Soft Computing (2022) 26:11695–11711
https://doi.org/10.1007/s00500-022-07489-8

OPTIMIZATION

A hybrid genetic-particle swarm optimization algorithm for multi-constraint optimization problems

Bosong Duan[1] · Chuangqiang Guo[1] · Hong Liu[1]

Accepted: 21 August 2022 / Published online: 17 September 2022

Abstract
This paper presents a new hybrid genetic-particle swarm optimization (GPSO) algorithm for solving multi-constrained optimization problems. This algorithm is different from the traditional GPSO algorithm, which adopts genetic algorithm (GA) and particle swarm optimization (PSO) in series, and it combines PSO and GA through parallel architecture, so as to make full use of the high efficiency of PSO and the global optimization ability of GA. The algorithm takes PSO as the main body and runs PSO at the initial stage of optimization, while GA does not participate in operation. When the global best value (gbest) does not change for successive generations, it is assumed that it falls into local optimum. At this time, GA is used to replace PSO for particle selection, crossover and mutation operations to update particles and help particles jump out of local optimum. In addition, the GPSO adopts adaptive inertia weight, adaptive mutation parameters and multi-point crossover operation between particles and personal best value (pbest) to improve the optimization ability of the algorithm. Finally, this paper uses a nonlinear constraint problem (Himmelblau's nonlinear optimization problem) and three structural optimization problems (pressure vessel design problem, the welded beam design problem and the gear train design problem) as test functions and compares the proposed GPSO with the traditional GPSO, dingo optimization algorithm, whale optimization algorithm and grey wolf optimizer. The performance evaluation of the proposed algorithm is carried out by using the evaluation indexes such as best value, mean value, median value, worst value, standard deviation, operation time and convergence speed. The comparison results show that the proposed GPSO has obvious advantages in finding the optimal value, convergence speed and time overhead.

Keywords Particle swarm optimization · Genetic algorithm · Multi-constraint optimization problem · Genetic-particle swarm optimization algorithm

1 Introduction

In the fields of engineering and science, many practical problems can be regarded as optimization problems. The mathematical models of these problems are often complicated. The traditional algorithms are limited in solving these problems and cannot obtain ideal results (Garg 2016). Therefore, a series of metaheuristic algorithms came into being (Garg 2014); scholars around the world used their optimization algorithms to solve multi-constraint optimization problems in practical engineering. For example, the improved gradient-based optimization algorithm (IGBO) was used to accurately define the static model of photovoltaic panel characteristics under various environmental conditions (Abd Elaziz et al. 2022; Jamei et al. 2022); arithmetic optimization algorithm (AOA) and its improved algorithms such as improved algorithm optimization algorithm (IAOA), logarithmic spiral algorithm optimization algorithm (LS-AOA), chaotic quasi-oppositional arithmetic optimization algorithm (COAOA) were applied to wind power prediction (Al-qaness et al. 2022), optimized solar cell (Abbassi et al. 2022), power grid design (Kharrich et al. 2022), thermal economy optimization of shell-tube condenser with mixed refrigerant (Turgut et al. 2022), control of electric system (Ekinci et al. 2022) and other practical engineering problems; Hernan et al. (2021) proposed DOA to solve the optimization problem of

✉ Chuangqiang Guo
chuangqiang.guo@hit.edu.cn

[1] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

PID control parameters. Mirjalili et al. proposed WOA (2016) and GWO (2014) to solve traditional engineering problems such as tension/compression spring design; He et al. (2004) and He and Wang (2007) proposed an improved PSO to solve the optimization of multi-constraint mechanical design problem; Dimopoulos (2006) used GA to solve mixed-integer engineering design optimization problems; Zhang et al. (2019) proposed an algorithm to predict financial time series based on improved GA and neural network.

Among the above optimization algorithms, the swarm intelligence (SI) algorithm represented by PSO and the evolutionary algorithm (EA) represented by GA are widely used in practical engineering. Compared with various new algorithms, GA and PSO have extremely distinct advantages and disadvantages, such as GA has strong global optimization capability, but it has no memory and is easy to lose the optimal solution; moreover, the operation time is long and the efficiency is low (Coello 2000; Coello and Montes 2002); PSO has memory and high efficiency, but it is easy to fall into local optimum (Liu et al. 2015; Abdelhalim et al. 2019; Rahman et al. 2020). In view of the complementary advantages and disadvantages of GA and PSO, scholars are studying the hybrid algorithm of GA and PSO, namely GPSO, hoping to integrate the advantages of the two algorithms perfectly and get an algorithm that is most suitable for solving problems in practical engineering (Zhang 2021; Song 2018; Alrufaiaat and Althahab 2021; Zhao et al. 2019; Guan et al. 2019; Guo et al. 2018). However, the proposed GPSO is all serial algorithms, that is, on the basis of PSO algorithm, PSO and GA are used to update the particles at the same time. Although such GPSO algorithm can solve the problem that the traditional PSO is easy to fall into local optimum in the iterative process, due to the serial connection of GA, there are also problems such as the decrease of optimization efficiency and the large operation overhead.

Aiming at the above problems in the existing GPSO algorithm, this paper fully combines the efficiency of PSO with the global optimization ability of GA and proposes a new parallel GPSO optimization algorithm to better solve the multi-constrained optimization problem. This algorithm takes PSO as the main body, and GA as a means of updating particles in parallel to complete the optimization task. The parallel relationship refers to that when the algorithm does not fall into local optimum, PSO is used to update the particles alone. When the algorithm falls into local optimum, GA is used to select, crossover and mutation of the particles alone to update the particles, which helps the algorithm jump out of local optimum. Once the algorithm jumps out of local optimum, PSO is continued to be used for optimization. Since PSO has memory, GA does not destroy the found optimal value. The parallel structure gives full play to the advantages of PSO efficiency and GA global optimization ability, avoids the disadvantages of PSO easily falling into local optimum and GA low optimization efficiency. Moreover, to solve the multi-constrained optimization problem better and faster, this GPSO uses adaptive inertia weight, adaptive mutation parameters and multi-point crossover operation between particles and pbest to improve the global optimization ability of the algorithm. In Sect. 4, a classical nonlinear constrained optimization problem and three classical structural optimization problems are used to test the performance of the algorithm. The results show that the proposed GPSO can perform very well in solving multi-constrained optimization problems. The following points can summarize the main contributions of the paper:

- A new parallel GPSO algorithm is proposed, which fully integrates the efficiency of PSO and the global optimization ability of GA.
- Evaluation the proposed GPSO algorithm's efficiency and its performance on Himmelblau's nonlinear optimization problem.
- Evaluation of the proposed GPSO algorithm's efficiency and its performance on the structural optimization problems (pressure vessel design problem, welded beam design problem and gear train design problem).
- In order to verify the optimization performance of the proposed GPSO algorithm, the algorithm is compared with the latest GPSO algorithm and several other latest algorithms. The performance evaluation of the proposed algorithm is carried out by using the evaluation indexes such as best value, mean value, median value, worst value, standard deviation, operation time and convergence speed.

The rest of the paper is organized as follows: Sect. 2 introduces the GPSO algorithm proposed in recent years; Sect. 3 presents the proposed GPSO algorithm; Sect. 4 presents the results and discussions of the function test; Sect. 5 summarizes the conclusions of this paper and future work.

2 Relative work

In practice, we are often faced with many optimization problems. EA and SI algorithm are the two most commonly used algorithms to solve practical engineering problems, because these two types of algorithms are simple and easy to implement and do not require high resolution of the objective function. The most representative of them are GA and PSO. However, as mentioned above, GA has strong global optimization ability but low efficiency; PSO has high efficiency, but it is easy to fall into local optimum. So many scholars are studying GPSO, hoping to integrate the advantages of the two algorithms perfectly.

Zhang (2021) proposed GPSO to introduce crossover and mutation operations of GA into PSO. The fitness values of the particles are calculated and sorted, and then, the particles with lower fitness are eliminated, and the remaining particles are crossed and mutated. Particles with better fitness are crossed with pbest or gbest until the total number of particles reaches the total number of particles before elimination. This can fully improve the quality of particles, which improved the ability to search for optimization. Song et al. (2018) proposed a GPSO algorithm based on differential evolution. On the basis of PSO, the differential evolution of each three particles is carried out to replace the existing particles to improve the quality of particles. Su et al. (2021) integrated the ant colony algorithm (AC) on the basis of GPSO algorithm, used the high efficiency of GPSO to update the velocity and position of particles, and used the global optimization ability of AC to update the fitness value of particles. The GPSO algorithms proposed by Sheng et al. (2021) and Aravinth et al. (2021) are similar to the algorithm proposed by Zhang (2021). The difference is that the parent generations of the crossover operation are randomly selected in the current population, which can improve the diversity of the population and avoided falling into local optimum. The GPSO algorithm proposed by Salaria et al. (2021) is based on PSO, while the inertia weight is optimized by the algorithm, and the quasi-oppositional population-based global particle swarm optimizer is used to update and evolve the particles, which greatly improves the optimization ability of the algorithm. The GPSO algorithm proposed by Chen and Li (2021) has made two improvements based on GPSO. Firstly, the global inertia weight is introduced to improve the global search ability in the whole optimization process. Second, using a small probability mutation operation, GPSO algorithm is easy to jump out of local optimum. Alrufaiaat and Althahab (2021) proposed the GPSO algorithm, where a new update formula is innovated by combing the GPSO with the gradient ascent/descent algorithm, so that the update of particle position can be completed faster and better. GPSO algorithm proposed by Allawi et al. (2020) uses greedy algorithm to select and optimize pbest and gbest, which can avoid falling into local optimum. Gao et al. (2020) proposed a GPSO algorithm using binary coding system and gradient penalty. In this algorithm, the particles are arranged in gradient according to the size of the fitness value, so that the particles with high quality are updated preferentially. This purpose is to solve the problem that when the number of particles in PSO is large, PSO cannot judge the priority of particle update, which leads to low convergence efficiency. Mir et al. (2020) proposed a GPSO algorithm, which made two improvements based on PSO. The first is to change inertia weight into adaptive parameters. The second is to use Gaussian function to optimize the particle update formula, which can accelerate the particle update and improve the optimization speed and quality. Zhao et al. (2019) proposed a grouping GPSO algorithm, in which the population is composed of several groups. For each iteration, an elite group is constructed to replace the worst group. Grouping is conducive to improving the diversity of solutions, thereby enhancing the global search ability of the algorithm. Guan et al. (2019) proposed a GPSO algorithm, which integrates the crossover and mutation operation of GA into the optimization iteration process of PSO and adaptively processes the crossover parameters, mutation parameters, inertia weight and learning factor parameters to enhance the ability of population to jump out of local optimum. Guo et al. (2018) proposed a grouping GPSO-PG algorithm based on individual best position guidance, which maintains the diversity of population by preserving the diversity of samples. On the one hand, the uniform random allocation strategy is used to assign particles to different groups, and the loser in each group will learn from the winner. On the other hand, the pbest of each particle in social learning is used to replace the gbest. This not only increases the diversity of samples, but also eliminates the dominant influence of gbest on optimization and prevents falling into local optimum. Al-Bahrani and Patra (2018) proposed an OPSO algorithm. The algorithm uses the orthogonal diagonalization between the particles with good performance and the residual particles to replace the crossover operation of GA to update the particles. All the GPSO algorithms mentioned above are series algorithms. Although they can also ensure the optimization ability of the algorithm, the series algorithm will reduce the optimization efficiency of the algorithm and increase the overhead of the algorithm. Therefore, parallel GPSO algorithm is required to improve the optimization efficiency and reduce the algorithm overhead while ensuring the optimization ability. Table 1 shows the similarities and differences between the proposed GPSO algorithm and the other GPSO algorithm (except the differences in parallel architecture).

3 GPSO

The GPSO proposed in this paper takes PSO as the main body and GA as a means of updating particles in parallel to complete the optimization task. The algorithm can fully combine the advantages of PSO efficiency and GA excellent global optimization ability. The operation of the algorithm can be divided into the following three stages:

1. First stage

Table 1 Similarities and differences between the proposed GPSO and the other GPSO

Algorithm	Similarities	Differences
GPSO by Zhang (2021)	PSO as main body, GA as auxiliary;	The mutation parameter of the GPSO by Zhang is constant, the mutation effect is not good, and it is not easy to jump out of local optimum;
	Crossing particles with pbest	GPSO in this paper adopts adaptive mutation parameters, which can improve the mutation probability and easily jump out of local optimum
GPSO by Song et al. (2018)	PSO as main body, EA as auxiliary	GPSO by Song uses the differential evolution method of three particles to update the particles, but if the particles are similar, it cannot produce new particles and is easy to fall into local optimum;
		GPSO in this paper updates the particles with the selection, crossover and mutation of GA. GA has a strong global optimization ability to help the algorithm jump out of local optimum
GPSO by Sheng et al. (2021)	PSO as main body, GA as auxiliary	GPSO by Sheng randomly selects the parent generations to cross in the particle, increasing the randomness of the offspring, and the quality of the particles after crossover is not high;
		GPSO in this paper uses particles to cross with pbest, and the quality of particles after crossover is high, which can improve the optimization efficiency
GPSO by Aravinth et al. (2021)	PSO as main body, GA as auxiliary;	GPSO by Aravinth randomly selects the parent generations to cross in the particle, increasing the randomness of the offspring, and the quality of the particles after crossover is not high;
	Adopts adaptive mutation parameters	GPSO in this paper uses particles to cross with pbest, and the quality of particles after crossover is high, which can improve the optimization efficiency
GPSO by Chen and Li (2021)	PSO as main body, GA as auxiliary;	The mutation parameter of GPSO by Chen is constant, the mutation effect is not good, and it is not easy to jump out of local optimum;
	Adopts adaptive inertial weight	GPSO in this paper adopts adaptive mutation parameters, which can improve the mutation probability and easily jump out of local optimum
GPSO by Allawi et al. (2020)	PSO as main body	GPSO by Allawi uses greedy algorithm to calculate multiple pbest and gbest, and selects multiple pbest and gbest to avoid falling into local optimum, but the convergence speed will slow down
GPSO by Gao et al. (2020)	PSO as main body	GPSO by Gao adds gradient penalty, so that the particles with high fitness will be updated first, and the particles with fitness will be eliminated. Although it will improve the convergence speed, it is easy to fall into local optimum;
	Sorting and selecting particles	GPSO in this paper updates the particles with the selection, crossover and mutation of GA. GA has a strong global optimization ability to help the algorithm jump out of local optimum
GPSO by Mir et al. (2020)	PSO as main body;	GPSO by Mahai uses Gaussian function to update particles. It can improve the convergence speed, but it is easy to fall into local optimum;
	Adopts adaptive inertial weight	GPSO in this paper updates the particles with the selection, crossover and mutation of GA, which is easy to jump out of local optimum
GPSO by Zhao et al. (2019)	PSO as main body, GA as auxiliary	GPSO by Zhao population is grouped, and the elite group is used to eliminate the worst group to further optimize the particles. Each step of this algorithm will produce large amount of calculation, resulting in slow convergence speed and large overhead;
		GPSO in this paper only uses particles to cross with pbest, which can ensure the high quality of particles after crossover and improve the optimization efficiency
GPSO by Guan et al. (2019)	PSO as main body, GA as auxiliary;	GPSO by Guan randomly selects the parent generations to cross in the particle, increasing the randomness of the offspring, and the quality of the particles after crossover is not high;
	Adopts adaptive inertial weight;	GPSO in this paper uses particles to cross with pbest, and the quality of particles after crossover is high, which can improve the optimization efficiency
	Adopts adaptive mutation parameters	
GPSO-PG by Guo et al. (2018)	PSO as main body, GA as auxiliary	GPSO-PG by Guo uses pbest instead of gbest in each particle sociology department to prevent falling into local optimum, but it will lead to slower convergence speed and lower optimization efficiency than GPSO proposed in this paper

Table 1 (continued)

Algorithm	Similarities	Differences
OPSO by Al-Bahrani and Patra (2018)	PSO as main body	OPSO by Al-Bahrani uses the orthogonal pairing of high-quality particles and residual particles to update particles, so the optimization efficiency is not high, and it is easy to fall into local optimum;
		GPSO in this paper uses particles to cross with pbest, and the quality of particles after crossover is high, which can improve the optimization efficiency. Moreover, using adaptive mutation parameters, improve the mutation probability, easy to jump out of local optimum
GPSO by Alrufaiaat and Althahab (2021)	PSO as main body	GPSO by Alrufaiaat uses gradient ascent/descent algorithm to update particles. It can improve the convergence speed, but it is easy to fall into local optimum;
		GPSO in this paper updates the particles with the selection, crossover and mutation of GA, which is easy to jump out of local optimum

In the first stage, in order to ensure the efficiency of optimization, GA does not participate in the updating of particles and only uses PSO to update particles, so that gbest quickly approaches the optimal value. During this period, the velocity and position of particles are updated by Eqs. (1) and (2) as follows:

$$v_{k+1}^{i} = w \cdot v_{k}^{i} + c_1 \cdot \mathrm{rand}_1 \cdot \left(p_{k}^{i} - x_{k}^{i}\right) + c_2 \cdot \mathrm{rand}_2 \cdot \left(p_{k}^{g} - x_{k}^{i}\right) \tag{1}$$

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i} \tag{2}$$

where v_k^i is the particle velocity,x_k^i is the particle position, w is the inertia weight, c_1 and c_2 are constants known as the social and cognitive parameters, rand_1 and rand_2 are random numbers in the range of [0,1], p_k^i is pbest, p_k^g is gbest.

The right side of Eq. (1) consists of three terms: The first item is the "inertia" part, which represents the tendency of particles to maintain their own speed; the second item is the "cognition" part, which indicates that the particle has a tendency to approach the best position in its own history; the third item is the "social" part, which represents the tendency of the particle to approach the best position in the history of the group or neighborhood. The first item guarantees the global convergence performance of the algorithm, and the second and third items guarantee the ability of local convergence. Therefore, in order to ensure the global search capability of the algorithm, w is dynamically adjusted during the search process in Eq. (3):

$$w = w_{\max} - \frac{\left(w_{\max} - w_{\min}\right) \cdot t}{T_{\max}} \tag{3}$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum of the inertia weight; t is the current number of iterations; $T_{\max}$ is the maximum number of iterations.

2. Second stage

When the continuous $\mathrm{num}_{\max}$ generation of gbest does not change, the algorithm assumes that it falls into the local optimum (It may also find the optimal value, while since the particle swarm algorithm has memory, it will not destroy the optimal value). At this time, it is difficult to help the algorithm jump out of the local optimum by continuing to use the PSO update particles in the first stage. Therefore, the algorithm enters the second stage, that is, the selection, crossover and mutation of GA are used to help the algorithm jump out of the local optimum. In the crossover operation, multi-point crossover operation between particles and pbest can quickly improve the quality of the population and improve the optimization efficiency. In the second stage, once gbest changes, it indicates that GA has completed its mission and has helped the algorithm jump out of local optimum, so the algorithm jumps back to the first stage to continue optimization. Equations (4) and (5) for updating particles by GA are as follows:

$$x_{k+1}^{i(\mathrm{rand}<p_c)} = \mathrm{rand}i_D \cdot x_{k}^{i(\mathrm{rand}<p_c)} + \left(1 - \mathrm{rand}i_D\right) \cdot p_{k}^{i(\mathrm{rand}<p_c)} \tag{4}$$

$$x_{k+1}^{i(\lceil N\cdot \mathrm{rand}\rceil)}{}_{\lceil \mathrm{rand}\cdot D\rceil} = x_{k+1_{\lceil \mathrm{rand}\cdot D\rceil_{\max}}}^{i} - \mathrm{rand} \cdot \left(x_{k+1_{\lceil \mathrm{rand}\cdot D\rceil_{\max}}}^{i} - x_{k+1_{\lceil \mathrm{rand}\cdot D\rceil_{\min}}}^{i}\right) \tag{5}$$

where p_c is probability of crossover, $x_k^{i(\mathrm{rand}<p_c)}$ are the crossing particles, D is the dimension of the particles, $\mathrm{rand}i_D$ is the binary vector with length D, $p_k^{i(\mathrm{rand}<p_c)}$ is the pbest of crossing particle, N is the number of particle swarm, $\lceil\ \rceil$ represents an upward integer, $x_{k+1}^{i(\lceil N\cdot \mathrm{rand}\rceil)}{}_{\lceil \mathrm{rand}\cdot D\rceil}$ represents the random particles in the particle swarm participating in the mutation operation, Mutation operation $N{\cdot}p_m$ times, p_m is the mutation probability. $x_{k+1_{\lceil \mathrm{rand}\cdot D\rceil_{\max}}}^{i}$ and $x_{k+1_{\lceil \mathrm{rand}\cdot D\rceil_{\min}}}^{i}$ are the maximum and minimum values of the particle.

3. Third stage

When the continuous $\mathrm{NUM}_{\max}$ generation of gbest does not change, it indicates that the operation in the second stage of the algorithm cannot help the algorithm jump out of the local optimum. Moreover, after several generations of evolution, most of the particles have been trapped in the local optimum. At this time, the population structure needs to be greatly changed. By using Eqs. (6) and (7), the mutation probability is greatly improved, so that the individuals in most particles are randomly updated and the maximum possible jump out of local optimum. Once gbest changes, it indicates that GA has completed its mission and has helped the algorithm jump out of the local optimum. Therefore, the algorithm jumps back to the first stage for further optimization, otherwise, it will always use GA to update particles.

$$p_m = \left(\frac{t}{T_{\max}}\right)^{\gamma} \cdot p_{m_{\max}} \tag{6}$$

$$\gamma = \begin{cases} 1 & \mathrm{NUM} < \mathrm{NUM}_{\mathrm{MAX}} \\ 0.5 & \mathrm{NUM} \geq \mathrm{NUM}_{\mathrm{MAX}} \end{cases} \tag{7}$$

where $p_{m_{\max}}$ is the maximum probability of mutation, γ is the coefficient of mutation.

Through the above steps, the whole process of GPSO optimization is completed. The GPSO flowchart is shown in Fig. 1:

3.1 Time complexity analysis

Since the running time of an algorithm is greatly affected by computer performance, the time complexity of an algorithm is generally used to evaluate the execution efficiency of an algorithm. Time complexity can also be understood as a relative measure of the running time of an algorithm, usually described by the big O notation, that is, $T\ (n)\ =\ O\ (f(n))$.

Because PSO and GA are included in the algorithm, there are many iterations in the algorithm. The time complexity of the algorithm is estimated by taking the frequency of the most repeated sentences in the algorithm. The most frequently executed statement in the algorithm is the judgment statement in PSO whether the velocity and position of the particle after updating exceed the boundary. Three loops are needed to execute the statement. The first loop is the iteration of the whole algorithm, the second loop is the PSO population updating, and the third loop is to determine whether the velocity and position of each particle exceed the boundary. Since the algorithm has three loops, its time complexity is $T\ (n)\ =\ O\ (n^3)$. The time complexity of other serial GPSO algorithms is also $O\ (n^3)$. However, since GA is always involved in the operation in the serial GPSO algorithm, and the parallel GPSO only participates in the operation when the algorithm falls into local optimum, the operation overhead of the parallel GPSO is lower than that of the serial GPSO. The analysis of time complexity is only a rough estimate of the efficiency and overhead of the algorithm. The comparison of the running time of the specific algorithm is shown in Sect. 4.

4 Verification and analysis of algorithm performance

In order to verify the effectiveness of the proposed GPSO, this paper selects several classical multi-constrained optimization problems to test the algorithm. In order to eliminate the influence of computer performance on the algorithm, this paper reproduces several latest GPSO algorithms and tests them under the same computer; in order to eliminate the influence of algorithm stability on the results, this paper takes the best value, mean value, median value, worst value, standard deviation, operation time and convergence speed of each algorithm in 100 experiments as the evaluation indexes of algorithm performance, so as to fully eliminate the contingency of the results and ensure that the results have statistical significance.

4.1 Constrained optimization problem

4.1.1 Himmelblau's nonlinear optimization problem

The performance of the proposed GPSO is verified by Himmelblau's nonlinear optimization problem. The problem was first proposed by Himmelblau and is now widely used to validate nonlinear constrained optimization algorithms. This problem contains five optimization variables and six nonlinear constraints, as follows:

$$x = [x_1, x_2, x_3, x_4] \tag{8}$$

$$\mathrm{Min} f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \tag{9}$$

s.t.

$$0 \leq g_1(x) \leq 92 \tag{10}$$

$$90 \leq g_2(x) \leq 110 \tag{11}$$

$$20 \leq g_3(x) \leq 25 \tag{12}$$

where

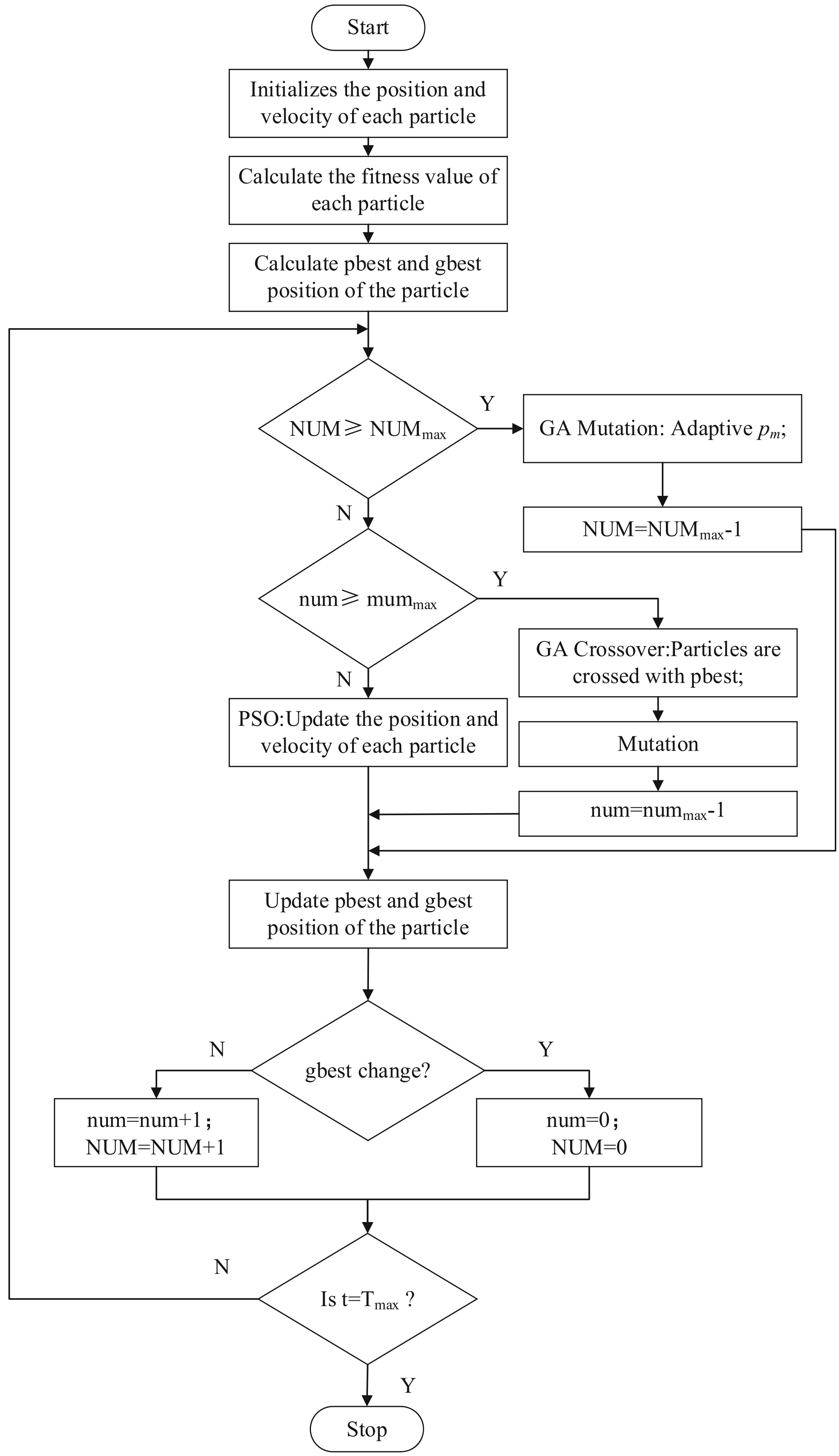

Fig. 1 Flowchart of GPSO

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \quad (13)$$

$$g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_3^2 \quad (14)$$

$$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \quad (15)$$

$$78 \le x_1 \le 102 \quad (16)$$

$$33 \le x_2 \le 45 \quad (17)$$

$$27 \le x_3, x_4, x_5 \le 45 \quad (18)$$

This paper uses this problem to test the performance of the proposed GPSO algorithm and compares it with several latest GPSO algorithms. The comparison results are shown in Tables 2 and 3.

In Table 2, different algorithms have achieved good results in optimizing this problem, but there are small differences in the results of the optimal value. The optimal value of GPSO proposed in this paper is ***− 30,659.81***, which is closer to the real optimal value compared with other results in the table. At this time, the parameter is: [x_1, x_2, x_3, x_4, x_5] = [78.01, 33.01, 30.01, 44.94, 36.78], which is within the constraint range of the parameter and meets the parameter requirements. The constraint result obtained by substituting the parameters into the constraint equations is: [g_1, g_2, g_3] = [91.99, 98.85, 20.00], which also satisfies the constraint condition. Through comparative analysis, it is proved that the optimization algorithm proposed in this paper can obtain a better optimal value to Himmelblau's nonlinear optimization problem.

Table 3 shows the algorithm performance of each algorithm in solving Himmelblau's nonlinear optimization problem. The optimal value of GPSO proposed in this paper is ***− 30,659.81***, which is closer to the real optimal value compared with other results in the table; it shows that the GPSO proposed in this paper can get a better optimal value. The average value of the optimal value obtained by 100 experiments is ***− 30,597.25***, which is smaller than the optimal value of − 30,595.62 in the above algorithm. It shows that the algorithm can obtain better results in many repeated experiments and proves that the algorithm has a better stability. The standard deviation is ***59.54***, which is smaller than the minimum deviation of 60.26 in the above algorithm, indicating that the stability of the algorithm is better. The time of 100 experimental results is only ***2.62*** s. Compared with the shortest time of 2.85 s in the above algorithm, the time can be shortened by 8%, indicating that the algorithm has high optimization speed and low time complexity. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can obtain better optimal value and has good performance in stability and efficiency.

In addition, Fig. 2 shows the convergence performance of each algorithm in solving Himmelblau's nonlinear optimization problem. It can be seen from Fig. 2 that the GPSO proposed in this paper has a higher convergence speed compared with other algorithms. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can get a better optimal value and has good performance in stability, convergence speed and time overhead.

4.2 Structural optimization problems

In order to verify the effectiveness of the algorithm in solving multi-objective optimization problems in practical

Table 2 Comparison of the optimal value for Himmelblau's nonlinear optimization problem found by different algorithms

Method	Variables					$f(x)$
	x_1	x_2	x_3	x_4	x_5	
GPSO by Zhang (2021)	78.01	33.01	30.03	44.94	36.72	− 30,658.18
GPSO by Zhao et al. (2019)	78.01	33.03	30.03	44.60	36.86	− 30,649.58
GPSO by Guan et al. (2019)	78.03	33.08	30.04	44.96	36.66	− 30,655.55
GPSO-PG by Guo et al. (2018)	78.07	33.06	30.03	44.94	36.70	− 30,653.97
OPSO by Al-Bahrani and Patra (2018)	78.76	33.37	30.81	44.56	34.91	− 30,472.52
GPSO by Song et al. (2018)	78.00	33.06	30.07	44.99	36.60	− 30,653.27
DOA by Hernan et al. (2021)	78.05	33.01	30.02	44.98	36.73	− 30,658.72
WOA by Mirjalili and Lewis (2016)	78.08	33.01	30.01	44.99	36.75	− 30,656.05
GWO by Mirjalili et al. (2014)	78.02	33.04	30.02	44.92	36.74	− 30,657.58
GPSO by Alrufaiaat and Althahab (2021)	78.14	33.18	30.14	44.99	36.37	− 30,634.98
GPSO in this paper	78.01	33.01	30.01	44.94	36.78	***− 30,659.81***

Bolditalics values indicated by optimal valve

Table 3 Performance of different algorithms for Himmelblau's nonlinear optimization problem

Method	Best	Mean	Median	Std	Worst	Time
GPSO by Zhang (2021)	− 30,658.18	− 30,595.62	***− 30,652.57***	59.83	− 30,335.32	2.85
GPSO by Zhao et al. (2019)	− 30,649.58	− 30,578.38	− 30,637.13	73.94	− 30,325.04	2.89
GPSO by Guan et al. (2019)	− 30,655.55	− 30,585.12	− 30,634.39	70.96	− 30,359.93	2.87
GPSO-PG by Guo et al. (2018)	− 30,653.97	− 30,572.10	− 30,638.37	76.62	− 30,310.31	2.98
OPSO by Al-Bahrani and Patra (2018)	− 30,472.52	− 30,006.17	− 29,869.29	149.98	− 29,627.54	3.58
GPSO by Song et al. (2018)	− 30,653.27	− 30,573.08	− 30,415.40	71.88	− 30,312.98	3.65
DOA by Hernan et al. (2021)	− 30,658.72	− 30,573.27	− 30,429.80	75.20	− 30,325.79	2.89
WOA by Mirjalili and Lewis (2016)	− 30,656.05	− 30,588.02	− 30,343.84	66.41	− 30,343.84	2.91
GWO by Mirjalili et al. (2014)	− 30,657.58	− 30,584.63	− 30,606.36	60.26	***− 30,404.35***	2.90
GPSO by Alrufaiaat and Althahab (2021)	− 30,634.98	− 30,498.90	− 30,549.20	111.86	− 30,101.50	6.46
GPSO in this paper	***− 30,659.81***	***− 30,597.25***	− 30,611.42	***59.54***	− 30,360.36	***2.62***

In the table, Best is the best of the optimal value in 100 experimental results; mean is the average value of the optimal value in 100 experimental results; median is the median of the optimal value in 100 experimental results; Std is the standard deviation between the optimal value and Best in 100 experimental results; Worst is the worst value of the optimal value in 100 experimental results; Time is the total running time for 100 experiments. All the tables below are the same

Bolditalics values indicated by optimal valve

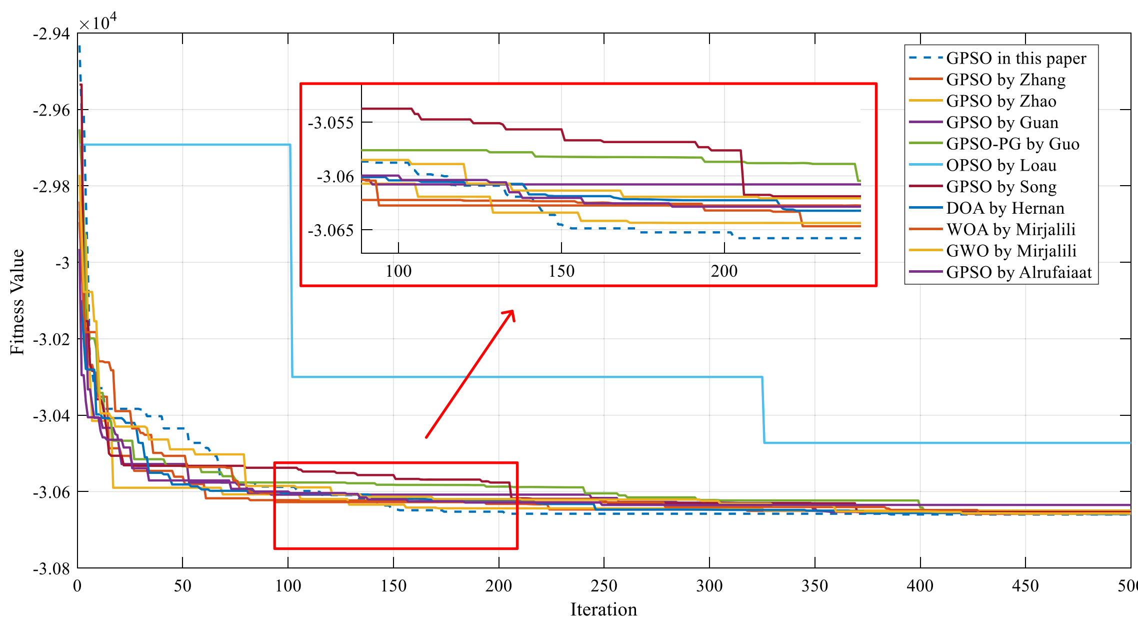

Fig. 2 The convergence performance of each algorithm in solving Himmelblau's nonlinear optimization problem

engineering, comparative tests were carried out on three classic practical engineering problems, including pressure vessel design problem, the welded beam design problem and the gear train design problem.

4.2.1 Pressure vessel problem

The objective of pressure vessel design problem is to minimize the cost of pressure vessel fabrication (matching, forming and welding). The design of the pressure vessel is shown in Fig. 3. Both ends of the pressure vessel have a

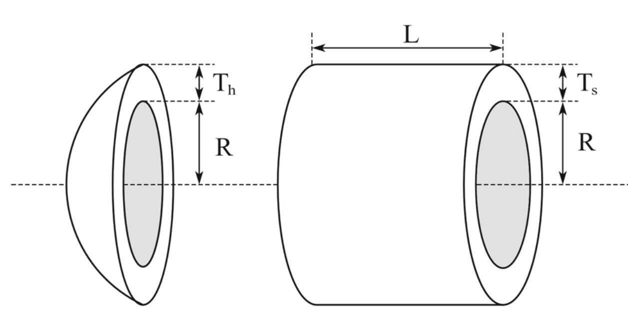

Fig. 3 The design of the pressure vessel

cover top. The cover at one end of the head is hemispherical. The maximum working pressure is 2000 psi, and the maximum volume is 750 ft^3. L is the section length of the cylinder part without considering the head, R is the inner wall diameter of the cylinder part, T_s and T_h represent the wall thickness of the cylinder part and the head. L, R, T_s and T_h are the four optimization variables of the pressure vessel design problem. The objective function and constraints of the problem are expressed as follows:

$$x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L] \tag{19}$$

$$\begin{aligned}\mathrm{Min} f(x) = 0.6224x_1x_3x_4 + 1.7781x_1x_3^2 + 3.1661x_1^2x_4 \\ + 19.84x_1^2x_3\end{aligned} \tag{20}$$

s.t.

$$g_1(x) = -x_1 + 0.0193x_3 \le 0 \tag{21}$$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0 \tag{22}$$

$$g_3(x) = -\pi x_3^2 - \frac{4}{3}\pi x_3^2 + 1296000 \le 0 \tag{23}$$

$$g_4(x) = x_4 - 240 \le 0 \tag{24}$$

$$1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625 \tag{25}$$

$$10 \le x_3, x_4 \le 200 \tag{26}$$

This paper uses this problem to test the performance of the proposed GPSO algorithm and compares it with several latest GPSO algorithms. The comparison results are shown in Tables 4 and 5.

In Table 4, different algorithms have achieved good results in optimizing this problem, but there are small differences in the results of the optimal value. The optimal value of GPSO proposed in this paper is ***5881.0474***, which is closer to the real optimal value compared with other results in the table. At this time, the parameter is: [x_1, x_2, x_3, x_4] = [0.7782, 0.3831, 40.3199, 199.9980], which is within the constraint range of the parameter and meets the parameter requirements. The constraint result obtained by substituting the parameters into the constraint equations is: [g_1, g_2, g_3, g_4] = [− 3.8027e−6, − 1.0516e−4, − 8.6355, − 40.0020], which also satisfies the constraint condition. Through comparative analysis, it is proved that the optimization algorithm proposed in this paper can obtain a better optimal value to pressure vessel design problem.

Table 5 shows the algorithm performance of each algorithm in solving pressure vessel design problem. The optimal value of GPSO proposed in this paper is ***5881.0474***, which is closer to the real optimal value compared with other results in the table, and it shows that the GPSO proposed in this paper can get a better optimal value. The average value of the optimal value obtained by 100 experiments is ***6001.3177***, which is smaller than the optimal value of 6006.4589 in the above algorithm. It shows that the algorithm can obtain better results in many repeated experiments and proves that the algorithm has a better stability. The median value of the optimal value obtained from 100 experiments is ***5882.6562***, which is smaller than the minimum median value of 5892.4465 in the above algorithm, indicating better stability of the algorithm. The time of 100 experimental results is only ***2.3047*** s. Compared with the shortest time of 2.5270 s in the above algorithm, the time can be shortened by 9%, indicating that the algorithm has high optimization speed and low time complexity. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can obtain better optimal value and has good performance in stability and efficiency.

Table 4 Comparison of the optimal value for pressure vessel design problem found by different algorithms

Method	Variables				$f(x)$
	x_1	x_2	x_3	x_4	
GPSO by Zhang (2021)	0.7783	0.3832	40.3241	199.9385	5881.5499
GPSO by Zhao et al. (2019)	0.7785	0.3832	40.3354	199.7809	5881.5583
GPSO by Guan et al. (2019)	0.7788	0.3834	40.3540	199.5214	5881.7955
GPSO-PG by Guo et al. (2018)	0.8167	0.4299	42.1756	177.2461	6091.7792
OPSO by Al-Bahrani and Patra (2018)	0.9607	0.5855	47.2922	127.1722	7161.8969
GPSO by Song et al. (2018)	0.7788	0.3833	40.3505	199.5716	5881.7177
DOA by Hernan et al. (2021)	0.7784	0.3831	40.3286	199.8787	5881.5046
WOA by Mirjalili and Lewis (2016)	0.7784	0.3832	40.3254	199.9366	5882.7597
GWO by Mirjalili et al. (2014)	0.7784	0.3830	40.3196	199.9997	5882.2191
GPSO by Alrufaiaat and Althahab (2021)	0.7964	0.4049	40.9939	191.9829	6012.6369
GPSO in this paper	0.7782	0.3831	40.3199	199.9980	***5881.0474***

Bolditalics values indicated by optimal valve

Table 5 Performance of different algorithms for pressure vessel design problem

Method	Best	Mean	Median	Std	Worst	Time
GPSO by Zhang (2021)	5881.5499	6057.3225	5900.6996	209.4077	6790.7251	6.4480
GPSO by Zhao (2019)	5881.5583	6099.3547	5912.4757	279.1119	7031.5241	2.5270
GPSO by Guan (2019)	5881.7955	6006.4589	5965.4783	***161.1013***	***6555.6666***	2.5667
GPSO-PG by Guo (2018)	6091.7792	8594.3849	13,031.6004	2316.0683	20,570.9776	2.4790
OPSO by Al-Bahrani and Patra (2018)	7161.8969	10,103.2843	7940.2697	1589.4665	15,299.6134	3.7434
GPSO by Song (2018)	5881.7177	6049.3820	5991.5861	207.9317	6819.0669	3.4975
DOA by Hernan (2021)	5881.5046	6017.0942	5985.6417	167.6150	6572.3540	2.5978
WOA by Mirjalili and Lewis (2016)	5882.7597	6047.3411	5892.4465	230.9826	6913.6343	2.5839
GWO by Mirjalili et al. (2014)	5882.2191	6198.6201	6793.7182	237.5923	6930.7830	2.5811
GPSO by Alrufaiaat and Althahab (2021)	6012.6369	6549.9268	6303.4466	267.1786	7333.9928	5.1600
GPSO in this paper	***5881.0474***	***6001.3177***	***5882.6562***	222.4062	6919.2537	***2.3047***

Bolditalics values indicated by optimal valve

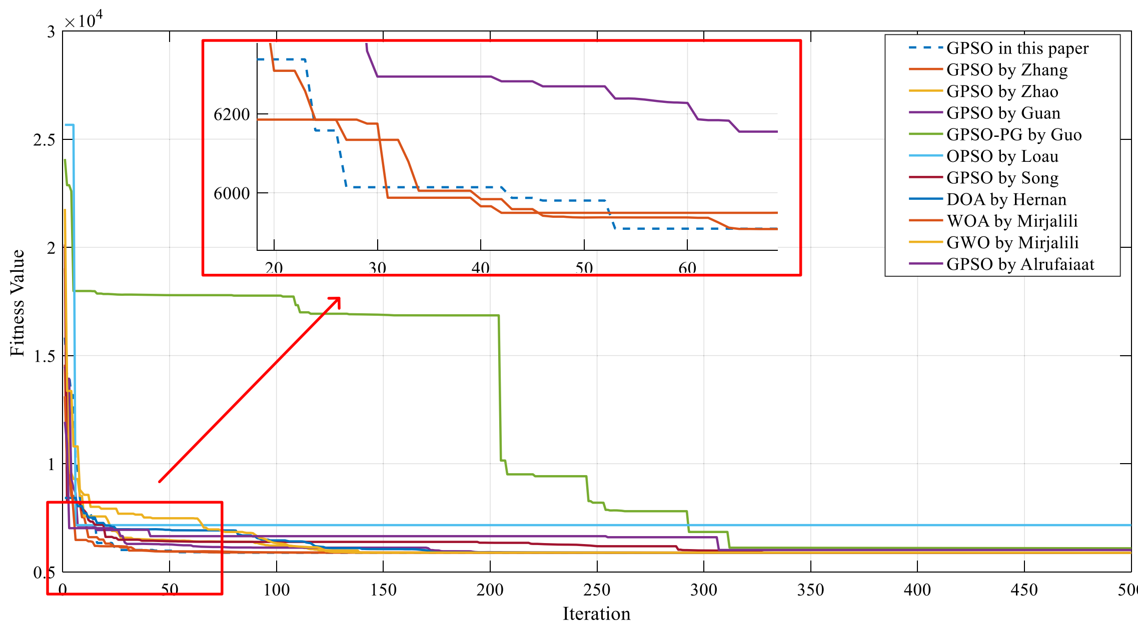

Fig. 4 The convergence performance of each algorithm in solving pressure vessel design problem

In addition, Fig. 4 shows the convergence performance of each algorithm in solving pressure vessel design problem. It can be seen from Fig. 4 that the GPSO proposed in this paper has a higher convergence speed compared with other algorithms. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can get a better optimal value and has good performance in stability, convergence speed and time overhead.

4.2.2 Welded beam problem

The welded beam design problem is also a classic optimization problem. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress in the beam (θ), buckling load on the bar (P_c), the end deflection of the beam (δ), and side constraint; the design of welded beam is shown in Fig. 5. This problem has four variables that need to be optimized, namely the thickness of the welded layer h, the length of the welded layer l, the width of the beam t, and the thickness of the beam b. The objective function and constraints of the problem are expressed as follows:

$$x = [x_1, x_2, x_3, x_4] = [h, l, t, b] \quad (27)$$

$$\text{Min} f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \quad (28)$$

s.t.

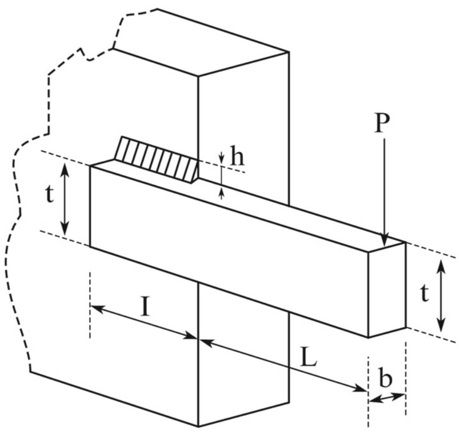

Fig. 5 The design of the welded beam

$$g_1(x) = \tau(x) - \tau_{max} \le 0 \quad (29)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \le 0 \quad (30)$$

$$g_3(x) = x_1 - x_4 \le 0 \quad (31)$$

$$g_4(x) = 0.125 - x_1 \le 0 \quad (32)$$

$$g_5(x) = \delta(x) - 0.25 \le 0 \quad (33)$$

$$g_6(x) = P - P_c(x) \le 0 \quad (34)$$

$$g_7(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \quad (35)$$

$$0.1 \le x_1, x_4 \le 2 \quad (36)$$

$$0.1 \le x_2, x_3 \le 10 \quad (37)$$

where $\tau(x)$ is the shear stress in the weld, τ_{max} is the allowable shear stress of the weld (= 13600 psi), $\sigma(x)$ is the normal stress in the beam, σ_{max} is the allowable normal stress for the beam material (= 30000 psi), $P_c(x)$ is the bar buckling load, P is the load (= 6000 lb), and $\delta(x)$ is the beam end deflection:

$$\tau(x) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2} \quad (38)$$

$$\tau_1 = \frac{P}{\sqrt{2}x_1x_2} \quad (39)$$

$$\tau_2 = \frac{MR}{J} \quad (40)$$

$$M = P\left(L + \frac{x_2}{2}\right) \quad (41)$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{2} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \quad (42)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (43)$$

$$\sigma(x) = \frac{6PL}{Ex_3^3x_4} \quad (44)$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \quad (45)$$

$$G = 12 \times 10^6 \text{ psi}, E = 30 \times 10^6 \text{ psi}, P = 6000 \text{ lb}, L = 14 \text{ in} \quad (46)$$

This paper uses this problem to test the performance of the proposed GPSO algorithm and compares it with several latest GPSO algorithms. The comparison results are shown in Tables 6 and 7.

In Table 6, the optimal value of GPSO proposed in this paper is ***1.7249***; this value is the same as the optimal value obtained by the existing algorithm, indicating that the algorithm can get a good optimal value. At this time, the parameter is: [x_1, x_2, x_3, x_4] = [0.2057, 3.4705, 9.0366, 0.2057], which is within the constraint range of the parameter and meets the parameter requirements. The constraint result obtained by substituting the parameters into the constraint equations is: [g_1, g_2, g_3, g_4, g_5, g_6, g_7] = [0, − 7.2760e−12, − 2.2204e−16, − 3.4330, − 0.0807, − 0.2355, 0], which also satisfies the constraint condition. Through comparative analysis, it is proved that the optimization algorithm proposed in this paper can obtain a better optimal value to welded beam design problem.

Table 7 shows the algorithm performance of each algorithm in solving welded beam design problem. The optimal value, average value and median value of GPSO proposed in this paper are all ***1.7249***, which are the same as the optimal value obtained by the existing algorithm, indicating that the algorithm can get a good optimal value and have a good stability. The standard deviation is ***6.6979e−6***, which is smaller than the minimum deviation of 3.4796E−05 in the above algorithm, indicating that the stability of the algorithm is better. The worst value of the optimal value obtained from 100 experiments is ***1.7251***, which is smaller than the minimum worst value of 1.7255 in the above algorithm, indicating better accuracy of the algorithm. The time of 100 experimental results is only ***5.1190*** s. Compared with the shortest time of 5.2413 s in the above algorithm, the time can be shortened by 2.3%, indicating that the algorithm has high optimization speed and low time complexity. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can obtain better optimal value and has good performance in stability and efficiency.

In addition, Fig. 6 shows the convergence performance of each algorithm in solving welded beam design problem. It can be seen from Fig. 6 that the GPSO proposed in this paper has a higher convergence speed compared with other

Table 6 Comparison of the optimal solution for welded beam design problem found by different algorithms

Method	Variables				$f(x)$
	x_1	x_2	x_3	x_4	
GPSO by Zhang (2021)	0.2057	3.4705	9.0366	0.2057	1.7249
GPSO by Zhao et al. (2019)	0.2057	3.4705	9.0366	0.2057	1.7249
GPSO by Guan et al. (2019)	0.2057	3.4705	9.0366	0.2057	1.7249
GPSO-PG by Guo et al. (2018)	0.1985	3.6784	8.9249	0.2111	1.7628
OPSO by Al-Bahrani and Patra (2018)	0.2006	3.5166	9.8080	0.2108	1.8984
GPSO by Song et al. (2018)	0.2057	3.4705	9.0366	0.2057	1.7249
DOA by Hernan et al. (2021)	0.2057	3.4705	9.0366	0.2057	1.7249
WOA by Mirjalili and Lewis (2016)	0.2057	3.4705	9.0366	0.2057	1.7249
GWO by Mirjalili et al. (2014)	0.2057	3.4705	9.0366	0.2057	1.7249
GPSO by Alrufaiaat and Althahab (2021)	0.2035	3.5075	9.0913	0.2055	1.7338
GPSO in this paper	0.2057	3.4705	9.0366	0.2057	***1.7249***

Bolditalics values indicated by optimal valve

Table 7 Performance of different algorithms for welded beam design problem

Method	Best	Mean	Median	Std	Worst	Time
GPSO by Zhang (2021)	1.7249	1.7249	1.7249	3.4796E−05	1.7255	8.375
GPSO by Zhao (2019)	1.7249	1.7252	1.7249	1.2859E−03	1.7365	5.2817
GPSO by Guan (2019)	1.7249	1.725	1.7265	3.6884E−04	1.7272	5.3727
GPSO-PG by Guo (2018)	1.7628	2.4487	2.6693	5.3189E−01	4.1438	5.2413
OPSO by Al-Bahrani and Patra (2018)	1.8984	2.8782	2.6013	4.4612E−01	3.925	8.7204
GPSO by Song et al. (2018)	1.7249	1.7251	1.7249	9.5686E−04	1.7335	5.9953
DOA by Hernan et al. (2021)	1.7249	1.725	1.7249	1.0065E−03	1.735	5.2648
WOA by Mirjalili and Lewis (2016)	1.7249	1.7249	1.7249	2.8723E−04	1.7277	5.2432
GWO by Mirjalili et al. (2014)	1.7249	1.7279	1.7249	9.5964E−03	1.8108	5.3352
GPSO by Alrufaiaat and Althahab (2021)	1.7338	2.2772	2.0483	3.3751E−01	3.2246	7.5093
GPSO in this paper	***1.7249***	***1.7249***	***1.7249***	***6.6979e−6***	***1.7251***	***5.1190***

Bolditalics values indicated by optimal valve

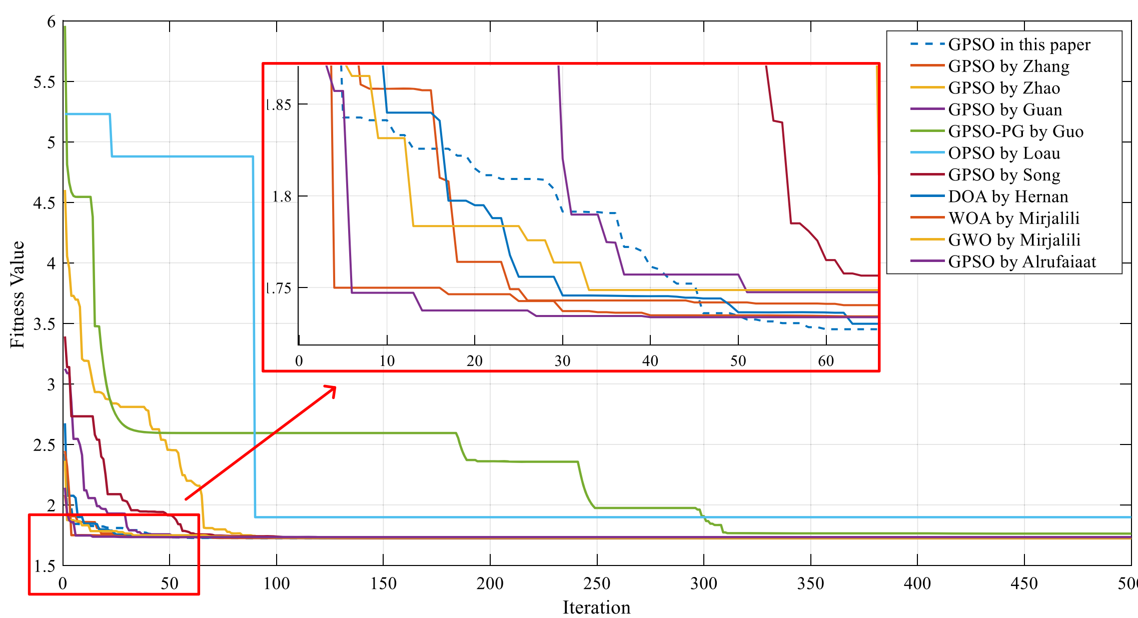

Fig. 6 The convergence performance of each algorithm in solving welded beam design problem

Table 8 Comparison of the optimal solution for gear train design problem found by different algorithms

Method	Variables				$f(x)$
	x_1	x_2	x_3	x_4	
GPSO by Zhang (2021)	16	19	49	43	2.7009E−12
GPSO by Zhao et al. (2019)	16	19	43	49	2.7009E−12
GPSO by Guan et al. (2019)	16	19	43	49	2.7009E−12
GPSO-PG by Guo et al. (2018)	19	16	49	43	2.7009E−12
OPSO by Al-Bahrani and Patra (2018)	26	15	53	51	2.3078E−11
GPSO by Song et al. (2018)	16	19	43	49	2.7009E−12
DOA by Hernan et al. (2021)	19	16	43	49	2.7009E−12
WOA by Mirjalili and Lewis (2016)	16	19	49	43	2.7009E−12
GWO by Mirjalili et al. (2014)	19	16	49	43	2.7009E−12
GPSO by Alrufaiaat and Althahab (2021)	26	15	51	53	2.3078E−11
GPSO in this paper	19	16	43	49	**2.7009E−12**

Bold values indicated by optimal valve

Table 9 Performance of different algorithms for gear train design problem

Method	Best	Mean	Median	Std	Worst	Time
GPSO by Zhang (2021)	2.7009E−12	5.5781E−10	2.7009E−12	8.7412E−10	4.5033E−09	6.8976
GPSO by Zhao (2019)	2.7009E−12	9.0259E−10	1.1661E−10	1.9240E−09	1.3125E−08	2.4033
GPSO by Guan (2019)	2.7009E−12	1.9522E−10	8.8876E−10	4.1787E−10	2.3576E−09	2.6563
GPSO-PG by Guo (2018)	2.7009E−12	9.3414E−09	3.3667E−08	1.6663E−08	1.1555E−07	2.3780
OPSO by Al-Bahrani and Patra (2018)	2.3078E−11	9.5187E−07	2.7265E−08	1.5744E−06	9.8207E−06	3.3740
GPSO by Song et al. (2018)	2.7009E−12	5.4215E−10	2.7009E−12	9.1924E−10	4.5033E−09	3.1630
DOA by Hernan et al. (2021)	2.7009E−12	3.6887E−10	1.1661E−10	7.2821E−10	2.3576E−09	2.6096
WOA by Mirjalili and Lewis (2016)	2.7009E−12	4.1501E−10	2.3078E−11	7.5615E−10	2.3576E−09	2.6874
GWO by Mirjalili et al. (2014)	2.7009E−12	4.5905E−10	2.3078E−11	1.2605E−09	1.1173E−08	2.5031
GPSO by Alrufaiaat and Althahab (2021)	2.3078E−11	1.8633E−07	2.3576E−09	3.5199E−07	2.0226E−06	6.0757
GPSO in this paper	***2.7009E−12***	***5.1431E−11***	***2.7009E−12***	***8.8063E−11***	***2.3576E−09***	***2.3258***

Bolditalics values indicated by optimal valve

algorithms. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can get a better optimal value and has good performance in stability, convergence speed and time overhead.

4.2.3 Gear train problem

Gear train design problem is also a common optimization problem in engineering. The optimization goal is to make the gear transmission ratio reach or close to the given transmission ratio by optimizing the number of gear teeth. The optimized variable is the number of teeth of 4 gears (gear range [12, 60]). The objective function and constraints of the problem are expressed as follows:

$$x = [x_1, x_2, x_3, x_4] = [T_1, T_2, T_3, T_4] \quad (47)$$

$$\text{Min} f(x) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2 \quad (48)$$

s.t.

$$12 \le x_1, x_2, x_3, x_4 \le 60 \quad (49)$$

$$x_i \in Z^+ \quad (50)$$

This paper uses this problem to test the performance of the proposed GPSO algorithm and compares it with several latest GPSO algorithms. The comparison results are shown in Tables 8 and 9.

In Table 8, the result of using the GPSO proposed in this paper to optimize the problem is: 2.7009e−12, the parameter is: [x_1, x_2, x_3, x_4] = [19, 16, 43, 49]. Since the parameters of this problem are finite integers, the optimal value is known. The significance of solving this optimization problem is to compare the advantages and disadvantages of the algorithms by comparing the performance of the algorithms.

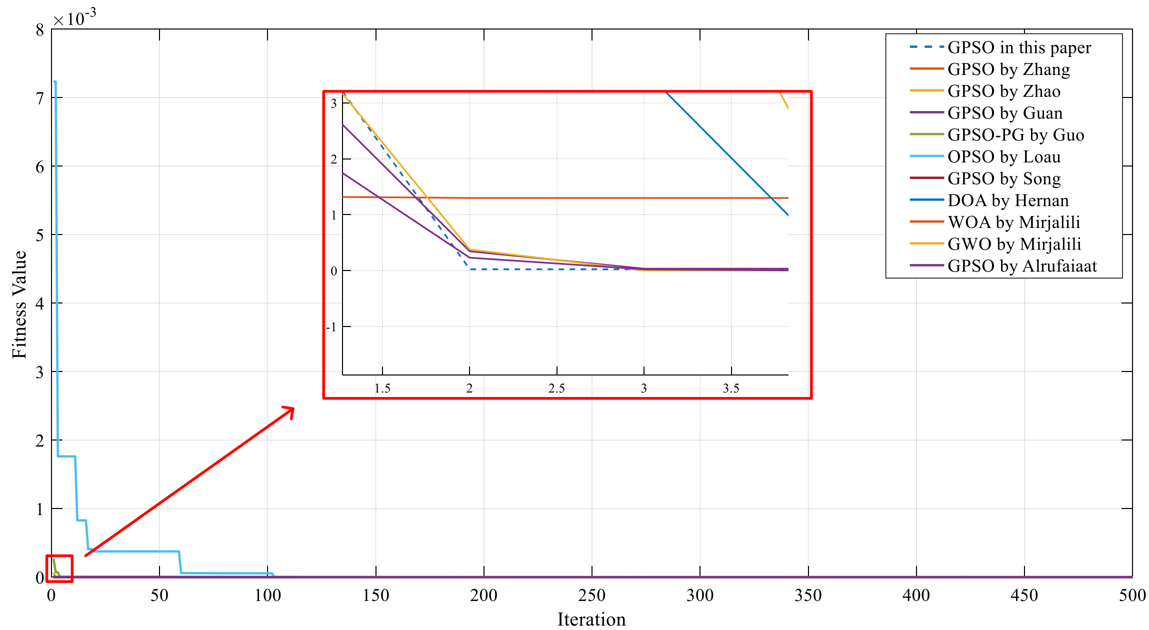

Fig. 7 The convergence performance of each algorithm in solving gear train design problem

Table 9 shows the algorithm performance of each algorithm in solving gear train design problem. The optimal value and median value of GPSO proposed in this paper are all ***2.7009E−12***, which are the same as the optimal value obtained by the existing algorithm, the average value is ***5.1431E−11*** which is smaller than the minimum optimal value of 1.9522E−10 in the above algorithm, indicating that the algorithm can get a good optimal value and have a good stability. The standard deviation is ***8.8063E−11***, which is smaller than the minimum deviation of 4.1787E−10 in the above algorithm, indicating that the stability of the algorithm is better. The worst value of the optimal value obtained from 100 experiments is ***2.3576E−09***, which is the same as the optimal value obtained by the existing algorithm, indicating good accuracy of the algorithm. The time of 100 experimental results is only ***2.3258*** s. Compared with the shortest time of 2.3780 s in the above algorithm, the time can be shortened by 2.2%, indicating that the algorithm has high optimization speed and low time complexity. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can obtain better optimal value and has good performance in stability and efficiency.

In addition, Fig. 7 shows the convergence performance of each algorithm in solving gear train design problem. It can be seen from Fig. 7 that the GPSO proposed in this paper has a higher convergence speed compared with other algorithms. Through the comparative analysis of the above performance, it is proved that the GPSO proposed in this paper can get a better optimal value and has good performance in stability, convergence speed and time overhead.

In summary, through the comparative analysis of the results of the classical nonlinear multi-constrained optimization problem and the classical multi-constrained optimization problems in practical engineering, GPSO proposed in this paper has obvious advantages over other GPSO algorithms in solving the multi-constrained optimization problem in terms of accuracy, stability, convergence speed and time overhead.

5 Conclusion

In this paper, a parallel GPSO algorithm is proposed based on the high efficiency of PSO and the global optimization ability of GA. This algorithm takes PSO as the main body, only when gbest has not changed for many generations, GA is enabled to replace PSO to update the particles and help the particles jump out of the local optimum. Since PSO has memory, GA does not destroy the found optimal value. In addition, this paper compares the performance of GPSO algorithm with several newly proposed algorithms on several multi-constrained optimization problems (Himmelblau's nonlinear optimization problem, pressure vessel design problem, welded beam design problem and gear train design problem) and evaluates the performance of the proposed algorithms with optimal value, average value, median value, worst value, standard deviation, time overhead and convergence speed. The results show that the GPSO proposed in this paper can get a better optimal value and has good performance in stability, convergence speed and time overhead. It shows that GPSO can ensure the effectiveness of solving multi-constrained optimization problems, ensure the accuracy of optimization results and the high speed of convergence.

This algorithm is only a basic version of the parallel GPSO algorithm. In fact, due to the high compatibility of the GPSO algorithm, it can continue to integrate other latest algorithms to improve the optimization performance and speed of the algorithm. In the future, the algorithm will continue to be developed and studied to improve its performance and speed, and the algorithm will be applied to various complex engineering optimization design to solve various problems in practice.

Author contributions Each author has equally contributed in conceptualization, model building, calculation and writing of the article.

Funding This work is supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 51521003), Heilongjiang Province Postdoctoral Research Startup Fund (No. LBH-Q18059).

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants nor any studies with animals, performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study

References

Abbassi A, Ben Mehrez R, Bensalem Y, Abbassi R, Kchaou M, Jemli M, Abualigah L, Altalhi M (2022) Improved arithmetic optimization algorithm for parameters extraction of photovoltaic solar cell single-diode model. Arab J Sci Eng. https://doi.org/10.1007/s13369-022-06605-y

Abd Elaziz M, Almodfer R, Ahmadianfar I, Ibrahim IA, Mudhsh M, Abualigah L, Lu S, Abd El-Latif AA, Yousri D (2022) Static models for implementing photovoltaic panels characteristics under various environmental conditions using improved gradient-based optimizer. Sustain Energy Technol Assess 52:102150. https://doi.org/10.1016/j.seta.2022.102150

Abdelhalim A, Nakata K, Alem ME, Eltawil A (2019) A hybrid evolutionary-simplex search method to solve nonlinear constrained optimization problems. Soft Comput 23(22):12001–12015. https://doi.org/10.1007/s00500-019-03756-3

Al-Bahrani LT, Patra JC (2018) A novel orthogonal PSO algorithm based on orthogonal diagonalization. Swarm Evol Comput 40:1–23. https://doi.org/10.1016/j.swevo.2017.12.004

Allawi HM, Al Manaseer W, Al Shraideh M (2020) A greedy particle swarm optimization (GPSO) algorithm for testing real-world smart card applications. Int J Softw Tools Technol Transf 22(2):183–194. https://doi.org/10.1007/s10009-018-00506-y

Al-qaness MAA, Ewees AA, Fan H, Abualigah L, Elaziz MA (2022) Boosted ANFIS model using augmented marine predator algorithm with mutation operators for wind power forecasting. Appl Energy 314:118851. https://doi.org/10.1016/j.apenergy.2022.118851

Alrufaiaat SAK, Althahab AQJ (2021) Robust decoding strategy of MIMO-STBC using one source Kurtosis based GPSO algorithm. J Ambient Intell Hum Comput 12(2):1967–1980. https://doi.org/10.1007/s12652-020-02288-1

Aravinth SS, Senthilkumar J, Mohanraj V, Suresh Y (2021) A hybrid swarm intelligence based optimization approach for solving minimum exposure problem in wireless sensor networks. Concurr Comput Pract E. https://doi.org/10.1002/cpe.5370

Chen CH, Li CL (2021) Process synthesis and design problems based on a global particle swarm optimization algorithm. IEEE Access 9:7723–7731. https://doi.org/10.1109/ACCESS.2021.3049175

Coello CAC (2000) Use of a self -adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127. https://doi.org/10.1016/S0166-3615(99)00046-9

Coello CAC, Montes EM (2002) Constraint- handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 16:193–203. https://doi.org/10.1016/S1474-0346(02)00011-3

Dimopoulos GG (2006) Mixed-variable engineering optimization based on evolutionary and social metaphors. Comput Methods Appl Mech Eng 196(4):803–817. https://doi.org/10.1016/j.cma.2006.06.010

Ekinci S, Izci D, Al Nasar MR, Abu Zitar R, Abualigah L (2022) Logarithmic spiral search based arithmetic optimization algorithm with selective mechanism and its application to functional electrical stimulation system control. Soft Comput. https://doi.org/10.1007/s00500-022-07068-x

Gao ZK, Li YL, Yang YX, Wang XM, Dong N, Chiang H-D (2020) A GPSO-optimized convolutional neural networks for EEG-based emotion recognition. Neurocomputing 380:225–235. https://doi.org/10.1016/j.neucom.2019.10.096

Garg H (2014) Solving structural engineering design optimization problems using an artificial bee colony algorithm. J Ind Manag Optim 10(3):777–794. https://doi.org/10.3934/jimo.2014.10.777

Garg H (2016) A hybrid PSO-GA algorithm for constrained optimization problems. Appl Math Comput 274:292–305. https://doi.org/10.1016/j.amc.2015.11.001

Guan JS, Hong SJ, Kang SB, Zeng Y, Sun Y, Lin C-M (2019) Robust adaptive recurrent cerebellar model neural network for non-linear system based on GPSO. Front Neurosci Switz 13:390. https://doi.org/10.3389/fnins.2019.00390

Guo WA, Si CY, Xue Y, Mao YD, Wang L, Wu QD (2018) A grouping particle swarm optimizer with personal-best-position guidance for large scale optimization. IEEE ACM Trans Comput Biol Bioinform 15(6):1904–1915. https://doi.org/10.1109/TCBB.2017.2701367

He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99. https://doi.org/10.1016/j.engappai.2006.03.003

He S, Prempain E, Wu QH (2004) An improved particle swarm optimizer for mechanical design optimization problems. Eng Optim 36(5):585–605. https://doi.org/10.1080/03052150410001704854

Hernan PV, Adrian FPD, Gustavo EC et al (2021) A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. Math Probl Eng 202:11–19. https://doi.org/10.1155/2021/9107547

Jamei M, Karbasi M, Mosharaf-Dehkordi M, Adewale Olumegbon I, Abualigah L, Said Z, Asadi A (2022) Estimating the density of hybrid nanofluids for thermal energy application: application of non-parametric and evolutionary polynomial regression data-intelligent techniques. Measurement 189:110524. https://doi.org/10.1016/j.measurement.2021.110524

Kharrich M, Abualigah L, Kamel S, AbdEl-Sattar H, Tostado-Véliz M (2022) An improved arithmetic optimization algorithm for design of a microgrid with energy storage system: Case study of El Kharga Oasis. Egypt J Energy Storage 51:104343. https://doi.org/10.1016/j.est.2022.104343

Liu Y, Mu CH, Kou WD, Liu J (2015) Modified particle swarm optimization-based multilevel thresholding for image segmentation. Soft Comput 19(5):1311–1327. https://doi.org/10.1007/s00500-014-1345-2

Mir M, Dayyani M, Sutikno T, Mohammadi Zanjireh M, Razmjooy N (2020) Employing a Gaussian particle swarm optimization method for tuning multi input multi output-fuzzy system as an integrated controller of a micro-grid with stability analysis. Comput Intell-US 36(1):225–258. https://doi.org/10.1111/coin.12257

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Rahman IU, Zakarya M, Raza M, Khan R (2020) An n-state switching PSO algorithm for scalable optimization. Soft Comput (Prepublish). https://doi.org/10.1007/s00500-020-05069-2

Salaria UA, Menhas MI, Manzoor S (2021) Quasi oppositional population based global particle swarm optimizer with inertial weights (QPGPSO-W) for solving economic load dispatch problem. IEEE Access 9:134081–134095. https://doi.org/10.1109/ACCESS.2021.3116066

Sheng ZQ, Li YH, Shi SS (2021) Multi-objective robust optimization of EMU brake module. In: 2021 IEEE 24th international conference on computer supported cooperative work in design (CSCWD) pp 702–707. https://doi.org/10.1109/CSCWD49262.2021.9437623

Song MX, Chen K, Wang J (2018) Three-dimensional wind turbine positioning using Gaussian particle swarm optimization with differential evolution. J Wind Eng Ind Aerodyn 172:317–324. https://doi.org/10.1016/j.jweia.2017.10.032

Su SB, Zhao W, Wang CS (2021) Parallel swarm intelligent motion planning with energy-balanced for multirobot in obstacle environment. Wirel Commun Mob Comput 2021:1–16. https://doi.org/10.1155/2021/8902328

Turgut MS, Turgut OE, Abualigah L (2022) Chaotic quasi-oppositional arithmetic optimization algorithm for thermo-economic design of a shell and tube condenser running with different refrigerant mixture pairs. Neural Comput Appl 34(10):8103–8135. https://doi.org/10.1007/s00521-022-06899-x

Zhang ZM (2021) Abnormal detection of pumping unit bearing based on extension theory. IEEJ Trans Electr Electron 16(12):1647–1652. https://doi.org/10.1002/tee.23468

Zhang WY, Zhang SX, Zhang S, Huang NN (2019) A novel method based on FTS with both GA-FCM and multifactor BPNN for stock forecasting. Soft Comput 23(16):6979–6994. https://doi.org/10.1007/s00500-018-3335-2

Zhao XR, Zhou YR, Xiang Y (2019) A grouping particle swarm optimizer. Appl Intell 49(8):2862–2873. https://doi.org/10.1007/s10489-019-01409-4