

# 一种动态改变权值的简化粒子群算法

贾瑞玉, 黄义堂, 邢 猛

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039;

安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

**摘 要:**基本粒子群优化算法(bPSO)具有容易陷入局部极值、进化后期收敛速度慢、精度低等缺陷,而舍弃了速度项的简化粒子群算法(sPSO)在保证收敛速度和精度的同时使算法更加简练。文中提出了一种动态改变权值的简化粒子群算法。并经实验证明,该算法在搜优精度和收敛速度上具有明显的优势。

**关键词:**粒子群算法;简化粒子群算法;惯性权值

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2009)02-0137-03

## A Modified Simple Particle Swarm Optimization Using Dynamically Decreasing Inertia Weight

JIA Rui-yu, HUANG Yi-tang, XING Meng

(Ministry of Edu. Key Lab. of Intelligent Computing & Signal Processing, Anhui Univ., Hefei 230039, China;

School of Computer Science and Technology of Anhui University, Hefei 230039, China)

**Abstract:** The basic particle swarm optimization (bPSO) has some demerits, such as relapsing into local extremum, slow convergence velocity and low convergence precision in the late evolutionary. The simple PSO discards the particle velocity and improves extraordinarily the convergence velocity and precision in the evolutionary optimization, and it looks more legible and terse. A modification to dynamically decreasing inertia weight strategy in this article is presented. It is demonstrated that there are evident superiorities in computational precision, searching speed and steady convergence.

**Key words:** particle swarm optimization; simple PSO; inertia weight

### 0 引 言

粒子群算法(Particle Swarm Optimization, PSO)是美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 于 1995 年提出的一种演化计算技术<sup>[1]</sup>。它源于对鸟群和鱼群群体觅食运动行为的模拟,是一种全局优化进化算法,最初用于处理连续优化问题,目前已可以用于解决大量非线性、不可微和多峰值的复杂问题优化,并已广泛用于科学和工程领域,如函数优化、神经网络训练、模式分类和模糊系统控制等领域<sup>[2]</sup>。

同遗传算法类似,粒子群优化算法也有群体和适应度这两个概念。粒子群的个体代表问题的一个可能解。每个粒子有位置与速度两个概念来描述其当前状

态,粒子位置坐标对应的目标函数值即为该粒子的适应度,PSO 算法通过适应度来衡量粒子的优劣性。但是,PSO 不是通过遗传算子进化而是通过个体间协作与竞争来寻找最优解。在进化初期,PSO 收敛速度快,运算简单,易于实现。

然而,粒子群优化算法是根据全体粒子和自身粒子的搜索经验向最优解的方向发展,在进化后期收敛速度明显变慢,同时,算法收敛精度不高。针对这些问题,学者们提出了不同的优化策略。SHI Y<sup>[3,4]</sup>提出了线性递减权策略,文献[4]采用模糊规则策略动态地调整的值来视算法自适应地调整全局系数,这种方法同时兼顾了搜索效率和搜索精度,但是对于许多复杂的非线性优化问题,通过改变一个全局系数提高搜索精度其效果是极其有限的。吕振肃等人<sup>[5]</sup>提出了协同 PSO 使得粒子更容易跳出局部最小点,并达到了较高的收敛精度,但是其出现了“启动延迟现象”,在迭代初期收敛速度较慢。文中参考了 SHI Y<sup>[3]</sup>的思路,在简化 PSO<sup>[6]</sup>的基础上,利用余弦函数的性质,使惯性权值

收稿日期:2008-05-21

基金项目:安徽省自然科学基金项目(kj2008B092)

作者简介:贾瑞玉(1965-),女,副教授,研究方向为计算机图形学、数据挖掘、人工智能。

随迭代次数增加而非线性地减少,经优化领域内具代表性的 3 个测试函数分别对改进前后的 PSO 进行试验,结果证明,所提出的改进算法在搜优精度和收敛速度上与基本 PSO 及简化 PSO 相比都具有极其明显的进步。

## 1 基本粒子群算法

bPSO 算法首先初始化一群随机的粒子,然后通过迭代来最终找到最优解。每一次迭代过程中,粒子通过个体极值和全局极值来更新自己的速度与位置。在  $D$  维目标搜索空间中,由  $m$  个种群构成粒子群体,其中,第  $i$  个粒子在第  $d$  维的位置为  $x_{id}$ ,其速度为  $v_{id}$ ,该粒子搜索到的最优位置为  $p_{id}$ (个体极值),而粒子群当前的最优位置为  $p_{gd}$ (全局极值),James Kennedy 和 Russell Eberhart 最早提出的 bPSO 算法公式为:

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

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

其中,  $i = 1, 2, \dots, m; d = 1, 2, \dots, D; r_1, r_2$  为服从  $U(0, 1)$  分布的随机数;学习因子  $c_1$  和  $c_2$  为非负常数,它们使粒子向  $p_{id}$  和  $p_{gd}$  加速运动,故也称加速常数,通常情况取  $c_1 = c_2 = 2$ (文中在试验中选取 1.4962 为多次试验的经验值),  $v_{id} \in [-v_{\max}, v_{\max}]$ ,  $v_{\max}$  为用户设定的常数。迭代的终止条件为最大迭代次数或者最小适应度阈值,这些视具体情况而定。

而 SHI Y<sup>[3]</sup> 等人通过添加惯性系数  $\omega$  来增强粒子跳出局部极值的能力,其修改后的公式为:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (3)$$

而 Clerc<sup>[7]</sup> 等人则对公式(3) 添加了约束因子  $\alpha$ , 其修改后的公式为:

$$x_{id}^{t+1} = x_{id}^t + \alpha v_{id}^{t+1} \quad (4)$$

目前很多学者直接将公式(3)、(4) 视为 bPSO 算法,笔者在后面的试验比较中也是这样认定的。

## 2 一种改进的简化粒子群算法

文献[4]提出了线性递减权策略,即

$$\omega(t) = \frac{(\omega_i - \omega_e)(T_{\max} - t)}{T_{\max}} + \omega_e \quad (5)$$

其中  $t$  是当前迭代次数,  $T_{\max}$  为最大进化代数,  $\omega_i$  为惯性权初值,  $\omega_e$  为进化至最大代数时权值。通常在初期权值较大,而在后期较小。如果  $\omega_i$  过大则局部搜优能力较差,易导致错过全局最优,后期  $\omega_e$  较小则易导致全局搜优能力差而陷入局部极值。文献[8] 和文献[9] 也分别提出了改变权值的算法。

目前, bPSO 及其改进算法大都基于粒子的“位置”和“速度”两个关键概念,因而在其改进方案中的方程中都包含位置和速度两个变量,而采取了遗传算法策略的算法中,其描述更加复杂,胡旺<sup>[6]</sup> 通过证明 bPSO 进化过程与粒子速度无关,并且证明了舍弃了速度的进化方程的收敛性,提出了简化粒子群算法,大刀阔斧地舍弃了粒子速度这个参数,从而避免了由粒子速度引起得粒子发散使得后期收敛速度变慢及精度低的缺陷。改进后的算法方程更加简练清晰,即

$$x_{id}^{t+1} = \omega x_{id}^t + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (6)$$

并且避免了人为确定的参数  $v_{\max}$  对粒子收敛速度和收敛精度的影响。基于以上的分析,文中的改进策略就是在简化粒子群算法的基础上对权值  $\omega$  采取动态调整的办法:

$$\omega(t) = \omega_i + \cos(2(t-1)/T_{\max}) \quad (7)$$

其中  $\omega_i$  为权初值,  $t$  为当前迭代次数,  $T_{\max}$  为最大进化代数;  $\cos(2 \frac{(t-1)}{T_{\max}})$  中 2 只是为了保证余弦函数中变量值  $\in [0, \pi]$ , 以使  $\omega(t)$  随  $t$  增大递减, 故可以根据具体试验情况从区间  $(0, \pi]$  中选取。迭代初期,  $\omega(t)$  较大, 粒子较快地搜索整个空间来确定最优值, 而随着迭代次数的增加,  $\omega(t)$  非线性地减小, 将搜索范围逐渐缩小, 直至迭代末期最终确定最优值。

## 3 实验及结果分析

选取了 3 个常用于优化算法比较的基准函数(见表 1)使用 matlab7.0 工具在 CPU 为 SP2400+, 内存 512M 的机器上进行试验。

表 1 3 个测试函数

函数名称	函数	维数	取值范围
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	10	$[-100, 100]^{30}$
Griewank	$f_2(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	10	$[-600, 600]^{30}$
Rastrigin	$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	10	$[-100, 100]^{30}$

这些函数的理论最小值均为 0, 其中 Sphere 函数为单峰函数, 在  $x_i = 0$  时取得极小值; Griewank 函数和 Rastrigin 函数则均有多个局部极小点。

以下实验均给定  $c_1 = c_2 = 1.4962$ , 粒子数为 40, 为方便比较, 各种实验结果均取其中 3 个结果列于表中, 表中实验结果均精确到小数点后 4 位。

3.1 Sphere 及 Griewank 函数测试

由表 2、表 3 可以看出,不论是 Sphere 函数还是 Griewank 函数,sPSO 算法和文中的改进算法均在收敛速度和精度上大大优于 bPSO,然而 sPSO 算法更依赖于权初值的选取,当选取较好的权初值(0.7298,为多次实验得出的经验值)时,得出的结果较好,而当任意选取权初值时(取 1 时),结果并不理想,相比之下,文中的改进算法在任意选取权初值时也能得出较好结果,而选取较好的权初值时,实验结果和 sPSO 相似。

表 2 Sphere 函数测试结果比较

Sphere 函数	$\omega_i$ (权初值)	最大迭代次数:50	最大迭代次数:100	最大迭代次数:500
b PSO (基本粒子群算法)	1	0.6058	0.3179	0.2697
		0.7763	0.2502	0.6889
		0.9026	0.6517	0.3456
	0.7298	0.0094	2.9814e-5	9.3799e-7
		0.0100	1.2365e-4	3.2429e-7
		0.0185	5.4685e-5	5.3676e-7
s PSO (简化粒子群算法)	1	0.1042	0.0286	0.0075
		0.1065	0.0679	0.0723
		0.0297	0.0036	0.1037
	0.7298	7.7174e-54	8.4436e-107	0
		1.6676e-55	9.1618e-105	0
		3.5580e-52	5.4606e-105	0
Mypso (文中的改进粒子群算法)	1	2.6748e-11	5.2407e-23	2.3628e-127
		1.6305e-10	1.3293e-24	2.0875e-108
		1.1745e-9	2.0617e-25	6.7451e-112
	0.7298	2.2065e-53	5.8848e-104	0
		9.0914e-53	6.7497e-102	0
		6.8327e-53	5.7905e-105	0

表 3 Griewank 函数测试结果比较

Griewank 函数	$\omega_i$ (权初值)	最大迭代次数:50	最大迭代次数:100	最大迭代次数:500
b PSO (基本粒子群算法)	1	1.0594e-4	1.1297e-4	4.6017e-5
		3.6864e-4	8.7796e-5	8.0402e-5
		1.2930e-4	1.08173e-4	1.5631e-4
	0.7298	1.3529e-6	6.5519e-7	3.0953e-11
		1.5165e-6	2.3617e-8	1.6181e-9
		1.8309e-7	3.5174e-7	1.4384e-11
s PSO (简化粒子群算法)	1	8.2112e-6	3.6786e-5	7.1581e-5
		2.2740e-5	3.7714e-5	1.9022e-6
		2.4029e-5	9.9581e-6	4.6856e-6
	0.7298	1.1103e-56	1.7309e-109	0
		9.7158e-60	1.0981e-111	0
		9.0278e-58	8.9632e-108	0
Mypso (文中的改进粒子群算法)	1	2.8573e-17	5.2497e-23	2.3628e-127
		6.7564e-12	1.3293e-24	2.0875e-108
		1.0758e-14	2.0617e-25	6.7451e-112
	0.7298	5.4179e-57	1.2346e-107	0
		1.8013e-59	1.4455e-108	0
		3.2252e-56	5.9999e-108	0

3.2 Rastrigin 函数测试

在同类试验中,一般视实验结果在小于 50 为优,

而在文献[10]中则以小于 25 为优。而本算法的实验结果均小于 50,并且大部分在 10 以内,故将实验结果分为 3 个区间方便比较(见表 4)。

表 4 Rastrigin 函数测试结果比较

Rastrigin 函数	$\omega_i$ (权初值)	实验结果	最大迭代次数:50(百分比)	最大迭代次数:100(百分比)	最大迭代次数:500(百分比)
b PSO (基本粒子群算法)	1	<10	10	20	40
		10-25	80	65	55
		25-50	10	5	5
	0.7298	<10	50	45	50
		10-25	40	50	45
		25-50	10	5	5
s PSO (简化粒子群算法)	1	<10	55	85	95
		10-25	35	10	5
		25-50	10	5	0
	0.7298	<10	100(全为 0)	100(全为 0)	100(全为 0)
		10-25	0	0	0
		25-50	0	0	0
Mypso (文中的改进粒子群算法)	1	<10	75	95	95
		10-25	20	5	5
		25-50	5	0	0
	0.7298	<10	100(全为 0)	100(全为 0)	100(全为 0)
		10-25	0	0	0
		25-50	0	0	0

由表 4 易发现,当选取权初值为 0.7298 时,sPSO 和文中改进算法在迭代 50 次以上所得的结果几乎都可以得到理论最小值 0(matlab 中计算结果近似为 0),而一般误差在 50 以内就已经可视为达优。当任意取权值 1 时,文中算法所得的结果则明显优于 sPSO。

4 结束语

分析了 bPSO 算法陷入局部极值的原因,在简化粒子群算法的基础上,对其中的惯性权值引入余弦函数进行动态调整,使得算法迅速摆脱局部极值,并提高了收敛速度。通过对 3 个经典的优化测试函数进行试验,结果表明,改进算法能在搜优精度收敛速度以及收敛精度上都有显著的效果。然而算法中某些参数的选取是否最佳,以及如何有效地进行调整还有待进一步研究。

参考文献:

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proc 1995 ICEC. Perth, Australia;[s. n. ],1995.

[2] Kennedy J, Eberhart R C. Particle swarm optimization[EB/OL]. 1995. <http://www.engr.iupui.edu/~shi/Conference/psopap4.html>.

[3] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]// In: Proc. of the IEEE Int'l Conf. of Evolutionary Computation. Piscataway: IEEE Press, 1998:303-308.

[4] Shi Y, Eberhart R C. Empirical study of particle swarm opti-

(massager MA)等的运行状态。网络管理代理和信使代理负责监视任务完成情况,及时添加和删除移动代理的登记,并负责移动代理返回结果的正确接收存储(这些数据可能要经过星上系统分析处理、转发)。对于任务代理,它负责维护和记录代理任务的完成状况,在一个任务卫星漫游出区的时候,移动 Agent 状态管理子库负责记录任务的完成情况,并将任务代理交给另一个经过目标区域的卫星。

⑤网络管理模块:由于将中间管理的功能置于静止轨道卫星之上,因此由位于静止轨道卫星上的常驻管理程序完成网络管理、任务管理及移动代理管理。该模块是中间管理节点的核心部分,它同时负责与地面派遣的移动代理、资源库以及 MNMP 管理代理之间的协调工作。地面管理站根据用户提供的要求生成相应的移动 Agent,并派遣到中间管理节点上;移动 Agent 与网络管理模块进行交互,将必要的信息存入对应的资源库;同时网络管理模块又与 MNMP 管理代理进行交互,将要对被管节点进行的网络管理经过 MNMP 与被管节点进行通信,获得所需的信息。

### (3)被管节点层。

被管节点层是由一些任务星(星座)、单星、近空间平台以及飞行器等组成。该层可以看成一种特殊的 Ad Hoc 网络,将 Ad Hoc 网络中的分簇方法引入被管节点层,这样将规模较大的异构天基综合信息网络分成了各个规模较小的域管理,从而减小管理的复杂性,进一步提高网络管理的效率。中间管理层通过 MNMP 与被管节点层的簇头节点进行通信,再经过转换网关,将其转换为 SNMP 或 CMIP,从而实现与对应的被管节点进行通信。被管节点完成信息收集后,再转换为 MNMP 格式发送给中间管理节点。

## 4 结束语

天基综合信息网络将是一个异构和繁杂的网络,传统的地面网络管理已经不能适应这种网络的管理。

通过对天基综合信息网络进行分层分域管理,将移动代理技术和 MNMP 结合进行天基综合信息网络的的管理,可以有效平衡网络负载,降低网络的高动态、高时延等影响。下一步的工作是在仿真平台上实现该模型,并对其性能进行分析。

### 参考文献:

- [1] 李小将,李志德,杨 健,等. 近空间装备体系概念及关键问题研究[J]. 装备指挥技术学院学报,2007,18(4):72-77.
- [2] Courtney W, Freitag J, Kelly C. Developing Functional and Performance Requirements for Broadband Satellite Networks (BSN)[C]//Fifth Ka-Band Utilization Conference. Taromina, Sicily Island, Italy:[s. n.],1999.
- [3] Ramachandran, Padmanabham, Wipro. Application of Next Generation Telecom Network Management Architecture to Satellite round Systems[C]// In Proceeding of the Ground System Architecture Workshop (GSAW2005). Manhattan Beach Marriott, Manhattan Beach, California:[s. n.],2005.
- [4] 闻英友,赵建立,王光兴.一种面向卫星综合信息网的网络管理系统[J]. 兵工学报,2005(3):231-235.
- [5] 赵建立,闻英友,潘成胜,等.多元网关协议的设计、验证及其仿真[J]. 通信学报,2005,26(4):1-6.
- [6] ISO/IEC 9596-1. Information Processing, Open System Interconnection, Common Management Information Protocol (CMIP)[S].1990.
- [7] Shen Jun, Gu Guanqun, Luo Junzhou. Research and development of network management[J]. Journal of Computer Research and Development,2002,39(10):1153-1167.
- [8] Li Zhi-wen, Meng Luo-ming, Qi Feng. The study and perspective of mobile agent applications in network management environment[J]. ACTA Electronica Sinica, 2002, 30(4):564-569.
- [9] Zhang Pu-han, Sun Yu-fang. Evaluating the performance of a network management based on mobile agents[J]. Journal of Software, 2002,13(11):2090-2098.

(上接第 139 页)

- mization[C]//Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway, NJ:IEEE Press,1999:1945-1950.
- [5] 吕振肃,侯志荣.自适应变异的粒子群优化算法[J]. 电子学报,2004,32(3):416-420.
  - [6] 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(4):861-868.
  - [7] Clerc M. The swarm and the queen: Towards a deterministic

and adaptive particle swarm optimization[C]// In: Proc. of the ICEC. Washington:[s. n.], 1999:1951-1957.

- [8] 王俊伟. 粒子群算法中惯性权重的实验与分析Ⅲ[J]. 系统工程学报,2005,20(2):194-197.
- [9] 王启付. 一种动态改变惯性权重的粒子群优化算法[J]. 中国机械工程,2005,16(11):945-948.
- [10] 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.