一种邻域重心反向学习的粒子群优化算法

周凌云[1,2],丁立新[1],彭 虎[3],强小利[2]
(1. 武汉大学软件工程国家重点实验室 武汉大学计算机学院,湖北武汉 430072;
2. 中南民族大学计算机科学学院,湖北武汉 430074;
3. 九江学院信息科学与技术学院,江西九江 332005)

摘 要: 粒子群优化算法使用反向学习技术可以提高性能. 然而,现有的反向学习粒子群优化算法仅采用粒子最大最小边界计算反向解,没有充分利用群体搜索经验. 针对此问题,提出了一种邻域重心反向学习策略,使用邻域重心作为参考点计算反向解,充分吸收群体搜索经验的同时保持种群多样性;采用收缩因子拓展反向解搜索范围,增加找到更高质量解的机率. 在典型的基准测试函数、CEC'13 测试函数和一个实际工程优化问题上进行验证,实验结果说明了邻域重心反向学习策略的有效性和本文算法的竞争力.

关键词: 反向学习; 邻域重心; 多样性; 粒子群优化
中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2017)11-2815-10
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.11.032

Neighborhood Centroid Opposition-Based Particle Swarm Optimization

ZHOU Ling-yun[1,2], DING Li-xin[1], PENG Hu[3], QIANG Xiao-li[2]
(1. *State Key Lab of Software Engineering, Wuhan University, Computer School, Wuhan University, Wuhan, Hubei* 430072, *China*;
2. *College of Computer Science, South-Central University for Nationalities, Wuhan, Hubei* 430074, *China*;
3. *School of Information Science and Technology, Jiujiang University, Jiujiang, Jiangxi* 332005, *China*)

Abstract: Using opposition-based learning can improve the performance of particle swarm optimization (PSO) algorithm. However, the current opposition-based learning particle swarm optimization algorithms calculate the opposite solution by using coordinates of the candidate solution, the maximum and the minimum of a population, without making full use of the search experience of the population. A neighborhood centroid opposition-based learning strategy is proposed to improve this issue. First, the neighborhood centroid is used as reference point for the generation of the opposite particle, absorbing the population search experience and maintaining diversity. Second, contraction factor is used to expand the reverse search space, increasing the probability of finding a better solution. Experiments are conducted on typical benchmark functions, CEC 13 test functions and also on a practical engineering optimization problem. The results verify the effectiveness of the neighborhood centroid opposition-based learning and the competitiveness of the NCOPSO.

Key words: opposition-based learning; neighborhood centroid; diversity; particle swarm optimization (PSO)

1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是一种群智能优化算法[1]. PSO 因简单易实现而获得广泛关注,并已成功应用到神经网络训练、模糊控制等多个领域[2]. 为获得更好的性能,学者们对 PSO 进行了许多改进。这些改进主要分为三类:一是参数控制和拓扑结构设计,如 Shi 等人在速度更新中引入惯性权重,平衡算法的全局探索与局部勘探[3]. Mendes 等人结合不同的拓扑结构,充分利用邻居最优位置信息提高算法收敛速度和精度[4]. 二是速度或位置更新规则的设计,如 CLPSO 算法[5]中粒子速度的更新是参照一个不同维度从各粒子学习而得到的样例. 三是与智能方法结合,反向学习(Opposite-Based Learning, OBL)就是其中一种.

OBL 应用到 PSO 算法中表现出很好的效果. Wang 等提出一种反向学习粒子群优化算法 OPSO,其中 OBL 不仅以一定的概率用于算法迭代过程中粒子位置的更

收稿日期:2016-08-16;修回日期:2017-05-12;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61379059)

新,也用于选择种群初始化位置[6]. Chi 等人提出了一种反向扰动的概念,当个体最优更新时,根据一定概率执行反向扰动,增加了种群多样性[7]. 这方面比较有代表性的工作还有将反向计算一般化提出通用反向学习粒子群优化算法[8];利用透镜成像原理扩展反向粒子学习位置[9];用折射原理改进粒子反向学习策略[10]. OBL 因其简单有效,已成为 PSO 算法中最常使用的学习策略之一[11]. 然而,纵观这些研究工作,反向粒子的计算均采用搜索空间每一维上的最大最小边界,没有充分考虑到群体作为一个寻优整体所携带的有利信息. Rahnamayan 等人提出重心反向学习(Centroid Opposition-Based Learning,COBL),以整个群体重心为参考点计算反向点,这样反向点就包含了群体搜索经验. 将 COBL 应用到差分演化算法中,实验结果表明这种模型比传统反向学习模型产生了更好的效果[12].

受 COBL 的启发,本文提出了一种邻域重心反向学习的粒子群优化算法(Neighborhood Centroid Opposition-Based Particle Swarm Optimization,NCOPSO). 以邻域重心作为参考点计算反向粒子,从而在充分利用群体搜索经验的同时保持种群多样性. 其次,采用缩放因子拓展反向空间搜索范围,增加找到高质量解的概率. 实验结果表明,NCOPSO 算法对多数测试函数和实际工程优化问题——扩频雷达相位编码问题都表现出明显优势.

2 基本粒子群优化算法与重心反向计算

2.1 基本粒子群优化算法

基本 PSO 算法中,粒子通过学习自身历史经验(个体最优 ***pbest***)与群体经验(全局最优 ***gbest***)以达到寻优目的. 对于一个 D 维的搜索空间,设群体粒子个数为 n,第 i 个粒子的位置和速度分别表示为 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{iD})$ 和 $\boldsymbol{V}_i=(v_{i1},v_{i2},\cdots,v_{iD})$,$i=1,2,\cdots,n$. 第 i 个粒子的 $d(1\leqslant d\leqslant D)$ 维速度与位置的计算公式分别为式(1)和式(2).

$$v_{id}(t+1)=\omega v_{id}(t)+c_1r_1(\boldsymbol{pbest}_{id}-x_{id}(t))+c_2r_2(\boldsymbol{gbest}_d-x_{id}(t)) \tag{1}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \tag{2}$$

其中,$v_{id}(t+1)$ 和 $x_{id}(t+1)$ 分别为粒子 i 在第 $t+1$ 代的速度和位置. ω 是惯性权重,c_1 和 c_2 表示学习系数,r_1 和 r_2 是[0,1]间均匀分布的随机数. 式(1)中的 ***gbest*** 若表示群体最优,则是全局 PSO 算法;若 ***gbest*** 表示局部最优,则是局部 PSO 算法.

2.2 重心反向计算

OBL 是 Tizhoosh 提出的一种智能技术,主要思想是同时评估当前解和其反向解,择优使用,以此来加速搜索进程[13]. 传统 OBL 是采用最大最小边界来计算反向点,这种反向学习模型应用到 PSO 中已获得较好的结果[11]. 但这种反向模型没有利用整个种群的搜索信息,由此 Rahnamayan 等人提出 COBL,重心与基于重心的反向点定义[12]如下:

定义 1 重心. 设 $(\boldsymbol{X}_1,\cdots,\boldsymbol{X}_n)$ 是 D 维搜索空间中带有单位质量的 n 个点,则整体的重心定义为

$$M=\frac{X_1+\cdots+X_n}{n} \tag{3}$$

即得:

$$M_j=\frac{\sum_{j=1}^{D}x_{i,j}}{n},i=1,2,\cdots,n \tag{4}$$

定义 2 重心反向点. 若一个离散均匀的整体的重心为 $\boldsymbol{M}$,则该整体中某一点 $\boldsymbol{X}_i$ 的反向点定义为

$$\bar{X}_i=2\times M-X_i,i=1,2,\cdots,n \tag{5}$$

反向点位于一个具有动态边界的搜索空间,记为 $x_{i,j}\in[a_j,b_j]$. 动态边界可以让反向点位于一个不断缩小的搜索空间中. 动态边界可以按式(6)计算.

$$a_j=\min(x_{i,j}),\quad b_j=\max(x_{i,j}) \tag{6}$$

反向点超出边界时,按式(7)重新计算反向点.

$$\bar{x}_{i,j}=\begin{cases}a_j+rand(0,1)\times(M_j-a_j), & \text{if } \bar{x}_{i,j}<a_j\\ M_j+rand(0,1)\times(b_j-M_j), & \text{if } \bar{x}_{i,j}>b_j\end{cases} \tag{7}$$

其中,$rand(0,1)$ 是[0,1]上的一个随机数.

3 邻域重心反向学习的粒子群优化算法

3.1 邻域重心反向学习策略

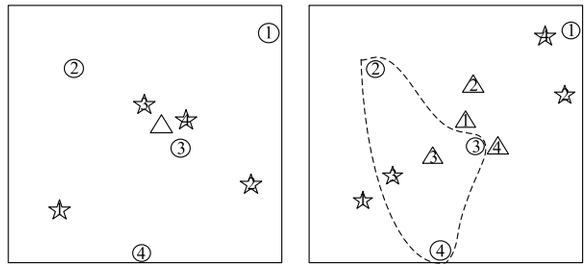
3.1.1 邻域重心反向计算

COBL 的优势主要在于计算反向点的参考点是群体重心,它携带了群体的搜索经验,克服了 OBL 的不足. 然而,COBL 用来计算反向解的参考点只有一个重心,没有考虑促进反向解的多样性. 因此 COBL 还有进一步提升的空间.

如果用多个重心来计算反向解,那么可以在利用群体搜索经验的同时也能保持一定的多样性. 局部 PSO 算法[14]中,粒子根据拓扑结构划分成多个邻域,邻域内的粒子直接相互影响,然后这种影响通过邻居粒子再扩散到其它邻域. 这种方法的优点是不同邻域的小群体能够搜索问题空间中的不同区域. 受此启发,本文提出邻域重心反向计算,一个邻域计算一个重心,邻域内粒子以邻域重心为参考点计算反向粒子,这样可以充分利用邻域这个小群体的搜索经验,同时又有多个参考点,计算的反向解可以探索更多的搜索空间.

为了直观地说明重心反向计算与邻域重心反向计算的区别,将它们进行对比,如图 1 所示. "○"表示个体,"△"表示重心或邻域重心,"☆"表示反向解. 图 1(*b*)中的虚线区域表示其中一个领域.

重心反向计算如图 1(*a*). 算法运行初期,个体位置



(a)重心反向解　　(b)邻域重心反向解

图1　重心反向解与邻域重心反向解的区别

较分散,以群体重心为参考点计算反向解,能充分利用群体搜索经验,具有较好的全局搜索能力. 随着迭代不断进行,个体位置逐渐集中,群体多样性降低,群体经验会加速反向解的靠拢,使得反向解也集中在一个较小的搜索空间,算法转为局部寻优. 这个过程中,没有充分利用个体携带的有用信息,缺少了多样性. 而邻域重心反向计算中,每个邻域计算一个重心,邻域内的个体采用邻域重心作为参考点计算反向解,如图 1(*b*)所示. 算法执行前期,个体差异较大,邻域重心反向解不但能够利用群体搜索经验,还可以探索问题空间中更多区域. 算法执行后期,群体多样性减少时,此时邻域重心反向解能更多的保留有用个体信息,保持多样性.

3.1.2　拓扑结构与收缩因子

邻域重心计算与拓扑结构相关,但最佳拓扑结构往往依赖于求解问题. 在不了解具体问题的情况下,随机拓扑结构往往优于固定拓扑结构[15]. 本文提出的邻域重心反向计算中采用随机拓扑结构,随机选择粒子的邻居,当群体最优解没有更新时,重新选择邻居.

值得注意的是,随机拓扑结构某个邻域中粒子可能仅包含它自己,或除它以外仅包含另 1 个粒子. 这两种情况下,按照(5)式计算的反向粒子会与粒子自身或邻域中的另 1 个粒子重合. 这样对反向粒子的评估就会造成重复,从而降低算法的收敛速度. 此外,对一些 OBL 的研究工作[16]已说明通过一个系数拓展反向搜索空间有利于提升反向学习的性能. 基于以上分析,我们提出在邻域重心计算中引入收缩因子 k,下面给出邻域重心反向解的定义.

定义 3　领域重心反向解. 设 $\boldsymbol{X}_i$ 是群体 X 中的 i 个个体,$M_i^{'}$ 是它所在邻域的重心,则邻域重心反向解 X_i^* 可定义为

$$X_i^* = 2 \times k \times M_i^{'} - X_i, i = 1,2,\cdots,m_i \tag{8}$$

其中,m_i 表示个体 i 所在邻域内个体的数目. k 是[0,1]之间均匀分布的随机数. 它拓展了反向搜索范围,避免了因重合而造成的重复评估.

3.2　NCOPSO 算法

本节把提出的领域重心反向计算应用到 PSO 中,提出一种邻域重心反向学习的粒子群优化算法,简称为 NCOPSO. 算法 1 描述了 NCOPSO.

算法 1　NCOPSO

① Initialize the population $\boldsymbol{X}_i$ and $\boldsymbol{V}_i$ of population size N;
② Calculate the fitness of particles $f(X)$;
③ Build neighbor matrix;
④ Calculate the neighborhood centroid M';
⑤ Calculate opposite of initial population OX according to (8)
⑥ Check for opposite particle boundary constraint violations
⑦ Calculate the fitness of opposite particles $f(\mathrm{OX})$;
⑧ for $i = 1$ to N do
⑨　if $f(OX_i) < f(X_i)$ then $X_i = OX_i$;end if
⑩ end for
⑪ while $FE \leqslant FE\_Max$
⑫　if (no improvement in the best solution) then
⑬　　Rebuild particle neighbor matrix;
⑭　end if
⑮　if rand < JR then
⑯　　Calculate the neighborhood centroid M';
⑰　　Calculate opposite particles OX according to (8);
⑱　　Update the dynamic interval boundaries according to (6);
⑲　　Check for boundary constraint violations according to (7);
⑳　　Calculate the fitness of opposite particles $f(\mathrm{OX})$;
㉑　　for $i = 1$ to N do
㉒　　　if $f(OX_i) < f(X_i)$ then $X_i = OX_i$; end if
㉓　　end for
㉔　else
㉕　　Update the vecolity and position according to (1) and (2);
㉖　　Check for particle boundary constraint violations
㉗　　Calculate the fitness of particles $f(X)$;
㉘　end if
㉙　Update the personal best and the the global best;
㉚ end while

NCOPSO 算法中要保存一个 $N \times N$ 的邻居矩阵,用来记录粒子间的邻居关系. 算法采用与 SPSO 2011 算法相同的自适应随机拓扑[15]. 邻域重心反向学习采用动态边界. 动态边界按照式(6)计算. 对反向粒子的越界处理按照式(7)计算,即当粒子出界时,将生成一个邻域重心到对应边界之间的随机点作为粒子位置. JR 是一个参数,表示 NCOBL 策略的执行概率,文献[17]中有对 *JR* 的讨论,一般取 0.3.

3.3　算法复杂度分析

设目标函数的维度为 D,种群规模为 N,最大迭代次数为 T. 初始化阶段,第③步计算邻居矩阵的时间复杂度为 $O(NN)$,其他步的时间复杂度都是 $O(ND)$. 迭代过程,⑫－⑭步时间复杂度为 $O(NN)$,其他步的时间复杂度都是 $O(ND)$,则⑪－㉚步总的时间复杂度为 $O(T(ND + NN))$. 因此,NCOPSO 总时间复杂度为 O

$(ND)+O(NN)+O(TN(D+N))$,仅与 T,N 和 D 有关. 因问题维度 D 往往大于或近似于群体个数 N,所以,略去低阶项,NCOPSO 算法的时间复杂度为 $O(TND)$,与 PSO 算法的时间复杂度一致.

4 仿真实验与结果分析

4.1 测试函数与相关设置

为了全面客观地对 NCOPSO 算法的性能做出评价,我们选择了具有不同解空间结构的优化函数. 它们分为 5 组,分别具备不同的特性. 第 1 组是普通单峰函数 Sphere,Step,Rosenbrock 和 Quartic function,i. e. noise,分别记为 f_1 ~f_4. 第 2 组是普通多峰函数 Rastrigin,Ackley,Griewank 和 Normalized Schwefel,分别记为 f_5 ~f_8. 这两组函数是演化算法最常使用的基准测试,具体定义见文献[18,19]. 第 3、4 和 5 组是 CEC'13 建议的测试函数,可以更好地代表广泛的实际优化问题,具体定义见文献[20]. 其中第 3 组是单峰函数,分别记为 f_9 ~f_{13},第 4 组是多峰函数,分别记为 f_{14} ~f_{28},第 5 组是组合函数,分别记为 f_{29} ~f_{36}. 所有 CEC'13 测试函数的全局最优解都做了偏移,且每组测试函数都包括了旋转函数.

从三类 PSO 变种算法中选择有代表性的与本文算法进行比较. 一类是惯性权重粒子群算法[3],记为 PSO,和标准粒子群算法 SPSO 2011[15],记为 SPSO. 第二类是有代表性的基于 OBL 的 PSO 算法:OPSO[6] 和 GOPSO[8]. 第三类是国际上有一定影响力的知名 PSO 算法:FIPS[4]、CLPSO[5]. 实验中,种群规模 $N=40$,最大函数评估次数设为 100000,问题维数 $D=30$,NCOPSO 的参数 K 取与 SPSO 2011 算法中相同的值 3,NCOPSO 中的参数 JR 取与 OPSO、GOPSO 相同的值 0.3,比较算法的其他参数与原文献相同. 迭代终止条件设为到达最大迭代次数,每个算法独立运行 25 次,记录平均误差(算法找到的最优解与函数最优值之差的均值(Mean))与标准差(SD).

4.2 实验结果分析

表 1 中显示了 NCOPSO 算法与各比较算法的平均误差、标准差和 Wilcoxon 秩和检验结果,符号"-","+","≈"分别表示比较算法劣于、优于和相当于 NCOPSO 算法的结果. 该表中 Wilcoxon 秩和检验用来分析每一个比较算法与 NCOPSO 算法在各测试函数上独立运行 25 次的结果之间是否存在显著差异. 本文实验中 Wilcoxon 秩和检验的显著性水平设为 0.05. 图 2 是各算法随机选择一次运行结果的收敛曲线,表现了算法多数情况下的收敛行为,受篇幅限制,仅选择了其中 6 个函数上的收敛曲线. 图 3 则显示了所有算法在 5 个分组上和整个测试集上的 Friedman 检验排名结果,用来对多个算法比较排名,值越小的排名越好.

第 1 组测试函数 f_1 和 f_4 上,NCOPSO 获得的解具有最高精度. 在 f_2 上,GOPSO、FIPS、CLPSO 和 NCOPSO 都找到了全局最优解. f_3 上,NCOPSO 取得的最优值均值劣于 CLPSO、SPSO 和 GOPSO,但与之相差不大. 另外,NCOPSO 的标准差都是最小的,说明 NCOPSO 求解单峰函数时的稳定性较好. NCOPSO 展现出快收敛速度和高求解精度,这一点从图 2(a)和(b)上也可看出. 从图 3 可看出,NCOPSO 在这组函数上的 Friedman 检验排名第一. 其主要原因是邻域重心反向学习策略充分利用了群体搜索经验,从而具有更快的学习速度. 另一方面邻域重心反向学习策略中的收缩因子也进一步强化了局部区域的搜索能力.

第 2 组测试函数除了 f_8 上,NCOPSO 的收敛精度均胜过其它算法. 其中,f_5 是具有大量局部极值的复杂多峰函数,f_7 的各变量相互关联,但这两个函数上 NCOPSO 都找到了全局最优解. f_6 上 NCOPSO 取得的解精度也远高于其它算法. 从图 2(c)和(d)可以看出,NCOPSO 在 f_5 和 f_6 上的收敛曲线快速下降,这是因为邻域重心学习策略在算法运行初期探索解空间中多个区域,一旦发现最有希望的区域之后,粒子能迅速收敛到最优区域. f_8 函数上 CLPSO 取得了最好结果,可能是因为 f_8 具有很深且远离全局最优的局部极值. 相比 CLPSO,NCOPSO 因搜索空间不够充足,从而在 f_8 上陷入局部极值,但 NCOPSO 能在全局搜索和局部搜索之间取得较好的平衡. 这一组其他函数上,NCOPSO 都比 CLPSO 的收敛速度快且收敛精度高. 这得益于邻域重心反向学习策略在吸收群体搜索经验的同时保持了算法的多样性.

表 1 各算法结果的均值与标准差和 Wilcoxon 秩和检验结果(Mean ± SD)

	PSO	SPSO	OPSO	GOPSO	FIPS	CLPSO	NCOPSO
f_1	3.00e+03 ± 4.83e+03 -	1.17e-63 ± 1.30e-63 -	2.21e-28 ± 5.24e-28 -	1.86e-27 ± 3.25e-27 -	2.19e-04 ± 1.22e-04 -	3.70e-08 ± 1.83e-08 -	8.77e-135 ± 1.84e-134
f_2	4.00e+03 ± 5.16e+03 -	4.00e-01 ± 5.16e-01 -	3.00e-01 ± 9.49e-01 -	2.00e-01 ± 4.22e-01 -	0.00e+00 ± 0.00e+00 ≈	0.00e+00 ± 0.00e+00 ≈	0.00e+00 ± 0.00e+00
f_3	1.71e+02 ± 2.50e+02 -	2.51e+01 ± 1.47e+00 +	1.97e+01 ± 3.34e+00 +	2.04e+01 ± 1.14e+00 +	2.58e+01 ± 5.05e-01 -	2.40e+01 ± 3.52e+00 +	2.54e+01 ± 1.14e-01

续表

	PSO	SPSO	OPSO	GOPSO	FIPS	CLPSO	NCOPSO
f_4	1.91e+00 ± 2.55e+00 −	3.61e−03 ± 1.59e−03 −	1.16e−02 ± 5.64e−03 −	2.16e−02 ± 9.04e−03 −	1.03e−02 ± 3.10e−03 −	1.12e−02 ± 3.59e−03 −	3.65e−04 ± 2.54e−04
f_5	1.54e+02 ± 5.10e+01 −	2.58e+01 ± 6.09e+00 −	3.75e+01 ± 1.80e+01 −	5.17e+01 ± 6.23e+01 −	1.02e+02 ± 1.13e+01 −	2.95e−03 ± 2.12e−03 −	0.00e+00 ± 0.00e+00
f_6	1.25e+01 ± 8.64e+00 −	7.86e−01 ± 7.12e−01 −	2.39e+00 ± 6.23e+00 −	6.84e−14 ± 1.84e−13 −	3.40e−03 ± 6.39e−04 −	8.35e−05 ± 2.31e−05 −	8.88e−16 ± 0.00e+00
f_7	1.81e+01 ± 3.82e+01 −	3.70e−03 ± 4.96e−03 −	3.41e−02 ± 2.65e−02 −	1.03e−02 ± 9.49e−03 −	1.32e−02 ± 1.91e−02 −	3.98e−06 ± 3.23e−06 −	0.00e+00 ± 0.00e+00
f_8	1.10e+02 ± 1.71e+01 ≈	1.30e+02 ± 9.81e+00 −	5.25e+01 ± 3.04e+01 +	8.42e+01 ± 1.46e+01 +	1.89e−01 ± 1.25e−01 +	4.52e−08 ± 3.25e−08 +	9.60e+01 ± 2.41e+01
f_9	2.76e+01 ± 1.89e+01 −	2.27e−13 ± 0.00e+00 ≈	2.46e−04 ± 3.20e−04 −	9.36e−04 ± 1.57e−03 −	2.70e−04 ± 1.35e−04 −	3.10e−07 ± 1.69e−07 −	2.27e−13 ± 0.00e+00
f_{10}	5.31e+07 ± 1.84e+07 −	8.54e+05 ± 3.79e+05 +	3.26e+07 ± 1.79e+07 −	3.19e+07 ± 1.56e+07 −	5.27e+07 ± 1.23e+07 −	3.10e+07 ± 9.22e+06 −	3.65e+06 ± 5.43e+06
f_{11}	5.62e+09 ± 3.90e+09 −	2.44e+08 ± 3.17e+08 ≈	4.03e+09 ± 2.04e+09 −	6.20e+09 ± 4.64e+09 −	3.00e+08 ± 2.30e+08 −	3.70e+09 ± 1.09e+09 −	7.28e+07 ± 6.94e+07
f_{12}	2.27e+04 ± 5.75e+03 ≈	2.07e+04 ± 6.41e+03 ≈	3.47e+04 ± 8.05e+03 −	3.49e+04 ± 6.94e+03 −	5.99e+04 ± 9.60e+03 −	6.93e+04 ± 1.39e+04 −	2.05e+04 ± 4.31e+03
f_{13}	1.16e+01 ± 3.67e+00 −	1.54e−03 ± 1.27e−04 +	2.74e−02 ± 2.82e−02 −	2.22e−02 ± 2.19e−02 −	1.00e−02 ± 1.42e−03 −	1.45e−04 ± 6.16e−05 +	2.85e−03 ± 2.56e−04
f_{14}	2.60e+02 ± 7.24e+01 −	3.07e+01 ± 2.86e+01 +	1.16e+02 ± 4.28e+01 ≈	1.32e+02 ± 3.18e+01 ≈	2.70e+01 ± 6.69e−01 +	6.78e+01 ± 1.24e+01 ≈	1.08e+02 ± 6.42e+01
f_{15}	7.78e+01 ± 2.19e+01 −	5.29e+01 ± 1.19e+01 −	1.13e+02 ± 2.83e+01 −	1.16e+02 ± 1.96e+01 −	6.75e+01 ± 1.12e+01 −	1.04e+02 ± 1.94e+01 −	2.93e+01 ± 1.45e+01
f_{16}	2.10e+01 ± 5.24e−02 ≈	2.10e+01 ± 5.18e−02 ≈	2.10e+01 ± 3.84e−02 −	2.10e+01 ± 3.03e−02 ≈	2.10e+01 ± 6.03e−02 ≈	2.10e+01 ± 5.19e−02 −	2.10e+01 ± 6.55e−02
f_{17}	3.21e+01 ± 2.96e+00 −	2.66e+01 ± 6.86e+00 ≈	3.55e+01 ± 2.71e+00 −	3.41e+01 ± 3.15e+00 −	3.61e+01 ± 3.39e+00 −	3.18e+01 ± 1.80e+00 −	2.34e+01 ± 7.34e+00
f_{18}	1.98e+02 ± 9.15e+01 −	3.12e−01 ± 1.44e−01 ≈	1.50e+01 ± 2.00e+01 −	3.73e+01 ± 4.43e+01 −	1.21e+01 ± 6.39e+00 −	3.24e+01 ± 1.06e+01 −	2.36e−01 ± 1.51e−01
f_{19}	1.10e+02 ± 2.25e+01 −	8.88e+01 ± 4.09e+01 −	5.74e+01 ± 2.74e+01 ≈	5.38e+01 ± 2.06e+01 ≈	1.03e+02 ± 1.31e+01 −	4.01e−02 ± 1.93e−02 +	6.02e+01 ± 1.78e+01
f_{20}	2.37e+02 ± 1.90e+01 −	6.41e+01 ± 1.73e+01 ≈	1.53e+02 ± 4.11e+01 −	1.44e+02 ± 2.75e+01 −	1.98e+02 ± 1.12e+01 −	1.77e+02 ± 2.62e+01 −	5.93e+01 ± 2.02e+01
f_{21}	2.41e+02 ± 1.70e+01 −	1.24e+02 ± 3.70e+01≈	2.28e+02 ± 5.23e+01 −	2.31e+02 ± 5.13e+01 −	1.89e+02 ± 1.14e+01 −	2.09e+02 ± 1.26e+01 −	1.33e+02 ± 2.71e+01
f_{22}	3.78e+03 ± 1.10e+03 +	5.44e+03 ± 5.17e+02 +	1.63e+03 ± 3.24e+02 +	1.52e+03 ± 3.90e+02 +	6.08e+03 ± 1.96e+02 +	1.81e+02 ± 4.40e+01 +	6.98e+03 ± 9.36e+02
f_{23}	7.05e+03 ± 2.89e+02 ≈	5.56e+03 ± 6.34e+02 −	4.54e+03 ± 9.48e+02 −	4.68e+03 ± 7.81e+02 −	7.53e+03 ± 2.22e+02 +	5.90e+03 ± 3.97e+02 ≈	6.23e+03 ± 1.29e+03
f_{24}	2.56e+00 ± 3.34e−01 ≈	1.99e+00 ± 3.01e−01 +	2.62e+00 ± 2.92e−01 ≈	2.46e+00 ± 1.96e−01 ≈	2.68e+00 ± 3.39e−01 ≈	2.46e+00 ± 4.35e−01 ≈	2.51e+00 ± 4.13e−01
f_{25}	2.37e+02 ± 3.28e+01 −	1.64e+02 ± 2.54e+01 ≈	6.59e+01 ± 1.29e+01 +	6.92e+01 ± 2.28e+01 +	2.02e+02 ± 1.19e+01 −	4.16e+01 ± 2.18e+00 +	1.83e+02 ± 2.70e+01

续表

	PSO	SPSO	OPSO	GOPSO	FIPS	CLPSO	NCOPSO
f_{26}	3.26e+02 ± 3.03e+01 −	1.51e+02 ± 1.83e+01 +	2.57e+02 ± 5.31e+01 −	2.86e+02 ± 7.32e+01 −	2.23e+02 ± 1.84e+01 −	2.37e+02 ± 1.46e+01 −	1.80e+02 ± 2.23e+01
f_{27}	2.87e+01 ± 6.40e+00 −	8.94e+00 ± 4.34e+00 ≈	5.70e+00 ± 1.34e+00 +	5.61e+00 ± 1.24e+00 +	1.55e+01 ± 1.02e+00 −	3.20e+00 ± 7.18e−01 +	9.39e+00 ± 2.12e+00
f_{28}	1.46e+01 ± 9.68e−01 −	1.40e+01 ± 1.41e+00 ≈	1.47e+01 ± 7.21e−01 −	1.45e+01 ± 8.40e−01 −	1.50e+01 ± 0.00e+00 −	1.46e+01 ± 3.03e−01 −	1.32e+01 ± 5.56e−01
f_{29}	3.77e+02 ± 7.80e+01 ≈	3.25e+02 ± 8.15e+01 ≈	2.93e+02 ± 7.54e+01 ≈	3.16e+02 ± 7.43e+01 −	3.25e+02 ± 7.20e+01 −	3.04e+02 ± 2.71e+01 +	3.06e+02 ± 6.58e+01
f_{30}	3.35e+03 ± 6.86e+02 +	4.83e+03 ± 9.79e+02 −	1.86e+03 ± 7.06e+02 +	2.04e+03 ± 7.74e+02 +	6.21e+03 ± 4.13e+02 ≈	4.11e+02 ± 1.20e+02 +	6.69e+03 ± 1.32e+03
f_{31}	7.53e+03 ± 4.44e+02 ≈	4.91e+03 ± 8.34e+02 +	4.87e+03 ± 8.39e+02 +	5.26e+03 ± 3.35e+02 +	7.83e+03 ± 3.64e+02 −	6.70e+03 ± 3.05e+02 +	7.23e+03 ± 8.26e+02
f_{32}	2.82e+02 ± 1.46e+01 −	2.63e+02 ± 1.09e+01 −	2.92e+02 ± 1.25e+01 −	2.96e+02 ± 1.30e+01 −	3.00e+02 ± 4.79e+00 −	2.81e+02 ± 6.48e+00 −	2.40e+02 ± 6.86e+00
f_{33}	3.29e+02 ± 6.25e+00 −	2.76e+02 ± 7.73e+00 +	3.19e+02 ± 1.10e+01 ≈	3.20e+02 ± 8.69e+00 ≈	3.01e+02 ± 3.76e+00 +	3.06e+02 ± 4.29e+00 ≈	3.11e+02 ± 1.70e+01
f_{34}	2.96e+02 ± 8.84e+01 −	2.60e+02 ± 7.65e+01 ≈	2.02e+02 ± 1.37e+00 ≈	2.02e+02 ± 1.41e+00 ≈	2.84e+02 ± 9.53e+01 ≈	2.04e+02 ± 1.30e+00 ≈	2.74e+02 ± 7.16e+01 ≈
f_{35}	1.12e+03 ± 1.30e+02 −	8.40e+02 ± 5.96e+01 −	1.22e+03 ± 6.55e+01 −	1.22e+03 ± 8.34e+01 −	1.30e+03 ± 3.17e+01 −	9.39e+02 ± 3.19e+02 ≈	7.14e+02 ± 1.09e+02
f_{36}	8.15e+02 ± 3.21e+02 −	2.73e+02 ± 7.04e+01 +	8.78e+02 ± 5.15e+02 −	1.09e+03 ± 5.22e+02 −	3.01e+02 ± 5.53e−01 −	3.31e+02 ± 1.20e+01 −	3.00e+02 ± 7.29e−14
−/+/≈	27/2/7	13/10/13	23/7/6	23/7/6	26/5/5	19/10/7	

第 3 组测试函数是 CEC'13 带偏移的单峰函数. NCOPSO 在 f_9, f_{11} 和 f_{12} 上均优于或相当于其他比较算法，在 f_{10} 上仅次于 SPSO，在 f_{13} 上次于 CLPSO、SPSO. 图 2(e) 可看出 NCOPSO 在 f_9 上的收敛曲线. 另外两种反向学习粒子群算法 OPSO 和 GOPSO 找到的解均差于本文算法. 从图 3 也可以看出，OPSO 和 GOPSO 在这组函数上的 Friedman 检验排名值也比前两组下降较大，而本文算法的 Friedman 检验排名值还是第一. 这主要是因为 OPSO 和 GOPSO 中反向解是用当前解每一维的最大最小坐标来计算的，这种计算方式容易造成坐标搜索偏向，当最优解位于坐标轴上或是坐标系中间时，较易找到最优解. 但对于偏移函数，最优解的位置随机偏移，这种方式就不易找到最优解了. 而 NCOPSO 以重心为参考点计算反向解，重心是群体的几何中心，没有坐标搜索偏向，所以仍能表现出不错的收敛性.

第 4 组测试函数解空间更复杂. NCOPSO 在 f_{15}、f_{16} 和 f_{17} 等 8 个函数上均优于或相当于其他比较算法，在 f_{14} 和 f_{26} 上次于 SPSO，在 f_{14} 和 f_{23} 上次于 FIPS，在 f_{19} 上次于 CLPSO. f_{22} 函数是带偏移的 f_8，NCOPSO 找到的最优解劣于所有其它比较算法. 图 2(f) 可看出 NCOPSO 在 f_{15} 上的收敛曲线. 这一组函数上，各算法取得的结果精度差距都不大.

第 5 组函数具有多峰、不可分、不对称，不同局部极值有不同特性的特点. NCOPSO 在 f_{32}, f_{34} 和 f_{35} 上优于或相当于其他比较算法，在 f_{29}, f_{30}, f_{33} 和 f_{36} 上表现比较一般，因为这几个组合函数中均包含了 f_8 函数. 邻域重心反向学习策略在这类函数上发挥的作用有限. 解决这个问题可以结合一些变异策略来进一步提高算法跳出局部极值的能力. 与上一组函数相似，这组函数上各算法求解精度相差不大.

对表 1 的结果再次进行分析，将 NCOPSO 的结果与其它算法的结果两两分别进行 Wilcoxon 秩和检验，表 2 记录下渐近显著性值(p-value)，且 p-value 值小于 0.05 的用粗体表示.

表 2 各算法的 Wilcoxon 秩和检验的渐近显著性值(p-value)

NCOPSO vs.	PSO	SPSO	OPSO	GOPSO	FIPS	CLPSO
p-value	**0.000**	0.694	0.201	0.140	**0.002**	0.824

从表 2 可以看出，NCOPSO 明显优于 PSO 和 FIPS. 虽然从统计意义上本文算法性能不显著优于其它算法. 从图 3 看出，虽然在第 4、5 组函数上 NCOPSO 的 Friedman 检验排名不是第一，但是在第 1、2、3 组和总体

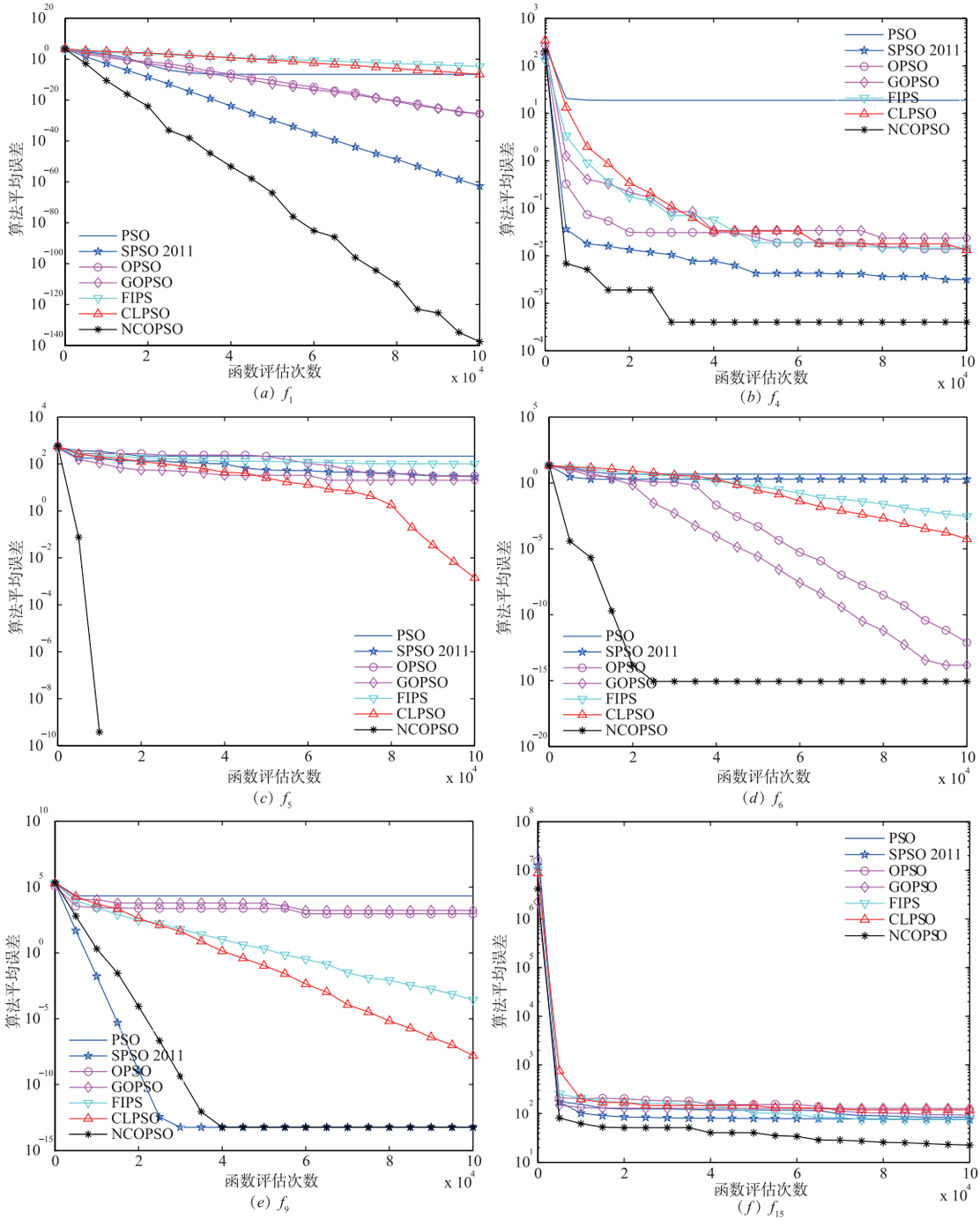


图2　各算法在f_1、f_4、f_5、f_6、f_9和f_{15}上的收敛曲线

上 NCOPSO 算法排名都是第一.

综合以上实验分析,说明领域重心反向学习策略有效提高了算法的性能.

4.3　邻域重心反向学习策略有效性分析

为了进一步对邻域重心反向学习策略进行分析,构建了重心反向学习粒子群优化算法 COPSO,它以粒子群体重心为参考点计算反向解. 表 3 给出了它们在测试集前 4 组函数上的结果. 分析表 3 的结果可以发现,NCOPSO 算法在 21 个函数上都胜过 COPSO 算法,5 个函数上与它相当,说明 NCOPSO 算法采用邻域重心计算反向解,保留了 COBL 利用群体搜索经验的优点,而且弥补了 COBL 的多样性不足,能够协调整体性与多

图3 各算法在不同分组上的Friedman检验排名

样性之间的平衡,对提高算法收敛能力发挥了积极作用.

4.4 拓扑结构对算法影响

为了分析拓扑结构对算法的影响,分别构建环形拓扑的 NCOPSO 算法,记为 NCOPSO_Ring,和正方形拓扑的 NCOPSO 算法,记为 NCOPSO_Square. 这两种算法中粒子邻居个数固定,不需要额外增加参数,其它设置与 NCOPSO 算法相同. 表4 给出了3 种算法在测试集前4 组函数上的结果,其中最后一行是 3 种算法的 Friedman 检验排名. 分析表 4 可以看出,不同拓扑结构的 NCOPSO 算法在大部分函数上的结果差距不大. 从表 4 最后一行的 Friedman 检验排名上来看,NCOPSO 算法略优于其它两个算法.

表3 NCOPSO 算法及变体的实验结果

Function	NCOPSO	COPSO
f_1	8.77e-135 ±1.84e-134	4.93e-76 ±9.21e-76
f_2	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00
f_3	2.54e+01 ±1.14e-01	2.56e+01 ±1.25e-01
f_4	3.65e-04 ±2.54e-04	1.05e-03 ±3.10e-04
f_5	0.00e+00 ±0.00e+00	4.69e-09 ±1.05e-08
f_6	8.88e-16 ±0.00e+00	4.44e-15 ±0.00e+00
f_7	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00
f_8	9.60e+01 ±2.41e+01	1.72e+01 ±5.28e+00
f_9	2.27e-13 ±0.00e+00	2.27e-13 ±0.00e+00
f_{10}	3.65e+06 ±5.43e+06	2.42e+06 ±2.22e+06
f_{11}	7.28e+07 ±6.94e+07	7.22e+07 ±8.20e+07
f_{12}	2.05e+04 ±4.31e+03	2.38e+04 ±4.70e+03
f_{13}	2.85e-03 ±2.56e-04	2.84e-03 ±1.94e-04
f_{14}	1.08e+02 ±6.42e+01	1.15e+02 ±6.32e+01
f_{15}	2.93e+01 ±1.45e+01	2.90e+01 ±8.44e+00

续表

Function	NCOPSO	COPSO
f_{16}	2.10e+01 ±6.55e-02	2.10e+01 ±3.67e-02
f_{17}	2.34e+01 ±7.34e+00	2.90e+01 ±7.69e+00
f_{18}	2.36e-01 ±1.51e-01	2.69e-01 ±1.44e-01
f_{19}	6.02e+01 ±1.78e+01	7.37e+01 ±2.14e+01
f_{20}	5.93e+01 ±2.02e+01	8.65e+01 ±4.04e+01
f_{21}	1.33e+02 ±2.71e+01	1.40e+02 ±2.65e+01
f_{22}	6.98e+03 ±9.36e+02	7.34e+03 ±3.52e+02
f_{23}	6.23e+03 ±1.29e+03	7.23e+03 ±3.11e+02
f_{24}	2.51e+00 ±4.13e-01	2.64e+00 ±2.23e-01
f_{25}	1.83e+02 ±2.70e+01	1.79e+02 ±1.94e+01
f_{26}	1.80e+02 ±2.23e+01	1.95e+02 ±1.75e+01
f_{27}	9.39e+00 ±2.12e+00	9.80e+00 ±3.90e+00
f_{28}	1.32e+01 ±5.56e-01	1.31e+01 ±8.36e-01
Friedman	1.32	1.68

4.5 应用到扩频雷达相位编码问题

为了进一步验证 NCOPSO 算法的性能,我们将 NCOPSO 算法应用于一个实际工程优化问题——扩频雷达相位编码问题. 该问题可以定义为[21]:

$$\text{Global } \min_{x\in X} f(x) = \max(\phi_1(x),\cdots,\phi_{2m}(x)) \quad (9)$$

其中 $\boldsymbol{X}=\{(x_1,\cdots,x_D)\in R^D \mid 0\leqslant x_j\leqslant 2\pi, j=1,\cdots,D\}$, $m=2D-1$,且

$$\phi_{2i-1}(x)=\sum_{j=i}^{n}\cos\Big(\sum_{k=|2i-j-1|+1}^{j}x_k\Big), i=1,\cdots,D,$$

$$\phi_{2i}(x)=0.5+\sum_{j=i+1}^{n}\cos\Big(\sum_{k=|2i-j|+1}^{j}x_k\Big), i=1,\cdots,D-1,$$

$$\phi_{m+i}(x)=-\phi_{2i}(x), i=1,\cdots,m \quad (10)$$

我们采用本文算法及相关比较算法求解该问题,将式(9,10)作为粒子适应值评估函数,式(9)中的自变量 $\boldsymbol{X}$ 表示为粒子,边界约束为[0,2π],在这个区域内随机产生粒子的初始位置. 为了比较的公平性,比较算法相同的参数均按照文献[22]建议的值来设置,即种群规模 $N=50$,最大函数评估次数 FE_max = 10000,问题维数为 $D=10$. 其它参数均保持 4.1 节的设置. 每种算法独立运行25 次,记录下平均最优值和标准方差,最好的结果用加粗表示,见表 5. DNLPSO 的结果来自于文献[22,Table 7]. 从表 5 可以看出,NCOPSO 取得的结果是最好的.

表 4　不同拓扑结构的 NCOPSO 算法比较

Function	NCOPSO	NCOPSO _ Ring	NCOPSO _ Square
f_1	8.77e-135 ±1.84e-134	4.79e-136 ±1.17e-135	1.92e-109 ±3.99e-109
f_2	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00
f_3	2.54e+01 ±1.14e-01	2.73e+01 ±5.43e-02	2.56e+01 ±1.06e-01
f_4	3.65e-04 ±2.54e-04	7.17e-05 ±4.77e-05	1.27e-04 ±5.26e-05
f_5	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00
f_6	8.88e-16 ±0.00e+00	8.88e-16 ±0.00e+00	3.97e-15 ±1.25e-15
f_7	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00	0.00e+00 ±0.00e+00
f_8	9.60e+01 ±2.41e+01	1.67e+02 ±2.68e+01	1.72e+02 ±2.09e+01
f_9	2.27e-13 ±0.00e+00	1.82e-13 ±9.41e-14	1.21e-13 ±1.17e-13
f_{10}	3.65e+06 ±5.43e+06	2.63e+06 ±9.38e+05	4.57e+06 ±6.96e+06
f_{11}	7.28e+07 ±6.94e+07	3.04e+08 ±3.72e+08	1.50e+08 ±7.50e+07
f_{12}	2.05e+04 ±4.31e+03	2.73e+04 ±3.52e+03	1.26e+04 ±2.26e+03
f_{13}	2.85e-03 ±2.56e-04	6.77e-03 ±1.00e-03	2.40e-03 ±3.13e-04
f_{14}	1.08e+02 ±6.42e+01	5.66e+01 ±3.30e+01	8.30e+01 ±3.11e+01
f_{15}	2.93e+01 ±1.45e+01	8.91e+01 ±2.11e+01	7.14e+01 ±1.39e+01
f_{16}	2.10e+01 ±6.55e-02	2.10e+01 ±3.66e-02	2.10e+01 ±6.60e-02
f_{17}	2.34e+01 ±7.34e+00	3.14e+01 ±1.99e+00	2.85e+01 ±2.50e+00
f_{18}	2.36e-01 ±1.51e-01	3.56e-01 ±1.15e-01	2.26e-01 ±6.89e-02
f_{19}	6.02e+01 ±1.78e+01	1.67e+02 ±4.13e+01	1.11e+02 ±2.57e+01
f_{20}	5.93e+01 ±2.02e+01	1.71e+02 ±3.42e+01	1.12e+02 ±2.99e+01
f_{21}	1.33e+02 ±2.71e+01	2.64e+02 ±5.66e+01	1.71e+02 ±2.85e+01
f_{22}	6.98e+03 ±9.36e+02	6.38e+03 ±5.70e+02	6.48e+03 ±1.03e+03
f_{23}	6.23e+03 ±1.29e+03	5.92e+03 ±7.77e+02	5.33e+03 ±1.25e+03
f_{24}	2.51e+00 ±4.13e-01	2.44e+00 ±4.15e-01	2.64e+00 ±3.66e-01
f_{25}	1.83e+02 ±2.70e+01	2.62e+02 ±3.51e+01	1.78e+02 ±3.81e+01
f_{26}	1.80e+02 ±2.23e+01	2.65e+02 ±4.39e+01	2.07e+02 ±1.93e+01
f_{27}	9.39e+00 ±2.12e+00	1.48e+01 ±4.21e+00	7.52e+00 ±2.04e+00
f_{28}	1.32e+01 ±5.56e-01	1.25e+01 ±3.51e-01	1.20e+01 ±3.84e-01
Friedman	1.88	2.23	1.89

表 5　扩频雷达相位问题的结果

SPSO	OPSO	GOPSO	FIPS	CLPSO	DNLPSO	NCOPSO
7.18e-1 (8.73e-2)	8.35e-1 (9.89e-2)	7.97e-1 (1.16e-1)	8.21e-1 (1.12e-1)	7.86e-1 (7.68e-2)	8.18e-1 (1.13e-1)	5.64e-1 (8.96e-2)

5　结论

本文提出了一种领域重心反向学习粒子群优化算法,采用随机拓扑结构,以粒子所在的邻域重心为参考点计算反向粒子,采用收缩因子拓展反向搜索区域. 通过理论分析和大量的实验结果表明,领域重心反向学习策略充分利用了种群搜索信息且增强了种群多样性,与其它反向学习粒子群优化算法和多种先进粒子群优化算法相比,本文算法具有更好的收敛精度. 在求解扩频雷达相位问题时,本文算法也取得了最好的结果. 后续工作是将本文算法应用于解决其它实际优化问题中.

参考文献

[1] Kennedy J, Eberhart R C. Particle swarm optimization[A]. Proceedings of IEEE International Conference on Neural Networks[C]. New York: IEEE, 1995. 1942 - 1948.

[2] Bonyadi M R, Michalewicz Z. Particle swarm optimization for single objective continuous space problems: a review [J]. Evolutionary Computation, 2016. Published online. http://dx. doi. org/10. 1162/EVCO _ r _ 00180

[3] Shi Y, Eberhart R C. A modified particle swarm optimizer [A]. IEEE World Congress on Computational Intelligence [C]. IEEE, 1998. 69 - 73.

[4] Mendes R, Kennedy J, et al. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204 - 210.

[5] 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.

[6] Wang H, Li H, Liu Y, et al. Opposition-based particle swarm algorithm with Cauchy mutation[A]. Proc of IEEE Congress on Evolutionary Computation[C]. New York: IEEE, 2007. 4750 – 4756.

[7] Chi Y, Cai G. Particle swarm optimization with opposition-based disturbance[A]. Informatics in Control, Automation and Robotics (CAR)[C]. New York: IEEE, 2010. 2: 223 – 226.

[8] Wang H, Wu Z, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. Information Sciences, 2011, 181(20): 4699 – 4714.

[9] 喻飞,李元香,等. 透镜成像反学习策略在粒子群算法中的应用[J]. 电子学报,2014,42(2):230 – 235.
Yu Fei, Li Yuan-xiang, et al. The application of a novel OL based on lens imaging principle in PSO[J]. Acta Electronica Sinica, 2014, 42(2): 230 – 235. (in Chinese)

[10] 邵鹏,吴志健,周炫余,等. 基于折射原理反向学习模型的改进粒子群算法[J]. 电子学报,2015,43(11):2137 – 2144.
Shao Peng, Wu Zhi-jian, Zhou Xuan-yu, et al. Improved Particle swarm optimization algorithm based on opposite learning of refraction[J]. Acta Electronica Sinica, 2015, 43(11): 2137 – 2144. (in Chinese)

[11] Xu Q, Wang L, Wang N, et al. A review of opposition-based learning from 2005 to 2012[J]. Engineering Applications of Artificial Intelligence, 2014, 29: 1 – 12.

[12] Rahnamayan S, Jesuthasan J, et al. Computing opposition by involving entire population[A]. IEEE Congress on Evolutionary Computation [C]. New York: IEEE, 2014. 1800 – 1807.

[13] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence[A]. Proceedings of International Conference on Intelligent Agent, Web Technologies and Internet Commerce[C]. Vienna: IEEE, 2005. 695 – 701.

[14] Kennedy J, Mendes R. Population structure and particle swarm performance[A]. IEEE Congress on Evolutionary Computation[C]. New York: IEEE, 2002. 1671 – 1676.

[15] Zambrano-Bigiarini M, Clerc M, Rojas R. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements[A]. IEEE Congress on Evolutionary Computation [C]. New York: IEEE, 2013. 2337 – 2344.

[16] Rahnamayan S, Wang G G. Center-based sampling for population-based algorithms[A]. IEEE Congress on Evolutionary Computation[C]. New York: IEEE, 2009. 933 – 938.

[17] Rahnamayan S, Tizhoosh H R, et al. Opposition-based differential evolution[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64 – 79.

[18] Suganthan P N, Hansen N, Liang J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization [R]. KanGAL: Nanyang Technological University, 2005.

[19] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82 – 102.

[20] Liang J J, Qu B Y, et al. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization[R]. Zhengzhou: Zhengzhou University, 2013.

[21] Das S, Suganthan P N. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems[R]. Singapore: Nanyang Technological University, 2010.

[22] Nasir M, Das S, Maity D, et al. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization[J]. Information Sciences, 2012, 209: 16 – 36.

作者简介

周凌云 女,1979 年生,武汉大学计算机学院博士研究生,研究方向为智能计算.
E-mail: zhouly@ mail. scuec. edu. cn

丁立新 男,1967 年生,博士,教授,博士生导师,研究方向为演化算法、机器学习.
E-mail: lxding@ whu. edu. cn