

文章编号: 1001-0920(2008) 11-1253-05

一种动态改变惯性权重的自适应粒子群算法

张顶学¹, 关治洪², 刘新芝²

(1. 长江大学 石油工程学院, 湖北 荆州 434203; 2. 华中科技大学 控制科学与工程系, 武汉 430074)

摘 要: 针对标准粒子群算法在进化过程中种群多样性降低而早熟的问题, 提出一种动态改变惯性权重的自适应粒子群算法. 采用种群中平均粒子相似程度作为种群多样性的测度, 并用于平衡算法的全局探索和局部开发. 基于对惯性权重随种群多样性测度变化的动态分析, 建立了惯性权重随种群多样性测度的变化关系, 并将其引入该算法中. 最后对 6 个经典测试函数进行仿真, 结果表明该算法在平均最优值和成功率上都有所提高, 特别是对多峰函数效果更明显.

关键词: 粒子群算法; 惯性权重; 自适应; 种群多样性

中图分类号: TP18

文献标识码: A

Adaptive particle swarm optimization algorithm with dynamically changing inertia weight

ZHANG Ding-xue¹, GUAN Zhi-hong², LIU Xin-zhi²

(1. Petroleum Engineering College, Yangtze University, Jingzhou 434203, China; 2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. Correspondent: ZHANG Ding-xue, E-mail: zdx7773@126.com)

Abstract: To overcome the premature caused by standard particle swarm optimization (PSO) algorithm searching for the large lost in population diversity, an adaptive PSO with dynamically changing inertia weight is proposed. The average of similarity of particles in the population as the measure of population diversity is introduced into proposed algorithm to balance the trade-off between exploration and exploitation. A function relationship between inertia weight and the measure of population diversity is established by analyzing the dynamically relationship between them, which is embedded into the algorithm. The simulation results show that the algorithm has better probability of finding global optimum and mean best value, especially for multimodal function.

Key words: Particle swarm optimization; Inertia weight; Adaptive; Population diversity

1 引 言

粒子群优化(PSO)算法是一种新的基于群体智能的优化算法,它是在模拟鸟群觅食行为规律的基础上提出的^[1,2].由于其结构简单、参数调整简单易行以及适合计算机编程处理等优点,一提出就受到广泛关注,目前已成功应用于函数优化、神经网络训练及系统控制等领域^[3].

PSO 算法作为一种随机搜索策略的优化方法,其算法的探索能力较好,收敛速度较快,但也存在易陷入局部最优的问题,对于复杂问题的开发能力较弱,在后期进化中很难获得更精确的解.在 PSO 中可使用惯性权重来改善算法的性能,很多学者在这

方面做了工作.文献[2]提出的随着进化代数的增加线性减小惯性权重,虽然能起到一定作用,但在复杂问题求解时效果不是很明显.[4]提出了用模糊控制器来自适应改变惯性权重,该算法需建立模糊规则,实现比较复杂.[5]采用随机惯性权重策略,仿真实验结果表明,该方法对于多峰函数能在一定程度上避免陷入局部最优.[6]提出利用进化过程中粒子适应度的差异来评价种群的早熟收敛程度,然后动态地改变惯性权重,仿真结果表明能取得较好的效果.本文则提出一种基于种群多样性测度动态改变惯性权重的自适应粒子群算法.仿真结果表明,该算法具有很强的全局探索和局部开发的能力.

收稿日期: 2007-09-23; 修回日期: 2008-01-07.

基金项目: 国家自然科学基金项目(60573005, 60603006); 湖北省高等学校优秀中青年团队计划项目(T200803).

作者简介: 张顶学(1975—),男,湖北潜江人,讲师,博士,从事智能控制理论与应用的研究;关治洪(1955—),男,湖北仙桃人,教授,博士生导师,从事复杂网络、非线性系统控制等研究.

2 PSO 算法

PSO 算法是由 Kennedy 和 Eberhart^[1,2] 在 1995 年提出的,是一种模拟鸟群飞行觅食行为,并通过个体之间的协作来寻找最优解的进化计算技术.假设其搜索空间为 N 维,粒子总数为 n ,第 i 个粒子在 N 维空间的位置为 x_i ,飞行速度为 v_i .每个粒子都具有一个由被优化的目标函数决定的适应值,并知道自己到目前为止所发现的最好位置 p_i 和现在的位置 x_i ,每个粒子都知道目前为止整个群体所发现的最好位置 p_g .每个粒子的位置按下式进行变化:

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times r_1 \times (p_{id} - x_{id}^k) + c_2 \times r_2 \times (p_{gd} - x_{id}^k), \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (2)$$

其中: v_{id}^k 为第 i 个粒子第 k 次迭代中飞行速度的第 d 维分量, x_{id}^k 为第 i 个粒子第 k 次迭代中位置的第 d 维分量, p_{gd} 为群体最好位置 p_g 的第 d 维分量, p_{id} 为粒子 i 最好位置 p_i 的第 d 维分量, r_1 和 r_2 为随机数, c_1 和 c_2 为权重因子, w 为惯性权重^[2],即

$$w = w_{\max} - (w_{\max} - w_{\min}) \times k / G. \quad (3)$$

这里: $w_{\max}$, $w_{\min}$ 分别是 w 的最大和最小值; G , k 分别为最大迭代次数和当前迭代次数.

3 种群多样性的测度及计算方法

目前常用平均粒距^[7] (种群中各个个体相互之间的离散程度) 和粒子适应度之间的差异^[6] 表示种群多样性.因为粒子的适应度是粒子位置的反应,所以这两种种群多样性的测度实质上是以粒子目前位置的离散程度表现种群多样性,而没有考虑粒子未来位置的差异,因此并不完善.种群多样性应该表示种群中个体的相似程度.若两个粒子目前的位置较近,而历代最优位置差别较大,则在进化的下一代,粒子的位置可能会出现很大差别.因此,说这两个粒子的相似程度较大并不妥当.

式(1)中设 $\phi = c_1 r_1$, $\phi = c_2 r_2$, $\phi = \phi + \phi$, 在参数选择合理的情况下,粒子将收敛于 $(\phi p_i + \phi p_g) / \phi$ 位置^[8-10], 种群的全局最优位置 p_g 对于每个粒子是相同的,因此,粒子的收敛位置的差别和粒子的历代最优位置有关.而粒子的目前位置表示了粒子目前位置与收敛位置之间的差异以及达到收敛位置的难易程度.因此,粒子之间的相似程度可用粒子目前位置和历代最优位置的相似程度表示.这样,种群中平均的粒子相似程度可作为种群多样性的测度.

设粒子群算法中个体数为 n ,第 i 代种群 X_i 中第 j 个个体为 $x_i^j = (x_i^{j(1)}, x_i^{j(2)}, \dots, x_i^{j(N)})$, 其中 N 为

解空间的维数.第 j 个个体历代最优位置为 $y^j = (y^{j(1)}, y^{j(2)}, \dots, y^{j(N)})$.将个体的目前位置和历代最优位置串在一起,记为 $T^j = (x_i^{j(1)}, x_i^{j(2)}, \dots, x_i^{j(N)}, y^{j(1)}, y^{j(2)}, \dots, y^{j(N)})$.对于种群中所有粒子的 T^j ($1 \leq j \leq n$) 可组成一个 $n \times 2N$ 阶的矩阵 T .对 T 作归一化处理,可得 $n \times 2N$ 阶矩阵 T' , 其中

$$T'_{uv} = \frac{T_{uv} - \min_{1 \leq u \leq n, 1 \leq v \leq 2N} T_{gl}}{\max_{1 \leq u \leq n, 1 \leq v \leq 2N} T_{gl} - \min_{1 \leq u \leq n, 1 \leq v \leq 2N} T_{gl}}, \quad 1 \leq u \leq n, 1 \leq v \leq 2N;$$

每个行向量 T_j' 可看成一个模糊集合,表示第 j 个粒子目前位置和历代最优位置各分量的隶属度;任意两个 T_u' , T_v' 的相似程度可用贴近度表示,记为

$$L(u, v) = 1 - \frac{1}{2N} \sum_{t=1}^{2N} |T_u^t - T_v^t|.$$

种群的多样性测度 F 用种群平均贴近度表示,即

$$F = \frac{2}{n(n-1)} \sum_{u=1}^{n-1} \sum_{v=u+1}^n L(u, v).$$

由 $0 \leq L(u, v) \leq 1$, 得到 $F \leq 1$.若种群中粒子完全相同,则多样性最差, F 为最大值 1.

4 动态改变惯性权重的自适应粒子群算法

4.1 惯性权重的改变策略

文献[2]研究了惯性权重 w 对算法性能的影响,较大的惯性权重有利于全局探索,而较小的惯性权重有利于算法的局部开发,加速算法的收敛.当 F 较大时,种群多样性较差,可能出现早熟,应增大惯性权重,加强全局探索,提高种群的多样性;当 F 较小时,种群多样性较好,应加强局部开发,减小惯性权重.设惯性权重的调整公式

$$w(k) = 1 / (a - bF(k-1)). \quad (4)$$

其中: $w(k)$ 为第 k 代进化的惯性权重; $F(k-1)$ 为第 $k-1$ 代种群多样性的测度; a , b 为常数.

为了保证算法收敛到最优值,在进化后期应加强局部开发能力,种群的多样性必然会降低, F 必然会增大.如果增大惯性权重,必然会导致优秀粒子被破坏的可能性增加,因此,在调整惯性权重时,应允许种群 F 适当增大.设初始 F 的允许值为 F_0 , 则第 k 代种群 F 的允许值为

$$F_k = F_0 + k(1 - F_0) / G. \quad (5)$$

若第 k 代 F 大于允许值 F_k , 则采用式(5)对惯性权重进行调整;否则表明种群多样性相对较好,应加强局部开发,在局部搜索更优解,此时惯性权重的选择应考虑适当的降低.在进化初期可能出现种群多样性相对较好,因此应加强全局探索.综合考虑这两方面因素,惯性权重应采用指数下降形式,即

$$w = w_{\min} + (w_{\max} - w_{\min}) \cdot e^{-(4w/G)^2}, \quad (6)$$

惯性权重调整方式(6)在进化早期具有较大的惯性权重,而随着进化代数的增加,惯性权重迅速下降.与惯性权重的线性调整方式相比,该方式更能提高算法的局部开发能力.

4.2 参数分析

设种群初始成熟度为 F_0 , 则 F 允许值的范围为 $[F_0, 1]$. a, b 的值确定了惯性权重的调整范围, 在进化初期, 算法全局探索的功能应增强, 因此若出现种群多样性变差, 惯性权重的调整幅度应大一些; 而在进化后期, 若出现种群多样性变差, 惯性权重的调整幅度应小一些. 由此, 将惯性权重的改变范围设为 $[w_0, 1]$, 可得

$$a = (1 - w_0 F_0) / (w_0 (1 - F_0)),$$
$$b = (1 - w_0) / (w_0 (1 - F_0)).$$

将其代入式(4) 可得

$$w = \frac{w_0 (1 - F_0)}{1 - w_0 F_0 - (1 - w_0) F(t - 1)}. \quad (7)$$

F_0 在计算开始时便可得到, 因此惯性权重的改变只与 w_0 有关.

4.3 算法的实现

算法的流程描述如下:

Step1: 初始化粒子群, 包括各参数: 粒子速度、位置和个体极值 p_i , 全局极值 p_g ;

Step2: 按式(1), (2) 更新粒子的速度和位置;

Step3: 计算种群中各粒子的适应度;

Step4: 根据适应度更新粒子的个体极值 p_i ;

Step5: 根据适应度更新粒子的全局极值 p_g ;

Step6: 计算种群多样性测度 F , 并与式(5) 所计算的允许值相比较, 按式(6) 或(7) 更新惯性权重; 判断算法的终止条件, 如满足则停止, 输出相关结果; 否则转 Step2.

5 算法有效性的测试

为了测试算法的有效性, 应用本文算法 (APSO) 对 6 个经典的函数优化进行仿真, 并与提出的自适应算法^[6] (FAPSO)、惯性权重线性下降的标准粒子群算法^[2] (LWPSO)、带压缩因子的粒子群算法^[8] (CFPSO) 和随机惯性权重粒子群算法^[5] (RWPSO) 进行比较. 6 个优化函数如表 1 所示. 表中: f_1 为简单的单峰函数, 在 原点达到极小值; f_2 为非凸的病态函数, 在 $x_i = 1$ 时达到极小值; f_3 为多峰函数, 当 $x_i = 0$ 时达到全局极小值, 在其周围存在多个局部极小点; f_4 在 $x_i = 0$ 时达到全局极小值, 当 $x_i = \pm k\pi\sqrt{i}$ ($i = 1, 2, \cdots, n, k = 1, 2, \cdots, n$) 时, 达到局部极小值; f_5 的全局极大值在 $(0, 0)$, 它的全局最优点被次优点所包围; f_6 为多峰函数,

表 1 测试函数的维数、初值范围和目标值				
简称	函数名	维数	初值范围	目标值
f_1	Sphere	30	$[-100, 100]^{30}$	0.01
f_2	Rosenbrock	30	$[-100, 100]^{30}$	100
f_3	Rastrigrin	30	$[-100, 100]^{30}$	100
f_4	Griewank	30	$[-600, 600]^{30}$	0.1
f_5	Schaffer f_6	2	$[-100, 100]^2$	0.991
f_6	Ackley	30	$[-32, 32]^{30}$	0.1

存在多个局部最优值.

为了测试算法的性能, 组织两类实验模式. 实验模式 I: 给定最大进化代数 $T_{\max}$, 执行测试算法; 实验模式 II: 给定最大进化代数 $T_{\max}$ 和目标最优值 f_b , 执行测试算法. 在实验模式 II 中, 经过 $T_{\max}$ 迭代, 算法搜索到的最优值为 f , 若 $f \leq f_b$, 则实验成功; 否则实验失败. 为了评价算法的优劣, 采用以下两个评价指标:

- 1) 平均最优适应值 f_{avg} : M 次实验 I 所得函数值的算术平均值.
- 2) 平均成功率 R : 对算法进行 M 次实验 II, 若其中 M_s 次实验成功, 则平均成功率 $R = M_s / M \times 100\%$.

各种算法种群数取为 20, $w_{\max} = 0.95, w_{\min} = 0.4$, 除 f_5 中两种实验模式的 最大进化代数都为 1000 外, 其余各函数的最大进化代数都为 2000. 每种算法对每个函数独立运行 20 次.

5.1 不同 w_0 下 APSO 算法的性能

选取 0.4, 0.5, 0.6 和 0.7 四个值进行仿真实验, 其结果如表 2 所示.

从表中可以看出, 随着 w_0 的增大, f_{avg} 和 R 先有所改善而后又变差, 在 $w_0 = 0.5$ 时基本达到最优. 其主要原因是较大的 w_0 能增强粒子的探索能力, 更好地保持种群的多样性, 但算法的局部开发能力受到了限制; 而较小的 w_0 能增强算法的开发能力, 但算法的探索能力较差, 粒子跳出局部最优的可能性降低. 在后面的对比中采用 $w_0 = 0.5$.

5.2 不同算法之间的对比

将本文算法与 LWPSO, FAPSO, CFPSO 和 RWPSO 4 种算法进行对比, 结果如表 3 所示.

从表 3 中可以看出, LWPSO 和 CFPSO 算法性能最差. RWPSO 算法在对多峰函数优化上, 平均成功率和平均最优值都有所改善, 但对单峰函数 f_1 效果较差, 其主要原因是采用随机惯性权重策略虽然能增大算法的探索能力, 保持种群的多样性, 但算法在后期缺乏局部开发, 很难搜索到最优解. APSO 算法相比 FAPSO 算法, 无论是在平均成功率上还是

表 2 不同 w_0 下 APSO 算法实验结果

函数	w_0							
	0.4		0.5		0.6		0.7	
	f_{avg}	R	f_{avg}	R	f_{avg}	R	f_{avg}	R
f_1	1.93e-24	100	1.43e-18	100	2.69e-15	100	7.98e-17	100
f_2	65.48	85	36.09	95	65.39	80	45.95	90
f_3	63.88	95	65.47	100	58.50	100	71.06	90
f_4	0.02035	100	0.00824	100	0.01611	100	0.01719	100
f_5	0.99514	50	0.99951	95	0.99611	60	0.99757	75
f_6	0.0751	95	1.12e-5	100	0.1682	90	0.1970	90

表 3 各种算法性能的比较结果

算法	f_1		f_2		f_3		f_4		f_5		f_6	
	f_{avg}	R	f_{avg}	R	f_{avg}	R	f_{avg}	R	f_{avg}	R	f_{avg}	R
APSO	1.43e-18	100	36.09	95	65.47	100	0.00824	100	0.99951	95	1.125e-5	100
FAPSO	1.99e-10	100	48.28	95	83.08	80	0.01838	100	0.99660	65	0.1329	90
LWPSO	1.16e-6	100	358.52	20	91.44	65	0.01833	100	0.99351	40	0.2221	80
CFPSO	1.63e-23	100	150.47	65	170.73	10	0.03712	95	0.99514	50	2.9282	0
RWPSO	118.2367	0	68.92	85	86.36	75	0.01682	95	0.99852	85	3.5926	0

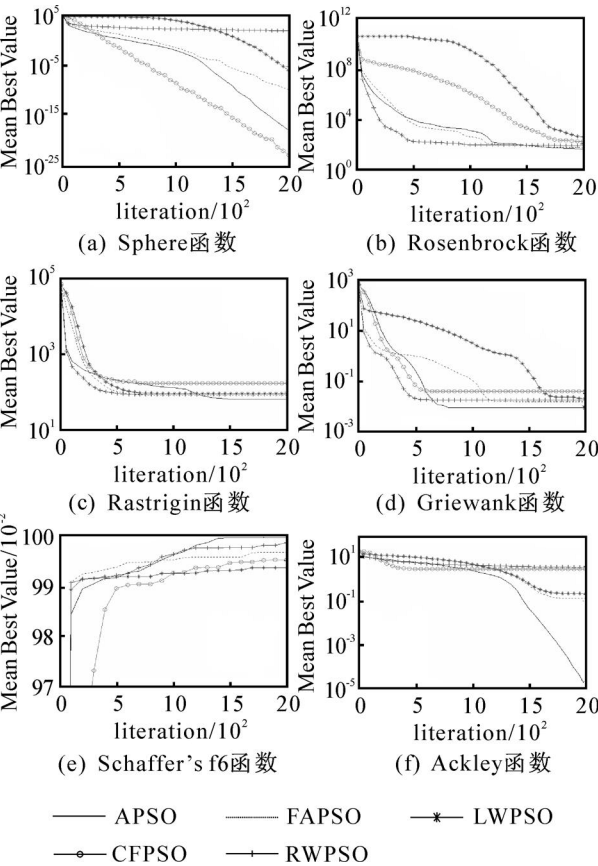


图 1 6 个函数进化过程中的平均适应度变化曲线

在平均最优值上都有所改善,但 APSO 算法对 f_1 和 f_4 函数的平均最优值改善较小,其主要原因是 f_1 为单峰函数, f_4 在维空间超过 15 维后, Griewank 函数特性趋向于单峰,从而 APSO 算法的优越性显示不

出来. 各函数进化过程中平均适应度变化曲线如图 1 所示.

图 2 和图 3 给出了 Rastrigin 函数各算法在进化过程中的种群多样性测度变化曲线和在 APSO 算法中的惯性权重的变化曲线.

由图 2 可以看出, RWPSO 算法在进化过程中种群多样性最好,其主要原因是由于该算法采用随

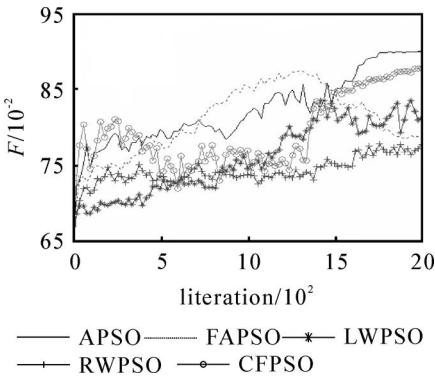


图 2 种群多样性测度的变化曲线

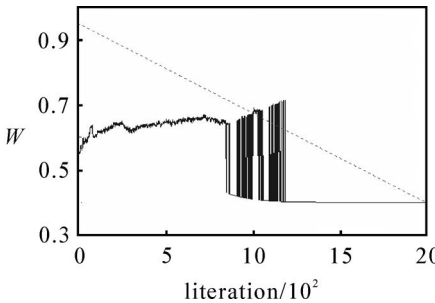


图 3 APSO 算法中惯性权重的变化

机惯性权重策略使其探索能力较强而局部开发能力较弱,因此在多峰函数优化问题上相比于 LWPSO 和 CFPSO 算法,RWPSO 算法具有较好的成功率.FAPSO 相比于 APSO 算法在进化过程中种群多样性保持较好,但在进化后期局部开发能力较弱.APSO 算法在进化后期种群多样性最差,说明该算法具有很强的局部开发能力.由图 3 可以看出,在进化早期,APSO 算法的惯性权重较小,局部开发能力加强,种群的多样性急剧变差;在进化中期,该算法采用动态的惯性权重平衡了其全局探索和局部开发能力,种群多样性测度呈现出稳中有升的趋势;在进化后期该算法的局部开发能力较强,种群多样性又急剧变差并趋于稳定.正是由于 APSO 算法中惯性权重的动态改变平衡了全局探索和局部开发的能力,使算法的性能得到很大提高.

6 结 论

本文以种群中个体目前位置和历代最优位置的相似程度作为种群多样性的测度,并由此动态调整惯性权重,使算法的全局探索和局部开发能力得到平衡,从而能有效地逃出局部最优.6 个函数优化的仿真结果表明,本文算法无论是在成功率上还是在平均最优值上都有很大改善,特别是多峰函数,其算法性能的改善更明显.

参考文献(References)

[1] Kennedy J, Eberhart R. Particle swarm optimization [C]. IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.
[2] Shi Y, Eberhart R. A modified particle swarm optimizer [C]. IEEE World Conf on Computational Intelligence. Piscataway: IEEE Press, 1998: 69-73.
[3] 谢晓锋, 张文俊, 杨之廉. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-134.

(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-134.)
[4] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization [C]. Proc of the IEEE Conf on Evolutionary Computation. Piscataway: IEEE Press, 2001: 101-106.
[5] Zhang L P, Yu H J, Hu S X. A new approach to improve particle swarm optimization[C]. Lecture Notes in Computer Science. Chicago: Springer-Verlag, 2003: 134-139.
[6] 韩江洪, 李正荣, 魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10): 2969-2971.
(Han J H, Li Z R, Wei Z C. Adaptive particle swarm optimization algorithm and simulation[J]. J of System Simulation, 2006, 18(10): 2969-2971.)
[7] Krink T, Vesterstroem J S, Riget J. Particle swarm optimization with spatial particle extension[C]. Proc of the IEEE Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1474-1479.
[8] Clerc M. The swarm and queen: Towards a deterministic and adaptive particle swarm optimization [C]. Proc of IEEE Conf on Evolutionary Computation. Washington D C, 1999: 1951-1957.
[9] 刘洪波, 王秀坤, 谭国真. 粒子群优化算法的收敛性分析及混沌改进算法[J]. 控制与决策, 2006, 21(6): 636-640.
(Liu H B, Wang X K, Tan G Z. Convergence analysis of particle swarm optimization and its improved algorithm based on chaos [J]. Control and Decision, 2006, 21(6): 636-640.)
[10] Frans van den Bergh. An analysis of particle swarm optimizers[D]. Pretoria: University of Pretoria, 2001.

下 期 要 目

量子进化算法研究进展 王 凌
一种快速稀疏最小二乘支持向量回归机 赵永平, 孙健国
基于控制微元的统计自适应增量控制方法 孙 波, 等
基于多指标灰区间数关联决策模型的产品方案设计 钟诗胜, 等
基于变论域模糊系统的空天飞行器直接自适应预测控制 方 炜, 姜长生
基于双基点法的多传感器数据融合 万树平
基于粒子滤波的模型自适应机动目标跟踪算法 胡振涛, 等