

基于多种群的自适应迁移 PSO 算法

邓先礼¹, 魏 波¹, 曾 辉², 桂 凌³, 夏学文¹

(1. 华东交通大学软件学院, 江西南昌 330013; 2. 新疆工程学院计算机工程系, 新疆乌鲁木齐 830023;
3. 华东交通大学经济管理学院, 江西南昌 330013)

摘 要: 针对标准 PSO 中单一社会学习模式造成的算法容易陷入局部最优和后期收敛速度慢等问题, 提出了一种基于多种群的自适应迁移 PSO 算法 (Multi-population based self-adaptive migration PSO, MSMP SO). 通过融合两种常用的邻居拓扑结构, 赋予个体更多的信息来源; 在多个子种群并行进化的基础上, 利用不同加速因子的组合赋予各子种群不同的搜索特性, 进而通过周期性对子种群的历史性能进行评估, 以此为基础指导个体的迁移操作, 实现子种群间的协作与计算资源的合理分配, 并最终提升算法的综合性能. 对 CEC2013 测试函数的优化结果表明, MSMP SO 在求解精度、收敛速度等方面均表现出较好的性能.

关键词: 粒子群算法; 社会学习; 多种群; 个体迁移; 历史性能评估

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2018)08-1858-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.08.009

A Multi-Population Based Self-Adaptive Migration PSO

DENG Xian-li¹, WEI Bo¹, ZENG Hui², GUI Ling³, XIA Xue-wen¹

(1. School of Software, East China Jiaotong University, Nanchang, Jiangxi 330013, China;

2. Department of Computer Engineering, Xinjiang Institute of Engineering, Urumqi, Xinjiang 830023, China;

3. School of Economics and Management, East China Jiaotong University, Nanchang, Jiangxi 330013, China)

Abstract: The performance of particles' "social-learning" ability directly affects the search capability of PSO. To overcome some shortcomings caused by mono-social-learning model, such as premature convergence and slow convergence speed at later evolution stage, a multi-population based self-adaptive migration PSO (MSMP SO) is proposed. In MSMP SO, the two common neighbor typologies are integrated into particle's social-learning part aiming to give more information source for the particles. Furthermore, the entire population is divided into three sub-populations which are evolved in parallel. Based on the multi-population mechanism, different search characteristics caused by three different combinations of acceleration coefficients are assigned to the three sub-populations. To take advantage of different merits of different sub-populations, and realize the reasonable allocation of computing resources, individuals carry out a migration operator based on sub-populations' historical performance during the last period. Simulation results based on CEC2013 test suite manifest that the favorable comprehensive performance of MSMP SO, in terms of convergence speed and solutions accuracy.

Key words: particle swarm optimization; social learning; multi-population; individual migration; historical performance evaluation

1 引言

粒子群算法 (particle swarm optimization, PSO) 是 James Kennedy^[1] 等人于 1995 年提出的一种基于群智能的优化算法. 由于 PSO 基本理论简单, 且易于编码实

现, 因此在提出后被迅速应用于各种优化领域. 和其它许多群智能优化算法一样, PSO 算法也存在早熟和收敛速度慢等缺点, 而且随着优化问题维数的增加, 这些缺点愈发突出. 影响 PSO 性能的一个主要因素是粒子个体的速度更新方式, 因为它决定着粒子个体的移动

收稿日期: 2017-06-14; 修回日期: 2018-01-22; 责任编辑: 蓝红杰

基金项目: 国家自然科学基金 (No. 61663009, No. 61602174, No. 61763010, No. 61762036); 江西省科技厅科技项目 (No. 2017D0038); 江西省教育厅自然科学基金 (No. GJJ160469); 新疆维吾尔自治区教育厅高校科研计划基金 (No. 2014JYT041606); 新疆工程学院博士科研启动基金 (No. 2015BQJ011712)

方向及步长,进而决定其最终的搜索能力.因此,近年来众多学者对粒子速度更新时所涉及的参数以及学习对象的选择进行了广泛的研究.实验表明,合理利用种群或个体的历史经验来调节相关参数、选择学习对象能有效提升 PSO 的综合性能^[2].

本文从参数设计与学习对象选择这两个方面对 PSO 进行改进,提出了一种基于多种群的自适应迁移 PSO 算法(Multi-population based self-adaptive migration PSO, MSMP SO).首先,对标准 PSO 中的社会学习部分进行了扩展,使个体学习具有更丰富的信息来源;同时,将种群分为 3 个子种群,通过设置不同的加速因子使种群分别侧重于全局搜索、局部搜索和平衡搜索;最后,根据各子种群在进化过程中的表现,决定个体在各子种群间的迁移行为,以实现高效搜索策略的有效传播及计算资源的合理分配.

2 PSO

2.1 标准 PSO

在 PSO 中,第 i 个粒子可用 2 个向量来描述:位置向量 $\mathbf{X}_i^t = [x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t]$ 和速度向量 $\mathbf{V}_i^t = [v_{i,1}^t, v_{i,2}^t, \dots, v_{i,D}^t]$,其中 D 为目标问题的变量维数, N 为种群规模, t 为进化代数.搜索过程中, \mathbf{X}_i^t 可视为问题的一个候选解,而 \mathbf{V}_i^t 则可视为粒子飞行的方向和步长.粒子种群的飞行过程即为算法搜索最优解的过程.若第 i 个粒子进化至第 t 代时其自身的历史最优解为 $\mathbf{PB}_i^t = [pb_{i,1}^t, pb_{i,2}^t, \dots, pb_{i,D}^t]$ 、其邻居的历史最优解为 $\mathbf{NB}_i^t = [nb_{i,1}^t, nb_{i,2}^t, \dots, nb_{i,D}^t]$,则在第 $t+1$ 代时,该个体的速度和位置更新过程如(1)、(2)所示.

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^t + c_1 \cdot r_1 \cdot (pb_{i,j}^t - x_{i,j}^t) + c_2 \cdot r_2 \cdot (nb_{i,j}^t - x_{i,j}^t) \quad (1)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (2)$$

其中, w 为惯性权重,表示上一时刻的速度对本次移动的影响程度; c_1 和 c_2 为加速因子,分别表示个体进行“自我认知”和“社会学习”的能力大小; r_1 和 r_2 为 $[0, 1]$ 内均匀分布的随机数.

2.2 PSO 的研究现状

由式(1)和(2)可以看出, \mathbf{V}_i^t 直接决定着粒子个体的移动方向与步长,进而影响该个体的搜索能力.因此,近年来,众多学者对速度更新策略(即式(1))进行了广泛的研究.根据改进动机及对象的不同,这些改进策略大致可分为三类.

第一类是调整 PSO 的参数.如 Shi^[3]和 Ratnaweera^[4]分别对 w, c_1 和 c_2 进行线性动态调整,满足种群在不同进化时期的要求,达到平衡全局搜索能力和收敛速度的目的.但此类改进策略未考虑不同目标问题的具体特性,算

法的自适应能力较弱.2009年, Zhan^[5]根据预先定义的 4 种进化状态,在每次迭代时对当前种群执行实时状态评估,并根据评估结果对 w, c_1 和 c_2 进行自适应调整. Tanweer^[6]则在其提出的自整定 PSO (self regulating PSO, SRPSO)中利用个体适应值优劣来整定 w .实验表明,利用种群或个体进化过程中体现的一些统计特性来自适应地调整 PSO 中的参数能赋予种群或个体更好的搜索能力.

第二类是改变粒子的邻居拓扑结构. Suganthan^[7]和 Peram^[8]分别利用粒子间欧式距离和粒子适应值来确定粒子的学习模式,通过这种动态选择学习对象的策略,改善种群的多样性,提高了算法全局搜索能力. Mendes^[9]和 Liang^[10]则先后提出了完全感知 PSO 算法(Fully informed particle swarm, FIPS)和综合学习 PSO 算法(Comprehensive learning particle swarm optimizer, CLPSO).这两种算法的共同特点就是每个粒子在进行社会学习时会向种群中其它所有粒子的历史最优解进行学习,从而丰富了粒子的社会学习模式,改善了种群多样性; Li^[11]提出了一种基于信息共享机制的竞争-协作 PSO (competitive and cooperative PSO with information sharing mechanism, CCP-SO-ISM).在 CCPSO-ISM 中,当个体陷入停滞时,通过竞争机制为其重新选择较优的学习对象.同时,不同个体间通过共享其历史最优解实现信息合作.实验结果表明,丰富个体间的信息交互方式、自适应调整学习目标均能在不同程度上克服种群早熟收敛现象,在复杂多峰函数优化中也取得了很好的综合性能^[12-14].

第三类是混合不同的优化策略.由于不同的优化算法和搜索策略都有其各自的优点,为了借鉴其它优化算法(或策略)的先进思想,一些学者也尝试将其他优化算法(或策略)融入 PSO 以提高其性能.如 Angeline^[15]和 Higashi^[16]分别将遗传算法中的选择操作和杂交操作引入到 PSO 算法中,有效提高了算法的收敛速度. Xin B.^[17]将 PSO 和差分进化算法(Differential Evolution, DE)相结合,实现两种算法在不同搜索阶段的优势互补.此外,还有一些学者将反向学习策略^[18,19]、局部搜索策略^[20]、免疫算法^[21]等引入 PSO 算法中,都在不同程度上改善了算法的性能.

3 MSMP SO

尽管针对特定问题的性质为 PSO 选择合适的参数和邻居拓扑结构可以得到较好的求解结果,但很多工程应用是“黑盒”问题,因此为 PSO 预设最优的参数及邻居拓扑结构就变得十分困难.此外,利用种群或个体的历史行为信息合理调整其学习模式与参数能赋予 PSO 更好的鲁棒性和通用性^[22].基于此,本文提出了一种 MSMP SO 算法.该算法将两种典型的邻居拓扑结构相融合,同时,借助多种群的不同搜索行为所体现出的

性能来引导个体进行迁移,从而实现种群间的协作与计算资源的合理分配,并最终提升算法的综合性能。

3.1 扩展社会学习

在标准 PSO 中,“社会学习”部分所采用的邻居拓扑结构直接影响着算法的搜索特性,如全局结构(Global PSO, GPSO)有利于加快收敛速度,而局部结构(Local PSO, LPSO)则有利于保持种群多样性。这也使得这两种结构分别在单峰函数和多峰函数中表现出较好的性能。因此,为了充分利用这两种基本拓扑结构的优点,在 MSMP SO 中,粒子个体的“社会学习”包含 2 个部分,即 GPSO 和 LPSO。这样,式(1)被扩展为式(3)所示。

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^t + c_1 \cdot r_1 \cdot (pb_{i,j}^t - x_{i,j}^t) + c_2 \cdot r_2 \cdot (lb_{i,j}^t - x_{i,j}^t) + c_3 \cdot r_3 \cdot (gb_{i,j}^t - x_{i,j}^t) \quad (3)$$

式中, $lb_{i,j}^t$ 为环形拓扑结构下的邻居历史最优解; $gb_{i,j}^t$ 为种群历史最优解; c_1, c_2, c_3 为 3 个加速因子。为描述方便, $(lb_{i,j}^t - x_{i,j}^t)$ 和 $(gb_{i,j}^t - x_{i,j}^t)$ 分别被称为“局部社会学习”和“全局社会学习”部分。

3.2 子种群协作

式(3)中, 3 个加速因子表征着粒子个体在不同学习部分的权重,其取值大小决定着粒子搜索行为的一些特征。例如,当 c_1 取较大值, c_3 为取较小值时,种群的全局搜索能力较强,但收敛速度较慢;反之,当 c_1 取较小值, c_3 为取较大值时,种群的局部搜索能力较强,收敛速度较快,但全局搜索能力越差。鉴于上述分析,在 MSMP SO 中,种群被分为 3 个子种群,通过对 $c_1 \sim c_3$ 设定不同的取值组合赋予各子种群不同的搜索任务。具体细节如下:

(1) 子种群 1 负责保持种群多样性,目的是加强全局搜索能力。加速因子取值为: $c_1 = 2.0, c_2 = 1.0, c_3 = 0.2$;

(2) 子种群 2 负责加强局部搜索,目的是加快收敛速度。加速因子取值为: $c_1 = 0.2, c_2 = 1.0, c_3 = 2.0$;

(3) 子种群 3 兼顾两类搜索,起到较为平衡的作用。加速因子取值为: $c_1 = 1.0, c_2 = 1.0, c_3 = 1.0$ 。

需要指出的是,这里 3 组不同的 c_1, c_2, c_3 的组合取值为经验值。

在进行个体迁移操作时有两个问题需要明确。一是何时进行迁移。本文采用的是周期性执行该操作,即每隔 $cycle$ 代就执行个体迁移,这里 $cycle$ 为迁移周期。根据实验结果,本文取 $cycle = 10$ 。另一个问题是如何评价各子种群的搜索性能。本文综合考虑子种群历史最优解及平均适应值在一个迁移周期内的进化程度。具体如式(4)所示。

$$adv_i = (|gbf_{i,new} - gbf_{i,old}| / N_i) \times (|avgf_{i,new} - avgf_{i,old}|) \quad (4)$$

式中, $adv_i (1 \leq i \leq 3)$ 表示第 i 个子种群的优劣度,值越大表示该子种群性能越优; $gbf_{i,new}$ 和 $gbf_{i,old}$ 分别表示迁移周期到来时子种群新的最优适应值和旧的最优适应值; $avgf_{i,new}$ 和 $avgf_{i,old}$ 分别表示迁移周期到来时子种群中所有个体的最新历史最优解的平均适应值和旧的个体历史最优解平均适应值; N_i 为第 i 个子种群的规模大小。

从式(4)中可以看出,子种群在一个迁移周期内,其最优适应值及种群平均适应值改进程度越大,表明该子种群的性能越优。根据该结果,性能最差的子种群将随机选择一个粒子迁移至性能最优的子种群中。同时,这两个子种群将按照环形拓扑结构重新初始化其内部的邻居关系。迁移操作的目的一方面是为了及时将计算资源分配给表现更优的子种群,另一方面则是通过迁入的个体改善原子种群的多样性。

3.3 MSMP SO 算法流程

根据上述对各策略的介绍, MSMP SO 的伪代码可描述如下。

Step 1 初始化种群; $t = 1$;

Step 2 将种群随机等分为 3 个子种群: Pop_1, Pop_2, Pop_3 ; 为各子种群分配相应的加速因子;

Step 3 评价各子种群; 更新个体历史最优解、子种群最优解、全局历史最优解及子种群平均适应值;

Step 4 按照式(3)和(2)更新各子种群中个体的速度和位置; $t = t + 1$;

Step 5 评价个体; 更新个体历史最优解、子种群最优解、全局历史最优解;

Step 6 如果 $\text{mod}(t, \text{cycle}) \neq 0$ goto Step 4;

Step 7 记录各子种群新的历史最优解及子种群群体历史最优解的平均适应值;

Step 8 按照式(4)计算各子种群的优劣度;

Step 9 根据子种群的优劣度进行个体迁移操作; 更新相关子种群的邻居关系;

Step 10 判断算法是否满足停机条件,若满足则结束; 否则 goto Step4。

4 实验结果及分析

4.1 实验设置

本文选取的测试函数为 CEC2013 测试集。该测试集包含 28 个测试函数,根据特性可将其划分为 3 组: $f_1 - f_5$ 为单峰函数, $f_6 - f_{20}$ 为多峰函数, $f_{21} - f_{28}$ 为复合函数。测试集的详细信息见文献[23]。

为了验证 MSMP SO 算法的综合性能,本文选取 FDRPSO^[8]、FIPSO^[9]、CLPSO^[10]、PSODDS^[12]、SRPSO^[6] 和 CCPSO-ISM^[11] 作为对比算法。实验中,每个算法在任一测试函数上独立运行 30 次,测试函数的维数 $D = 30$,每次运行的最大评价次数 $MaxFEs = 10000D$ 。

4.2 实验结果

4.2.1 求解精度

本文将 30 次独立运行结果的均值及标准差来评价算法的求解精度. 具体结果见表 1 所示, 表中灰色背景表示相应算法在测试函数上求解结果的均值最优, 最后一行的数值表示相应算法在所有测试函数上取得最优结果的个数.

从表 1 可以看出, MSMP SO 在 12 个函数上取得了最优的均值结果, FIP SO 和 CLP SO 则分别在 7 个和 5 个函数上得到了最优的结果. 从这一指标来看, MSMP SO 明显要优于其它对比算法. 进一步分析可以看出,

MSMP SO 在多峰函数和复合函数上的优势更为明显, 因为其取得的 12 次最优结果有 11 次是集中在这 2 类函数上, 而在 5 个单峰函数上仅在 f_3 上取得的最优均值. 而 FIP SO 则在单峰函数上表现出更优的性能, 因为其不仅在 f_1 和 f_3 上取得的最优均值, 而且在对这 2 个函数各自的 30 次求解中均找到了理论最优解. 通过结果可以看出, MSMP SO 所采用的基于多种群的自适应迁移策略较好地利用了全局结构和局部结构两种邻居拓扑方式, 合理分配了计算资源, 尤其是在多峰函数及复合函数等复杂问题上表现出较优的性能.

表 1 7 个算法在 CEC2013 测试集上 30 次独立运行的求解结果

	FDRPSO	FIPSO	CLPSO	PSODDS	SRPSO	CCPSO-ISM	MSMPSO
f_1	4.55e-14 ± 7.28e-14	0.00e+00 ± 0.00e+00	7.58e-14 ± 1.01e-13	8.45e+01 ± 1.46e+02	1.31e+02 ± 2.01e+02	3.78e-11 ± 6.58e-11	1.67e-13 ± 8.89e-14
f_2	3.56e+05 ± 1.53e+05	3.69e+05 ± 1.51e+05	1.58e+07 ± 3.04e+06	1.16e+06 ± 1.30e+06	2.94e+06 ± 2.22e+06	5.74e+06 ± 3.30e+06	4.66e+06 ± 4.22e+06
f_3	2.06e+07 ± 2.38e+07	5.31e+10 ± 2.97e+10	4.46e+07 ± 2.13e+07	7.78e+09 ± 7.80e+09	3.74e+09 ± 4.67e+09	1.92e+07 ± 2.37e+07	1.55e+07 ± 1.37e+07
f_4	1.11e+03 ± 4.84e+02	3.83e+04 ± 9.77e+03	1.34e+04 ± 2.84e+03	5.66e+03 ± 4.03e+03	2.11e+02 ± 7.89e+01	2.23e+02 ± 1.26e+02	1.26e+03 ± 1.21e+03
f_5	4.05e-13 ± 9.15e-14	0.00e+00 ± 0.00e+00	1.14e-13 ± 0.00e+00	6.11e+01 ± 6.72e+01	1.14e+02 ± 1.48e+02	3.74e-12 ± 3.53e-12	2.43e-13 ± 6.57e-14
f_6	1.39e+01 ± 8.12e+00	1.77e+01 ± 6.57e+00	2.61e+01 ± 7.69e+00	6.05e+01 ± 2.95e+00	6.95e+01 ± 1.90e+01	6.37e+01 ± 2.32e+01	7.91e+01 ± 2.85e+01
f_7	3.11e+01 ± 1.28e+01	2.63e+02 ± 8.36e+01	5.48e+01 ± 7.44e+00	1.05e+02 ± 1.98e+01	4.44e+01 ± 1.96e+01	6.40e+00 ± 3.30e+00	1.99e+01 ± 1.13e+01
f_8	2.09e+01 ± 4.09e-02	2.09e+01 ± 4.39e-02	2.09e+01 ± 3.98e-02	2.09e+01 ± 4.61e-02	2.10e+01 ± 2.91e-02	2.09e+01 ± 4.06e-02	2.09e+01 ± 5.45e-02
f_9	1.84e+01 ± 3.06e+00	1.99e+01 ± 2.84e+00	2.70e+01 ± 1.23e+00	2.61e+01 ± 2.26e+00	1.79e+01 ± 2.96e+00	1.71e+01 ± 2.98e+00	1.45e+01 ± 2.68e+00
f_{10}	1.28e-01 ± 5.43e-02	1.59e-01 ± 5.75e-02	7.43e-01 ± 2.57e-01	7.63e+01 ± 4.89e+01	9.07e+01 ± 6.78e+01	1.18e-01 ± 4.60e-02	8.67e-02 ± 3.79e-02
f_{11}	2.86e+01 ± 5.41e+00	4.12e+01 ± 2.11e+00	1.57e-11 ± 2.73e-11	7.98e+01 ± 2.17e+01	3.86e+01 ± 8.66e+00	2.39e+01 ± 6.32e+00	2.59e+01 ± 6.38e+00
f_{12}	5.52e+01 ± 1.18e+01	2.91e+01 ± 1.06e+01	1.03e+02 ± 1.04e+01	1.62e+02 ± 3.56e+01	5.60e+01 ± 1.46e+01	3.91e+01 ± 1.34e+01	5.16e+01 ± 1.67e+01
f_{13}	1.18e+02 ± 1.75e+01	8.23e+01 ± 2.45e+01	1.30e+02 ± 1.48e+01	2.71e+02 ± 4.49e+01	1.19e+02 ± 2.15e+01	9.44e+01 ± 2.42e+01	1.04e+02 ± 2.70e+01
f_{14}	1.24e+03 ± 2.06e+02	2.01e+03 ± 1.69e+03	5.39e+01 ± 1.01e+01	2.00e+03 ± 4.55e+02	1.37e+03 ± 2.20e+02	1.40e+03 ± 3.17e+02	1.44e+03 ± 3.86e+02
f_{15}	3.85e+03 ± 7.63e+02	5.84e+03 ± 5.91e+02	5.28e+03 ± 3.34e+02	4.09e+03 ± 3.99e+02	3.02e+03 ± 6.53e+02	3.67e+03 ± 1.28e+03	3.21e+03 ± 4.22e+02
f_{16}	2.01e+00 ± 3.03e-01	2.47e+00 ± 2.46e-01	2.44e+00 ± 1.90e-01	8.72e-01 ± 4.15e-01	2.50e+00 ± 2.32e-01	2.39e+00 ± 2.33e-01	2.29e+00 ± 3.62e-01
f_{17}	7.13e+01 ± 8.90e+00	1.11e+02 ± 1.93e+01	5.00e+02 ± 2.54e+00	1.08e+02 ± 1.48e+01	6.16e+01 ± 7.09e+00	5.85e+01 ± 6.96e+00	5.60e+01 ± 5.86e+00
f_{18}	1.28e+02 ± 5.92e+01	1.74e+02 ± 7.54e+00	2.03e+02 ± 1.04e+01	1.71e+02 ± 3.18e+01	1.99e+02 ± 2.02e+01	1.86e+02 ± 1.55e+01	6.95e+01 ± 1.44e+01
f_{19}	3.18e+00 ± 6.91e-01	6.65e+03 ± 1.23e+04	3.74e+00 ± 3.29e-01	1.02e+01 ± 5.16e+00	4.05e+00 ± 1.81e+00	2.67e+00 ± 4.17e-01	3.27e+00 ± 6.16e-01
f_{20}	1.47e+01 ± 4.57e-01	1.47e+01 ± 4.57e-01	1.31e+01 ± 3.90e-01	1.42e+01 ± 8.48e-01	1.39e+01 ± 1.29e+00	1.23e+01 ± 1.64e+00	9.92e+00 ± 6.68e-01
f_{21}	3.29e+02 ± 8.36e+01	1.46e+01 ± 4.88e-01	2.80e+02 ± 4.17e+01	3.42e+02 ± 8.51e+01	3.44e+02 ± 7.94e+01	3.05e+02 ± 5.51e+01	3.02e+02 ± 7.55e+01
f_{22}	1.21e+03 ± 2.40e+02	3.55e+02 ± 8.82e+01	1.96e+02 ± 3.51e+01	2.35e+03 ± 4.31e+02	1.64e+03 ± 3.63e+02	1.25e+03 ± 3.43e+02	1.33e+03 ± 3.46e+02
f_{23}	3.83e+03 ± 7.07e+02	6.46e+03 ± 1.21e+02	5.86e+03 ± 3.48e+02	4.98e+03 ± 7.14e+02	3.37e+03 ± 5.86e+02	3.34e+03 ± 9.50e+02	3.12e+03 ± 6.29e+02
f_{24}	2.40e+02 ± 1.38e+01	2.73e+02 ± 7.83e+00	2.58e+02 ± 7.72e+00	2.76e+02 ± 6.36e+00	2.60e+02 ± 1.12e+01	2.54e+02 ± 1.36e+01	2.39e+02 ± 1.45e+01
f_{25}	2.83e+02 ± 7.18e+00	2.83e+02 ± 6.64e+00	2.89e+02 ± 3.55e+00	2.99e+02 ± 7.98e+00	2.89e+02 ± 1.06e+01	2.82e+02 ± 7.90e+00	2.76e+02 ± 6.29e+00
f_{26}	3.00e+02 ± 6.01e+01	3.07e+02 ± 7.13e+01	2.01e+02 ± 3.36e-01	2.43e+02 ± 6.32e+01	2.79e+02 ± 6.79e+01	2.87e+02 ± 5.80e+01	2.37e+02 ± 5.36e+01
f_{27}	7.44e+02 ± 1.05e+02	9.30e+02 ± 6.29e+02	8.14e+02 ± 2.32e+02	9.99e+02 ± 9.51e+01	7.77e+02 ± 9.73e+01	7.53e+02 ± 5.43e+01	7.15e+02 ± 1.26e+02
f_{28}	4.39e+02 ± 2.81e+02	7.54e+02 ± 5.76e+02	3.00e+02 ± 8.35e-11	1.10e+03 ± 4.79e+02	6.53e+02 ± 4.14e+02	4.41e+02 ± 2.45e+02	2.93e+02 ± 1.29e+01
	2	7	5	2	2	3	12

4.2.2 收敛过程

为了更直观地呈现算法的收敛过程, 本文给出了不同算法对多个测试函数进行单次优化的收敛曲线. 由于篇幅有限, 这里选取了 11 个测试函数, 包括 3 个单峰函数($f_1 - f_3$)、4 个多峰函数($f_7 - f_{10}$)和 4 个复合函数($f_{22} - f_{24}$). 对比算法在 3 类测试函数上的收敛过程分别见图 1、图 2 及图 3 所示.

从图 1 可以看出, MSMP SO 在 f_1 和 f_3 上表现出较好的收敛性能. 具体而言, MSMP SO 和 FIP SO 在 f_1 上迅速地找到了全局最优解. 在 f_3 上, 尽管 MSMP SO 前期的收敛速度要略慢于 FDRPSO, 但 MSMP SO 在后期的搜索性能更优, 因此其取得了比 FDRPSO 更高精度的解. 图 2 的结果表明 MSMP SO 在多峰函数上取得了很好的综合性能. 因为 MSMP SO 不仅在 f_8 上表现出最快的收敛

速度和最高的求解精度,而且在 f_{10} 上也表现出较为稳定、持续的收敛过程和较高的求解精度.此外,MSMPSO 和 FDRPSO 在 f_7 上也取得了类似的迅速的收敛过程.在 4 个复合函数上的收敛结果显示,MSMPSO 在 f_{23} 和

f_{24} 上得到了最好的表现,同时,在 f_{21} 上也取得了仅次于 CCPSO-ISM 的表现.通过在 3 类函数上的比较可以看出,MSMPSO 在算法求解精度和加快收敛过程方面取得了较好的综合性能.

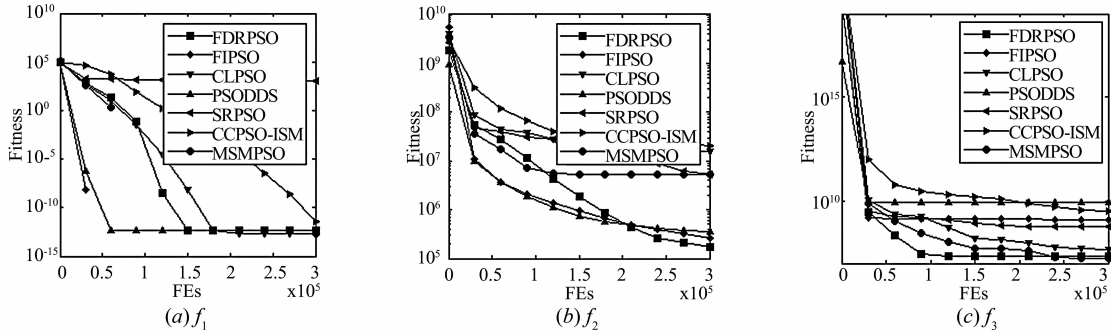


图1 对比算法在单峰函数上的收敛曲线

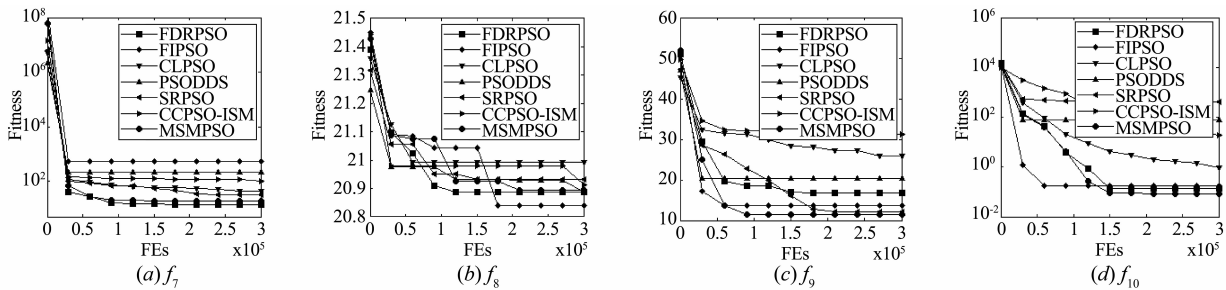


图2 对比算法在多峰函数上的收敛曲线

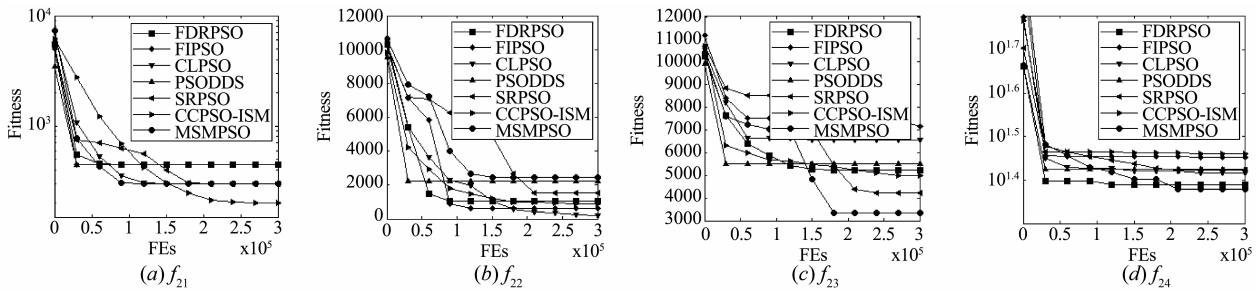


图3 对比算法在复合函数上的收敛曲线

4.2.3 显著性检验

(1) t -检验. 这里,我们对 MSMPSO 与其它 PSO 算法在 28 个测试函数上的优化结果进行了 t -检验,检验结果如表 2 所示.表中“+”、“-”和“=”分别表示 MSMPSO 在相应函数上明显优于、劣于对比算法或二者之间不存在显著性差异. B 、 S 和 W 分别表示 MSMPSO 比相应的对比算法在 t -检验结果中具有显著更优、相同和显著更劣的函数个数. $B-W$ 表示 MSMPSO 显著性优于对比算法的函数个数与其显著性劣于相应算法的函数个数的差值,该值越大说明相应算法在 t -检验中体现的性能越差.

从表 3 可以看出,MSMPSO 比其它 6 个算法具有更好的综合性能,因为在 6 个函数上的 $B-W$ 值均大于 0.需要注意的是,尽管 FDRPSO 和 CCPSO-ISM 取得最优均值的函数个数较少(见表 1),但其 $B-W$ 值较小,即这

两个算法在 t -检验中表现出较好的性能.

(2) Friedman-检验. 本文对 7 个算法在 CEC2013 测试集上的整体表现进行了 Friedman-检验.同时,为了进一步分析各算法在不同类型函数上的性能差异,还分别对这些算法在单峰函数、多峰函数和复合函数上的性能分别进行了检验.实验结果如表 3 所示.表中,算法的 Rank 值越小表示该算法越优,显著性概率 $p < 0.05$ 则表明结果各算法性能存在显著性差异.

从表 3 可以看出,MSMPSO 在整个测试集上具有最好的性能表现.但在单峰函数的检验结果表明,FDRPSO 具有最好的性能,MSMPSO 则取得了第二好的性能.需要注意的是,尽管 CCPSO-ISM 在整个测试集上取得了仅次于 MSMPSO 的表现,但其在单峰函数上的表现不甚理想.此外,在单峰函数上测试结果的 p 值为

表 2 MSMP SO 与其它 6 个算法 t -检验结果

	FDRPSO	FIPSO	CLPSO	PSODDS	SRPSO	CCPSO-ISM
f_1	=	-	=	+	+	+
f_2	-	-	=	-	-	=
f_3	=	+	+	+	+	+
f_4	-	+	+	+	-	-
f_5	+	-	-	+	+	+
f_6	-	-	-	=	-	=
f_7	=	+	=	+	+	-
f_8	=	=	=	=	+	=
f_9	+	=	+	=	=	=
f_{10}	=	+	+	+	+	=
f_{11}	=	+	-	+	=	=
f_{12}	+	-	+	+	=	=
f_{13}	+	=	+	+	=	=
f_{14}	-	+	-	=	-	=
f_{15}	+	=	=	=	=	+
f_{16}	=	=	+	=	+	+
f_{17}	+	+	-	+	=	=
f_{18}	+	=	=	+	=	=
f_{19}	=	+	+	+	+	-
f_{20}	=	=	+	=	+	+
f_{21}	=	=	-	=	=	=
f_{22}	=	+	-	=	=	=
f_{23}	=	+	+	=	=	=
f_{24}	=	+	+	+	=	=
f_{25}	=	=	+	=	+	=
f_{26}	=	+	-	=	+	=
f_{27}	=	+	+	=	=	+
f_{28}	+	+	+	+	+	+
B	8	14	14	14	12	8
S	16	9	6	13	12	17
W	4	5	8	1	4	3
$B-W$	4	9	6	13	8	5

0.650,这也说明这些函数在单峰函数上的性能不存在显著性差异.在多峰函数和复合函数的 Friedman-检验结果可以看出,MSMPSO 和 CCPSO-ISM 均取得了最优和次优的结果,这也说明对子种群间或个体间的信息进行合理的共享与利用有利于复杂问题的优化.

4.2.4 加速因子性能

本节将利用实验来说明不同加速因子对算法性能的影响以及 MSMPSO 的复合加速因子策略的性能特征.鉴于篇幅有限,本文仅选取了多峰函数 f_6 和复合函数 f_{21} 进行实验,收敛曲线如图 4 所示.为描述方便,图中 Global 表示算法的整个种群选用的加速因子注重全局搜索,Local 表示算法的整个种群选用的加速因子注重局部搜索,Balance 则表示算法的整个种群选用的加速因子注重平衡搜索.具体的加速因子取值见 3.2 节.

从图中可以看出,在 2 个测试函数上,Local 的前期收敛速度最快,但种群很快就陷入局部最优.Global 和 Balance 的表现也不尽如人意,不仅收敛速度慢,其寻优能力也较弱.而 MSMPSO 则表现出较好的综合性能,尽管其在进化初期的收敛速度要略慢于 Local,但其中后期跳出局部最优的能力更强.可以说,MSMPSO 中采用的多种群迁移模式充分利用了不同加速因子的优势.

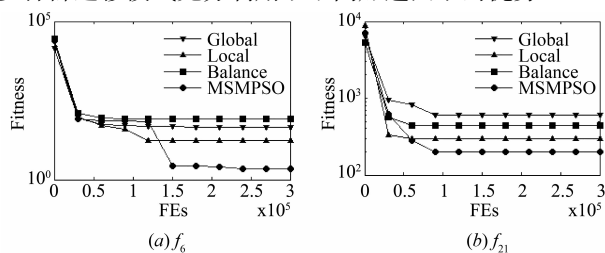


图 4 不同加速因子组合在 f_6 和 f_{21} 上的收敛曲线

表 3 7 个算法在 CEC2013 测试集上均值结果的 Friedman-检验分析

Average Rank	所有函数($f_1 - f_{28}$)		单峰函数($f_1 - f_5$)		多峰函数($f_6 - f_{20}$)		复合函数($f_{21} - f_{28}$)	
	Algorithm	Rank	Algorithm	Rank	Algorithm	Rank	Algorithm	Rank
1	MSMPSO	2.48	FDRPSO	3.00	MSMPSO	2.70	MSMPSO	1.63
2	CCPSO - ISM	3.09	MSMPSO	3.20	CCPSO - ISM	2.77	CCPSO - ISM	3.13
3	FDRPSO	3.29	FIPSO	3.60	FDRPSO	3.37	CLPSO	3.19
4	CLPSO	3.93	CCPSO - ISM	4.00	CLPSO	4.23	FDRPSO	3.31
5	SRPSO	4.80	CLPSO	4.20	SRPSO	4.80	SRPSO	4.81
6	FIPSO	4.93	SRPSO	4.80	FIPSO	4.90	FIPSO	5.94
7	PSODDS	5.45	PSODDS	5.20	PSODDS	5.23	PSODDS	6.00
Statistic	44.959		4.2000		22.263		27.471	

5 结束语

本文提出了一种 MSMPSO 算法,该算法将 GPSO 和 LPSO 两种模型融入 PSO 的“社会学习”部分,将该部分

扩展为“局部社会学习”和“全局社会学习”两个部分.同时,通过不同加速因子的设置将种群划分为具有不同搜索特性的 3 个子种群.在此基础上,通过周期性地对子种群进行性能评估,从而指导粒子的迁移操作,实

现了高效搜索策略的有效传播及计算资源的合理分配. 实验结果表明 MSMP SO 具有较好的综合性能, 尤其是在复杂的多峰函数表现出较好的鲁棒性.

需要指出的是, MSMP SO 中对 3 个加速因子及迁移周期的选择并非最优. 一方面, 这些参数对算法在不同优化问题中的影响有待进一步探究; 另一方面, 参数之间存在的相关性也需要进行深入的定性与定量分析. 这些问题将在我们后续工作中进一步研究.

参考文献

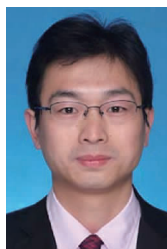
- [1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. Proceedings of the IEEE International Conference on Neural Networks [C]. Piscataway: IEEE Press, 1995. 1942 – 1948.
- [2] Wang D S, Tan D P, Liu L. Particle swarm optimization algorithm: an overview [J]. Soft Computing, DOI 10.1007/s00500-016-2474-6
- [3] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization [A]. Proceedings of IEEE Congress on Evolutionary Computation [C]. Seoul, Korea: IEEE Press, 2001. 101 – 106.
- [4] Ratnaweera A, Saman K H. Self-organizing hierarchical particle swarm 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, et al. Adaptive particle swarm optimization [J]. IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 2009, 39(6): 1362 – 1381.
- [6] Tanweer M R, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm [J]. Information Sciences, 2015, 294: 182 – 202.
- [7] Suganthan P N. Particle swarm optimizer with neighborhood operator [A]. Proceedings of IEEE Congress on Evolutionary Computation [C]. Washington D C, USA: IEEE Press, 1999. 1958 – 1962.
- [8] Peram T, Veeramachaneni K, Mohan C K. Fitness-distance-ratio based particle swarm optimization [A]. Proceedings of Swarm Intelligence Symp. [C]. Indianapolis, Indiana, USA: IEEE Press, 2003. 174 – 181.
- [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] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transaction on Evolutionary Computation, 2006, 10(3): 281 – 295.
- [11] Li Y H, Zhan Z H, Lin S, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems [J]. Information Sciences, 2015, 293(3): 370 – 382.
- [12] Jin X, Liang Y Q, Tian D P, et al. Particle swarm optimization using dimension selection methods [J]. Applied Mathematics and Computation, 2013, 219: 5185 – 5197.
- [13] 夏学文, 桂凌, 戴志锋 等. 基于多尺度选择性学习和探测-收缩机制的 PSO 算法 [J]. 电子学报, 2016, 44(5): 1090 – 1100.
XIA Xue-wen, GUI Ling, DAI Zhi-feng et al. A PSO algorithm based on multiscale-selective-learning and detecting-shrink strategies [J]. Acta Electronica Sinica, 2016, 44(5): 1090 – 1100. (in Chinese)
- [14] Liu Q, Wei W, Yuan H, et al. Topology selection for particle swarm optimization [J]. Information Sciences, 2016, 363(1): 154 – 173.
- [15] Angeline P J. Using selection to improve particle swarm optimization [A]. Proceedings of IEEE International Conference on Evolutionary Computation [C]. Alaska, USA: IEEE Press, 1998. 84 – 89.
- [16] Higashi H, Iba H. Particle swarm optimization with gaussian mutation [A]. Proceedings of IEEE swarm intelligence symposium [C]. Indianapolis, Indiana, USA: IEEE Press, 2003. 72 – 79.
- [17] Xin B, Chen J, Peng Z H, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization [J]. Science China (Information Sciences), 2010, 53(5): 980 – 989.
- [18] 周新宇, 吴志建, 王晖 等. 一种精英反向学习的粒子群算法 [J]. 电子学报, 2013, 41(8): 1647 – 1652.
Zhou Xin-yu, Wu Zhi-jian, Wang Hui et al. Elite opposition-based particle swarm optimization [J]. Acta Electronica Sinica, 2013, 41(8): 1647 – 1652. (in Chinese)
- [19] 喻飞, 李元香, 魏波 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, Wei Bo et al. The Application of a Novel OBL based on lens imaging principle in PSO [J]. Acta Electronica Sinica, 2014, 42(2): 230 – 235. (in Chinese)
- [20] Xia X W, Liu J N, Hu Z B. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space [J]. Applied Soft Computing, 2014, 23: 76 – 90.
- [21] 刘朝华, 李小花, 章兢. 精英免疫克隆选择的协同进化粒子群算法 [J]. 电子学报, 2013, 41(11): 2167 – 2173.
LIU Zhao-hua, LI Xiao-hua, ZHANG Jing. Co-evolutionary particle swarm optimization algorithm based on elite immune clonal selection [J]. Acta Electronica Sinica, 2013, 41(11): 2167 – 2173. (in Chinese)

- [22] Xia X W, Xie C W, Wei B, et al. Particle swarm optimization using multi-level adaptation and purposeful detection operators [J]. *Information Sciences*, 2017, 385: 174 - 195.
- [23] Liang J J, Qu B Y, Suganthan P. N. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization [R]. Nanyang Technological Univ, Singapore, 2013.

作者简介



邓先礼 男,1978 年生,湖北十堰人,硕士,现为华东交通大学软件学院讲师,研究方向为计算智能、复杂网络.



魏波 男,1983 生,湖北天门人,博士,博士后,现为华东交通大学软件学院讲师,研究方向为计算智能、机器学习.

曾辉 男,1980 年生,湖北麻城人,博士,现为新疆工程学院计算机工程系副教授,研究方向为计算智能.

桂凌 女,1977 年生,湖北应城人,学士,现为华东交通大学经济管理学院实验师,研究方向为智能物流.

夏学文(通信作者) 男,1974 年生,湖北云梦人,博士,博士后,现为华东交通大学软件学院副教授,研究方向为计算智能、机器学习.

E-mail: xwxia@whu.edu.com