doi:10. 16055/j. issn. 1672-058X. 2022. 0005. 005

基于侦察学习策略的粒子群算法优化

张　倩[1], 刘衍民[2], 杨妹兰[1], 舒小丽[3]
(1. 贵州大学 数学与统计学院, 贵阳 550025;2. 遵义师范学院 数学学院, 贵州 遵义 563006;
3. 贵州民族大学 数据科学与信息工程学院, 贵阳 550025)

摘　要:针对粒子群算法收敛能力不足和易陷入局部最优的问题,提出了一种基于侦察学习策略的新型粒子群算法。算法首先利用拓扑结构构建粒子种群,其次采用联合因子均衡算法的局部搜索能力和全局搜索能力,并通过侦察学习策略改进算法的速度和位置公式进而产生候选解;Wilcoxon 秩和检验结果和 CEC2017 基准函数检测结果表明,新型粒子群算法的收敛能力,最优解精度以及算法稳定性更好,说明算法性能得以提升。

关键词:粒子群算法;拓扑结构;侦察学习策略
中图分类号:TP18　　**文献标志码**:A　　**文章编号**:1672-058X(2022)05-0034-08

0　引　言

群智能优化算法衍生于生物群体行为,其中,粒子群算法(PSO)[1]由于其并行性高,全局搜索性强等优点,被广泛应用于配电网输送[2]领域,但此算法在迭代过程中易出现“早熟”现象。

为消除这一弊端,研究者对 PSO 算法进行了多方向改进。一是结合 PSO 算法与其他优化算法。文献[3]将差分进化算法(DE)的突变过程与 PSO 算法结合,此方法保证了粒子的多样性,但其计算复杂度却大大增加。文献[4]将 PSO 算法、生物地理信息优化算法(BBO)[5]以及侦察学习策略三者互相融合,这样虽然满足算法对结果精度的需求,但降低了算法的收敛速率。二是对粒子间的信息交流进行改进。文献[6]提出全信息 PSO 算法(wFIPS)。根据群体粒子经历的最优值和其邻域中表现优异的粒子分量调整粒子速度,达到对粒子信息的“充分了解”,此算法在一定程度上提高了 PSO 算法的局部搜索性能,但是其局部搜索能力和全局搜索能力并未均衡。文献[7]提出了合同协作 PSO 算法(CPSO)。该算法利用多个群体进行不同粒子间的交叉操作,改进粒子间的信息交换过程,此算法在增加粒子间信息交流的基础上,造成了数据大量冗余。

为改善这些不足,本文提出了一种结合拓扑结构与侦察学习策略的新型 PSO 算法(BUPSO),通过实验对比分析,该算法可兼顾算法的局部搜索能力和全局搜索能力,在提高算法收敛速率的同时规避算法陷入“早熟”现象,提升了算法的整体性能。

1　算法基础理论

1.1　标准粒子群算法

在标准粒子群算法中,假设当前种群规模为 N,

收稿日期:2021-08-11;**修回日期**:2021-09-30.
基金项目:国家自然科学基金资助项目(71461027);贵州省科技创新人才团队(黔科合平台人才[2016]5619).
作者简介:张倩(1997—),女,山东德州人,硕士研究生,从事优化理论及智能算法研究.
通讯作者:刘衍民(1978—),男,山东临沂人,教授,博士研究生导师,从事优化理论及智能算法研究. Email:yanmin7813@163. com

维度为 D,粒子的位置表示为 $x_i=\{x_1,x_2,x_3,\cdots,x_N\}$,速度表示为 $v_i=\{v_1,v_2,v_3,\cdots,v_N\}$,整个搜寻过程如式(1)、式(2)所示:

$$\bar{v}_{ij}^{(t+1)}=w\bar{v}_{ij}^{(t+1)}+c_1r_1[\bar{p}_{ij}^{(t)}-\bar{x}_{ij}^{(t)}]+c_2r_2[\bar{p}_{gj}^{(t)}-\bar{x}_{ij}^{(t)}] \quad (1)$$

$$\bar{x}_{ij}^{(t+1)}=\bar{x}_{ij}^{(t)}+\bar{v}_{ij}^{(t+1)} \quad (2)$$

其中,$\bar{v}_{ij}^{(t+1)}$ 和 $\bar{x}_{ij}^{(t+1)}$ 分别代表粒子 i 第 j 维在第 $t+1$ 代的速度和位置分量;w 为惯性权重,使粒子具有扩展搜索空间的能力,取值为 0.729;c_1 和 c_2 代表社会学习系数,取值为 1.494 45;r_1 和 r_2 为[0,1]内的随机数;$\bar{p}_{ij}^{(t)}$ 和 $\bar{p}_{gj}^{(t)}$ 分别代表粒子 i 在第 j 维分别经历的个人最优位置和全局最优位置,$c_1r_1[\bar{p}_{ij}^{(t)}-\bar{x}_{ij}^{(t)}]$ 代表粒子的“认知部分”;$c_2r_2[\bar{p}_{gj}^{(t)}-\bar{x}_{ij}^{(t)}]$ 代表粒子的“社会部分”。

1.2　侦察学习策略

PSO 算法只有粒子的学习阶段,若在此阶段对粒子的速度和位置进行多次迭代,算法搜索能力会因此降低而导致粒子群体在某一局部极值点停滞。因此,寻找一种合适的策略,使得算法跳出局部收敛显得尤为重要。文献[4]将侦察学习策略引入 PSO 算法。若 PSO 算法经过数次迭代后发现最优解的位置未被更新,则通过侦察学习策略产生新解,产生过程如式(3):

$$\bar{x}_{ij}=\bar{x}_{\min j}+r(\bar{x}_{\max j}-\bar{x}_{\min j}) \quad (3)$$

其中,x_{ij} 代表粒子 i 在第 j 维产生的随机位置,$x_{\max}j$,$x_{\min}j$ 代表侯选解的范围,r 为[0,1]的随机数。

2　结合拓扑结构和侦察学习策略的新型粒子群算法

2.1　基于拓扑结构的种群构建

粒子群体应用不同的拓扑结构,粒子间的信息交流方式与信息交流量也不同。以 All 型、Ring 型为例,Ring 型拓扑结构中粒子间的交流局限于相邻两粒子间,而 All 型拓扑结构中所有粒子可实现互相交流。两类拓扑结构如图 1 和图 2 所示。

图 1　Ring 型结构

Fig. 1　Ring-type structure

图 2　All 型结构

Fig. 2　All-type structure

依据种群粒子间信息交流量的不同,可将搜索方式分为局部版本和全局版本。若群体中所有粒子均产生信息交换,则为全局版本。PSO 算法使用的即为 All 型全局版本搜索方式。若粒子的信息交流局限于邻居粒子中,则为局部版本。文献[8]指出 Ring 型拓扑结构收敛精度与收敛速率最高,因此本算法应用 Ring 型局部版本搜索方式。两种搜索方式的不同之处在于粒子的速度更新方式不同,局部版本速度的“社会认知”项由邻居中表现最好的粒子引导,而全局版本速度的“社会认知”项由群体中表现最好的粒子引导。式(4)、式(5)分别代表全局搜索版本以及局部搜索版本的粒子速度。具体表述如式(4)、式(5):

$$\bar{v}_{gbest(i)}^{(t+1)}=w\bar{v}_i^{(t)}+c_1r_1(\bar{p}_{pbest(i)}^{(t)}-\bar{x}_i^{(t)})+c_2r_2(\bar{p}_{gbest(i)}^{(t)}-\bar{x}_i^{(t)}) \quad (4)$$

$$\bar{v}_{pbest(i)}^{(t+1)}=w\bar{v}_i^{(t)}+c_1r_1(\bar{p}_{pbest(i)}^{(t)}-\bar{x}_i^{(t)})+c_2r_2(\bar{p}_{neighbor(i)}^{(t)}-\bar{x}_i^{(t)}) \quad (5)$$

其中,$\bar{p}_{pbest(i)}^{(t)}$ 代表粒子 i 在第 t 次迭代时经历的局部最优位置,$\bar{p}_{gbest(i)}^{(t)}$ 代表截止到迭代时刻 t 整个群体中的最优位置,$\bar{p}_{neighbor(i)}^{(t)}$ 代表在迭代时刻 t 粒子 i 的邻居粒子所经历的最好位置。

局部版本搜索方式有利于粒子勘探更优质解,而全局版本搜索方式有利于粒子开发新区域。为充分融合两种搜索方式,本文引入联合因子 m 这一指标将两种搜索方式均衡结合,既能保证粒子搜索的广度,又可满足算法的整体收敛精度。两种搜索的结合方式如式(6):

$$\bar{v}_i^{(t+1)}=m\bar{v}_{gbest(i)}^{(t+1)}+(1-m)\bar{v}_{pbest(i)}^{(t+1)} \quad (6)$$

其中,$\bar{v}_{gbest(i)}^{(t+1)}$ 代表全局版本速度,$\bar{v}_{pbest(i)}^{(t+1)}$ 代表局部版本速度。m 取值范围为[0,1],根据 m 的不同取值,式(6)有不同的数值意义。当 m 取 0 时,式(6)表示此时的粒子速度为局部最优速度;当 m 取 1 时,式(6)表示此时的粒子速度为全局最优速度。

此外,为模拟生物进化算法中的突变过程,规避算法陷入局部最优的风险,本算法引入随机参数 r_3。文献[9]指出当 r_3 均值为 0,标准差为 0.01 的正态分布随机向量且 m 取 0.1 时,该算法收敛于全局最优值的效果更优。此更新过程如式(7):

$$\bar{v}_i^{(t+1)}=r_3m\bar{v}_{gbest(i)}^{(t+1)}+(1-m)\bar{v}_{pbest(i)}^{(t+1)} \quad (7)$$

2.2　侦察学习策略的引入

传统的侦察学习策略会在确定范围内随机更新粒子位置,因而无法加强粒子间的信息交流,导致算法的运算效率降低。故此,本文对侦察学习策略进行了相应改进,引入了粒子的速度公式,并在粒子位置公式中引入新变量,以确保该粒子做到绝对更新,增加粒子寻求到最优解的概率以及侦察学习策略在本文中的普适性。为了方便描述,用 x_i 表示产生坏解的粒子,x_j 表示更新后的粒子。此阶段的改进思

想如下：

在速度更新方面，当 PSO 算法陷入局部停滞时，说明此时粒子的状态不能促使算法找到最优解。通过 x_j 与 x_i 进行信息交流，由 x_i 指导 x_j 的飞行方向，可加快粒子的搜索效率。此外，在确定粒子速度飞行方向的基础上增加该粒子的搜索步长项，即 $cr_4(\bar{v}_{Rmaxj}-\bar{v}_{Rminj})$，可扩大粒子的搜索区域。

在粒子位置更新方面，在边界下限的基础上引入随机数 r_4，将此随机数与改进后的速度结合，以增强粒子的局部极值逃逸能力。从算法求解模型方面来说，粒子的位置分量，即 $\bar{p}_i=(\bar{p}_{i1},\bar{p}_{i2},\bar{p}_{i3},\cdots,\bar{p}_{ij})$，代表待检测函数模型中的自变量。若粒子的位置分量变化，此时检测函数中的自变量产生相应变化，最终会促使算法跳出局部停滞点。

改进的侦察学习策略阶段的速度和位置公式如式(8)、式(9)：

$$\bar{v}_{ij}^{(t+1)}=\bar{v}_{ij}^{(t)}+cr_4(\bar{v}_{Rmaxj}-\bar{v}_{Rminj}) \tag{8}$$

$$\bar{p}_{ij}^{(t+1)}=\bar{v}_{\min j}^{(t)}+r_5\bar{v}_{ij}^{(t+1)} \tag{9}$$

$\bar{v}_{Rmax\,j}$、$\bar{v}_{Rmin\,j}$ 代表粒子飞行速度边界；$\bar{v}_{ij}^{(t)}$ 代表粒子 i 第 j 维在 t 迭代时刻的速度分量；$\bar{v}_{\min j}^{(t)}$ 代表粒子第 j 维分量在 t 迭代时刻的位置下限；$\bar{p}_{ij}^{(t+1)}$ 代表粒子 i 第 j 维搜索产生的新位置；r_4、r_5 为[0,1]的随机变量。

虽然如式(9)所示，新产生的粒子位置没有超出粒子的位置下界，但是不排除粒子飞出位置上界的可能。所以，为将粒子的位置限定在 $[V_{\min},V_{\max}]$，在此公式后加入粒子的位置边界处理条件。若粒子位置超出边界，则抛弃掉此粒子，进入下一次搜寻过程，以保证运算结果的有效性和准确性。同时，为了方便记录最优解未被更新的次数，每个最优解都设一个标记参数。若参数的数值达到阈值，则通过式(8)、式(9)产生候选解，并将此参数重置为 0；否则参数加 1，并进入下一次迭代。此策略可有效规避算法过早收敛的问题，有助于算法寻求到更有效的解。

2.3 算法模型

环形拓扑结构可提高粒子收敛于全局最优解的效率，而侦察学习策略可扩大粒子的全局搜索范围。因此，将 PSO 算法与环形拓扑结构以及侦察学习策略的优点相互融合，形成一种基于拓扑结构并且结合侦察学习策略的新型 PSO 算法(BUPSO)。此算法以 PSO 算法为主框架，构建环形结构粒子群体，同时结合侦察学习策略对粒子速度和位置进行改进，以增加粒子种群的多样性，避免 PSO 算法陷入“早熟”问题。BUPSO 算法运行流程如图 3 所示。

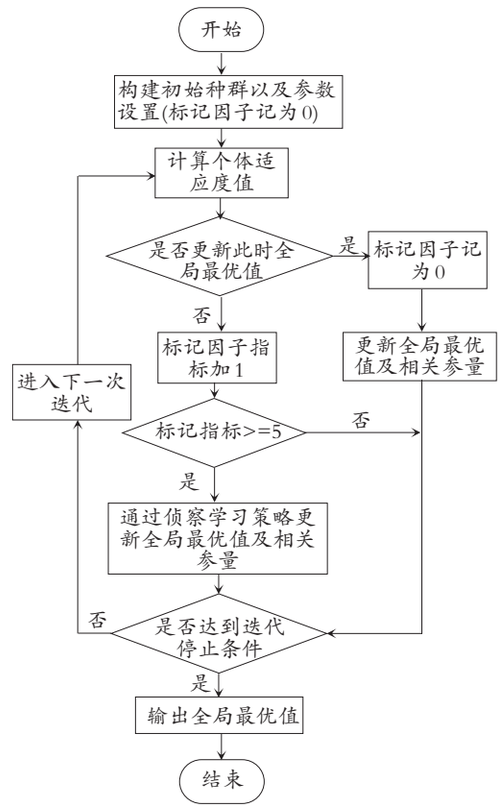


图 3 BUPSO 算法流程图

Fig. 3 Flow chart of BUPSO

其步骤可总结如下：

1) 种群初始化，包括参数设置和种群构建。参数设置包括标记因子、最大飞行速度、种群规模以及粒子边界条件，群体结构设为环形拓扑结构，标记因子阈值设为 5；

2) 计算个体适应度值；

3) 判断是否更新全局最优值；

4) 若更新全局最优值，标记因子重置为 0；否则加 1；

5) 若标记因子重置为 0，则将粒子的全局最优值以及全局最优变量更新，并转到步骤 7；若标记因子加 1，则判断标记因子是否超过阈值；

6) 若标记因子超过 5，则粒子通过侦察学习策略更新全局最优值以及相关参量；否则，进入下一迭代过程，返回步骤 2)；

7) 判断是否达到迭代停止条件。若达到迭代停止条件则输出最优解并结束算法；否则，返回步骤 2)。

2.4 算法复杂度分析

与 PSO 算法相比，BUPSO 算法改进了粒子群体构建方式，并增加了新的学习策略。为计算该算法的复杂性，用 ps 表示粒子种群规模，me 表示最大迭代次数，D 表示决策变量维数，k 为标记因子的阈

值。当通过粒子环形群体进行全局搜索时,计算复杂度为 $T_1(N)=O(ps\times me\times D)$;当通过侦察学习策略产生候选解时,计算复杂度为 $T_2(N)=O(ps\times me\times D\times \frac{1}{k})$。由于在计算复杂度时忽略常数,所以算法整体的计算复杂度为 $T_3(N)=O(ps\times me\times D)+O(ps\times me\times D)$,即 $T_3=O(ps\times me\times D)$。PSO 算法的复杂度为 $T(N)=O(ps\times me\times D)$,因此 BUPSO 算法与 PSO 算法相比,并没有产生额外计算量。

3　实验仿真分析

3.1　实验参数设置

选取 BUPSO 算法、CLPSO 算法[9]、UPSO 算法[10]、PSO 算法、wFIPS 算法在 CEC2017 检测函数[11]旋转环境和非旋转环境下进行收敛性和鲁棒性对比分析。其中,用“*”表示对检测函数进行旋转。选取的检测函数包括:Rosenbrock′s 函数、Griewank′s 函数、Levy 函数、HappyCat 函数、Expanded Griewank′s plus Rosenbrock′s 函数、Zakharov 函数、Rastrigin′s 函数、Expanded Schaffer′s F6 函数。

5 种算法均设置同一参数:种群规模为 30 个粒子、9×10^4 次函数评价、独立运行 20 次。其他参数设置与上文描述一致。

3.2　收敛性能分析

为检验 BUPSO 算法的优化效果,在确定的函数评价次数下对不同算法的收敛特性进行对比。如图 3 和图 4 所示,给出了 5 种算法在同一检测函数下不同环境中的收敛特征图。此外,本文选用两种方式对算法性能测评比较。其一,通过不同算法在同一测试函数下的最小值(Min)、均值(Mean)以及方差(Std)进行比较分析,判定算法的稳定性和搜索精度;其二,在 $\alpha=5\%$的显著水平下,对 50 次独立运算下的 BUPSO 算法与其他 4 种算法进行 Wilcoxon 秩和检验,判定算法在结果上的优越性。其中,p 代表 BUPSO 算法与其他 4 种算法最优结果是否具有显著差异。$p=0$ 表示结果无显著差异,$p=1$ 代表结果有显著差异。以部分函数为例,非旋转环境下的实验结果见表 1,旋转环境下的实验结果见表 2,秩和检验结果见表 3。

表 1　各优化算法对非旋转测试函数的实验结果

Table 1　Experimental results of various optimization algorithms for non-rotating test functions

函　数	维　度	指　标	PSO	CLPSO	UPSO	wFIPS	BUPSO
f_3	20	Min	4. 10E-03	4. 46E-05	3. 23E-03	3. 23E-03	**3. 75E-06**
		Mean	1. 77E-03	4. 84E-03	1. 57E-03	3. 23E-03	**3. 38E-04**
		Std	2. 48E-06	1. 38E-05	2. 96E-06	6. 77E-06	**1. 13E-07**
f_4	20	Min	6. 96E+05	7. 0E+05	6. 73E+05	**4. 97E+05**	6. 96E+05
		Mean	1. 09E+06	9. 67E+05	1. 07E+06	1. 02E+06	**1. 46E+05**
		Std	3. 40E+10	1. 32E+10	2. 87E+10	2. 88E+10	**1. 10E+09**
f_5	20	Min	8. 67E-01	1. 57E+00	9. 32E+00	1. 21E+01	**1. 95E-01**
		Mean	2. 12E+01	8. 77E+00	9. 96E+00	1. 49E+01	**8. 65E+00**
		Std	5. 12E+02	4. 51E+01	7. 80E+00	3. 11E-01	**3. 00E-02**
f_{14}	20	Min	2. 15E-01	1. 75E-01	1. 113E-01	1. 51E+01	**7. 80E-02**
		Mean	3. 91E-01	3. 08E-01	2. 93E-01	2. 98E-01	**2. 70E-01**
		Std	9. 00E-03	1. 00E-02	9. 00E-03	2. 30E-03	**2. 00E-03**

表 2　各优化算法对旋转测试函数的实验结果

Table 2　Experimental results of each optimization algorithm for rotation test functions

函　数	维　度	指　标	PSO	CLPSO	UPSO	wFIPS	BUPSO
f_{2*}	20	Min	5. 71E+01	7. 26E+01	5. 07E+01	7. 92E+01	**2. 84E+01**
		Mean	7. 73E+01	9. 33E+01	7. 68E+01	1. 00E+02	**4. 98E+01**
		Std	1. 40E+02	6. 41E+01	9. 38E+01	6. 22E+01	**5. 32E+01**
f_{6*}	20	Min	8. 81E+01	1. 07E+02	9. 20E+01	1. 22E+02	**6. 64E+01**
		Mean	1. 20E+02	1. 46E+02	1. 23E+02	1. 44E+02	**9. 46E+01**
		Std	1. 55E+02	1. 56E+02	1. 13E+02	1. 03E+02	**5. 05E+01**
f_{14*}	20	Min	7. 78E-12	4. 52E-14	9. 27E-13	1. 27E-12	**7. 77E-16**
		Mean	9. 45E-08	1. 36E-07	1. 13E-07	7. 61E-07	**1. 02E-08**
		Std	3. 19E-14	7. 02E-14	3. 29E-14	3. 1069E-12	**1. 7E-17**

表 3 Wilcoxon 秩和检验 p 值

Table 3 P values of Wilcoxon rank sum test

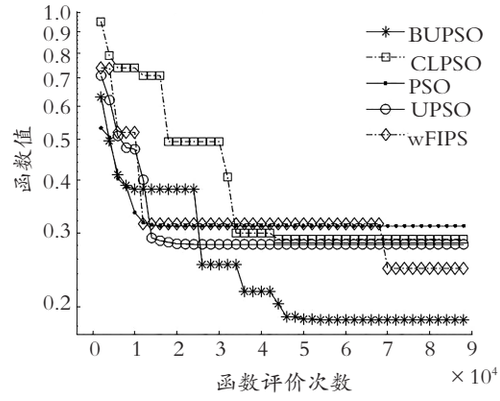
算 法	f_1	f_2	f_3	f_4	f_5	f_{2*}	f_{3*}	f_{4*}	f_{7*}
BUPSO-UPSO	3.07e-02	2.53e-12	8.39e-05	6.37e-03	4.44e-15	8.39e-02	1.30e-02	3.50e-03	3.30e-03
BUPSO-CLPSO	7.20e-03	3.22e-22	1.52e-06	8.32e-08	1.37e-02	1.73e-14	2.51e-14	1.02e-04	7.07e-18
BUPSO-PSO	2.81e-06	2.12e-19	1.80e-03	2.00e-09	8.15e-05	7.92e-16	2.26e-14	2.64e-09	4.16e-12
BUPSO-wFIPS	1.32e-06	7.6847e-12	9.54e-18	0.2868	8.32e-08	2.27e-16	2.02e-16	4.20e-03	9.70e-08
p	0	1	1	0	1	1	1	1	1

如表 1 和表 2 所示:数据表中加粗部分代表各算法在不同测试指标下通过对比产生的最小值。针对这些函数,该算法的值基本上达到最小,说明该算法的性能较优。无论检测函数在旋转环境还是非旋转环境下,从 Min 指标来看,除了旋转环境下的 f_4 检测函数,BUPSO 算法的结果更接近于最优值;从 Mean 指标来看,该算法相比其他算法均值更小,说明运行多次,该新型算法的求解精度更好;从 Std 指标来看,此算法的标准方差更小,说明算法每次运算得到的数值相对较稳定,说明该算法的收敛性能较好,且比较稳定。

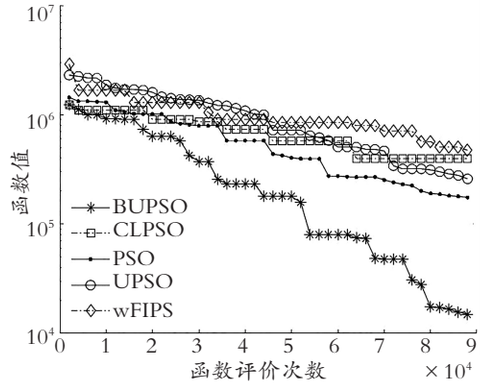
如表 3 所示:将该新型算法与其他算法的最优结果进行 Wilcoxon 秩和检验,通过 p 值发现在对测试函数 f_1 和 f_4 进行秩和检验时,p 值为 0;在其他测试函数进行对比检验,发现其 p 值为 1,说明除了检测函数 f_1 和 f_4,BUPSO 算法结果与其他 4 种算法最优结果的差别在 95%的置信区间内有统计意义,进而表明算法的优越性在统计上是显著的。

如图 4 所示:BUPSO 算法在求解多峰函数方面,尤其对于 Rosenbrock′s 函数、Griewank′s 函数、Levy 函数、HappyCat 函数、Expanded Griewank′s plus Rosenbrock′s 函数、Rastrigin′s 函数、Expanded Schaffer′s F6 函数,在收敛结果和收敛速率方面都表现突出,说明此算法对求解多峰函数具有明显优势;同样,在求解单峰函数方面,对于 Zakharov 函数,发现其收敛精度与收敛速率相较于其他算法,表现更优。

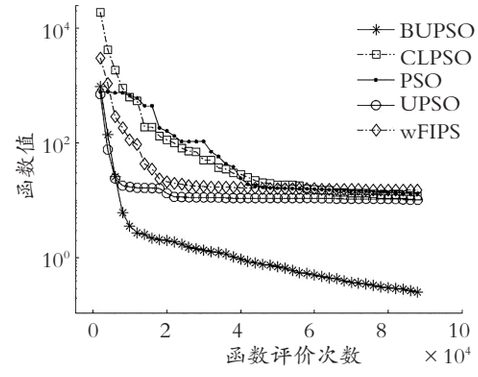
如图 5 所示:在旋转环境下,无论多峰函数还是单峰函数,BUPSO 算法都表现出良好的收敛性能。结合图 4 和图 5,算法在前期过程的收敛曲线明显比其他 4 种算法的收敛曲线下降更快,说明 BUPSO 算法与拓扑结构的结合提升了解的质量。此外,由收敛曲线整体趋势可得,侦察学习策略的引入,明显提高了算法的求解能力,说明 BUPSO 算法相较于其他算法跳出局部最优空间的能力显著。



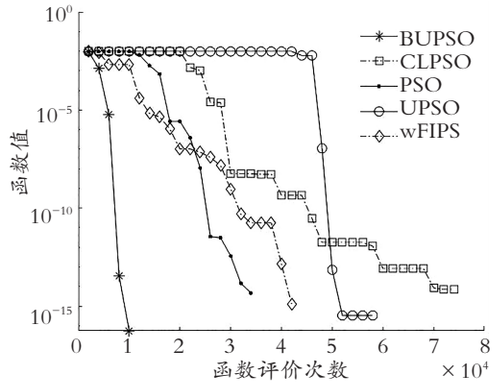
(a) Happy Cat 函数



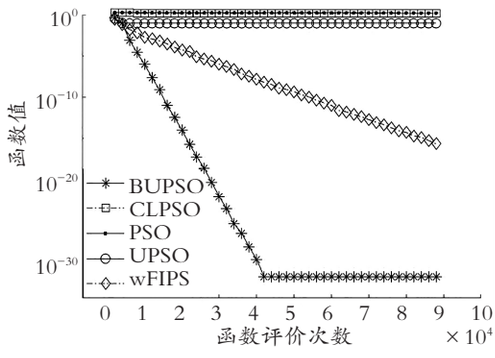
(b) Zakharov 函数



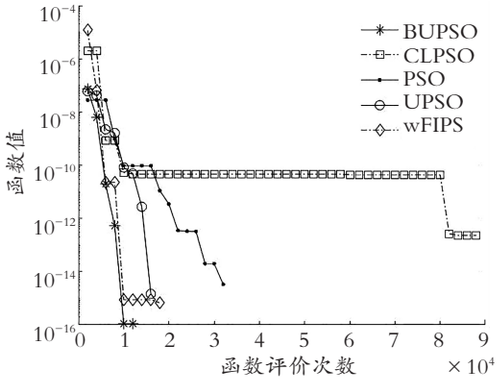
(c) Rosenbrock′s function 函数



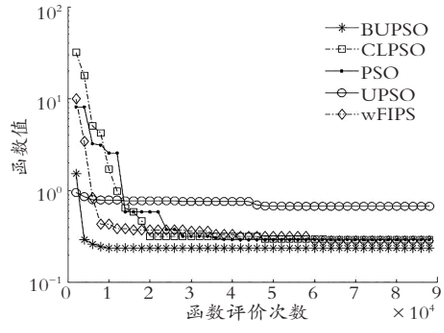
(d) Expanded Schaffer's F6 函数



(e) Levy 函数



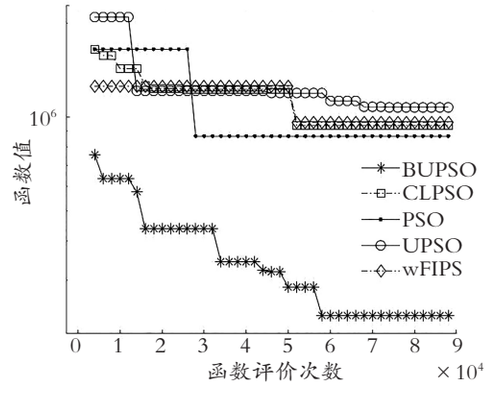
(f) Expanded Griewank's plus Rosenbrok's 函数



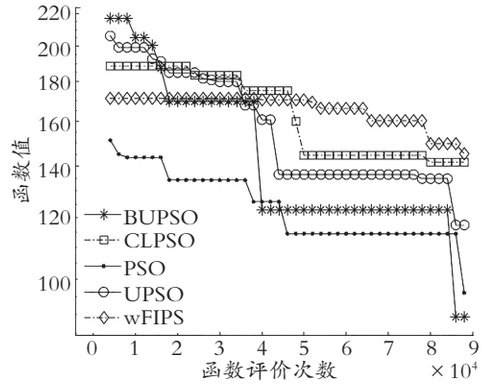
(g) HGBat 函数

图 4　非旋转检测函数收敛特征图

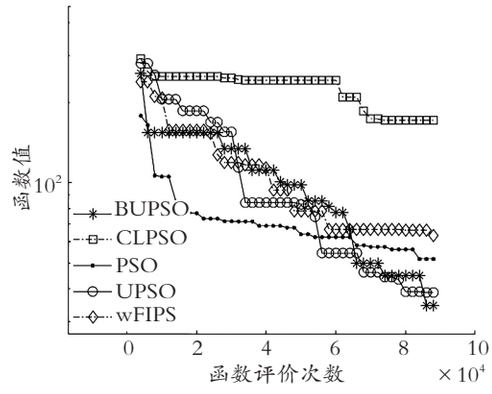
Fig. 4　Convergence features of non-rotating detection functions



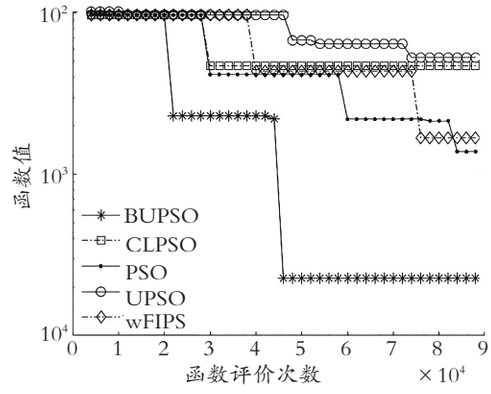
(a) Zakharov 函数



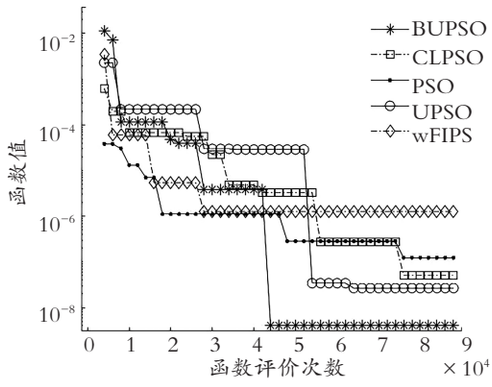
(b) Non-continuous Rotated Rastrigin's 函数



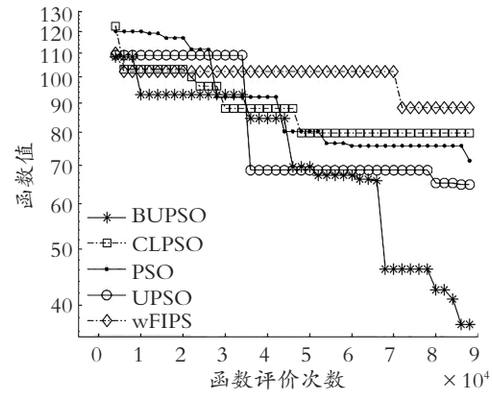
(c) Grivwank's 函数



(d) Expanded Schaffer's F6 函数



(e) EXpanded Griewank's plus Rosenbrok's 函数



(f) Rastrigin's 函数

图5 旋转检测函数收敛特征图

Fig. 5 Convergence features of rotation detection functions

3.3 鲁棒性分析

为探测5种算法的运行稳定性,每次运行都选用同一测试函数在不同的环境下进行测评比较。以Griewank's函数、Zakharov函数、Expanded Schaffer's F6函数为例,表4给出了算法的鲁棒性分析。这里的鲁棒性分析指各算法独立运行50次,在算法评价次数内所得到的全局最小值达到给定阈值的水平。*FEs*代表达到给定阈值时算法的评价次数,*S*代表在算法评价次数内所得到的全局最小值达到给定阈值的比例,给定的阈值分别为1e-15、2.5e+04、1e-15、50、2e-04、1e+06。

如表4所示:在非旋转环境下,对于Griewank's函数,CLPSO算法和BUPSO算法都100%达到了阈值,但BUPSO算法比CLPSO算法用了较少的函数评价次数,对于Zakharov函数和Expanded Schaffer's F6函数,仅BUPSO算法成功达到阈值且评价次数最少;在旋转环境下,对于Expanded Schaffer's F6函数,CLPSO算法与BUPSO算法都达到了阈值,对于Zakharov函数和Griewank's函数,只有BUPSO算法达到了阈值,且该算法对Zakharov函数的评价次数最少。

总之,不论从算法的收敛特征结果还是从算法的鲁棒性分析结果来看,BUPSO算法都表现出了较好的收敛特性和稳定性。

表4 各种算法的鲁棒性分析

Table 4 Robustness analysis of various algorithms

算 法	f_3		f_4		f_7		f_{3^*}		f_{4^*}		f_{7^*}	
	FEs	*S*	*FEs*	*S*	*FEs*	*S*	*FEs*	*S*	*FEs*	*S*	*FEs*	*S*
PSO	80 580	88	30 720	48	59 700	10	90 000	100	90 000	46	90 000	12
CLPSO	90 022	10	28 429	44	67 410	100	92 052	64	90 258	40	90 845	0
wFIPS	90 000	6	33 480	54	90 000	34	90 000	44	90 000	60	90 000	10
UPSO	90 000	66	21 690	52	13 320	58	93 000	36	81 000	72	72 000	34
BUPSO	66 000	100	19 890	100	21 450	100	90 000	100	69 000	100	93 000	100

4 结束语

PSO算法只有单一阶段,存在收敛能力不足,粒子跳出局部最优空间动力不足,解的精度不高等弊端。为解决这些问题,提出了基于侦察学习策略的BUPSO算法。该算法利用联合因子对算法的全局搜索能力和局部搜索能力进行均衡,提高了算法的收敛能力和运算精度;当粒子陷入局部收敛时,该算法引入侦察学习策略,对该粒子的速度和位置进行更新,提高了粒子跳出局部最优解的能力。通过Wilcoxon秩和检验和基准函数寻优实验结果显示,该算法有一定的有效性和优越性,既保持了PSO算法简单、并行性高等特点,又在一定程度上提高了算法的收敛能力、局部极值逃逸能力和求解能力,具有深远的研究意义。

参考文献(References):

[1] KEENEDY J, EBERHART R C. Particle swarm optimization[C]//Proceedings of IEEE Conference on Neural Networks. New York:IEEE,1995:1942—1948.

[2] MAHMOUD P, HERIS M N, IVATLOO B M, et al. A hybrid genetic particle swarm optimization for distributed generation allocation in power distribution networks[J/OL]. Energy,2020, 209 (C):S0360544220313256[2021-

08-11]. https://www.sciencedirect.com/science/article/pii/S0360544220313256.

[3] ZHAO L, ZHI W Q, ZHU P, et al. An adaptive switchover hybrid particle swarm optimization algorithm with local search strategy for constrained optimization problems[J/OL]. Engineering Applications of Artificial Intelligence,2020, 95(5):103771[2021-08-11]. https://www.sciencedirect.com/science/article/pii/S095 2197620 301718.

[4] CHEN X, TIANFIELD H, DU W. Bee-foraging learning particle swarm optimization[J/OL]. Applied Soft Computing, 2021, 102(11): 107134 [2021-08-11]. https://www.sciencedirect.com/science/article/pii/S15 68494621000570.

[5] HAIPING M, DAN S, PATRICK S, et al. Biogeography-based optimization:a 10-year review[J]. IEEE Transactions on Emerging Topics in Computational Intelligience, 20171 (5):391—407.

[6] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Compotation, 2004, 7(8): 204—210.

[7] VAN D B F, ENGELBRECHT P A. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Compotation, 2004, 6(8): 225—239.

[8] KENNEY J, MENDES R. Population structure and particle swarm performance [C]//Proceedings of Congress on Evolutionary Computation. IEEE, 2002: 1671—1676.

[9] LIANG J J, QIN A K. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3):281—295.

[10] PARSOPOULOS K E, VRAHATIS M N. Unified particle swarm optimization in dynamic[J]. Lecture Notes in Computer Sciense, 2005, 3449 (2):590—599.

[11] AWAD N H, ALI M Z, SUGANTHAN P N, et al. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization [R]. Singapore: Nanyang Technological University, 2016.

Particle Swarm Optimization Based on Scout Learning Strategy

ZHANG Qian[1], LIU Yan-min[2], YANG Mei-lan[1], SHU Xiao-li[3]

(1. School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China;
2. School of Mathematics, Zunyi Normal University, Guizhou Zunyi 563006, China;
3. School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China)

Abstract: Aiming at the problem of insufficient convergence ability of particle swarm optimization and easy to fall into local optimum, a novel particle swarm optimization based on scout learning strategy is proposed. The algorithm first uses the topology structure to construct the particle population, and then uses the local search ability and global search ability of the joint factor equalization algorithm, and improves the speed and position formula of the algorithm through the scout learning strategy to generate candidate solutions. The results of Wilcoxon rank sum test and CEC2017 benchmark function test show that this novel particle swarm algorithm has better convergence ability, optimal solution precision and algorithm stability, indicating that the performance of the algorithm has been improved.

Key words: particle swarm optimization; topology structure; scout learning strategy

责任编辑:陈　芳

引用本文/Cite this paper:

张倩,刘衍民, 杨妹兰,等. 基于侦察学习策略的粒子群算法优化[J]. 重庆工商大学学报(自然科学版),2022,39(5):34—41.

ZHANG Qian, LIU Yan-min, YANG Mei-lan, et al. Particle swarm optimization based on scout learning strategy[J]. Journal of Chongqing Technology and Business University (Natural Science Edition),2022,39(5):34—41.