

一种改进的动态改变惯性权重的粒子群算法

杜振鑫, 王兆青

(浙江理工大学 计算机技术教研部, 浙江 杭州 310018)

摘 要: 惯性权重在粒子群算法中起到重要的作用, 综合考虑了影响惯性权重的几种因素, 提出基于进化速度、聚集度和相似度的动态改变惯性权重的粒子群算法, 实验证明改进算法在收敛率、收敛精度和全局寻优能力方面都优于几种有代表性的动态改变惯性权重的算法。

关键词: 粒子群算法; 相似度; 进化度; 聚集度; 惯性权重

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2011)03-0085-04

Improved PSO Algorithm with Dynamic Change of Inertia Weights

DU Zhen-xin, WANG Zhao-qing

(Instructional Division of Computer Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Inertia weight is an important parameter in particle swarm optimization algorithm. In this paper, evolution speed, aggregation degree and similarity is integrated into particle swarm optimization organically to control the inertia weight better, and enhance the escaping ability from local optimum when used on complicated problems. The modified PSO algorithm improves the abilities of seeking the global excellent result and convergence accuracy. The experiment results demonstrate that the proposed algorithm are superior to several typical particle swarm optimization algorithms based on dynamic change of inertia weights.

Key words: particle swarm optimizer; similarity; evolution speed; gather degree; inertia weight

1 引言

粒子群算法(Particle Swarm Optimization, PSO)自 Kennedy 和 Eberhart^[1]于 1995 年提出以后, 由于原理简单, 可调参数少等优点, 迅速得到了广泛的应用. 文献[2]研究了粒子群算法在光源优化布局中的应用, 文献[3]提出用粒子群算法设定传递函数, 都取得了理想的效果. 但是粒子群算法同时也存在容易早熟等缺点. 惯性权重在粒子群算法中是一个重要的控制参数, 能够起到平衡全局搜索和局部搜索, 影响种群多样性的作用. 本文根据算法进化机理, 综合分析了几种影响惯性权重的因素, 提出了基于进化速度、聚集度和相似度动态改变惯性权重的算法, 使得算法能够较好地保持种群多样性, 实验证明算法好于几种改进的动态改变惯性权重的算法, 收敛率和寻优精度都得到较大提高.

2 基本粒子群算法

粒子群优化算法的原理来源于对鸟群觅食活动的仿真研究. PSO 首先初始化 n 个粒子($X_1, X_2, \dots, X_n$), 第 i 个粒子的空间坐标是($X_{i1}, X_{i2}, \dots, X_{id}$), D 代表每个粒子的维数. 同时, 每个粒子随机初始化一个方向($V_{i1}, V_{i2}, \dots, V_{id}$). 每次迭代中, 每个粒子都向两个位置靠近: 一个是个体当前最优极值 P_{best} , 一个是当前全局最优极值 g_{best} . 每次迭代结束后, 计算每个粒子的适应度, 通过式(1)更新每个粒子的速度, 通过式(2)更新每个粒子的位置. 每轮迭代结束后, 更新每个粒子的当前最优个体极值 P_{best} 和全局最优极值 g_{best} , 如此循环直至全部粒子都收敛于全局最优粒子, 则迭代结束.

$$V_{id} = V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

式中, 常数 c_1 和 c_2 是正的加速系数, r_1 和 r_2 是 $[0, 1]$ 之间服从均匀分布的随机数, P_{id} 是个体极值的第 d 维分量, P_{gd} 是全局极值的第 d 维分量。

粒子群算法提出以后, 立刻受到学术界和工程界的关注, 其中对于惯性权重的改进是其中重要的改进。Shi 和 Eberhart 于 1998 年在基本粒子群算法的基础上引入了惯性权重的概念, 将式(1) 修改为

$$V_{id} = w * V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id}) \quad (3)$$

此处 w 称为惯性权重, 它决定粒子对以前的速度的记忆程度, 具有平衡粒子勘探和开采的能力。随后, 他们二人再次提出线性递减控制惯性权重的方法^[4]:

$$w = w_{\max} - (w_{\max} - w_{\min}) \times k / iter_{\max},$$

式中, $w_{\max}$ 为初始惯性权重, $w_{\min}$ 为最终权重, $iter_{\max}$ 为最大迭代次数, k 为当前迭代次数。这个改进使得算法初期注重全局搜索, 后期注重局部搜索。随后, 文献[5] 提出了基于模糊系统的惯性因子动态调整方法, 实验结果表明此方法比线性调整方法具有更好的性能。

3 改进的粒子群算法

3.1 聚集度

粒子群算法跟其他进化算法一样, 存在容易早熟收敛的问题, 因此准确评价聚集度就成为重要的问题。文献[6] 提出了一种计算聚集度的方法:

$$g(i) = 1 - \frac{1}{s \cdot L} \sum_{d=1}^S \sqrt{\sum_{i=1}^n (P_{id} - \bar{P}_d)^2} \quad (4)$$

式中, t 代表迭代次数, g 代表聚集度, L 为整个搜索空间的最长半径, S 为粒子种群规模, n 为解空间的维数, P_{id} 是第 i 个粒子的第 d 维坐标, $\bar{P}_d$ 是所有粒子第 d 维坐标的平均值。

3.2 改进的粒子群算法

在粒子群算法中, 惯性权重是一个十分重要的参数。较大的惯性权重有利于全局搜索, 增加种群多样性; 较小的惯性权重有利于局部搜索。如果对所有粒子都赋予同样的惯性权重, 就很难平衡全局搜索和局部搜索之间的矛盾, 所以应当对不同粒子采用不同的惯性权重。距离全局最优值 P_g 较近的粒子找到真正的全局最优值的可能性非常大, 应当让这些粒子侧重局部搜索, 越是靠近 P_g 的粒子其惯性权重应该越小, 并且聚集度越大, 这部分粒子的惯性权重也应该越小。距离全局最优值 P_g 较远的粒子,

应当赋予较大的权重, 以侧重全局搜索, 并且聚集度越大, 权重也应当越大。惯性权重的大小还应当与进化速度有关, 当进化速度变化时, 惯性权重也应当作出相应的调整。根据上述分析, 本文提出如下改进公式:

$$\bar{d}(t) = \frac{1}{n} \sum_{i=1}^n |f_i - f_{\min}| \quad (5)$$

$$sim(i) =$$

$$\begin{cases} 1, & |f_i - f_{\min}| < \alpha_1 \times \bar{d}(t) \\ 1 - \frac{|f_i - f_{\min}|}{\alpha_2 \times \bar{d}(t)}, & \alpha_1 \times \bar{d}(t) \leq |f_i - f_{\min}| \leq \alpha_2 \times \bar{d}(t) \\ 0, & |f_i - f_{\min}| > \alpha_2 \times \bar{d}(t) \end{cases} \quad (6)$$

式(5) 中, $\bar{d}(t)$ 表示第 t 次迭代中, 所有粒子与全局最优粒子 $f_{\min}$ 的适应度距离的平均值, 其中 f_i 表示第 i 个粒子的适应度, n 表示粒子的个数。式(6) 中, $sim(i)$ 表示第 t 次迭代中, 第 i 个粒子与全局最优粒子 $f_{\min}$ 的相似程度, α_1 、 α_2 是控制参数。式中使用了适应度距离的概念, 而未使用欧氏距离, 可以大大减少计算复杂度, 尤其当粒子维数较大时, 节省的计算次数更多。同时, 相似度分母使用随迭代动态改变的平均距离 $\bar{d}(t)$, 而不是固定的值, 使得文献[7] 中当搜索范围较大时所有粒子的相似度趋同的问题得到有效解决。式(7) 中, $w(i)$ 表示第 i 个粒子的惯性权重, T 表示总共迭代次数, $g(t)$ 表示第 t 次迭代的聚集度, e 表示自然对数的底, ρ 、 γ 、 φ 是控制参数。 $1 - t/T + \rho \times g(t)$ 表示惯性权重整体随迭代增加而递减, 局部随聚集度增加而增加, 这样当聚集度上升时, 较差粒子会自适应地加大开拓空间, 防止聚集度过度上升。 $0.5 - \varphi \times g(t)$ 的部分表示当聚集度上升时, 核心粒子根据聚集度自适应的减少惯性权重, 保证对最优粒子周围的精细搜索。在惯性权重公式中引入 $\bar{d}(t)/\bar{d}(t-1)$ 表示相邻两次迭代中, 全局平均距离的变化程度(即进化速度), 当 $\bar{d}(t)/\bar{d}(t-1) < 1$ 时, 说明该次迭代正处于收敛状态, 比值越小, 粒子聚集的趋势越明显, 此时应当适当加大 $w(i)$, 防止粒子过分聚集, 公式正好满足要求; 当 $\bar{d}(t)/\bar{d}(t-1) > 1$ 时, 说明该次迭代正处于发散状态, 比值越大, 粒子发散的趋势越明显, 此时应当适当减少 $w(i)$, 防止粒子过分发散, 公式也正好满足要求。 e 的指数部分的值基本在 $[0, 1]$ 区间内, 满足了惯性权重一般属于 $[0, 1]$ 区间的要求。同时, 公式考虑了每个粒子与 $f_{\min}$ 的相似度 $sim(i)$, 使得每次迭代内, 每个粒子都有不同的惯性权重: 与 $f_{\min}$ 的适应度距

离越小,则 $sim(i)$ 越大, $w(i)$ 越小,满足了贴近 f_{min} 的粒子应当精细搜索的要求. 本算法综合考虑了粒子相似度 (Similarity degree)、聚集度 (Gather degree)、进化速度 (Evolution speed) 的影响,所以称为 SGEPSO. 算法流程如下:

Step 1: 随机初始化 n 个粒子的位置及其初始速度;

Step2: 计算各个粒子的适应度值,确定各个粒子的当前最优个体位置,确定当前整个种群的最优位置;

Step3: 根据式(5)计算第 t 次迭代的种群平均距离 $\bar{d}(t)$; 根据式(6)计算各个粒子与全局最优位置的相似程度 $sim(i)$, 根据式(4)计算第 t 次迭代的种群聚集度, 根据式(7)计算各个粒子的惯性权重 $w(i)$, 其中当 $t=1$ 时 $\bar{d}(t)/\bar{d}(t-1)$ 设为 1.

Step4: 根据式(2)和(3)分别更新粒子的速度和位置.

Step5: 若未满足结束条件,则算法转 Step2, 否则算法结束并输出全局最优值.

4 实验与分析

算法首先在同样条件下对 f_1 和 f_2 两个函数进行了两次实验,在同等条件下,比较各个算法的平均收敛率(误差范围内的收敛次数与总的实验次数的比值)、收敛时的平均收敛代数和平均函数值(算法执行到规定进化代数后的函数值的多次平均值).

$$f_1(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2) + 48x_2 - 36x_1x_2 + 27x_2^2], |x_i| \leq 2$$

$$f_2(x) = \frac{\sin \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001 * (x_1^2 + x_2^2))^2} + 0.5,$$

$$-5.12 \leq x_1, x_2 \leq 5.12$$

为了同文献[7]提出的动态改变惯性权重的算法 DPSO 进行比较,本文第一次实验使用与文献[7]相同的实验条件. 对于函数 f_1 , DPSO 的参数设置为 $c_1 = 1.8, c_2 = 1.8, w_{max} = 0.8, w_{min} = 0.2, v_{max} = 1, d_{min} = 0.5, d_{max} = 1$; 对于函数 f_2 , DPSO 的设置 为 $c_1 = 1.8, c_2 = 1.8, w_{max} = 1.0, w_{min} = 0.4, v_{max} = 10, d_{min} = 10, d_{max} = 50$. SGEPSO 的参数设置始终保持不变, $c_1 = 2.05, c_2 = 2.05, \alpha_1 = 0.2, \alpha_2 = 3, \rho = 0.5, \gamma = 2, \varphi = 0.25$. 表 1 所示为两个函数两次实验的实验条件,表 2 所示为第一次实验的数据是算法按上述设置对两个函数进行 100 次测试的平均结果.

表 1 两次实验的条件

函数	群体规模	进化代数	收敛误差
f_1	10	100	1e-20
f_1	20	100	1e-20
f_2	20	100	0.001
f_2	50	200	0.001

从表 2 可以看出,对于函数 f_1 和 f_2 ,不论是平均收敛率还是平均函数值,SGEPSO 都比 DPSO 在性能上有了较大的提高,这是因为 SGEPSO 更好的考虑了聚集度和进化速度对算法的影响,使得算法不容易陷入局部最优,搜索到全局最优解的可能性增大,因此平均收敛率和平均函数值都有较大改善. 但是 SGEPSO 的平均收敛代数比 DPSO 有所增加,这是因为 SGEPSO 考虑了聚集度和进化速度的影响,不容易过早收敛,使得算法收敛速度下降. 从 DPSO 参数设置可以看出, DPSO 的参数 $w_{max}, w_{min}, v_{max}, d_{min}, d_{max}$ 对具体问题的依赖性很强,不同的问题要设置不同的参数,否则会较大影响算法的性能,造成算法的性能下降. 对于上述最优值已知的测试函数, DPSO 可以通过调整参数来达到算法的最优性能,而在实际问题当中由于全局最优值无法预知,所以很难找到一个合适的参数组合,这就使得 DPSO 算法的性能受到很大限制,而 SGEPSO 的参数设置不依赖于任何具体的问题,所以通用性较好.

表 2 DPSO 与 SGEPSO 两次实验各 100 次测试平均值

函数	实验 序号	收敛率		收敛代数		函数值	
		DPSO	SGEPSO	DPSO	SGEPSO	DPSO	SGEPSO
f_1	1	84	95	47.66	69.87	12.72	6.24
	2	78	92	68.32	70.54	17.46	7.05
f_1	1	91	99	40.54	61.60	8.94	3.81
	2	85	98	62.10	59.24	14.34	3.82
f_2	1	21	28	41.85	43.58	0.008 3	0.007 4
	2	18	26	44.94	38.87	0.009 1	0.007 3
f_2	1	62	73	47.97	56.77	0.005 1	0.004 4
	2	57	71	63.54	62.69	0.005 5	0.003 8

进一步测试 DPSO 对参数设置的敏感性,重新设置 DPSO 的参数,对于函数 f_1 , DPSO 参数设置为 $c_1 = 1.8, c_2 = 1.8, w_{max} = 1.0, w_{min} = 0.4, v_{max} = 1, d_{min} = 10, d_{max} = 50$.

对于函数 f_2 , DPSO 参数设为 $c_1 = 1.8, c_2 = 1.8, w_{max} = 0.8, w_{min} = 0.2, v_{max} = 10, d_{min} = 0.5, d_{max} = 1$. 而 SGEPSO 的参数保持不变. 表 2 中第二次实验的数据是按上述设置 100 次测试的平均值. 可以看出,当 DPSO 改变参数后, f_1 函数的收敛率明显下降,收敛代数和平均函数值都明显上升;对于

f_2 函数, 更换参数的影响不是很明显, 但是收敛率和收敛代数也有一定程度的变坏. 而 SGEPSO 函数始终保持了良好的收敛性能, 没有大的波动. 主要原因如前所述, 因 DPSO 的 $d_{\min}, d_{\max}$ 参数受到具体问题的影响, 设置不当或者优化问题的初值范围较大时会使得各个粒子的相似性趋同, 失去了作用; 而 SGEPSO 是基于平均值原理, 与具体的优化问题的特点无关, 所以保持较好的稳定性, 参数也无需根据具体的函数进行调整.

本文将 SGEPSO、DPSO 和几种典型的惯性权重改进 PSO 算法进行了比较. 算法测试使用常用的典型病态函数 Rosenbrock:

$$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],$$

$$-30 \leq x_i \leq 30, \min(f(x)) = f(1, \dots, 1) = 0$$

此函数是极难优化的典型单峰病态函数, 其全局最优值与可到达的局部最优之间有一道狭窄的山谷, 曲面山谷中的点的最速下降方向几乎与到函数最小值的最好方向垂直. 由于此函数对搜索提供很少信息, 使算法很难辨别搜索方向, 找到全局最优解的机会微乎其微, 因此这个函数常用来评价优化算法的执行性能.

分别将 LWPSO^[4], CBPEPSO^[5], DPSO^[7] 和本文 SGEPSO 算法进行了比较. 函数维数设为 30, 粒子数目为 100, 最大迭代次数为 2 000, SGEPSO 参数设置与前面一样保持不变, 其它参数设置同原文献. 图 1 是对函数运行 100 次后的平均收敛曲线图.

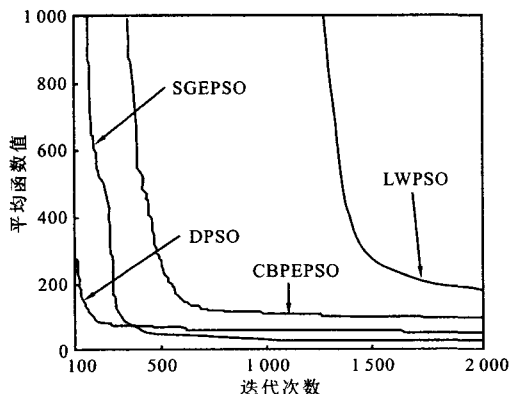


图 1 Rosenbrock 100 次平均收敛曲线

从图 1 中可以看出, SGEPSO 在迭代次数内一直保持着稳定的进化能力, 因此收敛曲线比较平缓, 而其它几种算法则较快就陷入局部最优, 停止进化.

5 结束语

本文综合考虑了影响惯性权重的几种因素, 提出了基于相似性、聚集度、进化速度的惯性权重调整公式, 经过与几种有代表性的动态改变惯性权重的 PSO 算法进行比较, 实验结果表明改进算法优于其它几种算法, 收敛率和精度都得到了较大提高.

参考文献:

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C]//Proceedings of IEEE International Conference on Neural Networks. Piscataway: IEEE Service Center, 1995:1942-1948.
- [2] 曹秒, 安志勇, 张影. 改进的粒子群算法及在光源优化布局中的应用[J]. 微电子学与计算机, 2009, 26(7):69-72.
- [3] 吴莲贵, 齐蕊, 李肯立. 一种基于粒子群优化的传递函数设定方法[J]. 微电子学与计算机, 2009, 26(9):187-190.
- [4] Shi Y, Eberhart R. Empirical Study of Particle Swarm Optimization[C]//Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway: IEEE Service Center, 1999.
- [5] Shi Y, Eberhart R. Fuzzy Adaptive Particle Swarm Optimization[C]//Proceedings of the IEEE Conference on Evolutionary Computation. Soul: IEEE Press, 2001:103-106.
- [6] Riget J, Vesterstroem J S. A diversity-guided particle swarm optimizer—the ARPSO, 2002-02 [R]. Aarhus: University of Aarhus, 2002:1-12.
- [7] 刘建华. 粒子群算法的基本理论及其改进研究[D]. 长沙: 中南大学, 2009.

作者简介:

杜振鑫 男, (1976—), 硕士研究生. 研究方向为计算智能与嵌入式系统.

王兆青 男, (1965—), 博士, 教授, 硕士生导师. 研究方向为网络计算、协同作业、嵌入式系统、传感网络以及可移动代码技术.