

文章编号:2095-7386(2016)02-0083-04

DOI:10.3969/j.issn.2095-7386.2016.02.015

基于区域均衡和压缩因子的混合粒子群算法的研究

陈欣, 刘朔

(武汉轻工大学 数学与计算机学院, 湖北 武汉 430023)

摘要: 针对传统粒子群算法容易陷入局部最优解的问题, 本文提出一种在采用全局最优和区域均衡最优相结合来共同更新粒子速度的基础上, 融合压缩因子的混合算法。将粒子的局部区域均衡的信息加入到速度更新公式, 使得粒子的速度同时受到全局最优和区域平衡最优的作用, 提高了粒子的寻优性能, 平衡粒子的全局寻优能力和局部寻优能力。通过选取几个多峰值的测试函数进行仿真实验, 表明该算法粒子的寻优性能有较大优势。

关键词: 混合粒子群算法; 区域均衡; 压缩因子

中图分类号: TP 301.6

文献标识码: A

Hybrid particle swarm optimization algorithm of regional equilibrium and compression factor

CHEN Xin, LIU Shuo

(School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China)

Abstract: In order to avoid traditional PSO algorithm converging too fast and easily falling into local optimum value, the PSO algorithm is proposed with regional equilibrium and global optimal Value, adding the regional equilibrium optimum value into the updating formula of particle speed, and improving the search ability of the particle. Compared three classic test function, the algorithm can greatly improve the search ability of the particle.

Key words: Hybrid particle swarm optimization; Regional equilibrium; Compression factor

1 引言

在众多群体智能算法中, 粒子群算法的结构简单, 易于实现, 并有着非常广泛的应用。但是, 粒子群算法和其他的智能算法一样, 也存在陷入局部最优解的弱点。为了克服这些弱点, 很多学者提出了不同的改进方案。马国庆等^[1]提出了利用惯性权重线性递减的策略。任圆圆^[2]给出非线性的动态惯性权重设置即自适应权重的公式。Shi Y 等^[3]则

从控制微粒的飞行速度入手, 构造了设置压缩因子的改进粒子群算法, 都取得了一定的效果。以上这些算法, 粒子速度的更新仍然极大取决于自身最优解和全局最优解, 容易出现早熟的现象。本文提出一种混合粒子群算法, 在微粒的速度更新公式中引入区域均衡最优解和飞行速度控制的压缩因子, 使得粒子速度的更新更为合理, 使混合粒子群算法的搜索能力和收敛精确度进一步增强。

收稿日期: 2016-03-04. 修回日期: 2016-03-31

作者简介: 陈欣(1978-), 男, 讲师, 硕士, E-mail: 2924892402@qq.com

2 标准的粒子群算法

经典的粒子群算法是在 1995 年由 Eberhart 等人提出来的,通过针对鸟类群体觅食行为的模拟研究。鸟类用简单的规则确定自己的飞行速度和飞行方向(彼此间不发生碰撞),当一只鸟飞离鸟群而飞向栖息地时,将导致它周围的其他鸟也飞翔该栖息地^[4]。与其他进化类算法相似,粒子群算法也采用“种群”和“进化”的概念,根据个体的适应值大小进行操作。假设在一个 S 维的目标搜索空间内,有一个种群包含 m 个粒子,其中第 i 个粒子表示为一个 S 维向量 $X_i = (X_{i1}, X_{i2}, \dots, X_{is})$, $i = 1, 2, 3 \dots m$, 每个粒子的位置就是一个潜在的解(或者称其为初始位置),将 X_i 代入一个目标函数就可以算出其适应值,然后根据适应值的大小衡量解的优劣。第 i 个粒子当前搜索到的最优位置记为 $P_i = (P_{i1}, P_{i2}, \dots, P_{is})$, 整个粒子群当前搜索到的最佳位置为 $P_g = (P_{g1}, P_{g2}, \dots, P_{gs})$, 假设 $f(x)$ 是求最小值的目标函数,则第 i 个粒子当前搜索到的最佳位置见公式为:

$$P_i(t+1) = \begin{cases} P_i(t), & \text{iff}(x_i(t+1)) \geq f(P_i(t)) \\ x_i(t+1), & \text{iff}(x_i(t+1)) < f(P_i(t)) \end{cases} \quad (1)$$

而第 i 个粒子的速度和位置更新见公式(2)和(3):

$$V_i(t+1) = V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t)) \quad (2)$$

$$x_i(t+1) = x_i(t) + V_i(t+1) \quad (3)$$

其中 c_1, c_2 是非负常数, r_1, r_2 是相互独立的随机数, $V_i \in [-v_{\max}, v_{\max}]$, 其中 $v_{\max}$ 由用户根据实际问题具体设定。迭代的终止条件一般选定为最大迭代次数 T 。

3 引入区域均衡最优解和压缩因子的混合粒子群算法

3.1 融合区域均衡的最优解

传统粒子群算法中,种群中的每一个粒子的速度更新都是依赖于该粒子本身的历史最优解和整个种群的最优解。在传统粒子群算法的前期搜寻阶段,很有可能会出现粒子搜寻过快,与最优解擦肩而过的情况,这样接下来的搜寻就很容易陷入局部最优的陷阱。陈群林等^[5]为了提高算法搜寻的准确度,针对粒子群算法局部搜寻能力引入了粒子加速

和变异的机制。张水平等^[4]就提出一种用每个粒子的邻居的当前最优解代替全局最优解来平衡局部搜索能力的方案,但是这种做法仍然具有一定的局限性,特别是每个粒子之间的距离的计算和存储将会使得算法的运算时间和存储空间的花销大大增加。本文把该粒子作为一个局部区域的中心,设定一个圆形的邻域,以该邻域内部所有粒子的当前最优位置的几何中心 P_n 作为该区域内的全局最优位置,这样,每个粒子周围的局部信息将得到更加高效和准确的传递。即使在最极端的情况下,可能出现在所给邻域内并没有出现其他的粒子,这样了邻域的范围也就应该做出相应的扩大调整。邻域扩大过程如下:

$$r = k;$$

$$\text{count} = \text{count}(\overset{\circ}{U}(P_i, r));$$

$$\text{search} = 1;$$

$$\text{while count} = 0 \text{ or search} \neq 3$$

$$r = 1.5 * r;$$

$$\text{count} = \text{count}(\overset{\circ}{U}(P_i, r));$$

$$\text{search} = \text{search} + 1;$$

如果三次调整后,周围仍然没有其余粒子出现,恰好说明该粒子周围局部信息不足,无需平衡,此时直接应用原始的速度更新公式即可。综合以上的讨论,本文提出一个基于局部范围内区最优解的速度更新公式为:

$$V_i(t+1) = V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_n(t) - x_i(t))$$

以往的研究表明,传统的全局 PSO 算法收敛速度快,但容易陷入局部最优的陷阱^[6]。而结合区域均衡最优解的搜索算法可以搜索到相对更优更精确的解,但是搜索速度会更慢一些。因此,本文采用一种采用全局最优和区域均衡最优相结合来共同更新粒子速度的混合算法,公式如下:

$$H_i(t+1) = u_1 M_i(t+1) + u_2 N_i(t+1)$$

其中:

$$M_i(t+1) = V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_n(t) - x_i(t)) \text{ 是针对局部均衡寻优。}$$

$$N_i(t+1) = V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t)) \text{ 是针对全局均衡寻优。}$$

并且 $u_1 = t/T, u_2 = 1 - u_1$ 的设置可以保证算法在兼顾了局部寻优和全局寻优的情况下,前期粒子的速度不会太快以至于错过最优解,在算法后期也会更具有精确性。

3.2 压缩因子

在速度更新的公式中, c_1, c_2 这两个非负常数的值决定了微粒本身的经验和其他微粒的经验信息对运动轨迹的影响,反映了微粒之间的信息交流^[7]。设置 c_1 较大,会使微粒更多地在局部徘徊,而设置 c_2 较大,又会使微粒过早收敛,为了达到全局探测和局部开采的两者间的平衡,Shi Y 等^[3] 提出了设置收缩因子的策略,速度更新公式如下:

$$V_i(t+1) = \varphi \{ V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t)) \} \text{ 其中}$$

$$\varphi = 2 / \sqrt{2 - c - \sqrt{c^2 - 4c}}, c = c_1 + c_2, c > 4。$$

文献[3]提出的相比文献[1],[2]提出的针对权重的优化,设置压缩因子比权重系数,更能有效地控制约束微粒的飞行速度,同时增强了算法的局部搜索能力。本文对压缩因子的设置做了一定的改进,在控制了微粒飞行的速度的情况下,完整地保留对微粒间个体最佳适应值和全局最佳适应值信息的传递,更新后的速度更新公式如下:

$$V_i(t+1) = \varphi V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t))$$

其中

$$\varphi = 2 / \sqrt{2 - c - \sqrt{c^2 - 4c}}, c = c_1 + c_2, c > 4$$

3.3 区域均衡最优解和压缩因子的混合

本文在引入区域均衡最优解和设定压缩因子这两个方面,改进了粒子的更新公式和位置更新公式,提出了一种混合粒子群算法。新的速度和位置更新公式如下:

$$M_i(t+1) = \varphi V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_n(t) - x_i(t))$$

$$N_i(t+1) = \varphi V_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (P_g(t) - x_i(t))$$

$$H_i(t+1) = u_1 M_i(t+1) + u_2 N_i(t+1)$$

$$X_i(t+1) = X_i(t) + H_i(t+1)$$

4 仿真实验

为了测试本文算法的实际性能效果,将其与文献^[1]提出的惯性权重线性递减算法与文献^[2]提出的自适应权重算法经行实验比较,采用三个多峰值测试函数进行测试比较每个函数的参数设置如表1,表2,表3所示,算法测试结果如表4,表5,表6所示。

表1 Ackley 函数的初始参数设置

种群	C1	C2	初始区间	目标值	迭代数
40	1.5	1.5	[-32,32]	0	100

表2 Rastrigrin 函数的初始参数设置

种群	C1	C2	初始区间	目标值	迭代数
40	1.5	1.5	[-5.12,5.12]	0	100

表3 Schaffer 函数的初始参数设置

种群	C1	C2	初始区间	目标值	迭代数
40	1.5	1.5	[-10,10]	0	100

三个多峰测试函数^[8]如下:

Ackley 函数:

$$f(x_1, x_2, \dots, x_n) = -c_1 e^{0.2 \sqrt{\frac{1}{n} \sum_{j=1}^n x_j^2}} - e^{\frac{1}{n} \cos(2\pi x_j)} + c_1 + e$$

Rastrigrin 函数:

$$f(x_1, x_2, \dots, x_n) = \frac{1}{n} \sum_{j=1}^n [x_j^2 - 10 \cos(2\pi x_j) + 10]$$

Schaffer 函数:

$$f(x_1, x_2) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$$

表4 对于 Ackley 测试函数的不同算法运行结果

	惯性权重递减的 PSO 算法	自适应权重调整的 PSO 算法	本文算法
算法运行次数(次)	30	30	30
算法平均运行时间(s)	0.0760	0.0786	0.2323
最佳平均适应度(10^{-6})	-0.525	-0.539	-0.255

表5 对于 Rastrigrin 测试函数的不同算法运行结果

	惯性权重递减的 PSO 算法	自适应权重调整的 PSO 算法	本文算法
算法运行次数(次)	30	30	30
算法平均运行时间(s)	0.0650	0.0586	0.1883
最佳平均适应度(10^{-6})	-0.425	-0.434	-0.221

表 6 对于 Schaffer 测试函数的不同算法运行结果

	惯性权重递减的 PSO 算法	自适应权重调整的 PSO 算法	本文算法
算法运行次数(次)	30	30	30
算法平均运行时间(s)	0.0692	0.0667	0.2011
最佳平均适应度(10^{-6})	-0.423	-0.512	-0.105

本文选取的三个测试函数都是经典的多峰值测试函数,即可以对算法的准确度进行测试,也可以对算法的寻优精度进行评价。从图 7,图 8,图 9 可以看出,本文算法的收敛速度和寻优精度都是优于文献^[1,2]提出的针对权重调整的算法。而且本算法的鲁棒性较好,不容易掉入局部最优的陷阱。

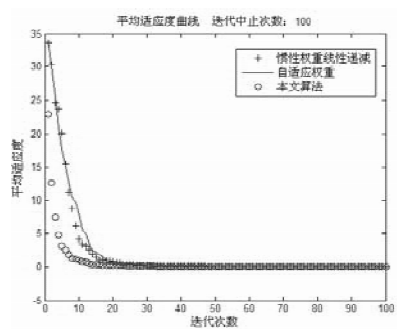


图 7 Ackley 函数测试图

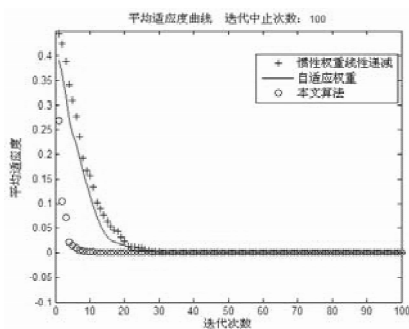


图 8 Rastrigrin 函数测试图

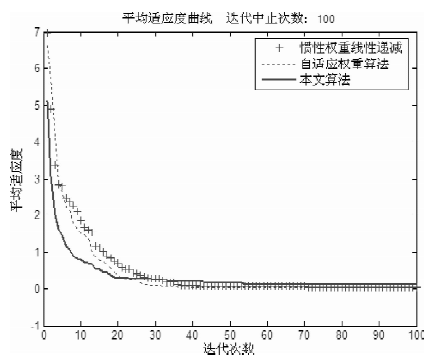


图 9 Schaffer 函数测试图

5 结束语

针对传统粒子群算法容易陷入局部最优的问题,提出一种混合粒子群算法,在微粒的速度更新公式中引入区域均衡最优解和飞行速度控制的压缩因子,将全局最优解和区域平衡最优解的影响因素同时用于更新粒子的速度。实验表明,本文算法在优化能力和性能上都有了很大的提高,但是算法的运行速度和稳定性还有待于进一步提高。

参考文献:

- [1] 马国庆,李瑞峰,刘丽.学习因子和时间因子随权重调整的粒子群算法[J].计算机应用研究,2014,31(11):3291-3294.
- [2] 任圆圆,刘培玉,薛素芝.一种新的自适应动态文化粒子群优化算法[J].计算机应用研究,2013,30(11):3240-3243.
- [3] Shi Y, Eberhart RC. Empirical Study of Particle Swarm Optimization [C]//Proc of Congress on Evolutionary Computation. 1999
- [4] 张水平 仲伟彪.改进学习因子和约束因子的混合粒子群算法[J].计算机应用研究,2015,32(12)3626-3628.
- [5] 陈群林,高岳林,郭祥.一种带有加速策略和变异策略的粒子群算法[J].兰州文理学院学报(自然科学版),2015,29(6):41-47.
- [6] 孙振龙,李晓晔,王颖.一种改进的简化粒子群优化算法[J].计算机科学,2015,42(11A):86-88.
- [7] 冯浩,李现伟.一种改进的粒子群优化算法惯性权值递减策略[J].蚌埠学院学报,2015,4(6):21-24.
- [8] 宋锦,高浩.基于双重更新策略的粒子群算法.南京邮电大学学报(自然科学版),2015,35(6):84-88.