

并行协作骨干粒子群优化算法

申元霞¹, 曾传华², 王喜凤¹, 汪小燕¹

(1. 安徽工业大学计算机科学与技术学院, 安徽马鞍山 243032; 2. 安徽工业大学数理科学与工程学院, 安徽马鞍山 243032)

摘 要: 为解决骨干粒子群优化(Bare-Bone Particle Swarm Optimization, BBPSO)的早期收敛问题, 本文通过粒子的运动行为分析了导致 BBPSO 早期收敛的因素, 并提出并行协作 BBPSO, 该算法采用并行的主群和从群之间的协作学习来平衡勘探和开采能力. 为了增强主群的勘探能力, 提出动态学习榜样策略以保持群体多样性; 同时提出随机反向学习机制以实现从群的从全局到局部的自适应搜索功能. 在 14 个不同特征的测试函数上将本文算法与 6 种知名的 BBPSO 算法进行对比, 仿真结果和统计分析表明本文算法在收敛速度和精度上都有显著提高.

关键词: 骨干粒子群优化; 协作学习; 反向学习; 多样性

中图分类号: TP38

文献标识码: A

文章编号: 0372-2112 (2016)07-1643-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2016.07.018

A Parallel-Cooperative Bare-Bone Particle Swarm Optimization Algorithm

SHEN Yuan-xia¹, ZENG Chuan-hua², WANG Xi-feng¹, WANG Xiao-yan¹

(1. School of Computer Science and Technology, Anhui University of Technology, Maanshan, Anhui 243002, China;

2. School of Mathematics & Physics, Anhui University of Technology, Maanshan, Anhui 243002, China)

Abstract: To deal with the premature convergence of the bare-bone particle swarm optimization (BBPSO) algorithm, we make the analysis of the motion behavior of the particles and point out the reasons leading to the premature convergence. According to the analysis results, a parallel-cooperative BBPSO (PCBBPSO) algorithm is proposed in which the parallel-cooperative learning of a master swarm and a slave swarm balances between exploration and exploitation abilities. In order to improve the exploration ability of the master swarm, a dynamic learning exemplar strategy is presented to preserve the swarm diversity. Meanwhile, a stochastic opposition-based learning mechanism is developed to achieve the abilities of the slave swarm from the global search to the local search. The proposed algorithm was evaluated on 14 benchmark functions with different characteristics. The experimental results and statistic analysis show that the proposed method significantly outperforms six state-of-the-art BBPSO variants in terms of convergence speed and solution accuracy.

Key words: BBPSO; cooperative learning; opposition-based learning; diversity

1 引言

粒子群优化(Particle Swarm Optimization, PSO)^[1]是一种基于群体的随机优化技术, 源于对鸟群、鱼群及人类集体行为的研究. 由于 PSO 概念简洁、易于实现和优良的性能使其在诸多领域的优化问题中得到应用. 但是 PSO 的优化性能和收敛行为依赖学习参数的选择, 如惯性权重, 加速系数等^[2,3]. 为了提高 PSO 的优化性能, 很多改进策略被提出^[4-7]. 但是在保持算法简洁结构的前提下, 同时提高算法的收敛速度和精度仍是 PSO 一个富有挑战性的问题^[8].

2003 年, Kennedy 提出了一种无参数的骨干粒子群优化^[9](BBPSO), 该算法采用基于群体最优和个体历史最优信息的高斯采样在解空间中进行搜索. 与传统的 PSO 相比, BBPSO 减少了速度项、加速系数、速度阈值等参数, 使得算法结构更为简洁易于操作. 目前, BBPSO 在图像的特征选择, 经济调度和故障诊断等实际问题中的应用取得优良的效果^[10,11].

由于 BBPSO 收敛速度快, 当解决复杂多峰问题时, 群体容易迅速聚集, 从而陷入早期收敛. 为了提高算法的勘探能力, Krohling 等^[12]提出了高斯和柯西跳跃策略的 BBPSO 算法. Orman 等^[13]将差分算法操作引入 BBP-

SO 以增强群体多样性. Zhang 等^[14] 提出了基于突变和交叉策略 BBPSO 算法以提高粒子的逃逸能力克服早期收敛现象. Riccardo 等^[15] 指出采样方法的微小变化利于 BBPSO 在解空间中开拓搜索新的空间. Blackwell^[16] 对 BBPSO 做了的群体行为分析, 给出了群体坍塌 (collapse) 条件, 提出了跳跃策略的 BBPSO 算法. Zhang 等^[17] 提出了具有自适应突变的 BBPSO 算法, 并将该算法应用于图像的特征选择.

上述研究表明, 突变和跳跃等辅助策略可以增加群体的多样性, 但是也降低了算法的收敛速度. 为了同时提高算法的收敛速度和精度, 本文首先分析了导致 BBPSO 早期收敛的因素, 依据分析结果, 提出了并行协作 BBPSO (Parallel-Cooperative BBPSO, PCBBPSO). PCBBPSO 算法利用两个并行异构种群的协作学习来平衡群体的勘探和开采能力. 两个种群分别称为主群和从群. 为了增强主群的勘探能力, 提出动态学习榜样策略以保持群体多样性; 同时提出随机反向学习机制使从群随着进化代数的增加自适应地从全局搜索转向局部搜索. 当主群和从群的进化状态满足交互条件时, 启动群体间的交互机制, 该机制可以提高两群的收敛精度.

2 经典 PSO 模型

设搜索空间为 D 维, 群体规模为 N , 粒子 i 的位置和速度, 分别用 D 维向量表示 $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ 和 $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$. 第 i 个粒子迄今为止搜索到的历史最优位置 $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$, 记为 P_{best} ; 整个粒子群迄今为止搜索到的最优位置 $P_g = (P_{g1}, P_{g2}, \dots, P_{gD})$, 记为 G_{best} . 经典 PSO 模型中粒子的速度和位置按如下公式更新:

$$V_i(t+1) = wV_i(t) + c_1r_1(P_i(t) - X_i(t)) + c_2r_2(P_g(t) - X_i(t)) \quad (1)$$

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

其中 $i=1, 2, \dots, N$; w 为惯性权重, r_1 和 r_2 为 $[0, 1]$ 之间独立均匀分布的随机数; c_1 和 c_2 为加速系数.

3 BBPSO 算法及其运动行为分析

3.1 BBPSO 算法

文献[2]指出经典 PSO 算法中每个粒子收敛于自身个体历史极值和全局极值的加权平均值, 即:

$$\lim_{t \rightarrow \infty} X_i(t) = \frac{\varphi_1 P_i + \varphi_2 P_g}{\varphi_1 + \varphi_2} \quad (3)$$

其中 φ_1 和 φ_2 为随机数, 分别为 $\varphi_1 = r_1 c_1$, $\varphi_2 = r_2 c_2$. 依据此结论, Kennedy 提出了骨干粒子群优化 (BBPSO), 算法中没有速度项, 粒子 i 的位置更新方程如下式:

$$X_{ij}(t+1) = N(u_{ij}(t), \sigma_{ij}^2(t)) \quad (4)$$

其中符号 $N(\cdot)$ 为高斯分布, 均值 $u_{ij} = 0.5(P_{ij}(t) + P_{gj}(t))$, 标准差 $\sigma_{ij} = |P_{ij}(t) - P_{gj}(t)|$, $j=1, 2, \dots, D$. 潘峰等^[18] 指出经典 PSO 同 BBPSO 具有近似的搜索机制.

3.2 BBPSO 行为分析

在 BBPSO 中, 粒子的更新位置 $X(t+1)$ 是个随机变量. 为了描述粒子的一维搜索空间, 本文引入置信区间的概念, 即对一个给定的 $a(0 < a < 1)$, 并满足 $P(\theta_L \leq X \leq \theta_U) = 1 - a$, $[\theta_L, \theta_U]$ 称为粒子位置 $X(t+1)$ 的置信度为 $1 - a$ 的置信区间. 该区间可看作粒子更新位置 $X(t+1)$ 置信水平为 $1 - a$ 的搜索空间.

由于群体中的粒子均为独立进化, 任意选取一个粒子 i 的运动做分析, 得到的结论可以推广至其他粒子. 不失一般性, 将式(4)简化为一维模型, 即

$$X_i(t+1) = N(0.5(P_i(t) + P_g(t)), |P_i(t) - P_g(t)|^2) \quad (5)$$

依据置信区间的概念, 求最短的置信区间是困难的, 可转求等尾置信区间.

结论 1 $[u_i(t) - \sigma_i(t)\Phi^{-1}(1 - a/2), u_i(t) + \sigma_i(t)\Phi^{-1}(1 - a/2)]$ 为粒子 i 更新位置 $X_i(t+1)$ 置信水平为 $1 - a$ 的等尾置信区间, 其中 $X_i(t+1) \sim N(u_i(t), \sigma_i^2(t))$, $u_i(t) = 0.5(P_i(t) + P_g(t))$, $\sigma_i(t) = |P_i(t) - P_g(t)|$, Φ 为标准正态分布函数, a 是置信水平.

证明 令 $[\theta_L, \theta_U]$ 为随机变量 $X_i(t+1)$ 取值的某一区间, 并满足 $P(\theta_L \leq X_i(t+1) \leq \theta_U) = 1 - a$, 将该式变形得

$$P\left(\frac{\theta_L - u_i(t)}{\sigma_i(t)} \leq \frac{X_i(t+1) - u_i(t)}{\sigma_i(t)} \leq \frac{\theta_U - u_i(t)}{\sigma_i(t)}\right) = 1 - a \quad (6)$$

令 $c = (\theta_U - u_i(t))/\sigma_i(t)$, 欲求等尾置信区间, 则 $c = -(\theta_L - u_i(t))/\sigma_i(t)$. 由式(6)可得式(7),

$$\Phi(c) - \Phi(1 - c) = 1 - a \quad (7)$$

由式(7)可解得 $c = \Phi^{-1}(1 - a/2)$. 置信下限和上限分别为: $\theta_L = u_i(t) - \sigma_i(t)\Phi^{-1}(1 - a/2)$, $\theta_U = u_i(t) + \sigma_i(t)\Phi^{-1}(1 - a/2)$.

根据结论 1 可知, 粒子更新位置的置信区间的大小依赖于正态分布的标准差. 当置信水平为 0.05 时, 粒子更新位置 $X_i(t+1)$ 的置信区间为 $[u_i(t) - 1.96\sigma_i(t), u_i(t) + 1.96\sigma_i(t)]$, 即粒子更新的位置以 95% 概率将落在这个区间. 下面结合图 1 分析群体陷入早期收敛的原因, 图 1 为一维粒子置信区间示例图, 其中置信水平为 0.05.

图 1 中 P_0 和 P_1 分别表示粒子 X_0 和 X_1 的历史最优位置 P_{best} , P_g 表示群体的历史最优位置 G_{best} , A 和 B

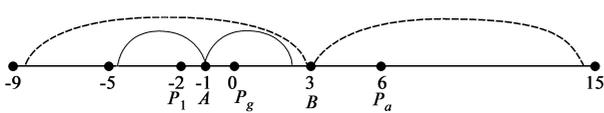


图1 粒子置信区间示例图

分别是 P_1 和 P_0 与 P_g 的中心点. 点 P_1, A, P_g, B 和 P_0 在数轴上位置分别为 $(-2, -1, 0, 3, 6)$. X_0 和 X_1 在下次更新且置信水平为 0.05 的置信区间分别为 $[-8.76, 14.76]$, $[-4.92, 2.92]$. 图中虚弧线表示 X_0 的下次更新的置信区间, 实弧线表示 X_1 的下次更新的置信区间. 由图 1 可知, 与 P_1 比较, P_0 与 P_g 的距离较远, P_0 的下次更新的置信区间较大, 表明粒子有较大的搜索空间. 在进化初期, 粒子的 P_{best} 与 G_{best} 相距较远时, 群体具有较强的勘探能力. 随着 P_{best} 向 G_{best} 靠近, 更新粒子的置信区间随之变小, 粒子开始集聚. 另外, 从图中可以看出, 群体中所有更新粒子的置信区间会覆盖 P_g 附近的区域, 使得更新粒子落在该区域的概率增加, 这也是 BBPSO 快速收敛的重要原因.

4 并行协作 BBPSO

4.1 主群的学习机制

为了解决群体多样性丧失问题, 主群利用动态学习榜样策略来实现全局搜索. 动态学习榜样策略是指粒子不再只学习单一的群体最优值 G_{best} , 还可以向其他粒子的历史最优值 P_{best} 学习. 如何从多个 P_{best} 中选择自己的学习榜样, 本文采用优胜榜样的方法. 下面给出优胜榜样的定义.

定义 1 优胜榜样 设群体历史最优个体集合 $P = \{P_1, \dots, P_i, \dots, P_N\}$ ($i \in 1, 2, \dots, N$) 对于随机变量 t , $k(t, k \in 1, 2, \dots, N, t \neq k)$, 有 $F(P_k) > F(P_i)$, 那么粒子 i 的第 j 维优质学习榜样是 P_{kj} .

定义 1 中的 F 是适应度函数, 适应度值越大代表解越好. 优胜榜样是指在每次比较中获胜的 P_{best} . 主群中第 i 个粒子位置的更新方程为:

$$X_{ij}(t+1) = N(0.5(P_{ij}(t) + P_{kj}(t)), |P_{ij}(t) - P_{kj}(t)|^2) \quad (8)$$

式(8)中的 P_{kj} 是粒子 i 的第 j 维优胜榜样. 从式(8)可知, 每个粒子的每一维都有一个不同的优胜榜样, 这种学习方式不仅减小了粒子在主群的群体最优值 (P_g^m) 附近搜索的概率, 而且利于扩大粒子的搜索空间和维持群体的多样性.

4.2 从群的学习机制

从群的功能是随着进化代数的增加由全局搜索转向局部搜索, 以提高进化后期解的质量. 由 BBPSO 机理分析可知, 粒子更新位置的搜索区间的大小依赖于高斯抽样的标准差. 为了保证从群能够快速的收敛, 粒子

仍只向从群的群体最优值 P_g^s 学习, 但是更新粒子高斯抽样的标准差依据概率选择当前粒子到 P_g^s 或是当前粒子到 P_g^s 反向解 P_g^{*s} 的距离. 进化前期, 粒子以较大概率接收 P_g^s . 增大搜索空间以提高勘探能力, 避免陷入早期收敛; 进化后期, 粒子以较大概率接收 P_g^s , 利于群体的局部搜索, 提高解的收敛精度.

从群粒子 i 位置的更新方程为:

$$X_{ij}(t+1) = N(0.5(P_{ij}(t) + P_{gj}^s(t)), |P_{ij}(t) - G_j(t)|^2) \quad (9)$$

式(9)中的 G_j 按下式生成:

$$G_j(t+1) = \begin{cases} P_{gj}^{*s}(t), & (\text{rand} < p_c) \wedge (F(P_g^s(t)) = F(P_g^s(t-1))) \\ P_{gj}^s(t), & \text{otherwise} \end{cases} \quad (10)$$

式(10)中 P_g^{*s} 是 P_g^s 的随机反向解; 由式(9)和(10)可知, 当从群的全局最优解 P_g^s 得到更新时, 高斯抽样的标准差为当前粒子到 P_g^s 的距离, 以增强粒子的开采能力; 当从群的 P_g^s 保持不变时, 高斯抽样的标准差依概率选择当前粒子到 P_g^s 的距离或到 P_g^{*s} 的距离. 式(10)中 p_c 为接收概率, p_c 的值随着迭代次数从 1 线性递减到 0, 即 $p_c = 1 - t/t_{\max}$, t 为当前迭代次数, t_{\max} 为最大的迭代次数. 进化前期, p_c 的值大利于群体的勘探; 进化后期, p_c 的值递减, 利于群体的开采. P_g^s 随机反向解 P_g^{*s} 按下式生成.

$$P_{gj}^{*s}(t) = \begin{cases} X_j^u + X_j^l - P_{gj}^s(t), & \text{rand} < 0.5 \\ P_{gj}^s(t), & \text{otherwise} \end{cases} \quad (11)$$

式(11)中的 X_j^u 和 X_j^l 分别为粒子位置第 j 维的取值的上界和下界. P_g^s 每维都以 50% 的概率进行反向操作.

4.3 交互机制

在本文算法中, 随着反向解接收概率的减少, 从群加快收敛, 提高解的质量. 如果主群的群体最优值 (P_g^m) 优于从群的群体最优值 (P_g^s), 从群则接收主群的 P_g^m , 即 $P_g^s = P_g^m$. 这种交互是单向的, 信息是从主群流向从群. 为了保持自身的探测能力, 主群不接收从群的信息. 如果从群接收了主群的信息, 从群将不启用反向学习. 交互机制利于提高主群和从群解的收敛精度.

4.4 算法流程

BBPSO 算法的算法流程如下:

算法 1 BBPSO 算法

Step1 初始化种群: 分别随机初始化 N_m 个主群和 N_s 个从群的粒子位置, 其中 $N = N_m + N_s$.

Step2 将群体当前的位置作为自身历史最优位置; 分别计算主群和从群粒子的适应度, 将主群中适应度最大的个体赋值给主群的群体最优值 P_g^m , 将从群中适应度最大的个体赋值给主群的群体最优

值 P_g^s .

Step3 对于主群,根据式(8)调整粒子的位置;对于从群,根据式(9)、(10)和(11)调整粒子的位置.

Step4 计算主群和从群粒子的适应度值,主群和从群分别记录种群目前所搜索到的最优值 P_g^m 和 P_g^s .

Step5 判断是否满足群体交换条件,如若满足条件则将 $P_g^s = P_g^m$;否则,转向下一步.

Step6 判断迭代次数是否达到最大迭代次数,如若满足条件则停止迭代,将 P_g^m 和 P_g^s 的适应度做比较,将适应度值大的个体作为输出结果;否则,转至 Step 2.

表 1 测试函数

函数	函数名	搜索空间	最优值
f_1	Sphere	$[-100, 100]$	0
f_2	Rosenbrock	$[-30, 30]$	0
f_3	Ackley	$[-32, 32]$	0
f_4	Griewank	$[-600, 600]$	0
f_5	Rastrigin	$[-5.12, 5.12]$	0
f_6	Rastrigin_noncont	$[-5.12, 5.12]$	0
f_7	Schewfel 2.26	$[-500, 500]$	-12569.5
f_8	Rotated Ackley	$[-32, 32]$	0
f_9	Rotated Griewank	$[-600, 600]$	0
f_{10}	Rotated Rastrigin	$[-5.12, 5.12]$	0
f_{11}	Rotated Rastrigin_noncont	$[-5.12, 5.12]$	0
f_{12}	Shifted Rosenbrock	$[-100, 100]$	390
f_{13}	Shifted Ackley	$[-32, 32]$	-140
f_{14}	Shifted Griewank	$[-600, 600]$	-180

5 数值实验及分析

5.1 测试函数和参数设置

为了验证本文算法的优化性能,采用了 14 个测试函数,包括常规测试函数($f_1 \sim f_7$),带旋转函数($f_8 \sim f_{11}$)和漂移函数($f_{12} \sim f_{14}$),表 1 列出了函数的名称,取值范围和最优值.本文算法与传统 BBPSO^[9]算法及其改进的算法 BBExp^[9],BBDE^[14],BBPSO_MC^[15],BBPSO_J^[17]和 ABPSO^[18]进行了实验比较.对比算法的参数设置均参照相应的文献,其中 BBDE 的重组概率为 0.7;BBPSO_MC 的突变概率为 0.5,方差固定值为 0.001,邻域规模为 2;BBPSO_J 的学习率为 0.7.本文设置进化代数为 30000 代,运行次数为 30 次,种群规模为 40,本文算法中的主群和从群的规模分别为 20.

5.2 评价指标

本文采用适应度均值和方差评价算法的优化性能,其中适应度值为 $(f(x) - f(x^*))$ (x^* 和 x 分别代表理论最优值和算法获得最优值).为了判断本文算法与 6 种对比算法的性能是否存在显著性差异,采用 wilcoxon rank-sum test 检验对结果进行统计分析,显著性水平设为 0.05,分别用符号“+”,“-”和“=”表示本文算法的性能要优于,劣于和相当于对比算法.实验结果如表 2 所示,获得的最优结果用粗体显示.

表 2 7 种算法在 14 个测试函数上的均值和方差

函数	BBExp	BBPSO	BBPSO_MC	BBDE	BBPSO_J	ABPSO	PCBBPSO
f_1	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	4.94E-324 = (0.00E+00)	1.37E-02 + (1.31E-02)	4.74E-09 + (3.93E-09)	0.00E+00 (0.00E+00)
f_2	3.31E+00- (1.36E+00)	1.70E+00- (3.76E+00)	1.05E+01 + (5.32E+00)	4.77E+01 + (3.63E+01)	2.74E+01 + (1.72E+01)	2.43E+01 + (3.46E+01)	6.51E+00 (1.50E+00)
f_3	7.11E-15 = (0.00E+00)	7.11E-15 = (0.00E+00)	2.58E-14 = (7.88E-15)	1.86E-01 + (4.16E-01)	1.10E-02 + (1.26E-02)	3.71E-07 + (1.76E-07)	4.33E-15 (1.05E-15)
f_4	1.85E-03 + (3.70E-03)	4.92E-03 + (8.53E-03)	0.00E+00 = (0.00E+00)	5.51E-03 + (2.46E-03)	1.45E-01 + (6.67E-02)	7.88E-03 + (8.23E-03)	0.00E+00 (0.00E+00)
f_5	7.21E+00 + (3.84E+00)	4.38E+01 + (1.32E+01)	6.37E+00 + (2.87E+00)	8.16E+00 + (3.61E+00)	1.28E-02 + (2.05E-02)	2.19E+00 + (1.09E+00)	0.00E+00 (0.00E+00)
f_6	1.20E+00 + (1.64E+00)	1.04E+01 + (5.08E+00)	1.40E+00 + (1.14E+00)	1.30E+01 + (4.53E+00)	4.02E-05 + (6.96E-05)	4.00E-01 + (5.48E-01)	0.00E+00 (0.00E+00)
f_7	3.44E+02 + (1.01E+02)	2.00E+03 + (6.44E+02)	5.65E+02 + (3.21E+02)	6.74E+02 + (3.13E+02)	1.55E-02 + (8.05E-03)	2.05E-10 + (3.10E-10)	0.00E+00 (0.00E+00)
f_8	7.11E-15 = (0.00E+00)	8.11E-01 + (7.58E-01)	2.13E-14 = (6.65E-15)	1.21E-1 + (8.56E-1)	3.08E+00 + (2.49E-01)	4.15E-07 + (3.00E-07)	4.97E-15 (1.98E-15)
f_9	5.22E-03 + (4.08E-03)	4.11E-02 + (7.11E-03)	3.47E-03 + (6.42E-03)	1.52E-02 + (2.47E-02)	3.74E-01 + (1.57E-01)	6.03E-03 + (6.12E-03)	0.00E+00 (0.00E+00)
f_{10}	2.57E+01 + (8.51E+00)	5.04E+01 + (1.98E+01)	3.20E+01 + (2.09E+01)	1.65E+01 + (9.40E+00)	6.56E+01 + (2.49E+01)	4.51E+01 + (1.47E+01)	1.27E+01 (1.18E+01)
f_{11}	2.98E+01 + (4.07E+00)	6.63E+01 + (1.88E+01)	2.41E+01- (7.26E+00)	2.24E+01- (2.07E+00)	6.70E+01 + (4.36E+00)	3.70E+01 + (1.41E+01)	2.56E+01 (7.50E+00)
f_{12}	1.94E+01 + (3.00E+01)	1.91E+00- (1.89E+00)	2.14E+01 + (1.97E+01)	8.21E+01 + (3.94E+01)	1.81E+02 + (3.05E+01)	3.40E+01 + (2.19E+01)	1.11E+01 (7.41E+00)
f_{13}	4.97E-14 = (1.72E-14)	5.49E-01 + (9.50E-01)	1.93E-13 + (5.08E-14)	6.25E-14 = (4.67E-14)	1.71E-03 + (1.27E-03)	2.47E-07 + (1.43E-07)	3.98E-14 (1.56E-14)
f_{14}	2.84E-14 + (0.00E+00)	9.86E-02 + (2.46E-02)	1.02E-13 + (8.67E-14)	1.14E-14 + (1.56E-14)	2.14E-01 + (5.24E-02)	2.55E-02 + (3.20E-02)	0.00E+00 (0.00E+00)

5.3 算法的收敛精度和速度比较

对于典型测试函数($f_1 \sim f_7$),除了函数 f_2 ,本文提出的 PCBBPSO 均获得最优结果,其中找到了单峰函数 f_1 和多模函数 $f_4 \sim f_7$ 理论最优解.对于函数 f_2 ,BBPSO 获得了最好的结果,PCBBPSO 的结果为第三.特别地,对于复杂多模函数($f_5 \sim f_7$),PCBBPSO 是唯一可以找到理论解的算法.对于带旋转函数($f_8 \sim f_{10}$),PCBBPSO 获得的结果均明显优于其他 6 种算法,其中 f_9 函数找到了全局最优解;对于偏移函数($f_{11} \sim f_{14}$),PCBBPSO 获得了偏移函数 f_{13} 和 f_{14} 的最优结果,并找到了 f_{14} 的全局最优值.对于 f_{11} ,PCBBPSO 优越于 BBExp, BBPSO, BBPSO_J 和 ABPSO 4 种算法.对于偏移函数 f_{12} ,PCBBPSO 获得的结果仅次于 BBPSO,优于其他 5 种算法.

由 wilcoxon rank-sum test 检验结果可知,PCBBPSO 在 14 个函数上明显优于 ABPSO 和 BBPSO_J.与 BBDE, BBPSO_MC, BBPSO 和 BBExp 相比,本文算法分别在 12 个, 8 个, 10 个和 9 个函数上表现更优.为了综合比较算法之间的性能,本文采用 Friedman 检验对结

果进行分析,表 3 给出了 7 种算法的平均排名,本文算法综合性能为第一.

限于篇幅,本文只给出 3 个测试函数收敛曲线的对比,如图 2 所示.为便于比较,图中纵坐标为算法获取最优值的对数值.由图 2 可知,对于多模函数 f_6, f_7 和 f_9 ,本文算法不仅找到全局最优解而且均在 15000 代内收敛,说明算法较好地平衡了算法的收敛精度和速度.

表 3 7 种算法的平均排名

算法	平均排名
PCBBPSO	1.176
BBExp	2.587
BBPSO_MC	3.285
ABPSO	4.241
BBDE	4.642
BBPSO	4.714
BBPSO_J	5.071

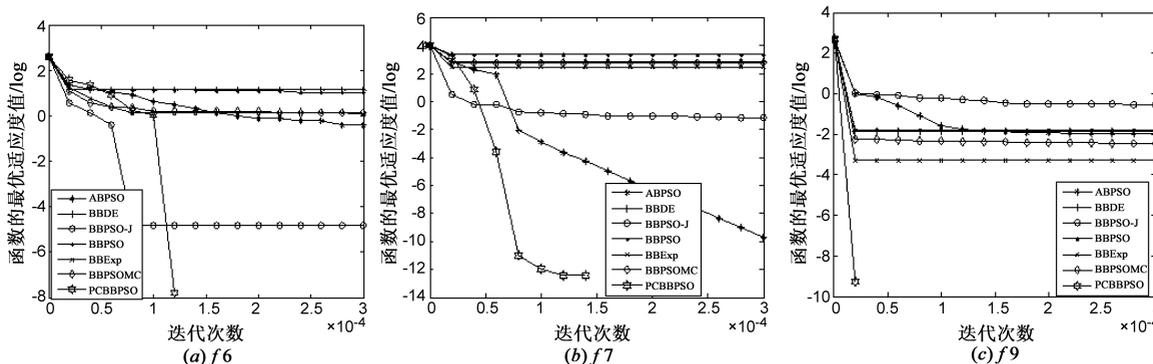


图 2 算法的收敛曲线比较

5.4 策略分析

PCBBPSO 利用动态学习榜样 (Dynamic Learning Example, DLE) 策略和随机反向学习 (Stochastic Opposition-based Learning, SOL) 机制协作搜索,其中 DLE 是实现全局搜索,而 SOL 是实现全局搜索自适应转向局部搜索.为了研究策略的有效性,将这两个策略分别独立优化表 1 中的函数,实验结果如表 4 所示.

由表 4 可知,对于复杂多模函数 $f_7, f_8, f_9, f_{10}, f_{11}, f_{13}$ 和 f_{14} , DLE 比 SOL 获得了更优的结果,这表明 DLE 具有全局搜索的能力;对于多模函数 f_4, f_5 和 f_6 , DLE 和 SOL 均获得最优值,说明 SOL 在进化前期具有较强的探测能力,有效地避免早期收敛问题;对于函数 f_1, f_2, f_3 和 f_{12} ,与 DLE 相比, SOL 优化的结果具有更高的精度,这表明 SOL 在进化后期加强了局部搜索.与 DLE 和 SOL 的独立优化结果相比,PCBBPSO 获得了所有函数的最优结果,表明 DLE 和 SOL 协作

优化可以更好平衡群体的勘探和利用,展现出满意的整体优化性能.

表 4 不同学习策略的比较结果

函数	DLE	SOL	PCBBPSO
f_1	6.90E-54 ± 7.27E-55	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
f_2	1.95E+01 ± 5.81E-01	9.24E+00 ± 2.17E+00	6.51E+00 ± 1.50E+00
f_3	5.33E-15 ± 2.51E-15	3.55E-15 ± 1.87E-15	4.33E-15 ± 1.05E-15
f_4	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
f_5	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
f_6	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00
f_7	0.00E+00 ± 0.00E+00	8.09E+02 ± 4.32E+01	0.00E+00 ± 0.00E+00
f_8	5.33E-15 ± 2.51E-15	7.11E-15 ± 3.16E-15	4.97E-15 ± 1.98E-15
f_9	5.22E-14 ± 1.04E-14	9.86E-03 ± 5.31E-03	0.00E+00 ± 0.00E+00
f_{10}	2.31E+01 ± 5.79E+00	3.04E+01 ± 1.04E+01	1.27E+01 ± 1.18E+01
f_{11}	3.18E+01 ± 1.11E+01	4.63E+01 ± 2.17E+01	2.56E+01 ± 7.50E+00
f_{12}	5.02E+01 ± 3.00E+01	1.13E+01 ± 9.33E+00	1.11E+01 ± 7.41E+00
f_{13}	5.68E-14 ± 2.21E-14	1.90E+00 ± 6.11E-01	3.98E-14 ± 1.56E-14
f_{14}	2.84E-14 ± 6.39E-15	6.15E-02 ± 4.68E-02	0.00E+00 ± 0.00E+00

6 结论

本文提出了一种并行协作 BBPSO (PCBBPSO) 算法,采用并行的主群和从群的协调进化以平衡算法的速度和精度.主群利用动态学习榜样策略扩大搜索的区域范围,保持群体多样性;从群通过随机反向学习机制使群体在进化前期提高群体的全局勘探能力,而进化后期增强群体的局部开采能力.通过 14 个典型函数进行测试,对比结果表明本文算法在解的精度与收敛速度上具有更优的性能.

参考文献

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. Proceedings of the IEEE International Conference on Neural Networks [C]. Perth, Australia: IEEE, 1995. 1942 – 1948.
- [2] Bergh F Van den, Engelbrecht A. A convergence proof for the particle swarm optimizer [J]. Fundamenta Informaticae, 2010, 105(4): 341 – 374.
- [3] Clerc M, Kennedy J. The particle swarm explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58 – 73.
- [4] Ratnaweera A, Halgamuge S, Waston H C. Self-organizing hierarchical particle optimizer with time-varying acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240 – 255.
- [5] Zhan Z H, Zhang J, Li Y, Chung H S. Adaptive particle swarm optimization [J]. IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics, 2009, 39(6): 1362 – 1381.
- [6] Akay B. A study on particle swarm optimization and artificial bee colony algorithms [J]. Applied Soft Computing, 2013, 13(6): 3066 – 3091.
- [7] 周新宇, 吴志健, 等. 一种精英反向学习的粒子群优化算法 [J]. 电子学报, 2013, 8(41): 1647 – 1652.
ZHOU Xin-yu, WU Zhi-jian, et al. Elite opposition-based particle swarm optimization [J]. Acta Electronica Sinica, 2013, 41(8): 1647 – 1652. (in Chinese)
- [8] Lim W H, Isa N A Mat. Bidirectional teaching and peer-learning particle swarm optimization [J]. Information Sciences, 2014, 280(10): 111 – 134.
- [9] Kennedy J. Bare bones particle swarms [A]. Proceedings of the IEEE Swarm Intelligence Symposium [C]. Indianapolis, USA: IEEE, 2003. 80 – 87.
- [10] Zhang Y, Gong D W, Hu Y, Zhang W Q. Feature selection algorithm based on bare-bones particle swarm optimization [J]. Neurocomputing, 2015, 148(19): 150 – 157.
- [11] Zhang Y, et al. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch [J]. Information Sciences, 2012, 192(1): 213 – 227.
- [12] Krohling R A, Mendel E. Bare bones particle swarm optimization with Gaussian or Cauchy jumps [A]. Proceedings of the IEEE Congress on Evolutionary Computation [C]. Trondheim, Norway: IEEE, 2009. 3285 – 3291.
- [13] Omran M G H, Engelbrecht A P, Ayed S. Bare bones differential evolution [J]. European Journal of Operational Research, 2009, 196(1): 128 – 139.
- [14] Zhang H, Kennedy D D, Rangaiah G P, Bonilla-Petriciolet A. A novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data [J]. Fluid Phase Equilibria, 2011, 301(1): 33 – 45.
- [15] Riccardo P, William B L. Markov chain models of bare-bones particle swarm optimizers [A]. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation [C]. London, England: ACM, 2007. 142 – 149
- [16] Blackwell T. A study of collapse in bare bones particle swarm optimization [J]. IEEE Trans Evolutionary Computation, 2012, 16(3): 354 – 375.
- [17] Zhang Y, Gong D W, Sun X Y, Geng N. Adaptive bare-bones particle swarm optimization algorithm and its convergence analysis [J]. Soft Computing, 2014, 18(7): 1337 – 1352.
- [18] 潘峰, 陈杰, 等. 粒子群优化方法若干特性分析 [J]. 自动化学报, 2009, 35(7): 1010 – 1015.
Pan Feng, Chen Jie, et al. Several characteristics analysis of particle swarm optimizer [J]. Acta Automatica Sinica, 2009, 35(7): 1010-1016. (in Chinese)

作者简介



申元霞 (通信作者) 女, 1979 年生于安徽六安, 博士, 讲师, 研究方向: 智能计算、智能信息处理。

E-mail: yuanxiashen@163.com

曾传华 男, 1975 年生于重庆, 讲师, 研究方向: 概率与统计理论、偏微分方程数值求解。

E-mail: stonzch@163.com

王喜凤 女, 1980 年生于山东成武, 讲师, 研究方向: 信息安全、智能信息处理。

E-mail: wxf80106@163.com

汪小燕 女, 1974 年生于安徽桐城, 硕士, 副教授, 研究方向: 数据挖掘、粗糙集理论。

E-mail: wxyzjx@126.com