文章编号:1001 - 9081(2006)10 - 2513 - 03

飞行时间自适应调整的粒子群算法

张建科[1,2],刘三阳[2],张晓清[3]
(1. 西安邮电学院 应用数学与物理系,陕西 西安 710121;
2. 西安电子科技大学 理学院,陕西 西安 710071;
3. 西安电子科技大学 软件学院,陕西 西安 710071)
(jiankezh@163.com)

摘　要:为改善粒子群优化算法的搜索性能,提出一种飞行时间自适应调整的粒子群算法(FAA-PSO)。该算法在粒子群进化过程中随着进化代数增大自适应调整粒子的飞行时间,从而克服了传统粒子群算法中粒子飞行时间固定为 1 导致的粒子在迭代后期搜索性能下降的困难。数值结果表明,该算法有利于加速收敛,提高收敛精度。

关键词:粒子群算法;进化算法;优化
中图分类号: TP18　　**文献标识码**:A

Particle swarm optimization with flying time adaptively adjusted

ZHANG Jian-ke[1,2], LIU San-yang[2], ZHANG Xiao-qing[3]
(1. *Department of Mathematics and Physics, Xi'an University of Post and Telecommunications, Xi'an Shaanxi* 710121, *China;*
2. *School of Science, Xidian University, Xi'an Shaanxi* 710071, *China;*
3. *School of Software, Xidian University, Xi'an Shaanxi* 710071, *China*)

Abstract: To improve the searching performance of Particle Swarm Optimization (PSO), a modified PSO algorithm with flying time adaptively adjusted was proposed and named FAA-PSO algorithm. The flying time of every particle in this algorithm was adaptively adjusted in pace with addition of the evolutionary generations; Thus, the algorithm overcomes the difficulty of the traditional PSO that the searching ability of particle is decreasing during the later time of iteration, which is caused by that the flying time of every particle is fixed on one. Numerical results show that this algorithm is of advantage to accelerate convergence and improve calculation accuracy.

Key words: Particle Swarm Optimization(PSO); evolutionary computation; optimization

0　引言

粒子群算法(Particle Swarm Optimization, PSO)[1,2]是 Kennedy 和 Eberhart 于 1995 年提出的一种群体智能算法。因其易理解,易实现,很多情况下比遗传算法更有效,近年来受到学术界的广泛关注,并提出了很多改进算法。现在,作为一种新的全局优化算法,PSO 算法在很多问题中已得到成功应用,如:函数优化[3]、系统辨识[4]、神经网络训练[5]等领域。

和其他的智能优化算法一样,粒子群算法也存在易收敛到局部最优点的缺陷。尤其是对复杂的多峰函数优化问题,传统的粒子群算法效率较低。为提高算法的性能,已有很多学者通过分析 PSO 算法的参数对优化性能的影响提出了很多改进方案,例如:惯性权重法[6]、压缩因子法(也称收缩因子法)[7,10]、杂交粒子群算法[8]等。这些方法很好的解决了早熟收敛问题,但仍存在对有的测试函数收敛速度较低或稳定性差的问题。

本文通过分析粒子群算法中粒子的飞行时间对优化性能的影响,提出了一种飞行时间自适应调整的粒子群算法(FAA-PSO)。该方法很好的克服了由于传统粒子群算法固定粒子飞行时间从而导致的粒子在进化后期搜索性能下降的问题。本文选取几个基准测试函数进行了测试,仿真结果表明,该算法与传统的粒子群算法相比其性能有了很大的提高。

1　PSO 算法思想

假设在一个 S 维的目标搜索空间中,有 m 个粒子组成一个群体,其中第 i 个粒子表示为一个 S 维的向量 $\vec{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iS})$, $i = 1, 2, \cdots, m$,即第 i 个粒子在 S 维搜索空间中的位置即每个粒子的位置就是一个潜在的解。将 $\vec{x}_i$ 代入一个目标函数就可以算出其适应值,根据适应值的大小衡量解的优劣。第 i 个粒子的飞翔的速度是 s 维向量,记为 $\vec{V} = (v_{i1}, v_{i2}, \cdots, v_{iS})$。记第 i 个粒子迄今为止搜索到的最优位置为 $\vec{P}_{is} = (p_{i1}, p_{i2}, \cdots, p_{is})$,整个粒子群迄今为止搜索到的最优位置为 $\vec{P}_{gs} = (p_{g1}, p_{g2}, \cdots, p_{gs})$。

最早提出的粒子群算法采用下列公式对粒子操作:

$$v_{is}^{t+1} = v_{is}^{t} + c_1 r_1 (p_{is} - x_{is}^{t}) + c_2 r_2 (p_{gs} - x_{is}^{t}) \tag{1}$$

$$x_{is}^{t+1} = x_{is}^{t} + v_{is}^{t} \tag{2}$$

收稿日期:2006 - 04 - 10　　**基金项目**:陕西省自然科学研究项目(2003A09)
作者简介:张建科(1978 -),男,陕西汉中人,硕士研究生,主要研究方向:智能优化算法、粒子群优化算法;　刘三阳(1959 -),男,陕西临潼人,教授,博士生导师,主要研究方向:最优化理论及应用;　张晓清(1981 -),陕西汉中人,硕士研究生,主要研究方向:软计算.

其中,$i = 1,2,\cdots,m, s = 1,2,\cdots,S$;学习因子$c_1$和$c_2$是非负常数;$r_1$和$r_2$为伪随机数,服从$[0,1]$上的均匀分布。$v_{is} \in [-v_{\max}, v_{\max}]$,$v_{\max}$为常数,由用户设定。

文献[6]对(1)作了如下改进:

$$v_{is}^{t+1} = \omega v_{is}^t + c_1 r_1 (p_{is} - x_{is}^t) + c_2 r_2 (p_{gs} - x_{is}^t) \tag{3}$$

在(3)中ω为非负数,称为动力常量,控制着前一速度对当前速度的影响,ω较大时,前一速度影响较大,全局搜索能力较强;ω较小时,前一速度影响较小,局部搜索能力较强。通过调正ω的大小来跳出局部极小值。

文献[10]提出了压缩因子(也称收缩因子)的概念。该方法描述了一种选择ω,c_1和c_2的方法,通过正确选择这些控制参数,就没必要限制速度范围。对(1)作了如下改进[10]:

$$v_{is}^{t+1} = \alpha (v_{is}^t + c_1 r_1 (p_{is} - x_{is}^t) + c_2 r_2 (p_{gs} - x_{is}^t)) \tag{4}$$

α为压缩因子,用来控制速度的权值。

$$\alpha = \frac{2}{|2 - l - \sqrt{l^2 - 4l}|},\text{且 } l = c_1 + c_2,\ l > 4$$

文献[7]分别利用$v_{\max}$和α控制速度,并作了比较,结果表明后者较好;但对有的函数压缩因子法在给定的迭代次数内无法达到全局极值点。他们认为这是由于粒子偏离所期望的搜索空间太远造成的,因此建议预先设置搜索空间大小或者设参数$v_{\max} = x_{\max}$。

迭代终止条件根据具体问题一般选为最大迭代次数或粒子群迄今为止搜索到的最优位置满足的预定最小适应阈值。

2 飞行时间自适应调整的粒子群算法

为提高粒子群算法的收敛速度和收敛精度,降低早熟收敛的比率,本文提出了飞行时间自适应调整的粒子群算法(FAA-PSO)。该方法在传统方法的基础上进行如下的改进。

2.1 粒子群运行机制分析

粒子群算法中粒子通过跟踪两个"极值"来更新自己的位置,算法结构简单,运行速度较快。但是,基本的粒子群算法在解空间搜索时,有时会出现粒子在最优解的附近来回"振荡"现象,调整学习因子和惯性权重因子也无法完全避免这种现象,而且这个最优解可能就是局部最优解。作者对这种现象作了分析,并且通过实验发现:传统的粒子群算法中粒子在位置更新时,每次粒子的飞行时间都保持不变,固定为1,这是导致产生"振荡"现象的一个重要因素。因为在进化初期,粒子距离最优位置较远,粒子的飞行时间应长一些,这样有利于更快的移向最优位置;当粒子距离最优位置较近时,粒子的飞行时间应短一些,这样可以避免因飞行时间过长而导致的粒子"飞过"最优位置从而产生的"振荡"现象。而传统的粒子群算法在进化初期和后期都固定飞行时间为1,将导致粒子群在进化后期搜索性能下降。如式(2)实际上就是:

$$x_{is}^{t+1} = x_{is}^t + v_{is}^{t+1} * T \tag{5}$$

上式中T表示第i个粒子的飞行时间,在传统的粒子群算法中$T = 1$。

从现实中的鸟群觅食行为来看,观察者不难发现:鸟群在从一个位置飞向下一个目标时,并不是每次都保持相同的飞行时间的。在实际中,鸟群在做位置迁移时不单单是速度在不断变化,而且它们的飞行时间(对每个速度而言)也在不断变化。所以,本文提出的飞行时间自适应调整的粒子群算法(FAA-PSO)是符合生物进化机制的。

2.2 粒子飞行时间调整方案

鉴于粒子在进化过程中,其飞行时间应不断变化。作者已经通过实验验证飞行时间随迭代代数的增加而增长并不能改善粒子群算法的搜索性能,这在实际中也不符合鸟群觅食行为机制。既然飞行时间随迭代代数的增加而缩短能改善粒子群算法的搜索性能,那么粒子的飞行时间应该以何种方式随迭代代数的增加而缩短才能最大限度的提高粒子群算法的性能,本文在参考粒子群算法的研究者们在设计惯性权重以及其他参数变化机制[6,7,9]的基础上,提出了以下几种备选方案:

方案1:

$$T = T_0 [1 - k \frac{iter}{I_{\max}}] \tag{6}$$

其中:T_0为最长飞行时间,$iter$为当前进化代数,$I_{\max}$为最大进化代数,k为比例系数,取常数。

方案2:

$$T = T_0 [1 + k\cos(\frac{iter}{I_{\max}}\pi)] \tag{7}$$

参数意义同上,T先增大后减小,和方案1、3不同(T不断减小)。

方案3:

$$T = T_0 [1 - k(\frac{n-1}{n})^{iter}] \tag{8}$$

其中:T_0为最长飞行时间,$iter$为当前进化代数,n为一个常数,k为比例系数。

以上方案中,k起调节作用。因为在很多情况下粒子不需要进化$I_{\max}$次就可收敛,这样可能$I_{\max}$比$iter$的最大值大得多,k在$I_{\max}$和$iter$之间起平衡作用。

2.3 FAA-PSO 算法步骤

1)初始化一个规模为N的粒子群,设定初始位置和速度。

2)计算每个粒子的适应值。

3)对每个粒子将其适应值和其经历过的最好位置P_{is}的适应值进行比较,若较好,则将其作为当前的最好位置。

4)对每个粒子将其适应值和全局经历过的最好位置P_{gs}的适应值进行比较,若较好,则将其作为当前的全局最好位置。

5)根据式(3),(5)分别对粒子的速度和位置进行进化,T按上述的各方案进行自适应变化。

6)如果满足终止条件,则输出解;否则返回2)。

2.4 必要说明

虽然本文的算法中粒子飞行时间T和文献[7]、[10]中的压缩因子α都乘在粒子更新位置公式中的速度项上,然而两者有本质的不同:

首先,意义不同。T表示粒子飞行时间,在物理意义上来说,速度乘以时间才等于距离,即粒子当前位置等于先前位置加上它移动的距离;而α无此意义。

其次,取值不同。粒子飞行时间T随迭代代数的增加自适应的减小;而压缩因子α在学习因子和惯性权重固定后就是确定值了[7]。

最后,优化效果不同。在后面的仿真实验中可以看出。

3 数值结果

为验证本文算法的性能,取文献[7]中四个常用基准函数做测试,并和文献[7]结果比较。在此选取的参数和文献

[7]一样。学习因子 $c_1 = c_2 = 2$,ω 从 1.0 减小到 0.4,群体规模数为 30,最大进化代数为 10000。方案 1、2 中 T_0 分别为 0.9,0.1;k 分别为 4,3;方案 3 中取 T_0,n 和 k 分别为 1,100,2 。用 VC++6.0 编程,算法运行 20 次,取平均值。计算结果见表 2,惯性权重法,压缩因子法,改进的压缩因子法的计算结果引自文献[7]。

表 2 中"平均"和"范围"分别表示 20 次运行中收敛情况下进化代数的平均值和进化代数的范围。例如:若 20 次中有 2 次不收敛,则"平均"和"范围"分别表示 18 次收敛情况下进化代数的平均值和进化代数的范围。由表 2 可以看出,FAA-PSO 算法各方案比之惯性权重法优势明显,在对 f_2 和 f_3 的优化结果上,方案 1、2 比其他算法收敛较快,运行稳定;对 f_1 和 f_4 的优化结果上,FAA-PSO 算法各方案比之压缩因子法和改进的压缩因子法收敛慢。但是 FAA-PSO 算法各方案的收敛次数明显较多,比压缩因子法更稳定。就 FAA-PSO 算法各方案的优化效果而言,方案 2 较好,方案 3 较差。

表 1　测试函数

函数名	函数表达式	维数	变量范围	最优值	目标值
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0	0.01
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$[-30,30]^n$	0	100
Rastrigrin	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12,5.12]^n$	0	100
Griewank	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600,600]^n$	0	0.1

表 2　各种 PSO 算法对四个测试函数优化结果

	函数	惯性权重	压缩因子	改进压缩因子	FAA-PSO(1)	FAA-PSO(2)	FAA-PSO(3)
$f_1(x)$	平均	1537.8	552.05	529.65	965.900000	857.250000	2974.052632
	范围	1485 - 1615	503 - 599	495 - 573	707 - 1306	646 - 1643	2608 - 3981
	收敛	20 次	20 次	20 次	20 次	20 次	19 次
$f_2(x)$	平均	3517.35	1424.1	992.0	761.350000	618.500000	2212.800000
	范围	2866 - 4506	475 - 4793	402 - 1394	581 - 1268	362 - 1530	1928 - 2371
	收敛	20 次	20 次	20 次	20 次	20 次	20 次
$f_3(x)$	平均	1320.9	6823.0	213.45	503.578947	219.421053	1215.050000
	范围	743 - 1704	233 - 7056	161 - 336	293 - 1013	103 - 980	186 - 1950
	收敛	20 次	19 次	19 次	19 次	19 次	20 次
$f_4(x)$	平均	2900.5	437.0	312.6	1088.900000	1311.100000	1677.250000
	范围	2556 - 3891	384 - 663	282 - 366	1082 - 1143	1179 - 1438	1590 - 1838
	收敛	20 次	17 次	19 次	20 次	20 次	20 次

4　结语

本文针对传统 PSO 算法的不足,对其进行了改进,提出了粒子飞行时间自适应调整的 PSO 算法,数值结果表明,该算法比之惯性权重法优势明显,和压缩因子法性能相当,但机制不同。作为一种新的思想有待进一步研究。另外,文中各方案参数如何影响算法性能也有待深入探讨。另外,本算法也可以和其他算法混合构造混合算法。

参考文献:

[1] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [J]. In: Proc of the 6th Int'l Symposium On Micro Machine and Human Science. Piscataway . NJ: IEEE Service Center 1995, 39 - 43.

[2] KENNEDY J, EBERHART R. Particle Swarm Optimization [A]. Proc IEEE Int Conf Neural Networks[C]. Piscataway: IEEE Press, 1995. 1942 - 1948.

[3] PARSOPOULOS RE, VRAHATIS MN. On Computation of all Global Minimizes through Particle Swarm Optimization [J]. IEEE, Transactions on Evolutionary Computation, 2004, 8(3): 211 - 223.

[4] VOSS MS, FENG X. ARMA Model Selection Using Particle Swarm Optimization and AIC Criteria [A]. 15th Triennial World Congress [C]. Barcelona, Spain: IFAC, 2002.

[5] VAN DEN BERGH F, ENGELBRECHT AP. Cooperative Learning In Neural Networks Using Particle Swarm Optimization [J]. South African Computer Journal, 2000, (11) : 84 - 90.

[6] SHI Y, EBERHART R. A Modified Particle Swarm Optimizer[A]. In: Proceedings of the IEEE International Conference on Evolutionary Computation[C]. Piscataway, NJ: IEEE Press, 1998. 69 - 73.

[7] EBERHART R, SHI YH. Comparing Intertia Weights and Constriction Factors In Particle Swarm Optimization[A]. Proc 2000 Congress on Evolutionary Computation[C]. IEEE Press, 2000. 84 - 88.

[8] LOVBJERG M , RASMUSSEN TK , KRINK T . Hybrid Particle Swarm Optimizer with Breeding and Subpopulation [A]. Proc of the 3rd Genetic and Evolutionary Computation Conference[C]. Sanfrancisco, 2001. 469 - 476.

[9] 陈国初,俞金寿. 增强型微粒群优化算法及其在软测量中的应用[J]. 控制与决策, 2005, 20(4): 377 - 381.

[10] CLERC M. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization[J]. In Proc. CEC 1999. 1999: 1951 - 1957.