

带扰动因子的自适应粒子群优化算法

赵志刚¹ 张振文¹ 石辉磊²

(广西大学计算机与电子信息学院 南宁 530004)¹ (91630 部队 广州 510000)²

摘要 针对标准粒子群优化算法搜索精度不高、易陷入局部最优的问题,提出了一种带扰动因子的自适应粒子群优化算法。该算法进行混沌初始化,采用自适应的惯性权重,并将扰动因子加入粒子个体极值、全局极值和位置更新公式中。通过与其它算法的数值实验对比,新算法能够有效避免局部最优,全局收敛性能显著提高,收敛速度更快。

关键词 粒子群优化算法,混沌初始化,惯性权重,扰动因子

中图分类号 TP18 **文献标识码** A

Adaptive Particle Swarm Optimization Algorithm with Disturbance Factors

ZHAO Zhi-gang¹ ZHANG Zhen-wen¹ SHI Hui-lei²

(College of Computer and Electronics Information, Guangxi University, Nanning 530004, China)¹

(91630 Army, Guangzhou 510000, China)²

Abstract A new particle swarm optimization (PSO) algorithm was presented to overcome disadvantages that standard PSO has shown in solving complex functions, including slow convergence rates, low precisions and premature convergence, etc. The proposed algorithm improves the performances of standard PSO by following methods: a) applying chaotic initialization for swarm, b) using adaptive inertia weight to enhance the balance of global and local search of algorithm, and c) introducing disturbance factors to avoid being trapped in local optimum. The experimental results show that the new algorithm has great advantages of convergence property than standard PSO and some other modified algorithms.

Keywords PSO, Chaotic initialization, Inertia weight, Disturbance factors

1 引言

粒子群优化(Particle Swarm Optimization, PSO)算法由 James Kennedy 和 Russell Eberhart 于 1995 年共同提出,是一种基于群体智能的随机优化算法^[1]。PSO 算法结构简单,需要控制和调整的参数较少,收敛速度较快,算法容易实现,自提出以来就受到广泛关注,已陆续成功应用到许多领域。比如,文献[2]基于 PSO 算法诊断机械故障,有效地剔除了冗余信息,缩短了网络训练的时间,提高了故障诊断的准确度;文献[3]基于 PSO 算法处理图像配准,提高了图像配准的速度和精度;文献[4]将 PSO 算法应用于双层油船,以有效优化其复杂的结构设计问题;文献[5]基于 PSO 算法有效地控制锅炉过热现象。

很多研究者从不同方面对 PSO 算法进行了改进。文献[6]将混沌理论应用到 PSO 算法中,能够有效避免算法的早熟收敛问题,具有较强的全局搜索能力。文献[7]提出了带极值扰动的粒子群优化算法,可以避免算法陷入局部极值点。文献[8]提出了简化的自适应粒子群优化算法,该算法采用去掉速度因子的算法结构,并对全局极值进行选择变异,能有效

避免局部最优,提高算法收敛性能。

2 标准粒子群优化算法

PSO 算法首先随机初始化一个种群,种群中有很多的“粒子”,每个粒子都有两个参数:速度和位置。算法采用一个适应度函数来评价粒子当前位置的优劣,通过不断迭代,最终找到最优解或近似最优解。在每次迭代中,粒子通过跟踪个体极值和全局极值来更新自己的速度和位置。

Kennedy 和 Eberhart 最早提出的 PSO 算法的速度与位置更新公式如下^[1]:

$$v(t+1) = v(t) + c_1 r_1 (p_{best} - x(t)) + c_2 r_2 (g_{best} - x(t)) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

式中, t 为当前迭代次数; c_1 和 c_2 为学习因子,一般取值均为 2; r_1 、 r_2 是分布在 $[0, 1]$ 内的随机数; p_{best} 为粒子本身找到的最优解,即个体极值; g_{best} 为整个粒子群当前找到的最优解,即全局极值。

Shi 等提出通过添加惯性系数 ω 来增强粒子跳出局部最优的能力^[9],式(1)变为:

$$v(t+1) = \omega v(t) + c_1 r_1 (p_{best} - x(t)) + c_2 r_2 (g_{best} - x(t)) \quad (3)$$

到稿日期:2013-05-21 返修日期:2013-07-11 本文受国家自然科学基金项目(61063031),广西教育厅科研项目(201106LX035)资助。

赵志刚(1973—),男,博士,副教授,主要研究方向为智能优化算法, E-mail: zzgmail2002@163.com; 张振文(1987—),女,硕士生,主要研究方向为粒子群优化算法; 石辉磊(1987—),男,助理工程师。

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * t / T_{\max} \quad (4)$$

式中, $\omega_{\max}$ 、 $\omega_{\min}$ 分别是最大惯性权重和最小惯性权重; t 、 $T_{\max}$ 为当前迭代次数和最大迭代次数。随着迭代的进行, ω 线性减小, 这使得算法在迭代初期具有较强的探索能力, 迭代后期具有较强的发掘能力。

3 改进的粒子群优化算法

3.1 混沌初始化

混沌(chaos)是一种貌似无规则的运动, 在确定性非线性系统中, 不需附加任何随机因子就可以产生随机的行为。混沌系统具有特殊的运动规律, 主要表现为随机性、规律性和遍历性。本文将混沌引入到 PSO 算法中进行随机初始化。采用 Logistic 方程 $y_{k+1} = s * y_k (1 - y_k)$, $k=0, 1, 2, \dots, N-1$, 得到 N 个初始粒子, 从 N 个粒子中选较好的 n (n 为设定的粒子群规模, $n \leq N$) 个作为初始位置项, 同理可以得到 n 个初始速度项。在 Logistic 方程中, s 为控制参数, 当 $s=4$, $0 \leq y_0 \leq 1$ 时, 方程处于完全混沌状态。

混沌初始化使得取值范围分布更广泛, 能够为粒子的后续迭代保存有用信息, 增大发现最优解的概率。

3.2 扰动因子

为增强粒子跳出局部最优的能力, 受文献[7]的启发, 本文采用停滞步数作为触发条件, 对个体极值 p_{best} 、全局极值 g_{best} 和位置更新公式加入扰动因子, 扰动方式如下:

$$p_{best} = (\alpha^{>T}) * p_{best}$$

$$g_{best} = (\beta^{>Tg}) * g_{best}$$

$$x(t+1) = x(t) + (\epsilon^{>To}) * v(t+1)$$

式中, t 、 tg 分别表示个体极值和全局极值迭代停滞的步数, T 、 Tg 表示个体极值和全局极值产生扰动对应的停滞步数的阈值; to 表示全局极值执行扰动操作的次数, 它的值受限于 tg 和 Tg 的值, 当全局极值扰动达到一定次数, 说明扰动效果不佳, 就对位置项添加扰动, To 表示位置项需要添加扰动所对应的阈值。另外,

$$\alpha^{>T} = \begin{cases} 1, & t \leq T \\ rand(1), & t > T \end{cases}$$

$$\beta^{>Tg} = \begin{cases} 1, & tg \leq Tg \\ rand(1), & tg > Tg \end{cases}$$

$$\epsilon^{>To} = \begin{cases} 1, & to \leq To \\ rand(1), & to > To \end{cases}$$

表示带条件的随机函数。

对连续函数来说, 在粒子尚未达到最优点时, 在它附近存在比它更优的点。本文通过扰动因子调整个体极值、全局极值和位置项, 以增大发现更好解的概率。

式(2)、式(3)添加扰动因子后变为:

$$x(t+1) = x(t) + \epsilon^{>To} * v(t+1) \quad (5)$$

$$v(t+1) = \omega v(t) + c_1 r_1 (\alpha^{>T} * p_{best} - x(t)) + c_2 r_2 (\beta^{>Tg} * g_{best} - x(t)) \quad (6)$$

3.3 自适应惯性权重

惯性权重 ω 是 PSO 算法中的一个重要参数。较大的 ω 有利于提高算法的全局探索能力, 较小的 ω 有利于提高算法的局部发掘能力, ω 的选取对平衡这两种能力有重要的作用。

受文献[10]的启发, 本文提出以下惯性权重修正公式:

$$\omega = \omega_{\min} + \exp(-(a * t / T_{\max})^2) * (\omega_{\max} - \omega_{\min}) \quad (7)$$

式中, a 的取值是通过经验来决定的, 一般为 $a \in [45, 65]$ 。通过这种方式调整, 算法在迭代的前期惯性权重较大, 从而有较强的全局探索能力; 随着迭代的进行, 惯性权重迅速减小, 算法就具有较强的局部发掘能力, 从而加快了收敛速度。

综合 3.1—3.3 节, 本文提出一种带扰动操作的自适应粒子群算法(Adaptive PSO with disturbance factors), 简称 tAPSO, 该算法的步骤如下:

步骤 1 $t=1$ 时, 在允许的范围内, 对所有粒子混沌初始化位置 x_i 与速度 v_i , $i \in [1, n]$, n 为粒子个数;

步骤 2 将第 i 个粒子的 p_{best} 设置为当前位置, g_{best} 设置为初始群体的最佳位置;

步骤 3 按式(5)—式(7)更新粒子的速度和位置;

步骤 4 计算粒子 i 的适应度 $f(x_i)$, $i \in [1, n]$, $f(x)$ 为适应度函数;

步骤 5 如果粒子 i 的适应度优于自身个体极值 p_{best} 的适应度, 就将该粒子的 p_{best} 更新为 x_i ;

步骤 6 如果在当前的迭代中, 粒子 i 的适应度优于全局极值 g_{best} 的适应度, 就将 g_{best} 更新为 x_i ;

步骤 7 判断算法是否达到最大迭代次数, 如果满足就转步骤 8, 否则就转步骤 3;

步骤 8 输出全局最优解 g_{best} , 并求出相应的目标函数值, 算法结束。

4 仿真实验

对标准粒子群优化算法(PSO)、文献[7]提出的带极值扰动的粒子群优化算法(tPSO)、文献[10]提出的混合变异算子的自适应粒子群优化算法(APSO)和本文提出的带扰动因子的自适应粒子群优化算法(tAPSO), 选用 4 个基准测试函数(理论最小值均为 0), 分别进行优化性能测试。

Sphere 函数

$$f_1(x) = \sum_{j=1}^D x_j^2, x \in [-100, 100]^{30}$$

Ackley 函数

$$f_2(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{j=1}^D x_j^2}) - \exp(\frac{1}{D} \sum_{j=1}^D \cos 2\pi x_j) + 20 + \exp(1), x \in [-100, 100]^{30}$$

Griewank 函数

$$f_3(x) = \frac{1}{4000} \sum_{j=1}^D x_j^2 - \prod_{j=1}^D (\cos \frac{x_j}{\sqrt{j}}) + 1, x \in [-100, 100]^{30}$$

Rastrigrin 函数

$$f_4(x) = \sum_{j=1}^D [x_j^2 - 10 \cos(2\pi x_j) + 10], x \in [-10, 10]^{30}$$

粒子群的种群规模 $n=30$, SPSO 和 tAPSO 算法中: $c_1=c_2=1.4962$, $\omega_{\max}=0.95$, $\omega_{\min}=0.5$; tPSO 和 APSO 算法的参数按原文献的设置。针对每个函数运行 30 次求平均值, 仿真结果与统计数据见表 1。为了进一步比较各算法的性能, 图 1—图 4 给出了用这 4 种算法分别测试每个函数的适应度进化曲线。

(下转第 103 页)

mization [J]. Applied Soft Computing, 2012, 12: 320-332

[5] Kang F, Li J J, Ma Z Y. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions [J]. Information Sciences, 2011, 181(1): 3508-3531

[6] Gao W F, Liu S Y. Improved artificial bee colony algorithm for global optimization [J]. Information Processing Letters, 2011, 111(17): 871-882

[7] 毕晓君, 王艳娇. 加速收敛的人工蜂群算法[J]. 系统工程与电子技术, 2011, 33(12): 2755-2761

[8] 张新明, 孙印杰. 基于混沌优化的自适应中值滤波[J]. 电子技术应用, 2007, 33(9): 63-65

[9] 张新明, 徐久成. 基于混沌理论和支持向量机的人脸识别方法[J]. 高技术通讯, 2009, 19(5): 494-500

[10] Alatas B. Chaotic bee colony algorithms for global numerical op-

mization [J]. Expert Systems with Applications, 2010, 37: 5682-5687

[11] 罗钧, 李研. 具有混沌搜索策略的蜂群优化算法[J]. 控制与决策, 2010, 25(12): 1913-1916

[12] 李志勇, 李玲玲, 王翔, 等. 基于 Memetic 框架的混沌人工蜂群算法[J]. 计算机应用研究, 2012, 29(11): 4045-4049

[13] 张新明, 雷冠军, 闫林, 等. 一种新型快速的直接随机优化算法[J]. 吉林大学学报: 理学版, 2012, 50(4): 750-756

[14] Dong H B, He J, Huang H K, et al. Evolutionary programming using a mixed mutation strategy [J]. Information Sciences, 2007, 177(1): 312-327

[15] Zhang J, Sanderson A C. JADE: adaptive differential evolution with optional external archive [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958

(上接第 69 页)

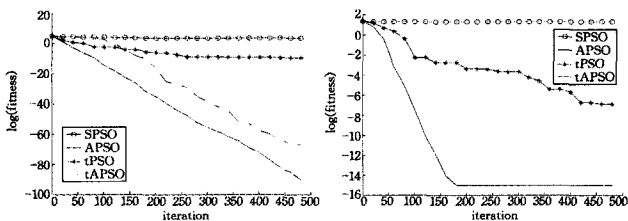


图 1 函数 f_1 寻优进化曲线

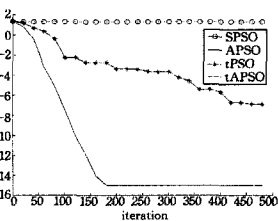


图 2 函数 f_2 寻优进化曲线

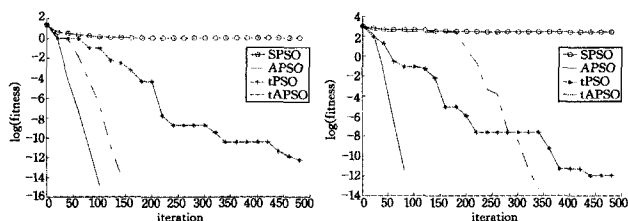


图 3 函数 f_3 寻优进化曲线

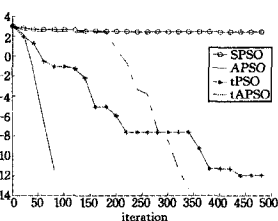


图 4 函数 f_4 寻优进化曲线

表 1 测试结果

函数	算法	最佳值	最差值	平均值
f_1	SPSO	39.7475	1.0347×10^4	753.0247
	tPSO	2.8097×10^{-20}	2.3204×10^{-11}	9.7722×10^{-12}
	APSO	1.6456×10^{-70}	1.2172×10^{-37}	4.0573×10^{-39}
	tAPSO	1.1700×10^{-101}	1.4614×10^{-89}	5.4920×10^{-91}
f_2	SPSO	20	20	20
	tPSO	2.8050×10^{-111}	1.3377×10^{-005}	1.3159×10^{-006}
	APSO	20	20	20
	tAPSO	8.8818×10^{-016}	8.8818×10^{-016}	8.8818×10^{-016}
f_3	SPSO	0.9760	3.1590	1.1514
	tPSO	0	4.1296×10^{-012}	3.5261×10^{-013}
	APSO	0	0	0
	tAPSO	0	0	0
f_4	SPSO	113.5420	370.7751	221.7465
	tPSO	8.8818×10^{-015}	1.7926×10^{-009}	9.0128×10^{-011}
	APSO	0	214.3522	24.1629
	tAPSO	0	0	0

从图 1 可以看出 tAPSO 的收敛精度在指数级别上优于其它 3 种算法, 收敛速度也更快, 能够收敛于很接近理论最优值的全局最优解, 具有很好的优化性能。从图 2 可以看出, 只有 tAPSO 算法在进化的前期突破了障碍物, 越过了局部最优解, 从而在进化的末期能够收敛到比较接近理论最优值的全局最优解, 具备良好的收敛效果。从图 3 可以看出: SPSO 算法不能找到全局最优解, APSO 算法绕过了一部分的局部最

优点, 但是进化的过程比较坎坷, 优化性能不够稳定; tPSO 算法和 tAPSO 算都能够最后收敛于比较接近理论最优解的全局最优解, 并且收敛速度很快, 获得了较好的收敛效果。从图 4 可以发现: SPSO 算法优化性能不好; APSO 算法虽然取到全局最优值, 但优化效果不明显; tPSO 算法最终收敛到全局最优, 且收敛的速度也较快, 收敛精度也较高, 具有良好的优化性能; tAPSO 算法具有比较突出的优势, 算法的收敛速度、收敛精度和平均收敛值都要优于其它的算法。

从表 1 和图 1—图 4 中可以看出, 对于每一个测试函数, 本文提出的 tAPSO 算法在收敛速度和收敛精度上都明显优于其它 3 种算法。

结束语 本文在标准 PSO 算法的基础上进行改进, 提出了带扰动因子的自适应粒子群优化算法。实验研究表明, 提出的新算法具有较强的全局搜索能力、较高的收敛精度和较快的收敛速度。

参 考 文 献

[1] Kennedy J, Eberhart R. Particle swarm optimization[C]//IEEE International Conference on Neural Networks. Perth, 1995: 1942-1948

[2] 吴伟, 李楠, 郭茂耘. 粗糙集及 PSO 优化 BP 网络的故障诊断研究[J]. 计算机科学, 2011, 11(38): 200-203

[3] 苗启广, 王明静, 王宝树. 基于归一化互信息与模糊自适应 PSO 的图像自动配准方法[J]. 计算机科学, 2008, 6(35): 175-177

[4] 秦洪德, 石丽丽. PSO 算法在油船双层结构优化设计中的应用研究[J]. 哈尔滨工程大学学报, 2010, 8(31): 1007-1011

[5] 肖奔贤. 基于改进 PSO 算法的过热汽温神经网络预测控制[J]. 控制理论与应用, 2008, 3(25): 569-573

[6] 刘军民, 高岳林. 混沌优化算法[J]. 计算机应用, 2008, 28(2): 322-325

[7] 胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868

[8] 赵志刚, 常成. 简化的自适应粒子群优化算法[J]. 广西大学学报, 2010, 35(5): 793-798

[9] Shi Y, Eberhart C. A modified particle swarm optimizer[C]//IEEE International Conference Evolutionary Computation. Anchorage, Alaska, 1998, 5: 4-9

[10] 安晓会, 高岳林. 混合变异算子的自适应粒子群优化算法[J]. 计算机应用, 2008, 28(6): 28-30