for particle i; and N_T is the number of iterative cycles among velocity adjustments.

[36] indicated that an excellent PSO-based algorithm needs to consider both abilities of exploration and exploitation. The reinforcement best position and reinforcement updating velocity formula schemas enhance the traditional updating velocity formula's searchability in Equation (4). However, the traditional updating velocity formula in Equation (4) does not consider the status of particles entering the infeasible regions. It will cause the social factor (c_2) and cognitive factor (c_1) to be mutually restrained on particles' directions. The reinforcement best position and reinforcement updating velocity formula schemas based on the traditional updating velocity formula in Equation (4) are inappropriate to solve the NCO problems. The penalty schema can easily handle the constraint's issue but pollute the objective function. The boundary schema blocks the paths of particles.

Based on the above discuss, Table I lists the advantages and disadvantages of the literature.

TABLE I
THE ADVANTAGES AND DISADVANTAGES OF THE REFERENCES

Schema and references	Advantage	Disadvantage
A. Penalty schema [23] [24] [25] [26] [27].	• It is straightforward in addressing the constraints issue. • It does not involve modifications of the used PSO.	• It results in an infeasible solution for NCO problems.
B. Boundary schema [28] [29] [30].	• It let the particle explore feasible regions and avoids the particles entering infeasible regions.	• It does not offer a formula to calculate the distance between the boundary and current infeasible position. • It generates many dummy moves. • It cannot escape a local optimal region completely enclosed by infeasible regions.
C. Reinforcement best position schema [32] [33] [34] [43].	• The local search PSO improves the exploration ability because many best positions $\mathbf{p}_{ng}$ exist for the related groups.	• It remains the issue of the tradeoff between global search PSO and local search PSO. • It still does not solve the mutual restraint on directions.
D. Reinforcement updating velocity formula schema [35] [36].	• It enhances the traditional updating velocity formula's searchability.	• It does not consider the status of particles entering the infeasible regions.

Therefore, the current PSO-based approaches still lack a novel schema to explore undiscovered regions blocked by the infeasible regions generated by constraints (2)–(3) [37]. For solving premature convergence, the current PSO-based approaches need different behavior of particles to break through the traditional particle that restricted by the social (c_2) and cognitive (c_1) factors.

In a biological study, [38] observed that not all ants were active in the ant colony. In 1999, Gordon and Mehdiabadi [39] observed that the ant colony does the task allocation according to the environment and food reserved. [40] found that most ants harvest food with effort, but the lazy ants roam around everywhere and seemingly do nothing in the ant colony. Without those lazy ants, the ant colony cannot change the harvest target immediately during food shortages. Lazy ants spent the most time exploring and detecting unknown regions to continuously ensure the colony owned food sources. More precisely, lazy ants are not lazy; they are not tempted by present foods but spend the most time exploring additional food sources everywhere. [41] indicated that the inactive lazy ants are also a biological activity of swarm. Inspired by the lazy ant, this study proposes an easy particle that simulates lazy ant behavior. This study will embed the proposed easy particles into referenced PSO-based approaches to effectively address the stagnant particle and solve the NCO problems.

III. PROPOSED EASY PARTICLE

A. Easy particle concept

To reiterate, [40][41] indicated that lazy ants are not tempted by present food but spend the most time exploring anywhere. [42] analyzed the ants' trails and obtain that the lazy ants move straight in most times, sometimes turn right or turn left, and move backward within seldom times. Figure 1 represents the excepted exploration trajectory of easy particles. The proposed easy particles' movements are not influenced by the individual best position $\mathbf{p}_i$ and group best position $\mathbf{p}_g$ in Equation (4) (as the present food for the lazy ant). Thus, the lazy ants have more probabilities to find the undiscovered optimal solutions and to never be mutually restrained on directions by social and cognitive factors.

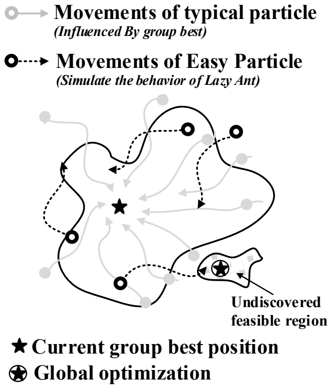

FIGURE 1. **Expected exploration movements of easy particle.**

Based on the analysis in [42], the trails of Lazy Ants are not purely random. The easy particle is the same as Lazy Ant. The trails of easy particle are not simply purely random moving. [12] indicated that the randomized velocity always obtained terrible results, such as stochastic particles [13] or wandering particles [14].

The concept of movements of the easy particle is shown in Figure 2. In Figure 2, an easy particle in each iteration has four directions relative to its previous movement: move forward, turn left, turn right, and move backward. To simulate lazy ant behavior, the easy particle moves forward most times, sometimes turns right or left, and seldom goes backward. Therefore, the design principles of the easy particle are shown in Table II.

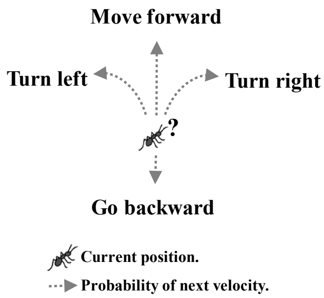

FIGURE 2. **Concept of movements of easy particle.**

TABLE II
DESIGN PRINCIPLES OF EASY PARTICLE

Direction	Purpose	Probability	Directions of dimensions in velocity
Moving forward	Moving forward to ensure the particle is exploring the undiscovered regions.	High.	Greater and equal than 50% of the dimension's directions in velocity is the same as its previous ones.
Turning left/right	Both exploration and exploitation orientation. Turning left/right to keep the balance of exploring the undiscovered regions and exploiting the current region.	Medium.	Less than 50% of the dimension's directions in velocity are the same as its previous ones, and less than 50% of the dimension's directions in velocity are the opposite of its previous ones.
Moving backward	Fully exploitation orientation. Move backward to refine the solution by exploiting the current region.	Low.	Greater and equal than 50% of the dimension's directions in velocity is the opposite of its previous ones.

B. Moving schema of easy particle

For implementing the easy particle based on Table II, seven hyperparameters are introduced in Table III.

TABLE III
HYPERPARAMETERS OF EASY PARTICLE

Notation	Meaning
r_{EP}	$r_{EP} \in [0,1)$. The rate of easy particles in the swarm.
r_{FW}	$r_{FW} \in [0,1]$. The rate of easy particle's directions moving forward
r_{TU}	$r_{TU} \in [0,1]$. The rate of easy particle's directions turning left and turning right, respectively.
r_{BW}	$r_{BW} \in [0,1]$. The rate of easy particle's directions going backward.
r_S	$r_S \in [0,1]$. The rate of dimension's directions in easy particle's velocity that are the same as its previous ones.
r_O	$r_O \in [0,1]$. The rate of dimension's directions in easy particle's velocity that are the opposite of its previous ones.
r_R	$r_R \in [0,1]$. The rate of random direction dimensions.

In Table III, r_{EP} decides the size of easy particles in a swarm; more easy particles will improve the swarm's exploration ability but counteract the exploitation. In our experiential, $r_{EP} = 0.1$ can appropriately increase the exploration ability and retain its exploitation ability as much as possible. The other hyperparameters must follow constraints (14) and (15).

$$r_{FW} + 2r_{TU} + r_{TR} = 1,\ r_{FW} > r_{TU} > r_{BW}, \tag{14}$$

$$r_S + r_O + r_R = 1. \tag{15}$$

The appropriate ranges of r_S and r_O for each direction are deduced by Proposition.

PROPOSITION 1

On the basis of the design principles in Table II *and constraint* (15)*, the appropriate ranges of* r_S *and* r_O *for each direction of easy particle are listed as follows.*

(i) *Moving forward:* $r_S \geq 0.75$. (16)

(ii) *Turning left/right:* $0.25 < r_S, r_O < 0.75$. (17)

(iii) *Moving backward:* $r_O \geq 0.75$. (18)□

Proof

(i) Let the direction of dimension in the previous easy particle's velocity be positive, then the dimensions belonging to r_S are positive, and the dimensions belonging to r_O are negative. Denote $D(r_S)$, $D(r_O)$, and $D(r_R)$ as the direction values of dimensions belonging to r_S, r_O, and r_R, respectively. Then, $D(r_S) = r_S$, $D(r_O) = -r_O$, and $D(r_R) \in [-r_R, r_R]$. (ii) Base in Table II, an easy particle moving forward should follow constraint (19).

$$D(r_S) + D(r_O) + D(r_R) \geq 0.5. \quad (19)$$

In the most conservative case, $D(r_R) = -r_R$, constraint (19) can be expressed as constraint (20).

$$r_S - r_O - r_R \geq 0.5. \quad (20)$$

Based on constraint (15), constraint (20) deduces constraint (16). (iii) An easy particle turning left/right should follow constraint (21).

$$-0.5 < D(r_S) + D(r_O) + D(r_R) < 0.5. \quad (21)$$

In the most conservative case for the lower bound of constraint (21), $D(r_R) = -r_R$; constraint (21) can be expressed as constraint (22).

$$-0.5 < r_S - r_O - r_R < 0.5. \quad (22)$$

Based on constraint (15), constraint (22) deduces constraint (23).

$$0.25 < r_S < 0.75. \quad (23)$$

In the most conservative case for the upper bound of constraint (21), $D(r_R) = r_R$; constraint (21) can be expressed as constraint (24).

$$-0.5 < r_S - r_O + r_R < 0.5. \quad (24)$$

Constraint (24) deduces constraint (25) based on constraint (15).

$$0.25 < r_O < 0.75. \quad (25)$$

Constraints (23) and (25) can be expressed as constraint (17). (iv) An easy particle moving backward should follow constraint (26).

$$D(r_S) + D(r_O) + D(r_R) \leq -0.5. \quad (26)$$

In the most conservative case, $D(r_R) = r_R$, constraint (26) can be expressed as constraint (27).

$$r_S - r_O + r_R \leq -0.5 \quad (27)$$

Constraint (27) deduces constraint (18) based on constraint (15). ■

Let d as the number of dimensions of the NCO problem. Denote S_S as a set of dimensions whose directions are the same as their previous ones in velocity. Denote S_O as a set of dimensions whose directions are the opposite of their previous ones in velocity. Denote $S(v)$ as the sign of the element v in velocity. r_3 and r_4 are random variables that follow a uniform distribution. The following processes can calculate the velocity of the easy particle:

(1) Decide the moving direction of the easy particle by a random variable r_3.
   - $0 \leq r_3 < r_{FW}$: moving forward.
   - $r_{FW} \leq r_3 < r_{FW} + r_{TU}$: turning left.
   - $r_{FW} + r_{TU} \leq r_3 < r_{FW} + 2r_{TU}$: turning right.
   - $r_{FW} + 2r_{TU} \leq r_3 \leq 1$: moving backward.

(2) Move the easy particle with random variable r_4.
   - (i) Randomly pick up $\lceil dr_S \rceil$ and $\lceil dr_O \rceil$ dimensions of velocity for the sets S_S and S_O, respectively.
   - (ii) Calculate the velocity of easy particle through the updating velocity formulas (28).

$$v_{k,i}^{(t+1)} = \begin{cases} S(v_{k,i}^t) r_4 (v_{\min} + (v_{\max} - v_{\min})), v_{k,i}^t \in S_S \\ -S(v_{k,i}^t) r_4 (v_{\min} + (v_{\max} - v_{\min})), v_{k,i}^t \in S_O \\ 2(r_4 - 0.5)(v_{\min} + (v_{\max} - v_{\min})), \text{otherwise} \end{cases}. \quad (28)$$

C. PSO-based approach embedded with easy particle

The PSO-based approach embedded with easy particles is shown in Figure 3. Major processes in Figure 3 are the same as those of the standard PSO-based approach except for updating the velocity of the easy particle. The typical particles calculate velocity by PSO-based approaches' updating velocity formulas, whereas the easy particle calculates velocity by formula (28). When the easy particle obtains a better feasible result, the best position $\mathbf{p}_g$ of the entire swarm will be updated. Therefore, the typical particles will be influenced by the new group best position $\mathbf{p}_g$.

That is, the typical particles moving in the traditional manner attracted by social and cognitive factors and easy particles moving in a diversity trajectory simulated the behavior of the lazy ant. Hence, the particle swarm closer to the ants' behavior in nature.

Moreover, the easy particles only influence the convergence of PSO when they search a better position than the current group best position, and the easy particles will help the whole swarm to find a better solution. The easy particles impossibly find the better solution continuously; therefore, the solution must be converged within an acceptable evolution time in the late evolution process.

IEEE Access Multidisciplinary : Rapid Review : Open Access Journal

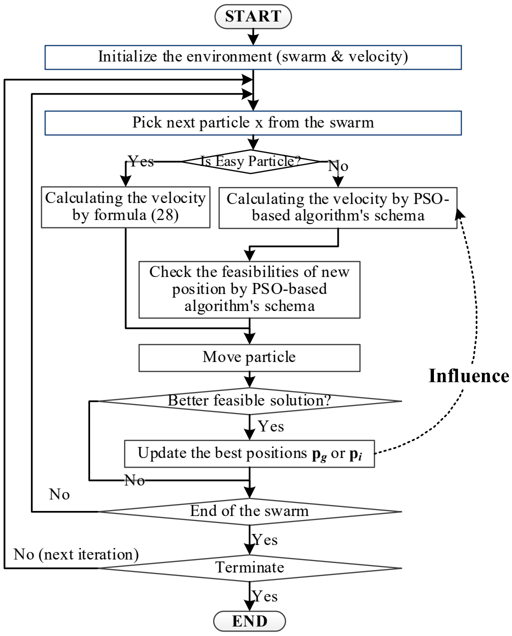

FIGURE 3. The flowchart of PSO-based approach embedded with the easy particles.

IV. EXPERIMENTAL RESULTS

This study uses pretest–posttest design to validate the effectiveness of the proposed easy particle. Five PSO-based algorithms are used as the referenced algorisms. Pretests are the referenced algorithms without easy particles, and the referenced algorithms with easy particles are posttests. Given that the penalty schema should pollute the objective function, it will cause the best group position as an infeasible solution and distort the behavior of the easy particle. The easy particle is inappropriate for the penalty schema. Therefore, the following five referenced algorithms do not include the penalty schema.

Algorithm 1	The traditional PSO that was proposed by Shi and Eberhart [20].
Algorithm 2	The UPSOm proposed by Parsopoulos and Vrahatis [33] incorporated a stochastic parameter that imitates mutation in UPSO [32] to enhance the exploration capabilities.
Algorithm 3	The RWDEPSO proposed by Lin et al [43] is a mutate DEPSO [34] in which the inertia weight was based on a random number that obeys the standard state distribution.
Algorithm 4	The algorithm proposed by He et al. [30] uses the fly-back manner to handle the infeasible solution spaces in problems.
Algorithm 5	The PSO+ proposed by Kohler et al. [35] uses the crossover operator between the infeasible particle and feasible particle until a new feasible particle can solve the infeasible particles.

Five examples (three well-known benchmark problems and two well-known real-world optimization problems) are used in this section. 25 pretest–posttest experiments (five referenced algorithms with five examples) are performing in this section to demonstrate the effectiveness of the proposed easy particle.

The values of standard PSO hyperparameters and easy particle hyperparameters are listed as following: (i) swarm size (G) is 30, (ii) executing time per run is 60 seconds, (iii) inertia weight (w) is descending by iteration from 0.9 to 0.4 on the basis of Shi and Eberhart [20], (iv) cognitive and social parameters (c_1, c_2) are 1.7 based on Bonyadi and Michalewicz [44], (v) r_{EP} = 0.1, (vi) r_{FW} = 0.5, r_{TU} = 0.2, and r_{BW} = 0.1 based on constraint (14). The value ranges of r_S, r_O, and r_r are following constraints (15)-(18). Based on constraints (15)-(18), the full factorial design of experiment (DOE) for r_S, r_O, and r_r is conducted to find the appropriate settings listed in Table IV.

TABLE IV
VALUES OF HYPERPARAMETERS IN THIS STUDY

Direction	Values of r_S, r_O, r_r
Moving forward.	$r_S = 0.75,\ r_O = 0,\ r_r = 0.25.$
Turning left/right.	$r_S = 0.35,\ r_O = 0.35,\ r_r = 0.3.$
Going backward.	$r_S = 0,\ r_O = 0.75,\ r_r = 0.25.$

All algorithms are implemented by C# in Microsoft Visual Studio Community 2019. All pretest–posttest experiments are run on a PC equipped with an Intel® CoreTM i7-930 CPU, 16 GB RAM, and Windows 10 operating system. Each pretest–posttest experiment is performed at 30 runs, and it will let the average of objective values following the normal distribution based on the central limit theorem. Finally, this study uses paired *t*-test to verify the performance of each pretest–posttest experiment. The format of the result of the paired *t*-test is "*p*-value (* or ** or ***)" where "*", "**", and "***" denote *p*-value <0.05, *p*-value <0.01, and *p*-value <0.001, respectively. Standard deviation (Stdev) and quartile deviation (QD) are used to verify the stability of each pretest–posttest experiment. The tables provide precision experiment results, while the boxplots provide a visual for observing the difference simply. The experiment results of all examples are shown as follows.

Example 1. Rosenbrock Problem

The Rosenbrock problem was first proposed by Rosenbrock [45]. It is a famous testing problem that includes two local optimal regionals. The constraints form several infeasible regions within the search space that restrict the movements of traditional particles. They usually cause the particles to fall into premature convergence. Experiment results in Table V and Figure 4 show that the proposed easy particles help the PSO swarm to discover the new feasible region and obtain better objective values.

$$\text{Min} \quad f(\mathbf{x}) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$$
$$s.t. \quad (x_1-1)^3 - x_2 + 1 \le 0\,,\ x_1 + x_2 - 2 \le 0\,.$$

TABLE V
EXPERIMENT RESULT OF EXAMPLE 1

	Algorithm				
	1	2	3	4	5

Pretest	Max	0.9989	0.9989	0.9989	0.9994	0.9991
	Mean	0.9989	0.9989	0.9989	0.9656	0.9989
	Min	0.9989	0.9989	0.9989	0.0000	0.9989
	Stdev	**0.0000**	**0.0000**	**0.0000**	0.1824	**0.0000**
	QD	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Posttest	Max	0.0281	0.1261	0.9989	0.6517	0.4809
	Mean	**0.0087**	**0.0218**	**0.3819**	**0.1101**	**0.0798**
	Min	0.0001	0.0002	0.0000	0.0028	0.0001
	Stdev	0.0067	0.0297	0.4686	**0.1547**	0.1260
	QD	0.0033	0.0137	0.4994	0.0626	0.0253
p-value		0.0000***	0.0000***	0.0000***	0.0000***	0.0000***

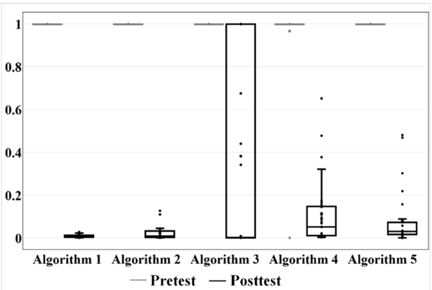

FIGURE 4. **Box plot of Example 1.**

Example 2. Three-hump Camelback Problem

The three-hump camelback problem that is modified from [46] has three local optimal regionals. It increased the challenge for the typical particle to find the global optimal region when it has lacked exploring capability. Experiment results in Table VI and Figure 5 show that the proposed easy particles improve the exploring capability of all referenced algorithms and obtain better objective values.

$$\text{Min} \quad f(\mathbf{x}) = 2x_1^2 - 1.081x_1^4 + \tfrac{1}{6}x_1^6 - x_1x_2 + x_2^2 + 0.01x_1$$

$$\text{s.t.} \quad -2.5 \le x_1 \le 2.5\,,\ -2.5 \le x_2 \le 2.5\,.$$

TABLE VI
EXPERIMENT RESULT OF EXAMPLE 2

		Algorithm				
		1	2	3	4	5
Pretest	Max	0.0088	0.0088	0.0000	0.3263	0.3365
	Mean	-0.0164	-0.0085	-0.0100	0.0094	0.0167
	Min	-0.0272	-0.0272	-0.0272	-0.0272	-0.0272
	Stdev	0.0147	0.0148	0.0133	0.0615	0.0884
	QD	0.0136	0.0136	0.0136	0.0044	0.0092
Posttest	Max	-0.0272	-0.0272	0.0000	0.0000	0.0000
	Mean	**-0.0272**	**-0.0272**	**-0.0200**	**-0.0218**	**-0.0263**
	Min	-0.0272	-0.0272	-0.0272	-0.0272	-0.0272
	Stdev	**0.0000**	**0.0000**	**0.0122**	**0.0110**	**0.0050**
	QD	**0.0000**	**0.0000**	**0.0102**	**0.0000**	**0.0000**
p-value		0.0002***	0.0000***	0.0045**	0.0058**	0.0063**

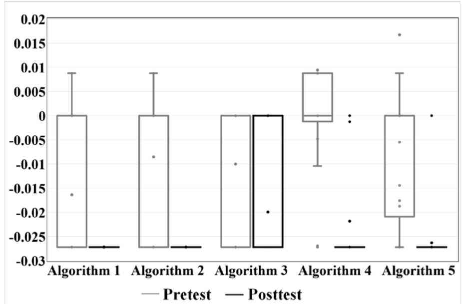

FIGURE 5. **Box plot of Example 2.**

Example 3. Townsend Problem

The Townsend problem, modified from Townsend [47], contains trigonometric functions in objective function and constraints. A ridge is present around the global optimization that restricts the crossing of typical particles. Experiment results in Table VII and Figure 6 show that the proposed easy particles led the PSO swarm across the ridge successfully.

$$\text{Min} \quad f(\mathbf{x}) = -\left[\cos((x_1 - 0.1)x_2)\right]^2 - x_1 \sin(3x_1 + x_2)$$

$$\text{s.t.} \quad x^2 + y^2 < (2\cos t - 0.5\cos 2t - 0.25\cos 3t - 0.125\cos 4t)^2 + (2\sin t)^2\,,\quad t = \text{Atan2}(x_1, x_2)\,.$$

TABLE VII
EXPERIMENT RESULT OF EXAMPLE 3

		Algorithm				
		1	2	3	4	5
Pretest	Max	-1.6397	-1.6397	-1.6397	-1.6397	-1.6397
	Mean	-1.8314	-1.7689	-1.8088	-1.6777	-1.6696
	Min	-2.0240	-2.0240	-2.0236	-2.0240	-2.0219
	Stdev	0.1827	0.1701	0.1695	0.0938	0.0668
	QD	0.1814	0.1697	0.1672	0.0099	**0.0000**
Posttest	Max	-2.0159	-2.0192	-1.6595	-2.0135	-2.0087
	Mean	**-2.0225**	**-2.0229**	**-1.9510**	**-2.0205**	**-2.0204**
	Min	-2.0240	-2.0240	-2.0240	-2.0239	-2.0240
	Stdev	**0.0019**	**0.0015**	**0.1483**	**0.0031**	**0.0037**
	QD	**0.0009**	**0.0010**	**0.0001**	**0.0024**	0.0018
p-value		0.0000***	0.0000***	0.0011**	0.0000***	0.0000***

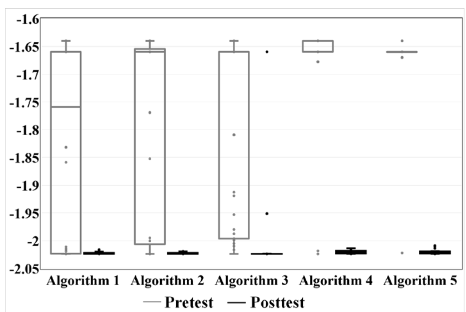

FIGURE 6. **Box plot of Example 3.**

Example 4. Welded Beam Design Problem

Example 4 is a particle design problem for welded beams, which was first proposed by Rao [48]. The four decision variables are the thickness of the welded joint ($h = x_1$), the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3110708, IEEE Access

length of the welded joint ($l = x_2$), the width of the beam ($t = x_3$), and the thickness of the beam ($b = x_4$). It is also a popular machine design optimization problem that includes nonlinear objective function and nonlinear constraints. Experiment results in Table VIII and Figure 7 show that the proposed easy particles help all referenced algorithms obtain improved objective value and stability.

Min $1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$

s.t. $\tau(\mathbf{x}) - \tau_{\max} \le 0$, $\sigma(\mathbf{x}) - \sigma_{\max} \le 0$,

$x_1 - x_4 \le 0$,

$0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0$,

$0.125 - x_1 \le 0$,

$\delta(\mathbf{x}) - \delta_{\max} \le 0$, $P - P_c(\mathbf{x}) \le 0$,

$0.1 \le x_i \le 2,\ i = 1,4$, $0.1 \le x_i \le 10,\ i = 2,3$,

where

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P(L + \frac{x_2}{2}),$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2},$$

$$J = 2\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2\right],$$

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4x_3^2},\ \delta(\mathbf{x}) = 4PL^3/Ex_3^3x_4,$$

$$P_c(\mathbf{x}) = \frac{4.013\sqrt{(EGx_3^2x_4^6)/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$P = 6000lb$, $L = 14in$, $E = 30\times10^6\ psi$,

$G = 12\times10^6\ psi$, $\tau_{\max} = 13{,}600\,psi$,

$\sigma_{\max} = 30{,}000\,psi$, and $\delta_{\max} = 0.25in$.

TABLE VIII
EXPERIMENT RESULT OF EXAMPLE 4

		Algorithm 1	2	3	4	5
Pretest	Max	7.4251	3.7842	3.6703	11.9458	14.4426
	Mean	3.7221	3.0462	3.1755	5.3372	5.6572
	Min	2.2600	2.5067	2.3931	2.4842	2.4982
	Stdev	2.0873	0.3558	0.3501	2.4744	2.8029
	QD	0.4084	0.2809	0.2425	1.2801	1.2943
Posttest	Max	2.7116	2.8223	2.7357	2.5961	2.5795
	Mean	**2.5926**	**2.6465**	**2.5839**	**2.4908**	**2.4919**
	Min	2.4503	2.4590	2.3812	2.4267	2.4096
	Stdev	**0.0645**	**0.0975**	**0.1236**	**0.0337**	**0.0423**
	QD	**0.0401**	**0.0721**	**0.0935**	**0.0206**	**0.0380**
p-value		0.0030**	0.0000***	0.0000***	0.0000***	0.0000***

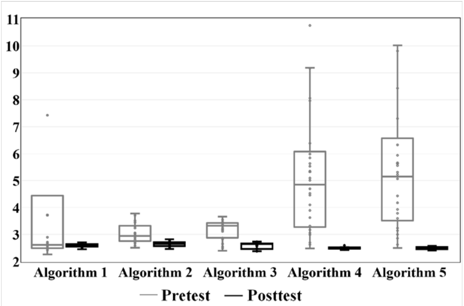

FIGURE 7. Box plot of Example 4.

Example 5. Pressure Vessel Design Problem

The pressure vessel design problem introduced by Sandgren [49] aims to minimize the total cost of materials when forming and welding pressure vessels. In Table IX and Figure 8, the proposed easy particles have improved the performances and stabilities of all referenced algorithms as well for Example 5.

Min $f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

s.t. $-x_1 + 0.0193x_3 \le 0$, $-x_2 + 0.00954x_3 \le 0$,

$-\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$,

$x_4 - 240 \le 0$, $0.0625 \le x_1, x_2 \le 6.1875$,

$10 \le x_3, x_4 \le 200$.

TABLE IX
EXPERIMENT RESULT OF EXAMPLE 5

		Algorithm 1	2	3	4	5
Pretest	Max	11946.73	11549.74	11523.64	23385.96	21633.41
	Mean	6591.82	6735.73	6974.48	11668.12	11063.45
	Min	6058.87	5977.99	5915.86	6355.25	5897.48
	Stdev	1291.65	1547.71	1569.72	5184.27	5265.78
	QD	**85.22**	130.86	332.29	4260.13	3738.06
Posttest	Max	6264.68	6538.04	7144.50	7950.01	6084.48
	Mean	**6027.54**	**6145.40**	**6219.87**	**6758.61**	**6011.27**
	Min	5895.68	5938.13	5895.68	6029.14	5892.29
	Stdev	**108.63**	**174.72**	**298.34**	**543.28**	**57.58**
	QD	86.01	**111.61**	**229.29**	**279.84**	**43.28**
p-value		0.0094**	0.0227*	0.0083**	0.0000***	0.0000***

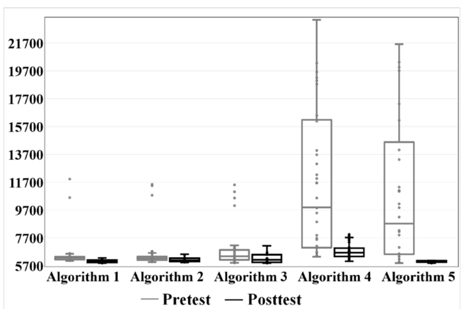

FIGURE 8. Box plot of Example 5.

Table X shows the significance level of the experiment results of 25 pretest–posttest experiments (five examples

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

with five referenced algorithms). All experiment results of posttest are better than the ones of pretest significantly.

TABLE X
SUMMARY OF THE EXPERIMENT RESULTS OF PRETEST-POSTTEST EXPERIMENTS

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5
Example 1	***	***	***	***	***
Example 2	***	***	**	**	**
Example 3	***	***	**	***	***
Example 4	**	***	***	***	***
Example 5	**	*	**	***	***

*: p-value<0.05; **: p-value<0.01; ***: p-value<0.001.

Table X demonstrates that the proposed easy particles improve all referenced algorithms in all examples. The experiment using 10% of swarm size to improve the referenced algorithms and 90% of swarm size keeps the features of the referenced algorithms. The experiment results demonstrate that the proposed easy particles are helpful to improve the referenced algorithms for obtaining better results.

V. CONCLUSION

This study proposes the easy particle inspired by the lazy ant's effect in the ant colony for PSO to solve the issue of constraints in NCO problems, and the easy particle is very convenient to embed the current PSO-based approaches. Based on lazy ant behavior, the proposed easy particle unrestricted by social and cognition factors can break through the containment of constraint for enhancing the probabilities of exploring undiscovered regions. The experiments demonstrate that the proposed easy particles embedded in all referenced algorithms can effectively reduce premature convergence for significantly improving the NCO problems' performances.

Reducing the number of easy particles will enhance the exploitation capability but weakening the exploration capability; on the contrary, increasing the number of easy particles will enhance the exploration capability but weakening the exploitation capability. Sometimes the PSO-based approach needs more easy particles to enhance the exploration capability, and sometimes it needs fewer easy particles to enhance the exploitation capability. How to determine the optimal number of easy particles for both maintaining the exploration and exploitation is always a dilemma problem for PSO.

Appropriately refreshing the number of easy particles seems a good strategy; therefore, the elastic size of easy particles is a topic for future research.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Dec. 1995, vol. 4, pp. 1942–1948 vol.4, doi: 10.1109/ICNN.1995.488968.
[2] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 2, pp. 397–407, Feb. 2004, doi: 10.1109/TAP.2004.823969.
[3] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013, doi: 10.1109/TII.2012.2198665.
[4] B. Gordan, D. J. Armaghani, M. Hajihassani, and M. Monjezi, "Prediction of seismic slope stability through combination of particle swarm optimization and neural network," *Eng. with Comput.*, vol. 32, no. 1, pp. 85–97, 2016, doi: 10.1007/s00366-015-0400-7.
[5] S. Upadhyay and M. P. Sharma, "Development of hybrid energy system with cycle charging strategy using particle swarm optimization for a remote area in India," *Renewable Energy*, vol. 77, pp. 586–598, May 2015, doi: 10.1016/j.renene.2014.12.051.
[6] R. Ge and J. Gao, "Improved PSO algorithm for energy saving research in the double layer management mode of the cloud platform," Jul. 2016, pp. 257–262, doi: 10.1109/ICCCBDA.2016.7529567.
[7] L. Zhang, W. Lu, X. Liu, W. Pedrycz, C. Zhong, and L. Wang, "A Global Clustering Approach Using Hybrid Optimization for Incomplete Data Based on Interval Reconstruction of Missing Value: GLOBAL CLUSTERING APPROACH FOR INCOMPLETE DATA," *Int. J. Intell. Syst.*, vol. 31, no. 4, pp. 297–313, Apr. 2016, doi: 10.1002/int.21752.
[8] M. S. Rahman, A. K. Manna, A. A. Shaikh, and A. K. Bhunia, "An application of interval differential equation on a production inventory model with interval-valued demand via center-radius optimization technique and particle swarm optimization," *Int J Intell Syst*, vol. 35, no. 8, pp. 1280–1326, Aug. 2020, doi: 10.1002/int.22254.
[9] P. H. Mohan. Performance Review of Harmony Search, Differential Evolution and Particle Swarm Optimization. *IOP Conference Series: Materials Science and Engineering*. 2017;225(1):012221. doi: 10.1088/1757-899X/225/1/012221
[10] B. Al-kazemi, C. K. Mohan. Discrete Multi-Phase Particle Swarm Optimization. *Information Processing with Evolutionary Algorithms.* 2005:305-327. doi: 10.1007/1-84628-117-2_20
[11] B. Wang, Z. Wu, Z. Zhao. Performance comparison of GA, PSO, and DE approaches in estimating low atmospheric refractivity profiles. *Wuhan Univ. J. Nat. Sci.* 2010;15(5):433-439. doi: 10.1007/s11859-010-0679-6
[12] A. Engelbrecht, "Particle swarm optimization: Velocity initialization," in *2012 IEEE Congress on Evolutionary Computation*, Brisbane, Australia, Jun. 2012, pp. 1–8, doi: 10.1109/CEC.2012.6256112.
[13] Cui Zhihua, Zeng Jianchao, and Cai Xingjuan, "A new stochastic particle swarm optimizer," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, Portland, OR, USA, 2004, pp. 316–319, doi: 10.1109/CEC.2004.1330873.
[14] J. An, Q. Kang, L. Wang, and Q. Wu, "Mussels Wandering Optimization: An Ecologically Inspired Algorithm for Global Optimization," *Cogn Comput*, vol. 5, no. 2, pp. 188–199, Jun. 2013, doi: 10.1007/s12559-012-9189-5.
[15] A. L. Cronin, "Individual and Group Personalities Characterise Consensus Decision-Making in an Ant," *Ethology*, vol. 121, no. 7, pp. 703–713, 2015, doi: 10.1111/eth.12386.
[16] M. W. Moffett, The human swarm: how our societies arise, thrive, and fall. New York: Basic Books, 2019.
[17] I. Maák, G. Roelandt, and P. d'Ettorre, "A small number of workers with specific personality traits perform tool use in ants," *eLife*, vol. 9, p. e61298, Dec. 2020, doi: 10.7554/eLife.61298.
[18] T. O. Richardson, A. Coti, N. Stroeymeyt, and L. Keller, "Leadership – not followership – determines performance in ant teams," *Communications Biology*, vol. 4, no. 1, p. 535, May 2021, doi: 10.1038/s42003-021-02048-7.
[19] M. W. Moffett et al., "Ant colonies: building complex organizations with minuscule brains and no leaders," *Journal of Organization Design*, vol. 10, no. 1, pp. 55–74, Mar. 2021, doi: 10.1007/s41469-021-00093-4.

[20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," May 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.

[21] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. Part I: background and development," *Natural Computing*, vol. 6, no. 4, pp. 467–484, Dec. 2007, doi: 10.1007/s11047-007-9049-5.

[22] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Natural Computing*, vol. 7, no. 1, pp. 109–124, Mar. 2008, doi: 10.1007/s11047-007-9050-z.

[23] C. A. C. Coello, "A Survey of Constraint Handling Techniques used with Evolutionary Algorithms," *Techn. Rep. Lania–RI–99–04*, p. 33, 1999.

[24] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evol. Comput.*, vol. 7, no. 1, pp. 19–44, 1999, doi: 10.1162/evco.1999.7.1.19.

[25] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, Mar. 2000, doi: 10.1016/S0166-3615(99)00046-9.

[26] Ö. Yeniay, "Penalty Function Methods for Constrained Optimization with Genetic Algorithms," *MCA*, vol. 10, no. 1, pp. 45–56, Apr. 2005, doi: 10.3390/mca10010045.

[27] D. S. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "On Solving Multiobjective Bin Packing Problems Using Particle Swarm Optimization," in *2006 IEEE International Conference on Evolutionary Computation*, Vancouver, BC, Canada, 2006, pp. 2095–2102, doi: 10.1109/CEC.2006.1688565.

[28] M. Sanaz, M. Sanaz, H. Werner, and W. Anja, "Linear Multi-Objective Particle Swarm Optimization," in *Stigmergic Optimization*, vol. 31, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 209–238.

[29] Li-Yeh Chuang, Sheng-Wei Tsai, and Cheng-Hong Yang, "Catfish particle swarm optimization," in *2008 IEEE Swarm Intelligence Symposium*, St. Louis, MO, USA, Sep. 2008, pp. 1–5, doi: 10.1109/SIS.2008.4668277.

[30] S. He, E. Prempain, and Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems," *Engineering Optimization*, vol. 36, no. 5, pp. 585–605, Oct. 2004, doi: 10.1080/03052150410001704854.

[31] S. L. Ho, S. Y. Yang, G. Z. Ni, and K. F. Wong, "An Improved PSO Method With Application to Multimodal Functions of Inverse Problems," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1597–1600, Apr. 2007, doi: 10.1109/TMAG.2006.892108.

[32] K. E. Parsopoulos, "UPSO : A Unified Particle Swarm Optimization Scheme," *Lecture Series on Computer and Computational Science*, vol. 1, pp. 868–873, 2004, [Online]. Available: https://ci.nii.ac.jp/naid/10018505420/en/.

[33] K. E. Parsopoulos and M. N. Vrahatis, "Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems," 2005, pp. 582–591, doi: 10.1007/11539902_71.

[34] Z. Wen-Jun and X. Xiao-Feng, "DEPSO: hybrid particle swarm with differential evolution operator," Oct. 2003, vol. 4, pp. 3816–3821 vol.4, doi: 10.1109/ICSMC.2003.1244483.

[35] M. Kohler, L. Forero, M. Vellasco, R. Tanscheit, and M. A. Pacheco, "PSO+: A nonlinear constraints-handling particle swarm optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2016, pp. 2518–2523, doi: 10.1109/CEC.2016.7744102.

[36] H. Mohan Pandey, "Performance Review of Harmony Search, Differential Evolution and Particle Swarm Optimization," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 225, p. 012221, Aug. 2017, doi: 10.1088/1757-899X/225/1/012221.

[37] N. K. Jain, U. Nangia, and J. Jain, "A Review of Particle Swarm Optimization," *J. Inst. Eng. India Ser. B*, vol. 99, no. 4, pp. 407–411, Aug. 2018, doi: 10.1007/s40031-018-0323-y.

[38] D. M. Gordon, B. C. Goodwin, and L. E. H. Trainor, "A parallel distributed model of the behaviour of ant colonies," *Journal of Theoretical Biology*, vol. 156, no. 3, pp. 293–307, Jun. 1992, doi: 10.1016/S0022-5193(05)80677-0.

[39] D. M. Gordon and N. J. Mehdiabadi, "Encounter rate and task allocation in harvester ants," *Behavioral Ecology and Sociobiology*, vol. 45, no. 5, pp. 370–377, Apr. 1999, doi: 10.1007/s002650050573.

[40] D. Charbonneau, N. Hillis, and A. Dornhaus, "'Lazy' in nature: ant colony time budgets show high 'inactivity' in the field as well as in the lab," *Insect. Soc.*, vol. 62, no. 1, pp. 31–35, Feb. 2015, doi: 10.1007/s00040-014-0370-6.

[41] D. Charbonneau and A. Dornhaus, "When doing nothing is something. How task allocation strategies compromise between flexibility, efficiency, and inactive agents," *J Bioecon*, vol. 17, no. 3, pp. 217–242, Oct. 2015, doi: 10.1007/s10818-015-9205-4.

[42] N. Imirzian, Y. Zhang, C. Kurze, R. G. Loreto, D. Z. Chen, and D. P. Hughes, "Automated tracking and analysis of ant trajectories shows variation in forager exploration," *Sci. Rep.*, vol. 9, no. 1, p. 13246, Dec. 2019, doi: 10.1038/s41598-019-49655-3.

[43] M. Lin, Z. Wang, and F. Wang, "Hybrid Differential Evolution and Particle Swarm Optimization Algorithm Based on Random Inertia Weight," in *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Jun. 2019, pp. 411–414, doi: 10.1109/YAC.2019.8787698.

[44] M. Bonyadi and Z. Michalewicz, "Impacts of coefficients on movement patterns in the particle swarm optimization algorithm," *IEEE Trans. Evol. Computat.*, pp. 1–1, 2016, doi: 10.1109/TEVC.2016.2605668.

[45] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960, doi: 10.1093/comjnl/3.3.175.

[46] L. C. W. Dixon and G. P. Szegö, *Towards Global Optimisation: Proceedings of a Workshop*. North-Holland; New York: American Elsevier, 1975.

[47] Alex Townsend, "Constrained optimization in Chebfun," *Chebfun*, Jan. 2014. http://www.chebfun.org/examples/opt/ConstrainedOptimization.html.

[48] S. S. Rao, *Engineering Optimization*. JOHN WILEY & SONS, INC., 1996.

[49] E. Sandgren, "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990, doi: 10.1115/1.2912596.