李桐鸥, 潘大志. 动态分组机制的多策略改进粒子群算法实现[J]. 智能计算机与应用,2026,16(5):54-60. DOI:10.20169/j.issn.2095-2163.25101406

动态分组机制的多策略改进粒子群算法实现

李桐鸥, 潘大志
(西华师范大学 数学科学学院, 四川 南充 637009)

摘 要: 针对传统粒子群优化(PSO)算法在迭代过程中易陷入局部最优、早熟收敛的问题,本文提出一种基于动态分组的多策略改进粒子群算法(DGMS-PSO)。该算法在每次迭代时,根据粒子的适应度值实现动态划分,将种群划分为具备全局引导能力的精英粒子与仍有优化潜力的普通粒子,并针对两类粒子分别采用不同的速度更新机制与惯性权重调整策略,平衡全局探索与局部开发能力。此外,引入一种自适应正弦变异策略,根据粒子搜索进程动态调整变异概率,有效增强种群多样性,帮助算法跳出局部最优。为验证 DGMS-PSO 的性能,在多个标准测试函数上将该算法与多种 PSO 变体进行对比实验。结果表明,DGMS-PSO 在收敛速度、求解精度方面均表现出优势,验证了该方法的有效性。

关键词: 粒子群优化算法; 动态分组; 惯性权重; 正弦变异策略; 局部最优解
中图分类号: TP182 **文献标志码**: A **文章编号**: 2095-2163(2026)05-0054-07

Multi-strategy particle swarm optimization algorithm based on dynamic grouping

LI Tong'ou, PAN Dazhi
(School of Mathematical Sciences, China West Normal University, Nanchong 637009, Sichuan, China)

Abstract: To address the tendency of traditional Particle Swarm Optimization (PSO) to fall into local optima and converge prematurely, this paper proposes a multi-strategy improved PSO algorithm based on dynamic grouping (DGMS-PSO). During each iteration, the population is dynamically divided into elite particles with global guidance capability and ordinary particles with remaining optimization potential, based on their fitness values. Different velocity update mechanisms and inertia weight adjustment strategies are applied to the two groups to balance global exploration and local exploitation. Additionally, an adaptive sine mutation strategy is introduced, which dynamically adjusts the mutation probability according to the search progress, effectively enhancing population diversity and helping the algorithm escape local optima. To evaluate the performance of DGMS-PSO, comparative experiments with several PSO variants are conducted on multiple benchmark test functions. Results show that DGMS-PSO outperforms others in convergence speed and solution accuracy, demonstrating its effectiveness.

Key words: Particle Swarm Optimization algorithm; dynamic grouping; inertia weight; sine mutation strategy; local optimum

0 引 言

粒子群优化算法(PSO)是 Kenny 和 Eberhart 受鸟群觅食过程启发而提出的一种基于随机搜索的智能优化算法[1]。在寻优过程中,搜索空间随机生成一定规模的粒子,当前粒子通过向自身最佳位置及群体最佳位置学习,最终引导整个群体向最优解的方向移动[2]。PSO 具有概念简单、参数少、收敛迅速的优点,在函数优化、工程问题、路径规划问题等领域得到了广泛的应用。在处理复杂问题时,为解决 PSO 易陷入局部最优和收敛过早等问题,研究者们在控制参数、粒子群拓扑结构与学习对象、融合其他算法这几方面进行了改进。

惯性权重和加速因子是 PSO 的重要参数。惯性权重的改进方法有线性递减、非线性递减[3-4]等。Liu 等[5]利用 logistic 映射构造了一种具有很强波动性的非线性惯性权重,有利于平衡全局搜索与局部开发。康岩松等[6]利用粒子适应度构造函数作为

基金项目: 国家自然科学基金(11871059);四川省教育厅自然科学基金(18ZA0469)。
作者简介: 李桐鸥(2002—),女,硕士研究生。主要研究方向:智能计算,组合优化。
通信作者: 潘大志(1974—),男,博士,教授,硕士生导师。主要研究方向:智能计算,算法设计。Email:pdzzj@126.com。
收稿日期: 2025-10-14

惯性权重，赋予粒子与之适应的搜索能力。Xu 等[7]提出了随迭代次数变化而变化的学习因子，在算法的最优值和收敛速度方面都有显著改善。

改变粒子群的拓扑结构，就是改变粒子间的信息交流方式。当前粒子向全局最优粒子学习，同时，向拓扑邻域粒子中的局部最优位置学习[8]。研究者们已设计出包括环形[9]、星型[9]、冯诺依曼结构[10]等拓扑结构。王浩丞等[11]提出了一种基于环形拓扑结构的动态更新机制，有效扩大了算法在迭代空间的搜索范围。高海军等[12]扩充了邻域半径，将线性结构连接变成星型结构，提高了算法的全局搜索能力。Feng 等[13]提到了改进的二维方形晶体模型，并引入魔方的旋转操作，动态地改变邻域并增加信息交流。Zhang 等[14]提出了一种具有自组织拓扑结构的新型粒子群优化算法。在学习样本方面，Zhao 等[15]将精英粒子的个体历史最优位置的加权中心作为普通粒子的学习对象。精英档案库中的粒子[16]也可作为普通粒子的学习样本。

PSO 与其他进化算法融合，往往会形成一种性能更佳的混合粒子群算法。徐展鹏等[17]在 PSO 中引入遗传算法的思想，提升了算法的全局优化能力。此外，PSO 还可与萤火虫算法[18]、鲸鱼优化算法[19]、海洋捕食者算法[20]、蝴蝶优化算法[21]等相结合。

本文提出一种基于动态分组的多策略改进粒子群算法(DGMS-PSO)。在每次迭代过程中，根据粒子的适应度，按比例将种群动态划分为精英粒子与普通粒子，针对两类粒子分别设计不同的速度更新机制与自适应的惯性权重，并引入一种自适应正弦变异策略，在粒子搜索进程中动态调整变异概率。

1　粒子群优化算法

PSO 中每个粒子被抽象成没有质量、体积但有速度的点，这种简化的模型决定了粒子可在空间中移动并不断开发与探索搜索空间。第 t 次迭代时第 i 个粒子的速度和位置更新公式为：

$$V_i^d(t+1)=w\cdot V_i^d(t)+c_1\cdot r_1\cdot(p\text{best}_i^d(t)-X_i^d(t))+c_2\cdot r_2\cdot(g\text{best}^d(t)-X_i^d(t)) \quad (1)$$

$$X_i^d(t+1)=X_i^d(t)+V_i^d(t+1) \quad (2)$$

$$(1\leqslant i\leqslant N,1\leqslant d\leqslant D)$$

其中，$V_i^d(t)$ 表示第 t 次迭代第 i 个粒子的速度向量的第 d 维；$X_i^d(t)$ 表示第 t 次迭代第 i 个粒子的位置向量的第 d 维；$p\text{best}_i^d(t)$ 表示第 t 次迭代第 i 个粒子的历史最优位置的第 d 维；$g\text{best}^d(t)$ 表示第 t 次迭代群体的历史最优位置的第 d 维；w 为惯性权重；c_1、c_2 分别为个体学习因子和群体学习因子；r_1、r_2 为均匀分布在(0,1)以内的随机数。

对于局部类型的粒子群算法，全局最优历史位置被邻域局部最优位置替代，将第 i 个粒子局部历史最优位置的第 d 维表示为 $n\text{best}_i^d$，则 PSO 的速度更新公式为：

$$V_i^d(t+1)=w\cdot V_i^d(t)+c_1\cdot r_1\cdot(p\text{best}_i^d(t)-X_i^d(t))+c_2\cdot r_2\cdot(n\text{best}_i^d(t)-X_i^d(t)) \quad (3)$$

2　改进的粒子群算法

2.1　八粒子邻域拓扑结构

拓扑结构显著地影响着粒子群优化算法的性能。标准的 PSO 算法中，全局最优位置引导其他粒子向最优解的方向移动，群体中的每个粒子都可以与其他粒子连接并交换信息，全局最优位置的信息可以在整个群体中共享，这也是全局型 PSO 搜索速度快，但容易陷入局部最优的原因。

为加强粒子之间的联系，避免陷入局部最优，本文提出了一种八粒子的邻域拓扑结构。如图 1 所示，将 N 个粒子序号排列成 m 行 n 列的队列，中心粒子 i 的上、下、左、右、右上、左上、右下、左下 8 个方位的邻居粒子，构成了二维平面上的八粒子邻域结构。为保证八粒子邻域结构的完整性，增加了第一行粒子与最后一行粒子、第一列粒子与最后一列粒子的连接，最终形成三维“甜甜圈”结构，如图 2 所示。

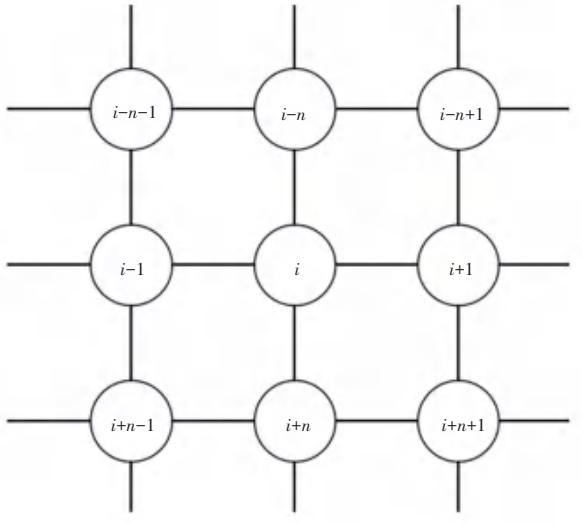


图 1　八粒子邻域结构

Fig. 1　Eight-particle neighborhood structure

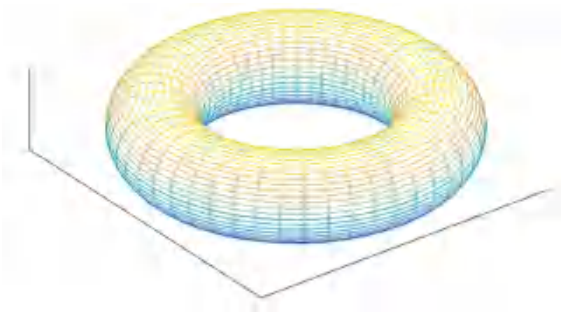


图 2 三维“甜甜圈”结构

Fig. 2 Three-dimensional torus structure

2.2 基于动态分组的速度更新策略

在每次迭代中,按照粒子适应度表现由优到劣排列,根据比例动态地选择前 $a\%$ 的粒子作为精英粒子(Elite particle)。这类粒子的适应度表现较好,让这类粒子在保持自身信息的同时,向八粒子邻域结构中适应度表现最好的粒子学习;其余粒子为普通粒子(Ordinary particle),向粒子群中最优的粒子学习。对于第 t 次迭代的第 i 个粒子,其速度和位置的更新公式如下:

$$V_i^d(t+1)=w_i^t\cdot V_i^d(t)+c_1\cdot r_1\cdot(p\text{best}_i^d(t)-X_i^d(t))+c_2\cdot r_2\cdot(g\text{best}^d(t)-X_i^d(t)),i\in O \quad (4)$$

$$V_i^d(t+1)=w_i^t\cdot V_i^d(t)+c_3\cdot r_3\cdot(p\text{best}_i^d(t)-X_i^d(t))+c_4\cdot r_4\cdot(n\text{best}_i^d(t)-X_i^d(t)),i\in E \quad (5)$$

$$X_i^d(t+1)=\eta\cdot X_i^d(t)+V_i^d(t+1),1\leqslant i\leqslant N,\ 1\leqslant d\leqslant D \quad (6)$$

其中,w_i^t 是第 i 个粒子在第 t 次迭代的自适应惯性系数;η 为位置更新加速系数,是一个常数;c_1、c_2、c_3、c_4 是学习因子;r_1、r_2、r_3、r_4 为均匀分布在(0,1)以内的随机数;集合 O 是包含所有普通粒子序号的集合;集合 E 是包含所有精英粒子序号的集合。

本文采用 DCIW 法[6]更新权重,自适应权重可加强算法局部搜索能力和跳出局部最优解的能力,计算公式为:

$$n_i=\frac{F_i-F_{g\text{best}}}{F_m-F_{g\text{best}}} \quad (7)$$

$$\text{wa}=\frac{\text{ws}+\text{we}\left(1-\frac{1}{N}\right)}{2-\frac{1}{N}} \quad (8)$$

$$w_i=\begin{cases}\text{wa}+(\text{wa}-\text{we})\left(1-\frac{1}{n_i}\right), & n_i\geqslant 1\\ \text{we}+n_i(\text{wa}-\text{we}), & n_i<1\end{cases} \quad (9)$$

其中,w_i 为第 i 个粒子的惯性权重;F_i 表示第 i 个粒子的适应度值;$F_{g\text{best}}$ 表示当前全局最优粒子的适应度值;F_m 表示其余粒子与劣势粒子(适应度由好到劣排列后 $b\%$ 的粒子)适应度的分界值;ws、we 为惯性系数上下限;n_i 为第 i 个粒子的偏离度;N 为 n_i 的最大值。

2.3 正弦变异策略

在处理多峰问题和复杂问题时,为保持算法全局搜索的鲁棒性与收敛的有效性,引入正弦变异策略,帮助粒子在当前解附近再次进行探索,提升算法跳出局部最优的概率,又有效避免随机变异带来的盲目性与搜索效率低下的问题,变异公式如下:

$$X_{i\ \text{new}}^d=X_{i\ \text{old}}^d+\left(1+\sin\left(\frac{pi}{2}+\left(r-\frac{1}{2}\right)\cdot pi\right)\right)\cdot n\text{best}_j^d,\ 1\leqslant i\leqslant N,1\leqslant d\leqslant D \quad (10)$$

其中,$X_{i\ \text{old}}^d$ 是变异前的解;$X_{i\ \text{new}}^d$ 是变异后的解;r 为均匀分布在(0,1)以内的随机数;若 i 为精英粒子,则 $j=i$;否则,$n\text{best}_j^d$ 为粒子 i 的八邻域拓扑最优粒子。

正弦变异策略流程如下:

输入 N,D,F,x_i;

输出 $x_{i\text{new}},F(x_{i\text{new}})$

1 For $i=1,2,\cdots N$

2 For $j=1,2,\cdots D$ %对每个粒子的每一维进行变异操作

3 ratios (i) = initratio %初始化每个粒子的变异概率为 initratio

4 If rand(1) <ratios (i) then

5 按式(10)对 x_i^j 进行变异,变异后的粒子位置为 $x_{i\text{new}}^j$

6 End

7 End

8 计算 $F(x_{i\text{new}})$;

9 $R=F(x_{i\text{new}})/F(x_i)$;

10 If $R>0.9$ 连续 5 次 then

11 ratios (i) = ratios(i) +ratio;%在当前变异概率的基础上增加 ratio

12 End

13 If 累计 5 次增大变异概率 then

14 ratios(i) = initratio;%重置变异概率为初始变异概率

15 End

16 End

2.4 DGMS-PSO 算法的流程

本文改进的 DGMS-PSO 算法的流程如图 3 所示。

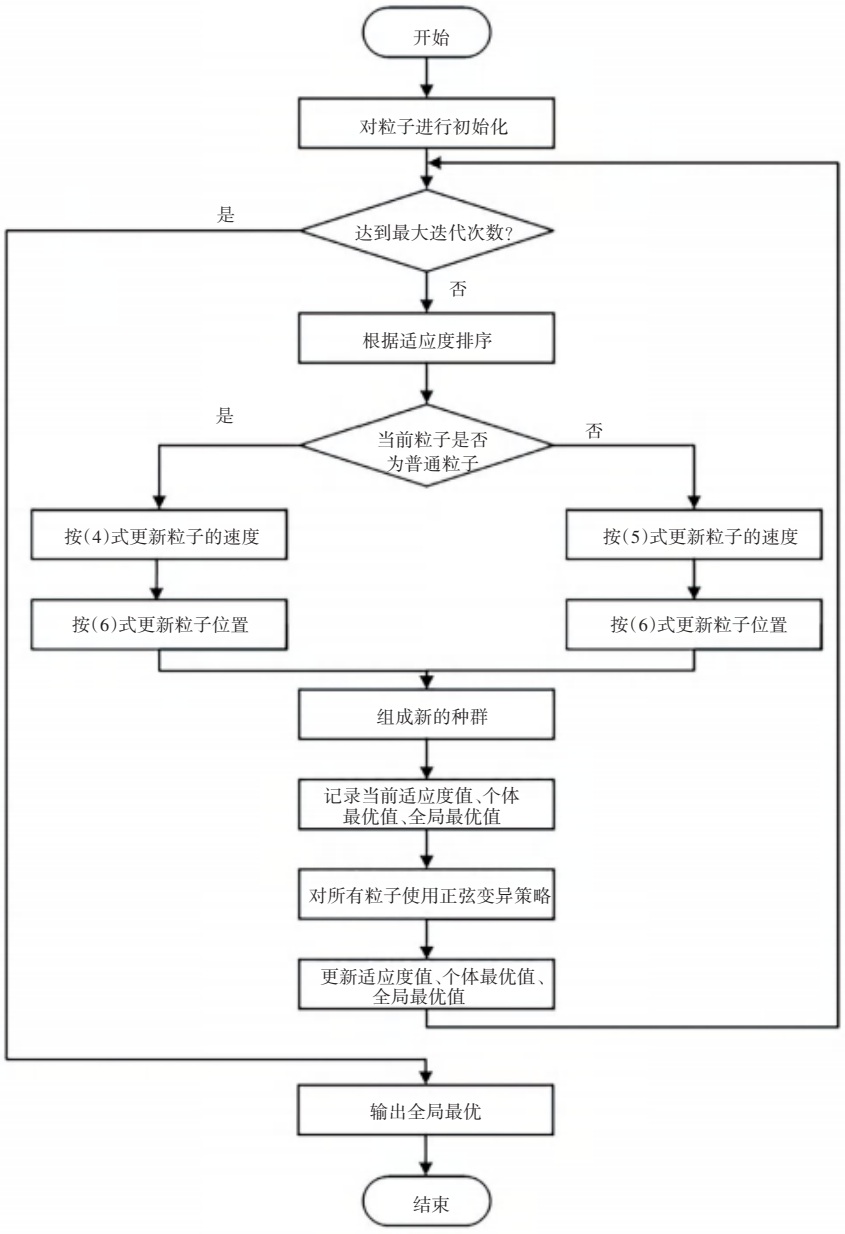


图 3　DGMS-PSO 算法流程

Fig. 3　Flowchart of the DGMS-PSO algorithm

3　实验仿真与结果分析

3.1　参数设置

本文选用 14 个标准测试函数验证 DGMS-PSO 算法的性能，并与 PSO 算法[1]、LDIWPSO 算法[22]、PSO_CF 算法[23]、FIPS 算法[24]、HDPSO 算法[6]、CBPPSO 算法[25]、MPSO 算法[16]的测试结果进行对比分析，为保证公平，每个算法运行 50 次，最大迭代次数 maxgen = 500，种群数量 $N = 30$，维数 $D = 30$，速度初始系数 $v = 0.12$，表 1 给出了粒子群优化算法及其变体算法的参数，表 2 为测试函数的定义。

表 1　粒子群优化算法及其变体算法的参数

Table 1　Parameters of Particle Swarm Optimization Algorithm and its Variant Algorithms

算法	参数设置
PSO	$w = 0.7, c_1 = c_2 = 2$
LDIWPSO	w:[0.4,0.9], $c_1 = c_2 = 2$
PSO_CF	w:[0.4,0.9], $c_1 = c_2 = 2$
HDPSO	w:[0.4,0.9], $c_1 = c_2 = 2$, initratio = 0.000 5, ratio = 0.001, factor = 15, $\eta = 0.97$
FIPS	$\chi = 0.729, c_1 = c_2 = 2$
MPSO	w:[0.4,0.9], $cw = 4 \cdot r \cdot (1 - r), r \in (0,1)$
CBPPSO	$c_1 = c_2 = 2$, w:[0.4,0.9]
DGMS-PSO	we = 0.4, ws = 0.9, $c_1 = c_2 = c_3 = c_4 = 2$, $\eta = 0.97$, initratio = 0.000 5, ratio = 0.001, $a = 20, b = 20, R = 0.9$

表 2 测试函数

Table 2 Test functions

名称	表达式	参数范围	函数特性
$F1$	$\sum_{i=1}^{n} x_i^2$	$[-100,100]$	单峰函数
$F2$	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10,10]$	单峰函数、非光滑函数
$F3$	$\sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	$[-100,100]$	单峰函数
$F4$	$\max(\lvert x_i \rvert, 1 \leqslant i \leqslant n)$	$[-100,100]$	单峰函数、非光滑函数
$F5$	$\sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-10,10]$	多峰函数
$F6$	$\sum_{i=1}^{n} \lvert x_i + 0.5 \rvert^2$	$[-100,100]$	单峰函数
$F7$	$\sum_{i=1}^{n} ix_i^4 + \mathrm{rand}[0,1)$	$[-1.28,1.28]$	单峰函数、引入随机噪声
$F8$	$\sum_{i=1}^{n} (-x_i \sin \sqrt{\lvert x_i \rvert})$	$[-500,500]$	多峰函数
$F9$	$\sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]$	多峰函数
$F10$	$\frac{1}{4\,000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]$	多峰函数
$F11$	$\frac{\pi}{n}[10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_n^2 - 1)^2] + \sum_{i=1}^{n} u(x_i,10,100,4)$, $y_i = 1 + \frac{1}{4}x_{i+1}$, $u(x_i,a,k,m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a < x_i < a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	$[-50,50]$	多峰函数
$F12$	$0.1[\sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + \sin^2(3\pi x_1) + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]] + \sum_{i=1}^{n} u(x_i,5,100,4)$	$[-50,50]$	多峰函数
$F13$	$\sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	$[-50,50]$	多峰函数
$F14$	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	$[-5,5]$	多峰函数、低维函数，但具有复杂的局部最小值结构

3.2 结果分析

每个算法在每个测试集上运行 50 次，基于结果的平均值、标准差排名来评价收敛精度。由表 3 可知，DGMS-PSO 算法在函数 $F_1 \sim F_5$、$F_7 \sim F_{10}$、F_{12} 上的平均值、标准差均优于其余算法；其算法的平均排名为 1.79，综合排名为 1。无论是针对单峰函数还是多峰函数，DGMS-PSO 算法均表现出优秀的搜索精度和一定的稳定性，具有较好的全局搜索和跳出局部最优解的能力。

表 3　测试结果

Table 3　Test results

测试函数		PSO	PSO_CF	LDIWPSO	FIPS	MPSO	CBPPSO	HDPSO	DGMS-PSO
*F*1	平均值	2.87E+01	7.61E+02	1.95E+01	3.50E+02	4.78E−07	4.70E−06	2.90E−24	**3.12E−41**
	标准差	1.20E+01	3.51E+02	7.37E+00	1.55E+02	2.95E−06	8.71E−06	1.64E−23	**1.13E−40**
	排名	6	8	5	7	3	4	2	1
*F*2	平均值	5.86E+00	1.48E+01	4.21E+00	7.54E+00	7.48E−02	3.69E−04	5.34E−13	**9.50E−21**
	标准差	2.09E+00	2.99E+00	1.73E+00	1.50E+00	2.76E−01	6.16E−04	1.42E−12	**2.79E−20**
	排名	6	8	5	7	4	3	2	1
*F*3	平均值	4.77E+02	3.73E+03	6.35E+02	5.51E+02	5.81E+00	3.64E+02	2.67E−22	**2.77E−37**
	标准差	2.44E+02	2.71E+03	2.59E+02	2.56E+02	1.00E+01	1.52E+02	1.27E−21	**1.16E−36**
	排名	5	8	7	6	3	4	2	1
*F*4	平均值	9.54E+00	1.75E+01	8.72E+00	6.57E+00	1.20E+01	3.87E+00	4.72E−11	**4.98E−16**
	标准差	2.25E+00	4.06E+00	2.58E+00	1.21E+00	2.69E+00	8.45E−01	7.49E−11	**1.66E−15**
	排名	6	8	5	4	7	3	2	1
*F*5	平均值	1.39E+02	1.29E+03	1.23E+02	4.07E+02	2.76E+01	4.56E+01	2.74E+01	**2.60E+01**
	标准差	6.39E+01	8.69E+02	6.47E+01	2.49E+02	1.47E+01	2.76E+01	6.91E−01	**4.54E−01**
	排名	6	8	5	7	3	4	2	1
*F*6	平均值	3.35E+01	6.51E+02	2.00E+01	3.70E+02	7.92E−05	**6.55E−06**	1.14E−01	1.45E−02
	标准差	1.63E+01	2.93E+02	8.88E+00	1.39E+02	5.51E−04	**1.48E−05**	1.24E−01	3.73E−02
	排名	6	8	5	7	2	1	4	3
*F*7	平均值	5.78E−02	1.99E−01	4.65E−02	3.74E−02	2.10E−02	2.48E−02	9.27E−03	**3.62E−03**
	标准差	3.30E−02	1.41E−01	2.04E−02	1.95E−02	1.06E−02	8.48E−03	4.99E−03	**1.36E−03**
	排名	7	8	6	5	3	4	2	1
*F*8	平均值	−6.13E+03	−5.69E+03	−6.24E+03	−3.45E+03	−7.79E+03	−6.74E+03	−6.30E+03	**−9.58E+03**
	标准差	8.75E+02	9.01E+02	9.26E+02	5.69E+02	7.15E+02	8.48E+02	9.80E+02	**7.13E+02**
	排名	6	7	5	8	2	3	4	1
*F*9	平均值	4.45E+01	9.17E+01	3.91E+01	3.99E+01	3.19E+01	5.00E+01	3.34E+01	**2.46E+01**
	标准差	1.33E+01	1.86E+01	1.44E+01	8.95E+00	1.10E+01	1.20E+01	1.39E+01	**8.37E+00**
	排名	6	8	4	5	2	7	3	1
*F*10	平均值	1.33E+00	7.72E+00	1.18E+00	4.18E+00	8.14E−02	1.69E−02	1.63E−02	**8.44E−03**
	标准差	1.71E−01	2.84E+00	1.09E−01	1.16E+00	9.10E−02	2.28E−02	3.01E−02	**1.05E−02**
	排名	6	8	5	7	4	3	2	1
*F*11	平均值	−3.76E−01	4.15E+00	−4.74E−01	**−2.36E−02**	−9.66E−01	−9.29E−01	−8.99E−01	−9.68E−01
	标准差	6.69E−01	2.56E+00	5.12E−01	4.75E−01	1.24E−01	1.47E−01	2.15E−01	**9.10E−02**
	排名	2	8	3	1	6	5	4	7
*F*12	平均值	3.38E+01	2.16E+03	1.62E+01	2.01E+01	1.21E+01	5.00E−01	1.69E+00	**9.60E−02**
	标准差	1.90E+01	8.45E+03	1.58E+01	1.14E+01	5.91E+00	1.26E+00	6.84E−01	**1.03E−01**
	排名	7	8	5	6	4	2	3	1
*F*13	平均值	1.88E+00	8.30E+00	1.85E+00	2.40E+00	4.29E−03	**2.87E−04**	1.93E+00	6.81E−02
	标准差	1.37E+00	2.78E+00	1.31E+00	7.29E−01	1.72E−02	**1.29E−03**	2.19E+00	8.34E−02
	排名	5	8	4	7	2	1	6	3
*F*14	平均值	1.98E+01	1.97E+01	1.98E+01	**1.97E+01**	1.98E+01	1.98E+01	1.97E+01	1.97E+01
	标准差	1.03E−01	9.77E−02	1.01E−01	**4.10E−02**	1.04E−01	9.92E−02	9.40E−02	8.38E−02
	排名	7	4	6	1	8	5	3	2
	平均排名	5.79	7.64	5.00	5.57	3.79	3.50	2.93	1.79
	最终排名	7	8	5	6	4	3	2	1

4　结束语

本文结合动态分组策略、八粒子邻域拓扑结构、自适应惯性权重和正弦变异策略对粒子群优化算法进行改进，平衡了全局探索与局部开发能力，并有效增强种群多样性，帮助算法跳出局部最优解。使用标准测试函数对改进算法和粒子群算法的变种进行了测试，结果表明该算法具有优秀的收敛速度，并提高了搜索精度、增强了稳定性。此外，八粒子邻域拓扑结构中还有许多可以挖掘的信息，有待在未来工作中需要进一步研究分析。

参考文献

[1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//

Proceedings of International Conference on Neural Networks. Piscataway,NJ:IEEE,1995:1942-1948.
[2] 许胜才, 蔡军, 程昀, 等. 基于拓扑结构与粒子变异改进的粒子群优化算法[J]. 控制与决策,2019,34(2):419-428.
[3] SHI Y, EBERHART R C. Parameter selection in particle swarm optimization [C]//Proceedings of International Conference on Evolutionary Programming. Cham: Springer , 1998: 591-600.
[4] SHI Y, EBERHART R C. Fuzzy adaptive particle swarm optimization [C]//Proceedings of the 2001 Congress on Evolutionary Computation. Piscataway,NJ:IEEE, 2001:101-106.
[5] LIU H, ZHANG X W, TU L P. A modified particle swarm optimization using adaptive strategy [J]. Expert Systems with Applications, 2020, 152: 113353.
[6] 康岩松, 臧顺来. 基于多种策略的改进粒子群优化算法[J]. 东北大学学报(自然科学版), 2023,44(8):1089-1097.
[7] XU H, DENG Q, ZHANG Z, et al. A hybrid differential evolution particle swarm optimization algorithm based on dynamic strategies[J]. Scientific Reports,2025,15(1):4518.
[8] CLERC M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization[C]//Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE,1999:1951-1957.
[9] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.
[10] MENDES R, KENNEDY J, NEVES J. Watch thy neighbor or how the swarm can learn from its environment[C]//Proceedings of the 2003 IEEE Swarm Intelligence Symposium. Piscataway,NJ: IEEE,2003: 88-94.
[11] 王浩丞, 李凌. 动态拓扑结构混合粒子群算法及其应用[J]. 计算机科学与探索,2024,18(8): 2065-2079.
[12] 高海军, 潘大志. 星型结构的多目标粒子群算法求解多模态多目标问题[J]. 计算机工程与科学,2020,42(8):1472-1481.
[13] FENG X, CHEN Y, ZHANG J, et al. Improved Particle Swarm Algorithm using Rubik′s Cube Topology for bilevel building energy transaction[J]. Journal of Energy Resources Technology, 2022, 144(7):4053768. DOI:10.1115/1.4053768.
[14] ZHANG D, MA G, DENG Z, et al. A self-adaptive gradient-based particle swarm optimization algorithm with dynamic population topology[J]. Applied Soft Computing, 2022, 130: 109660.
[15] ZHAO S, WANG D. Elite-ordinary synergistic particle swarm optimization[J]. Information Sciences, 2022, 609: 1567-1587.
[16] ZHANG Y. Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications [J]. Swarm and Evolutionary Computation, 2023, 76: 101212.
[17] 徐展鹏, 陈福新, 杨雪凡, 等. 一种基于遗传算法改进粒子群算法的光储氢并网型微电网容量配置优化模型研究[J]. 太阳能学报,2025,46(7):144-153. DOI:10.19912/j.0254-0096.tynxb.2024-0319.
[18] SINGH A K, KUMAR A. Multi-objective: Hybrid particle swarm optimization with firefly algorithm for feature selection with Leaky ReLU[J]. Discover Artificial Intelligence, 2025, 5(1): 192.
[19] QIAO J, WANG G, YANG Z, et al. A hybrid particle swarm optimization algorithm for solving engineering problem [J]. Scientific Reports, 2024, 14(1): 8357.
[20] HAN B, LI B, QIN C. A novel hybrid particle swarm optimization with marine predators[J]. Swarm and Evolutionary Computation, 2023, 83: 101375.
[21] WANG Y, DAI S, LIU P, et al. A hybrid particle swarm optimization with butterfly optimization algorithm based maximum power point tracking for photovoltaic array under partial shading conditions[J]. Sustainability, 2023, 15(16): 12402.
[22] SHI Y, EBERHART R. A modified particle swarm optimizer [C]//Proceedings of the 1998 IEEE International Conference on Evolutionary Computation . Piscataway,NJ:IEEE, 1998: 69-73.
[23] EBERHART R C, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]// Proceedings of the 2000 Congress on Evolutionary Computation. Piscataway,NJ:IEEE, 2000: 84-88.
[24] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.
[25] AGRAWAL A, TRIPATHI S. Particle swarm optimization with adaptive inertia weight based on cumulative binomial probability [J]. Evolutionary Intelligence, 2021, 14(2): 305-313.