اونيۏرسيتي مليسيا ڤهڠ
UNIVERSITI MALAYSIA PAHANG

ORIGINAL ARTICLE

Modified Particle Swarm Optimization with Chaotic Initialization Scheme for Unconstrained Optimization Problems

K.M. Ang[1], Z.S. Yeap[1], C.E. Chow[1], W. Cheng[1], S.S. Tiang[1] and W.H. Lim[1*]

[1]Faculty of Engineering, Technology and Built Environment, UCSI University, 56000 Kuala Lumpur, Malaysia.

ABSTRACT – Different variants of particle swarm optimization (PSO) algorithms were introduced in recent years with various improvements to tackle different types of optimization problems more robustly. However, the conventional initialization scheme tends to generate an initial population with relatively inferior solution due to the random guess mechanism. In this paper, a PSO variant known as modified PSO with chaotic initialization scheme is introduced to solve unconstrained global optimization problems more effectively, by generating a more promising initial population. Experimental studies are conducted to assess and compare the optimization performance of the proposed algorithm with four existing well-establised PSO variants using seven test functions. The proposed algorithm is observed to outperform its competitors in solving the selected test problems.

INTRODUCTION

Majority of the real-world applications contain a set of parameters considered as decision variables that need to be optimized. The objective function is formulated to represent a distinct target to be accomplished by an optimization problem. In particular, the optimal combinations of decision variables can be either the lowest and highest objective function values for minimization and maximization problems, respectively. In the past decades, metaheuristic search algorithm (MSA) emerged as a promising decision-making tool to determine the optimal set of decision variables for various type of optimization problems, due to its characteristic of independent from the gradient information. MSAs can be grouped into two types based on their inspration sources, known as swarm intelligence (SI) algorithm and evolutionary algorithm (EA). SI algorithms, including artificial bee colony (ABC) [1] and bat algorithm (BA) [2], are motivated by the cooperative behavior of animals, while the EAs, such as genetic algorithm (GA) [3] and differential evolution (DE) [4], are inspired by the Darwin's Theory of Evolution. MSAs are widely implemented to solve various type of optimization problems with different complexity level [5-19], due to its strengths of fast convergence speed and promising global search ability.

Particle swarm optimization (PSO) algorithm emerges as one of the most popular MSAs due to its significant strengths such as simplicity implementation, high convergence rate, etc. During the searching process, the search trajectory of PSO algorithm are dependent on the collaborative behavior of the swarm members in sharing the useful information to locate the superior region of the search space. However, the conventional population initialization scheme adopted by the conventional PSO tends to generate an initial population with relatively inferior particles due to random guess mechanism [20]. Furthermore, the poor balancing of the explorative and exploitative search behavior of the conventional PSO tends to suffer with premature convergence in dealing with complex optimization problems. Extensive studies were performed in recent years to strengthen the performance of PSO in solving single-objective optimization problems (SOPs) [6, 8], multi-objectives optimization problems (MOPs) [7], constrained optimization problems (COPs) [5], etc.

Although signficant numbers of PSO variants were introduced in the past decades such as those reported in [5, 8, 17, 18, 21-27], the high tendency of these PSO variants to suffer with premature convergence issue remains as an on-going challenge. There are several factors that can contribute to these undesirable issue such as the initial population with poor quality and strong dependence of particles on histrical best positions (e.g., personal best position, global best position and etc.) [28]. The initial populations of most existing PSO variants are randomly generated using uniform distributions without considering current condition of search environment. These particles might be occasionally initialized nearby local optima or solution region far away from global optimum, hence leading to premature convergence and slow convergence speed of algorithm, respectively. Conventional PSO are also found to strongly rely on the guidance of their historical best solutions such as personal best position and global best position during the search process. However, it is notable that these solutions might not be frequently updated in the middle or later stages of optimization process, therefore have high tendency of diversity loss in population. In order to address the aforementioned issues, it is necessary to design a modified initizlization scheme that is able to produce initial population with better quality at the beginning of

optimization process. In addition, a more robust scheme is also designed to generate new examplars that can guide particles more effectively with better solution diversity by levering the useful search information stored in other non-fittest particles. To this end, a modified particle swarm optimization with chaotic initialization scheme (MPSO-CIS) is introduced. The main contributions of MPSO-CIS are presented as follows:

1. A chaotic initialization scheme is designed by hybridizing the chaotic system and oppositional based learning strategies to initialize the population with improved qualities.
2. A unique exemplar derivation module (UEDM) is implemented to produce a distinctive model to substitute the self-cognitive component of each particle in searching for the promising region of the search space.
3. Performance assessments of the proposed MPSO-CIS algorithm in solving seven global optimization test problems and compared with four well-established PSO variants.

The remaining parts of this article are presented as follows. Section 2 summarizes the conventional PSO and the opposition-based learning strategy. The proposed MPSO-CIS frameworks are presented in Section 3. The performance assessments and comparisons are reported in Section 4. At last, the conclusion and future studies are summarized in Section 5.

RELATED WORK

Conventional Particle Swarm Optimization

Inspiration from the group behavior of bird flocks in locating the food sources, conventional particle swarm optimization was introduced as a powerful optimization algorithm in solving SOPs in year 1995 [29]. The search mechanisms of the conventional PSO allow each particle to search based on their best self-knowledge and the best knowledge of the whole swarm during the optimization process. Given that each particle n of the population refers to a candidate solution of a specified optimization problem, $V_n = [V_{n,1},...,V_{n,d},...,V_{n,D}]$ and $X_n = [X_{n,1},...,X_{n,d},...,X_{n,D}]$ define the velocity and position vectors of the particle n, respectively. Note that $d \in [1,D]$ indicates the dimensional index with the total dimension of D, and $n \in [1,N]$ refers to the particle index with the population size of N. Given the personal best position $P_{best,n} = [P_{best,n,1},...,P_{best,n,d},...,P_{best,n,D}]$ of particle n and the global best position $G_{best} = [G_{best,1},...,G_{best,d},...,G_{best,D}]$ of the entire poulation found during the optimizaion process, the new velocity $V_{n,d}^{New}$ and new position $X_{n,d}^{new}$ of each particle n in dimension d are calculated as follows:

$$V_{n,d}^{new} = \omega V_{n,d} + c_1 r_1 (P_{best,n,d} - X_{n,d}) + c_2 r_2 (G_{best,d} - X_{n,d}) \tag{1}$$

$$X_{n,d}^{new} = X_{n,d} + V_{n,d}^{new} \tag{2}$$

where ω refers to the inertia weight; c_1 and c_2 represent the acceleration coefficients; r_1 and r_2 denote two numbers stochastically produced from uniform distribution in range of $[0,1]$. The fitness value of the new position of particle n is assessed and its personal best position as well as the global best position are replaced if the new position of particle n has fitter value. The optimization process of the conventional PSO is replicated until the termination criteria is fulfilled. The global best position G_{best} is treated as the optimal solution to the specified problem.

Opposition-Based Learning (OBL)

The opposition-based learning (OBL) concept [30] is applied to produce a set of opposite solutions that potentially contributed to the searching process. Given the intention is to seek for solution x and the searching of solution x in opposite direction is agreed, the opposite x is calculated as follows:

$$x = a + b - x \tag{3}$$

where x refers to a real number in range of a and b. Besides that, given that D as the total number of dimensional components, the opposite number $x = (x_1,...,x_d,...,x_D)$ in a multidimensional case P are calculated as follows:

$$x_d = a_d + b_d - x_d, d = 1,...,D \tag{4}$$

Based on the potential solution $P = (x_1,...,x_d,...,x_D)$ with D dimensional size, the potential solution is replaced by its opposite solution if the latter solution is fitter than the former solution. Otherwise, the potential solution remains as its own value. During the searching process, the search interval is recursively decreased until either the potential solution or the opposite slution is close to an existing solution for the given problem.

Metaheuristic Search Algorithms with Opposition-Based Learning Scheme

In the past decades, OBL scheme was widely modified and implemented into MSAs to enhance the performance of the optimization algorithm in dealing with optimization problems with different complexity level. In [31], a probabilistic opposition-based learning (OBL) scheme was designed into PSO to tackle noisy optimization problems effectively. A number of best particles were picked from the current swarm and the opposite swarm generated by the adopted OBL scheme, in order to preserve the swarm diversity in noisy environments. In [32], OBL scheme was implemented to an improved sine cosine algorithm (SCA) to deal with SOPs more effectively. The influence of the OBL scheme adopted in SCA algorithm was investigated and reported to improve the search accuracy, convergence rate and time complexity. In [33], an enhanced PSO algorithm known as GOPSO was proposed with the incorporation of generalized OBL (GOBL) and Cauchy mutation. The GOBL sampling scheme was adopted to convert the potential solutions from current population into another search space, in order to increase thr probability in locating a number of better solutions. In [34], OBL scheme was designed into PSO algorithm to decrease the tendency of particles being stucked in local optima, and to boost the convergence speed of PSO algorithm. In [35], a gravitational search algorithm (GSA) variant is introduced with the OBL sheme to deal with combined economic and emission dispatch problems of power systems. The OBL scheme was employed to generate initial swarm with superior diversity level and to enable the generation jumping, leading to the improvement of convergence rate.

MODIFIED PARTICLE SWARM OPTIMIZATION WITH CHAOTIC INITIALIZATION SCHEME

Chaotic Initialization Scheme

A chaotic oppositional based initialization scheme (COBIS) is adopted in this work to replace the conventional population initialization scheme of PSO. At the initial stage of the optimization process, the population of MPSO-CIS is initialized with improved diversity, aiming to prevent premature convergence by leveraging the stochasticity and non-repetition characteristics of chaotic maps. Given that ζ_0 and ζ_z are the preliminary condition and z-th sequence of chaotic variable in range of $[0,1]$, respectively, the chaotic sequence is generated by using a modified sine map as expressed [36]:

$$\zeta_{z+1} = \sin(\rho \zeta_z), z = 1,...,Z \tag{5}$$

where $\rho = \pi$ is a bifurcation coeffcient of modified sine map; Z refers to the maximum chaotic sequences. Denote $X_d^{\min}$ and $X_d^{\max}$ as the lowest and the highest values of design variables, respectively, where $d = 1,...,D$. Given that ζ_Z represents the final sequence of chaotic variable produced, the dimension d of each chaotic swarm member n denoted as $X_{n,d}^{CS}$, is computed as follows:

$$X_{n,d}^{CS} = X_d^{\min} + \zeta_Z \left(X_d^{\max} - X_d^{\min} \right) \tag{6}$$

where $n = 1,...,N$. The dimension d of each opposite swarm member n denoted as $X_{n,d}^{OBL}$ is computed using oppositional based learning strategy as follows:

$$X_{n,d}^{OBL} = X_d^{\min} + X_d^{\max} - X_{n,d}^{CS} \tag{7}$$

A new population set $\varphi^{Comb} = \varphi^{CS} \cup \varphi^{OBL}$ with population size of $2N$ is formed by combining both φ^{CS} and φ^{OBL}, where $\varphi^{CS} = \left[X_1^{CS},...,X_n^{CS},...,X_N^{CS} \right]$ and $\varphi^{OBL} = \left[X_1^{OBL},...,X_n^{OBL},...,X_N^{OBL} \right]$ refer to the population sets of chaotic and opposite swarm members, respectively. After arranging all solution members in φ^{Comb} from best to worst based on their objective function values, the top N members of φ^{Comb} with relatively superior objective function values are selected to form an initial population of MPSO-CIS denoted as $\varphi^{Initial} = \left[X_1,...,X_n,...,X_N \right]$. The chaotic initialization scheme of MPSO-CIS is described in Figure 1.

Unique Exemplar Derivation Module

In this study, the unique exemplar derivation module (UEDM) is adopted to produce the unique exemplar of each particle n to lead its search process. At the beginning stage, two exemplar candidates of particle n are calculated. Given that $P_{best,n1}$, $P_{best,n2}$ and $P_{best,n3}$ are the personal best positions of three different stochastically chosen particles with indices $n1$, $n2$ and $n3$, respectively, the first exemplar candidates $X_n^{\exp,1}$ are obtained as follows:

$$X_n^{\exp,1} = \begin{cases} r_1 P_{best,n1} + r_2 \left(P_{best,n2} - P_{best,n3} \right), f\left(P_{best,n2} \right) < f\left(P_{best,n3} \right) \\ r_1 P_{best,n1} + r_2 \left(P_{best,n3} - P_{best,n2} \right), Otherwise \end{cases} \tag{8}$$

where r_1 and r_2 refer to two stochastic numbers produced from uniform distribution in range of $[0,1]$ and $r_1 + r_2 = 1$; $f(\bullet)$ represent the fitness value of the particle. Based on Equation 8, neighborhood search is conducted around a random particle $n1$ with $P_{best,n1}$ that is located far from the global best position G_{best}. Hence, $X_n^{\exp,1}$ tends to be explorative to avoid the stagnation of particle n by leading it to explore other regions of the search space.

Algorithm 1: Chaotic Initialization Scheme

Input: D, N, $X_d^{\min}$, $X_d^{\max}$, Z, ρ

01:     Initialize $\varphi^{CS} \leftarrow \varnothing$ and $\varphi^{OBL} \leftarrow \varnothing$

02:     **for** each particle n **do**
03:             **for** each dimension d **do**
04:                     Initialize $\zeta_0 \in [0,1]$ and $z = 1$;
05:                     **while** $z \leq Z$ **do**
06:                             Calculate ζ_{z+1} using Eq. (5), $z \leftarrow z+1$;
07:                     **end while**
08:                     Produce $X_{n,d}^{CS}$ and $X_{n,d}^{OBL}$ using Eqs. (6) and (7), respectively;
09:             **end for**
10:             $\varphi^{CS} \leftarrow \varphi^{CS} \cup X_n^{CS}$, $\varphi^{OBL} \leftarrow \varphi^{OBL} \cup X_n^{OBL}$;
11:     **end for**
12:     $\varphi^{Comb} = \varphi^{CS} \cup \varphi^{OBL}$;
13:     Evaluate objective function values of all φ^{Comb} members and sort from best to worst;
14:     Select the first N members of φ^{Comb} to form initial population;

Output: $\varphi^{Initial} = [X_1,...,X_n,...,X_N]$ and the associated objective function values

Figure 1. Pseudocode of initialization scheme of MPSO-CIS

Given that $P_{best,n4}$ and $P_{best,n5}$ are the personal best positions of two different stochastically chosen particles with indices $n4$ and $n5$, respectively, the second exemplar candidates $X_n^{\exp,2}$ is computed as follows:

$$X_n^{\exp,2} = \begin{cases} r_3 G_{best} + r_4 \left(P_{best,n4} - P_{best,n5} \right), f\left(P_{best,n4} \right) < f\left(P_{best,n5} \right) \\ r_1 G_{best} + r_2 \left(P_{best,n5} - P_{best,n4} \right), Otherwise \end{cases} \tag{9}$$

where r_3 and r_4 refer to two stochastic numbers produced from uniform distribution in range of $[0,1]$ and $r_3 + r_4 = 1$. In contrast to $X_n^{\exp,1}$, $X_n^{\exp,2}$ is produced by performing neighborhood search around G_{best} that is relatively nearer to the global optimum. Therefore, the latter exemplar tends to be more exploitative in searching around the superior regions of the search space.

Given that both of the $X_n^{\exp,1}$ and $X_n^{\exp,2}$ produced randomly, two scenarios are stated to compare the fitness values of particle n, $X_n^{\exp,1}$ and $X_n^{\exp,2}$, as follows:

1.   If the best exemplar candidate selected between $X_n^{\exp,1}$ and $X_n^{\exp,2}$ are better than particle n, the better exemplar candidate is assigned as the exemplar of particle n indicated as ε_n.

2.   If the best exemplar candidate selected between $X_n^{\exp,1}$ and $X_n^{\exp,2}$ are worse than particle n, crossover mechanism is performed to produce the third exemplar $X_n^{\exp,3}$ by leveraging the information of $X_n^{\exp,1}$ and $X_n^{\exp,2}$. The weightage value $W_{n,k}$ indicates the tendency of each exemplar candidate k to contribute its information in generating each dimensional component of $X_n^{\exp,3}$ is computed as follows:

$$W_{n,k} = \begin{cases} 1 / \left[1 + f\left(X_n^{\exp,k} \right) \right], f\left(X_n^{\exp,k} \right) \geq 0 \\ 1 + \left| f\left(X_n^{\exp,k} \right) \right|, Otherwise \end{cases} \tag{10}$$

where $k = 1$ and 2. Given the $W_{n,k}$ values, roulette wheel selection is performed to choose the exemplar candidate that contributes to each dimensional component of $X_n^{\exp,3}$. The exemplar candidate with fitter objective function values has higher probability to contribute to compute $X_n^{\exp,3}$, and vice versa. In order to prevent the domination of the superior exemplar candidate in contribute to the formation of $X_n^{\exp,3}$, a stochastic dimensional index d_r is generated and the dimensional component with index of d_r is contributed by the relatively worse exemplar candidate. The best exemplar candidate among $X_n^{\exp,1}$, $X_n^{\exp,2}$ and $X_n^{\exp,3}$ is allocated as the exemplar of particle n indicated as ε_n.

Given the fitness evaluation counter *fes* and the personal best positions of all particles $P_{best} = \left[P_{best,1}, ..., P_{best,n}, ..., P_{best,N} \right]$ in MPSO-CIS, the crossover mechanism and the unique exemplar derivation module are described in Figures 1 and 2, respectively.
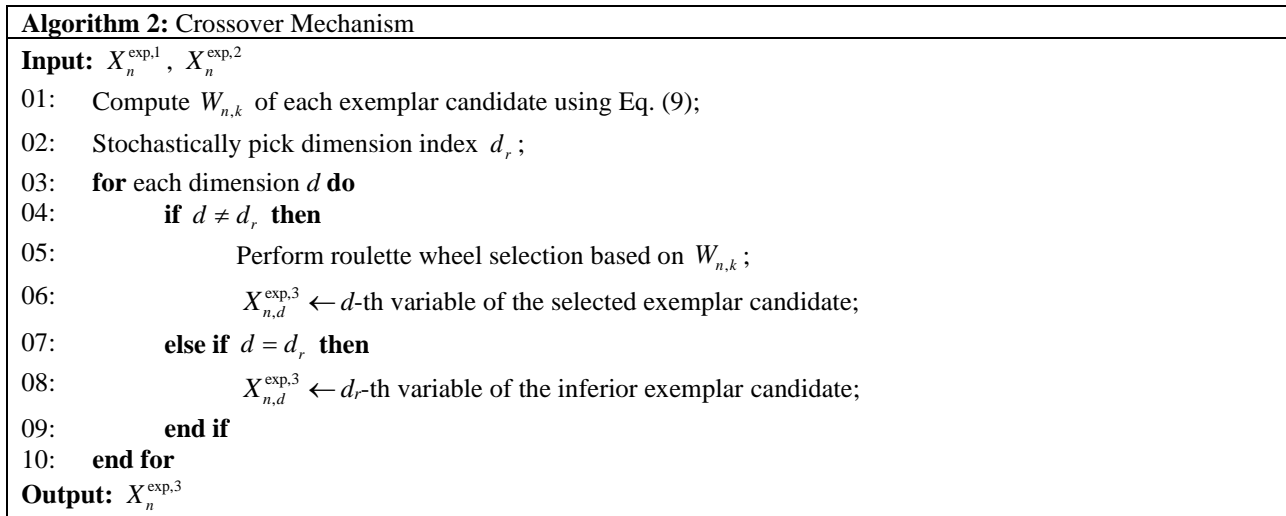

Algorithm 2: Crossover Mechanism

Input: $X_n^{\exp,1}$, $X_n^{\exp,2}$

01:    Compute $W_{n,k}$ of each exemplar candidate using Eq. (9);

02:    Stochastically pick dimension index d_r;

03:    **for** each dimension d **do**

04:        **if** $d \neq d_r$ **then**

05:            Perform roulette wheel selection based on $W_{n,k}$;

06:            $X_{n,d}^{\exp,3} \leftarrow d$-th variable of the selected exemplar candidate;

07:        **else if** $d = d_r$ **then**

08:            $X_{n,d}^{\exp,3} \leftarrow d_r$-th variable of the inferior exemplar candidate;

09:        **end if**

10:    **end for**

Output: $X_n^{\exp,3}$

Figure 2. Pseudocode of the crossover mechanism adopted in MPSO-CIS.

Algorithm 3: Unique Exemplar Derivation Module

Input: Particle n, P_{best}, G_{best}

01:    Produce $X_n^{\exp,1}$ and $X_n^{\exp,2}$ using Eqs. (8) and (9), respectively;

02:    Evaluate $f\left(X_n^{\exp,1}\right)$ and $f\left(X_n^{\exp,2}\right)$;

03:    $fes \leftarrow fes + 2$;

04:    **if** $\min\left[f\left(X_n^{\exp,1}\right), f\left(X_n^{\exp,2}\right) \right] < f\left(P_{best,n}\right)$ **then**

05:        $\varepsilon_n \leftarrow$ exemplar candidate $X_n^{\exp,1}$ or $X_n^{\exp,2}$ with the best objective function value;

06:    **else if** $\min\left[f\left(X_n^{\exp,1}\right), f\left(X_n^{\exp,2}\right) \right] \geq f\left(P_{best,n}\right)$ **then**

07:        Generate $X_{n,d}^{\exp,3}$ using crossover mechanism; */*Algorithm 2*/*

08:        Evaluate $f\left(X_n^{\exp,3}\right)$;

09:        $fes \leftarrow fes + 1$;

10:        $\varepsilon_n \leftarrow$ exemplar candidate $X_n^{\exp,1}$, $X_n^{\exp,2}$ or $X_{n,d}^{\exp,3}$ with the best objective function value;

11:    **end if**
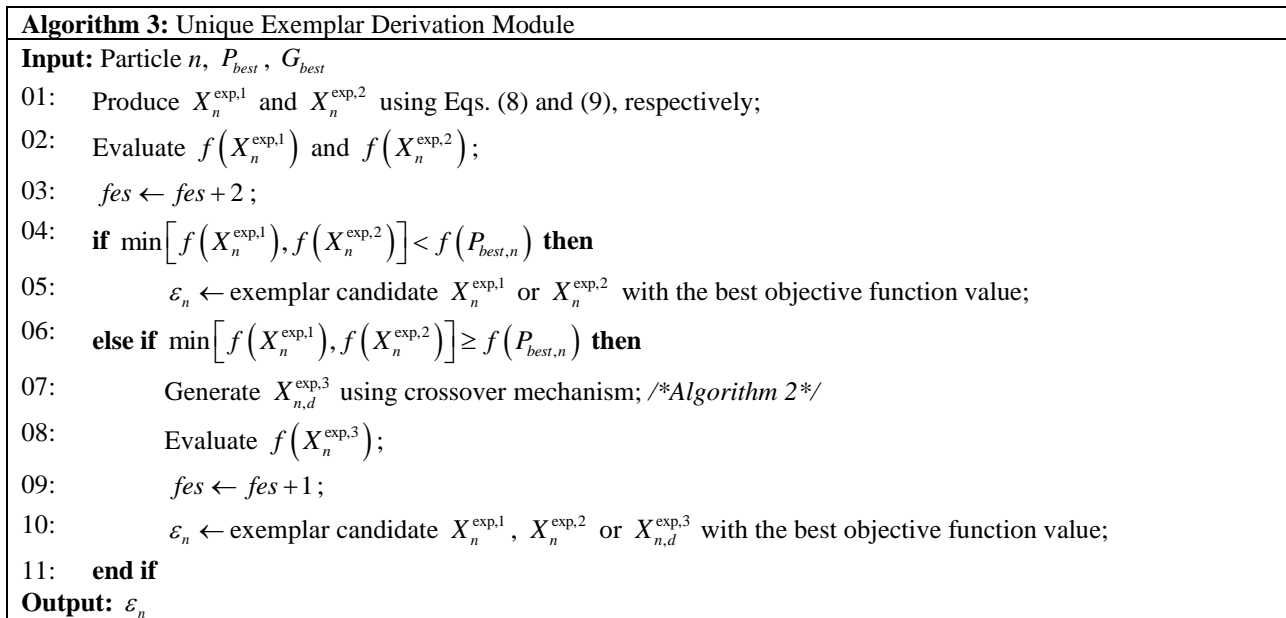
Output: ε_n

Figure 3. Pseudocode of the unique exemplar derivation module adopted in MPSO-CIS.

Overall Framwork of MPSO-CIS

The proposed MPSO-CIS calculates the new velocity of each particle n based on the exemplar ε_n formulated by the unique exemplar derivation module and the global best particle G_{best}. The d-th dimensional component of the new velocity and new position of each particle n are calculated as follows, repetively:

$$V_{n,d}^{new} = \omega V_{n,d} + c_1 r_5 \left(\varepsilon_{n,d} - X_{n,d} \right) + c_2 r_6 \left(G_{best,d} - X_{n,d} \right) \tag{11}$$

$$X_{n,d}^{new} = X_{n,d} + V_{n,d}^{new} \tag{12}$$

where ω represents the inertia weight; $n = 1,...,N$ refers to the particle index; c_1 and c_2 represent the acceleration coefficients; r_5 and r_6 refer to two stochastic numbers produced from uniform distribution in range of $[0,1]$.

The overall frame work of MPSO-CIS is described in Figure 4. In the early stage of the algorithm, the population is initialized via the chaotic initialization scheme. The unique exemplar derivation module is them performed to generate an exemplar ε_n to contribute in calculating the new velocity and position vectors of each particle n. The personal best position of particle n and the global best particle are replaced if a better particle is found. The reformulation of ε_n is performed only if the particle n fails to update its personal best position for S successive times as recorded by its counter variable s. The optimization process is iterated until the fitness evaluation counter fes reach the maximum fitness evaluation number η_{max}. The global best position G_{best} produced at the end of searching process is considered as the optimal solution for a specified problem.
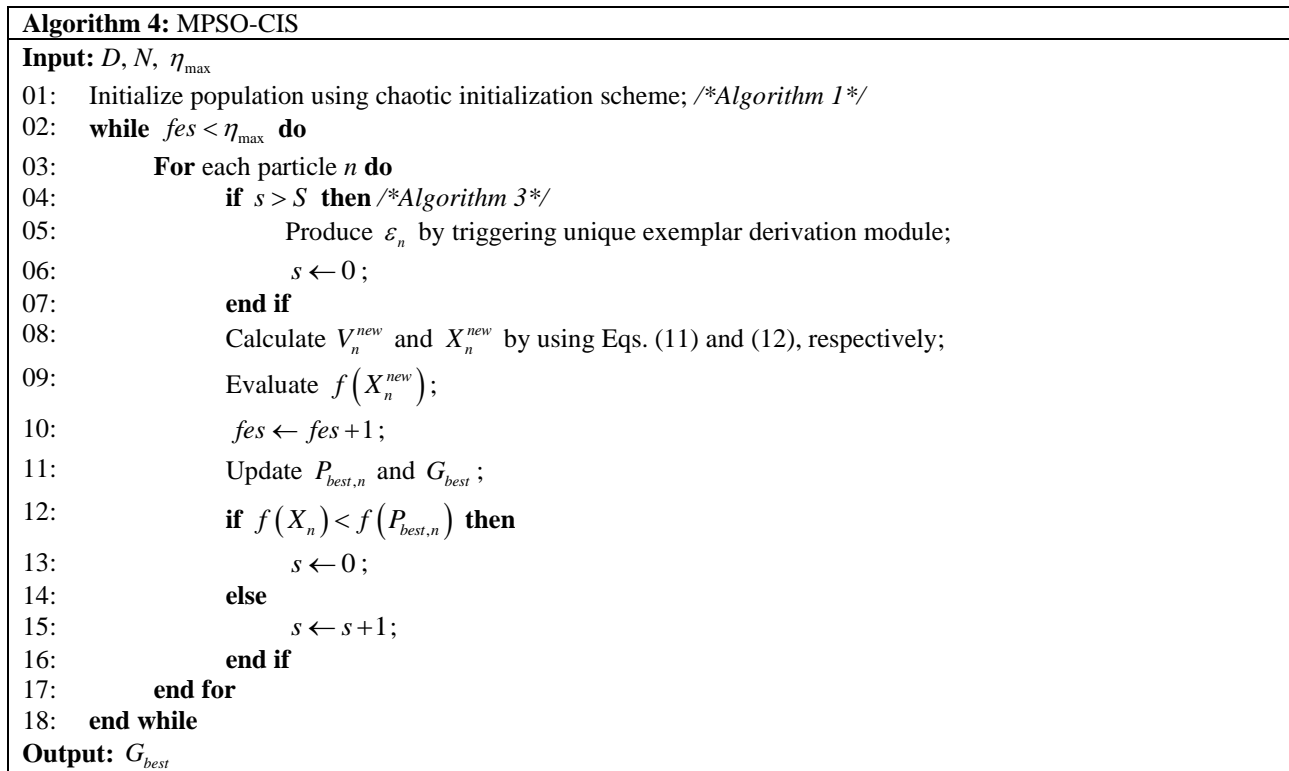

Algorithm 4: MPSO-CIS

Input: D, N, η_{max}

01:   Initialize population using chaotic initialization scheme; */Algorithm 1*/
02:   **while** $fes < \eta_{max}$ **do**
03:       **For** each particle n **do**
04:           **if** $s > S$ **then** */Algorithm 3*/
05:               Produce ε_n by triggering unique exemplar derivation module;
06:               $s \leftarrow 0$;
07:           **end if**
08:           Calculate V_n^{new} and X_n^{new} by using Eqs. (11) and (12), respectively;
09:           Evaluate $f\left(X_n^{new} \right)$;
10:           $fes \leftarrow fes + 1$;
11:           Update $P_{best,n}$ and G_{best};
12:           **if** $f\left(X_n \right) < f\left(P_{best,n} \right)$ **then**
13:               $s \leftarrow 0$;
14:           **else**
15:               $s \leftarrow s + 1$;
16:           **end if**
17:       **end for**
18:   **end while**

Output: G_{best}

Figure 4. Overall framework of MPSO-CIS.

SIMULATION ANALYSIS OF MPSO-CIS

Simulation Settings

Performance assessment of the proposed MPSO-CIS is conducted using seven global optimization test problems as described in Table 1, where LU refers to the lower and upper boundaries of the search region, and F_{min} represents the fitness value in global optimum. The optimization performance of MPSO-CIS is compared with four well-stablished pso variants known as: adaptive particle swarm optimization (APSO) [24], Frankenstein's particle swarm optimization (FPSO) [25], comprehensive learning particle swarm optimization (CLPSO) [26] and feedback learning particle swarm optimization with quadratic inertia weight (FLPSO-QIW) [27]. In particular, APSO adopted parameter adaptation strategy to improve the adaptivity of the algorithm in solving various types of problems. FPSO adopted the modified population topology strategy to enable information sharing between the particles. It is notably that the strategies adopted by CLPSO and FLPSO-QIW have similarities with MPSO-CIS which also compute exemplars from non-best solutions using different learning strategies.

In this study, the inertia weight ω of MPSO-CIS is initially set as 0.9 and recursively reduced to 0.4 at the end of the optimziation process. Both of the acceleration coefficients c_1 and c_2 are set as 2.0. The threshold value of S is set as 8 proved by experimental study in offering satisfactory performance for MPSO-CIS. The maximum fitness evaluation number η_{max}, population size N and dimensional size are set as 300000, 30, and 50, respectively, for all compared

algorithms. The simulations are conducted using MATLAB 2021b on a desktop computer utilized with Intel® Core™ i9-10900K CPU @ 3.70GHz for 25 independent runs in solving each problem.

Table 1. Global optimization benchmark functions.

Func.	Name	LU	F_{min}
A1	Sphere	$[-100,100]^D$	0
A2	Schwefel 1.2	$[-100,100]^D$	0
A3	Rastrigin	$[-5.12,5.12]^D$	0
A4	Noncontinuous Rastrigin	$[-5.12,5.12]^D$	0
A5	Griewank	$[-600,600]^D$	0
A6	Ackley	$[-32,32]^D$	0
A7	Weierstrass	$[-0.5,0.5]^D$	0

Performance Analysis of MPSO-CIS

The search accuracy and consistency of each algorithm in solving all benchmark problems are indicated by the mean fitness F_{mean} and standard deviation SD values and presented in Table 2. In particular, the lowest and the second-lowest F_{mean} values produced for each problem are indicated in boldface and underlined, respectively. The performance comparison of the compared algorithms are summarized as $w/t/l$ and $\#BMF$. Specifically, the proposed MPSO-CIS outperforms its competitor in w function, similar performance with its competitor in t function, underperform its competitor in l function, and $\#BMF$ indicates the number of best mean fitness value produced by the algorithm. Furthermore, Friedman test [37] is also performed as multiple comparison between the algorithms in solving the seven test problems and the average ranking of each algorithm is reported in Table 3.

Table 2. Performance comparison of MPSO-CIS and four PSO variants.

Func.	Criteria	APSO	FPSO	CLPSO	FLPSO-QIW	MPSO-CIS
A1	F_{mean}	2.50E-01	7.03E+01	3.30E-48	<u>2.89E-81</u>	**4.90E-162**
	SD	1.84E-01	6.99E+01	1.27E-47	5.96E-81	2.03E-161
A2	F_{mean}	1.49E+03	3.45E+03	5.14E+03	2.61E+02	**1.21E-03**
	SD	4.79E+02	1.34E+03	1.01E+03	8.89E+01	3.03E-03
A3	F_{mean}	**5.75E-01**	1.83E+01	9.09E+01	<u>2.59E+00</u>	4.81E+01
	SD	6.26E-01	1.00E+01	1.07E+01	1.51E+00	1.48E+01
A4	F_{mean}	**3.62E-02**	<u>1.61E+01</u>	8.12E+01	5.57E+00	4.90E+01
	SD	3.24E-02	9.57E+00	9.77E+00	2.35E+00	1.54E+01
A5	F_{mean}	1.67E-01	1.88E+00	<u>3.38E-11</u>	5.74E-04	**0.00E+00**
	SD	8.20E-02	9.30E-01	1.71E-10	2.20E-03	0.00E+00
A6	F_{mean}	6.63E-02	1.81E+00	<u>1.16E-14</u>	3.42E-14	**5.86E-15**
	SD	2.60E-02	1.11E+00	2.57E-15	1.06E-14	1.79E-15
A7	F_{mean}	5.42E-01	3.33E+00	**0.00E+00**	<u>1.87E-05</u>	**0.00E+00**
	SD	1.86E-01	2.33E+00	0.00E+00	8.28E-05	0.00E+00
	$w/t/l$	5/0/2	5/0/2	6/1/0	5/0/2	-
	$\#BMF$	2	0	1	0	5

Table 3. Average ranking produced by Friedman test.

Algorithms	APSO	FPSO	CLPSO	FLPSO-QIW	MPSO-CIS
Ranking	3.0000	4.2857	3.2143	2.2857	2.2143
Chi-square Statistic			7.914286		
P value			9.48E-02		

By referring to Table 2, the proposed MPSO-CIS shows the best search accuracy among its competitors by producing five best mean fitness values out of seven benchmark problems. APSO is observed to have competitive search accuracy in dealing with rastrigin and non-continuoud rastrigin problems. For griewank function, MPSO-CIS is reported to be the algorithm ables to locate the global optimum solution. CLPSO and FLPSO-QIW that have similar algorithmic framework

design with MPSO-CIS is reported to have competitive search accuracy in dealing with weierstrass fucntion. CLPSO and MPSO-CIS are able to locate the global optimum solution of weierstrass function, while FLPSO-QIW demonstrate its competitive performance by producing second best mean fitness value. The competitive simulation results imply that the directional information extracted from non-best solutions are beneficial to lead the searching process more effective. According to Table 3, the compared algorithms are ranked from best to worst as MPSO-CIS, FLPSO-QIW, APSO, CLPSO and FPSO with the average ranking values of 2.2143, 2.2857, 3.0000, 3.2143 and 4.2857, respectively.

CONCLUSIONS

In this articles, a new PSO variant named as MPSO-CIS is proposed with the incorporation of chaotic initialization scheme to iniatilize the population. The adopted initializtaion scheme tends to enhance the qualty and diversity of initial population. An unique exemplar derivation module is designed to produce exemplars by leveaging the useful information of non-best solutions, aiming to preserve the population diversity. In contrary to conventional PSO, the exemplars substitute the self-cognitive component of particle, aiming to accomplish better balancing of exploration and exploitation strengths of algorithm. Performance analysis report that MPSO-CIS demonstrates competitive search accuracy and consistency to the existing well-establised PSO variants.

In future works, the convergence characteristics of MPSO-CIS can be investigated. The search performance of MPSO-CIS in tackling complex optimization problems with multi-objectives, constraints, and large dimensional size is worth to be investigated also. Futhermore, the feasibility of MPSO-CIS in optimizing the architecture of artificial neural networks for image processing task can be explored also.

ACKNOWLEDGEMENT

REFERENCES

[1] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of global optimization 39(3) (2007) 459-471. https://doi.org/10.1007/s10898-007-9149-x.

[2] X.S. Yang, Bat algorithm: a novel approach for global engineering optimization, Engineering Computations 29(5) (2012) 464-483. https://doi.org/10.1108/02644401211235834.

[3] M. Melanie, An Introduction to Genetic Algorithms, Massachusetts Institute of Technology1996.

[4] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization 11(4) (1997) 19. https://doi.org/10.1023/A:1008202821328.

[5] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.H. Wong, A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems, Expert Systems with Applications 140 (2020) 112882. https://doi.org/10.1016/j.eswa.2019.112882.

[6] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.K. Ang, C.E. Chow, Z.S. Yeap, Modified Particle Swarm Optimization with Unique Self-cognitive Learning for Global Optimization Problems, Springer Singapore, Singapore, 2022, pp. 263-274.

[7] W.S. Koh, W.H. Lim, K.M. Ang, N.A.M. Isa, S.S. Tiang, C.K. Ang, M.I. Solihin, Multi-objective Particle Swarm Optimization with Alternate Learning Strategies, Springer Singapore, Singapore, 2022, pp. 15-25.

[8] Z.C. Choi, K.M. Ang, W.H. Lim, S.S. Tiang, C.K. Ang, M.I. Solihin, M.R.M. Juhari, C.E. Chow, Hybridized Metaheuristic Search Algorithm with Modified Initialization Scheme for Global Optimization, Advances in Robotics, Automation and Data Analytics: Selected Papers from ICITES 2020 1350 (2021) 172. https://doi.org/10.1007/978-3-030-70917-4_17.

[9] O.T. Chong, W.H. Lim, N.A.M. Isa, K.M. Ang, S.S. Tiang, C.K. Ang, A Teaching-Learning-Based Optimization with Modified Learning Phases for Continuous Optimization, Science and Information Conference, Springer, 2020, pp. 103-124.

[10] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.K. Ang, E. natarajan, M.I. Solihin, A Constrained Teaching-learning-based Optimization with Modified Learning Phases for Constrained Optimization, Journal of Advanced Research in Dynamical and Control Systems 12(04) (2020) 15. https://doi.org/https://doi.org/10.5373/JARDCS/V12SP4/20201623.

[11] M. Alrifaey, W.H. Lim, C.K. Ang, A Novel Deep Learning Framework Based RNN-SAE for Fault Detection of Electrical Gas Generator, IEEE Access 9 (2021) 21433-21442. https://doi.org/10.1109/ACCESS.2021.3055427.

[12] M.I. Solihin, W.H. Lim, S.S. Tiang, C.K. Ang, Modified particle swarm optimization for robust anti-swing gantry crane controller tuning, Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019, Springer, 2021, pp. 1173-1192.

[13] L. Yao, Y.-Q. Chen, W.H. Lim, Internet of things for electric vehicle: An improved decentralized charging scheme, 2015 IEEE International Conference on Data Science and Data Intensive Systems, IEEE, 2015, pp. 651-658.

[14]   L. Yao, W.H. Lim, Optimal purchase strategy for demand bidding, IEEE Transactions on Power Systems 33(3) (2017) 2754-2762. https://doi.org/10.1109/TPWRS.2017.2743765.

[15]   L. Yao, W.H. Lim, S.S. Tiang, T.H. Tan, C.H. Wong, J.Y. Pang, Demand bidding optimization for an aggregator with a Genetic Algorithm, Energies 11(10) (2018) 2498. https://doi.org/10.3390/en11102498.

[16]   S.K. Mohammed, A.A.M. Faudzi, H. Daniyal, B. Muhammad, N.A.A. Aziz, Z.M. Yusof, K.Z.M. Azmi, A. Adam, Improving black hole algorithm using gravitational search, white hole operator, and local search, Mekatronika 1(2) (2019) 8-14. https://doi.org/10.15282/mekatronika.v1i2.4891.

[17]   N.A.A. Aziz, N.H. Abd Aziz, T. Ab Rahman, N. Mokhtar, M. Mubin, Random Synchronous Asynchronous PSO– A Particle Swarm Optimization Algorithm with a New Iteration Strategy, MEKATRONIKA 1(2) (2019) 81-92. https://doi.org/10.15282/mekatronika.v1i2.4989.

[18]   N.M. Yahya, A.R. Yusoff, A. Senawi, M.O. Tokhi, Dual Level Searching Approach for Solving Multi Objective Optimisation Problems using Hybrid Particle Swarm Optimisation and Bats Echolocation-inspired Algorithms, MEKATRONIKA 1(1) (2019) 45-57. https://doi.org/10.15282/mekatronika.v1i1.155.

[19]   H.A. Kasdirin, S.K. Ali, M. Tokhi, Spread enhancement for firefly algorithm with application to control mechanism of exoskeleton system, Mekatronika–Journal of Intelligent Manufacturing & Mechatronics 1(1) (2019) 27-34. https://doi.org/10.15282/mekatronika.v1i1.163.

[20]   X.-S. Yang, Metaheuristic optimization: algorithm analysis and open problems, International Symposium on Experimental Algorithms, Springer, 2011, pp. 21-32.

[21]   A.A. Karim, N.A.M. Isa, W.H. Lim, Hovering Swarm Particle Swarm Optimization, IEEE Access (2021). https://doi.org/10.1109/ACCESS.2021.3106062.

[22]   A.A. Karim, N.A.M. Isa, W.H. Lim, Modified particle swarm optimization with effective guides, IEEE Access 8 (2020) 188699-188725.

[23]   W.H. Lim, N.A.M. Isa, S.S. Tiang, T.H. Tan, E. Natarajan, C.H. Wong, J.R. Tang, A self-adaptive topologically connected-based particle swarm optimization, IEEE Access 6 (2018) 65347-65366. https://doi.org/10.1109/ACCESS.2018.2878805.

[24]   Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39(6) (2009) 1362-1381. https://doi.org/10.1109/TSMCB.2009.2015956.

[25]   M.A.M. De Oca, T. Stutzle, M. Birattari, M. Dorigo, Frankenstein's PSO: a composite particle swarm optimization algorithm, IEEE Transactions on Evolutionary Computation 13(5) (2009) 1120-1132. https://doi.org/10.1109/TEVC.2009.2021465.

[26]   J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE transactions on evolutionary computation 10(3) (2006) 281-295. https://doi.org/10.1109/TEVC.2005.857610.

[27]   Y. Tang, Z. Wang, J.-a. Fang, Feedback learning particle swarm optimization, Applied Soft Computing 11(8) (2011) 4713-4725. https://doi.org/10.1016/j.asoc.2011.07.012.

[28]   F. Van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE transactions on evolutionary computation 8(3) (2004) 225-239. https://doi.org/10.1109/TEVC.2004.826069.

[29]   J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4.

[30]   H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), IEEE, 2005, pp. 695-701.

[31]   M. Zhou, Z. Zhao, C. Xiong, Q. Kang, An opposition-based particle swarm optimization algorithm for noisy environments, 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2018, pp. 1-6.

[32]   M. Abd Elaziz, D. Oliva, S. Xiong, An improved opposition-based sine cosine algorithm for global optimization, Expert Systems with Applications 90 (2017) 484-500. https://doi.org/10.1016/j.eswa.2017.07.043.

[33]   H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Information sciences 181(20) (2011) 4699-4714. https://doi.org/10.1016/j.ins.2011.03.016.

[34]   H. Wang, H. Li, Y. Liu, C. Li, S. Zeng, Opposition-based particle swarm algorithm with Cauchy mutation, 2007 IEEE congress on evolutionary computation, IEEE, 2007, pp. 4750-4756.

[35]   B. Shaw, V. Mukherjee, S. Ghoshal, A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, International Journal of Electrical Power & Energy Systems 35(1) (2012) 21-33. https://doi.org/10.1016/j.ijepes.2011.08.012.

[36]   F.B. Demir, T. Tuncer, A.F. Kocamaz, A chaotic optimization method based on logistic-sine map for numerical function optimization, Neural Computing and Applications 32(17) (2020) 14227-14239. https://doi.org/10.1007/s00521-020-04815-9.

[37]   J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1(1) (2011) 3-18. https://doi.org/https://doi.org/10.1016/j.swevo.2011.02.002.