

# 基于函数最优解问题的粒子群算法改进

王莉荣, 祁云嵩

(江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003)

**摘要:**通过对粒子群算法的深入研究,鉴于其具有容易陷入局部极值、迭代后期收敛速度慢、精度低等情况,众多学者对其作出改进,并都已成功应用到各种实际问题中。为了改善粒子群算法性能,能够快速准确地求解出函数的最优解,文中在基于粒子群最优算法及其改进算法研究的基础上,结合时变权重与压缩因子,对粒子群算法进行改进,并将改进算法应用于求解函数最优解问题中。实验表明,该方法具有了带时变权重或带压缩因子算法的优点,同时加快了函数的收敛速度,提高了最优解的准确度,通过参数调整,性能得到了有效改善。

**关键词:**粒子群算法;压缩因子;时变权重

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2013)02-0049-03

doi:10.3969/j.issn.1673-2013.02.012

## Partical Swarm Optimization Improvement Based on Optimal Solution to Functions

WANG Li-rong, QI Yun-song

(School of Computer Science and Eng., Jiangsu Univ. of Science and Techn., Zhenjiang 212003, China)

**Abstract:** A depth-research found that partical swarm algorithm is easy to fall into local minima, iterative post-slow rate of convergence, low accuracy and so on, then many scholars have made improvements and successfully applied to a variety of practical problems. In order to improve the performance of partical swarm optimization, it works over the particle swarm optimization and the proved methods in order to solve the function of the optimal solution fastly and accurately. And it combines the time-varying weights with the constriction factor to improve the partical swarm optimization. Then use the method to solve the optimal solution of functions. Experiments show that the method has the advantages with the band time-varying weights or with a compression factor algorithm, at the same time makes the convergence faster, improves the accuracy of the optimal solution of the function, and improve the performance by adjusting parameters.

**Key words:** partical swarm optimization algorithm; compression factor; time-varying weights

## 0 引言

粒子群算法(Partical swarm optimization)是1995年由Eberhart和Kennedy<sup>[1-5]</sup>提出的一种新的群体智能(swarm intelligence)算法,也是一种非常有效并被广泛应用的迭代优化算法,源于对鸟群、鱼群捕食行为的研究。最初粒子群优化算法是用来处理连续优化问题的,逐渐应用到组合优化问题中。由于其算法简单,易实现,可调参数少,21世纪以来国内外关于粒子群算法出现很多新的研究成果,对于PSO本身的研究很多学者认为惯性权重的选择是相当重要的。

Shi Y提出了线性递减权重策略,通常迭代初期 $\omega$

取值较大,使得粒子具有较大的速度,从而使粒子具有较强的全局探索能力,后期 $\omega$ 取值较小,粒子具有较小的速度,使得粒子具有较强的搜索能力。整个过程 $\omega$ 随着迭代的进行逐渐减小,但是若 $\omega$ 取值过大则局部搜索能力太差,容易错过全局最优点;而后期 $\omega$ 取值过小全局探索能力太差,容易陷入局部最优的极值<sup>[6-9]</sup>。

随后Clerc提出在PSO中加入压缩因子的概念,用于确保粒子群最优优化算法能够收敛, $K$ 的取值对参数 $c_1, c_2$ 的依赖比较大,使得收敛速度不容易控制。

文中基于基本粒子群算法,结合线性递减权重和压缩因子对参数 $\omega$ 进行了适当的改进,使其能够在进化过程中加快收敛速度,提高准确度;文中利用具有代表性的5个测试函数对改进算法进行测试,并且根据文献给定的参数求解测试函数的最优解。实验结果表明,文中所提出的改进算法在求解最优解过程中收敛速度、精度以及稳定性方面都有优势。

收稿日期:2012-05-29;修回日期:2012-09-02

基金项目:国家自然科学基金资助项目(61100116)

作者简介:王莉荣(1985-),女,硕士生,研究方向为模式识别与智能系统;祁云嵩,副教授,硕士研究生导师,研究方向为生物信息学、图像分割、智能信息处理。

## 1 粒子群算法简介

粒子群优化算法是一种新的全局优化进化算法,最初是由模拟简单个体组成的群落与环境以及个体之间的互动行为。这一新算法涉及到以下常用概念<sup>[10]</sup>。

### 1.1 粒子群算法的基本概念

(1) 粒子类似于遗传算法中的染色体, PSO 中的粒子为基本的组成单位,代表解空间的一个候选解。

(2) 粒子种群由  $m$  个粒子组成,代表  $m$  个候选解,经过  $t$  次迭代产生种群  $pop(t) = [X_1(t), X_2(t), \dots, X_m(t)]$ 。其中,  $X_i(t)$  为种群中的第  $i$  个粒子。

(3) 粒子速度表示粒子在单位迭代次数位置的变化,即代表解变量的粒子在  $d$  维空间的位移。

(4) 适应函数由优化目标决定,用于评价粒子的搜索性能,指导粒子种群的搜索过程。

(5) 个体极值  $P_{id}$  是单个粒子到目前为止对应的适应值最优解。

(6) 全局极值  $P_{gd}$  是整个种群到目前为止对应的适应值最优解。

### 1.2 基本粒子群算法原理

PSO 算法是一种有效的启发式全局优化算法,仿真简化了的社会行为,人工生命和演化计算理论是 PSO 算法的思想来源。Reynolds 通过对鸟群飞行行为的研究发现,在飞行过程中,鸟仅跟随相邻的有限数量的鸟飞行,但最终整个种群是朝着同一目标行进的。PSO 算法受鸟群捕食行为的启发,即鸟群在只有一块食物的环境中随机地搜食物,那么找到食物的最优策略就是搜寻离食物最近的鸟的周围。在这一算法中将鸟视为一个质点,称为“粒子”,在搜索过程中,每个粒子都有自己特定的位置和速度,还有一个由目标函数决定的适应值。粒子在空间中以一定的速度飞行,其速度由粒子的飞行经验和群体飞行经验动态调整,在整个飞行过程中保留自己发现的最好位置<sup>[11~16]</sup>。

假设在一个  $d$  维的搜索空间中,初始化  $m$  为种群规模,其中,在  $t$  时刻第  $i$  个粒子的位置表示为:  $X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{id}(t))$ , 相应的飞行速度为:  $V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{id}(t))$ 。

在执行算法之前先初始化粒子的位置和速度,在迭代过程中,每一次粒子都通过跟踪两个极值来更新自己的速度和位置:一个是粒子迄今为止搜索到的最优解,称为个体极值,表示为:  $P_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{id}(t))$ ;另一个是群体到目前为止搜索到的最优解,称为全局极值,表示为:  $P_g(t) = (p_{g1}(t), p_{g2}(t), \dots, p_{gd}(t))$ 。

该算法的更新公式如下:

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1)$$

$$v_{ik}(t+1) = v_{ik}(t) + c_1 r_1 (p_{ik}(t) - x_{ik}(t)) +$$

$$c_2 r_2 (p_{gk}(t) - x_{ik}(t))$$

$c_1, c_2$  为两个学习因子,分别调节向个体和全局最优解方向飞行的最大步长,若太小,粒子会远离目标,若太大,粒子会突然飞向目标区域或飞过目标区域。合适的  $c_1, c_2$  可以加快收敛速度并且不易陷入局部最优,通常令  $c_1 = c_2 = 2$ ;  $r_1, r_2$  是在  $[0, 1]$  区间的随机数;  $i = 1, 2, \dots, m; k = 1, 2, \dots, d$ 。为防止粒子偏离搜索区域,粒子的速度都限制在  $[-vmax, +vmax]$ 。在速度更新公式中,第一部分反应粒子自身的飞行惯性;第二部分理解为是粒子的“认知”行为,跟随自己的最优位置进行搜索;第三部分就是粒子的“社会”行为,反应了粒子间信息的传递。

### 1.3 算法流程

(1) 初始化。种群粒子的位置和速度都是在允许范围内随机初始化的,将  $pbest$  设置为当前位置,同时计算其个体极值,取个体极值中最好的作为全局极值,并记录其粒子序号,并将  $pgbest$  设置为该最好粒子的当前位置。

(2) 评价粒子。计算每个粒子的适应度值,与当前个体极值做比较,将  $pbest$  设置为更好的值,并更新个体极值。同理更新全局极值。

(3) 更新粒子。根据粒子的速度和位置公式进行更新。

(4) 迭代。如果达到预先设置好的结束条件则停止迭代,输出最优解,否则返回到第二步。

PSO 流程图如图 1 所示。

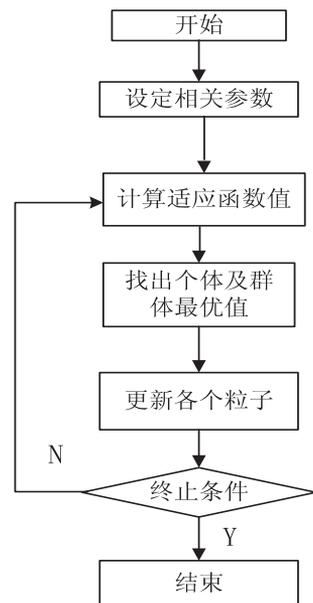


图 1 PSO 流程图

### 1.4 改进的粒子群算法

粒子群算法由于其算法简单容易实现,可调参数比较少,已经得到了广泛的研究和应用。到目前为止衍生出来的变形也很多,常见的两种变形是线性递减

权重与压缩因子。线性递减权重是由 Y. Shi 和 Eberhart 在 1998 年提出的,合适的权重  $\omega$  数值可以提供局部和全局一个平衡的探索及开发能力,遵循线性地从大到小地进行递减,当  $\omega$  较大时探索较大的区域,较快地定位最优解的大概位置,随着  $\omega$  的减小,可以比较精细地搜索目标区域,提高了 PSO 算法的性能。Clerc 于 1999 年提出了压缩因子的概念,这一方法确保了最优算法能够随时间而收敛。鉴于前辈的改进方法,文中提出非线性权重因子,合理地结合了线性递减权重和压缩因子,既保证算法收敛,又提高了算法的收敛速度和性能,其速度更新公式为:

$$v_{ik}(t + 1) = \omega v_{ik}(t) + c_1 r_1 (p_{ik}(t) - x_{ik}(t)) + c_2 r_2 (p_{gk}(t) - x_{ik}(t))$$

式中权重

$$\omega = (\omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{k_{\max}} * k) *$$

$$\frac{2}{|\psi + \sqrt{\psi^2 - 4 * \psi} - 2|}, \psi = c_1 + c_2, \psi > 4$$

这种算法调整相当于是在原始算法的基础上对速度进行了一个动态的非线性调整。

## 2 算法的实验验证

惯性权重是粒子群算法中一个重要的参数,常用的两种权重形式:固定权重和时变权重,本节通过实验比较来说明现有权重形式与非线性权重对算法性能的改进效果。实验使用 5 个常用的测试函数(见表 1<sup>[17]</sup>)作为实例,先从不同形式的权重对算法优化效果的影响来分析,再从非线性权重在不同变化区间内对优化效果的影响来研究。

表 1 5 个常用测试函数

函数编号	函数名	表达式	维数
1	Schaffer's	$f(x) = 0.5 - \frac{(\sin(\sqrt{x_1^2 + x_2^2}) - 0.5)}{(1 + 0.001 * (x_1^2 + x_2^2))^2}$	2
2	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	30
3	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} (100 * (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30
4	Rastrigrin	$f(x) = \sum_{i=1}^n (x_i^2 - 10 * \cos(2 * \pi * x_i) + 10)$	30
5	Griewank	$f(x) = \frac{1}{4000} * \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30

### 2.1 权重对算法性能的影响

参数设计:参数 1<sup>[18]</sup>(固定权重)  $\omega = 0.6, c_1 = c_2 = 1.7$ ;参数 2<sup>[19]</sup>(时变权重)范围 0.9~0.2,递减率为 0.9-0.2/100,  $c_1 = c_2 = 2$ ;参数 3(非线性权重)范围 0.9~0.2,递减率为 0.9-0.2/80,  $c_1 = c_2 = 2$ ;种群大小 10,最大迭代次数 100,实验结果如表 2 所示:

表 2 不同形式权重的优化效果

参数	指标	函数 1	函数 2	函数 3	函数 4	函数 5
1	平均最优解	0.2319	1.1259	0.4825 * 105	1.2399	1.0032
	运行时间(s)	135.8	37.3	42.5	44.1	43.5
2	平均最优解	0.1598	1.0229	0.3689 * 105	1.2596	0.7192
	运行时间(s)	98.7	29.6	31.5	35.6	34.8
3	平均最优解	0.0982	0.9748	0.1496 * 105	0.8671	0.3927
	运行时间(s)	111.7	23.8	25.6	31.2	31.8

由表 2 可以看出,非线性权重与时变权重,固定权重相比,最优解的值更加准确,只是收敛速度没有明显的提高,固定权重和时变权重的参数是文章推荐的,经过了详细准确的选择,非线性权重经过简单的参数调整,收敛速度就能得到明显的提高,同时算法性能能够有效改善。

### 2.2 时变部分区间变化对优化效果的影响

使用函数

$$3 * (1 - x) * \sqrt{2} * \exp(-(x * \sqrt{2}) - (y + 1) * \sqrt{2}) - 10 * (x/5 - x * \sqrt{3} - y * \sqrt{5}) * \exp(-(x * \sqrt{2} - y * \sqrt{2}) - \frac{1}{3} * \exp(-(x + 1) * \sqrt{2} - y * \sqrt{2}))$$

作为观察对象来研究时变部分区间的变化对优化效果的影响,设置种群大小为 30,每次运行的最大迭代次数为 400,实验结果如表 3:

表 3 不同变化区间的非线性权重的优化效果

时变范围	最优解	运行时间/s
0.7~0.2	3.8935	66.2
0.9~0.2	2.5153	103.1
0.9~0.7	7.7840	72.8

由表 3 实验数据可知,测试函数在较大的惯性权重范围下取得较好的优化效果(说明对于这个函数寻找最优解需要较强的探索能力,即全局寻优能力),尤其在 0.9~0.7 变化范围内,最优解的值最准确,而且收敛速度较快。

## 3 结束语

在粒子群算法中,惯性权重的选择相当重要,适当的选择可以有效地提升优化效果。文中提出了非线性权重粒子群算法,该算法结合线性递减权重和压缩因子,有效地加快了算法的收敛速度,提高了算法的收敛精度,具有较强的寻优能力,继承了标准粒子群算法的优化性能,并且合适的选择使粒子具有较强的开发能力和探索能力。

### 参考文献:

[1] 焉为家,郭雨珍.改进的粒子群算法求解蛋白质结构预测问题[J].计算机技术与发展,2011,21(12):109-112.

高效率,减小冗余也是研究重点<sup>[17]</sup>。

参考文献:

[1] 张巍巍. 虚拟环境中阴影的实时绘制算法研究[D]. 郑州: 解放军信息工程大学, 2009.

[2] Cass E, Kilgard M. Practical and Robust Stenciled Shadow Volumes for Hardware-accelerated Rendering[R]. [s. l.]: NVIDIA Corporation, 2002.

[3] McGuire M. Efficient Shadow Volume Rendering[M]. [s. l.]: GPU Gems, 2004; 137-166.

[4] McGuire M, Hughes J F, Egan K, et al. Fast, Practical and Robust Shadows[R]. [s. l.]: Brown Univ. , 2003.

[5] Williams L. Casting Curved Shadows on Curved Surfaces [J]. Computer Graphics, 1987, 12(3): 270-274.

[6] 过洁, 许晓, 潘金贵. 基于阴影图的阴影生成算法研究现状[J]. 计算机辅助设计与图形学学报, 2010(4): 579-591.

[7] Stamminger M, Drettakis G. Perspective Shadow Maps[J]. ACM Transactions on Graphics, 2002, 21(3): 557-562.

[8] Wimmer M, Scherzer D, Purgathofer W. Light Space Perspective Shadow Maps[C]//Proceedings of the Eurographics Symposium on Rendering. [s. l.]: [s. n.], 2004; 143-152.

[9] Lloyd B, Tuft D, Yoon Sung-Eui, et al. Warping and Partitioning for Low Error Shadow Maps[C]//Proceedings of the Eurographics Symposium on Rendering. [s. l.]: [s. n.],

2006; 215-226.

[10] Zhang Fan, Sun Hanqiu, Xu Leilei, et al. Parallel-Split Shadow Maps for Large-Scale Virtual Environments[C]//Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications. [s. l.]: [s. n.], 2006; 311-318.

[11] Zhang Fan, Sun Hanqiu, Xu Leilei, et al. Hardware-accelerated Parallel-split Shadow Maps[J]. International Journal of Image and Graphics, 2008, 8(2): 223-241.

[12] Nguyen H. GPU 精粹 3[M]. 北京: 清华大学出版社, 2010; 164-177.

[13] Reeves W, Salesin D, Cook R. Rendering Antialiased Shadows with Depth Maps[J]. Computer Graphics, 1987, 21(3): 283-291.

[14] Brabec S, Annen T, Seidel H P. Practical Shadow Mapping [J]. Journal of Graphical Tools, 2002, 7(4): 9-18.

[15] Donnelly W, Lauritzen A. Variance Shadow Maps[C]//Proceedings of the Symposium on Interactive 3D Graphics and Games. [s. l.]: [s. n.], 2006; 161-165.

[16] Annen T, Zhao D, Mertens T, et al. Real-time all-frequency shadows in dynamic scenes [J]. ACM Transactions on Graphics, 2008, 27(3): 341-348.

[17] Liang X H, Ma S, Cen L X, et al. Light Space Cascaded Shadow Maps Algorithm for Real Time Rendering[J]. Journal of computer science and technology, 2011, 26(1): 176-186.

(上接第 51 页)

[2] 邓璐娟, 卢化琦, 孙义坤, 等. 改进的粒子群算法在测试数据生成中的应用[J]. 计算机技术与发展, 2010, 20(7): 216-218.

[3] 宋晓琳, 李红, 郭孔辉. 基于改进的粒子群优化算法的轮胎参数辨识[J]. 科技导报, 2011, 29(9): 53-56.

[4] Yang Kai, Zhao Zhiqin, Nie Zaiping. Optimization of Unequally Spaced Antenna Arrays Using Fuzzy Discrete Partical Swarm Algorithm [J]. Journal of University of Electronic Science and Technology of China, 2012, 41(1): 1-6.

[5] Wen Guangrui, Zhang Xinning, Zhao Ming. Application of Partical Swarm Optimization Algorithm in Field Holo-balancing[M]. Berlin Heidelberg: Springer-Verlag, 2010.

[6] 王丽, 王晓凯. 一种非线性改变惯性权重的粒子群算法[J]. 计算机工程与应用, 2007, 43(4): 47-48.

[7] 刘长良, 高亚龙. 带压缩因子的粒子群算法在汽包压力控制系统中的应用[J]. 计算机系统应用, 2012, 21(1): 13-16.

[8] 贾冀婷. 基于粒子群算法的测试用例自动生成方法研究[J]. 计算机技术与发展, 2010, 20(9): 24-27.

[9] 苏守宝, 汪继文, 方杰. 粒子群优化技术的研究与应用进展[J]. 计算机技术与发展, 2007, 17(5): 249-252.

[10] 黄席樾, 向长城, 殷礼胜. 现代智能算法理论及应用[M]. 北京: 科学出版社, 2009; 180-247.

[11] 张国印, 程慧杰, 刘咏梅, 等. 一种新算法在基因表达谱聚类中的应用[J]. 计算机工程与应用, 2009, 45(36): 216-218.

[12] 贾瑞玉, 黄义堂, 邢猛. 一种动态改变权值的简化粒子群算法[J]. 计算机技术与发展, 2009, 19(2): 137-139.

[13] 田野. 粒子群优化算法及其应用研究[D]. 长春: 吉林大学, 2010.

[14] 王伯成, 施锦丹, 王凯. 粒子群优化算法的研究现状与发展概述[J]. 电视技术, 2008, 48(5): 7-11.

[15] 朱玉平. 一种改进粒子群优化算法[J]. 计算机技术与发展, 2008, 18(11): 106-108.

[16] 王俊伟, 汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报, 2005, 20(2): 194-198.

[17] 徐青鹤. 改进粒子群算法及其应用研究[D]. 杭州: 杭州电子科技大学, 2009.

[18] Trelea I. The partical swarm optimization algorithm: Convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 85(6): 317-325.

[19] Eberhart R, Shi Y. Comparing Inertia Weights and Constriction Factors in Partical Swarm Optimization [C]//IEEE Congress on Evolutionary Computation. Piscaway: IEEE Service Center, 2000; 84-88.

## 基于函数最优解问题的粒子群算法改进

作者: 王莉荣, 祁云嵩  
作者单位: 江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003  
刊名: 计算机技术与发展  
英文刊名: Computer Technology and Development  
年, 卷(期): 2013 (2)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjz201302014.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjz201302014.aspx)