A Fast-Converging Particle Swarm Optimization through Targeted, Position-Mutated, Elitism (PSO-TPME)

Tamir Shaqarin¹ · Bernd R. Noack²

Received: 25 October 2022 / Accepted: 16 January 2023 © The Author(s) 2023

Abstract

We improve convergence speed by two orders of magnitude and the global exploration capabilities of particle swarm optimization (PSO) through targeted position-mutated elitism (TPME). The proposed fast-converging TPME operator requires a fitness-based classification technique to categorize the particles. The introduced classification is motivated by its simplicity, low memory requirements, and automated termination criteria based on convergence. The three key innovations address particle classification, elitism, and mutation in the cognitive and social model. PSO-TPME is benchmarked against five popular PSO variants for multi-dimensional functions, which are extensively adopted in the optimization field. In particular, the convergence accuracy, convergence speed, and the capability to find global minima are investigated. The statistical error is assessed by numerous repetitions. The simulations confirmed that in ten of the thirteen investigated functions, the proposed PSO variant outperforms other variants in terms of convergence rate and accuracy by at least two orders of magnitude. On the other hand, the simulations demonstrated the early exploration capabilities of PSO-TPME in all tested functions. In the first ten iterations, PSO-TPME outperformed all the investigated PSO variants by at least two orders of magnitude.

Keywords Particle swarm optimization · Elitism · Mutation · Particles’ Classification

1 Introduction

Particle swarm optimization (PSO) is a metaheuristic optimization approach that imitates the dynamics of biological systems such as the swarm of birds. Kennedy and Eberhart [1] published the first seminal paper on PSO. As a result, the variety of research articles ascribed to the development of particle swarm optimization and swarm intelligence grown massively. The PSO algorithm is simple to implement and has a low memory requirements. Consequently, it has been used in a multitude of complex applications that involve optimization, control, machine learning, and so on.

PSO is widely used in the design, sizing, control and maximum power point tracking (MPPT) of renewable energy systems; such as photovoltaic (PV) systems [2–4], wind turbines [5–7] and Hybrid PV-wind systems [8–10]. The authors in [11–13] implemented PSO in the control of robotics for various applications. PSO is also popular in unmanned vehicles in performing path planning [14, 15] and path tracking [16–18].

PSO has been implemented widely in the sizing, design, modeling, and control of various types of refrigeration systems, such as reducing power consumption of multiple chiller systems [19], optimizing cascade refrigeration cycles [20], standing wave thermoacoustic refrigerators [21] and vapor compression refrigeration cycles [22].

Nevertheless, while PSO is efficient in optimizing multi-dimensional complexities [23], its drawbacks involve premature convergence and stagnation to local optima [24–29], as well as a slow convergence rate [24, 25, 28–30]. According to Jiao et al. and Chen [24, 25], the drawbacks of PSO originates from its own structure. The social model readily falls into local optima, practically, each particle in PSO follows rather dynamically the path of the global best position. This global best could be influenced by a local minimum, which leads all particles to a position with poor fitness. Whereas, the cognitive model has the drawback of slow convergence...
especially in the final phases of the evolving iterations when tackling various complexities. This motivates the particles’ classifications in order to treat each category with different updating strategy to achieve better convergence rates and to enhance global explorations capabilities [25, 31–34].

The authors in [24, 31, 35–37] implemented the particle elitism approach to speed up the convergence of PSO. Elitism is a process in which particles with poor fitness values are replaced with particles with the best fitness value (elite particles) after a certain number of iterations, resulting in the production of a new swarm with a better particle’s average fitness. The elitism process in nature requires a preliminary step which is the particle classification, in order to distinguish poor and elite particles. This process will indeed speed up the convergence but on the other hand, it will lessen the diversity of the particles and may increase the probability of falling into a local optimum.

In order to boost particle swarm diversity and mitigate the risk of falling into local optimum, references [24, 38–41] proposed the concept of mutation in PSO algorithm to improve the convergence accuracy and speed. In the literature there are mainly two mutation’s types implemented in the PSO algorithm. The first one is the global best mutation as proposed by [24, 41], in which the particles’ position are mutated indirectly in response to the change of global best. The second type is the direct position mutation as reported by [40], which is better emulating the mutation process inspired by genetic algorithms (GA). The mutation process will enhance the diversity of the swarm which results in better exploration capabilities but it may slow down the convergence as reported by [40].

Janson and Middendorf [42] introduced the concept of hierarchical particle swarm optimization (HPSO). Several authors [43–45] utilized and improved this method to improve the exploration and exploitation performance of the PSO. This approach arranges the particles into a dynamic hierarchy to construct a neighborhood structure. The particles move up or down the hierarchy based on the quality of their fitness. As a result, the higher fitness particles that advance up the hierarchy have a greater effect on the particles’ movements.

To address the above PSO’s disadvantages: slow convergence, loss of particles diversity, premature convergence, and stagnation to local optima. This work will propose a fast-converging targeted position-mutated elitism (TPME) operator that requires a fitness-based classification technique to categorize the particles. The introduced classification is motivated by its simplicity, low memory requirements, and automated termination criteria based on convergence. Initially, the basic PSO will be highlighted, along with several PSO variants that involve the three key elements affecting the convergence speed and global exploration capabilities: particle classification, elitism, and mutation. The proposed PSO method, particle swarm optimization with targeted position-mutated elitism (PSO-TPME), will benefit from these key features in order to enhance both convergence speed and global exploration capabilities by introducing an alternative classification technique, elitism, and targeted position mutation in the PSO algorithm. Following the No Free Lunch (NFL) theorem [46, 47], which establishes the risks of comparing algorithms based on their performance on a limited sample of problems, we will test the proposed PSO on thirteen multidimensional benchmark functions with various complexities in this study.

2 Related Works

Particle swarm optimization is efficient at optimizing multidimensional, complex applications. However, its drawbacks involve premature convergence, stagnation to local optima, [24–29], and a slow convergence rate [24, 25, 28–30]. Many PSO variants addressed the original PSO’s shortcomings. This work discusses the PSO variants incorporating bioinspired operators (mutation and elitism) that necessitate particle classification. The advantages and disadvantages of these operators were discussed in the introduction. It is worth noting that some PSO variants, including PSO-based feature selection [48, 49], effectively addressed the computational complexity of the PSO for extremely high-dimensional problems. This section illustrates basic PSO and several PSO variants incorporating mutation, elitism, and particle classification.

2.1 Particle Swarm Optimization (PSO)

PSO’s method generates a swarm of particles at random positions to represent possible solutions to an optimization task throughout the parameter space. The particle locations are hence iterated with the target of achieving a global optimum of a fitness value. The algorithm then examines each particle’s position and stores the best solution for each particle, which is called personal best ($P_b$). Moreover, during every iteration ($t$), PSO records the best solution throughout the swarm, which is called global best ($G_b$). For every subsequent iteration, the particles’ position ($x$) and velocity ($v$) are determined as a function of the swarm’s best position (social component), the particle’s best personal position (the cognitive component), and its prior velocity (memory component). In general, PSO has several versions, linearly decreasing weight particle swarm optimization is one of the basic versions of PSO, known as LDW-PSO, that reads [1, 50]:

$$ v_{ij}^{t+1} = w v_{ij}^t + c_1 r_1 (P_{bij} - x_{ij}^t) + c_2 r_2 (G_{bj} - x_{ij}^t) $$

(1)

$$ x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} $$

(2)
The subscript $i$ is ranging from 1 to $N$ number of particles and the subscript $j$ is ranging from 1 to $n$ dimensions. The factor $w$ is known as the inertia weight, $w_{\text{max}}$ and $w_{\text{min}}$ are the maximum and the minimum inertia weight, respectively, and the product $wv$ represents the particle’s momentum. The acceleration factors are $c_1$ and $c_2$, while $r_1$ and $r_2$ are the output of random generators ranging from 0 to 1.

### 2.2 PSO With Classification

Chen [25] proposed PSO-M, a study on particle classification that improves the convergence rate and convergence accuracy of PSO. The classification is based on observing the particles’ fitness value at each iteration, identifying the particles’ fitness mean ($\text{aver}$) and also the best and worst fitness values, ($f_{\text{max}}$) and ($f_{\text{min}}$), respectively. Then, two averages are calculated, $\text{aver}_1$ and $\text{aver}_2$, between $f_{\text{max}}$ and $\text{aver}$ and between $f_{\text{min}}$ and $\text{aver}$, respectively. Hence, the fitness space ($f_{\text{min}}$ – $f_{\text{max}}$) is divided into three categories. Considering maximization problem, the classification will be as follows; particles with fitness values in the range ($\text{aver}_1$ – $f_{\text{max}}$) are updated using the cognitive PSO component, particles with fitness values in the range ($f_{\text{min}}$ – $\text{aver}_2$) are updated using the social PSO component, and the remaining particles are updated using the basic PSO model. One drawback of this classification is that all particles are classified starting from the initialization and can even last after the convergence, that is, no classification termination criteria, as seen in Fig. 1.

Rezaei and Safavi [34] introduced a guided adaptive search particle swarm optimization (GuASPSO). With this method, the personal best particles are all classified into clusters using a self-organizing map (SOM). With the evolving iterations, the number of these clusters is decreasing linearly. The weighted average calculated across the best particles of other clusters is used to get the unique global best guide of a specific particle. Due to the well-dispersed particles across the search space, the reasonable distance between each particle and its unique global best guide prevents the particles from becoming trapped in local optima, drifting, or having loss of diversity.

### 2.3 PSO with mutation

Influenced by genetic algorithms (GA), Jiao et al. [24] proposed the concept of elite PSO with mutation (EPSOM) to improve the convergence accuracy and speed. The global best was mutated to boost particle swarm diversity and mitigate the risk of falling into local optimum. The mutation is performed as follows;

$$G_b' = G_b(1 + 0.5\eta)$$

where $\eta$ is a randomly generated number ranging from 0 to 1.

Another recent study by Lü et al. [41] that relies on the global best mutation in a similar manner as in (4), proposed an adaptive weighted and mutation particle swarm...
optimization (AWMPSO) to enhance global search capabilities of the PSO. In which the mutation probability of the global best is now adaptive, where the mutation probability of the global best depend on the population fitness’s variance. Apart from the significant improvement of this approach on the enhanced exploration ability and the convergence speed and accuracy, this approach still has limitations due to the fact that the mutation is implemented on the global best as seen in (4). Hence, the particles’ positions are mutated indirectly through the global best mutation. According to (4), the change in the global best is limited and can vary gradually with the evolving iterations. Furthermore, the mutation is not targeted, since the mutation of the global best affect all particles. In this case, the mutation will indeed be useful to poor or moderate particles but it may also deteriorate the fitness of good particle.

Wang et al. [40] proposed a new method that relies on the position mutation (M-PSO). The particles’ position mutation process is activated conditionally as an intermediate step just before the evaluation of the personal best and the global of the PSO algorithm. The condition requires that a randomly generated number to be less than an adaptively generated threshold value (TH). The threshold value was defined as:

\[
TH = \left( 1 - \frac{i - 1}{It_{\text{max}} - 1} \right)^{\mu_1}
\]

where \(\mu_1\) is the mutation factor. Wang et al. [40] reported that increasing mutation factor can enhance the convergence accuracy but it reduces the convergence speed. As seen in (5), the threshold value is updated by the evolving iterations, not the particles’ fitness, which can be mainly considered as a termination criterion. The threshold value is crucial parameter in M-PSO approach, not only because it is dictating the activation of the mutation process, but because it also decides the upper and the lower bounds of the mutated position. In M-PSO approach, there is no particle classifications, hence, the mutation is performed on all the particles. That is, M-PSO does not use targeted mutation which can contribute negatively on the movement of the good particles and consequently affect the convergence speed as reported by Wang et al. [40].

3 Proposed PSO (PSO-TPME)

The initial concept of the proposed PSO is automated-determination, adaptive particles’ classification with elitism based on particles’ fitness values (f). The classification is performed as follows; the mean (m) of the fitness values of all particles is calculated at each iteration. Then, a fixed percentage (p) around the mean is calculated to generate lower and upper bounds. Now the particles are divided into three categories: “good”, “fair”, and “bad”. Hence, considering a maximization problem, particles with fitness higher than the upper bound (“good” particles) can decrease their velocity to improve exploitation in the local domain via relying on their personal best (PSO cognitive component) instead of the global best (PSO social component). The particles with fitness located within the lower and upper bounds (“fair” particles) and having relatively average fitness can continue both exploration and exploitation while using the basic PSO algorithm. Particles with fitness less than the lower bound (“bad” particles) can initially increase their velocities to enhance exploration in the global search domain via relying on the global best instead of their personal best. If the “bad” particles after several iterations \(N_\epsilon\) are still classified as “bad” particles, then the elitism process is activated to speed up the convergence rate, where \(N_\epsilon\) is the iteration number that initiates the targeted position-mutated elitism. The elitism process is intended to cope with the so-called “hopeless particles,” those given the chance to explore the search space and fail to level up to a better category. Now the elitism simply locates the hopeless particles in the position of the particle with the maximum fitness \(f_{\text{max}}\). This process speeds up the convergence rate significantly with proper choice of \(N_\epsilon\) as shown in the Fig. 2.

Conducting PSO using the earlier approach will drastically reduce the diversity of the particle swarm during the preliminary stages of evolving iterations. As a result, the probability of entrapment in a region of the local optimum is high. Adding a mutation to the particle’s position of the maximum fitness \(x_{f_{\text{max}}}\) will boost the diversification of the elite particle and reduce the possibility of falling into a local optimum. Hence, the mutation is now targeting bad particles only, and the mutation is performed on the particle’s position directly instead of indirect mutation of the position using the global best as proposed by Jiao et al. [24]. The details of the proposed PSO algorithm considering a maximization problem are in (6), in which the elite particle’s positions \((x_{f_{\text{max}}})\) are mutated by \((2\eta + (1 - a))\), where \(\eta\) is a randomly generated number ranging from 0 to 1, \(a\) and \(mp\) are presetting parameters that define the mutation range and mutation probability, respectively. The basic idea is that the elite position \((x_{f_{\text{max}}})\) is mutated in the range \((x_{f_{\text{max}}} \pm (1 + a))\), in which the mean is unity multiplied by \(x_{f_{\text{max}}}\) which gives higher probability to \(x_{f_{\text{max}}}\), that is, exploitation of the elite particle’s position without neglecting the chances of exploring new particles’ positions. The aforementioned range can be increased or decreased via varying \(a\) for higher or lower dimension functions, respectively.

The proposed PSO is further explained in the pseudo code in algorithm 1 and the flowchart in Fig. 3. The algorithm’s inputs are an objective function with predetermined dimensions \((n)\) and the total number of particles \((N)\). The algorithm produces the global best \((G_b)\) at the end.
of the iterations. Two “for loops” are used in the code; the first one iterates till a specific maximum number of iterations, whereas the second loop iterates over the number of particles. The results section will analyze the algorithm’s key parameters ($a$, $mp$, and $N$).

This classification is called automated-termination classification because, after some iterations, all the particles will fall into the middle category since, on one hand, the particles’ fitness values will eventually be close and, on the other hand, the particles’ mean fitness is gradually increasing which will
expand the bounds of the middle category. Hence, the classification will stop automatically as depicted in Fig. 2. This classification is also considered adaptive classification because of particles’ mean variation at each iteration. The particles’ mean will change, therefore the upper and lower bound of the middle category will dynamically vary, as the calculation of these bounds depends only on a fixed percentage around the particles’ mean. This type of particles’ classification is motivated by its implantation simplicity, low memory requirements, and automated-termination criteria based on the particles’ convergence. The latter is very crucial since elitism with mutation takes part in the proposed PSO. Now, the automated termination criteria will stop the classification and the elitism with mutation as well, without manual or other termination criteria being involved, which otherwise may require extra processing.

\[
\begin{align*}
    v_{ij}^{k+1} &= w_{ij}^k + c_1 r_1 (P_{bij} - x_{ij}^k) \quad \text{if } f_{ij}^k > (1 + p)m \\
    x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1},
\end{align*}
\]

\[
\begin{align*}
    v_{ij}^{k+1} &= w_{ij}^k + c_1 r_1 (P_{bij} - x_{ij}^k) + c_2 r_2 (G_{bij} - x_{ij}^k) \quad \text{if } (1 - p)m \leq f_{ij}^k \leq (1 + p)m \\
    x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1},
\end{align*}
\]

\[
\begin{align*}
    v_{ij}^{k+1} &= w_{ij}^k + c_2 r_2 (G_{bij} - x_{ij}^k) \quad \text{if } f_{ij}^k < (1 - p)m \text{ & } It < N_e \\
    x_{ij}^{k+1} &= x_{ij}^k + v_{ij}^{k+1},
\end{align*}
\]

\[
\begin{align*}
    x_{ij}^{k+1} &= x_j(f_{\text{max}})(2a\eta + (1 - a)) \quad \text{if } f_{ij}^k < (1 - p)m \text{ & } It > N_e \text{ & } \eta < mp
\end{align*}
\]

3.1 Benchmark Problems

A group of recognized benchmark multi-dimensional functions, which are extensively adopted in the optimization field, were employed to assess the performance of EPSOM, LDW-PSO, PSO-M, M-PSO, GuAPSO and the suggested PSO algorithm in terms of convergence accuracy and convergence speed. It’s worth noting that all PSO variants were tested in the MATLAB environment on an Intel six-core PC with a 2.2 GHz CPU and 16 GB of RAM. The benchmark functions are used herein as minimization problems; each has various characteristics that offer a distinct level of complexity to the tested algorithm. Function characteristics include unimodality or multimodality, symmetric or asymmetric, and separable or inseparable variables. The benchmark multi-dimensional functions are shown in tables 1 and 2 for unimodal (F1-F7) and multimodal (F8-F13) function, respectively. The unimodal functions (F1-F7)

| Benchmark function | Search space |
|---------------------|--------------|
| \( F_1(x) = \sum_{i=1}^{n} x_i^2 \) | \([-100, 100]\) |
| \( F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \) | \([-10, 10]\) |
| \( F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{n} x_j)^2 \) | \([-100, 100]\) |
| \( F_4(x) = \max\{|x_i|, 1 \leq i \leq n\} \) | \([-100, 100]\) |
| \( F_5(x) = \sum_{i=1}^{n-1} 100(x_i - x_i^2)^2 + (1 - x_i)^2 \) | \([-30, 30]\) |
| \( F_6(x) = \sum_{i=1}^{n} |x_i + 0.5|^2 \) | \([-100, 100]\) |
| \( F_7(x) = \sum_{i=1}^{n} x_i^2 + \text{random}(0, 1) \) | \([1.28, 1.28]\) |
are suitable for examining algorithm exploitation since they have just a single global minimum. The Multimodal functions (F8-F13), on the other hand, contain an enormous number of local minima and can assess algorithm’s exploration capabilities [51]. The selected benchmark functions include or exceed the functions that were adopted by the competing PSO variants (Chen (2010) [25], Jiao et al. (2008) [24], Rezaei and Safavi (2020) [34], Wang et al. (2021) [40]).

Rosenbrock function ($F_3$) is a unimodal function that is extensively used for local exploration, which was first used in optimization assessment of Genetic Algorithms (GA) by De Jong [52]. According to Shang and Qiu [53], the n-dimensional Rosenbrock function ($n > 4$) can be a bimodal function, that makes it even a more complex minimization problem, this complexity also originates from the function’s asymmetry and variables inseparability. The global minimum value of Rosenbrock function is zero, which is located at (1, 1, ...).

Rastrigin function ($F_5$) is a multimodal function that is employed for the performance assessment of evolutionary algorithms [54]. Although it is extremely multimodal, the positions of the minima are evenly dispersed, the function is symmetric and separable. The global minimum value of Rastrigin function is zero that is located at (0, 0, ...). Griewank function ($F_{11}$) is also a multimodal and symmetric function that is widely used for global optimization. The global minimum value of Griewank function is zero that is located at (0, 0, ...). Following Locatelli [55], the function contains a huge number of local minima, which increases exponentially with the number of dimensions. According to Jumonji et al. [54], Griewank function is inseparable in its variables.

### 4 Results

Beneficial to evaluate the efficiency of the proposed PSO algorithm on large-scale problems, the benchmark functions (F1-F13) are set to have 30 dimensions. To assess the performance of the proposed PSO algorithm in comparison to EPSOM, LDW-PSO, PSO-M, GuAPSO and M-PSO, for comparison consistency, we have selected identical parameters settings for all the tested PSO variants. The search and initialization spaces are listed in tables 1 and 2 for unimodal (F1-F7) and multimodal (F8-F13) function, respectively. The number of particles is set to 40, and a maximum iteration of 2000. Furthermore, $w$ is declining linearly from $w_{max} = 0.9$ to $w_{min} = 0.1$, $c1$ is 1.4962, and $c2$ is 1.4962. Regarding the M-PSO, the mutation factor $mu$ is set to 0.05.

The proposed PSO method includes four more presetting parameters: one is the classification percentage around the fitness mean, denoted by $p$, and the other is the elitism with position mutation process initiation iteration, denoted by $N_e$, the last two parameters are the mutation range and mutation probability, that are denoted by $a$ and $mp$, respectively. In these simulations, these parameters are selected as $p = 0.02$, $N_e = 3$, $a = 0.5$, and $mp = 1$. Consequently, the particles with fitness in the range of $\pm 2\%$ of mean particles’ fitness value are classified in the middle category (“fair particles”), the elitism process starts after the third iteration, the mutation range is $\pm 50\%$ of $xf_{max}$, and the mutation probability is 100%.

Twenty independent simulations were carried out on the previously mentioned functions for each minimization algorithm, which is beneficial to minimizing the statistical errors of the optimization performance of the aforementioned...
Fig. 4 Fitness performance of PSO variants on multi-dimensional benchmark functions. Figures a–g depict comparison of EPSOM, LDW-PSO, PSO-M, M-PSO, and GuAPSO with the proposed PSO on uni-modal 30-dimensional functions (F1 - F7). Figure h shows a zoomed-in plot of the first 10 iterations performance of the proposed PSO on the F1 - F7.
algorithms. The outcomes of each algorithm were averaged across 20 simulations. The averaged fitness performance of PSO variants on 30-dimensional unimodal (F1-F7) benchmark functions is depicted in Fig. 4. The optimization results show that the proposed PSO excels EPSOM, LDW-PSO, PSO-M, GuAPSO and M-PSO in terms of convergence speed and exploitation capabilities, on all tested unimodal benchmark functions. Figure 4h shows a zoomed-in plot of the minimization performance of the proposed PSO on the 30-dimensional unimodal functions. The figure undeniably shows the gigantic improvement of the elitism with targeted position mutation on the proposed PSO after the iteration $N_e = 3$. The bad particles are now exploring the particle’s position with maximum fitness with high probability of exploring new particles’ positions as a consequence of position mutation.

To assess the global exploration capabilities of the studied optimization methods, the average fitness performance of PSO variants on 30-dimensional multimodal (F8-F13) benchmark functions is presented in Fig. 5. The optimization results for functions F9, F10, and F11 demonstrate that the proposed PSO highly outperforms EPSOM, LDW-PSO, PSO-M, GuAPSO, and M-PSO. This is supported by Fig. 5b, c, and d, in which PSO-TPME converged to global minimum in less than ten iterations, outperforming other evaluated PSO variants by at least two orders of magnitude in terms of convergence speed and accuracy. However, the optimization results for functions F8, F12, and F13 demonstrate that the proposed PSO was not able to converge to the global minimum, as seen in Fig. 5a, e, and f, respectively. The best-performing algorithms in terms of convergence accuracy in functions F8, F12, and F13 were EPSOM, M-PSO, and GuAPSO, respectively, which means each one of them outperforms PSO-TPME in only one function out of 30. On the other hand, PSO-TPME outperforms the other tested PSO variants in ten out of thirteen functions, as follows: three out of six multimodal functions and seven out of seven unimodal functions. Despite the fact that PSO-TPME with 100% mutation probability has low convergence accuracy in functions F12 and F13, it has high early convergence capabilities in the first ten iterations, as confirmed by Fig. 5g and h. The fast behavior of the proposed PSO and the early exploration capabilities are confirmed for all tested functions, outperforming all tested PSO variants. To investigate why the PSO-TPME with the proposed settings for F12 and F13 was underperforming, functions F12 and F13 were optimized using PSO-TPME with 100% and 1% mutation probability as shown in Fig. 6. The figure clearly shows that the early exploration capabilities are poorer with the 1% mutation probability. The 1% $mp$ case took more than 500 iterations to reach the same performance as the 100% mutation probability in 10 iterations. However, the late exploration capabilities are much better in the 1% $mp$ case, which has superior convergence accuracy. This can be explained by the fact that F12 and F13 are generalized penalized functions with several layers of multitude of local minima. As a result, mutating poor particles with a limited mutation level might result in them staying within one layer of local minima. To avoid falling into this layer, the mutation probability is reduced to allow the poor particle to be updated with the social model, which allows the particles to cross through these layers or at least approach the layer’s boundaries, allowing the poor particles to reach other layers in the case of mutation events.

The advantage of elitism-mutation is illustrated in Fig. 7 for function F11. Here, the learning performance of an arbitrary poor particle was tracked for the first ten iterations in the PSO-TPME and PSO w/o elitism-mutation. The classifier assigned the particle as a “poor” after the third iteration. In the fourth iteration, the particle undergoes the elitism routine, which shifts the formerly “poor” particle to the best-performing particle’s position. Subsequently, the particle position is mutated by a 50% mutation level. This behavior in the PSO-TPME is the driving force behind the excellent exploration and exploitation performance. The fitness of that replaced particle in PSO-TPME is better by more than 15 orders of magnitude as compared to the PSO w/o elitism-mutation. The enhancement in the exploration performance can be seen by the trajectory of the particle in PSO-TPME vs. PSO w/o elitism-mutation.

To assess the effect of the number of dimensions on the performance of the PSO-TPME in terms of convergence speed, early exploration, and late exploration capabilities, the PSO-TPME is tested on 3 selected benchmark functions F5, F9, and F11 with a larger number of dimensions. Figures 8, 9, and 10 show the optimization results for the PSO-TPME, EPSOM, LDW-PSO, PSO-M and M-PSO. The optimal fitness value in the figures is the average optimal one of the solutions in 20 trials. To evaluate the early and late exploration capabilities of the PSO variations, the optimum fitness is estimated after 10 and 2000 iterations, respectively. The optimization was carried out on the previously specified benchmark functions (F5, F9, and F11) with dimensions of 30, 60, and 90 while keeping the same presetting settings for all PSO variations, including the number of particles, constant. The figures undeniably reveal that the PSO-TPME’s
Fig. 5 Fitness performance of PSO variants on multi-dimensional benchmark functions. Figures a–f depict comparison of EPSOM, LDW-PSO, PSO-M, M-PSO, and GuAPSO with the proposed PSO on multi-modal 30-dimensional functions (F8 - F13). Figure g and h shows a zoomed-in plot of the first 10 iterations performance of the proposed PSO on F12 and F13, respectively.
early exploration performance outperforms the overall (early and late) exploration performance of the investigated PSO variants by orders of magnitude, based on the functions F5, F9, and F11. This clearly shows that PSO-TPME has remarkably fast and accurate convergence characteristics, which are supported by benchmark functions with various levels of complexity and a large number of dimensions.

Furthermore, the PSO-TPME is evaluated on the same three benchmark functions F5, F9, and F11 with varied mutation probabilities and mutation levels to analyze the influence of mutation probabilities and mutation levels on the performance of the PSO-TPME. Fig. 11a, c, and e show the optimization results for the PSO-TPME with mutation level $a = 0.5$ and varied mutation probabilities ($mp = [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ %). The figures show that...
increasing the mutation probability accelerates convergence for all investigated functions. Furthermore, late exploration capabilities like in function F5 can benefit from a relatively low mutation probability. On the other hand, very low mutation probabilities, such as in function F9, might result in immature convergence. The same analysis applies to the
mutation level, as seen in Fig. 11b, d, and f, which was tested with \(mp = 100\%\) and varied mutation level \((a = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1])\).

The PSO-TPME with the number of particles \(N = [5 10 30 40 60 80 90 100 400]\), \(mp = 100\%\) and \(a = 0.5\) is evaluated on the 30-dimensional benchmark functions (F5, F9, and F11) to assess the effect of the number of particles on the performance and the time complexity of the PSO-TPME. As seen in Fig. 12a, c, and e, the convergence accuracy is not affected by the number of particles, whereas, the convergence speed is improved in terms of iterations with a larger number of particles with the penalty of increasing evaluations per iteration. The optimal number of particles for the functions F5, F9, and F11 in terms of the number of evaluations was found to be \(N = 10, 30,\) and \(40\), respectively. Implying that the optimal number of particles lies in a limited range around the number of dimensions. Fig. 12b, d, and f depict the average calculation time of PSO-TPME.
for 2000 iterations at various particle numbers. The graphs clearly illustrate that the time complexity of the PSO-TPME increases linearly with particles’ number or in the big $O$ notation the time complexity of the PSO-TPME algorithm is $O(N)$.

5 Conclusions

Starting point are five popular variants of particle swarm optimization. The proposed new PSO variant (PSO-TPME) is shown to dramatically improve convergence speed and early exploration capabilities. This variant comprises the three major factors affecting convergence speed and global exploration capabilities: particles’ classification, elitism, and mutation, as well as the original PSO’s cognitive and social models. This variant introduced an alternative classification approach, elitism, and targeted position mutation, all of which were integrated into the basic PSO algorithm. The introduced particle classification process is simple to apply, requires low memory, is adaptive, and provides automated termination criteria based on convergence. These qualities of the proposed classification technique permitted the implementation of targeted elitism and mutation, in terms of targeting just the poor particles, terminating the elitism and mutation process automatically in the event of convergence.
and applying different updating models (social and/or cognitive) based on the particle’s category.

A set of benchmark 30-dimensional functions (F1 - F13) widely used in the optimization assessment were used to compare the performance of the proposed PSO-TPME to EPSOM, LDW-PSO, PSO-M, GuAPSO, and M-PSO, in terms of convergence accuracy and convergence speed. The tested functions provide different levels of complexity, such as unimodality or multimodality, symmetry or asymmetry, and separability or inseparability. For each benchmark function, many minimization simulations were performed, repeated, and averaged to reduce statistical errors. The simulations revealed that PSO-TPME surpasses the aforementioned variants by orders of magnitude in terms of convergence rate and accuracy in ten out of the thirteen tested functions and the early exploration capabilities in all of the tested functions.

To assess the effect of the number of dimensions on the performance of the PSO-TPME, the PSO-TPME is tested on three benchmark functions: F5 (Rosenbrock), F9 (Rastrigrin), and F11 (Griewank) with the dimensions of 30, 60, and 90. PSO-TPME maintained the same performance in terms of convergence rate and accuracy, as well as early exploration capabilities, with a larger number of dimensions while keeping the same number of particles. Using the same functions (F5, F9, and F11) with 30 dimensions, a complete sensitivity analysis is performed on PSO-TPME’s key parameters: mutation level, mutation probability, and the number of particles. According to the findings, mutation levels (α) and probabilities (mp) greater than 50% would be an appropriate setting for both faster convergence rates and global exploration capabilities. The optimal number of particles based on the fewest number of evaluations required for convergence falls within a narrow range centered on the number of dimensions.

The authors have already achieved impressive optimization performance in a turbulent jet control experiment in high-dimensional space, featuring many local optima. Fast convergence optimization algorithms such as the proposed PSO are expected to be fruitful for optimization-based control for high-dimensional real-life applications.

Acknowledgements The authors acknowledge the funding of the National Science Foundation of China (NSFC) through grants 12172109 and 12172111 and by the Guangdong province, P.R. China via the Natural Science and Engineering grant 2022A1515011492 and by the Shenzhen Research Foundation for Basic Research, China, through grant JCYJ20220531095605012. We thank the anonymous referees for their insightful suggestions. We gratefully acknowledge Zhutao Jiang, Songqi Li and Yutong Liu for discussions and the exciting successful applications of PSO-TPME in several turbulence control experiments.

Author Contributions Conceptualization, TS and BRN; methodology, TS; software, TS; validation, TS and BRN; formal analysis, TS; investigation, TS; resources, TS; data curation, TS; writing—original draft preparation, TS and BRN; writing—review and editing, TS and BRN; visualization, TS; supervision, TS and BRN; project administration, BRN; funding acquisition, BRN. All authors have read and agreed to the published version of the manuscript.

Funding This work is supported by the National Science Foundation of China (NSFC) through grants 12172109 and 12172111 and by the Natural Science and Engineering grant 2022A1515011492 of Guangdong province, P.R. China via the Natural Science and Engineering grant 2022A1515011492 and by the Shenzhen Research Foundation for Basic Research, China, through grant JCYJ20220531095605012.

Availability of Data and Materials The code of PSO-TPME is available upon request from the corresponding author.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical approval and Consent to participate Not applicable.

Consent for Publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN’95-international Conference on Neural Networks, IEEE 4, 1942–1948 (1995)
2. Harrag, A., Messalti, S.: PSO-based SMC variable step size P &O MPPT controller for PV systems under fast changing atmospheric conditions. Int. J. Numer. Model.: Electron. Netw. Devices Fields 32(5), 2603 (2019)
3. Eltamaly, A.M., Al-Saud, M., Abo-Khalil, A.: Performance improvement of PV systems’ maximum power point tracker based on a scanning PSO particle strategy. Sustainability 12(3), 1185 (2020)
4. Shaqarin, T.: A modified reinitialization mechanism for particle swarm optimization based control, case study: PV system. In: 1st International Conference on Automation, Robotics & Communications for Industry 4.0 (ARCI), p. 86 (2021). IFSA
5. Aguilar, M.E.B., Coury, D.V., Reginatto, R., Monaro, R.M.: Multi-objective PSO applied to PI control of DFIG wind turbine under electrical fault conditions. Electr. Power Syst. Res. 180, 106081 (2020)
6. Kamarzarrin, M., Refan, M.H.: Intelligent sliding mode adaptive controller design for wind turbine pitch control system using PSO-SVM in presence of disturbance. J. Control Autom. Electr. Syst. 31(4), 912–925 (2020)
7. Sushmitha, R., Devi, S.C., Manjula, A., Niraimathi, R.: A novel methodology to capture the maximum power from variable speed.
wind turbines using fuzzy proportional integral controllers, modified PSO-GSA algorithm. Materials Today: Proceedings (2021)
8. Ghorbani, N., Kasaeian, A., Toopshekan, A., Bahrami, L., Maghami, A.: Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability. Energy 154, 581–591 (2019)
9. Saad, S.S., Zainuri, M.A.A.M., Hussain, A.: Implementation of maximum power point tracking techniques for PV-wind hybrid energy system: A review. In: 2021 International Conference on Electrical Engineering and Informatics (ICEEI), IEEE pp. 1–6 (2021)
10. El Boujdaini, L., Mezrhab, A., Moussauvi, M.A., Jurado, F., Vera, D.: Sizing of a stand-alone PV–wind–battery–diesel hybrid energy system and optimal combination using a particle swarm optimization algorithm. Electr. Eng. 1–21 (2022)
11. Yifei, T., Meng, Z., Jingwei, L., Dongbo, L., Yulin, W.: Research on intelligent welding robot path optimization based on GA and PSO algorithms. IEEE Access 6, 65397–65404 (2018)
12. Wu, Z., Yu, J., Yan, S., Wang, J., Tan, M.: Motion control method and system for biomimetic robotic fish based on adversarial structured control. Google Patents. US Patent 10,962,976 (2021)
13. Zhang, H., Peng, Q.: PSO and K-means-based semantic segmentation toward agricultural products. Futur. Gener. Comput. Syst. 126, 82–87 (2022)
14. Guo, X., Ji, M., Zhao, Z., Wen, D., Zhang, W.: Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. Ocean Eng. 216, 107693 (2020)
15. Tavossi, V., Marzbanrad, J., Goln vaz, M.: Optimized path planning of an unmanned vehicle in an unknown environment using the PSO algorithm. In: IOP Conference Series: Materials Science and Engineering, vol. 671, p. 012009 (2020). IOP Publishing
16. Al-Mayyahi, A., Wang, W., Birch, P.: Path tracking of autonomous ground vehicle based on fractional order PID controller optimized by PSO. In: 2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMII), IEEE. pp. 109–114 (2015)
17. Amer, N.H., Zamzuri, H., Hudha, K., Aparow, V.R., Kadir, Z.A., Abidin, A.F.Z.: Path tracking controller of an autonomous armed vehicle using modified stanley controller optimized with particle swarm optimization. J. Braz. Soc. Mech. Sci. Eng. 40(2), 1–17 (2018)
18. Jiang, Y., Xu, X., Zhang, L., Zou, T.: Model free predictive path tracking control of variable-configuration unmanned ground vehicle. ISA trans. (2022)
19. Beghi, A., Cecchinato, L., Cosi, G., Rampazzo, M.: A PSO-based algorithm for optimal multiple chiller systems operation. Appl. Therm. Eng. 32, 31–40 (2012)
20. Ghorbani, B., Mafi, M., Shirmohammadi, R., Hamedi, M.-H., Amidpour, M.: Optimization of operation parameters of refrigeration cycle using particle swarm and NLP techniques. J. Nat. Gas Sci. Eng. 21, 779–790 (2014)
21. Rahman, A.A., Zhang, X.: Single-objective optimization for stack unit of standing wave thermoacoustic refrigerator through particle swarm optimization method. Energy Procedia 158, 5445–5452 (2019)
22. Kong, D., Yin, X., Ding, X., Fang, N., Duan, P.: Global optimization of a vapor compression refrigeration system with a self-adaptive differential evolution algorithm. Appl. Therm. Eng. 197, 117427 (2021)
23. Huang, K.-W., Wu, Z.-X., Peng, H.-W., Tsai, M.-C., Hung, Y.-C., Lu, Y.-C.: Memetic particle gravitation optimization algorithm for solving clustering problems. Ieee Access 7, 80950–80968 (2019)
24. Jiao, W., Liu, G., Liu, D.: Elite particle swarm optimization with mutation. In: 2008 Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing, IEEE. pp. 800–803 (2008)
25. Chen, G.: Simplified particle swarm optimization algorithm based on particles classification. In: 2010 Sixth International Conference on Natural Computation, IEEE. 5, 2701–2705 (2010)
26. Liu, Q., Wei, W., Yuan, H., Zhan, Z.-H., Li, Y.: Topology selection for particle swarm optimization. Inf. Sci. 363, 154–173 (2016)
27. Pahnehkolaei, S.M.A., Ali, A., Machado, J.T.: Analytical stability analysis of the fractional-order particle swarm optimization algorithm. Chaos, Solitons Fractals 155, 111658 (2022)
28. Ramírez-Ochoa, D.-D., Pérez-Domínguez, L.A., Martínez-Gómez, E.-A., Luviano-Cruz, D.: PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: a review. Symmetry 14(3), 455 (2022)
29. He, Y., Chen, W., Lei, K., Zhao, Y., Lv, P.: Semi-airborne electromagnetic 2.5 D inversion based on a PSO-LCI strategy. J. Appl. Geophys. 197, 104541 (2022)
30. Liu, S., Liang, M., Hu, X.: Particle swarm optimization inversion of magnetic data: Field examples from iron ore deposits in china. Geophysics 83(4), 43–59 (2018)
31. Angeline, P.J.: Using selection to improve particle swarm optimization. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), IEEE pp. 84–89 (1998)
32. Parrott, D., Li, X.: Locating and tracking multiple dynamic optimizers by a particle swarm model using speciation. IEEE Trans. Evolutionary Comput. 10(4), 440–458 (2006)
33. Zheng, B., Huang, H.-Z., Guo, W., Li, Y.-F., Mi, J.: Fault diagnosis method based on supervised particle swarm optimization classification algorithm. Intell. Data Anal. 22(1), 191–210 (2018)
34. Rezaei, F., Safavi, H.R.: Guaspso: a new approach to hold a better exploration-exploitation balance in pso algorithm. Soft Comput. 24(7), 4855–4875 (2020)
35. Yang, Z.-L., Wu, A., Min, H.-Q.: An improved quantum-behaved particle swarm optimization algorithm with elitist breeding for unconstrained optimization. Comput. Intell. Neurosci. 2015 (2015)
36. Alshammari, M.E., Ramil, M.A., Mehedii, I.M.: An elitist multi-objective particle swarm optimization algorithm for sustainable dynamic economic emission dispatch integrating wind farms. Sustainability 12(18), 7253 (2020)
37. Yang, Z., Wei, X., Qiu, M., Shi, K.: A quantum-behaved particle swarm optimization algorithm with self-adaptive elitist crossover. In: 2021 13th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), IEEE. pp. 177–180 (2021)
38. Higashi, N., Iba, H.: Particle swarm optimization with gaussian mutation. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS’03 (Cat. No. 03EX706), IEEE. pp. 72–79 (2003)
39. Zhang, X., Wang, X., Kang, Q., Cheng, J.: Differential mutation and novel social learning particle swarm optimization algorithm. Inf. Sci. 480, 109–129 (2019)
40. Wang, X., Henshaw, P., Ting, D.S.-K.: Exergoeconomic analysis for a thermoelectric generator using mutation particle swarm optimization (M-PSO). Appl. Energy 294, 116952 (2021)
41. Liu, X., Meng, L., Long, L., Wang, P.: Comprehensive improvement of camera calibration based on mutation particle swarm optimization. Measurement 187, 110303 (2022)
42. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer. In: The 2003 Congress on Evolutionary Computation, 2003. CEC’03. IEEE. 2, 770–776 (2003)
43. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer and its adaptive variant. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 35(6), 1272–1282 (2005)
44. Yu, H., Tan, Y., Zeng, J., Sun, C., Jin, Y.: Surrogate-assisted hierarchical particle swarm optimization. Inf. Sci. 454, 59–72 (2018)
45. Chu, S.-C., Du, Z.-G., Peng, Y.-J., Pan, J.-S.: Fuzzy hierarchical surrogate assists probabilistic particle swarm optimization for expensive high dimensional problem. Knowledge-Based Syst. 220, 106939 (2021)
46. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE trans. Evolutionary Comput. 1(1), 67–82 (1997)
47. Orosz, T., Rassolkin, A., Kallaste, A., Arsénio, P., Pánek, D., Kaska, J., Karban, P.: Robust design optimization and emerging technologies for electrical machines: challenges and open problems. Appl. Sci. 10(19), 6653 (2020)
48. Rostami, M., Forouzandeh, S., Berahmand, K., Soltani, M.: Integration of multi-objective pso based feature selection and node centrality for medical datasets. Genomics 112(6), 4370–4384 (2020)
49. Rostami, M., Berahmand, K., Nasiri, E., Forouzandeh, S.: Review of swarm intelligence-based feature selection methods. Eng. Appl. Artif. Intell. 100, 104210 (2021)
50. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary computation-CEC99 (Cat. No. 99TH8406), IEEE. 3, 1945–1950 (1999)
51. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowledge-Based Syst. 89, 228–249 (2015)
52. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. (Publication No. 7609381) [Doctoral dissertation, University of Michigan], ProQuest Dissertations Publishing, (1975)
53. Shang, Y.-W., Qiu, Y.-H.: A note on the extended rosenbrock function. Evolutionary Comput. 14(1), 119–126 (2006)
54. Jumonji, T., Chakraborty, G., Mabuchi, H., Matsuhara, M.: A novel distributed genetic algorithm implementation with variable number of islands. In: IEEE Congress on Evolutionary Computation, pp. 4698–4705 (2007). doi:10.1109/CEC.2007.4425088
55. Locatelli, M.: A note on the griewank test function. J. Glob. Optim. 25(2), 169–174 (2003)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.