

基于自适应认知域的粒子群性能改进方法^{*}

刘 瞰 东¹ 陈 得 宝² 李 素 文² 王 颖¹

¹(厦门大学 自动化系 厦门 361005)

²(淮北煤炭师范学院 物理与电子信息学院 淮北 235000)

摘 要 为提高粒子群算法的收敛性能,提出一种自适应粒子认知域方法.在粒子位置的更新方法中,粒子运动到当前的最好位置由计算得到的最好位置为中心,粒子的认知方向为导向来确定.利用线性惯性下降权重来实现粒子的优化.为验证该方法的有效性,将此方法应用于 3 种不同的粒子群方法,分别是固定权重粒子群方法、线性下降权重粒子群方法及阶梯形群体粒子群算法.实验结果表明此方法是较有效的.

关键词 粒子群优化 (PSO), 线性下降权重粒子群 (LDW PSO), 认知域, 阶梯形群体粒子群算法 (LPSO)

中图分类号 TP 24

An Improved PSO Method Based on Adaptive Cognitive Domain

LIU Tun Dong, CHEN De Bao, LI Su Wen, WANG Ying

¹ (Department of Automation, Xiamen University, Xiamen 361005)

² (Physics and Electronic Information Institute, Huaibei Coal Industry Teachers College, Huaibei 235000)

ABSTRACT

To improve the convergent performance of particle swarm optimization (PSO), an adaptive cognitive domain particle swarm optimization (ACDPSO) method is proposed. In the updating equations of particles, the current best position, which the particle achieves, is determined by the center of the best calculated position and the cognizant direction of the particle. Linear decreasing inertia weight is used to optimize particles. Three different PSOs, particle swarm with constant weight (CWPSO), linear decreasing inertia weight PSO (LDW PSO) and Ladder PSO (LPSO), are combined with the proposed method to test the performance of the proposed method, and the results indicate that the proposed method is effective.

Key Words Particle Swarm Optimization (PSO), Linear Decreasing Inertia Weight PSO (LDW PSO), Cognitive Domain, Ladder PSO (LPSO)

^{*}教育部科学技术研究重点项目 (No. 209057)、安徽省自然科学基金项目 (No. 090412070)、高等学校省级优秀青年人才基金项目 (No. 2009SQRZ088ZD)和高等学校省级自然科学基金项目 (No. KJ2009B062)资助

收稿日期: 2008-12-30 修回日期: 2009-03-09

作者简介: 刘瞰东, 男, 1970年生, 博士, 副教授, 主要研究方向为智能信息处理与控制. E-mail: ltd@xmu.edu.cn; 陈得宝, 男, 1975年生, 博士, 副教授, 主要研究方向为人工智能、进化计算、机器人等; 李素文, 女, 1968年生, 博士, 教授, 主要研究方向为信号处理; 王颖, 女, 1977年生, 博士, 讲师, 主要研究方向为智能优化模式分类、系统建模等.

1 引言

模拟鸟类或鱼类捕食特性的粒子群算法^[1],因其算法简单、易于实现的优点,已广泛用于函数优化^[2]、神经网络^[3]、分类器的设计^[4]等领域.作为一种群智能优化算法,粒子群算法同样存在局部收敛和对复杂优化问题精度难一保证的缺点.

为提高粒子群算法的整体性能,近些年出现很多改进算法,其中有一个重要的研究方法是改善粒子的更新方程出发来提高其性能.文献[5]中给出一种适应度距离比率粒子群算法(Fitness Distance-Ratio Based PSO) (FDRPSO),粒子不仅学习当前的最好粒子和到目前为止的最好位置,还向周围较好粒子学习.文献[6]设计一种具有理解力的粒子群算法,在原来粒子群的搜索空间中增加一维,来确定当前的搜索方向.这样在一定的半径内,粒子能找到当前周边的较好信息,用于粒子的更新.特别是当前代的最好信息对于某粒子难以得到时,这种方法非常有效.文献[7]根据生物体有尽量离开当前的最差位置和粒子到目前为止的最差位置,按最差的信息对粒子群速度和位置进行更新.

然而这些方法中,粒子到目前为止的最好位置,都是通过当前代计算得到,而且此位置对不同的粒子在每代是唯一的.实际上,粒子在飞翔的过程中,由于粒子的强壮程度不同,其认知的范围也不一样,在其认知域内可能存在一个位置比粒子到目前为止的最好位置更有利于全局最优解的获得.基于这样的思想,本文设计一种自适应更新方法,对粒子群更新方程中粒子运动到目前为止的最好位置进行重新分配,以提高粒子群算法的性能.

2 固定权重粒子群算法

设一个有 N 个解的解群 $X_i(d), i \in [1, N], d \in [1, M], M$ 为解的维数.对每个粒子定义一个速度矢量 $V_i(d)$,用它来衡量粒子位置的更新速度,速度的初始值随机生成.则对粒子群进化的第 $K+1$ 代而言,粒子位置的更新表示如下:

$$V_i(K+1) = wV_i(K) + \zeta \text{rand}_1(\cdot)(X_{\text{best}_i}(k) - X_i(k)) + \zeta \text{rand}_2(\cdot)(X_{\text{gbest}}(k) - X_i(k)),$$

$$X_i(K+1) = X_i(K) + V_i(K+1), \quad (1)$$

其中, $X_i(K+1)$ 、 $X_i(K)$ 分别为第 $K+1$ 时刻和第 K 时刻粒子的位置; $V_i(K+1)$ 为 $K+1$ 时刻粒子的运动速

度; w 为加权因子,其较大时有利于全局收敛,较小时有利于局部收敛,通常取 $0.4 \sim 0.9$ 之间^[8]; ζ 、 ζ 为常数,影响着粒子更新过程中认知特性和社会特性的比例; X_{best_i} 、 X_{gbest} 分别为系统进化到当前代第 i 个个体的最好位置和粒子群中最优粒子的位置; $\text{rand}_1(\cdot)$ 、 $\text{rand}_2(\cdot)$ 为 $[0, 1]$ 之间的随机数; $i = 1, 2, \dots, N$ 、 N 为粒子群的总体数量.按照上述的过程更新自己的位置,直到找到自己合适的位置为止.

3 粒子群算法的改进方法

与以前的粒子群算法相比,此方法的显著特点是,更新方程中粒子跟踪得到的目前最好位置不一定是当前算法计算得到的最好位置,而是以当前计算最好位置为中心的一定认知域内,其实际值的最好位置随粒子当前适应值自适应决定.除此之外,本文所设计的方法因其不改变粒子群算法的基本结构,能应用于其它任何改进粒子群方法.基于不同粒子认知域,对粒子群的更新方法进行改进,提高粒子群算法的性能.

基于不同粒子认知域,对粒子群的更新方法进行改进,提高粒子群算法的性能.

定义 1 以当前计算的粒子到目前为止的最好位置为中心,根据不同粒子当前适应度确定的半径为界限的一定区域,定义为此粒子的认知域,其中此域的边界称为最大认知域(Maximum Cognitive Domain, MCD).对第 i 个个体的最大认知域表示如下:

$$MCD_i(t) = \max \frac{\text{fit}(t) - \text{fit}(t)}{\text{fit}(t)}, \quad i = 1, 2, \dots, N$$

其中, $\max \text{fit}(t)$ 、 $\text{fit}(t)$ 分别为粒子 到目前为止所得到的最好适应度值和粒子 当前的适应度值.

定义 2 认知方向(Cognitive Direction, CD)定义为粒子 的第 k 维变量在认知域中的飞行方向.认知方向定义如下:

$$CD_{i,k}(t) = \frac{X_{\text{best}_i}(k, t)}{\text{dis}_{\text{best}_i}}, \quad k = 1, 2, \dots, D$$

其中, $X_{\text{best}_i}(k, t)$ 是粒子 在 t 时刻第 k 维变量所得到的到目前为止的最好值, D 粒子包含的变量的维数; $\text{dis}_{\text{best}_i}$ 是最好粒子各维变量的平方和,

$$\text{dis}_{\text{best}_i} = \sqrt{\sum_{k=1}^D X_{\text{best}_i}^2(k, t)}$$

通过以上的分析,定义更新方程中粒子到目前为止的最好位置如下:

$$X_{\text{best}_i}(i, k) = X_{\text{best}_i}(i, k) + w_i \times \text{sign}(\text{rand}_n(\cdot)) \times MCD_i \times \text{rand}(\cdot) \times CD_{i,k}$$

其中, $X_{pbest_i}^*(i, k)$ 、 $X_{gbest_i}(i, k)$ 分别为粒子 中第 k 维变量到目前为止所具有的新的最好位置和原来计算得到的最好位置; $\text{randn}(\cdot)$ 是均值为零, 标准差为 1 的正态分布随机数; $\text{rand}(\cdot)$ 为 $[0, 1]$ 之间的随机数; w_i 是原有最好位置的飞行率, 太大的取值将会使粒子破坏较严重, 太小的取值, 最好位置将不能达到更新的目的. 通常 w_i 取 $0.001 \sim 0.01$ 之间, 为维持粒子群的多样性, 开始时 w_i 取较小的值, 在进化后期取较大的值, 因此设计 w_i 的求取方法如下:

$$w_i = w'_{\max} - (\max \text{gen} - \text{gen}) \times \frac{(w'_{\max} - w'_{\min})}{\max \text{gen}},$$

其中, $w'_{\max} = 0.01$, $w'_{\min} = 0.001$, $\max \text{gen}$ 是给定最大进化代数, gen 是当前进化代数, $\text{Sign}(\cdot)$ 为符号函数, 使粒子变量能增加也能减少, $\text{randn}(\cdot)$ 是确定方向的随机变量, $\text{rand}(\cdot)$ 是均值为零, 方差为 1 的随机变量, 用于限制位置的变化幅度. 特别注意, 为避免较多的维数变化引起粒子的巨大破坏, 每个粒子每次变化仅限制在选中的一维. 故粒子群的更新方程变为

$$V_i(K+1) = wV_i(K) + \zeta_1 \text{rand}(\cdot) (X_{pbest_i}^*(k) - X_i(k)) + \zeta_2 \text{rand}(\cdot) (X_{gbest_i}(k) - X_i(k)). \quad (2)$$

4 算法的实现流程和收敛性说明

算法的流程如图 1 所示.

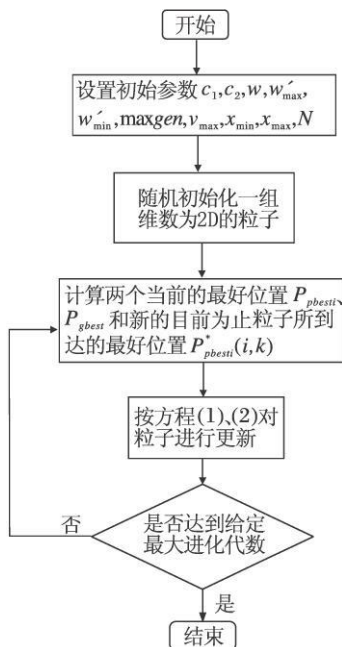


图 1 算法流程图

Fig 1 Flowchart for Proposed algorithm

与文献[9]类似, 将粒子群的更新方程表示为状态方程的形式:

$$V_d(t+1) = wV_d(t) + \phi_1 (X_{dp}(t) - X_d(t)) + \phi_2 (X_{dg}(t) - X_d(t)),$$

则粒子运动的递推方程变为

$$X_d(t+1) = (1 + w - \phi_1 - \phi_2) X_d(t) - wX_d(t-1) + \phi_1 X_{dp}(t) + \phi_2 X_{dg}(t).$$

此方程表示成矩阵方程形式如下:

$$\begin{bmatrix} X_d(t+1) \\ X_d(t) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + w - \phi_1 - \phi_2 & -w & \phi_1 X_{dp} + \phi_2 X_{dg} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_d(t) \\ X_d(t-1) \\ 1 \end{bmatrix}, \quad (3)$$

其中, X_{dp} 、 X_{dg} 分别为个体运动到当前时刻的最优值和当前代最优解, $\phi_1 = \zeta_1 \text{rand}(\cdot)$, $\phi_2 = \zeta_2 \text{rand}(\cdot)$, 式(3)特征方程为

$$(1 - \lambda)(w - \lambda(1 + w - \phi_1 - \phi_2) + \lambda^2) = 0$$

特征根

$$\lambda_1 = \frac{1 + w - \phi_1 - \phi_2 + \zeta}{2}, \quad \lambda_2 = \frac{1 + w - \phi_1 - \phi_2 - \zeta}{2},$$

其中

$$\zeta = \sqrt{(1 + w - \phi_1 - \phi_2)^2 - 4w}.$$

假定初始状态 $X_d(0)$, $X_d(1)$ 已知, 则方程(3)的解为

$$X_d(t) = k + k_2 \lambda_1^t + k_3 \lambda_2^t,$$

其中

$$k = \frac{\phi_1 X_{dp}(t) + \phi_2 X_{dg}(t)}{\phi_1 + \phi_2},$$

$$k_2 = \frac{\lambda_2 (X_d(0) - X_d(1)) + X_d(2) - X_d(1)}{\zeta(\lambda_1 - 1)},$$

$$k_3 = \frac{\lambda_1 (X_d(1) - X_d(0)) + X_d(1) - X_d(2)}{\zeta(\lambda_2 - 1)}.$$

当 $\|\max(\lambda_1, \lambda_2)\| < 1$ 时

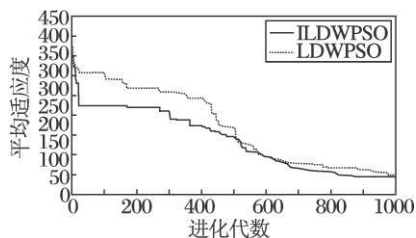
$$\lim_{t \rightarrow \infty} X_d(t) = k = \frac{\phi_1 X_{dp}(t) + \phi_2 X_{dg}(t)}{\phi_1 + \phi_2}.$$

由于本文方法不改变粒子群算法的基本框架, 其收敛点为

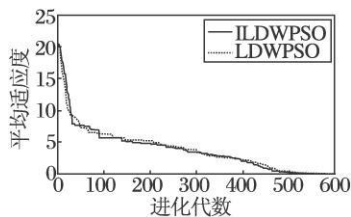
$$\lim_{t \rightarrow \infty} P(t) = \frac{\phi_1 P_{pbest}^*(t) + \phi_2 P_{gbest}(t)}{\phi_1 + \phi_2}.$$

5 仿真实验

为验证算法的有效性, 分别对 2 个典型函数用 3



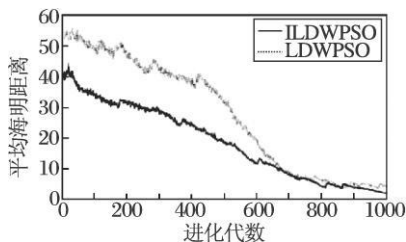
(a) 函数 1
(a) Function 1



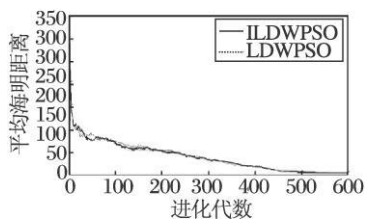
(b) 函数 2
(b) Function 2

图 2 2次实验的平均适应度比较

Fig 2 Comparison of average fitness value between 2 experiments



(a) 函数 1
(a) Function 1



(b) 函数 2
(b) Function 2

图 3 2次实验的平均海明距离比较

Fig 3 Comparison of average Hamming distance between 2 experiments

如下.

1) 函数 1.

$$F_{Ras}(x) = \sum_{i=1}^n (x_i - 10 \cos(2\pi x_i) + 10),$$

$$-5.12 \leq x_i \leq 5.12$$

2) 函数 2

$$F_{Ack}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n |x_i|}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e,$$

$$-32 \leq x_i \leq 32.$$

实验中以其函数值作为适应度函数, 目的是求取 2 个函数的最小值, 由于篇幅限制, 本文仅对线性下降权重粒子群方法 (Linear Decreasing Inertia Weight PSO, LDWPSO) 给出 30 维时 2 函数实验的平均最好适应度变化 (见图 2) 和平均海明距离变化图 (见图 3).

对 PSO 及其改进方法, 相同参数为, $\zeta = \xi = 2$, $w = 0.55$, $w'_{max} = 0.01$, $w'_{min} = 0.001$, $maxgen = 1000$ 对函数 1, $v_{max} = 2$, $x_{min} = -5.12$, $x_{max} = 5.12$, $N = 70$ 对函数 2, $v_{max} = 5$, $x_{min} = -32$, $x_{max} = 32$, $N = 50$, $maxgen = 600$ 对 LDWPSO 及其改进方法, 除权重采用

$$w = w_{max} - gen \times \frac{w_{max} - w_{min}}{maxgen},$$

其它参数与 PSO 一样. 对阶梯形群体粒子群算法 (Ladder PSO, LPSO)^[10], 函数 1 在实验中 $R = 20$, $T = 50$, $N = 100$, $\lambda = 0.2$, $\eta = 0.1$, 函数 2 在实验中 $R = 12$, $T = 50$, $N = 50$ 其它参数与函数 1 实验相同.

每组实验进行 60 次, 函数分别为 10、20、30 维, 其结果如表 1. 表 1 中 ILDWPSO, IPSO 分别表示改进的 LDWPSO, PSO. 对函数 1 在 10、20 维时, 6 种方法都能收敛到接近全局最优解. 对 30 维函数, 只有 LPSO 收敛到接近全局最优解, PSO 的收敛精度高于 LDWPSO. 从表中还可看出, 对于本文方法而言, 其收敛精度和均方差都优于 LDWPSO. 最好个体的平均适应度变化如图 2 为显示本文方法中粒子群的多样性, 用粒子的平均海名距离衡量粒子群的多样性, 实验结果如图 3. 由图可看出, 对同类方法而言, 本文方法有较好的多样性、算法精度较高. 由于在粒子群更新时, 根据粒子适应度的大小, 设计粒子飞翔时的认知域, 对适应度大的粒子其认知性能较好、认知半径较大, 这样在进化的后期, 粒子群依然保持良好的多样性, 使得粒子群不易收敛到局部极小点, 有利于算法收敛精度的提高.

种方法进行实验, 其中第 3 种方法阶梯形粒子群方法是作者在文献 [10] 中提出. 所用的 2 个函数

表 1 3种方法及其改进方法的实验结果

Table 1 Experimental results of 3 methods and their improved methods

维数	函数 1				函数 2				
	最好解	最差解	均方差	平均解	最好解	最差解	均方差	平均解	
LDWPSO	10	0.0073	0.0100	0.0031	0.0084	1.9849	4.9899	2.1482	3.0055
	20	0.0091	0.0098	0.0002	0.0096	15.9196	63.7510	34.0063	24.8740
	30	0.0169	1.8572	0.2002	0.0299	43.8121	61.7623	28.3306	46.7668
ILDWPSO	10	4.7595×10^{-5}	8.4854×10^{-5}	3.1266×10^{-5}	6.2873×10^{-5}	1.9849	4.9748	2.0368	3.0045
	20	1.0441×10^{-4}	1.9636×10^{-4}	0.5731×10^{-4}	1.9636×10^{-4}	13.0732	46.2587	26.6614	19.9748
	30	0.0093	1.3874	0.0316	0.0144	38.5412	55.4628	18.6792	43.7818
PSO	10	0.0081	0.0098	0.0012	0.0094	1.9899	2.9849	1.0368	2.4503
	20	0.0088	0.0098	0.0018	0.0089	18.7460	78.6425	33.6633	34.7882
	30	0.0096	0.0099	0.0001	0.0098	49.5784	91.3380	38.6731	58.9926
IPSO	10	2.5866×10^{-5}	4.2838×10^{-5}	1.3876×10^{-5}	3.6811×10^{-5}	1.9899	4.9899	1.8526	2.8874
	20	5.3244×10^{-4}	7.8637×10^{-4}	1.6843×10^{-4}	6.6329×10^{-4}	17.6899	58.7762	28.3761	23.6605
	30	0.0025	0.0037	0.0004	0.0028	36.3701	54.3398	8.9337	45.6226
LPSO	10	1.5410×10^{-9}	1.5410×10^{-9}	0.0000	1.5410×10^{-9}	1.9849	2.9849	0.4894	2.4428
	20	2.0476×10^{-9}	4.0476×10^{-9}	0.2864×10^{-9}	2.2971×10^{-9}	10.4782	22.8840	7.3638	15.7708
	30	9.3107×10^{-6}	1.3429×10^{-4}	8.7433×10^{-5}	1.3305×10^{-5}	30.6518	45.9902	9.3837	39.6633
ILPSO	10	0.5514×10^{-9}	0.5788×10^{-9}	0.0012×10^{-9}	0.5546	0.0136	1.0642	0.1366	0.0422
	20	1.4989×10^{-9}	5.0634×10^{-9}	0.2249×10^{-9}	2.2108×10^{-9}	9.9883	14.9244	2.6609	11.6637
	30	8.3304×10^{-6}	1.7687×10^{-5}	0.8823×10^{-5}	1.0827×10^{-5}	24.8803	42.0766	8.3765	32.9917

6 结束语

本文从改进粒子更新方程出发,设计一种基于个体认知域的改进方法,对粒子更新方程中的粒子运动到目前为止的最好位置进行修改.方法具有自适应性,通过3种不同粒子群方法的实验,证明方法是有效的.

参 考 文 献

- [1] Kennedy J, Eberhart R C. Particle Swarm Optimization // Proc of the IEEE International Conference on Neural Networks, Perth, Australia, 1995: 1942-1948.
- [2] Seo J H, In C H, Heo C G, et al. Multimodal Function Optimization Based on Particle Swarm Optimization. IEEE Trans on Magnetice, 2006, 42(4): 1095-1098.
- [3] Yi Da Ge, Xu Yun. An Improved PSO-Based ANN with Simulated Annealing Technique. Neurocomputing, 2005, 63: 527-533.
- [4] Chatterjee A, Siarry P. A PSO-Aided Neuro-Fuzzy Classifier Employing Linguistic Hedge Concepts. Expert Systems with Applications, An International Journal, 2007, 33(4): 1097-1109.

- [5] Peram T, Veeramachaneni K, Mohan C K. Fitness-Distance-Ratio Based Particle Swarm Optimization // Proc of the IEEE Swarm Intelligence Symposium, IndianaPolis, USA, 2003: 174-181.
 - [6] Kaewkarnerdpong B, Peter J B. Perceptive Particle Swarm Optimization: An Investigation // Proc of the IEEE Swarm Intelligence Symposium, Pasadena, USA, 2005: 169-176.
 - [7] Yang Chuming, Simon D. A New Particle Swarm Optimization Technique // Proc of the 18th International Conference on Systems Engineering, Las Vegas, USA, 2005: 164-169.
 - [8] Baskar S, Sugathan P N. A Novel Concurrent Particle Swarm Optimization // Proc of the Congress on Evolutionary Computation, San Diego, USA, 2004: 792-796.
 - [9] van den Bergh F, Engelbrecht A P. A Study of Particle Swarm Optimization Particle Trajectories. Information Sciences, 2006, 176(8): 937-971.
 - [10] Chen Dehao, Zhao Chunxia. Ladder Particle Swarm Optimization and Application in Function Optimization. Journal of System Simulation, 2007, 19(24): 5659-5662. (in Chinese)
- (陈得宝, 赵春霞. 阶梯型粒子群算法及在函数优化中的应用. 系统仿真学报, 2007, 19(24): 5659-5662)