

基于种群分区的多策略综合粒子群优化算法

李冰晓, 万睿之, 朱永杰, 赵新超

(北京邮电大学 理学院, 北京 100876)

摘要:针对标准粒子群优化算法容易陷入局部最优收敛精度不高的问题,提出一种基于种群分区的多策略综合学习粒子群优化算法(MSPSO).该算法利用竞争机制将种群分为两个子种群:潜力子群与普通子群,对这两个子群实行不同的进化策略,潜力子群中的粒子主要负责全局探索,普通子群中的粒子则侧重于局部勘探.为验证算法的性能,在不同类型的基准函数上与其他粒子群算法及其他群智能算法进行对比,所提算法都能取到最优的平均结果,证明所提算法具有更优异的算法性能.

关键词:粒子群优化;竞争机制;多策略学习;种群分区;综合学习

中图分类号:TP301.6

文献标志码:A

粒子群优化算法^[1]是由 KENNEDY 和 EBERHART 于 1995 年提出的一种全局优化算法,它源于对鸟群觅食行为的模拟.自提出以来,粒子群优化算法以其参数少、概念简单、易于实现等优点引起众多研究者关注,并在数据聚类^[2]、图像处理^[3]、特征选择^[4]、电力系统^[5]、人工神经网络^[6]等实际问题中得到了广泛应用.

然而,粒子群算法容易早熟收敛,算法后期多样性不足等问题限制了它的性能.为改善上述问题,众多学者做了大量的研究工作,主要分为四个方面:参数调整^[7]、拓扑结构改善^[8]、改进学习策略^[9]和与其他算法相结合.文献[10]提出了 CLPSO,算法中的粒子以一定概率向其他粒子的历史最佳经验学习,极大地增加了种群多样性;文献[11]提出了一种简化而高效的粒子群优化算法,算法中对粒子的学习样本以一定概率施加均匀扰动,能够有效改善算法陷入局部极值,增大找到最优解的概率;文献[12]提出了一种基于扰动全局最优概念的新粒子更新策略的扰动粒子群优化算法(pPSA),以解决粒子群中的早熟收敛和多样性保持问题;文献[13]提出了一种新的多阶段扰动差分进化算法(MPDE),采用方向性差异信息策略和多参数自适应实现了一种新的变异策略“多阶段扰动”;文献[14]提出一种基于分段优势学习的 SLPSO 算法,算法中的整个种群被分为精英子群与普通子群,普通子群中粒子分段向精英种群中的粒子进行学习,使得算法能够充分利用精英粒子的信息,避免了早熟收敛;文献[15]将差分变异操作与 SLPSO 算法相结合提出了差分变异与新型社会学习粒子群优化算法(DSPSO);文献[16]在 SLPSO 算法的启发下提出了带有混合变异策略的多种群协作粒子群优化算法(MPCPSO);文献[17-21]都采用了多种群的策略对各个子群进行不同的进化策略,研究结果表明多种群的多策略进化是改善粒子群算法性能的一个有效研究方向.

目前对粒子群算法的研究使得算法性能得到了不同程度的提高,多种群操作以及不同扰动策略的引进有效改善了粒子群算法收敛精度不高的问题.然而,在很多问题上粒子群陷入局部最优以及早熟收敛的问题还没有得到很好的解决.针对此现状,本文提出一种基于种群分区的多策略综合学习粒子群优化算法(MSPSO).在每次迭代中,利用一种新的竞争机制将种群分为两个子群:潜力子群与普通子群,对这两个子群采取不同的学习策略.在潜力子群中,粒子的进化除惯性部分外,还向该子群中随机选取的某粒子的历史最佳经验学习,这种学习策略能够在充分利用优秀粒子精英信息的同时保持种群多样性.在普通子群中,设置

收稿日期:2021-09-17;**修回日期:**2022-02-13.

基金项目:国家自然科学基金(61973042);北京市自然科学基金(1202020).

作者简介:李冰晓(1997-),女,河北邯郸人,北京邮电大学硕士研究生,研究方向为粒子群优化及其应用.

通信作者:赵新超(1976-),男,河南商丘人,北京邮电大学教授,博士生导师,研究方向为群体智能、运筹优化及其应用.

E-mail:zhaoxc@bupt.edu.cn.

了两种学习策略.算法前期,粒子向潜力子群中随机选取的某粒子的历史最佳经验学习;算法后期,则以一定概率直接向全局最优经验学习.这种进化方式既保持了两个子群的协同进化和种群多样性,又加快了收敛速度.实验结果表明,所提算法具有较快的收敛速度,并且能够有效防止陷入局部极值,获得极有竞争力的解.

1 相关工作

1.1 标准 PSO 算法

PSO 算法首先随机初始化一群粒子,然后通过飞行速度和位置的逐步更新迭代找到最优解.在每一次迭代中,粒子通过跟踪个体发现的最好极值和群体发现的全局极值更新自己的飞行速度和位置.假设在 D 维搜索空间中,有 N 个粒子组成一个群体, t 时刻粒子 i 由位置向量 $(x_{i1}, x_{i2}, \dots, x_{iD})$ 和速度向量 $(v_{i1}, v_{i2}, \dots, v_{iD})$ 表示.在优化过程中,下一代种群中个体的速度和位置向量由如下公式更新,

$$\begin{aligned} V_{ij}(t+1) &= \omega V_{ij}(t) + c_1 r_1 (P_{ij}(t) - X_{ij}(t)) + c_2 r_2 (P_{gj}(t) - X_{ij}(t)), \\ X_{ij}(t+1) &= X_{ij}(t) + V_{ij}(t+1). \end{aligned} \quad (1)$$

其中 $i=1, 2, \dots, N, j=1, 2, \dots, D, N$ 为群体规模, ω 为惯性权重, r_1, r_2 为 $(0, 1)$ 区间均匀分布的随机数, c_1, c_2 为加速系数. $P_{ij}(t)$ 为粒子 i 的历史最佳位置的第 j 维分量, $P_{gj}(t)$ 为整个种群的历史最佳位置的第 j 维分量, t 为当前迭代次数.

1.2 CLPSO 算法

与标准 PSO 算法不同的是, CLPSO 算法中的粒子以概率 $(1-p_c)$ 向个人历史最佳经验学习, 以概率 p_c 向由竞争机制选择出的其他粒子的历史最佳经验学习. 粒子更新过程由如下公式产生: $V_{ij}(t+1) = \omega V_{ij}(t) + cr(P_{f_i(j)}(t) - X_{ij}(t))$, 其中 ω 为惯性权重, r 为 $(0, 1)$ 区间内的随机数, c 为加速系数, $f_i(j)$ 为由 CLS 构建的学习样本的索引. 下面是 CLPSO 构建学习样本的过程.

对于粒子 i 的每一维度 j : [1]产生随机数 $p, p \in (0, 1)$; [2]若 $p \leq p_c$, 粒子向由竞争机制选择出的其他粒子的历史最佳经验学习; [3]若 $p > p_c$, 粒子向其个体历史最佳经验学习.

与标准 PSO 算法相比, 粒子可以向不同的粒子历史最佳个体学习, 有效保证了种群多样性, 提供了大范围而有潜力的搜索区域.

1.3 Nmp3PSO 算法

文献[22]提出一种基于非均匀变异和多阶段扰动的粒子群优化算法(Nmp3PSO).该算法在执行的不同阶段利用对当前最优解施加大小不同的邻域扰动操作, 并且引入非均匀变异运算适应性调整解向量的搜索步长, 具体操作如下.

1.3.1 非均匀变异算子

假设对粒子 $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}^T$ 的第 d 个分量执行变异运算, 而 x_{id} 的下界和上界分别记为 LB 和 UB , 则变异后的分量:

$$x'_{id} = \begin{cases} x_{id} - \Delta(t, UB - x_{id}), & r < 0.5, \\ x_{id} - \Delta(t, x_{id} - LB), & r \geq 0.5, \end{cases}$$

其中 $\Delta(t, y) = y \cdot (1 - r)^{\frac{t}{T}}$, t 是循环变量, T 是最大的循环变量, r 是 $(0, 1)$ 间均匀产生的随机数, b 是系统参数.

1.3.2 最优粒子扰动策略及粒子更新

对全局最优粒子 $gbest$ 依据方差可调的正态随机分布进行扰动得到新的最优粒子 $pgbest$, 然后选定的待更新粒子向 $pgbest$ 学习, 速度更新公式为:

$$\begin{aligned} p'_{gd} &= N(P'_{gd}, \sigma), \\ v'_{id} &= \omega v'_{id} + c_1 r_1 (p'_{id} - x'_{id}) + c_2 r_2 (p'_{id} - x'_{id}), \end{aligned}$$

其中 p'_{gd} 表示第 t 代扰动后的最优粒子 $pgbest$ 的第 d 维分量, $pgbest$ 由正态分布 $N(P'_{gd}, \sigma)$ 产生, σ 更新为:

$$\sigma = \begin{cases} \sigma_1, t < \alpha_1 T, \\ \sigma_2, \alpha_1 T \leq t < \alpha_2 T, \\ \sigma_3, t \geq \alpha_2 T, \end{cases}$$

其中 $\sigma_1 > \sigma_2 > \sigma_3$ 表示正态扰动幅度的半径参数; α_1, α_2 是半径变化的控制参数,且 $\alpha_1 < \alpha_2$; t 是当前函数值计算次数, T 是最大函数值计算次数。

算法性能分析表明, Nmp3PSO 能够较好兼顾群体优化算法的多样性和精英学习强度之间的平衡问题。

1.4 其他群体智能算法

文献[23]提出了一种自适应差分进化算法(SaDE),算法中的试验向量生成策略及其相关的控制参数通过学习它们以前生成的潜力解的经验而逐渐实现自适应,从而自适应地确定更合适的进化策略及参数设置来匹配搜索过程的不同阶段.实验结果与传统 DE^[24]以及几个先进的参数自适应 DE 变体相比, SaDE 能够获得更高质量的解、更小的标准差以及更高的成功率。

文献[25]提出了一种遗传算法的变体(GL25),算法在交叉操作之前执行了3个过程.首先是雌性和雄性的分化过程,它决定了种群中可能成为雌性或雄性父母的个体;然后采用两种不同的选择机制从每个群体中选择雌雄亲本;最后讨论了最适合的进化模型的选择以及从这些亲本选择机制中获益.实验表明这3个过程可以增强以父代为中心的交叉算子的操作性能。

2 多策略综合学习粒子群优化算法(MSPSO)

2.1 研究动机

对粒子群体中不同适应值的粒子信息如何个性化的进行开发和利用是本文最初的出发点,逐步具体化为如何设置一种新的竞争规则将整个种群划分成不同的子群,对携带精英信息较多的子群如何在加强精英搜索的基础上引进较多的群体多样性,对携带差异化信息较多的群体如何在扩大全局搜索的基础上适当加速算法的收敛,同时保证整个种群及多个子群能够保持协同进化来平衡算法的全局搜索能力和局部开发能力。

2.2 MSPSO 算法

首先设计合理的竞争机制将整个种群分为两个子群,对这两个子群进行不同偏向的进化操作。

为了利用群体进化过程中多种关键信息,本文考虑划分不同子群的两个指标:一是每个粒子相邻两次迭代适应值的改变量,二是当前迭代适应值的大小.粒子前后两次迭代适应值的改变量越大,说明该方向潜力越大;粒子当前适应值越大,也说明该方向潜力越大.竞争机制构建的两个指标 α 和 β 分别设置如下,

$$\alpha = \frac{(F(i) - F_N(i))}{\sum_{k=1}^N (F(k) - F_N(k))}, \quad (2)$$

$$\beta = \frac{F_N(i)}{\sum_{k=1}^N F_N(k)}, \quad (3)$$

$$R = \lambda_1 \alpha + \lambda_2 \beta. \quad (4)$$

综合竞争指标 R 既考虑解的改变量,又考虑解的优异程度,因此从算法搜索状态和搜索性能两方面构建了启发式信息.对 R 进行降序排列,前 1/2 作为潜力子群,后 1/2 作为普通子群.其中 F, F_N 分别是粒子前后两次迭代的适应值, λ_1, λ_2 为参数,本文初步设置为 1 和 3; α 初始化为 0。

针对潜力子群,粒子 i 的飞行速度和位置向量在第 j 维上的更新公式如下:

$$V_{ij}(t+1) = \omega V_{ij}(t) + r_3(\text{rand } n * P_{kj}(t) - X_{ij}(t)), \quad (5)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1), \quad (6)$$

其中 ω 为惯性权重,更新公式为 $0.5 \times (1 - FES/\max FES) + 0.15$. FES 为当前函数评价次数, $\max FES$ 为最大评价次数. P_k 为该子群中随机选择的粒子的历史最佳经验. r_3 为随机性参数,为保证(5)式中学习部分的有效性,此参数不宜过小,故设置为 $0.45 + r$,其中 r 为(0,1)区间内均匀分布的随机数, $\text{rand } n$ 为满足标准

高斯分布的随机数.对学习样本施加高斯扰动,目的是增加种群多样性,加大粒子的学习范围,研究动机是在精英综合学习的基础上增加一定的群体多样性,即选择多样化的精英搜索路径.

针对普通子群,粒子 i 的飞行速度和位置向量在第 j 维上的更新公式如下,

$$V_{ij} = \begin{cases} \omega V_{ij}(t) + r_1(P_{ij}(t) - X_{ij}(t)), & \text{若 } \text{rand} < \mu, \\ \omega V_{ij}(t) + r_1(P_{gj}(t) - X_{ij}(t)), & \text{若 } \text{rand} \geq \mu, \end{cases} \quad (7)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1), \quad (8)$$

其中 ω, r_1 的设置如(1)式; P_l 为潜力子群中随机选择的粒子 l 的历史最佳经验; P_g 为全局历史最佳经验; μ 为参数,设置为 $1/(1+(FES/N)^{(1/2)} \times \exp(i/N))$. 随着算法的进行,粒子飞向全局历史最佳经验的概率逐渐增加,能够加速算法收敛;对普通子群粒子更新公式的研究动机是在保持种群多样性的基础上加速算法的收敛,并且排名越靠后的粒子飞向全局历史最佳经验的概率越大,有助于加速粒子的收敛速度,从而提高整个种群解的质量.

2.3 算法步骤

步骤 1 初始化粒子种群规模 N 、粒子搜索空间 D 、函数最大评价次数 $\max FES$ 等算法参数,随机初始化粒子的飞行速度和位置;

步骤 2 初次按照适应值升序排序分为两个子群,分别执行不同搜索偏向的速度和位置更新操作,转到步骤 4;

步骤 3 按照竞争机制将整个种群分为两个子群,对不同子群中的粒子分别执行对应进化操作;

步骤 4 对溢出边界粒子进行修正操作;

步骤 5 执行位置更新公式;

步骤 6 保留粒子历史最佳经验,全局最佳经验;

步骤 7 若函数最大评价次数达到设置值,输出结果;否则回到步骤 3.

3 对比仿真实验与结果分析

3.1 测试函数与参数设置

为了验证所提出的 MSPSO 算法的性能,将 MSPSO 与标准 PSO, CLPSO, Nmp3PSO, 标准差分算法 DE, SaDE, 以及 GL25 这 6 种算法在 CEC2005 的 11 个经典基准测试函数上进行对比分析,为方便起见将 CEC 中的测试函数重新编号为 F1~F11,如表 1 所示.

表 1 基准测试函数

Tab. 1 The benchmark function

函数	搜索区间	最优值	函数	搜索区间	最优值
F1=Shifted Sphere Function	$[-100,100]^n$	-450	F6=Shifted Rosenbrock's Function	$[-100,100]^n$	390
F2=Shifted Schwefel's Problem 1.2	$[-100,100]^n$	-450	F7=Shifted Shifted Rastrigin's Function	$[-5,5]^n$	-330
F3=Shifted Rotated High Conditioned Elliptic Function	$[-100,100]^n$	-450	F8=Shifted Rotated Rastrigin's Function	$[-5,5]^n$	-330
F4=Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100,100]^n$	-450	F9=Shifted Rotated Weierstrass Function	$[-0.5,0.5]^n$	90
F5=Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100,100]^n$	-310	F10=Schwefel's Problem 2.13	$[-100,100]^n$	-460
			F11=Expanded Extended Griewank's plus Rosenbrock's Function(F8F2)	$[-3,1]^n$	-130

其中 F1~F5 为单峰函数,用于检验算法的收敛精度和速度;F6~F11 为多峰函数,用于检验算法的全局搜索能力.在实验中,5 种算法的种群规模都设置为 30,搜索空间维数都设置为 30,函数最大评估次数设置为 300 000,粒子最大飞行速度 $V_{\max} = \frac{1}{5} X_{\max}$, MSPSO 的其余参数如上文, PSO, CLPSO, Nmp3PSO, DE, SaDE, GL25 的其余参数如参考文献所述.为减少实验的随机性,将所有算法在 11 个基准测试函数上分别独

立运行 30 次,实验结果如表 2 所示.

表 2 7 种算法数值实验结果

Tab. 2 The numerical experiments results of 7 algorithms

函数	指标	数值结果						
		PSO	CLPSO	Nmp3PSO	MSPSO	DE	SaDE	GL25
F1	最优值 Min	0.051 27	0.001 8	1.26E-11	2.90E-95	4.20E-31	2.33E-26	9.14E-31
	平均值 Mean	55.552 8	0.003 7	1.80E-11	1.97E-81	0.207 8	9.80E-24	1.42E-23
	标准差 STD	92.461 1	0.001 2	3.01E-12	7.20E-81	1.138 2	2.30E-23	7.50E-23
F2	最优值 Min	8.384 8	0.004 9	1.82E-05	3.54E-52	5.49E-19	1.87E-16	7.39E-15
	平均值 Mean	40.622 2	0.008 9	0.001 7	2.96E-47	9.05E-06	1.81E-15	9.68E-09
	标准差 STD	24.388 4	0.001 8	0.005 1	9.63E-47	4.72E-05	1.16E-15	3.67E-08
F3	最优值 Min	662.911 5	375 4.967 0	3.79E-06	6.48E-69	337 1.156 0	0.861 1	79.364 1
	平均值 Mean	740 6.026 0	637 8.928 0	8.68E-05	1.89E-58	105 44.560 0	10.737 5	433.133 6
	标准差 STD	422 6.430 0	125 4.643 0	8.14E-05	9.21E-58	417 3.082 0	9.399 4	311.247 1
F4	最优值 Min	22.755 4	17.167 5	0.040 5	2.05E-40	0.010 7	0.001 8	1.664 9
	平均值 Mean	34.912 0	21.588 5	3.181 7	2.11E-34	4.821 1	0.758 2	6.097 9
	标准差 STD	5.599 8	1.872 7	3.343 3	9.56E-34	6.656 6	1.156 4	2.631 0
F5	最优值 Min	505.067 7	418.554 5	1.073 6	22.948 5	18.308 3	14.009 0	5.396 9
	平均值 Mean	825 41.390 0	725.401 5	43.634 6	23.886 0	125.515 6	72.399 4	133.587 2
	标准差 STD	171 699.300 0	226.848 2	51.472 8	1.358 0	311.153 1	53.232 4	138.046 6
F6	最优值 Min	84.000 0	0	0	0	0	0	0
	平均值 Mean	959.233 3	0	0.133 3	0	0	0	5.966 7
	标准差 STD	784.243 5	0	0.345 8	0	0	0	11.278 6
F7	最优值 Min	77.864 0	2.035 6	52.732 7	0	2.104 1	8.40E-09	12.905 7
	平均值 Mean	127.277 5	4.473 4	91.038 5	0	15.368 3	0.156 8	45.714 5
	标准差 STD	25.979 7	1.202 8	24.845 4	0	10.292 4	0.643 0	45.505 2
F8	最优值 Min	6.727 9	0.021 5	2.813 8	8.88E-16	7.99E-15	5.77E-14	3.29E-14
	平均值 Mean	11.195 4	0.037 1	6.012 8	8.88E-16	0.166 8	0.168 9	0.508 7
	标准差 STD	2.642 0	0.007 6	1.507 2	0	0.407 4	0.390 0	0.625 5
F9	最优值 Min	0.568 1	0.008 0	1.13E-12	0	0	0	0
	平均值 Mean	1.902 0	0.019 1	0.087 4	0	0.031 3	0.007 5	0.011 7
	标准差 STD	2.151 4	0.008 0	0.132 8	0	0.126 8	0.013 6	0.016 7
F10	最优值 Min	5.583 9	3.88E-05	4.62E-13	6.89E-08	1.40E-30	6.16E-28	3.49E-16
	平均值 Mean	15.086 9	0.000 1	4.308 0	2.62E-07	0.052 1	0.003 5	0.111 4
	标准差 STD	6.254 3	4.53E-05	3.709 8	4.00E-07	0.126 4	0.018 9	0.228 3
F11	最优值 Min	24.447 3	0.001 1	2.14E-12	3.45E-07	4.69E-28	5.70E-26	1.37E-17
	平均值 Mean	40.211 1	0.002 5	14.907 7	0.037 8	32.768 6	0.000 4	0.166 8
	标准差 STD	12.997 5	0.001 0	11.551 3	0.079 5	178.890 0	0.002 0	0.407 0
小计	最优值 Min	0	1	2	8	4	2	2
	平均值 Mean	0	2	0	10	1	1	0
总计	—	0	3	2	18	5	3	2

注:粗黑体表示最优的结果.

表 2 包含各算法运行最终所达到的最优值 Min、均值 Mean 和标准差 STD,另外小计部分显示在 11 个测试函数上各算法的最优值 Min 及均值 Mean 能达到相对最优结果的个数,总计部分显示各算法综合考虑

最优值 Min 以及均值 Mean 能达到相对最优结果的个数.各算法的收敛曲线如图 1 所示.

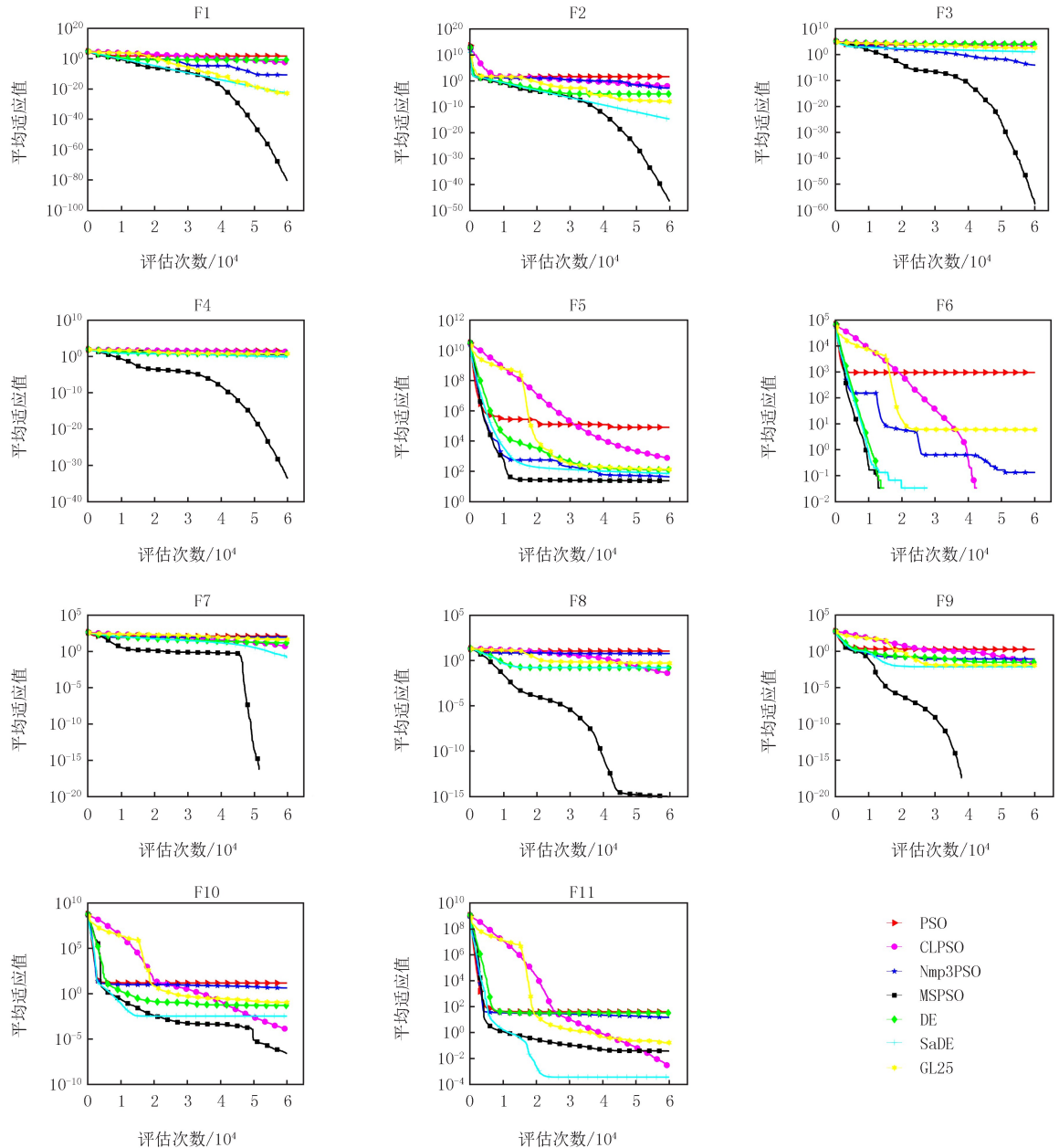


图1 7种算法的收敛曲线图

Fig.1 The Convergence graphs of 7 algorithms

3.2 算法仿真结果与分析

由表 2 可以看出,1)PSO,CLPSO,Nmp3PSO,MSPSO,DE,SaDE 和 GL25 在 11 个测试函数最终结果的最优值上分别取得 0 个、1 个、2 个、8 个、4 个、4 个和 2 个最优结果;2)PSO,CLPSO,Nmp3PSO,MSPSO,DE,SaDE 和 GL25 在 11 个测试函数最终结果的平均值上分别取得 0 个、2 个、0 个、10 个、1 个、1 个和 0 个最优结果;3)结合最优值与平均值来看 PSO,CLPSO,Nmp3PSO,MSPSO,DE,SaDE 和 GL25 分别取得 0 个、3 个、2 个、18 个、5 个、3 个和 2 个最优结果.综合表 2 和相应分析可以看出,在这 11 个基准测试函数上,MSPSO 算法相对优于其他对比算法.

图 1 给出 7 种算法在 11 个测试函数上平均适应值的进化性能对比,可以看出, MSPSO 比 PSO,CLPSO,Nmp3PSO,DE,SaDE 以及 GL25 有着较为明显的进化优势,说明 MSPSO 算法有着更快的收敛速度和更高的收敛精度.综上所述,本文提出的 MSPSO 算法相比其他对比算法有着较为优异的性能.

Friedman 检验^[26]是利用秩实现对多个总体分布是否存在显著差异的非参数检验方法.为讨论 7 个算法平均结果对比统计的显著性,对算法所得结果的平均值做 Friedman 统计检验,结果如表 3 所示.由此可见, MSPSO 对比 PSO, CLPSO, Nmp3PSO, DE, SaDE 以及 GL25 有显著性差异.

表 3 7 种算法的 Friedman 统计检验结果

Tab. 3 The Friedman Statistical Test Results of 7 Algorithm

算法	MSPSO	PSO	CLPSO	Nmp3PSO	DE	SaDE	GL25
平均秩	1.318 2	6.909 1	3.954 5	4.545 5	4.500 0	2.409 1	4.363 6

3.3 高斯扰动效用测试

为验证 MSPSO 算法对于潜力子群中局部历史最佳经验施加高斯扰动的有效性,本文设置了一组对照试验,其中 MSPSO1 为不加高斯扰动的对照组,其余设置与 MSPSO 相同.两种算法的测试函数仍然为表 1 中的 11 个经典基准函数,种群规模、粒子最大飞行速度、函数最大评估次数等参数也与上文保持一致.图 2 为 MSPSO 与 MSPSO1 在各测试函数上的收敛曲线对比图.

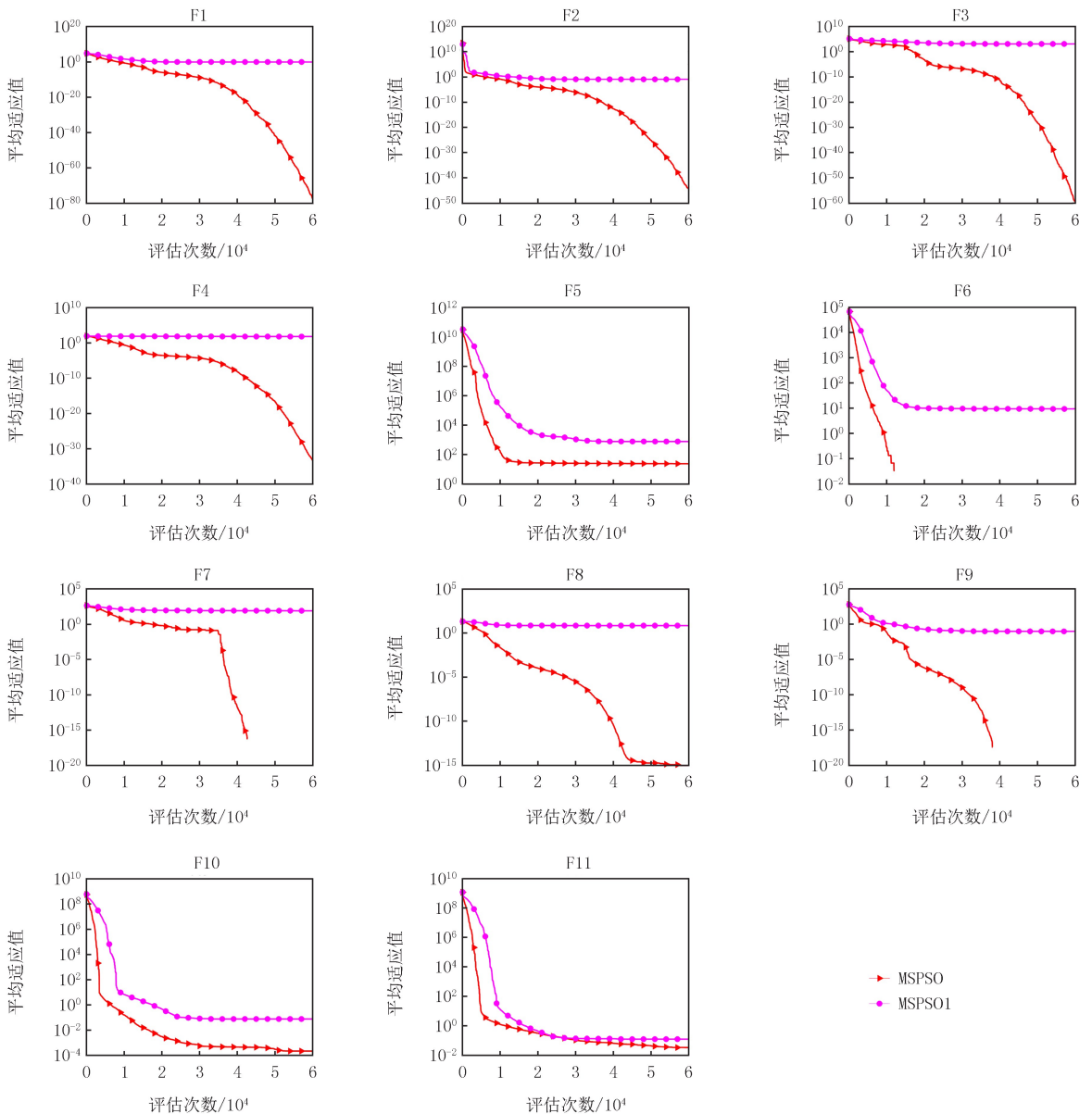


图2 MSPSO与MSPSO1收敛曲线图

Fig.2 The Convergence graphs of MSPSO and MSPSO1

从图 2 可以看出 MSPSO 在收敛速度和收敛精度上都比 MSPSO1 具有明显的优势.因此 MSPSO 中对

于潜力子群中优秀粒子的局部历史最佳经验施加高斯扰动是非常有效的,提升了算法找到最优解的能力.

3.4 竞争机制排序的参数分析

为检验 MSPSO 算法中竞争机制所考虑两个指标权重 λ_1, λ_2 的参数敏感性,本文设置了几组对比试验,其中 MSPSO2, MSPSO3, MSPSO4 和 MSPSO5 分别将 (λ_1, λ_2) 设置为 $(1, 2)$ 、 $(1, 1)$ 、 $(2, 1)$ 和 $(3, 1)$,其余设置与 MSPSO 相同.这 5 种算法的测试函数为由表 1 中选取的 3 个单峰函数 F1, F2 和 F5, 3 个多峰函数 F9~F11.

表 4 为这 5 种算法在 6 个测试函数上达到的平均值对比.由表 4 可看出这 5 种算法都能获得较高精度的解,说明不同的 λ_1, λ_2 设置对算法性能影响不大,但探索其更好的设置值能为算法提供较好的稳定性.本文中 MSPSO 算法的设置值要略好于其他 4 种对比算法.

表 4 不同设置值算法的平均值对比

Tab. 4 Comparison of the average value of different parameter setting

函数	平均值					函数	平均值				
	MSPSO	MSPSO2	MSPSO3	MSPSO4	MSPSO5		MSPSO	MSPSO2	MSPSO3	MSPSO4	MSPSO5
F1	6.42E-81	5.49E-82	4.09E-85	3.12E-93	3.33E-94	F9	0	0	0	0	0
F2	8.82E-47	1.82E-46	2.24E-47	1.55E-58	1.45E-60	F10	1.97E-07	6.66E-07	1.07E-03	3.95E-03	7.11E-03
F5	23.33	24.51	24.63	25.84	25.91	F11	1.14E-02	6.29E-02	1.06E-01	1.92E-01	2.913E-01

3.3 节对局部历史最佳经验高斯扰动的有效性进行了分析,表明算法对高斯扰动操作对潜力子群局部历史最佳经验具有较明显的影响提升;3.4 节对两个重要指标权重 λ_1, λ_2 参数敏感性和鲁棒性进行了分析,结果表明指标权重参数 λ_1, λ_2 对算法性能的影响不明显.除此之外,除了 30 维的测试结果,又补充了问题维数 50 和 100 时的结果,见表 5,可以看出 MSPSO 算法的数值结果并未随着测试函数维数上升而有明显下降,说明 MSPSO 算法较好的稳定性.结合 3.2 节算法仿真结果分析认为 MSPSO 算法性能不仅具有更高的收敛精度和更快的收敛速度,还具有较好的鲁棒性.

表 5 MSPSO 在不同维度测试函数上的数值结果

Tab. 5 The numerical experiments results of MSPSO in the benchmark function of different dimensions

函数	指标	函数维度			函数	指标	函数维度		
		30 维	50 维	100 维			30 维	50 维	100 维
F1	最优值 Min	2.90E-95	3.36E-96	2.48E-94	F7	最优值 Min	0	0	0
	平均值 Mean	1.97E-81	5.50E-82	3.34E-82		平均值 Mean	0	0	0
	标准差 STD	7.20E-81	1.90E-81	1.07E-81		标准差 STD	0	0	0
F2	最优值 Min	3.54E-52	1.94E-52	5.26E-52	F8	最优值 Min	8.88E-16	8.88E-16	8.88E-16
	平均值 Mean	2.96E-47	1.27E-47	3.05E-46		平均值 Mean	8.88E-16	8.88E-16	1.13E-15
	标准差 STD	9.63E-47	3.67E-47	1.27E-45		标准差 STD	0	0	9.01E-16
F3	最优值 Min	6.48E-69	2.15E-68	1.04E-70	F9	最优值 Min	0	0	0
	平均值 Mean	1.89E-58	2.07E-58	8.59E-57		平均值 Mean	0	0	0
	标准差 STD	9.21E-58	9.96E-58	4.66E-56		标准差 STD	0	0	0
F4	最优值 Min	2.05E-40	9.33E-40	8.17E-40	F10	最优值 Min	6.89E-08	7.05E-08	7.62E-08
	平均值 Mean	2.11E-34	1.57E-34	3.24E-35		平均值 Mean	2.62E-07	0.000 218	1.75E-07
	标准差 STD	9.56E-34	7.83E-34	9.30E-35		标准差 STD	4.00E-07	0.001 195	8.16E-08
F5	最优值 Min	22.948 5	22.615 9	22.928 1	F11	最优值 Min	3.45E-07	1.81E-07	4.86E-07
	平均值 Mean	23.886 0	23.863 7	23.992 7		平均值 Mean	0.037 8	0.041 343	0.019 826
	标准差 STD	1.358 0	1.252 1	1.427 6		标准差 STD	0.079 5	0.055 106	0.047 962
F6	最优值 Min	0	0	0					
	平均值 Mean	0	0	0					
	标准差 STD	0	0	0					

4 结 论

为改善粒子群算法早熟收敛、收敛速度慢的问题,提出了基于竞争机制和种群分区的多策略综合学习算法(MSPSO)。MSPSO 算法通过一种有效的竞争机制将整个种群划分为两个子群,对这两个子群采用了不同的学习策略,一个子群在精英学习的基础上引进更多种群多样性,另一子群在保持多样性的基础上加速了算法收敛,并且两个子群体和整个群体保持协同进化,较好地平衡了算法的寻优能力与收敛速度。在 11 个经典基准函数上对 MSPSO,PSO,CLPSO,Nmp3PSO,DE,SaDE,GL25 算法的性能进行了比较。结果表明,本文所提出的 MSPSO 算法性能具有更高的收敛精度和更快的收敛速度,并且具有较好的鲁棒性。

参 考 文 献

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. Piscataway: IEEE Press, 1995.
- [2] ALSWAIITI M, ALBUGHDADI M, ISA N A M. Density-based Particle Swarm Optimization Algorithm for Data Clustering[J]. Expert Systems with Applications, 2018, 91: 170-186.
- [3] YADAV S, EKBAL A, SAHA S. Feature Selection for Entity Extraction from Multiple Biomedical Corpora: A PSO-based Approach[J]. Soft Computing, 2018, 22(20): 6881-6904.
- [4] JORDEHI A R. Particle swarm optimisation(PSO) for allocation of FACTS devices in electric transmission systems: a review[J]. Renewable and Sustainable Energy Reviews, 2015, 52: 1260-1267.
- [5] PANG H, LIU F, XU Z R. Variable Universe Fuzzy Control for Vehicle Semi-active Suspension System with MR Damper Combining Fuzzy Neural Network and Particle Swarm Optimization[J]. Neurocomputing, 2018, 306: 130-140.
- [6] WANG F, ZHU H Q, LI W, et al. A Hybrid Convolution Network for Serial Number Recognition on Banknotes[J]. Information Sciences, 2020, 512: 952-963.
- [7] CHEN K, ZHOU F Y, LIU A L. Chaotic Dynamic Weight Particle Swarm Optimization for Numerical Function Optimization[J]. Knowledge-Based Systems, 2018, 139: 23-40.
- [8] 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.
- [9] LIU H R, CUI J C, LU Z D, et al. A Hierarchical Simple Particle Swarm Optimization with Mean Dimensional Information[J]. Applied Soft Computing, 2019, 76: 712-725.
- [10] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [11] 胡旺,李志蜀.一种更简化而高效的粒子群优化算法[J].软件学报, 2007, 18(4): 861-868.
HU W, LI Z S. A Simpler and More Effective Particle Swarm Optimization Algorithm[J]. Journal of Software, 2007, 18(4): 861-868.
- [12] ZHAO X C. A Perturbed Particle Swarm Algorithm for Numerical Optimization[J]. Applied Soft Computing, 2010, 10(1): 119-124.
- [13] XU G Z, LI R, HAO J L, et al. A new multi-stage perturbed differential evolution with multi-parameter adaption and directional difference [J]. Natural Computing, 2020, 19(4): 683-698.
- [14] YANG Q, CHEN W N, GU T L, et al. Segment-Based Predominant Learning Swarm Optimizer for Large-Scale Optimization[J]. IEEE transactions on cybernetics, 2017, 47(9): 2896-2910.
- [15] ZHANG X M, WANG X, KANG Q, et al. Differential Mutation and Novel Social Learning Particle Swarm Optimization Algorithm[J]. Information Sciences, 2019, 480: 109-129.
- [16] LI W, MENG X, HUANG Y, et al. Multipopulation Cooperative Particle Swarm Optimization with A Mixed Mutation Strategy[J]. Information Sciences, 2020, 529: 179-196.
- [17] 王永骥,苏婷婷,刘磊.基于柯西变异的多策略协同进化粒子群算法[J].系统仿真学报, 2018, 30(8): 2875-2883.
WANG Y J, SU T T, LIU L. Multi-strategy cooperative evolutionary PSO based on cauchy mutation strategy[J]. Journal of System Simulation, 2018, 30(8): 2875-2883.
- [18] 梁静,刘睿,于坤杰,等.求解大规模问题协同进化动态粒子群优化算法[J].软件学报, 2018, 29(9): 2595-2605.
LIANG J, LIU R, YU K J, et al. Dynamic Multi-Swarm Particle Swarm Optimization with Cooperative Coevolution for Large Scale Global Optimization[J]. Journal of Software, 2018, 29(9): 2595-2605.
- [19] 邓先礼,魏波,曾辉,等.基于多种群的自适应迁移 PSO 算法[J].电子学报, 2018, 46(8): 1858-1865.
DENG X L, WEI B, ZENG H, et al. A Multi-Population Based Self-Adaptive Migration PSO[J]. Acta Electronica Sinica, 2018, 46(8): 1858-1865.

- [20] 方景远,季益胜,赵新超.基于高斯分布估计的对位差分进化算法[J].河南师范大学学报(自然科学版),2021,49(3):27-32.
FANG J Y,JI Y S,ZHAO X C.Opposition-based differential evolution algorithm with Gaussian distribution estimation[J].Journal of Henan Normal University(Natural Science Edition),2021,49(3):27-32.
- [21] 赵新超,刘子阳.差分和扰动混合的多策略粒子群优化算法[J].计算机科学与探索,2014,8(2):218-225.
ZHAO X C,LIU Z Y.Hybrid Particle Swarm Optimization with Differential and Perturbation[J].Journal of Frontiers of Computer Science and Technology,2014,8(2):218-225.
- [22] 赵新超,刘国莅,刘虎球,等.基于非均匀变异和多阶段扰动的粒子群优化算法[J].计算机学报,2014,37(9):2058-2070.
ZHAO X C,LIU G L,LIU H Q,et al.Particle Swarm Optimization Algorithm Based on Non-Uniform Mutation and Multiple Stages Perturbation[J].Chinese Journal of Computers,2014,37(9):2058-2070.
- [23] QIN A K,HUANG V L,SUGANTHAN P N.Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization[J].IEEE Transactions on Evolutionary Computation,2009,13(2):398-417.
- [24] STORN R,PRICE K.Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces[J].Journal of Global Optimization,1997,11(4):341-359.
- [25] GARCÍA-MARTÍNEZ C,LOZANO M,HERRERA F,et al.Global and local real-coded genetic algorithms based on parent-centric crossover operators[J].European Journal of Operational Research,2008,185(3):1088-1113.
- [26] 陶澍.应用数理统计方法[M].北京:中国环境出版社,1994.

Multi-strategy comprehensive article swarm optimization algorithm based on population partition

Li Bingxiao, Wan Ruizhi, Zhu Yongjie, Zhao Xinchao

(School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In order to solve the problem that the canonical particle swarm algorithm is prone to fall into the local optima and is of low convergence accuracy, a multi-strategy comprehensive learning particle swarm optimization algorithm(MSPSO) is proposed based on population partition. In MSPSO the population is divided into potential subgroup and common subgroup based on the competition mechanism. Both subgroups use different evolutionary strategies. The particles in potential subgroup are mainly responsible for global exploration, while particles in common subgroup focus on local exploitation. In order to verify the performance of the proposed algorithm, MSPSO is compared with other PSO-based competitors and other swarm intelligence algorithms. It gets the best average results on different benchmark functions, and it is proved that the proposed algorithm has better performance.

Keywords: particle swarm optimization; competition mechanism; multi-strategy learning; population partition; comprehensive learning

[责任编辑 陈留院 赵晓华]