

具有快速收敛和自适应逃逸功能的粒子群优化算法

史小露¹, 孙 辉^{2*}, 李 俊¹, 朱德刚¹

(1. 南昌航空大学 信息工程学院, 南昌 330063; 2. 南昌工程学院 信息工程学院, 南昌 330099)

(* 通信作者电子邮箱 sun_hui2006@yahoo.com.cn)

摘 要: 为了克服标准粒子群优化算法(PSO)后期收敛速度慢、容易陷入局部最优等缺点, 借鉴人工蜂群算法的思想, 提出了一种提高收敛速度并且带有自适应逃逸功能的粒子群优化算法(FAPSO)。算法中每进化一次粒子搜索两次: 一次全局搜索, 一次局部搜索。当粒子陷入局部最优时, 通过逃逸功能使粒子重新搜索。8个经典基准测试函数仿真结果表明, 改进的粒子群优化算法在收敛速度和寻优精度上均有提高, 相对于目前常用的改进粒子群优化算法如CLPSO等, t 检验结果说明, 新算法具有明显的优势。

关键词: 粒子群优化算法; 全局搜索; 局部搜索; 快速收敛; 自适应逃逸

中图分类号: TP301.6 **文献标志码:** A

Particle swarm optimization algorithm with fast convergence and adaptive escape

SHI Xiaolu¹, SUN Hui^{2*}, LI Jun¹, ZHU Degang¹

(1. School of Information Engineering, Nanchang Hangkong University, Nanchang Jiangxi 330063, China;

2. School of Information Engineering, Nanchang Institute of Technology, Nanchang Jiangxi 330099, China)

Abstract: In order to overcome the drawbacks of Particle Swarm Optimization (PSO) that converges slowly at the last stage and easily falls into local minima, this paper proposed a new PSO algorithm with convergence acceleration and adaptive escape (FAPSO) inspired by the Artificial Bee Colony (ABC) algorithm. For each particle, FAPSO conducted two search operations. One was global search and the other was local search. When a particle got stuck, the adaptive escape operator was used to search the particle again. Experiments were conducted on eight classical benchmark functions. The simulation results demonstrate that the proposed approach improves the convergence rate and solution accuracy, when compared with some recently proposed PSO versions, such as CLPSO. Besides, the results of t -test show clear superiority.

Key words: Particle Swarm Optimization (PSO); global search; local search; fast convergence; adaptive escape

0 引言

粒子群优化算法^[1] (Particle Swarm Optimization, PSO) 是由 Kennedy 和 Eberhart 于 1995 年模拟鸟群的运动模型提出的。该算法概念简单、容易实现等突出的优点, 使其在短短几年获得了很大的发展, 并在众多领域得到了成功的应用, 如电力系统优化^[2]、神经网络进化^[3]、数字滤波设计^[4]、半导体器件综合、电路设计及布局优化等方面。PSO 算法提出后, 其存在的收敛速度慢、易于早熟等问题, 亦引起了众多学者的关注。因此, 许多学者对其进行改进研究, 如吴晓军等^[5] 提出均匀搜索粒子群优化算法; 陈伟等^[6] 为提高量子行为粒子群优化算法的全局收敛性能, 提出一种基于完全学习策略的量子行为粒子群优化算法; 刘黎黎等^[7] 利用物理学中的非线性复合效应, 提出一种复合粒子群优化算法。目前国内外研究者对其改进方面主要有: 1) 参数的选择与设计, 在 PSO 算法中存在几个显参数和隐参数, 控制着诸如收敛和发散的重要行为, 通过控制这些参数来优化算法的性能; 2) 种群拓扑结构, 生物组织内的交流及其性能要受其网络结构的影响, 因此有人提出不同的种群拓扑结构, 以改善算法的性能; 3) 混合粒子群优化算法, 将进化计算中的选择、交叉和变异等特性,

混沌、免疫系统的免疫信息处理机制与粒子群优化算法中的寻优机制相结合, 提出相应的混合算法; 4) 离散粒子群优化算法, Kennedy 和 Eberhart 在基本算法的基础上提出一种离散的二进制决策模型, 在二进制空间中微粒的移动是通过翻转位置来实现的, 而微粒的速度用每次迭代的位数来描述, 或等价于某微粒在 t 和 $t+1$ 时刻取值之间的海明距离。

受人工蜂群算法^[8] 的启发, 本文提出一种具有提高收敛速度和自适应逃逸功能的粒子群优化算法(FAPSO)。针对标准 PSO 算法收敛速度慢的缺点, FAPSO 算法增加局部搜索能力, 快速找到最优解; 针对 PSO 算法容易出现“早熟”的现象, FAPSO 算法对标准 PSO 算法增加了“逃逸”功能, 使得粒子能及时跳出局部极值点。通过 Benchmark 函数对算法的性能进行了验证。结果表明: 该算法能增加粒子群的多样性, 提高了算法的收敛速度, 避免早熟收敛, 能够跳出局部最优, 增强了 PSO 算法的全局收敛能力。

1 标准粒子群优化算法

粒子群优化算法是一种演化计算技术, 该算法中将鸟群运动模型中的栖息地类比为所求问题解空间中可能解的位置, 通过个体间的信息传递, 引导整个群体向可能解的方向移

收稿日期: 2012-10-29; 修回日期: 2012-12-13。

基金项目: 国家自然科学基金资助项目(61261039); 江西省自然科学基金资助项目(20122BAB201043)。

作者简介: 史小露(1987-), 女, 山西临汾人, 硕士研究生, 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: 智能计算、Rough 集与粒计算、变分不等原理与变分不等式; 李俊(1987-), 男, 江西赣州人, 硕士研究生, 主要研究方向: 群智能优化算法; 朱德刚(1988-), 男, 安徽芜湖人, 硕士研究生, 主要研究方向: 群智能优化算法。

动,在求解过程中逐步发现较好解的可能性。群体中的鸟被抽象为没有质量和体积的“粒子”,通过这些“粒子”相互协作和信息共享,使其运动速度受到自身和群体的历史运动状态信息的影响,以自身(p_{best})和群体(g_{best})的历史最优位置来对粒子当前的运动方向和速度加以影响,较好地协调粒子本身和群体运动间的关系,每个粒子的适应值取决于所选函数的优化值。粒子群优化算法的数学描述如下:

设粒子群规模为 n ,解空间为 D , $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ($i = 1, 2, \dots, n$) 表示第 i 个粒子的位置,每个 x_i 对应一个目标函数值 $f(x_i)$, $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ ($i = 1, 2, \dots, n$) 表示第 i 个粒子的速度,代表每次迭代中粒子移动的距离, $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ ($i = 1, 2, \dots, n$) 表示第 i 个粒子的历史最优位置, $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ ($i = 1, 2, \dots, n$) 表示全局最优位置。飞行速度和位置按下式计算:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

其中: $i = 1, 2, \dots, n$; $d = 1, 2, \dots, n$; c_1, c_2 为学习因子,一般取 $c_1 = c_2 = 2$; r_1, r_2 为 $0 \sim 1$ 均匀分布的随机数。

2 FAPSO 算法

目前关于 PSO 算法的改进很多,并且在众多领域内得到广泛的应用。然而,在算法的收敛速度和早熟收敛两方面很难得到平衡,这方面仍然是一个值得研究的问题。针对粒子群优化算法的收敛速度和早熟收敛的问题,本文提出 FAPSO 算法,从两方面进行改进:一方面增加局部搜索能力,以提高收敛速度;另一方面,增加逃逸功能,以便有效地逃离局部最优。

2.1 局部搜索的改进

在标准 PSO 算法中,较好粒子与较差粒子的搜索机会一样,导致较好的粒子不能充分在其周围寻找有利信息。为了能够快速找到有用信息,在较好粒子周围做精细的搜索,即在较优粒子周围做更多的搜索,可能有效增加算法的局部搜索能力,使得算法能快速找到全局最优解,提高收敛速度。FAPSO 算法中,每进化一次,粒子群搜索两次,一次全局搜索,一次局部搜索。首先设定初始种群个数为 n ,所有个体先按标准 PSO 算法进行一次全局搜索,记录全局最优值 g_{best} 。全局搜索完成后,对粒子进行评估如算法 1,即按锦标赛原则^[9]决定粒子局部搜索的概率。

算法 1 计算粒子局部搜索的概率。

```

 $a_i = 0$ 
for  $i = 1:n$ 
    for  $j = 1:n$ 
        if  $f_i \leq f_j$ 
             $a_i = a_i + 1$ ;
        end
    end
end
 $p_i = a_i / \sum_{i=1}^n a_i$ 

```

算法 1 中, f_i 为第 i 个粒子的适应值; a_i 为适应值收益度^[8], a_i 越大该粒子的位置就越优越; p_i 为第 i 个粒子被选中局部搜索的概率。

被选中的粒子按式(1)更新速度,按式(2)更新位置,算法 2 为粒子进行局部搜索的选择与更新过程。由算法 2 可知,拥有较优信息的粒子被选中的概率更大,进行局部搜索的

次数更多。

算法 2 局部搜索的选择与更新过程。

```

 $i = 0$ ;
 $t = 0$ ;
while ( $t < n$ ) do
     $r_i = \text{rand}[-1, 1]$ ;
    if ( $p_i > r_i$ )
         $t++$ ;
         $v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})$ ;
         $x_{id} = x_{id} + v_{id}$ ;
    end if
     $i++$ ;
    if ( $i = n - 1$ )
         $i = 0$ ;
    end while

```

2.2 自适应逃逸过程

在标准 PSO 算法中,随着进化迭代次数的增加,粒子逐渐向全局最优值聚集,群体的多样性减小,粒子的速度也不断减小。一旦粒子落入某个极值范围,则种群的粒子将很难重新移动,这时算法收敛到目前找到的最优解。若此时搜索到的极值 g_{best} 是优化问题的一个局部极值,那么算法就出现了早熟收敛现象。标准粒子群优化算法没有提供逃离局部解的策略,即标准粒子群的探索能力较弱。为了提高标准粒子群优化算法的探索能力,使粒子能及时跳出局部最优, FAPSO 算法对标准粒子群优化算法进行改进:

局部搜索后记录更新后的当前全局最优值 g_{best} ,并记录 g_{best} 在进化过程中连续未改进的次数 c ,若 $c > \text{limit}$,即陷入局部最优。此时,需要使粒子跳出局部最优重新搜索。

目前已有的一些算法利用变异操作使粒子逃离局部最优,但不能利用先前的经验做精细搜索。为了充分利用历史知识,同时增加种群多样性,使算法的搜索空间尽可能地扩大,从而以更大概率收敛到全局最优解,将群体按粒子编号分为规模相等的三个分群,分群 A、分群 B、分群 C。分群 A 中各粒子位置随机重置跳出局部最优,扩大搜索范围;分群 B 中各粒子随机选择某一维按式(3)向分群 A 各粒子学习,使得分群 B 中的粒子既保留有先前的经验又携带新的信息;分群 C 仍保留原来的位置,以充分利用之前的经验。

$$v_{ij} = x_{ij} + \varphi_{ij} (x_{ij} - x_{kj}) \quad (3)$$

其中: k 为分群 A 中的粒子; i 为分群 B 中的粒子; j 为随机数且 $j \in \{1, 2, \dots, D\}$ 。

2.3 FAPSO 算法的主要过程

根据上述思路,新算法的具体步骤如下:

- 1) 设置主要参数,如群体个数 n 、最大循环次数、阈值、学习因子和惯性权重;
- 2) 群体中各粒子按标准 PSO 进行一次全局搜索;
- 3) 对各粒子按算法 1 进行评估,按概率选择粒子进行局部搜索,并记录局部搜索后的全局最优值 g_{best} ;
- 4) 若全局最优值 g_{best} 在连续 limit 代没改进,进入 5),否则返回 2);
- 5) 将群体分为规模相等的三个分群,分群 A 中粒子的位置随机重置,分群 B 中的粒子随机选择一维向分群 A 各粒子学习,分群 C 中粒子仍保留在原来的空间搜索;
- 6) 记录局部搜索后最优值 g_{best} ;
- 7) 若满足终止条件则输出最优解,否则返回 2)。

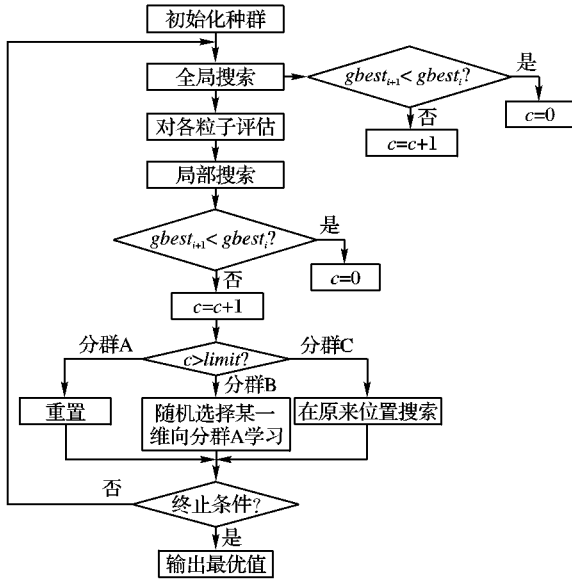


图1 进化过程曲线

3 仿真实验及结果分析

3.1 参数设置及基准测试函数

为了验证 FAPSO 算法的性能,本文选取标准 PSO 及改进粒子群优化算法进行对比实验,引入 8 个经典函数^[10-11]作为基准测试函数测试。实验中种群规模为 15。在 FAPSO 算法和 PSO 算法中参数选取均为: $limit = 200, c_1 = (2.3 - 1.6)(iter/maxCycle)^2 + (1.6 - 2.3)(2iter/maxCycle) + 2.3$; $c_2 = 4 - c_1$, 式中 $iter$ 为迭代次数, $maxCycle$ 为最大迭代次数。随着迭代次数的增加 c_1 递减, c_2 递增,使得算法在初期充分利用自身经验,在后期则增加社会经验。 w 随着迭代次数的增加从 0.8 线性递减到 0.2^[12],这样能够使算法在开始时,粒子以较大速度步长在全局范围内搜索,后期在极点周围做精细搜索。

下面给出这 8 个函数的定义、取值范围和理论最优值。其中: $f_1 \sim f_4$ 是单模态函数,在定义域内只有一个极值点,主要考察算法的寻优精度和执行性能; $f_5 \sim f_8$ 是多模态函数,在定义域内有许多局部极值点,主要检测算法的全局搜索性能和避免早熟的能力。其中: $Quadratic$ 是连续的凸面函数,理论最优值位于一条平滑狭长的波谷内,大多数算法在该函数均很难找到理论最优值; $Rosenbrock$ 的理论最优值位于一个弯曲的、平滑路径上的谷底,由于该函数较特殊,通常用来评价优化算法的性能^[13]; 函数 $Schwefel$ 理论最优值为 $-418.9829D$, 其余函数的理论最优值均为 0。

1) Sphere 函数 (取值范围 $[-100, 100]^D$, 理论最优值 0):

$$f_1(x) = \sum_{i=1}^D x_i^2$$

2) Schwefel's 函数 (取值范围 $[-10, 10]^D$, 理论最优值 0):

$$f_2(x) = \sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i|$$

3) Quadratic 函数 (取值范围 $[-100, 100]^D$, 理论最优值 0):

$$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$$

4) Rosenbrock 函数 (取值范围 $[-10, 10]^D$, 理论最优值 0):

$$f_4(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2]$$

5) Schwefel 函数 (取值范围 $[-500, 50]^D$, 理论最优值 $-418.9829D$):

$$f_5(x) = \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|})$$

6) Noncontinuous Rastrigin 函数 (取值范围 $[-5.12, 5.12]^D$, 理论最优值 0):

$$f_6(x) = \sum_{i=1}^D [y_i^2 - 10 \cos 2\pi y_i + 10];$$

$$y_i = \begin{cases} x_i, & |x_i| < 0.5 \\ \text{round}(2x_i)/2, & |x_i| \geq 0.5 \end{cases}$$

7) Griewank 函数 (取值范围 $[-600, 600]^D$, 理论最优值 0):

$$f_7(x) = \frac{1}{4000} \sum_{i=1}^D (x_i)^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$$

8) Generalized Penalized 函数 (取值范围 $[-50, 50]^D$, 理论最优值 0):

$$f_8(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4);$$

$$y_i = 1 + \frac{1}{4} (x_i + 1)$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

3.2 仿真实验结果

表 1 给出了 8 个基准函数在 30 维的仿真实验结果,为了进一步比较 FAPSO 算法的优越性,表 1 给出了 GPSO^[14]、LPSO^[15]、VPSO^[15]、FIPS^[16]、CLPSO^[17] 和标准 PSO 算法的结果进行对比,表中平均最优适应值 (Mean Best Fitness, MBF) 反映了限定的评估次数下算法的寻优精度,标准差 (Standard Deviation, SD) 反映了算法的稳定性和鲁棒性。为消除随机影响,每个函数独立运行 30 次,以平均值作为最终结果。每次运行评估 20 万次。

由表 1 可以看出,无论对是解的质量,还是算法的稳定性, FAPSO 算法都有很大的提高。与标准 PSO 相比在 8 个测试函数上寻优性能和稳定性上均有提高。CLPSO 是改进粒子群优化算法中公认的知名算法,目前在各类国际期刊上经常被引用。FAPSO 与 CLPSO 算法相比,在 f_1, f_2, f_3, f_7, f_8 5 个函数上寻优精度均有提升,与其他几种改进算法相比在各个测试函数上寻优精度均有提高。特别地,对 f_3 函数大多数算法都很难找到最优解, FAPSO 在该函数上能取得较好的优化结果。对 f_4 函数 FAPSO 寻优结果稍逊于 CLPSO,但稳定性优于 CLPSO。从表 1 可以看出对一些单峰函数,标准 PSO 也可以得到理论最优值,但整体来看标准 PSO 的鲁棒性较差。

表 2 为 t 检验结果,分位数为单侧 0.05,自由度为 30,查表得对应的临界值为 1.697,即当 t 值大于这个值时,两种算法存在显著性差异。表 3 中“+”表示 t 大于这个临界值,即本文算法明显优于其他算法;“=”表示两种算法无明显差异;“-”表示本文算法差于其他算法;“ $w/t/l$ ”表示 FAPSO 与所选取函数相比有 w 个函数优于该算法, t 个函数上无明显差异, l 个函数上差于该算法^[18]。

由表 2 的 t 检验结果可以看出,与其他几种算法相比, FAPSO 算法有明显优势。

表1 30维函数测试结果

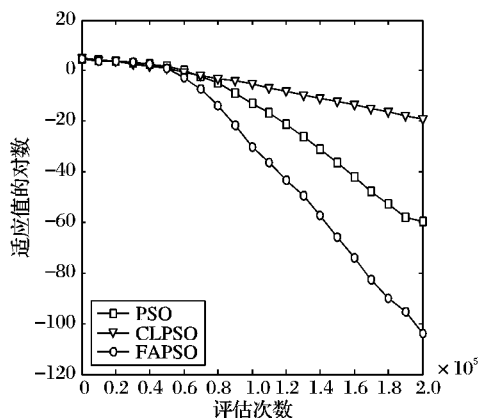
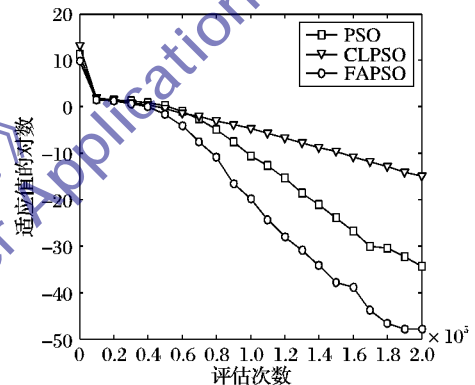
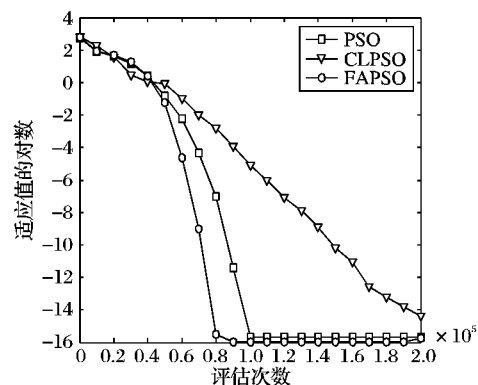
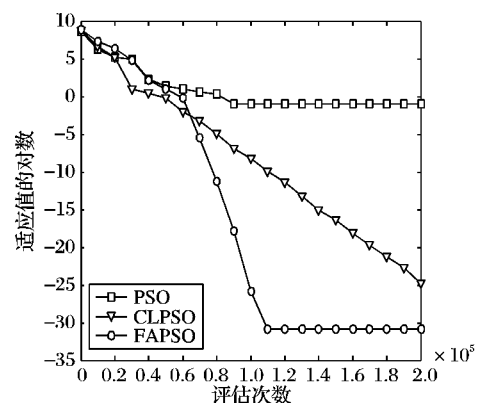
函数	性能指标	GPSO	LPSO	VPSO	FIPS	PSO	CLPSO	FAPSO
f_1	MBF	$1.98e-53$	$4.77e-29$	$5.11e-38$	$3.21e-30$	$1.65e-60$	$1.89e-19$	$1.7e-104$
	SD	$7.08e-53$	$1.13e-28$	$1.91e-37$	$3.60e-30$	$6.56e-64$	$1.49e-19$	$1.4e-101$
f_2	MBF	$2.51e-34$	$2.03e-20$	$6.29e-27$	$1.13e-17$	$2.25e-35$	$1.01e-13$	$1.12e-48$
	SD	$5.84e-34$	$2.89e-20$	$8.68e-27$	$7.86e-18$	$2.91e-35$	$6.51e-14$	$2.07e-36$
f_3	MBF	$6.45e-02$	18.60	1.44	0.77	$7.64e-03$	395	$6.9e-7$
	SD	$9.46e-02$	30.71	1.55	0.86	$9.4e-03$	142	$1.03e-06$
f_4	MBF	28.1	21.86	37.65	22.54	27.5	11	12.7
	SD	24.6	11.16	24.94	0.31	28.8	14.5	13.2
f_5	MBF	-10090.16	-9628.35	-9845.27	-10868.57	-1.0100.0	-12557.66	-11535.1
	SD	495	456.54	588.87	289	370.1	36.2	309.58
f_6	MBF	15.55	30.40	21.33	35.91	26.8	0.167	0.52
	SD	7.4	9.23	9.46	9.49	7.43	0.379	0.79
f_7	MBF	$2.37e-02$	$1.10e-02$	$1.31e-02$	$9.04e-04$	$2.22e-16$	$6.45e-13$	$1.2e-16$
	SD	$2.57e-02$	$1.60e-02$	$1.35e-02$	$2.78e-03$	$1.97e-17$	$2.07e-12$	$8.92e-17$
f_8	MBF	$1.04e-2$	$2.18e-30$	$3.46e-3$	$1.22e-31$	$6.9e-3$	$1.59e-21$	$1.01e-31$
	SD	$3.16e-02$	$5.14e-30$	$1.89e-02$	$4.85e-32$	0.025	$1.93e-21$	$8.28e-31$

表2 t 检验测试结果

函数	GPSO	LPSO	VPSO	FIPS	PSO	CLPSO
f_1	=	+	=	+	+	+
f_2	+	+	+	+	+	+
f_3	+	+	+	+	+	+
f_4	+	+	+	+	+	-
f_5	+	+	+	+	+	-
f_6	+	+	+	+	+	-
f_7	+	+	+	=	+	+
f_8	+	+	-	-	=	+
$w/t/l$	7/1/0	8/0/0	6/1/1	6/1/1	7/1/0	5/0/3

3.3 收敛速度分析

图2~5给出了两个单峰函数和两个多峰函数30维的进化过程曲线。由图可知,本文提出的FAPSO算法,由于增加了局部搜索的能力,增强了算法的开发能力,在处理单峰函数时,初期与PSO和CLPSO有相同的收敛速度,当评估次数达到20000后,FAPSO进化曲线几乎直线下降,有较快的收敛速度。在处理多峰函数时,初期能快速跳出局部最优,当评估次数达到40000后,收敛速度明显快于PSO和CLPSO。由图5知,对于 f_8 函数,由于函数自身的特性算法很容易陷入局部最优,而导致收敛速度变慢,甚至出现停滞现象,CLPSO在该函数上评估次数达到10000时,陷入局部最优,出现停滞现象。而FAPSO能跳出这些局部极值点,快速地找到最优解。

图2 f_1 函数进化过程曲线图3 f_2 函数进化过程曲线图4 f_7 函数进化过程曲线图5 f_8 函数进化过程曲线

4 结语

本文针对粒子群优化算法早熟收敛和收敛速度慢等问题,受人工蜂群算法的启发,对标准粒子群优化算法增加局部搜索能力,并能在粒子陷入局部最优时及时跳出极值点,从而既克服了早熟现象,同时提高了算法的收敛速度。对8个函数的寻优结果表明,FAPSO在单峰函数和多峰函数上均表现出良好的性能,与标准PSO及其他改进的粒子群优化算法相比有了较大的改善。 t 检验结果也表明该算法有明显优势。此外,FAPSO算法还有比较大的改进空间,如算法中的分群规则、变异方式及学习策略等;同时还可在此基础上作进一步的改进,如增加某些新改进算法的内容等。因此,FAPSO算法还有较大的研究空间,这也是该算法今后研究的内容。

参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995, 4: 1942-1948.
- [2] 刘丽军, 蔡金铤. 基于自适应强引导粒子群算法的电力系统无功优化[J]. 电工电能新技术, 2012, 31(4): 24-28.
- [3] 王勇. 基于混沌粒子群的神经网络主汽温控制系统[J]. 计算机仿真, 2012, 29(8): 322-325.
- [4] 李鹏, 马红梅, 张旭珍. 基于粒子群优化算法的模拟滤波器设计[J]. 计算机工程, 2011, 37(13): 246-250.
- [5] 吴晓军, 杨战中, 赵明. 均匀粒子群搜索算法[J]. 电子学报, 2011, 39(6): 1261-1266.
- [6] 陈伟, 周嶙, 孙俊, 等. 一种采用完全学习策略的两子行为粒子群优化算法[J]. 控制与决策, 2012, 27(5): 719-730.
- [7] 刘黎黎, 李国家, 汪定伟. 动态环境下带有非线性效益的复合粒子群优化算法[J]. 控制理论与应用, 2012, 29(10): 1253-1262.
- [8] KARABOĞA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3): 459-471.
- [9] BAO L, ZENG J C. Comparison and analysis of the selection in artificial bee colony algorithm [C]// Proceeding of the 9th International Conference on Hybrid Intelligent Systems. Washington, DC: IEEE

Computer Society, 2009: 411-416.

- [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 Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [11] SUGANTHAN P N, HANSEN N, LIANG J J, *et al.* Problem definitions and evaluation criteria for the CEC2005 special session on realparameter optimization, KanGAL Report Number 2005005[R/OL]. [2012-10-01]. <http://bschw.googlecode.com/svn-history/r599/trunk/EvalCompu/Tech-Report-May-30-05.pdf>.
- [12] JIAO B, LIAN Z G, GU X S. A dynamic inertia weight particle swarm optimization algorithm[J]. Chaos Solitons & Fractals, 2008, 37(3): 698-705.
- [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] SHI Y, EBERHART R C. A modified particle swarm optimizer [C]// The 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence. Piscataway: IEEE Press, 1998: 69-73.
- [15] KENNEDY J, MENDES R. Population structure and particle swarm performance [C] // Proceeding of the 2002 Congress on Evolutionary Computation. Washington, DC: IEEE Computer Society, 2002: 1671-1676.
- [16] 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.
- [17] LIANG J J, QIN A K, SUGANTHAN P N, *et al.* Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [18] GONG W Y, CAI Z H, CHARLES X, *et al.* Enhanced differential evolution with adaptive strategies for numerical optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, 41(2): 397-413.

(上接第1297页)

参考文献:

- [1] 蔡文沁, 彭培林, 姜寿山. 航空产品设计知识的表示与重用技术研究[J]. 计算机集成制造系统, 2004, 10(1): 55-58.
- [2] 殷国富, 干静, 胡晓兵, 等. 面向信息时代的机械产品现代设计理论与方法研究进展[J]. 四川大学学报: 工程科学版, 2006, 38(5): 38-47.
- [3] WU M-C, LO Y-F, HSU S-H. A fuzzy CBR technique for generating product ideas[J]. Expert Systems with Applications, 2008, 34(1): 530-540.
- [4] DURAN O, AGUILO J. Computer-aided machine-tool selection based on a fuzzy-AHP approach[J]. Expert Systems with Applications, 2008, 34(3): 1787-1794.
- [5] ROSS S, FANG L P, HIPEL K W. A case-based reasoning system for conflict resolution: design and implementation[J]. Engineering Applications of Artificial Intelligence, 2002, 15(3/4): 369-383.
- [6] 沈斌, 官大. 产品设计知识重用研究[J]. 计算机工程, 2006, 32(18): 186-210.
- [7] HUNTER R, VIZAN A, PEREZ J, *et al.* Knowledge model as an integral way to reuse the knowledge for fixture design process[J]. Journal of Materials Processing Technology, 2005, 164(15): 1510-

1518.

- [8] COSTA C A. Product range models supporting design knowledge reuse[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2001, 215(3): 323-337.
- [9] 罗仕鉴, 朱上上, 孙守迁, 等. 基于集成化知识的产品概念设计技术研究[J]. 计算机辅助设计与图形学学报, 2004, 16(3): 261-266.
- [10] SUH N P. Axiomatic design advances and applications[M]. New York: Oxford University Press, 2000.
- [11] MELVIN J W, SUH N P. Simulation within the axiomatic design framework[J]. CIRP Annals - Manufacturing Technology, 2002, 51(1): 107-110.
- [12] 唐敦兵, 钱晓明, 王晓勇, 等. 基于公理化设计矩阵与设计结构矩阵同步演化的产品设计[J]. 计算机辅助设计与图形学学报, 2007, 19(4): 541-544.
- [13] HEO G Y, LEE S K. Design evaluation of emergency core cooling systems using axiomatic design[J]. Nuclear Engineering and Design, 2007, 237(1): 38-46.
- [14] 宋慧军, 林志航. 基于域结构模板的机械产品概念设计方案生成[J]. 机械工程学报, 2001, 37(9): 24-29.
- [15] 于学军, 罗振璧, 朱立强. 公理设计中确定交互作用程度的方法[J]. 机械工程学报, 2007, 43(4): 39-43.