



Bilevel-search particle swarm optimization algorithm for solving LSGO problems

Yanan Wang¹ · Zhen Lei² · Jiekang Wu²

Received: 22 November 2022 / Accepted: 19 September 2023 / Published online: 23 October 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Improving the efficiency of solving complex optimization problems is the focus of intelligent algorithm research in recent years. Complex optimization problems have the characteristics of high dimension, multi-minimum and variable coupling. Using intelligent algorithms to solve such problems is prone to premature convergence, high-dimensional invalid and excessive computational costs, resulting in low optimization efficiency of complex problems. In this paper, a double-layer search particle swarm optimization algorithm is proposed. The algorithm derives the sub-objective function by grouping the particle population and variable dimension, constructs the bilevel-layer search space, and defines the dynamic parameter matrix to realize the information interaction and space reconstruction of the sub-space. For the optimization problem with complex characteristics, bilevel-search can realize the alternation of fine and rough search to reduce the computational cost and improve the optimization efficiency. The estimation program and test samples of the proposed method are tested, and compared with other intelligent algorithms to further prove the advantages of this method.

Keywords Complex optimization problem · Particle swarm optimization · Bilevel-search space · Dynamic matrix · Optimize efficiency · Computational cost

1 Introduction

In the process of innovative and green development in industrial design, numerous technical problems requiring optimization and innovation have emerged incessantly (Sellali et al. 2022; Soheyli et al. 2016; Sun et al. 2011). With the continuous expansion of industrial systems and the ongoing enhancement of technical requirements, these technical issues are exhibiting a trend of large-scale and complex development. The optimization of industrial development gives rise to a category of complex optimization problems characterized by large scale, multipolarity, variable coupling, and a complex structure of the objective function (Tey and Mekhilef 2014; Zhang et al. 2010). Typically, these problems establish solution models based on intelligent algorithms.

For complex optimization problems exhibiting multiple extremum characteristics, swarm intelligence algorithms are prone to premature convergence caused by individuals getting trapped in local optima. This significantly restricts the algorithm's solution performance for such problems and leads to low optimization efficiency (Wah and Ma 2006; Azar et al. 2011; Andre and Siarry 2001). In the case of optimization problems with large-scale characteristics, the increase in problem complexity results in the phenomenon known as 'dimension disaster' in variable dimensions (Kerner 1989; Xiang et al. 2006; Li and Xin 2012). This renders conventional optimization algorithms ineffective. Particularly, when large-scale optimization problems possess variable coupling characteristics, solving them becomes exceedingly complex (Dutta and Gandomi 2020).

The particle swarm optimization (PSO) algorithm is an iterative optimization method and widely employed as a mainstream approach for solving complex optimization problems. When compared to other algorithms such as genetic algorithms and simulated annealing, PSO algorithm demonstrates certain advantages in terms of convergence, diversity, and consistency (Yeh et al. 2010; Lin and Tang 2021; Fu-Shiung et al. 2019; Yan et al. 2021). In the study of

✉ Yanan Wang
y1355584199@126.com

¹ School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523106, China

² School of Automation, Guangdong University of Technology, Guangzhou 510006, China

asset markets, William R. introduced the concept of 'asymptotic inefficiency,' which refers to the scenario where the limited portfolio and return cannot align or approximate, indicating inefficiency in the long run (Zame 1989). In the context of global optimization problems, the phenomenon of 'gradual ineffectiveness' implies that the algorithm can approach the optimal solution with a low rate of progress, but achieving higher accuracy requires more computational resources (Liu and Zeng 2015; Liu et al. 2015, 2017; Liu et al. 2014).

In the process of calculating complex optimization problems, our objective is to find or approach the optimal solution while minimizing the computational cost to improve the efficiency of optimization. However, in actual calculations, as we strive for higher accuracy, the associated computational costs also increase. This phenomenon is referred to as 'Asymptotic Inefficiency' (Liu 2011; Liu et al. 2016, 2017; Zhao et al. 2015). When particle updating narrows down the range of the optimal solution, i.e., enhances the calculation accuracy, the inevitable consequence is an increase in computational cost. Therefore, our aim should be to minimize the impact of "Asymptotic Inefficiency".

When utilizing particle swarm optimization for various optimization problems, the manifestation of 'Asymptotic Inefficiency' can differ. Particularly, when addressing simple optimization problems, the impact of this phenomenon may be negligible. This paper, however, focuses exclusively on complex optimization problems. It compares the 'Asymptotic Inefficiency' exhibited by different particle swarm optimization algorithms and validates the performance of the bilevel-search particle swarm optimization algorithm (Zame 1993; Skevas and Skevas 2021; Tran et al. 2016).

The rest of this paper is organized as follows. In Sect. 2, the characteristics and influence of complex optimization problems are analyzed, especially the phenomenon of "premature convergence". In Sect. 3, we provide a detailed introduction to the principles of the bilevel-search particle swarm optimization algorithm. In Sect. 4, presents an empirical assessment of the optimization performance of the bilevel-search particle swarm optimization algorithm through example calculations and a comparison with three popular particle swarm optimization algorithms. In Sect. 5, we summarize the research findings and outline potential future research directions.

2 Background

Optimization problems are pervasive in various technical domains of social production (Abdullah and Hassan 2020; Kim et al. 2013). The key technical elements for solving diverse optimization problems lie in scientific mathematical models and intelligent solution algorithms. However,

specific models and algorithms do not uniformly yield significant effects on solving all problems. The effectiveness of optimization problem solutions is heavily influenced by the inherent characteristics of the problem itself. Factors such as the dimensionality of the objective function, the quantity and distribution of local extremum points, and variable coupling often contribute to the complexity of problem-solving (Lui and Teodorovi 2011; Dereli and Kker 2021). This chapter analyzes the characteristics of complex optimization problems, including their large-scale nature, multipolarity, and variable coupling. Furthermore, it investigates the impact of these characteristics on optimization efficiency.

2.1 Complex optimization problems with large-scale characteristics

Taking single objective unconstrained optimization as an example, search for the global optimal solution of the objective function within the solution space can be formulated as Eq. (1):

$$\min f(x) \text{ s.t. } x \in \Omega \subseteq R^n \quad (1)$$

where, f is the objective function; Ω is the solution space of the problem; n is the dimension value of the optimization vector x .

The dimensionality of the solution space Ω is directly determined by the number of optimization variables x . Real-world optimization problems, such as large-scale power system control optimization (Cagliari et al. 2021) and large-scale distribution route optimization (Wu 2014; Economics and Wisconsin 1996), often involve a significant number of decision variables. As the dimension of the objective function increases, it becomes increasingly challenging to find feasible solutions, regardless of the computational cost involved.

As shown in Fig. 1, PSO algorithm is used for Sphere-function experiments with different dimensions. The experimental results show that the higher the dimension, the less ideal the calculation effect is. After the function reaches 50 dimensions, even if it consumes more calculation resource, it can't give a feasible solution to the problem, resulting in "negative efficiency" (Ngango and Hong 2022).

2.2 Complex optimization problems with multi extremum characteristics

Multi extremum characteristic means that in addition to the global optimal value, a large number of local extremum points are distributed in the solution space, such as the Griewank function of Eq. (2). The optimization problem of multi pole points is easy to cause "premature convergence"

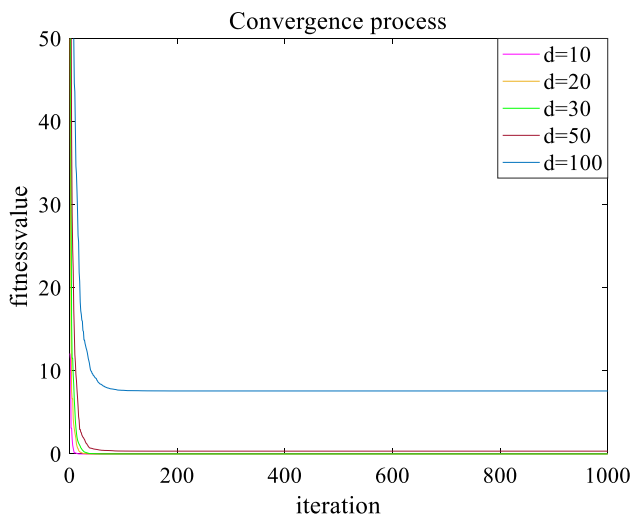


Fig. 1 Evolution curve of sphere-function in different dimensions

(El-Abd and Mohammed 2009) and make the algorithm stop evolution too early.

$$f = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, x \in [-600, 600]^D \tag{2}$$

After PSO algorithm solves the five-dimensional multimodal function Griewank, the top 10 particles with the best fitness are selected, and their positions are shown in Table 1. It can be seen that all particles cannot get rid of the local extreme value and stay away from the global optimal solution. Table 2 shows the particle motion speed. The top 10 particles with the fastest progress are selected to observe their motion speed. It is found that the particle speed is close to 0, that is, even if more computing resources are consumed, the particles do not have enough power to get rid of the local dilemma for multi extreme and multi-dimensional problems.

Table 1 Position of the top 10 elite particles

P	d1	d2	d3	d4	d5
x ₁	4.449	-1.089	1.164	-2.269	-2.035
x ₂	1.117	-2.084	2.741	-2.879	-2.170
x ₃	2.613	-4.210	4.009	6.558	2.532
x ₄	5.425	6.166	-1.503	-6.632	1.119
x ₅	-1.808	3.758	-1.496	3.663	1.119
x ₆	-1.880	3.758	-1.496	3.663	-1.087
x ₇	-1.880	-2.758	-1.247	2.4957	1.9409
x ₈	8.464	1.2794	-1.382	1.798	-5.227
x ₉	-1.161	-1.183	7.026	-1.286	3.085
x ₁₀	-7.953	-1.445	-1.122	-5.964	9.972

Table 2 Position of the top 10 profiteer particles

P	d1	d2	d3	d4	d5
v ₁	0.0042	-0.0034	-0.0661	0.0010	-0.0239
v ₂	0.0014	-0.0045	-0.0327	0.0085	0.0137
v ₃	0.0047	-0.0111	-0.0458	-0.0061	0.0081
v ₄	0.0049	-0.0022	-0.0384	-0.0881	0.0821
v ₅	0.0053	0.0652	-0.0390	1.9101	0.3684
v ₆	0.0074	-0.0026	-0.0762	0.0012	0.0150
v ₇	0.0065	-0.0022	-0.0594	0.0064	0.0148
v ₈	0.0050	0.0356	-0.0885	-1.002	0.3553
v ₉	0.0030	-0.0081	-0.0971	0.0084	0.0264
v ₁₀	0.0045	-0.0132	-0.0477	0.0105	0.0127

2.3 Complex optimization problems with variable coupling characteristics

The optimization problem of variable coupling characteristics is also called indivisible problem, which means that some or all variables depend on each other, and the value of the objective function is affected by both the variable itself and the coupling variable. The common method to solve this kind of problem is to divide the large-scale coupling problem into small-scale uncoupled subproblems, and finally reconstruct the solution of the original problem by using the results of the subproblems.

The optimization problems encountered in engineering design often involve complex characteristics. The bilevel-search particle swarm optimization method proposed in this paper takes into account the coupling relationships among variables and aims to solve complex high-dimensional problems by reducing their dimensions, thereby improving optimization efficiency.

3 Methods

When employing the bilevel-search framework in the PSO algorithm to address large-scale complex optimization problems with coupling characteristics, several challenges need to be addressed: 1. Hierarchical handling of complex optimization problems; 2. Determination of sub-layer optimization objectives; 3. Construction of a high-level search subpopulation.

3.1 Hierarchical processing method

Hierarchical processing of large-scale complex optimization problems with variable coupling characteristics is to divide the high-dimensional space layer into multiple low-dimensional combined subspace layers to realize the splitting and dimensionality reduction of high-dimensional problems. This chapter

mainly studies the two-layer space, and the space with higher layers can be treated similarly.

when the high-dimensional solution space is divided into multiple low-dimensional subspaces, the variables within these subspaces remain coupled. In this case, we employ multiple reference vectors to facilitate information transfer between different subproblems. This approach helps to partially restore the variable coupling relationship that may be disrupted by the segmentation of the solution space. As a result, it reduces computational costs and enhances the optimization efficiency of complex optimization problems.

The objective function of the bilevel-search space can be expressed as:

$$\begin{aligned} \min f(x_n) &= \min [F_1(X_1), F_2(X_2), \dots, F_k(X_k)] \\ &= \min [F_1(x_1 \dots x_s), F_2(x_{s+1} \dots x_{2s}), \dots, F_i(x_{(i-1)s+1} \dots x_{is})] \end{aligned} \tag{3}$$

The sub-objective functions of the layer-2 of the double-layer search space can be expressed as:

$$\min F_i(X_i) = f(c_1^i, \dots, c_{(i-1)s}^i, x_{(i-1)s+1}^i \dots x_{is}^i, c_{is+1}^i \dots c_n^i) \tag{4}$$

$$\begin{cases} c^i = 0, & i = 1 \text{ or } i = k \\ c^i \neq 0, & 1 < i < k \end{cases} \tag{5}$$

where, s is the dimension of subspace, $s = d / k$. c is the reference value.

Assuming that the d -dimensional problem sets the population number as n , the state of each particle can be expressed by matrix M :

$$M = P \cdot X = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \cdot [x_1, x_2 \dots x_d] = \begin{bmatrix} m_{11} & \dots & m_{1d} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{nd} \end{bmatrix} \tag{6}$$

Through dimensionality reduction, the coupling variables are divided into k groups, $X = [X_1, X_2 \dots X_k]$, and the population is also divided into sub-populations of group k , $P = [P_1, P_2 \dots P_k]$, The population and variables are evenly distributed. The relationship between population and sub population can be expressed by formula (4). The hierarchical strategy is summarized in algorithm 1.

$$M = \begin{bmatrix} M_{s1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & M_{sK} \end{bmatrix} \tag{7}$$

where, s represents the number of variables in each group.

Algorithm 1: Bilevel-search hierarchical process

Input: population P and number of sub-spaces K .

Output: Building layer-2 search space

- 1 Population matrix of layer-1 $\leftarrow M = P \cdot X$
 - 2 Sub-population matrix of layer-2 $\leftarrow M_{sk} = P_k \cdot X_s$
-

3.2 Hierarchical optimization sub-objectives

Dividing the complex optimization problem into a combination of subproblems implies a bilevel search approach, where the search space k is divided into two levels. The first level search, known as the generalized or fine search, involves locking onto multiple search spaces and narrowing down the search scope. The second level search, referred to as the regional or rough search, focuses on performing local optimization within each subspace. The optimization results obtained from the regional search are then fed back to the first level area to obtain the optimal solution.

The first layer search is well-understood and follows a conventional multi-objective function solution method. However, the second layer search involves multiple sub-objective functions. Since these sub-objective functions are still coupled, the optimization process requires not only updating the particle positions but also adjusting the dimensionality of the subpopulations and variable combinations. The hierarchical structure is depicted in Fig. 2, while the optimization sub-objectives strategy is summarized in Algorithm 2.

Algorithm 2: optimization sub-objectives strategy

Input: Variables and objective functions

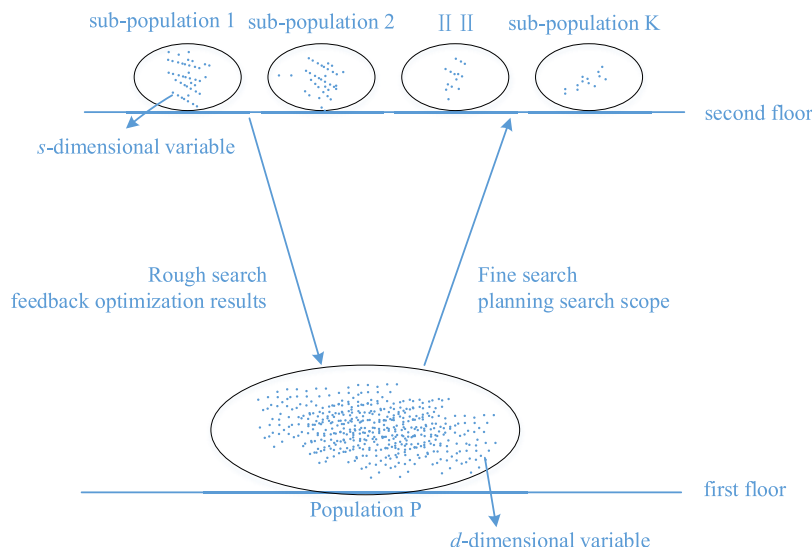
Output: sub-variables and sub-objective functions

- 1 Variable grouping $\leftarrow s = d / k$;
 K groups, each subspace containing s variables
 - 2 sub-objective function \leftarrow Combine the s variables
-

3.3 Population regeneration

When dealing with an uncoupled variable optimization problem, a reference vector Y is utilized to record and

Fig. 2 Bilevel-search space structure diagram



update the optimal solution for each subspace. The initial reference vector is obtained by combining any solution from each subspace and is recorded as $Y = [y_1, y_2 \dots y_k]$. Through the iteration of particles within each subspace, optimal solutions $y^* i(i = (1, 2 \dots k))$ are found. The high-dimensional optimal solution is represented as $Y = [y^* 1, y^* 2 \dots y^* k]$.

However, it is evident that a single set of reference vectors cannot accommodate high-dimensional problems with coupled variables. To address this, we introduce a reference matrix and update the worst vector within the matrix every few generations. From each sub-population in the second layer search, layer-1 elite particles and layer-2 profiteer particles are selected to form the sub-population reference matrix. The sub-population reference matrix of the second layer then becomes the overall reference matrix of the first layer. The optimal solution for the high-dimensional problem is chosen from this reference matrix. The reference matrix for iteration m can be expressed as:

$$C^m = \begin{bmatrix} C_{i,L_1}^m & \dots & C_{k,L_1}^m \\ C_{i,L_2}^m & \dots & C_{k,L_2}^m \end{bmatrix} \tag{8}$$

After several population iterations, the reference matrix is updated. First, update the reference matrix of the second layer search, then randomly combine the reference vectors of each group of matrices, and finally update the reference matrix of the first layer search. The optimal solution of

high-dimensional complex problems is obtained from the reference matrix. The population regeneration strategy is summarized in algorithm 3.

Algorithm 3: sub-population regeneration process

Input: sub-population after layer-1 allocation

Output: sub-population after n iterations

- 1 Initialize sub-population and reference matrix $\leftarrow M_{s1}, M_{s2}, \dots, M_{sk}, C^m;$
- 2 Update sub-population particles $\leftarrow \text{iter}=\text{iter}+1;$
sub-module of reference matrix \leftarrow select elite particles;
- 3 Update reference matrix $\leftarrow \text{iter}=\text{iter}+n;$

3.4 PSO based on bilevel-search framework algorithm flow

When dealing with high-dimensional complex optimization problems, PSO algorithm based on bilevel-search framework first divides the population P and space vector X , then searches hierarchically, and finally reconstructs the global optimal solution from the optimal solution of the subproblem. The algorithm flow is as follows:

Algorithm 4: BPSO

```

1  Initialization parameters:  $N, Kc$ 
2  search space of layer-1  $\rightarrow$  zbest, fitnessbest;
3  Select  $K_1$  group of elite particles to form a position
   reference matrix  $\rightarrow R_x$ ;
   Select  $K_1$  group of profiteer particles to form the velocity
   reference matrix  $\rightarrow R_v$ ;
4  Initialize layer-2 search space: Randomly divided into  $k$ 
   subspaces,
   subspace: variable dimension is  $s=D/k$ , fixed dimension is
    $m=D-s$ , population number  $sub\_p=N/k$ ;
5  Update layer-2 subspace particles  $\rightarrow$  sub $_i$  fitnessbest,
   sub $_i$  best; Regroup subspace  $\rightarrow$  fitnessbest; Update
    $R_x$ ;
6  iter= $n$ , Update  $R_v$ ;
7  Judging,  $Y^{m*}$  is better than  $Y^{(m-1)*}$ ;
   yes, update vector; no, continue 7;
7  Iteration  $q$  times, interactive subproblem vector
   information, update reference matrix  $C$ ;
8  Judgment, stop condition;
   yes, stop it; no, change  $K$ , return 2;

```

4 Numerical experiment

In this section, numerical simulation experiments are carried out to verify the performance of bilevel-search particle swarm optimization(BPSO) algorithm in solving LSGO problems, and the current fluid algorithms is compared to reveal the effectiveness of BPSO.

4.1 Selection of test function

For the LSGO problem, this paper selects 10 general LSGO test functions (Bergh and Engelbrecht 2004; Yang et al. 2008; Li and Yao 2012; Zhang and Sanderson 2009; Hedar 2005), and the relevant information of the test functions f_1 - f_{10} is shown in Table 3. Where, f_1 - f_5 are single peak functions and f_6 - f_{10} are multi peak functions. f_1, f_4 and f_6 are separable problems, and f_2, f_3, f_5 and f_7 are complex problems of variable coupling (Ke et al. 2009; Yu et al. 2013).

4.2 Influence of reference matrix on algorithm performance

For the same test function, under different reference matrix conditions, multiple independent operation results are shown in Table 4. List the BPSO- p (p is the order of the reference vector, $p=2,3,5,10,20$) results in turn.

The function dimensions of f_1, f_2 and f_4 in Table 4 are 30d, while the dimension of f_3 is 50d. As observed from Table 4, for the f_1 functions, increasing the reference vector improves the algorithm's performance, although the effect is not significant. On the other hand, for the f_4 functions, the algorithm's performance shows significant improvement with an increase in the order of the matrix, particularly with the 10th order matrix. Regarding the f_2 function, the impact of the reference matrix is not ideal, as it only offers moderate improvement while consuming substantial computational resources. For the f_3 function, which is 50-dimensional, increasing the order of the reference matrix enhances the algorithm's performance, albeit with slightly volatile data. Therefore, when dealing with high-dimensional functions, multiple experiments should be conducted to obtain average results.

Figure 3 shows the bilevel-search process of f_4 function, and set the number of sub populations to 5. It can be seen that after the large-scale fine search of the first layer, the search area can be quickly locked, and then enter the rough search of the second layer to further improve the search ability and find the optimal value more accurately.

Figure 4 illustrates the bilevel-search process applied to the f_3 function, with 3 subpopulations and p representing the order of the reference matrix. Through comparison, it is concluded that the search curves of the first layer are essentially the same. In the second layer search process, the p -10 search yields more accurate results, while the p -2 and p -3 searches exhibit similar performance.

In the realm of visualization, a comparison is made among the operational results of the f_1, f_3, f_4 functions, as depicted in Fig. 5, where the fitness values of the functions are displayed. The first 1000 iterations represent the first layer search process, while the last 1000 iterations represent the second layer search process. It is evident that the second layer search enhances the algorithm's accuracy.

Generally, increasing the order of the reference matrix can significantly improve the algorithm's performance.

Table 3 Function characteristics

Test function	Solution space	Theoretical optimal value	Dimension
$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100,100]^D$	0	30
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10,10]^D$	0	200
$f_3(x) = \max_i \{ x_i , 1 \leq i \leq 30\}$	$[-100,100]^D$	0	300
$f_4(x) = \sum_{i=1}^D (x_i + 0.5)^2$	$[-100,100]^D$	0	30
$f_5(x) = \sum_{i=1}^D ix_i^4 + random[0, 1)$	$[-1.28,1.28]^D$	0	500
$f_6(x) = -\sum_{i=1}^D x_i \sin\left(\left \sqrt{ x_i \right }\right)$	$[-500,500]^D$	-12,569.5	30
$f_7(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12,5.12]^D$	0	30
$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^D$	0	50
$f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^D$	0	50
$f_{10}(x) = \left[\frac{1}{500} + \sum_{i=1}^D \frac{1}{i + \sum_{j=1}^i (x_j - a_j)^6}\right]^{-1}$	$[-65.536,65.536]^D$	1	30

Table 4 Comparison of BPSO algorithms with different reference matrix orders

Function		BPSO-1	BPSO-2	BPSO-3	BPSO-5	BPSO-10
f_1	Variance	1.2135	0.9884	0.8854	0.5378	0.2336
	Expectation	0.5340	0.3224	0.0971	0.0771	0.0392
f_2	Variance	0.2867	0.2723	0.2503	0.1935	0.1905
	Expectation	0.0013	0.0020	0.0021	0.0009	0.0067
f_3	Variance	13.1412	10.4576	9.8573	8.0668	4.7939
	Expectation	1.2406	1.2745	1.1786	1.5808	1.7009
f_4	Variance	663.9219	110.7925	65.4422	19.3853	0.8622
	Expectation	45,572	1591.9	986.4549	284.8326	0.3248

However, excessively high orders will result in substantial computational resource consumption. Therefore, the following experiments will be conducted using a reference matrix of order 5.

4.3 Comparative analysis of experimental results

Compare the operation results of algorithms adaptive mutation particle swarm optimization (AMPSO), an improved backbone particle swarm optimization algorithm based

on learning strategy (Bare-bones PSO, BBPSO), adaptive weight adjustment AWPSO and adaptive mutation bilevel-search PSO (AM_BPSO) on functions $f_1 - f_{10}$. The results are shown in Table 5, including the average value, standard deviation and calculation time of fitness function.

For function f_1 , AM_BPSO algorithm has obvious advantages, accurate and stable calculation, but the calculation time is relatively long; For function f_2, f_3, f_7 and f_8 , the effect of AWPSO algorithm is similar to that of AM_BPSO algorithm, and even AWPSO algorithm is better. For functions f_4 ,

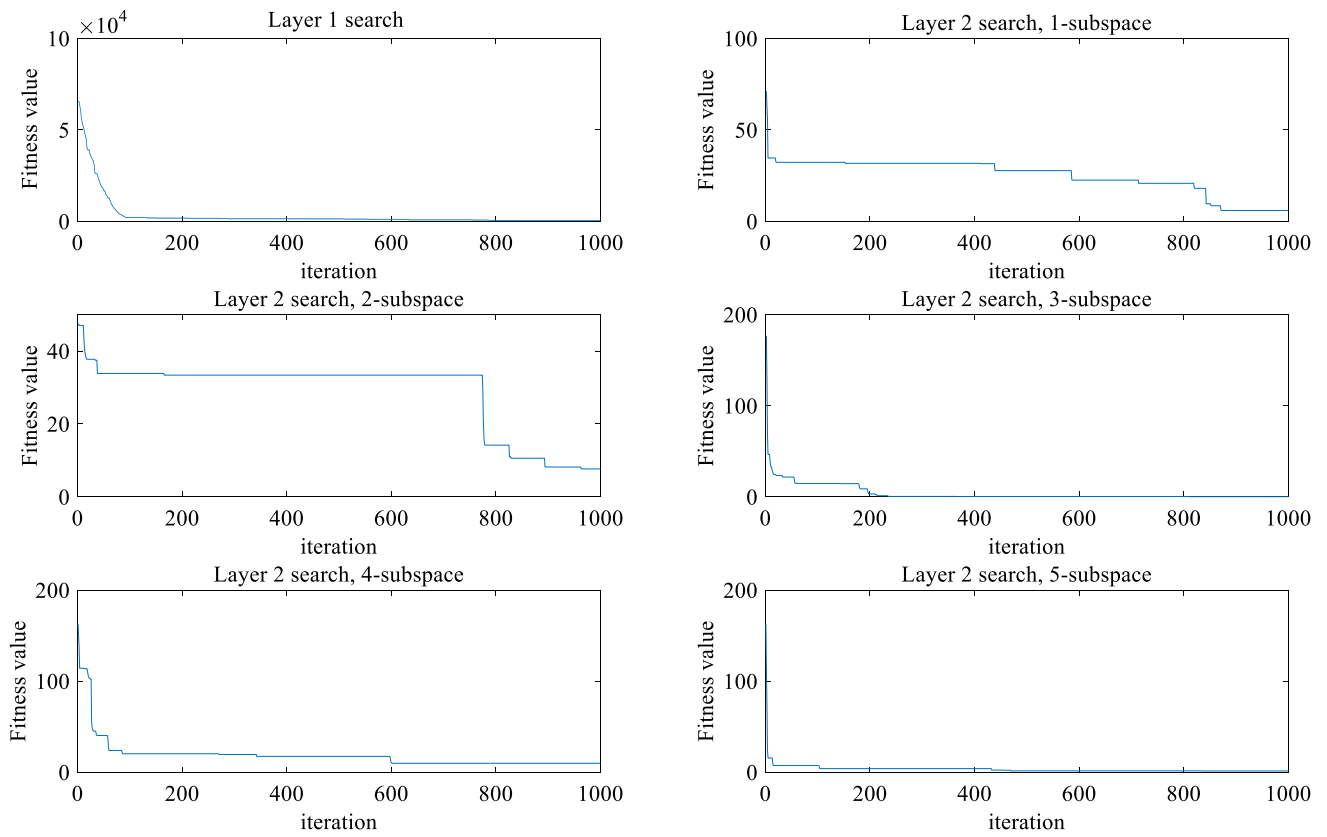


Fig. 3 Bilevel-search process of f_4 function

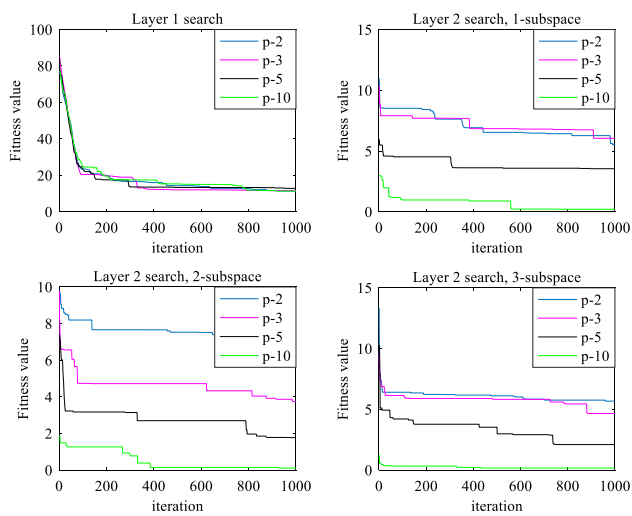


Fig. 4 bilevel-search process of f_3 function

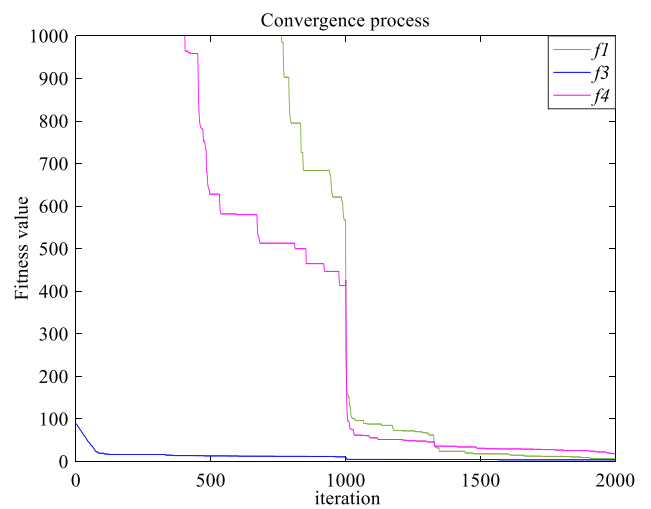


Fig. 5 Comparison chart of f_1, f_3, f_4 operation results

Table 5 Comparison of operation results of each method

Method		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
AMPSO	Mean	3700.7423	54.7673	41.0138	3267.551	11.4046	-3268.94	30.4804	5.2604	293.830	14.6226
	Std	913.2131	26.2802	3.5418	666.5678	1.91480	463.0304	5.7047	0.4805	20.8493	12.5951
	Time	0.1075	0.2506	0.3271	0.1131	0.8127	0.1588	0.1335	0.1655	1.4144	15.3795
BBPSO	Mean	52.0520	37.526	98.6596	62.6917	241.3108	-11,262.6	334.8293	11.1036	311.037	77.2873
	Std	14.4456	20.0351	0.8172	9.55175	12.5867	643.3748	29.8545	6.5610	0.0050	33.6178
	Time	0.4641	0.5746	0.6950	0.4647	1.0673	0.2819	0.2834	2.4168	4.8327	16.9560
AWPSO	Mean	701.4083	2.0070	21.8314	705.0734	0.6093	-3860.94	2.3523	0.3963	274.471	26.5763
	Std	115.1643	0.20710	1.3052	126.698	0.0646	357.6874	1.3214	0.2590	26.5541	24.5907
	Time	0.8404	1.9626	2.5404	0.8587	7.0440	1.1116	0.8846	1.0497	1.4121	15.3179
AM_BPSO	Mean	44.0831	3.92021	20.9776	47.9488	1.3150	-10,145	3.4589	0.2873	112.103	1.58661
	Std	24.1752	0.30900	1.4052	13.4614	0.4296	291.2264	2.1337	0.2631	19.4397	0.0280
	Time	0.8514	1.5382	1.9545	0.8933	6.2133	1.0237	0.8734	0.9873	1.2760	10.9572

f_5, f_6, f_9 and f_{10} , algorithm AM_BPSO has certain advantages in terms of calculation time and accuracy. Experimental results of f_4 function is shown in Fig. 6.

The bilevel-layer search particle swarm optimization algorithm don't have significant effects on all test functions. The above conclusions can be drawn that for some low-dimensional functions, the bilevel-search algorithm does not show advantages. For high-dimensional complex problems, the bilevel-search particle swarm optimization algorithm has obvious advantages. The bilevel-layer search particle swarm optimization algorithm has strong ability to solve complex problems, fast convergence speed and high calculation accuracy.

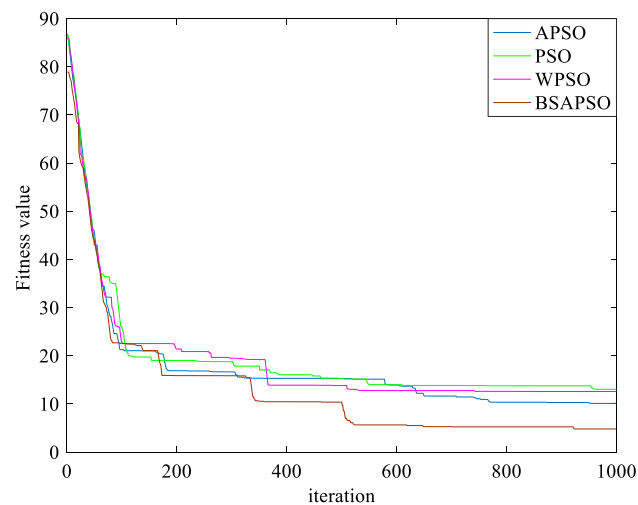


Fig. 6 Comparison of calculation process

Under the condition of ensuring the accuracy, stability and convergence speed of the algorithm, improve the accuracy and use the calculation cost as little as possible.

5 Discussion and conclusion

In this paper, we propose a particle swarm optimization algorithm based on a bilevel-search framework. The first-level search, known as fine search, efficiently narrows down the search space to identify extreme regions. The second-level search, referred to as rough search, further enhances the algorithm's accuracy.

During the second-level search, we introduce a reference matrix to record and update the optimal positions of particles within each subspace. This incorporation of the reference matrix significantly accelerates the convergence speed and improves the accuracy of the optimization process. Comparing our algorithm to other optimization algorithms, the double-layer search structure demonstrates significant advantages.

By utilizing the bilevel-search framework, our proposed algorithm establishes a particle swarm optimization approach that combines fine and rough searches. This strategy leads to faster convergence and more precise optimization positions, setting it apart from other optimization algorithms.

Funding This work is supported by the National Natural Science Foundation of China (Grant no. 51567002, 50767001), The National High Technology Research and Development of China (Grant no.

2007AA04Z197), Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20094501110002).

Data availability The datasets generated and analysed during the current study are not publicly available due the fact that they constitute an excerpt of research in progress but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest There are no known competing financial interests or personal relationships between authors that could have appeared to influence the work reported in this paper.

References

- Abdullah G, Hassan Z (2020) Using of particle swarm optimization (pso) to addressed reliability allocation of complex network. *J Phys: Conf Ser* 1664(1):1–12
- Andre J, Siarry P (2001) An improvement of the standard genetic algorithm ghting premature convergence in continuous optimization. *Adv Eng Softw* 32(1):49–60
- Azar FE, Perrier M, Srinivasan B (2011) A global optimization method based on multi-unit extremum-seeking for scalar nonlinear systems. *Comput Chem Eng* 35(3):456–463
- Bergh F, Engelbrecht A (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8(3):225–239
- Cagliari L, Hicken JE, Mishra S (2021) Using the lss adjoint for simultaneous plant and control optimization of chaotic dynamical systems. *Struct Multidiscip Optim* 64(4):2361–2376
- Dereli S, Kker R (2021) Strengthening the pso algorithm with a new technique inspired by the golf game and solving the complex engineering problem. *Complex Intell Syst* 1092(5):1515–1526
- Dutta S, Gandomi A H (2020) Surrogate model-driven bio-inspired optimization algorithms for large-scale and high-dimensional problems. *Biomimicry for Aerospace*
- Economics DO, Wisconsin UO (1996) Using randomization to break the curse of dimensionality. *EconWPA*
- El-Abd and Mohammed (2009) Preventing premature convergence in a PSO and EDA hybrid. *IEEE Congress on Evolutionary Computation*, pp 3060–3066
- Fu-Shiung H, Yi-Hong G (2019) A discrete cooperatively coevolving particle swarm optimization algorithm for combinatorial double auctions. *Appl Intell* 49(11):3845–3863
- Hedar A (2005) Hedar test set. [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\$_files/TestGO.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar$_files/TestGO.htm)
- Ke T, Li X, Suganthan P N, Yang Z, Weise T (2009) Benchmark functions for the cec'2010 special session and competition on large-scale global optimization. *Nat Inspired Comput Appl Lab*, pp 1–18
- Kerner W (1989) Large-scale complex eigenvalue problems. *J Comput Phys* 85(1):1–85
- Kim YR, Park MS, Moore TJ, Varma S (2013) Multiple levels of meta-cognition and their elicitation through complex problem-solving tasks. *J Math Behav* 32(3):377–396
- Li X, Xin Y (2012) Cooperatively coevolving particle swarms for large scale optimization. *Evol Comput IEEE Trans* 16(2):210–224
- Li XD, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput* 16(2):210–224
- Lin H, Tang C (2021) Analysis and optimization of urban public transport lines based on multi-objective adaptive particle swarm optimization. *IEEE Trans Intell Transp Syst* 99(1):1–13
- Liu Q (2011) Two minimal positive bases based direct search conjugate gradient methods for computationally expensive functions. *Numer Algor* 58(4):14357–14374
- Liu Q, Cheng W (2014) A modified DIRECT algorithm with bilevel partition. *J Global Optim* 60(1):483–499
- Liu Q, Zeng J (2015) Global optimization by multilevel partition. *J Global Optim* 61(1):47–69
- Liu Q, Zeng J, Yang G (2015) MrDIRECT: a multilevel robust direct algorithm for global optimization problems. *J Global Optim* 62(1):205–227
- Liu Q, Wei W, Yuan H, Zhan ZH, Li Y (2016) Topology selection for particle swarm optimization. *Inf Sci* 363(1):14357–14374
- Liu Q, Yang G, Zhang Z, Zeng J (2017) Improving the convergence rate of the DIRECT global optimization algorithm. *J Global Optim* 67(1):851–872
- Lui P, Teodorovi D (2011) Computing with bees: attacking complex transportation engineering problems. *Int J Artif Intell Tools* 12(3):375–394
- Ngango J, Hong S (2022) Assessing production efficiency by farm size in Rwanda: a zero-inefficiency stochastic frontier approach, pp 2–29
- Sellali M, Ravey A, Betka A, Kouzou A, Benbouzid M, Djerdir A, Kennel R, Abdelrahem M (2022) Multi-objective optimization-based health-conscious predictive energy management strategy for fuel cell hybrid electric vehicles. *Energies* 15(4):1–17
- Skevas I, Skevas T (2021) A generalized true random-effects model with spatially autocorrelated persistent and transient inefficiency. *Eur J Oper Res* 293(3):1131–1142
- Soheylly S, Mayam S, Hossein M, Mehrjoo M (2016) Modeling a novel CCHP system including solar and wind renewable energy resources and sizing by a CC-MOPSO algorithm. *Appl Energy* 184(1):375–395
- Sun D, Clinet A, Bayen AM (2011) A dual decomposition method for sector capacity constrained traffic flow optimization. *Transp Res Part B Methodol* 45(6):880–902
- Tey KS, Mekhilef S (2014) Modified incremental conductance algorithm for photovoltaic system under partial shading conditions and load variation. *IEEE Trans Industr Electron* 61(10):5384–5392
- Tran KC, Tsonas MG (2016) Zero-inefficiency stochastic frontier models with varying mixing proportion: a semiparametric approach. *Eur J Oper Res* 249(3):1113–1123
- Wah Ma (2006) MANIP—a multicomputer architecture for solving combinatorial extremum-search problems. *IEEE Trans Comput* 33(5):377–390
- Wu Z (2014) Optimization of distribution route selection based on particle swarm algorithm. *Int J Simul Model* 12(2):230–242
- Xiang Z, Chu C, Chen H (2006) A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *Eur J Oper Res* 174(2):1117–1139
- Yan Y, Zhou Q, Cheng S, Liu Q, Li Y (2021) Bilevel-search particle swarm optimization for computationally expensive optimization problems. *Optimization* 25(1):14357–14374
- Yang ZY, Tang K, Yao X (2008) Large scale evolutionary optimization using cooperative coevolution—sciencedirect. *Inf Sci* 178(5):2985–2999
- Yeh WC, Lin YC, Chung YY, Chih M (2010) A particle swarm optimization approach based on monte carlo simulation for solving the complex network reliability problem. *IEEE Trans Reliab* 59(1):212–221
- Yu W, Jin H, Wei SD, Yan JC, Tian CH, Min L, Wen TM (2013) Two-stage based ensemble optimization framework for large-scale global optimization. *Eur J Oper Res* 228(2):308–320
- Zame WR (1993) Asymptotic behavior of asset markets: asymptotic inefficiency. *Gen Equilib Growth Trade* 2:203–233
- Zame WR (1989) Asymptotic inefficiency. *UCLA Economics Working Papers*, pp 1–59

- Zhang J, Sanderson AC (2009) Adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13(5):945–958
- Zhang Y, Yan J, Geng W, Wu L (2010) Find multi-objective paths in stochastic networks via chaotic immune PSO. *Exp Syst Appl Int J* 37(3):1911–1919
- Zhao FQ, Liu Y, Zhang C, Wang JB (2015) A self-adaptive harmony PSO search algorithm and its performance analysis. *Expert Syst Appl* 42(21):7436–7455

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Yanan Wang was born in China, on Nov. 16, 1991. Her research topics lie at power system operation and analysis.