

基于惯性权重非线性动态变化的微粒群算法^{*})

王 辉 钱 锋

(华东理工大学化学工程联合国家重点实验室 上海 200237)

摘 要 本文提出一种惯性权重非线性动态变化的微粒群算法(NDPSO),根据微粒在群体中距群体最优位置的远近,选择不同的惯性权重非线性下降指数,每个微粒根据个体状况选择不同的惯性权重。在 NDPSO 算法中,距最优位置较远区域惯性权重选择较小的下降指数,这样利于微粒较快地飞向群体最优位置,提高算法的全局搜索性能;当微粒飞到距最优位置较近区域时,惯性权重采用较大的下降指数,这样微粒在此区域进行细致的搜索,提高算法的收敛速度。为了研究 NDPSO 算法的性能,对几种典型高维非线性函数进行了测试。测试结果表明,与其它几种典型的微粒群算法相比,NDPSO 明显地提高了算法的性能。

关键词 微粒群算法,惯性权重,非线性

Improved Particle Swarm Optimizer Based on Nonlinear Inertia Weight Dynamic Changing

WANG Hui QIAN Feng

(State Key Laboratory of Chemical Engineering, East China University of Science and Technology, Shanghai 200237)

Abstract This paper proposes a particle swarm optimizer (PSO) with nonlinear inertia weight dynamic changing (NDPSO) to improve the speed of convergence and fine tune the search in the multidimensional space. The decreasing exponent of inertia weight is selected by the region which particle stays in the swarm and each particle will be having different inertia weight according to the place which it stays. If the particle is far from the optimal point of the swarm the exponent of inertia weight will be smaller in order to fly to the optimal position quickly, on the other hand if particle flies near to the optimal position the exponent of inertia weight will be larger for particle searching for the optimal value carefully. The propose NDPSO is tested with a set of 4 benchmark problems and compares with linearly decreasing weight PSO (SPSO), nonlinearly decreasing weight PSO (C&S-PSO). Experiment results indicate that the proposed PSO improves the search performance on the benchmark functions significantly.

Keywords Particle swarm optimizer, Inertia weight, Nonlinear

1 引言

微粒群算法(Particle Swarm Optimizer, PSO)是由 Kennedy 与 Eberhart 模拟鸟类群体飞行特性于 1995 年提出的一种群体智能算法^[1]。微粒群算法在解决一些问题上与其它进化算法相比具有参数设置简单、收敛速度快的优点^[2],但也有易陷入局部最优和算法后期震荡现象。针对这些问题,国内外学者做了大量的研究,以提高算法的性能。

Eberhart 与 Shi 提出引入惯性权重来体现微粒以前的速度对当前速度的影响^[3],但是固定值的惯性权重只能对部分特定问题有效。为了提高算法性能,Eberhart 与 Shi 又提出了线性惯性权重递减微粒群算法(称为标准微粒群算法 Standard PSO—SPSO)^[4],即在算法的初始阶段惯性权重取值较大,这样有利于算法有较快的收敛速度,找到最优解所在的子区域。随着搜索的进行,惯性权重逐渐下降,这样有利于算法在最优解所在的子区域进行仔细搜索。SPSO 算法提高了算法的性能。但是微粒群算法在实际中的搜索过程是非线性和高度复杂性的,线性下降惯性权重并不能正确反映实际的优化过程。于是 Chatterjee 与 Siarry 提出非线性惯性权

重微粒群算法(称为 C&S-PSO)^[5],即惯性权重按非线性规律下降,对某些测试函数的研究表明,改进后的算法提高了算法的收敛速度。但是同 SPSO 一样,采用静态非线性下降惯性权重方法也不能准确和有效地反映搜索的过程,特别是对于一些问题,算法的收敛精度不能得到提高。

基于上述不足,本文提出一种惯性权重非线性动态改变的微粒群算法(nonlinear inertia weight dynamic changing in particle swarm optimizer—NDPSO)。在此算法中,根据每个微粒在群体中所处的不同区域,动态选择惯性权重不同的非线性下降指数,提高算法的性能。

2 标准微粒群算法

在一个 D 维搜索空间中,有 m 个微粒组成一微粒群,其中第 i 个微粒的空间位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,第 i 个微粒经历的最优位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$,每个微粒的飞行速度为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$,其中 $i = 1, 2, \dots, m$ 。所有微粒经过的最优位置为 $P_g = (g_{p1}, g_{p2}, \dots, g_{pD})$ 。每一代微粒在 d 维空间中($1 \leq d \leq D$)的速度和位置根据以下公式更新:

$$v_{id}(k+1) = w \cdot v_{id}(k) + c_1 \text{rand}_1() (p_{id} - x_{id}(k)) +$$

^{*} 国家杰出青年科学基金(60625302)、国家 973 计划(2002CB3122000)、国家 863 计划项目(20060104Z1081)、上海市科委重大基础研究(05DJ14002)、上海市自然科学基金(05ZR14038)基金。王 辉 博士生,主要从事计算智能、智能控制理论与应用的研究;钱 锋 教授,博士生导师,主要从事工业过程先进控制、优化与故障诊断、人工智能在流程工业过程建模、控制和优化中的应用等方面的研究。

$$c_2 \text{rand}_2() (p_{gd} - x_{id}(k)) \quad (1)$$

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

$$V_{\max} \geq V_i \geq -V_{\max} \quad (3)$$

其中, w 为惯性系数, 表明历史的速度信息对目前速度的影响。当 w 较大时, 可以加大微粒群的搜索空间, 提高搜索的全局性能; w 较小时, 微粒的局部搜索性能得到改善。 c_1 、 c_2 为正的加速度常数, $\text{rand}_1()$ 、 $\text{rand}_2()$ 为 $[0, 1]$ 之间的随机数。在(1)式中, 第二项表明微粒个体的认知能力, 来源于微粒自身的经验和思考; 第三项表明社会认知能力, 表明微粒间的信息共享和相互作用。微粒的飞行范围和飞行速度受最大、最小位置和最大、最小速度限制。

3 基于惯性权重非线性动态变化的微粒群算法

3.1 微粒群中不同区域的划分

研究表明, 群聚生活的生物个体间具有类似自然界中的排斥和吸引现象^[6], 远离群体区域为吸引区。当个体处于此区域时, 其生存受到饥饿、严寒以及来自其它动物捕食等威胁; 据群体较近的距离为排斥区, 此区域具有较多的食物和较为安全的外部环境。因此, 在微粒群中, 为了摆脱危险, 微粒就要加速飞向群体。当微粒处于距群体较近时, 微粒进入较为安全的区域。当其进一步向群体中心飞行时, 受到来自其它微粒排斥力的影响, 微粒就会放慢飞行的速度, 寻找最优位置。

在 2 维空间内, 吸引和排斥区域的划分如图 1 所示。

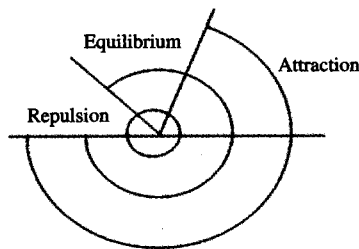


图 1 微粒在群体中所处区域划分

在图 1 中, 同心圆中心为要优化的目标值。“Attraction”标识的圆环为微粒距圆心最远距离——吸引距离, “Repulsion”标识的圆环为微粒距圆心最近距离——排斥距离。在“Attraction”与“Repulsion”中间位置的环形为平衡距离“Equilibrium”。吸引距离与平衡距离之间的为吸引区域, 平衡距离与排斥距离之间的为排斥区域。

3.2 惯性权重非线性变化的动态选择

在标准微粒群算法中, 惯性权重随算法迭代动态变化可表示为

$$w(\text{iter}) = ((\text{iter}_{\max} - \text{iter}) / \text{iter}_{\max})^n \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (4)$$

其中, iter 表示当前迭代次数, iter_{\max} 表示最大迭代次数。惯性权重从初始值 w_{initial} 随着指数 n 的不同值, 速率非线性下降, 在 $\text{iter} = \text{iter}_{\max}$ 时减小到最小值 w_{end} 。定义 $u = \frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max}}$, 则(4)式可表示为

$$w(\text{iter}) = u^n \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (5)$$

因此, 微粒群算法的速度更新为

$$v_{id}(k+1) = w(\text{iter}) \cdot v_{id}(k) + c_1 \text{rand}_1() (p_{id} - x_{id}(k)) + c_2 \text{rand}_2() (p_{gd} - x_{id}(k)) \quad (6)$$

在(5)式中, 当 $n=1$ 时, 为 ShiY 与 Eberhart 提出的惯性权重线性下降的标准微粒群算法, 即 SPSO。当 n 取不等于 1

的某一固定值时, 为 Chatterjee 与 siarry 所提出的惯性权重非线性下降的微粒群算法 (C&S-PSO)。

对于 NDPSO, 根据微粒所处的不同区域, 惯性权重按不同的非线性指数下降。当微粒群处于远离群体的吸引区域时, 惯性权重下降速度减慢, 这样惯性权重在这一阶段具有较大值, 微粒将以较快的速度飞向群体最优位置 (优化最优值)。当微粒逐渐靠近目标最优值时, 微粒进入排斥区, 惯性权重下降加快, 微粒在最优值所在区域对优化目标进行更加细致的搜索。

因此, 惯性权重下降的指数大小根据微粒所处位置按式(7)~(10)进行调整:

$$\textcircled{1} \text{ 当 } |x_{id} - att_d| < |equ_{ud}| \text{ 时, } n=0.9, \text{ 即}$$

$$w(\text{iter}) = u^{0.9} \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (7)$$

$$\textcircled{2} \text{ 当 } |x_{id} - att_d| \geq |equ_{ud}| \text{ 时, } n=1.1$$

$$w(\text{iter}) = u^{1.1} \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (8)$$

$$\textcircled{3} \text{ 当 } |x_{id} - ATT_d| < |equ_{ud}| \text{ 时, } n=0.8$$

$$w(\text{iter}) = u^{0.8} \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (9)$$

$$\textcircled{4} \text{ 当 } |x_{id} - ATT_d| \geq |equ_{ud}| \text{ 时, } n=1.2$$

$$w(\text{iter}) = u^{1.2} \cdot (w_{\text{initial}} - w_{\text{end}}) + w_{\text{end}} \quad (10)$$

其中, att 为个体所发现的最大距离, ATT 为整个群体所发现距中心的最大距离。 $equ_{ud} = (att_d - p_{id}) / 2$ 为中间点 (平衡距离), 是目前个体微粒发现的距中心点最大距离和最小距离的中间距离。 $equ_{ud} = (ATT_d - p_{gd}) / 2$ 为微粒群所发现的最大与最小距离的中间距离。

3.3 NDPSO 算法的计算过程

根据上述分析, NDPSO 算法的计算过程为:

Step1: 初始化所有微粒, 随机设置微粒的速度和位置, 每个微粒的最优位置设为其初始位置, 其中最好的设置为全局最优。

Step2: 评价每个微粒的位置, 计算每个微粒的目标函数值。

Step3: 对于每个微粒计算其位置和其经历过的最好位置进行比较, 优于最好历史位置则替代其历史最好位置。

Step4: 对于每个微粒将其位置与其群体最好位置进行比较, 优于群体最好位置则替代之。

Step5: 判断微粒在群体中所处的区域, 根据(7)~(10)式得出惯性权重下降指数 n , 根据(5)式计算当前的惯性权重。

Step6: 根据(6)式计算微粒当前的速度。

Step7: 根据(2)式计算微粒当前的位置。

Step8: 检查终止条件, 若条件满足则终止迭代, 否则返回 Step2。

4 实验和结果分析

为了研究算法的性能, 取 Sphere、Rosenbrock、Rastrigin 和 Griewank 四个难优化的非线性函数进行测试。四个测试函数维数均取 30 维, 函数介绍如下:

① Sphere 函数, 其表达式为

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

② Rosenbrock 函数, 其表达式为

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \\ x_i \in [-30, 30]$$

Sphere 与 Rosenbrock 由 K. A. DeJong 提出。Sphere 为单峰值函数, 在 $x_i = 0$ 时达到极小值; Rosenbrock 函数是非凸、病态函数, 难以进行全局极小化, 在 $x_i = 1$ 时达到极小值。

③ Rastrigin 函数为多峰值函数, 在 $x_i = 0 (i=1, 2, \dots, n)$

时达到全局极小点,在 $S=\{x_i \in (-5.12, 5.12), i=1, 2, \dots, n\}$ 范围内大约存在 $10n$ 个局部极小点。其表达式为

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

$$x_i \in [-5.12, 5.12]$$

④ Griewank 函数为多峰值函数,全局极小值在 $x_i=0 (i=1, 2, \dots, n)$ 时达到,局部极小值在 $x_i \approx \pm k \cdot \pi \sqrt{i}, i=1, 2, \dots, n; k=0, 1, 2, \dots, n$ 时达到。其表达式为

$$f(x) = \frac{1}{400} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos((x_i - 100)/\sqrt{i}) + 1$$

$$x_i \in [-600, 600]$$

算法参数设置:加速因子 c_1, c_2 取 2.0,惯性权重从 0.9 下降到 0.4。群体规模设为 100, V_{max} 与 V_{min} 分别取所优化问题变量最大取值。算法最大迭代次数为 2000,对每个测试函数运行 100 次求其平均值。

NDPSO 与 C&S-PSO, SPSO 所求测试函数均值(方差)见表 1。

表 1 不同微粒群算法所求测试函数均值(方差)

函数	算法		
	SPSO	C&S-PSO ($n=1, 2$)	NDPSO
Sphere	8.15×10^{-12} ($9.57E-12$)	3.61×10^{-26} (5.62×10^{-25})	1.17×10^{-36} (5.26×10^{-35})
Rosenbrock	50.79 (40.56)	53.08 (42.16)	23.84 (21.26)
Rastrigin	30.19 (11.52)	29.76918 (11.03)	24.68 (10.15)
Griewank	1.4×10^{-2} (1.6×10^{-2})	1.53×10^{-2} (2.58×10^{-2})	8.38×10^{-3} (5.6×10^{-3})

从表 1 可以看出,同 SPSO 和 C&S-PSO 相比,NDPSO 算法对单峰值函数 Sphere, Rosenbrock 和多峰值函数 Rastrigin, Griewank 都能获得更好的解。

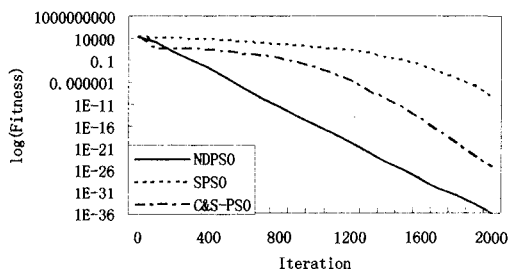


图 2 Sphere 函数

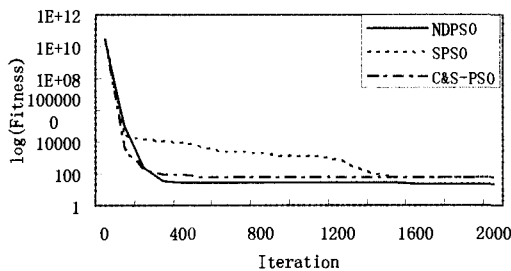


图 3 Rosenbrock 函数

图 2~5 给出了以上三种微粒群算法对四个测试函数的均值的求解过程,图中纵坐标为所优化函数对数值,横坐标为迭代次数。从图中可以看出,与 SPSO 和 C&S-PSO 相比,

NDPSO 在算法的初始阶段具有较快的收敛速度,随着搜索的进行,算法的收敛精度也得到了提高。

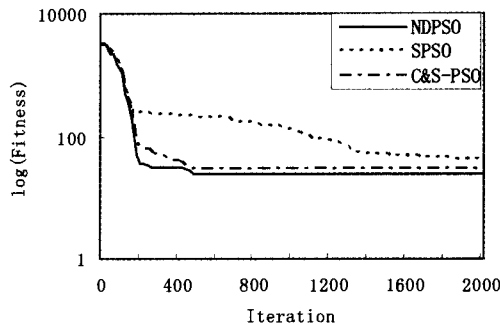


图 4 Rastrigin 函数

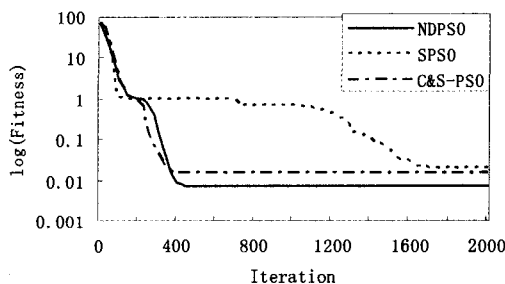


图 5 Griewank 函数

结论 本文提出了一种惯性权重非线性动态变化的微粒群算法,根据微粒个体在群体中所处不同区域,选取惯性权重的非线性下降指数。在吸引区域时,惯性权重非线性下降指数取值较小,这样有利于提高算法的全局搜索性能,随着微粒靠近最优位置,惯性权重非线性下降指数取值较大,有利于提高算法的收敛速度。为了防止因惯性权重下降指数过小和过大引起算法的性能变差,惯性权重的非线性下降指数取 $[0.8, 1.2]$ 之间较为合适。通过四个典型的测试函数的实验结果对比分析表明,改进的微粒群算法(NDPSO)提高了算法的收敛速度和收敛精度,改善了算法的性能。

参考文献

- Kennedy J, Eberhart R C. Particle swarm optimization[A]. Proc. of the IEEE International Conference on Neural Networks (Perth, Australia) [C]. IEEE Service Center, Piscataway, NJ, 1995(4):1942~1948
- Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources[A]. In: Proc. of the IEEE International Conference on Evolutionary Computation [C]. Seoul, Korea, IEEE Press, 2001. 81~86
- Shi Y, Eberhart R C. A modified particle swarm optimizer[A]. In: Proc. of the IEEE International Conference on Evolutionary Computation [C]. Piscataway NJ; IEEE Press, 1998. 69~73
- Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[A]. In: Proc. of Evolutionary Programming VII. Lecture Notes in Computer Science [C], Berlin: Springer, 1998(1447): 591~600
- Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers & Operations Research, 2006(33): 859~871
- Warburton K, Lazarus J. Tendency-distance models of social cohesion in animal groups[J]. Journal of Theoretical Biology, 1991 (150): 473~488