

# 一种基于距离行为模型的改进微粒群算法

王 辉, 钱 锋

WANG Hui, QIAN Feng

华东理工大学 化学工程联合国家重点实验室, 上海 200237

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

E-mail: zgwanh@163.com

WANG Hui, QIAN Feng. Improved particle swarm optimizer with behavior of distance models. Computer Engineering and Applications, 2007, 43(30): 30-32.

**Abstract:** Proposes an improved PSO (Particle Swarm Optimizer) called BDPSO (Behavior of Distance PSO) that particles do different behavior due to different regions of the swarm. In the attraction region particles speed up fly to the best position of the swarm and in the repulsion region they fly at normal speed. Some benchmark problems are tested for the performance of BDPSO and results show that BDPSO performances better than PSO both on convergence rate and accurate solutions.

**Key words:** Particle Swarm Optimizer (PSO); attraction; repulsion; Behavior of Distance PSO (BDPSO)

**摘 要:** 提出一种基于距离行为模型的改进微粒群算法, 根据微粒所处区域来调整其飞行的速度。在吸引区域微粒加速飞向群体最优位置, 在排斥区域按正常速度飞行。为了研究算法的性能, 对几种典型高维非线性函数进行了测试。研究表明, 与基本微粒群算法相比, 改进后的微粒群算法提高了算法的收敛速度和收敛精度, 改善了算法的性能。

**关键词:** 微粒群算法; 吸引; 排斥; 距离行为微粒群算法

**文章编号:** 1002-8331(2007)30-0030-03 **文献标识码:** A **中图分类号:** TP301

## 1 引言

微粒群算法 (Particle Swarm Optimizer, PSO) 是 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<sup>[1]</sup>。微粒通过自身的位置与整个群体进行对比, 根据个体及整体的信息来调整自己的速度和位置, 使个体保持最优, 群体根据个体表现使队伍保持最优。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<sup>[2]</sup>, 但是微粒群算法也有易陷入局部最优和算法后期震荡现象<sup>[3]</sup>, 针对这些问题国内外学者作了大量的研究, 通过提高算法的性能<sup>[4-8]</sup>。

微粒群算法通过群体中个体之间的信息交流实现寻优, 个体之间的相互关系对整个群体的性能必将产生影响。研究发现, 生物界群体生存的个体之间也具有类似自然界中吸引和排斥现象的吸引和排斥关系。吸引使得个体间距离变小, 群体趋于聚合, 而个体间间距过小时, 个体间的吸引关系就转化为排斥关系<sup>[9, 10]</sup>。基于上述思想, 本文提出一种基于距离行为模型的改进微粒群算法 BDPSO (Behavior of Distance PSO), 即以距微粒距最优位置的距离划分出吸引和排斥区域, 微粒根据自身所处区域执行不同的飞行行为, 提高整个群体寻优的能力。

BDPSO 算法的性能通过几个典型的基准测试函数进行了检验, 结果表明, 其性能明显优于 PSO。

## 2 基本微粒群算法

微粒在空间飞行的时候具有位置和速度, 其速度和位置按以下公式更新:

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

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad V_{\max} \geq v_i \geq -V_{\max} \quad (2)$$

其中, 每个微粒在 D 维空间中运动,  $v_{id}$  为微粒  $i$  的运动速度。  $w$  为惯性权重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,  $p_{id}$  为微粒  $i$  的最优位置 (问题的局部最优解),  $p_{gd}$  为整个微粒群的全局最优位置 (问题的全局最优解),  $x_{id}$  为 D 维空间中微粒所处的坐标位置。在式 (1) 中, 第 2 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 3 项表明社会认知能力, 表明微粒间的信息共享和相互作用。微粒的速度  $v_{id}$  通过  $v_{\max}$  进行限制。

**基金项目:** 国家重点基础研究发展规划 (973) (the National Grand Fundamental Research 973 Program of China under Grant No.2002CB3122000); 国家高技术研究发展计划 (863) (the National High-Tech Research and Development Plan of China under Grant No.20060104Z1081); 上海市自然科学基金 (the Natural Science Foundation of Shanghai of China under Grant No.05ZR14038); 国家杰出青年科学基金 (the National Science Fund of China for Distinguished Young Scholar under Grant No.60625302); 上海市科委重大基础研究基金 (No. 05DJ14002)。

**作者简介:** 王辉 (1972-), 男, 博士研究生, 主要从事计算智能、智能控制理论与应用的研究; 钱锋 (1961-), 男, 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。

### 3 距离行为微粒群算法

在微粒群中,距离群体中心最远的区域为吸引区,当个体处于此区域时其生存受到饥饿、严寒以及来自于其它动物捕食等威胁,为了摆脱危险微粒就要加速飞向群体。当微粒处于据群体中心较近的排斥区域时,因为此区域具有安全的环境和较多的食物,微粒就以正常的速度飞行。

在二维空间内,吸引和排斥区域的划分如图 1 所示。

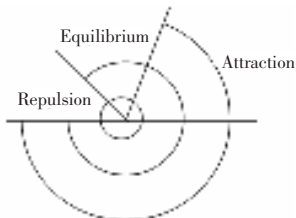


图 1 微粒在群体中所处区域划分

在图 1 中,同心圆中心是为要优化问题的目标值。“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。微粒根据自身当前所处区域进行速度更新:

$$v_{id}(k+1)=w \cdot v_{id}(k)+c_1 \text{rand}_1() \cdot (p_{id}-x_{id}(k))+c_2 \text{rand}_2() \cdot (p_{gt}-x_{id}(k))+T_{id}(k) \quad (3)$$

其中

$$T_{id}(k)=\begin{cases} c_3 \text{rand}_3() \cdot (equ_{ud}-x_{id}), |x_{id}-att_d| < lequ_{ud} \\ c_4 \text{rand}_4() \cdot (equ_{hd}-x_{id}), |x_{id}-ATT_d| < lequ_{hd} \\ 0, \text{otherwise} \end{cases} \quad (4)$$

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, $c_3$  与  $c_4$  均为正常数。 $equ_{ud}=(att_d-p_{id})/2$  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。 $equ_{hd}=(ATT_d-p_{gt})/2$  为微粒群所发现的最大与最小距离的中间距离。

BDPSO 算法的求解过程为:

**步骤 1** 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。

**步骤 2** 评价每个微粒的位置(适应值),计算每个微粒的目标函数。

**步骤 3** 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。

**步骤 4** 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。

**步骤 5** 根据式(3)、式(4)计算微粒当前速度,根据式(2)计算微粒当前位置。

**步骤 6** 检查终止条件,若条件满足则终止迭代,否则返回步骤 2。

### 4 实验和讨论

为了研究算法的性能,取 Rosenbrock,Rastrigin 和 Griewank 3 个进化计算常用的基准函数进行测试,其全局最小值均为 0。其中 Rosenbrock 为单峰函数,检验算法的局部搜索能力;Rastrigin 和 Griewank 为多峰函数,检验算法的综合(全局和局部)搜索能力。

Generalized Rosenbrock 函数为 Eq.(5):

$$f_1(x)=\sum_{i=1}^n [100(x_{i+1}-x_i^2)^2+(x_{i-1})^2], x_i \in [-30, 30] \quad (5)$$

Generalized Rastrigin 函数为 Eq.(6):

$$f_2(x)=\sum_{i=1}^n (x_i^2-10 \cos(2\pi x_i)+10), x_i \in [-5.12, 5.12] \quad (6)$$

Generalized Griewank 函数为 Eq.(7):

$$f_3(x)=\frac{1}{400} \sum_{i=1}^n (x_i-100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i-100}{\sqrt{i}}\right)+1, x_i \in [-600, 600] \quad (7)$$

3 个基准测试函数的维数分别设为 10、20 和 30,相应迭代次数设为 1 000、1 500、2 000,每个实验运行 50 次,群体规模  $m$  值取为 20、40、80、160。 $V_{\max}$  取测试函数自变量取值上界的一半,惯性权重从 0.9 到 0.4 线性下降, $c_1$  与  $c_2$  均为 2, $c_3$  与  $c_4$  均为 0.2。

表 1~表 3 分别为 TDPSO 与 PSO 对函数  $f_1$ 、 $f_2$  和  $f_3$  运算 50 次后的平均适应值。图 2~图 4 分别给出了 2 种算法对函数  $f_1$ 、 $f_2$  和  $f_3$  的优化过程,均取  $m=160$ ,维数为 30,迭代 2 000 次,运行 50 次的平均适应值的对数。

表 1 Generalized Rosenbrock 函数的平均适应值

Population Size	Iteration	Dimension	PSO	BDPSO
20	1 000	10	38.897 1	20.939 5
	1 500	20	86.108 7	57.070 0
	2 000	30	122.661 4	63.207 3
40	1 000	10	15.456 2	10.253 6
	1 500	20	70.303 5	47.004 6
	2 000	30	106.735 4	58.106 2
80	1 000	10	9.018 7	6.634 2
	1 500	20	36.877 4	22.848 1
	2 000	30	47.753 6	26.950 6
160	1 000	10	10.625 3	4.516 0
	1 500	20	41.654 3	26.650 8
	2 000	30	44.785 0	23.770 9

表 2 Generalized Rastrigin 函数的平均适应值

Population Size	Iteration	Dimension	PSO	BDPSO
20	1 000	10	4.238 9	3.452 5
	1 500	20	21.568 8	20.074 3
	2 000	30	50.216 5	34.660 7
40	1 000	10	3.465 3	3.121 8
	1 500	20	16.945 2	14.589 9
	2 000	30	35.732 2	28.121 2
80	1 000	10	2.256 8	1.977 5
	1 500	20	11.965 7	8.238 6
	2 000	30	32.135 8	25.069 0
160	1 000	10	1.352 0	1.002 5
	1 500	20	10.155 7	9.473 6
	2 000	30	26.858 8	23.499 2

通过表 1~表 3 可以看出,引入距离行为模型的 BDPSO 在求解能力上优于 PSO,不论是对考查局部搜索能力的 Rosenbrock 函数还是考查综合搜索能力的 Rastrigin 与 Griewank 函数,BDPSO 总体上都取得了优于 PSO 的结果,特别是当基准测试函数维数和群体规模变大时,改进效果更加明显。

从图 2~图 4 也可以看出,BDPSO 所求解的质量优于 PSO 算法。在优化的开始阶段,BDPSO 具有较快的收敛速度,能较快地找到最优解所在区域;在优化结束阶段,BDPSO 能够获得更好的收敛精度。

表3 Generalized Griewank 函数的平均适应值

Population Size	Iteration	Dimension	PSO	BDPSO
20	1 000	10	0.096 5	0.091 5
	1 500	20	0.035 3	0.020 2
	2 000	30	0.014 7	0.010 4
40	1 000	10	0.098 6	0.081 5
	1 500	20	0.032 8	0.019 3
	2 000	30	0.017 3	0.015 8
80	1 000	10	0.085 3	0.070 1
	1 500	20	0.024 3	0.022 2
	2 000	30	0.015 1	0.008 6
160	1 000	10	0.089 9	0.081 5
	1 500	20	0.029 6	0.021 1
	2 000	30	0.011 4	0.007 8

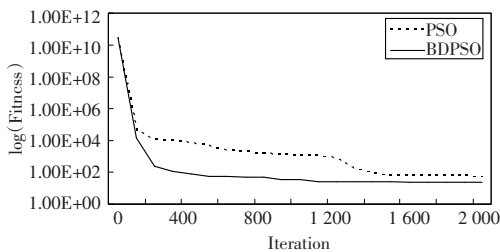


图2 Rosenbrock 函数优化过程对比

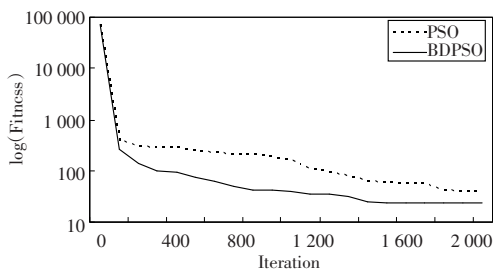


图3 Rastrigin 函数优化过程对比

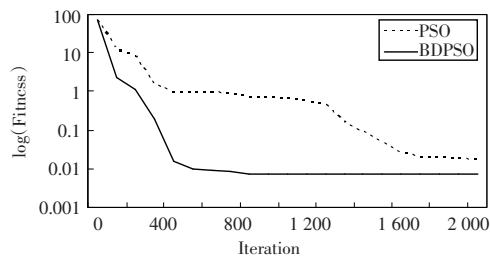


图4 Griewank 函数优化过程对比

(上接 25 页)

体也是分布式,每个机器人是独立的。多机器人系统可以被设计成动作层次和任务层次,流行的基于非平均回报的学习技术,像 Q 学习在动作层次上是有效的,但在任务层次上不是很有效,因为它们不能导致协作,不能根据功能和位置对工作进行划分理解,主要原因是回报值随时间减弱,引起所有机器人首先选择有较大立即回报值的动作,从理论分析和实际机器人实验都表明使用具有全局回报机制的蒙特卡罗学习能解决问题并导致协作,尽管蒙特卡罗学习很简单,但能减少训练样例的使用数量。今后的研究应该在理论上做进一步的分析,并在实验中增加机器人的数量。(收稿日期:2007 年 7 月)

## 参考文献:

[1] Mataric M J. Reinforcement learning in the multi-robot domain[J]. Autonomous Robots, 1997, 4(1): 73-83.

## 5 结束语

本文根据生物群体中存在的个体间吸引和排斥关系,将微粒所处区域划分为吸引和排斥区域,提出基于距离行为模型的微粒群算法 BDPSO。群体中的个体根据自身所处区域调整飞行速度,从而使个体和群体获得最优。BDPSO 与 PSO 算法的性能通过进化计算常用的几个测试函数进行了比较,结果表明,改进的微粒群算法(BDPSO)提高了算法的收敛速度和收敛精度,改善了算法的性能。(收稿日期:2007 年 7 月)

## 参考文献:

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proceedings of the IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, IV: 1942-1948.
- [2] Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources[C]//Proceedings of the IEEE International Conference on Evolutionary Computation Seoul, Korea: IEEE Press, 2001: 81-86.
- [3] Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference[C]//Proceedings of the Evolutionary Programming Conference, San Diego, USA, 1998: 601-610.
- [4] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]//Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway NJ: IEEE Press, 1998: 69-73.
- [5] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[C]//LNCS 1447: Proceedings of Evolutionary Programming VII. Berlin: Springer, 1998, 1447: 591-600.
- [6] Clerc M, Kennedy J. The particle swarm - explosion, stability and convergence in a multidimensional complex space [J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [7] Løbjerg M, Rasmussen T K, Krink K. Hybrid particle swarm optimizer with breeding and subpopulations[C]//Proceedings of the Third Genetic and Evolutionary Computation Conference, 2001, 1: 469-476.
- [8] Parsopoulos K E, Vrahatis M N. UPSO: a unified particle swarm optimization scheme [C]//Lecture Series on Computer and Computational Sciences: Proc Int Conf Comput Meth Sci Eng (ICCMSE 2004), VSP International Science Publishers, Zeist, The Netherlands, 2004, 1: 868-873.
- [9] Warburton K, Lazarus J. Tendency-distance models of social cohesion in animal groups[J]. Journal of Theoretical Biology, 1991, 150: 473-488.
- [10] Breder J M. Equations descriptive of fish schools and other animal aggregations[J]. Ecology, 1954, 35(3): 361-369.
- [2] Cao Y U, Fukunaga A S, Kahng A B. Cooperative mobile robotics: antecedents and directions[J]. Autonomous Robots, 1997, 4(1): 1-23.
- [3] Watkins C J C H, Dayan P. Technical note: Q-learning[J]. Machine Learning, 1992, 8(3/4): 279-292.
- [4] MacKay D J C. Introduction to Monte Carlo methods [M]//Jordan M I. Learning in Graphical Models[S.L.]: MIT Press, 1999: 175-204.
- [5] Mahadevan S. Average reward reinforcement learning: foundations, algorithms, and empirical results [J]. Machine Learning, 1996, 22(1/3): 159-195.
- [6] Raphen B, Shlomo Z, Victor L. Solving transition independent decentralized Markov decision processes[J]. Journal of Artificial Intelligence Research, 2004, 22: 423-455.
- [7] Marek P, Shlomo Z. Average-reward decentralized Markov decision processes[C]//Proceedings of the Twentieth International Joint Conference on Artificial Intelligences (IJCAI-07), Hyderabad, India, 2007: 1997-2002.