

基于自主学习和精英群的多子群粒子群算法

姜海燕^{1,2}, 王芳芳¹, 郭小清¹, 庄嘉祥¹

(1. 南京农业大学信息科学技术学院, 南京 210095; 2. 国家信息农业工程技术中心, 南京 210095)

摘要: 为了提高动态多子群粒子群算法中粒子学习的自主性, 提出一种基于自主学习和精英群的粒子群算法. 该算法借鉴教育心理学自主学习的理念, 用基础群中粒子自主选择学习对象的操作代替子群的重组操作, 并通过精英群局部搜索的配合来达到寻优的目的. 将所提出的算法应用于6个测试函数, 并与动态多子群PSO等算法进行了比较, 比较结果表明, 新算法在提高收敛速度、精度和寻优时间等方面具有良好的性能.

关键词: 粒子群优化; 多子群; 精英群; 自主学习; 多样性

中图分类号: TP18

文献标志码: A

Multi-swarm particle swarm optimization based on autonomic learning and elite swarm

JIANG Hai-yan^{1,2}, WANG Fang-fang¹, GUO Xiao-qing¹, ZHUANG Jia-xiang¹

(1. School of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, China; 2. National Engineering and Technology Center for Information Agriculture, Nanjing 210095, China. Correspondent: JIANG Hai-yan, E-mail: jianghy@njau.edu.cn)

Abstract: In order to promote the particles' autonomic learning ability in the dynamic multi-swarm particle swarm optimization algorithm, a multi-swarm particle swarm optimization algorithm based on autonomic learning and elite swarm is proposed. According to the concept of autonomic learning, the algorithm can achieve the purpose of optimization through replacing the regrouping operation with choosing learning objects autonomously in base swarms and conducting local searches in elite swarms. The proposed algorithm is applied to six test functions and compared with the dynamic multi-swarm particle swarm optimization algorithm. The results show that the proposed algorithm has a good performance in the convergence speed, the search accuracy, the time consumption and so on.

Key words: particle swarm optimization; multi-swarm; elite swarm; autonomic learning; diversity

0 引言

粒子群优化算法(PSO)^[1]具有概念简单和收敛速度较快等优点, 已被广泛应用于工程技术和科学研究等领域^[2]. 但是, PSO在应用中往往出现易陷入局部极值点导致早熟现象^[3], 进化后期收敛慢导致精度较差等问题^[4]. 造成PSO早熟的直接原因是粒子只接受其自身当前最好位置和种群当前最好位置的指导, 群体进化很快就失去多样性, 使算法易陷入局部极值点, 这种现象在解决复杂多峰优化问题时显得更为突出. 为了避免早熟, 研究者提出通过增加粒子多样性来提高算法的全局搜索能力. 多子群策略是目前增加粒子多样性的重要方法之一^[5-10].

Lovbjerg^[5]于2001年首次提出了多子群策略, 即将群体划分为多个子群分别独立进化, 各粒子以其所在子群的最优粒子lbest代替社会学习对象gbest, 进而增加粒子向不同粒子学习的概率, 避免全体粒子收敛于同一个局部极值点. 此后, 众多研究者对多子群策略做了进一步研究, 尤其以动态多子群PSO(DMS-PSO)较为突出. 在DMS-PSO^[6]中, 各子群分别独立进化, 并周期性地合并和随机重组操作, 有效地提高了粒子多样性. 实验表明, DMS-PSO提高了复杂多峰函数上的性能. 但另一方面, 多样性与收敛速度一直是矛盾的, DMS-PSO增加了粒子多样性的同时, 由于缺乏有效的局部搜索, 即使找到全局最优区域, 也

收稿日期: 2013-07-28; 修回日期: 2013-10-29.

基金项目: 国家自然科学基金项目(30971697); 国家863计划项目(2013AA100404); 国家科技支撑计划项目(2011BA-D21B03); 南京农业大学教学改革重点项目(2013Z004).

作者简介: 姜海燕(1967-), 女, 教授, 博士, 从事计算智能、作物系统建模与仿真等研究; 王芳芳(1988-), 女, 硕士生, 从事计算智能的研究.

难以快速收敛于全局最优位置^[7]. 因此, 研究者在保持 DMS-PSO 粒子多样性的同时改进其局部搜索能力. Liang 等^[7]在 DMS-PSO 的基础上, 周期性地将所有的子群 lbest 按适应度值大小排序, 选择其中前 25% 运用拟牛顿法来改善局部搜索能力. Zhao 等^[10]将和声算法 (HS) 引入到 DMS-PSO, 形成 DMS-PSO-HS 算法, 周期性地对全体粒子的当前最好位置 pbest 应用 HS, 以提高局部搜索能力, 仿真实验表明, DMS-PSO-HS 在单峰和多峰函数上表现突出.

以上改进算法主要依靠 DMS-PSO 周期性地重组操作来改善全局搜索能力, 并以典型的局部搜索算法来改善局部搜索能力. 实验发现, 周期性地重组操作虽然提高了固定代数下的寻优精度, 但寻优时间也相应地增加, 而且在固定时间下的精度并没有得到提高. 进一步分析 DMS-PSO 和其改进算法的重组过程可以发现: 粒子在周期性重组之后, 与子群中其他粒子并没有直接的信息交流, 其本质只是改变了所处子群的 lbest 学习对象. 由此可以想到, 是否可以不进行重组操作, 而是周期性地改变粒子的学习对象就能达到与重组操作相同的效果. 此外, 以上算法由于同一子群的不同粒子均学习相同的社会学习对象 lbest, 忽视了粒子在进化过程中的自主性, 进而无法合理地反映粒子个体差异和认知能力对其进化的影响, 造成粒子的进化带有一定的盲目性和强迫性, 产生较多无用的搜索, 降低了算法的性能.

针对以上问题, 本文提出一种基于自主学习和精英群的多子群粒子群算法 (ALEMSPSO). 算法引入自主学习的理念, 基础群中的粒子可以根据已学习对象对自身的领导能力自主地选择学习对象, 提高了粒子学习的自主性, 并通过精英群局部搜索的配合, 提高了算法的收敛速度和精度.

1 ALEMSPSO 算法

1.1 算法进化模型

教育心理学中, 将教师的领导方式分为专制型、自主型和放任型^[11]. 在自主型领导方式下, 教师的作用在于引导和调动学生的自主性和积极性, 学生能有效发挥学习主体的自主性, 工作绩效最高.

鉴于这样的思想, 研究标准 PSO 和 DMS-PSO 中粒子更新机制后发现: 粒子不能自主地选择学习对象, 每次迭代更新时, 必须向全局最优粒子 gbest 或者所在子群的 lbest 学习, 带有强迫性, 很大程度上扼杀了粒子进化的自主性和多样性, 从而导致粒子迅速向其所在子群的最优位置靠拢. 如果该位置为一局部最优点, 则算法将陷入局部最优. 本文将自主学习的思想引入到 DMS-PSO 中, 通过粒子自主选择学习对象的操作来代替原来的子群重组操作, 希望达到与周期性重组相同的效果.

ALEMSPSO 算法将整个种群划分为 n 个相同规模的子群, 各个子群根据其搜索行为的不同, 可分为

$n-1$ 个基础群和 1 个精英群. 基础群起主导作用, 精英群起辅助搜索的作用. 基础群中的粒子是普通粒子, 精英群中的粒子由适应度值较好的个体扰动生成. 精英群平均划分成 $n-1$ 个更小的子群, 称为精英小子群, 与基础群一一对应. 初始种群生成时, 首先随机生成 $n-1$ 个基础群, 计算每个子群中的最优粒子 lbest, 并以 lbest 为种子, 随机扰动生成基础群对应的精英小子群. 基础群采用自主学习的进化策略, 精英群采用局部搜索策略. 基础群进行自主学习时, 粒子每隔 R 代 (学习对象选择周期) 后, 通过判断已学习对象对自身的领导能力来自主选择或随机选择下一周期的学习对象. 精英群和基础群各自进化 L 代 (通信周期) 后进行通信, 分别对比精英小子群中最优粒子 Lbest 和对应基础群 lbest. 若 Lbest 较优, 则将其值直接赋予基础群中某一粒子; 否则, 重新以对应基础群 lbest 为种子随机扰动生成精英小子群. 为简单起见, 本文将学习对象选择周期 R 和通信周期 L 视为同一个周期, 即均为 R . 在进化后期, 由于只需在某一个粒子附近进行搜索, 无需同时追逐多个学习对象, 算法可以通过对基础群 lbest 多样性的度量实现子群合并操作, 以节约计算资源. 算法进化过程如图 1 所示.

1.2 基础群的进化策略——自主学习

1.2.1 自主选择学习对象

基础群中粒子更新公式如下:

$$\begin{aligned} v_{ij}^{t+1} = & \omega v_{ij}^t + c_1 r_1 (\text{pbest}_{ij}^t - x_{ij}^t) + \\ & c_2 r_2 (\text{lbest}_{ij}^t - x_{ij}^t), \end{aligned} \quad (1a)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}. \quad (1b)$$

其中: $i = 1, 2, \dots, N_1$, N_1 为基础群粒子总数; $j = 1, 2, \dots, \text{dim}$, dim 为问题维度; $t = 1, 2, \dots, T-1$, T 为最大迭代次数; x_{ij}^t, v_{ij}^t 分别为粒子的位置和速度; lbest_i^t 为粒子 i 的自我认知对象, 即自身最优位置; lbest_i^t 为粒子 i 的社会学习对象, 即某子群的最优粒子; c_1, c_2 为学习因子; r_1, r_2 为 $[0, 1]$ 内的随机数; ω 为惯性权重.

基础群自主选择学习对象的策略是: 通过衡量上一周期已学习对象对自身的领导能力来自主选择下一周期学习对象, 即粒子可以周期性地自主选择所在子群 lbest, 也可以选择其他子群 lbest 作为社会学习对象. 这里的周期称为学习对象选择周期 R . 领导能力的计算如下:

设 $f(\text{pbest}_i(t))$ 为第 t 代中, 粒子 i 历史最优位置 lbest 的适应度值. 如果待解问题是 1 个最小化问题, 则有

$$\begin{aligned} \delta_{\text{pbest}_i}(t, t-1) = \\ f(\text{pbest}_i(t)) - f(\text{pbest}_i(t-1)) \leq 0. \end{aligned} \quad (2a)$$

其中: $t = 2, 3, \dots, T$, $\delta_{\text{pbest}_i}(t, t-1)$ 为粒子 i 间隔一代的改进度. 粒子经历了向某个子群 lbest 学习 R 代后, 应满足

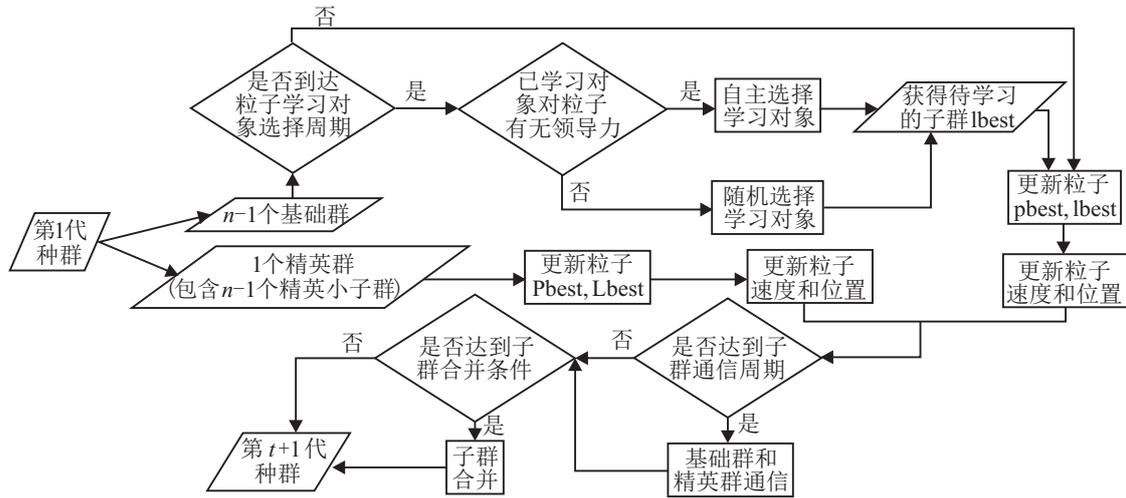


图1 ALEMSPSO算法进化模型

$$\delta_{\text{pbest}_i}(t, t-R) \leq 0. \quad (2b)$$

其中: $t = 2R, 3R, \dots, T$; $\delta_{\text{pbest}_i}(t, t-R)$ 为粒子 i 间隔 R 代的改进度. 基于这个改进度, 子群 lbest 对粒子 i 的领导能力可以划分为如下2种情形:

1) 有领导能力, 即 $\delta_{\text{pbest}_i}(t, t-R) < 0$.

在该情况下, 由于在学习对象 lbest 的领导下, 粒子 i 的 lbest 有所提升, 意味着该子群 lbest 具备带领其寻找更优解的潜力. 因此, 可以判断子群 lbest 对粒子 i 具有领导能力, 粒子在下一个 R 代内仍向该子群 lbest 学习.

2) 无领导能力, 即 $\delta_{\text{pbest}_i}(t, t-R) = 0$.

在该情况下, 可以判断子群 lbest 对粒子 i 没有领导能力, 粒子 i 在下一个 R 代内随机选择其他子群 lbest 学习, 选择方式见 1.2.2 节.

1.2.2 随机选择学习对象

粒子自主选择学习对象时, 需要利用粒子的历史信息来度量子群 lbest 对粒子的领导能力, 因此, 粒子在初始选择对象时, 采用随机选择方法确定社会学习对象. 另外, 当粒子判断已学习对象对自身无领导力时, 也可采用随机方法重新选择学习对象, 以提升粒子的多样性.

通过2个随机数 rand1 、 rand2 来选择学习对象, 这2个随机数称为学习对象选择的随机因子. rand1 决定粒子是否向所在子群 lbest 学习. 当粒子不向所在子群 lbest 学习时, 有均等的概率向其他某个子群 lbest 学习. rand2 用来确定子群, 每个子群分别对应一个概率区间, 如果 rand2 值落入某个子群对应的概率区间内, 则选择该子群的 lbest 为学习对象, 公式表示如下:

$$\text{lbest}_i^t = \begin{cases} \text{lbest}_k, & \text{rand1} \leq a; \\ \text{lbest}_l, & l \neq k, \text{rand1} > a, \\ & \min_l < \text{rand2} < \max_l. \end{cases} \quad (3)$$

其中: $t = 1, R+1, \dots, T-R+1$; k 为粒子 i 所在子群

的编号, $k = 1, 2, \dots, n-1$; l 为其他子群的编号, $l = 1, 2, \dots, n-1$; lbest_i^t 为粒子 i 在第 t 代时的学习对象; a 为 $[0, 1]$ 内的常数, 是粒子向所在子群 lbest 学习的概率阈值; \max_l 和 \min_l 为子群 l 对应概率区间的上下界.

1.3 精英群的进化策略——局部搜索

1.3.1 精英小子群的局部搜索

PSO 必须同时具有全局搜索能力和局部搜索能力才能表现出良好的性能. 目前常见的做法是在进化初期强化全局搜索能力, 并在后期强化局部搜索能力^[12]. 但这样往往会因为进化前期不能进行及时有效的局部搜索而使粒子只找到最优解附近的区域就受到其他粒子影响跳出了该搜索区域, 降低了收敛速度和精度. 本文算法在整个进化过程中兼顾全局搜索和局部搜索. 基础群和精英群分别独立地进化, 每隔通信周期 R 代后进行通信, 用精英小子群在其对应基础群 lbest 附近小范围的局部搜索结果代替基础群粒子, 既能使算法减少早熟, 又能提高收敛速度和精度.

精英小子群中的粒子更新公式类似于基础群中的粒子, 但搜索范围不同, 表示如下:

$$[\text{Region}_{j \max}, \text{Region}_{j \min}] = b[\text{region}_{j \max}, \text{region}_{j \min}]. \quad (4)$$

其中: $[\text{Region}_{j \max}, \text{Region}_{j \min}]$ 为精英小子群 j 维上搜索范围的上下界; $[\text{region}_{j \max}, \text{region}_{j \min}]$ 为优化问题, 即基础群 j 维上搜索范围的上下界; b 为 $[0, 1]$ 间的常数, 用以控制局部搜索范围缩小的幅度.

1.3.2 精英小子群的生成

精英小子群以对应基础群 lbest 为种子随机扰动生成. 生成方式如下:

$$X_{k1j} = \text{lbest}_{kj}, \quad (5a)$$

$$X_{kIj} = \text{lbest}_{kj} + 0.5(\text{Region}_{j \max} - \text{Region}_{j \min})\text{rand}(). \quad (5b)$$

其中: $I = 1, 2, \dots, N_2/(n-1)$, N_2 为精英群粒子总数; $k = 1, 2, \dots, n-1$; X_{kIj} 为第 k 个精英小子群中第 I

个粒子 j 维上的位置; $lbest_{kj}$ 为第 k 个基础群中的最优粒子 j 维上的位置; $rand()$ 为 $[-1, 1]$ 内的随机数.

1.4 子群合并操作

种群进化到一定代数以后, 如果大部分的 $lbest$ 均到达同一个目标区域内, 则粒子不管向哪个子群 $lbest$ 学习, 结果都相差无几, 此时继续分群反而会耗费计算资源. 本文采取的操作是度量基础群 $lbest$ 的多样性来决定子群合并操作, 即将 $n - 1$ 个基础群合并为 1 个大的基础群, 精英小子群合并为 1 个大的精英群. 精英群仍与基础群对应, 并且分别进行局部搜索和全局搜索.

本文将 $diversity(t) = \frac{1}{cn(t)}$ 称为基础群 $lbest$ 的多样性值, $cn(t)$ 统计第 t 代时基础群 $lbest$ 与基础群 $lbest$ 平均值的差值在每一维上均满足设定阈值的个数之和, 计算方程如下:

$$cn(t) = \sum_{k=1}^{n-1} F(lbest_k(t)); \tag{6a}$$

$$F(lbest_k(t)) = \begin{cases} 1, & \forall j \text{ 使得 } |lbest_{kj}(t) - \overline{lbest}_j(t)| < \varepsilon; \\ 0, & \exists j \text{ 使得 } |lbest_{kj}(t) - \overline{lbest}_j(t)| \geq \varepsilon; \end{cases} \tag{6b}$$

$$\overline{lbest}_j(t) = \frac{\sum_{k=1}^{n-1} lbest_{kj}(t)}{n - 1}. \tag{6c}$$

其中: $t = 1, 2, \dots, T$; $\overline{lbest}_j(t)$ 为各基础群 $lbest$ 的平均值; ε 为设定的阈值. 当 $diversity(t) < 1/[c(n - 1)]$ 时, 基础群和精英群分别进行合并操作, c 为 $[0, 1]$ 间的常数.

2 仿真实验与分析

2.1 实验设置

实验硬件环境为 Intel(R) Core(TM)i3-2120 CPU (3.30 GHz), 内存为 2 GB, 在 Mtalab 7.6 下完成.

本文选用文献[6]的 6 个经典测试函数进行验证: f_1 : Sphere; f_2 : Rosenbrock; f_3 : Griewank; f_4 : Rastrigrin; f_5 : Ackley; f_6 : Weierstrass. 测试函数的维度 dim 设为 10. 由于各函数寻优复杂程度的不同, f_2 精度要求为 2, f_4 精度要求为 1, 其他函数精度要求为 $1e-5$.

评价算法寻优性能优劣的指标包括平均适应度值 (Mean) 和平均代数 (MI)^[13]. 此外, 增加度量算法性能排名的指标 (Rank), 根据 Mean 升序排名. 算法稳定性通过标准差 (Deviation)^[13] 评价.

本文选取 3 个改进的 PSO 进行对比, 即 DMS-PSO^[6]、串行实现的 IPPSO^[8] 和 LDWPSO^[14]. 为了降低随机因子对实验结果的波动影响, 本文涉及的实验均重复运行 50 次, 即 $EN = 50$, 实验结果取 50 次的平均值. 为了增强可比性, 不同 PSO 算法设置了统一的参数, 即种群大小 $N = 300$, 最大迭代次数 $T = 1000$. 其余参数参照原文.

2.2 ALEMSPSO 算法参数确定

ALEMSPSO 中如表 1 所示的 5 个参数均依据经验得出.

表 1 ALEMSPSO 参数设置

参数符号	参数值
a	0.4
b	0.2
c	0.66
ε	0.05
ω	从 0.7 线性递减至 0.2

此外, 通信周期 R 和子群个数 n 通过实验获得.

1) 通信周期 R .

通信周期 R 是基础群和精英群相邻 2 次通信之间的间隔和基础群中粒子重新选择学习对象的间隔. 通信周期太长, 则算法将退化为只有基础群的模型, 不能发挥精英群局部精细搜索的优势, 基础群中的粒子也不能周期性地选择学习对象; 若周期过短, 则容易产生通信瓶颈, 导致算法效率降低. 为此, 需要通过实验获取合适的通信周期.

实验 1 测试函数为 f_1, f_4, f_5 . 停止准则设为 $T = 500$, 粒子总个数分为 300、600、900 三种情况, 子群

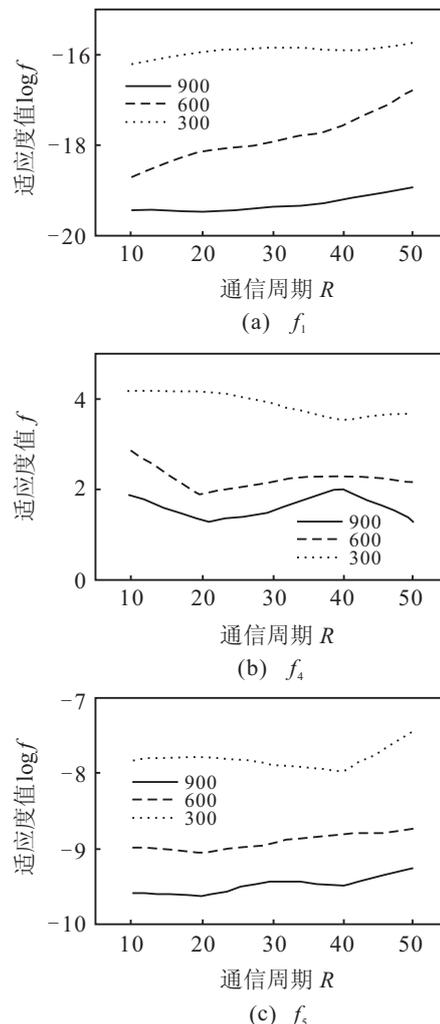


图 2 ALEMSPSO 在不同粒子数下的平均适应度值随通信周期的变化情况

个数为6,通信周期从第10代递增至50代,步长为10.统计平均适应度值的变化情况,结果如图2所示.为了清晰地反应曲线的性质,图2(a)和图2(c)的纵坐标为平均适应度值,取以10为底的对数.

由图2可以看出,在粒子数相同的情况下,随着通信周期的增加,适应度值没有明显的变化趋势.随着粒子数的增加,解的精度逐渐提高,但最适通信周期无明显变化趋势.综合考虑实验中的3个函数,当 $R = 20$ 时,各函数都能得到比较理想的结果,因此将 R 定为20.在应用算法时,建议将 R 设为20.

2) 子群个数 n .

子群个数是指基础群的个数加上一个精英群的总个数.一般而言,目标函数的局部极值点越多,子群个数越多,越有利于跳出局部极值.但考虑到基础群与精英小子群之间的通信开销,且精英小子群中需要一定量的粒子,子群个数不宜设为过大值.为此,通过实验获取合适的子群个数.

实验2 测试函数为 f_1, f_4, f_5 ,停止准则设为 $T = 500$,通信周期为20,粒子总个数分300和840两种情况.子群个数分别从2递增至6,从2递增至8,步长

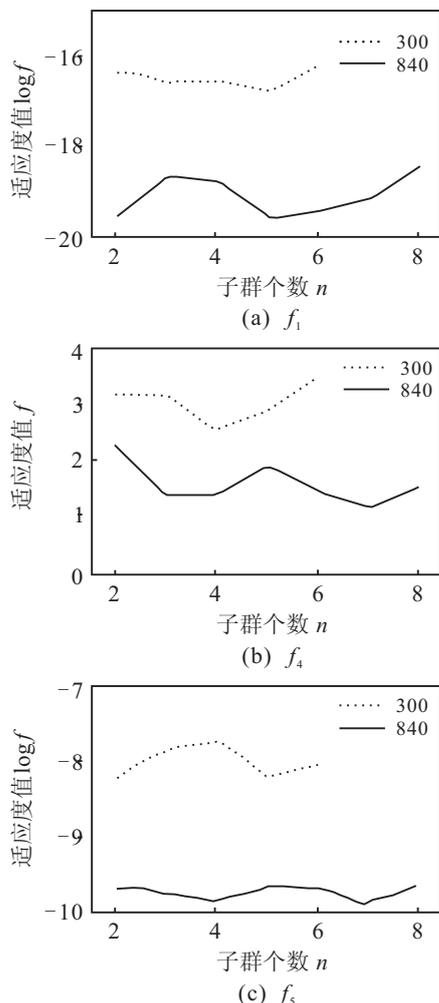


图3 ALEMSPSO在不同粒子数下的平均适应度值随子群个数的变化情况

均为1.统计平均适应度值随子群个数的变化情况,结果如图3所示.

由图3可以看出:粒子数为300的情况下,子群个数为5时3个函数均能取得较好的结果;在粒子数为840的情况下,子群个数为7时效果最好.在这2种情况下,精英小子群的个数分别为12和20.在具体应用算法时, n 的选取应使得精英小子群中的粒子数 $N/n(n-1)$ 不能过大也不能过小,在 $[10,20]$ 内为宜.因此, n 建议在 $\left[\frac{1}{2} + \frac{\sqrt{25+5N}}{10}, \frac{1}{2} + \frac{\sqrt{25+10N}}{10}\right]$ 内取值.在本文中,子群个数定为5.

2.3 不同算法的对比实验与分析

实验分别在2种方案下进行:一种是固定代数,另一种是固定时间.

实验3 固定代数实验.对象为ALEMSPSO和各对比算法,停止准则设为 $T = 1000$,实验结果如表2所示.

实验4 固定时间实验.对象为ALEMSPSO和各对比算法,停止准则设为固定运行时间.由于各测试函数寻优的复杂程度不同,本文保证算法大致运行600~900代, f_6 固定运行30s,其他函数固定运行3s,实验结果见表2.

由表2可以看出:

1) ALEMSPSO在固定代数和固定时间2种情况下的综合排名分别为1.3和1.2,在所有对比算法中均列第一,尤其是在 f_1 函数上,Mean分别比其他算法中的最优结果提高了55和24个数量级,精度得到很大提升.然而,ALEMSPSO在 f_4 上并没有达到最好,但仍优于DMS-PSO.

2) 在固定代数情况下,ALEMSPSO在6个测试函数上的MI都是最小的,从而说明ALEMSPSO能在最少的代数内满足精度要求,其中在 f_1 上需要177代,在 f_5 上需要330代,而其他算法中,速度最快的为493代和681代,ALEMSPSO的收敛速度大大提高.然而,在固定时间的情况下,MI在6个函数上均不是最大,说明每代运行时间不是最短,这是因为算法要进行选择学习对象和通信等操作,增加了额外的计算消耗.但是在相同的时间内,新算法的精度是最高的,因为算法能在较少的代数内获得更高的精度,总体性能得到很大的提升,说明额外的计算消耗是值得的.另外,DMS-PSO在2种实验方案下的MI均为最差,这是因为它增加粒子多样性的同时,丧失了局部空间的搜索能力,降低了收敛速度,而且频繁地重组操作增加了运行时间,使得在相同时间下DMS-PSO运行代数最少.

3) ALEMSPSO在标准差的指标上也有很好的结

果, 实验 3 中在 f_1 和 f_6 上分别达到 $8.17e-92$ 和 0, 显示了算法的稳定性。

为了更直观地说明本文算法求解 6 个测试函数的优越性, 以图形方式给出每个函数固定代数下的仿真结果, 如图 4 所示。每幅图均反映各算法在求解测试函数时对应的平均适应度值的变化趋势。

由图 4 可以看出, ALEMSPSO 能够迅速定位到最优解的附近区域, 在短时间内找到正确的搜索方向, 下降速度较快。而其他算法具有一定的盲目性, 没有发挥主体的自主性, 有选择地选择学习对象, 再加上没有局部搜索的配合, 因此下降得比较缓慢, 减缓了算法的收敛速度和精度。

表 2 固定代数和固定时间下各算法的统计结果

函数	评价指标	固定代数(实验 3)				固定时间(实验 4)			
		LDWPSO	DMS-PSO	IPPSO	ALEMSPSO	LDWPSO	DMS-PSO	IPPSO	ALEMSPSO
f_1	Mean	3.15e-37	5.52e-12	3.89e-31	1.59e-92	3.13e-20	1.962 593	3.2e-14	1.08e-44
	Deviation	1.18e-36	1.02e-11	1.24e-30	8.17e-92	2.18e-19	0.549 747	9.69e-14	6.03e-44
	MI	496.62	956.84	493.48	177.38	862.6	492.82	749.76	746.08
	Rank	2	4	3	1	2	4	3	1
f_2	Mean	2.835 826	5.912 558	3.519 514	1.980 804	3.573 438	31.335 74	4.380 635	2.515 851
	Deviation	0.469 172	0.545 154	0.660 869	0.346 375	0.478 636	5.796 323	0.742 482	0.366 42
	MI	993.92	1000	988.18	927.5	851.94	631.8	773.88	767.62
	Rank	2	4	3	1	2	4	3	1
f_3	Mean	0.062 256	0.113 264	0.062 274	0.049 927	0.085 042	6.501 302	0.106 496	0.059 385
	Deviation	0.020 228	0.072 352	0.021 81	0.023 306	0.037 933	1.640 272	0.051 578	0.024 933
	MI	921.14	994.72	939.28	737.1	760.66	605.6	688.96	648.36
	Rank	3	4	2	1	2	4	3	1
f_4	Mean	0.517 524	4.918 13	0.399 474	0.656 675	0.862 828	35.836 13	1.266 116	0.895 925
	Deviation	0.696 084	2.630 509	0.628 65	0.706 629	0.977 379	4.617 578	0.993 58	0.802 222
	MI	776.58	999.4	785.46	563.38	816.08	626.88	770.88	726.1
	Rank	2	4	1	3	1	4	3	2
f_5	Mean	4.44e-15	0.000 212	6.43e-15	4.37e-15	4.21e-08	10.029 23	1.33e-05	4.44e-15
	Deviation	0	0.000 853	3.42e-15	4.97e-16	1.54e-07	0.805 061	1.71e-05	7.11e-16
	MI	669.34	998.98	681.28	329.84	773.14	547.1	688.08	666.48
	Rank	2	4	3	1	2	4	3	1
f_6	Mean	0	0.000 931	0 0	1.45e-08	5.741 18	3.93e-05	0	0
	Deviation	0	0.000 717	0	0	2.74e-08	0.407 541	3.09e-05	0
	MI	742.72	1000	757.16	384.94	793.58	711.74	740.1	749.86
	Rank	2	4	3	1	2	4	3	1
Comprehensive Rank		2.2	4	2.5	1.3	1.8	4	3	1.2

注: 实验 3 的 MI 是指达到精度要求时的平均收敛代数, 实验 4 的 MI 是指在固定时间内运行的平均代数。

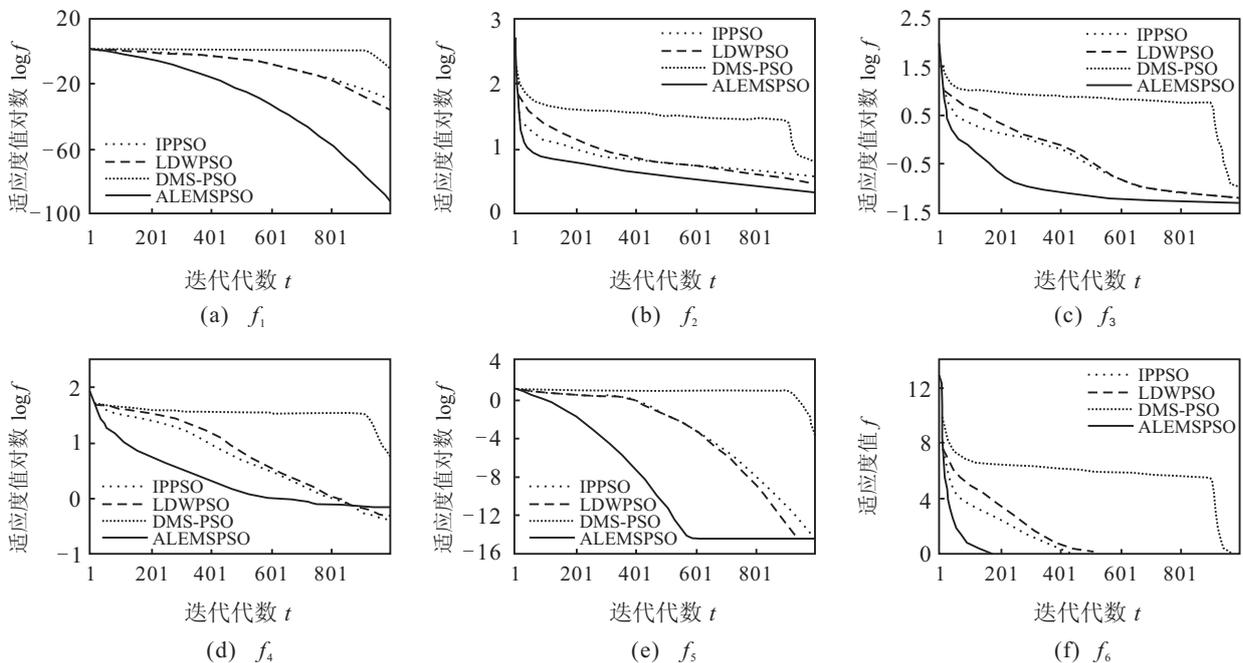


图 4 各算法在不同函数上的适应度变化曲线

3 结 论

本文提出了一种基于自主学习和精英群的多子群粒子群算法(ALEMSPSO). 基础群中的粒子通过自主选择学习对象操作来代替动态多子群PSO中的重组操作, 并通过精英群局部搜索的配合提高了算法的收敛速度和精度. 此外, 算法可通过对基础群最优粒子多样性的度量, 实现子群的合并操作, 减少无用的搜索, 节约计算资源. 将算法应用到6个测试函数, 与动态多子群PSO等算法比较的结果表明, 新算法在提高收敛速度、精度和时间方面具有良好的性能, 稳定性优良, 可以作为一种有效、稳健、快速的PSO改进算法.

本算法尚存一些不足, ALEMSPSO在Rastrigrin上部分指标未能达到最好, 需进一步改进. 同时, 对于子群个数的选取, 本文所做的实验仅考虑了粒子数为300和840的情况, 对于更多粒子数的情形尚有待于进一步验证.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Perth: IEEE Press, 1995: 1942-1948.
- [2] 纪震, 廖惠连, 吴青华. 粒子群算法及应用[M]. 北京: 科学出版社, 2009: 235-244.
(Ji Z, Liao H L, Wu Q H. Particle swarm algorithm and its application[M]. Beijing: Science Publishing House, 2009: 235-244.)
- [3] Bergh F V D. An analysis of particle swarm optimizers[D]. Pretoria: Faculty of Natural and Agricultural Science, University of Pretoria, 2006.
- [4] 倪庆剑, 邢汉承, 张志政, 等. 粒子群优化算法研究进展[J]. 模式识别与人工智能, 2007, 20(3): 349-357.
(Ni Q J, Xing H C, Zhang Z Z, et al. Survey of particle swarm optimization algorithm[J]. Pattern Recognition and Artificial Intelligence, 2007, 20(3): 349-357.)
- [5] Lovbjerg M, Rasmussen T K, Krink T, et al. Hybrid particle swarm optimizers with breeding and subpopulations[C]. Proc of the Genetic and Evolutionary Computation Conf. San Francisco: Morgan Kaufmann, 2001: 469-476.
- [6] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer[C]. Proc of IEEE Int Swarm Intelligence Symposium. Piscataway: IEEE Press, 2005: 124-129.
- [7] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer with local search[C]. Proc of IEEE Congress on Evolutionary Computation. Edinburgh: IEEE Press, 2005: 522-528.
- [8] 黄芳, 樊晓平. 基于岛屿群体模型的并行粒子群优化算法[J]. 控制与决策, 2006, 21(2): 175-180.
(Huang F, Fan X P. Parallel particle swarm optimization algorithm with island population model[J]. Control and Decision, 2006, 21(2): 175-180.)
- [9] 吴宪祥, 郭宝龙, 王娟. 基于Lotka-Volterra模型的双群协同竞争粒子群优化算法[J]. 控制与决策, 2010, 25(11): 1619-1624.
(Wu X X, Guo B L, Wang J. Lotka-Volterra model based particle swarm optimization[J]. Control and Decision, 2010, 25(11):1619-1624.)
- [10] Zhao S Z, Suganthan P N, Pan Q K, et al. Dynamic multi-swarm particle swarm optimizer with harmony search[J]. Expert Systems with Applications, 2011, 38(4): 3735-3742.
- [11] 申荷永. 充满张力的生活空间-勒温的动力心理学[M]. 武汉: 湖北教育出版社, 2001: 77-79.
(Shen H Y. Tension-filled living space—Lewin's dynamic psychology[M]. Wuhan: Hubei Education Press, 2001: 77-79.)
- [12] 陶新民, 刘福荣, 刘玉, 等. 一种多尺度协同变异的粒子群优化算法[J]. 软件学报, 2012, 23(7): 1805-1815.
(Tao X M, Liu F R, Liu Y, et al. Multi-Scale cooperative mutation particle swarm optimization algorithm[J]. J of Software, 2012, 23(7): 1805-1815.)
- [13] 曾华, 吴耀华. 基于进化停滞周期的局部变异PSO算法及其收敛性分析[J]. 控制与决策, 2010, 25(9): 1333-1337.
(Zeng H, Wu Y H. PSO algorithm with local mutation in evolution stagnation cycle and its convergence analysis[J]. Control and Decision, 2010, 25(9): 1333-1337.)
- [14] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. Proc of IEEE Int Congress on Evolutionary Computation. Anchorage: IEEE Press, 1998: 69-73.

(责任编辑: 闫 妍)