一种基于动态边界的粒子群优化算法

李迎秋[1,2],迟玉红[3],温 涛[1,2]
(1.东北大学软件中心,辽宁沈阳 110004;2.大连东软信息学院计算机系,辽宁大连 116023;3.中国人民解放军 65053 部队,辽宁大连 116113)

摘 要: 2007 年提出的标准粒子群优化算法(PSO-2007)在进化的后期容易出现停滞现象而导致早熟收敛,为此本文提出了一种基于动态边界的粒子群优化算法(DBPSO).该算法根据停滞期粒子运动的特点,将边界动态调整策略引入到 PSO-2007 中,通过跟踪粒子飞行位置的分布动态调整搜索空间的边界,引导粒子在更有效的区域内进行搜索,从而减轻早熟收敛,提高收敛精度.典型测试函数的求解实验结果表明 DBPSO 是可行而有效的.

关键词: 粒子群优化;停滞现象;早熟收敛;动态边界
中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2013)05-0865-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.05.006

A Dynamic Boundary Based Particle Swarm Optimization

LI Ying-qiu[1,2], CHI Yu-hong[3], WEN Tao[1,2]
(1. *Software Center, Northeastern University, Shenyang, Liaoning* 110004, *China*; 2. *Department of Computer Science and Technology, Dalian Neusoft Information Institute, Dalian, Liaoning* 116023, *China*; 3. *Unit* 65053, *PLA, Dalian, Liaoning* 116113, *China*)

Abstract: Standard particle swarm optimization presented in 2007(namely, PSO-2007) inclines towards stagnation phenomena in the later stage of evolution, which leads to premature convergence. Therefore, a PSO based on dynamic boundary(namely, DBPSO) is proposed in this paper. According to the movement characteristics of particles at stagnation stage, DBPSO introduces a strategy of boundary adjusting in PSO-2007. By tracking the distribution of the particles' locations, DBPSO adjusts the boundary of search space dynamically, which could guide the particles to more promising region. This strategy helps PSO-2007 decrease premature convergence and improve convergence precision. The results of experiments of four typical functions show that DBPSO are feasible and effective.

Key words: particle swarm optimization; stagnation phenomena; premature convergence; dynamic boundary

1 引言

PSO 算法是由 Kennedy 和 Eberhart 在鸟类和鱼类等生物群体觅食行为的启发下,于 1995 提出的一种智能优化算法[1].PSO 算法中,每个粒子都被随机地分配一个位置和速度,根据个体经验和群体经验,在优化问题空间内自由飞行,逐渐接近最优解.

PSO 算法具有参数少、实现容易、收敛速度快等特点,已经在求解组合优化、模糊系统控制、神经网络训练等许多领域得到广泛应用[2~5].但是 PSO 算法同其它优化算法一样在快速收敛的同时也容易陷入局部最优,影响收敛精度,因此如何克服早熟收敛,提高收敛精度一直是粒子群优化算法研究领域中的一个热点和难点问题.从近年发表的研究成果看,对 PSO 算法的改进主要包括调整算法参数[6~8]、改进拓扑结构[9~11]、混合其他算法[12,13]等策略.这些改进策略的目的都是使粒子的全局搜索能力和局部搜索能力达到较好的平衡状态,从而提高算法的性能.

事实上,要减轻早熟收敛,有必要对 PSO 算法进化过程中粒子的行为进行观察和分析,然后根据粒子运动的特点提出改进方法.

2 标准粒子群优化算法

PSO 算法中,每个粒子代表了优化问题的一个潜在解,粒子通过自己飞行获得的个体认知以及与群体交流获得的社会认知来不断调整自身的位置,逐渐向最优解靠近.粒子 i 调整自身的飞行速度和位置迭代方程如下:

$$v_{t+1} = \omega v_t + \tilde{c}_1(p_t - x_t) + \tilde{c}_2(g_t - x_t) \quad (1)$$

收稿日期:2012-03-16;修回日期:2012-10-10
基金项目:国家自然科学基金(No.61170168,No.61170169)

$$x_{t+1} = x_t + v_{t+1} \quad (2)$$

其中,t 是当前进化代数,ω 是惯性系数,$\tilde{c}_1$ 和 $\tilde{c}_2$ 是加速系数,x_t和 v_t分别是粒子 i 在 t 时刻的位置和速度,p_t 是粒子 i 所经过的最优位置,即个体极值,代表个体认知,g_t 是粒子 i 的邻居粒子发现的局部极值,代表社会认知.

迄今为止,已经有三个标准粒子群版本,分别于2006、2007 和 2011 年被提出.本文选择 2007 年被提出的标准粒子群(PSO-2007)算法作为参考算法,相关参数如下(更多细节参见文献[14,15]).

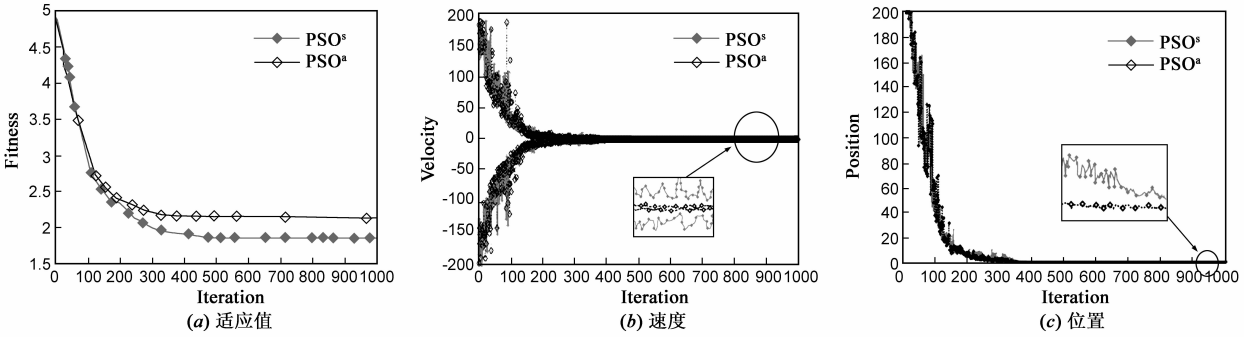
(a) 初始化:粒子在问题空间$[X_{\min}, X_{\max}]^D$ 内均匀分布,其中 D 是维数,采用"half-diff"方法初始化粒子速度;

(b) 惯性系数 $\omega = 1/(2\ln 2) \approx 0.721$;

(c) $\tilde{c}_1$ 和 $\tilde{c}_2$ 是在区间$[0, c]$上服从均匀分布的随机变量,$c = 0.5 + \ln 2 \approx 1.139$.

3 粒子动态行为分析

PSO-2007 算法在进化后期会出现停滞现象[14,16],下面我们通过求解 Rastrigin 函数来观察这种现象.求解过程中,粒子的适应值、速度和位置的进化曲线分别如图 1(a),(b)和(c)所示.



(a) 适应值 (b) 速度 (c) 位置

图1 求解Rastrigin函数的PSO-2007算法进化过程

图 1 中分别将带有同步更新和异步更新的 PSO-2007 称为 PSO^s 和 PSO^a,PSO^s 和 PSO^a 的不同在于极值更新的顺序.在 PSO^s 中,所有粒子在更新完毕它们的个体极值并且和邻居共享各自的适应值后,再更新自身的速度和位置,这样,所有粒子都有它们邻居的完整的信息.而在 PSO^a 中,每个粒子在更新个体极值并将适应值送给它的邻居后,立刻根据到目前为止的局部极值来更新自身的速度和位置,这意味着很多粒子更新依据的是它们邻居的非完全信息.

从图 1 中可以看到,PSO-2007 在进化过程的初期,收敛速度很快,收敛精度也令人满意,但是在进化过程的后期,却出现停滞现象,即适应值没有提高(图 1(a)),速度降低并趋近于零,尽管小幅振荡很频繁(图 1(b)),但是由于粒子失去能量,所以位置变化仍接近于停滞(图 1(c)).

通过对进化过程中粒子动态行为的观察可知,在 PSO-2007 中,粒子在进化后期移动速度变慢,很难逃离局部最优,从而导致算法早熟收敛.目前大多数粒子群优化算法都认为搜索空间是固定的,这意味着,当粒子聚集的时候,绝大部分空间被放弃了,如果最优解就处于聚集的位置,PSO-2007 会得到非常理想的优化结果,但遗憾的是,大多数情况下最优解却可能隐藏在聚集区域附近的某个地方,而发现该有效搜索区域并非易事.通过观察算法进化过程可以发现,粒子在群体与个体最佳经验的指导下,能够不断地进化,向最优解靠近,因此粒子的运动可以为发现该有效搜索区域提供指引,正是基于这一思想,本文提出了基于动态边界的粒子群优化算法,算法通过不断调整搜索空间边界对粒子飞行位置的分布进行跟踪,当发现粒子聚集时,对边界进行一定的扩展,然后在当前搜索空间内激活停滞粒子,同时清除这些粒子的个体认知,促使其从当前局部区域逃离,去寻找最优解.

4 DBPSO 算法

4.1 边界的动态调整

边界的调整策略简单易实现,我们可以把边界所围成的 D 维搜索空间想象成一个 D 维的盒子,当所有粒子都在盒子的当前边界$[B_l, B_r]^D$ 内飞行的时候,盒子就收缩边界,否则就扩展边界.以上动作在当前边界和全局极值点的距离大于阈值 ε 时才执行.若盒子过度收缩,即当前边界和全局极值点的距离小于或等于 ε 时,由第 3 节的分析可知,进化后期粒子往往在最优解的附近聚集,因此这时需要对搜索空间的边界进行重置,以扩大粒子的搜索范围.

在第 d 维,对左边界进行收缩、扩展及重置的方法如公式(3)~(5)所示.

$$B_l(d) = g_t(d) - |B_l(d) - g_t(d)| \times (c_b r_1 + c'_b) \quad (3)$$

$$B_l(d) = g_t(d) - |B_l(d) - g_t(d)| \times (o_b r_2 + 1) \quad (4)$$

$$B_l(d) = g_t(d) - |B_l(d) - X_{\min}| \times r_3 \quad (5)$$

在第 d 维,对右边界进行收缩、扩展及重置的方法如公式(6)~(8)所示.

$$B_r(d) = g_t(d) + |B_r(d) - g_t(d)| \times (c_b r_4 + c'_b) \quad (6)$$

$$B_r(d) = g_t(d) + |B_r(d) - g_t(d)| \times (o_b r_5 + 1) \quad (7)$$

$$B_r(d) = g_t(d) + |X_{\max} - B_r(d)| \times r_6 \quad (8)$$

公式(3)~(8)中,$g_t(d)$为全局极值点,c_b 为收缩率,$c'_b = 1 - c_b$,o_b 为扩展率,$r_1 \sim r_6$ 是[0,1]上服从均匀分布的随机数.公式(3)和公式(6)表明,当所有粒子都处于搜索空间的当前第 d 维边界内时,则在第 d 维对搜索空间边界进行收缩,即以全局极值点为核心对左或右边界进行随机收缩,使其接近全局极值点.公式(4)和公式(7)表明,当有粒子飞跃搜索空间的第 d 维边界时,则在第 d 维对搜索空间进行扩展,即以全局极值点为核心对左或右边界进行随机扩展,使其远离全局极值点;公式(5)和公式(8)表示以全局极值点为核心在$[X_{\min}, X_{\max}]$范围内对左右边界进行重置,从而扩大搜索空间.在算法进化过程中,伴随粒子的运动,通过公式(3)、(4)、(6)、(7)动态调整搜索空间边界,达到对粒子飞行位置分布进行跟踪的目的,当粒子聚集时(主要出现在进化后期),利用公式(5)和(8)扩大搜索空间的范围.

4.2 停滞粒子的处理

由第3节我们知道,算法在进化后期出现停滞现象时,粒子的飞行速度是趋近于0的,在这种情况下,需要提高粒子的多样性,因此,算法通过在搜索空间的第 d 维当前边界内,为速度加上一个服从正态分布的随机数来激活粒子.

当粒子在搜索空间内飞行时,根据两种经验更新位置和速度,即个体经验和社会经验,粒子的个体经验是在进化过程中的个体认知,为在已定义好的空间内搜索更好的可行解提供了搜索方向.但问题是进化过程中,粒子有可能因为受到它们个体经验的吸引过强而不能从所发现的极值中逃离.这样,当边界被重置后,随机挑选部分处于停滞状态的粒子,在当前边界内激活,同时还需要清除它们的个体经验,促使这些粒子飞离当前位置.清除粒子 i 的个体经验的方法如公式(9)所示.

$$p_t(d) = x_t(d) \quad (9)$$

公式(9)表明将粒子 i 的个体极值设置为其当前位置,该处理方式使粒子失去个体经验,从而克服以往经验的影响,有助于粒子逃离局部极值区域.

粒子激活及个体经验清除是当粒子聚集到一定程度时,为了增强种群的多样性,使粒子保持寻优能力而采取的必要措施,粒子激活本质上是对粒子施加的变异操作.为了不对种群的自然进化过程造成过多的干扰,在实施变异操作时,算法采取了一种随机的方式,即生成一个0到1之间的随机数,如果该随机数大于某个阈值,则随机挑选少量粒子施加变异操作,否则不执行任何操作.经过测试该阈值取0.9时,算法可以获得较好的搜索效果.

4.3 算法步骤

DBPSO算法在PSO算法的每一次迭代结束后,增加如下一些步骤.

(1) 标记搜索空间当前边界

首先,在第 d 维,检查是否有一些粒子飞出了当前边界$[B_l(d), B_r(d)]$,如果有粒子飞出左或右边界,则在第 d 维将对应的左边界标志位 $s_l(d)$或者右边界标志位 $s_r(d)$置为 *true*,否则置为 *false*.$[B_l, B_r]^D$ 被初始化为$[X_{\min}, X_{\max}]^D$.

(2) 动态调整搜索空间

D 维搜索空间的动态调整通过调整其每一维的边界实现,具体算法如下.

```
for d = 1:D% for each dimension
  if| B_l(d) - g_t(d) | > ε
    if s_l(d) is flase
      使用公式(3)收缩搜索空间左边界 B_l(d);
    else
      使用公式(4)扩展搜索空间左边界 B_l(d);
    end if
  else
    B_r(d) = true;
    使用公式(5)重置搜索空间左边界 B_l(d);
  end if
  if| B_r(d) - g_t(d) | > ε
    if s_r(d) is false
      使用公式(6)收缩搜索空间右边界 B_r(d);
    else
      使用公式(7)扩展搜索空间右边界 B_r(d);
    end if
  else
    B_r(d) = true;
    使用公式(8)重置搜索空间右边界 B_r(d);
  end if
end for
```

(3)处理停滞粒子

若 $B_r(d)$为 *true*,表明当前边界在第 d 维被重置,则随机挑选少量粒子,在第 d 维当前边界内为速度

加上一个服从正态分布的随机数,以激活停滞的粒子,同时,利用公式(9),清除这些粒子的个体认知.这一步骤有利于减轻顽固的局部最优问题.当达到最大迭代次数或指定目标精度时算法终止.

5 实验分析

5.1 测试函数

这里选择常见的四个 benchmark 函数[17]来评价 DBPSO 的性能,分别是 Sphere, Griewank, Rastrigin 和 Rosenbrock 函数,其中 Sphere 函数和 Rosenbrock 函数是单模函数,后者在极值点附近的曲线较平滑,属于非凸的病态函数,Griewank 和 Rastrigin 函数是多模函数,存在多个极值点,容易陷入局部最优.图 2 显示了在一维空间内,这四个函数的函数值变化情况.为保证一般性,本文采用极值点平移变换[17]对这四个函数在原标准函数的基础上进行了移动,尽管平移处理增加了函数发现全局最优解的难度,但是避免了对称和中心特性,能够确保优化问题的普遍性,因此我们选择这些移动后的 benchmark 函数来验证本文算法的有效性.为了叙述方便,文中分别将平移后的 Sphere, Griewank, Rastrigin 和 Rosenbrock 函数称为 $f1$, $f2$, $f3$ 和 $f4$.

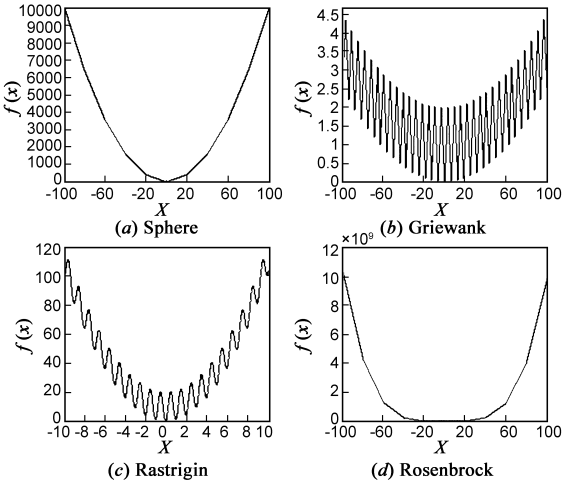


图2 一维空间四个标准函数的函数值变化

5.2 实验计划

本文将带有异步更新和同步更新的 DBPSO 算法分别记为 DBPSO^a 和 DBPSO^s.实验对 PSO^a, DBPSO^a, PSO^s 和 DBPSO^s 的优化性能进行比较.所有测试算法均使用第 2 节所描述的 PSO-2007 的参数.算法独立运行次数为 100 次,每次运行的最大迭代次数取 10000 次,当达到目标精度或者最大迭代次数时,算法运行终止.对于函数 $f1$ ~ $f3$ 实验指定目标精度达到 0,也就是事实上的全局最优,对于函数 $f4$,考虑到它的复杂性,指定目标精度设置为 0.1.

在算法性能对比实验开始前,首先通过实验为 DBPSO 算法获取最佳参数取值.在 DBPSO 算法中,重要的参数包括收缩率 c_b、扩展率 o_b 以及搜索空间边界和全局极值点的距离阈值 ε,为了确定这些参数的取值,本文通过实验,分别对不同参数下,DBPSO 算法针对四个测试函数求解问题的性能进行了比较.收缩率和扩展率所采用的测试参数集如表 1 所示.对于表 1 中每组不同的收缩率和扩展率参数,ε 分别取 1e-2, 1e-3, 1e-4, 1e-5, 1e-6 等 5 个值.通过实验发现,没有一个参数集可以使所有的测试函数都能获得最佳性能,但是相对而言,当 $c_b = 0.03$, $o_b = 0.1$, $\varepsilon = 1e-5$ 时,DBPSO 算法在大多数问题的求解中获得的效果要优于其它参数值,因此,DBPSO 算法采用这组参数值进行对比实验.

表 1 收缩率和扩展率参数集

No.	c_b	o_b
1	0.03	0.05
2	0.03	0.08
3	0.03	0.1
4	0.05	0.08
5	0.05	0.1
6	0.05	0.15
7	0.1	0.12
8	0.1	0.15
9	0.1	0.2

5.3 实验结果

实验中,种群规模为 30,好邻居个数为 3,实验对问题维数在 10、30 和 100 的情况下,四个比较算法的求解结果的质量进行了比较,图 3 是根据各测试算法所得优化结果绘制的箱形图,图中的每一个箱,即 x 轴的每一列对应着 100 个测试结果,箱的长短表示优化结果的分散情况,箱中的小横线是优化结果的中位数.x 轴被分为三个组,这三个组分别是由 1,2,3,4 组成的第一组,由 5,6,7,8 组成的第二组和由 9,10,11,12 组成的第三组,每个组依次由四个比较算法即 PSO^a, DBPSO^a, PSO^s 和 DBPSO^s 组成,例如,X 轴的第二组中,5 是 PSO^a,6 是 DBPSO^a,7 是 PSO^s,8 是 DBPSO^s.X 轴的这三组分别对应于维数为 10,30 和 100 的情况.从图中可以看到,虽然所有测试算法均随着问题维度的增加优化性能降低,但本文所提算法在 $f1$, $f2$, $f3$ 这 3 个函数中均有较稳定的优化性能,且有较好的优化精度,尤其是对两个多模函数 $f2$ 和 $f3$ 的优化效果非常突出.对于 $D = 100$ 的高维问题,本文算法的优势更加明显,但函数 $f4$ 的测试结果有所不同,这是由它本身的特点决定的,因为标准测试函数 $f4$ 有一个平滑狭长的山谷,无法为优化算法提供足够的搜索信息,所以算法常常找不到正确的搜索方向.

通过实验对算法的成功率进行了比较,这里成功

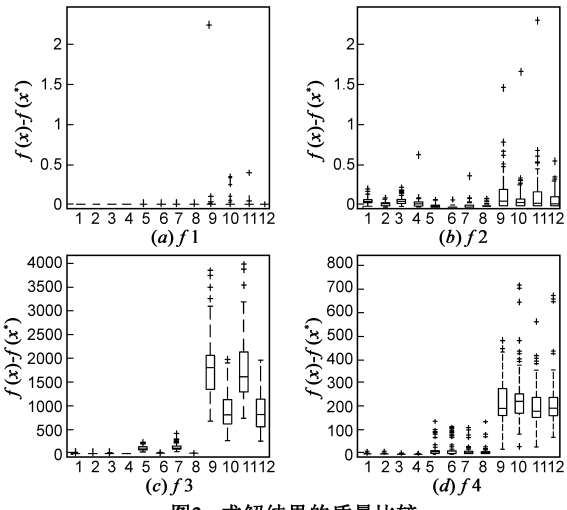


图3　求解结果的质量比较

率是指在 10000 次迭代步骤内成功得到的所有可接受优化解的运行次数的比率.四个比较算法针对测试函数在 10 维和 30 维空间中的平均成功率如表 2 所示.

表 2　成功率比较

测试函数	维数	PSO^a	DBPSO^a	PSO^s	DBPSO^s
$f1$	10	100%	100%	100%	100%
	30	55%	63%	63%	83%
$f2$	10	0	5%	2%	4%
	30	34%	38%	43%	44%
$f3$	10	0	99%	0	100%
	30	0	5%	0	4%
$f4$	10	99%	99%	100%	100%
	30	41%	45%	71%	73%

从表 2 中可以看到,对于单模函数 $f1$,$f4$,当问题维数为 10 维时,所有测试算法的成功率几乎均可达到 100%,没有明显的差别.但是对于复杂的多模优化以及当问题维度增加时,就可看出本文算法的成功率高于 PSO-2007 算法.

表 3 中列出了四个比较算法对每个测试函数在 30 维空间内优化的均值和方差,本文算法在 $f1$,$f2$,$f3$ 这三个函数中,特别是对多模函数的优化,表现出较好的性能.

表 3　均值和方差的比较

测试函数	维数	指标	PSO^a	DBPSO^a	PSO^s	DBPSO^s
$f1$	30	均值	4.29e-29	7.83e-30	3.55e-29	6.82e-30
		方差	7.76e-29	3.52e-29	6.82e-29	2.80e-29
$f2$	30	均值	1.82e-2	1.26e-2	1.43e-2	9.68e-3
		方差	6.15e-2	1.72e-2	1.89e-2	1.29e-2
$f3$	30	均值	1.1e+2	1.09e-1	1.05e+2	9.95e-3
		方差	3.38e+1	7.02e-1	4.24e+1	9.90e-2
$f4$	30	均值	7.500	11.290	7.484	8.793
		方差	1.22e+1	2.30e+1	2.66e+1	2.93e+1

四个比较算法在 30 维空间中针对测试函数的收敛曲线如图 4 所示.从图中可以看到,对于测试函数,DBPSO 算法和 PSO-2007 在进化初期适应值下降速度相当,进化后期 PSO-2007 出现停滞现象,算法收敛于局部最优,而 DBPSO 算法获得了较高的收敛精度.对测试函数在 10 和 100 维空间进行实验,所得结论与在 30 维空间中一致.

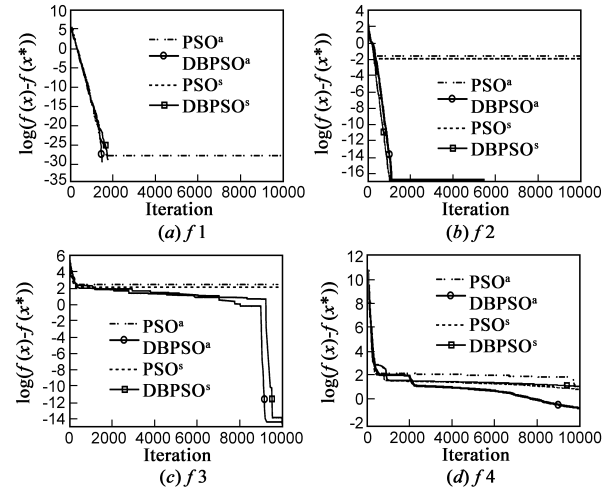


图4　4个测试函数的收敛曲线

上述实验结果表明,本文所提算法较适合于复杂的多模优化问题,DBPSO 算法无论在整体性能,还是稳定性方面都明显优于标准 PSO 算法.

6　结论

基于对粒子动态行为的分析,提出一种基于动态边界的粒子群优化算法 DBPSO,算法采用处理搜索空间边界和个体认知的新方法,通过动态调整搜索空间边界,引导粒子在更有效的区域内进行搜索.实验结果表明 DBPSO 算法可以提供稳定的收敛,从而获得更高的成功率和收敛精度.下一步,我们计划将 DBPSO 算法应用于大型和更复杂的实际多目标优化问题中,并对算法的性能以及平衡计算消耗与搜索精度等问题进行分析和改进.

参考文献

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

[2] 王文彬,孙其博,赵新超,等.基于非均衡变异离散粒子群算法的 QoS 全局最优 Web 服务选择方法[J].电子学报,2010,38(12):2774 – 2779.
Wang Wen-bin, Sun Qi-bo, Zhao Xin-chao, et al. Web services selection approach with QoS global optimal based on discrete particle swarm optimization with non-uniform mutation algorithm[J]. Acta Electronica Sinica, 2010, 38(12): 2774 – 2779.

(in Chinese)
[3] Modares H, Alfi A, Mohammad BB. Parameter estimation of bilinear systems based on an adaptive particle swarm optimization[J]. Engineering Applications of Artificial Intelligence, 2010,23(7):1105 – 1111.
[4] Karakuzu C. Parameter tuning of fuzzy sliding mode controller using particle swarm optimization[J]. Journal of Innovative Computing, Information and Control, 2010, 6(10): 4755 – 4770.
[5] Sabat Samrat L, Ali Layak, Udgata Siba K. Integrated learning particle swarm optimizer for global optimization[J]. Applied Soft Computing,2011,11(1):574 – 584.
[6] Shi Y, Eberchart R. A modified particle swarm optimizer[A]. Proceedings of IEEE Congress on Evolutionary Computation. Anchorage[C]. New York: IEEE Press,1998.69 – 73.
[7] Clerc M, Kennedy J. The particle swarm: Explosion, stability, and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation,2002,6(1):58 – 73.
[8] Chen D, Wang GF, Chen ZY. The inertia weight self-adapting in PSO[A]. Proceedings of the 7th World Congress on Intelligent Control and Automation[C]. New York: IEEE Press, 2008.5313 – 5316.
[9] James Kennedy, Rui Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms[J]. IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews,2006,36(4):515 – 519.
[10] Takeshi Korenaga, Nobuhiko Kondo, Toshiharu Hatanaka, et al. Topology-based personal selection in multi-objective particle swarm optimization[A]. Proceedings of SICE Annual Conf[C]. Tokyo,2008.3465 – 3469.
[11] 倪庆剑,张志政,王蓁蓁,等.一种基于可变多簇结构的动态概率粒子群优化算法[J].软件学报,2009,20(2):339 – 349.
Ni QJ, Zhang ZZ, Wang ZZ, et al. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure[J]. Journal of Software,2009,20(2):339 – 349.(in Chinese)
[12] Shelokar PS, Siarry Partrick, Jayaraman VK, et al. Particle swarm and ant colony algorithms hybridized for improved continuous optimization[J]. Applied Mathematics and Computation,2007,188(1):129 – 142.
[13] Chen MR, Li X, Zhang X, et al. A novel particle swarm optimizer hybridized with extremal optimization[J]. Applied Soft Computing,2010,10(2):367 – 373.
[14] M Clerc. From theory to practice in particle swarm optimization[A]. Handbook of swarm intelligence: concepts, principles and applications[C]. Berlin Heidelgerg: Springer,2011.3 – 36.
[15] M Clerc. Standard Particle Swarm Optimization[EB/OL]. http://clerc.maurice.free.fr/pso/.2011-07-12.
[16] R Poli. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation[J]. IEEE Trans. on Evolutionary Computation,2009,13(4):712 – 721.
[17] Tang K, Yao X, Suganthan P N, et al. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization[R]. Hefei, China: University of Science and Techno logy of China,2007.

作者简介

李迎秋 女,1972 年出生于河北,东北大学软件中心博士研究生,大连东软信息学院计算机系教师,副教授.研究方向为计算智能、服务计算.
E-mail: liyingqiu@neusoft.edu.cn

迟玉红 女,1972 年出生于黑龙江,中国人民解放军 65053 部队,讲师.主要研究方向为计算智能及应用.

温 涛 男,1962 年出生于陕西,东北大学信息科学与工程学院教授,博士生导师.主要研究方向为网络安全、服务计算.