

多策略并行学习的异构粒子群优化算法

王芸^{1*}, 孙辉^{1,2,3}

(1. 南昌工程学院 信息工程学院, 南昌 330099; 2. 南昌工程学院 协同感知与先进计算技术研究所, 南昌 330099;

3. 江西省水信息协同感知与智能处理重点实验室, 南昌 330099)

(*通信作者电子邮箱 5219476@qq.com)

摘要:针对标准粒子群优化(PSO)算法在复杂问题上收敛速度慢和早熟收敛的缺点,提出了一种多策略并行学习的异构 PSO 算法(MHPSO)。该算法首先从种群多样性和跳出局部极值的角度提出了两种新学习策略(局部扰动学习策略和高斯子空间学习策略),并将这两种策略与 MBB-PSO 策略融合组成高效稳定的策略池。其次提出了一种简单有效的策略更换机制,指导粒子迭代寻优中何时更换学习策略。基准测试函数的实验结果表明,改进的粒子群优化算法在求解精度和收敛速度上得到极大的提高。与一些改进 PSO 算法(如自适应的粒子群优化(APSO)算法等)相比,所提算法具有更优良的寻优性能。

关键词:粒子群优化算法;局部扰动学习策略;高斯子空间学习策略;策略池;策略更换

中图分类号: TP301.6 **文献标志码:** A

Heterogenous particle swarm optimization algorithm with multi-strategy parallel learning

WANG Yun^{1*}, SUN Hui^{1,2,3}

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang Jiangxi 330099, China;

2. Research Laboratory of Cooperative Sensing and Advanced Computing Techniques, Nanchang Institute of Technology, Nanchang Jiangxi 330099, China;

3. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Jiangxi 330099, China)

Abstract: The standard Particle Swarm Optimization (PSO) suffers from the premature convergence problem and the slow convergence speed problem when solving complex optimal problems, so a Heterogenous PSO with Multi-strategy parallel learning (MHPSO) was presented. Firstly two new learning strategies, named local disturbance learning strategy and Gaussian subspace learning strategy respectively, were proposed to maintain the population's diversity and jump out from the local optima. And an efficient and stable strategy pool was constructed by combining the above two strategies with the existed one (MBB-PSO); Secondly, a simpler and more effective strategy change mechanism was proposed, which could guide particles when to change the learning strategy. The experimental study on a set of classical test functions show that the proposed approach improves the solution accuracy and convergence speed greatly, and has a superior performance in comparison with several other improved PSO algorithms, such as APSO (Adaptive Particle Swarm Optimization).

Key words: Particle Swarm Optimization (PSO) algorithm; local disturbance learning strategy; gaussian subspace learning strategy; strategy pool; strategy change

0 引言

1995 年 Kennedy 和 Eberhart 通过模拟鸟群捕食行为提出了粒子群优化(Particle Swarm Optimization, PSO)算法^[1],该算法由于概念简单、高效且易实现,一提出就吸引了国内外学者的研究兴趣,并在众多领域中得到了成功的应用。

虽然 PSO 算法表现了良好的寻优性能,但在解决复杂优化问题时,易陷入局部最优且收敛速度较慢。为此人们对 PSO 算法提出了多种改进策略。如 Zhan 等^[2]提出自适应的粒子群优化(Adaptive Particle Swarm Optimization, APSO)算法,该算法依据种群分布和粒子适应值确定算法处于探索、开发、收敛还是陷入局部最优等哪种状态,并根据不同的状态自适应调整惯性权重、加速因子等参数,同时对全局最优粒子采

用精英学习策略帮助逃离局部最优。Liang 等^[3]为解决标准 PSO 算法在多峰问题上表现较差的问题,提出一个广泛学习的 PSO (Comprehensive Learning Particle Swarm Optimizer for global optimization of multimodal functions, CLPSO)算法,在该算法中每个粒子的各维可以向不同粒子的好经验进行学习,从而有效地保证了算法的多样性。

此外,许多学者将几种不同特点的学习策略融合组成策略池^[4-6]。迭代进化中,算法实时调整个体的学习策略,从策略池中选择更合适的策略指导个体完成全局寻优,即异构优化算法。如伍大清等^[7]提出了一种基于混合策略自适应学习的粒子群优化算法(improved parallel Particle Swarm Optimization algorithm with Hybrid strategy and self-adaptive Learning, HLPSPSO),该算法从收敛速度、跳出局部极值、探索、

收稿日期:2015-05-06;修回日期:2015-07-13。

基金项目:国家自然科学基金资助项目(61261039,61305150);教育部人文社科青年基金交叉项目(13YJCZH174);江西省教育厅落地计划项目(KJLD13096);江西省科技厅自然科学基金资助项目(20122BAB201043,20151BAB207067,20151BAB207032)。

作者简介:王芸(1979-),女,江西南昌人,讲师,硕士,主要研究方向:群智能优化算法;孙辉(1959-),男,江西九江人,教授,博士,主要研究方向:群智能优化算法、Rough 集与粒计算、变分不等原理。

开发几个不同角度融合了 4 种具有不同优势的变异策略,面对不同形态的复杂问题时通过自适应学习机制选择合适的策略来完成全局寻优。Li 等^[8]针对粒子群优化算法单一的学习模式易导致粒子群多样性的严重缺失,且不能很好地处理各种复杂优化问题的缺点,对每个粒子设置 4 个学习策略,并自适应地选择最适合自身的策略,提出自学习粒子群优化算法(Self-Learning Particle Swarm Optimizer for global optimization problems, SLPSO)。

由于各种学习策略在进化过程中所起的作用不同,因此有效控制和协调不同的学习策略就成为异构优化算法设计的核心问题。为此本文提出了一种多策略并行学习的异构 PSO 算法(Heterogenous PSO with Multi-strategy parallel learning, MHPSO)。该算法首先提出了两种新的学习策略:一种是消除速度更新公式中的惯性部分,引入扰动因子对粒子的个体最优实施扰动,有效地保持种群的多样性;另一种是提出了高斯子空间学习策略,通过对粒子的子空间实施高斯学习,有效增强了算法逃离局部最优的能力,并将这两种策略结合修改的 Bare-Bones PSO(Modified Bare-Bones PSO, MBB-PSO)组成 MHPSO 算法的策略池。其次,提出一种简单有效的策略更换机制,即,当粒子的适应值在连续的迭代次数内未发生改善时,则指导粒子更换当前学习策略从而完成全局寻优。通过经典基准函数仿真实验,验证了本文算法的优良性能。

1 标准 PSO

标准 PSO 中,每个粒子都有一个位置矢量和速度矢量。位置矢量代表求解问题的一个候选解,算法通过适应值函数评价每个候选解的优劣。速度矢量决定粒子的下一步飞行位置。每次迭代时,粒子按照式(1)和式(2)进行速度和位置更新:

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 r_{1j}(t)(\mathbf{pbest}_i(t) - \mathbf{x}_i(t)) + c_2 r_{2j}(t)(\mathbf{gbest}_j(t) - \mathbf{x}_i(t)) \quad (1)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (2)$$

其中: $\mathbf{v}_i(t)$ 表示第 t 次迭代时粒子 i 在第 j 维解空间的速度; $\mathbf{x}_i(t)$ 表示第 t 次迭代时粒子 i 在第 j 维解空间的位置; $\mathbf{pbest}_i(t)$ 表示粒子 i 的个体历史最优位置; $\mathbf{gbest}_j(t)$ 表示在第 t 次迭代时群体中的最优位置; ω 为惯性权重,用来衡量算法全局搜索能力和局部开采能力; c_1 和 c_2 为加速因子,分别用来衡量粒子个体历史最优位置和群体最优位置对下一次迭代飞行的影响,取值通常为 2; r_{1j} 和 r_{2j} 是符合均匀分布的随机数,即, $r_{1j}, r_{2j} \sim U(0,1)$ 。

2 多策略并行学习的异构 PSO 优化算法

2.1 策略池的构建

2.1.1 局部扰动学习策略

文献[9]通过严格的数学推导,证明了粒子在寻优过程中将聚集到由个体最优 \mathbf{pbest} 和全局最优 \mathbf{gbest} 决定的极值 \mathbf{p}^* 之上。

$$\lim_{t \rightarrow \infty} (\mathbf{x}(t)) = \mathbf{p}^* = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2} \mathbf{pbest} + \frac{c_2 r_2}{c_1 r_1 + c_2 r_2} \mathbf{gbest} \quad (3)$$

从式(3)可知,所有粒子在寻优过程中如果都没有找到比 \mathbf{pbest} 和 \mathbf{gbest} 更优的位置,则进化过程将处于停滞状态,

粒子逐渐聚集到 \mathbf{p}^* 。如果能在寻优过程中,不断调整粒子的自身个体最优 \mathbf{pbest} ,这将使得粒子们由 \mathbf{p}^* 转移聚集到一个新的位置,这个过程中所经历的新的搜索路径和领域将会提高发现更优解的概率。

因此,本文引入一扰动因子 λ , 在迭代搜索过程中对粒子的自身个体最优进行扰动,产生新的“个体引导者” \mathbf{pbest}^* 。同时去除速度公式中的惯性部分,使得粒子仅通过对自身的思考,和与群体中其他粒子的比较和模拟来调整本次寻优时的移动速度,加快算法的收敛速度。

新的速度更新公式如下:

$$\mathbf{v}_i(t+1) = c_1 r_{1j}(\mathbf{pbest}_i^*(t) - \mathbf{x}_i(t)) + c_2 r_{2j}(\mathbf{gbest}_j(t) - \mathbf{x}_i(t)) \quad (4)$$

$$\mathbf{pbest}_i^*(t) = \lambda \cdot \mathbf{pbest}_i(t) \quad (5)$$

其中: $\mathbf{pbest}_i^*(t)$ 是粒子学习的新对象,位于粒子自身个体最优的领域范围内,它可能比粒子自身个体最优更优秀。粒子在迭代寻优过程中,不再向“个体引导者”学习,而是向新的“个体引导者” \mathbf{pbest}^* 和“全局引导者” \mathbf{gbest} 学习,这有助于粒子勘探更广的搜索区域,有效地维持种群的多样性。

在式(5)中,扰动因子 λ 的值随着函数评估次数线性变化,其计算公式如下:

$$\lambda = 0.1 - (0.1 - 0.01) \cdot \text{fes}/\text{MAX_FEs} \quad (6)$$

其中: fes 表示函数当前的评估次数, MAX_FEs 表示函数的最大评估次数。算法初期, λ 的值较大,对粒子的个体历史最优实现大范围的扰动,从而勘探其他的搜索领域,保持了种群的多样性;在算法后期, λ 的值逐渐减小,使得算法能够进行局部精细搜索,提高了解的精度。

本文将该策略称为局部扰动学习策略(Local Disturbance learning Strategy, LDS)。

2.1.2 高斯子空间学习策略

复杂问题求解时,大多数改进的 PSO 算法进化后期易陷入局部最优。文献[2]提出了精英高斯学习策略,随机选择全局最优粒子的一维空间实施高斯学习,使得全局最优粒子有可能逃离局部最优,进入其他潜在更优解的区域,这将使得群体中的其他粒子也跟随跳出局部区域,聚集到一个新的搜索区域。

受精英高斯学习策略思想的启发,本文提出对粒子 i 随机选择其多维子空间位置进行高斯变异。对粒子 i 的第 d 维位置 x_i^d 的高斯学习方式如下:

$$x_i^d = x_i^d + \text{Gaussian}(\mu, \sigma^2) \quad (7)$$

在式(7)的基础上,本文对粒子 i 随机选择 M 维子空间实施高斯学习。如果粒子高斯学习后的适应值更优,则取代原粒子;否则保留原粒子。式(7)中高斯随机数的方差 σ^2 随粒子高斯学习的子空间维数而变化:

$$\sigma^2 = \frac{1}{2} (x_{\max}^d - x_{\min}^d) \frac{j}{M} \quad (8)$$

其中: j 取值分别为 $1, 2, \dots, M$; M 表示粒子进行高斯学习的子空间维数。

本文将这种学习策略称为高斯子空间学习策略(Gaussian Subspace learning Strategy, GSS)。

2.1.3 MBB-PSO 策略

Kennedy 等^[10]认为 PSO 算法在迭代过程中,粒子的位置分布呈现高斯随机数特征,提出了 MBB-PSO 算法。在该算法

中,粒子没有速度向量,只有位置向量。粒子的位置更新公式如下:

$$x_{i,j,G+1} = \begin{cases} pbest_{ij}(t), & rand_j(0,1) < 0.5 \\ N\left(\frac{gbest_j + pbest_{i,j,G}}{2}, |gbest_j - pbest_{i,j,G}|\right), & \text{其他} \end{cases} \quad (9)$$

其中: $x_{i,j,G+1}$ 为第 i 个粒子在第 $G+1$ 次迭代时的位置; N 表示均值为 $(gbest_j + pbest_{i,j,G})/2$, 方差为 $|gbest_j - pbest_{i,j,G}|$ 的高斯随机数; $rand_j(0,1)$ 是第 j 维解空间产生的随机数。在该算法的进化初期,由于个体历史最优位置和全局最优位置相差较远,有利于 MBB-PSO 算法进行勘探,随着算法的迭代进行,方差趋近于 0,则有利于算法后期的局部开采。

本文提出的 MHPSO 算法综合利用局部扰动学习策略、高斯子空间学习策略和 MBB-PSO 策略的特点,共同构成算法的策略池,使得各个学习策略在算法进化中具有不同的作用。

2.2 粒子学习策略的更换

异构优化算法在搜索过程中很难确定从勘探转入开采的最佳时机,进化过程中粒子何时更换学习策略以及选择哪个策略影响着异构优化算法的寻优性能。Engelbrecht 等在文献[4]中提出通过监视粒子自身个体最优的变化,决定是否从策略池中为该粒子随机分配另一个策略;Wang 等在文献[11]中提出个体只要在本次迭代的适应值未改善,则为该个体随机分配另一个策略指导其下一次迭代寻优。

本文在综合文献[4]和文献[11]的基础上,提出当粒子的适应值在一定的迭代次数内未发生改善时,则更换粒子的当前学习策略。

策略的更换算法如下:

```
if (粒子  $i$  当前的适应值未发生改善) then
     $count_i++$ ;
    if (粒子  $i$  的  $count_i$  值大于阈值  $K$ ) then
        为粒子  $i$  随机分配另一个学习策略;
         $count_i = 0$ ;
    end if
end if
```

以上伪代码中, $count_i$ 记录粒子 i 的适应值连续未发生改善的次数; K 为粒子适应值连续未发生变化的迭代次数的阈值,通过反复实验,本文中设定为 200。

2.3 MHPSO 算法流程

新算法步骤如下:

步骤 1 设定算法的各参数、种群规模以及阈值 K 。

步骤 2 初始化种群中各粒子的位置、速度及个体历史最优值,计算种群全局最优值。

步骤 3 随机初始化群体中粒子的学习策略。

步骤 4 对种群中粒子 i 做以下操作:

4.1) 速度和位置的更新:

```
if 粒子  $i$  的学习策略 = LDS 策略 then
    使用式(4)和式(2)对粒子速度和位置更新;
if 粒子  $i$  学习策略 = CSS 策略 then
    利用式(7)和式(8)对粒子位置更新;
if 粒子  $i$  学习策略 = MBB-PSO 策略 then
    使用式(9)对粒子位置更新;
```

4.2) 计算粒子 i 的适应值并判断:

```
if 粒子  $i$  适应值未发生改善 then
```

```
     $count_i++$ ;
```

```
    if 粒子  $i$  的  $count_i$  值大于阈值  $K$  then
```

```
        为粒子  $i$  随机分配另一个学习策略;
```

```
         $count_i = 0$ ;
```

```
    end if
```

```
else
```

```
    更新粒子  $i$  的个体历史最优信息;
```

```
     $count_i = 0$ ;
```

```
end if
```

步骤 5 判断本次迭代中,群体中粒子是否全部更新完成:如果是,则执行步骤 6;否则,继续执行步骤 4。

步骤 6 计算群体中的全局最优值。

步骤 7 判断是否已达到算法的迭代终止条件,若满足则输出全局最优值;否则转到步骤 4 继续执行。

从算法流程可知,MHPSO 算法每次迭代时,需对粒子当前使用的学习策略进行判断选择,其基本运算次数是 $O(3N)$;判断粒子适应值是否发生改善的基本运算次数为 $O(2N)$;粒子飞行速度和位置向量更新的基本运算是 $O(2ND)$;计算函数适应值、更新局部和全局最优位置需要的基本运算是 $O(3ND)$;算法一次迭代的时间复杂度是 $O(3N) + O(2ND) + O(3ND) + O(2N) = O(CND)$,因此 MHPSO 算法的复杂度是 $O(CNDT)$ 。其中: N 是群体规模, D 为问题维数, T 是算法最大迭代次数, C 是常数。显然,与标准 PSO 算法相比,MHPSO 算法的复杂度并不高。

3 实验结果分析

3.1 经典基准函数及参数设置

为验证 MHPSO 算法,本文选择 8 个经典函数^[2]作为基准测试函数。 $f_1 \sim f_3$ 为单峰函数, $f_4 \sim f_8$ 为多峰函数。函数的具体信息见表 1。

表 1 8 个经典基准测试函数的基本信息

函数编号	函数名	搜索空间	维数	最优值
f_1	Sphere	$[-100, 100]^D$	30	0
f_2	Schwefel_2_22	$[-10, 10]^D$	30	0
f_3	Schwefel_1_2	$[-100, 100]^D$	30	0
f_4	Schwefel_2_26	$[-500, 500]^D$	30	-418.9829D
f_5	Rastrigin	$[-5.12, 5.12]^D$	30	0
f_6	Ackley	$[-32, 32]^D$	30	0
f_7	Griewank	$[-600, 600]^D$	30	0
f_8	Generalized Penalize_1	$[-50, 50]^D$	30	0

注: D 代表问题求解空间的维数。

仿真实验时,MHPSO 算法种群规模为 20,求解问题维数为 30。每个基准函数独立运行 30 次,函数适应值的最大评估次数为 20 万次。策略池中各学习策略涉及的参数设置分别是:局部扰动策略中的学习因子 $c_1 = c_2 = 1.49618$;高斯子空间学习策略中的子空间维数 M 设置为 5。

3.2 与经典 PSO 改进算法的比较

为比较 MHPSO 算法的优越性,本节与信息全部共享的粒子群优化算法(Fully Informed Particle Swarm, FIPS)^[12]、线性时变加速因子的粒子群算法(self-organizing hierarchical Particle Swarm Optimizer with TimeVarying Acceleration Coefficients, PSO-TVAC)^[13]、动态多子群的粒子群优化算法(Dynamic Multi-Swarm Particle Swarm Optimizer, DMS-

PSO)^[14]、CLPSO^[3]及 APSO^[2]等 5 种经典改进优化算法进行比较。实验时, MHPSO 算法的参数设置同 3.1 节所述, 其他算法的参数设置及运行结果来自于文献[2]。实验对比数据如表 2 所示, 最优结果用下划线标注。表中平均最优适应值 (Mean Best Fitness, MBF) 表示算法的平均最优适应值, 反映了限定评估次数下算法的寻优精度; 标准差 (Standard Deviation, SD) 表示算法的标准方差, 表示算法每次运行结果偏离平均最优值距离的平均数, 反映算法的稳定性和鲁棒性。表 2 的最后一行为 t 检验统计结果, 反映了 MHPSO 算法和其他 5 种改进算法的寻优性能是否存在显著性差异。“w/t/l”表示与其他算法相比, MHPSO 算法寻优性能在 w 个函数上更优, 在 t 个函数上无明显差异, 在 l 个函数上更差。

CLPSO 和 APSO 等优化算法是改进粒子群优化算法中公认的知名算法。表 2 中的对比结果显示, MHPSO 算法与这些改进算法相比, 在各个测试函数上的寻优精度上不仅均有提高, 而且能找到大多数基准函数 (f_1 、 f_2 、 f_3 、 f_5 和 f_7) 的最优值。对于其他多峰函数 f_4 、 f_6 和 f_8 的寻优结果也略胜于其他改进算法。

此外表 2 中最后一行 t 统计结果显示, 与 FIPS 算法和 CLPSO 算法相比, MHPSO 算法在所有基准测试函数获得更优的解精度; 与 HPSO-TVAC 相比, 则有 7 个函数更优; 与 DMS-PSO 算法相比, 结果显示 MHPSO 算法在 6 个函数上更优, 2 个函数寻优性能无明显差异。而与 APSO 算法相比, MHPSO 算法在 5 个函数上更优, 3 个函数性能相等。

显然, 与其他改进算法相比, 无论解的质量还是算法的稳定性, MHPSO 算法都有很大的提高。

表 2 MHPSO 算法与经典改进算法在 30 维基准函数的测试结果

函数	FIPS		HPSO-TVAC		DMS-PSO	
	MBF	SD	MBF	SD	MBF	SD
f_1	3.21E-30	3.60E-30	3.38E-41	8.50E-41	3.85E-54	1.75E-53
f_2	1.32E-17	7.86E-18	6.9E-23	6.89E-23	2.61E-29	6.60E-29
f_3	0.77	0.86	2.89E-7	2.97E-7	47.5	56.4
f_4	-10 113.8	889.58	-10 868.57	289	-9 593.33	441
f_5	29.98	10.92	2.39	3.71	28.1	6.42
f_6	7.69E-15	9.33E-16	2.06E-10	9.45E-10	8.52E-15	1.79E-15
f_7	9.04E-4	2.78E-3	1.07E-2	1.14E-2	1.31E-2	1.73E-2
f_8	1.22E-31	4.85E-32	7.07E-30	4.05E-30	2.05E-32	8.12E-33
w/t/l	8/0/0		7/1/0		6/2/0	
函数	CLPSO		APSO		MHPSO	
	MBF	SD	MBF	SD	MBF	SD
f_1	1.89E-19	1.49E-19	45E-150	5.73E-150	0	0
f_2	1.01E-13	6.51E-14	5.15E-84	1.44E-83	0	0
f_3	395	142	1.0E-10	2.13E-10	0	0
f_4	-12 557.65	36.2	-12 569.5	5.22E-11	-12 569.46	0.61
f_5	2.57E-11	6.64E-11	5.8E-15	1.01E-14	0	0
f_6	2.01E-11	9.22E-13	1.11E-14	3.55E-15	5.89E-16	0
f_7	6.45E-13	2.07E-12	1.67E-2	2.14E-2	0	0
f_8	1.59E-21	1.93E-21	3.76E-31	1.2E-30	1.86E-32	2.96E-32
w/t/l	8/0/0		5/3/0		—	

3.3 算法的收敛速度对比

为考查算法的平均进化趋势, 图 1~4 给出了 2 个单峰函数和 2 个多峰函数 30 维的进化曲线图。其中, X 轴是函数的评估次数, Y 轴是 30 次独立运行中的平均最优适应值的对数值。

对于单峰函数 f_1 , 图 1 显示 MHPSO 算法大概评估 2 万

次达到最优值。图 2 显示 MHPSO 算法平均在 3 万次评估次数内寻找到单峰函数 f_2 最优值。这是因为策略池中的局部扰动策略不仅提高了算法的全局勘探能力, 而且提高了算法的收敛速度, 使得 MHPSO 算法能够迅速找到单峰函数的最优值。

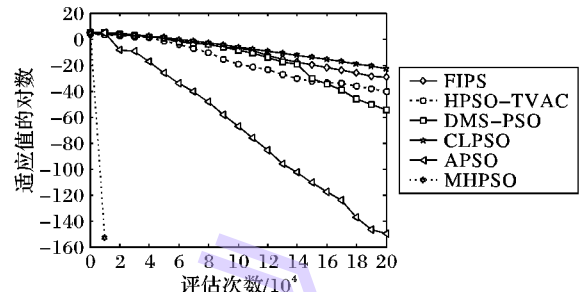


图 1 函数 f_1 的适应值与评估次数的关系

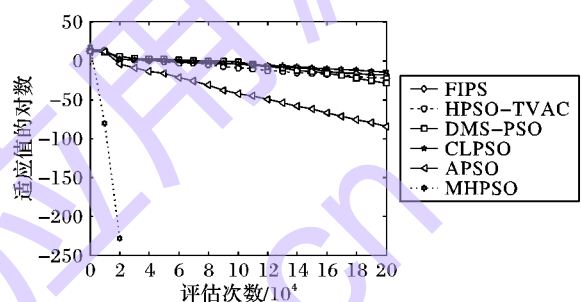


图 2 函数 f_2 的适应值与评估次数的关系

对于多峰函数 f_5 , 图 3 显示 MHPSO 算法不仅找到其最优值, 且收敛速度远远超过其他 5 个改进算法, 平均评估 3 万次达到最优值。对于多峰函数 f_6 , 图 4 显示进化前期新算法的收敛速度遥遥领先, 但到了进化后期接近函数评估 17 万次时, APSO 的收敛速度快速提高, 使得两者的寻优结果相当。这是由于 MHPSO 算法选择高斯子空间学习策略和 MBB-PSO 学习策略进入策略池, 有助于粒子逃离局部最优。

3.4 与其他异构优化算法的比较

为进一步对比算法 MHPSO 的性能表现, 本节选择了近几年提出的异构优化算法 HILPSO^[7]和 SLPSO^[8]作为比较对象。HILPSO 算法和 SLPSO 算法都融合了 4 种具有不同优势的策略, 对不同形态的复杂问题通过自适应学习选择合适的策略完成全局寻优。

本组实验采用文献[7]中的 7 个基准函数进行测试。算法 MHPSO 的实验参数设置同 3.1 节。其他算法的参数设置及运行结果来自文献[7]和文献[8]。表 3 是 3 种算法的实验对比结果, 表中最优的结果用下划线表示。

从表 3 可知, 除 Rosenbrock 函数外, MHPSO 算法在大部分函数的寻优性能均

优于 SLPSO 算法, 而 MHPSO 算法寻优性能与 HILPSO 算法相当。在这三个异构优化算法中, SLPSO 算法和 MHPSO 算法采用了 4 种策略构成策略池, 使用自适应选择机制选择合适的策略, MHPSO 算法则融合了 3 种学习策略并使用随机选择机制选择学

习策略,这说明在异构优化算法中,并不是学习策略越多越好,而是学习策略固有的特点决定了算法的寻优性能。因此,本文提出

的局部扰动学习策略和高斯子空间学习策略决定了 MHPSO 算法的高效寻优性能。

表3 MHPSO 算法与 HPSO 算法、SLPSO 算法在30维基准函数上的测试结果

函数名称	HPSO		SLPSO		MHPSO	
	MBF	SD	MBF	SD	MBF	SD
Sphere	6.03E-307	0	2.78E-50	8.09E-49	0	0
Rosenbrock	0.0246	0.135	2.06	12.3	25.2	0.795
Ackley	7.85E-016	0	3.47E-14	4.45E-14	5.89E-16	0
Griewank	0	0	0.0227	0.144	0	0
Rastrigin	0	1.0	0	0	0	0
Noncontiguous Restraining	0	1.0	0	0	0	0
Weierstrass	0	1.0	4.50E-15	3.55E-14	0	0

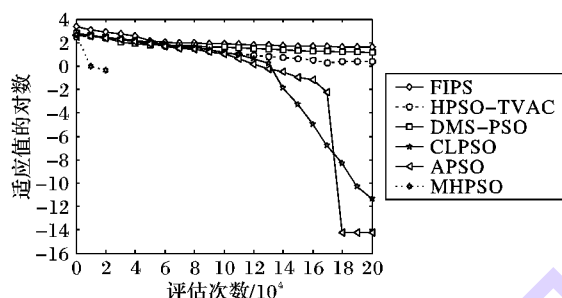


图3 函数 f_5 的适应值与评估次数的关系

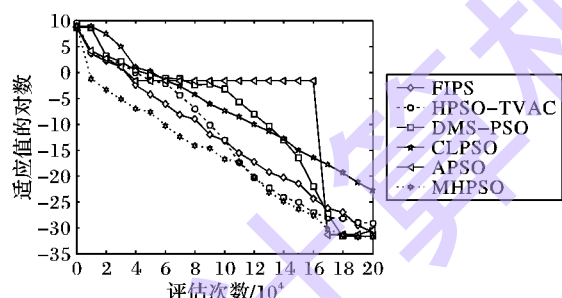


图4 函数 f_8 的适应值与评估次数的关系

4 结语

本文基于异构优化框架,提出了一种多策略并行学习的异构 PSO 算法。该算法首先提出一种局部扰动的学习策略,通过引入扰动因子,对粒子的自身个体最优进行扰动引导粒子勘探更广的搜索区域,有效地保持种群的多样性,增强了算法的全局搜索性能;其次为增强算法逃离局部最优的能力,提出了高斯子空间学习策略,对粒子的位置子空间实施多维高斯学习;并将这两种学习策略与 MBB-PSO 策略共同构建了算法的策略池。最后提出一种简单有效的策略更换机制,指导粒子进化过程中进行策略更换的最佳时机。

表2的仿真实验数据、t检验结果及收敛速度曲线图表明 MHPSO 算法在经典基准测试函数上显示了良好的寻优性能,在求解精度和收敛速度上要优于其他著名的改进优化算法。此外,表3的实验对比数据也验证了 MHPSO 算法在相同的异构优化算法中的优良性能。

今后将继续研究用于各种新的学习策略,放入策略池中,同时还将对粒子学习策略的自适应选择机制进行深入研究。

参考文献:

[1] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995, 4: 1942-1948.

[2] ZHAN Z, ZHANG J, LI Y, *et al.* Adaptive particle swarm optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2009, 39(6): 1362-1381.

[3] LIANG J J, QIN A K, SUGANTHAN P N, *et al.* Comprehensive learning particle swarm optimizer for global optimization of multi-modal function [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.

[4] ENGELBRECHT A P. Heterogeneous particle swarm optimization [C]// Proceedings of the 7th International Conference on Swarm Intelligence. Berlin: Springer-Verlag, 2010: 191-202.

[5] WANG Y, LI B, WEISE T, *et al.* Self-adaptive learning based particle swarm optimization [J]. Information Sciences, 2011, 181(20): 4515-4538.

[6] NEPOMUCENO F V, ENGELBRECHT A P. A self-adaptive heterogeneous PSO for real-parameter optimization [C]// Proceedings of the 2013 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2013: 361-368.

[7] WU D, ZHENG J. Improved parallel particle swarm optimization algorithm with hybrid strategy and self-adaptive learning [J]. Control and Decision, 2013, 28(7): 1087-1093. (伍大清, 郑建国. 基于混合策略自适应学习的并行粒子群优化算法 [J]. 控制与决策, 2013, 28(7): 1087-1093.)

[8] LI C, YANG S, NGUYEN T T. A self-learning particle swarm optimizer for global optimization problems [J]. IEEE Transactions on Systems, Man, and Cybernetics: Part B: Cybernetics, 2012, 42(3): 627-646.

[9] 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.

[10] KENNEDY J. Bare bones particle swarms [C]// Proceedings of the 2003 IEEE Swarm Intelligence Symposium. Piscataway: IEEE, 2003: 80-87.

[11] WANG H, WU Z, RAHNAMAYAN S, *et al.* Multi-strategy ensemble artificial bee colony algorithm [J]. Information Science, 2014 (279): 587-603.

[12] 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.

[13] RATNAWEERA A, HALGAMUGE S, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time varying acceleration coefficients [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.

[14] LIANG J J, SUGANTHAN P N. Dynamic multi-swarm particle swarm optimizer [C]// Proceedings of the 2005 IEEE Swarm Intelligence Symposium. Piscataway: IEEE, 2005: 124-129.