文章编号:1001-5132（2008）02-0150-05

求解多峰函数的改进粒子群算法的研究

江宝钏[1]，胡俊溟[2]

（1.宁波大学 信息科学与工程学院，浙江 宁波 315211；2.丽水市缙云县委办公室，浙江 丽水 321400）

摘要：针对标准粒子群算法进行多峰函数优化时存在的易陷入局部极值和搜寻效率低的问题，提出了子种群划分和自适应惯性权重改进方法来求解多峰函数. 根据群体微粒的相似度将粒子群分成子群体，各子群体围绕一个有最佳适应值的群体中心进行建立，并通过几个经典函数进行求解. 实验表明：改进的粒子群算法能快速有效地找到多峰函数的全局最佳值.

关键词：粒子群算法；多峰函数；子群体

中图分类号：TP301.6　　　　文献标识码：A

粒子群优化算法是由 Kennedy 和 Eberthart 等人提出的一种基于种群搜索的进化计算技术[1,2]，此算法目前已经成为演化领域的一个新的分支. 粒子群算法最初是受到飞鸟和鱼类集群活动的规律性启发，利用群体智能建立了一个简化模型，并用组织社会行为代替进化算法的自然选择机制，且通过种群间个体协作来实现对问题最优解的搜索. 近年来粒子群算法粒子群优化算法已广泛地应用于函数优化、神经网络训练、模式分类、参数优化、组合优化、模糊系统控制、机器人路径规划、信号处理、模式识别、旅行商问题以及车间调度等工程领域等多个领域[3,4].

本文研究的是改进粒子群算法求解多峰函数. 多峰函数一般是指有多个峰值点的函数，这些峰值点的函数值可能相同，也可能不同. 而多峰函数在许多方面都有应用，例如求解非线性方程组、线性规划问题及神经网络训练等.

粒子群算法虽然有较多的优点，但其算法有时会出现早期收敛，对具有多个峰值的函数而言，算法寻找的最优解可能是局部最优解而不是全局最优值，因此该算法不能较好地用于求解存在多个局部最优的多峰函数.

1　粒子群算法原理[1]

粒子群优化(Particle Swarm Optimization，PSO)算法是从生物种群行为特性中得到启发，并将它运用于求解优化问题，每个优化问题的解是搜索空间中的一个个粒子. 所有的粒子都有一个被优化的函数决定其适应值，且每个粒子还有一个速度决定它们飞行的方向和距离，然后这些粒子就追随当前的最优粒子在解空间中搜索. PSO 在初始化时将每个个体看作是在 n 维搜索空间中的一个没有质量和体积的微粒，并在搜索空间以一定的速度飞行，该飞行速度根据个体的飞行经验和群体的飞行经验进行动态的调整.

收稿日期：2007-10-22.　　　宁波大学学报（理工版）网址：http://3xb.nbu.edu.cn
基金项目：浙江省自然科学基金（Y106080）.
第一作者：江宝钏（1982－），女，浙江临安人，讲师，主要研究方向：图像处理及智能计算. E-mail: jiangbaochuan@nbu.edu.cn

设 $X_i=(x_{i1},x_{i2},\cdots,x_{in})$ 为粒子 i 的当前位置；$V_i=(v_{i1},v_{i2},\cdots,v_{in})$ 为粒子 i 的当前飞行速度；$P_i=(p_{i1},p_{i2},\cdots,p_{in})$ 为粒子 i 所经过的最好位置，也就是粒子 i 所经历过的具有最好适应值的位置，称为个体最好位置，可表示为 p_{best}. 在群体中所有粒子经过的最好位置用 p_g 表示，也称为 g_{best}.

在每次迭代中，粒子通过跟踪 2 个“极值”来更新自己. 第 1 个极值是粒子本身个体极值 p_i，另外的 1 个极值是全局极值 p_g. 粒子根据如下的公式来完成自己的速度和位置的变化：

$$v_{ij}(t+1)=v_{ij}(t)+c_1r_{1j}(t)(p_{ij}(t)-x_{ij}(t))+c_2r_{2j}(p_{gi}(t)-x_{ij}(t))\text{，}\tag{1}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1)\text{，}\tag{2}$$

其中：j 表示粒子的第 j 维；i 表示粒子的第 i 维；t 表示第 t 代；c_1 和 c_2 为加速常数，通常在 0～2 之间取值，r_1～$U(0,1)$, r_2～$U(0,1)$ 为 2 个相互独立的随机函数. 为减少在进化过程中粒子离开搜索空间的可能性，v_{ij} 通常限定在一定的范围内，即 $v_{ij}\in[-v_{\max},v_{\max}]$.

2　改进粒子群算法求解多峰函数

粒子群算法有时会出现早期收敛，对具有多个峰值的函数，算法寻找的最优解可能是局部最优解，而不是全局最优值. 其原因是在算法运行过程中，如果某粒子发现某个当前最优位置时，其他粒子将迅速向其靠拢；如果该最优位置为局部最优点，粒子群就无法在解空间内重新搜索，算法将陷入局部最优，因此该算法不能较好地用于求解存在多个局部最优的多峰函数.

基于以上情况的考虑，在如何解决算法应用时发生早熟的问题上，本文在粒子群算法中引进子群体的概念，让种群中的个体在不同特定的生存环境中进化，而不是全部聚集在同种环境中. 这样可以使算法在整个解空间中搜索，以找到更多的最优个体，从而得到更好的解.

2.1　多种群协同

协同 PSO 算法基本思想是用 N 个相互独立的粒子群分别在 D 维的目标搜索空间中的不同维度方向上进行搜索. 首先选定划分因子 N 和粒子群的粒子数，将输入的 D 维向量(粒子的速度和位置向量)划分到 N 个粒子群. 前 $D \bmod N$ 个粒子群中粒子位置和速度都是 D/N 维；后 $N-(D \bmod N)$ 个粒子群中粒子的位置和速度向量也是 D/N 维的. 然后在每步迭代的过程中，这些 N 个的粒子种群相互独立地进行状态更新，粒子群之间不共享信息. 在计算值时，将每个粒子群中最优粒子的位置向量拼接起来，组成 D 维向量并代入适应函数计算适应值. 此算法在迭代初期，虽然值下降较为缓慢(收敛速度缓慢)，且其收敛速度与种群所含粒子数目成反比. 但由于协同 PSO 算法采用的是局部学习策略，因此比基本 PSO 算法更容易跳出局部极值，从而达到较高的收敛精度.

基于上述想法发对整个种群进行子种群划分，2 个粒子间用欧氏距离来判定它们的亲缘关系. 粒子群系统中相似的粒子并行地划分成子群体，1 个子群就是同类且具有相似特点的粒子集合. 每个子群体中的粒子都围绕在本群体中具有最优值的粒子周围，该粒子称为“种子”. 利用 2 个粒子间的欧氏距离来判断它们之间的相似程度，距离越远，则 2 个粒子间的相似度越低.

设 2 个粒子分别为：$x_i=[x_{i1},x_{i2},\cdots,x_{in}]$, $x_j=[x_{j1},x_{j2},\cdots,x_{jn}]$，则它们之间的距离可定义为：

$$d(x_i,x_j)=\sqrt{\sum_{k=1}^{n}(x_{ik}-x_{jk})^2}\ .\tag{3}$$

比较粒子与各个有最佳适应值的群体中心间的相似程度，找到相似程度最高的那个种子，则该粒子就属于该子群体.

本文粒子群改进算法(Improved Particle Swarm Optimization，IPSO)在协同 PSO 的基础上，把 N 个粒子分成多个子群体，然后各个子群体独立地搜索最优解. 为了防止某个子群体陷入局部最优，各个

子种群之间需进行信息交换，并采用基于适应度的信息交流的方式，保留最佳适应值，再以原各子种群的最好值点为中心，对整个粒子群重新进行分区，直到满足停止条件(误差达到设定的误差或者迭代次数超过最大允许迭代次数)，然后搜索停止，输出搜索结果. 这样使得搜索不仅能朝着全局较优解的方向飞行，而且能朝着其他多个方向进行. 粒子飞行的方向多样化，可以使得一些孤立且狭长的可行解区域能够被算法高效地探测到，特别是在可行解区域占搜索空间比例非常小的情况，这样可以防止粒子群在搜索过程中陷入局部最优，增加其解的多样性.

2.2 自适应粒子群算法

为改变 PSO 所存在的缺陷，可根据不同问题提出了不同的改善方法. 文献[5]中，Shi 等人引入了惯性权重 w，并更新(1)式得到迭代速度为：

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}(t)(p_{ij}(t)-x_{ij}(t))+ c_2r_{2j}(p_{gi}(t)-x_{ij}(t)), \qquad (4)$$

其中 w 值随着迭代次数的增加从 0.95 递减至 0.4.

研究表明：比较小的惯性 w 有利于算法的收敛，而较大的 w 有利于跳出局部最优解[6]. 算法中本文定义 w 如下：

$$w=\begin{cases} w, & \text{算法在进行,} \\ w+0.1, & \text{当算法解在 } N \text{ 次循环内没有明显改变时,} \ w<w_{\max}. \end{cases}$$

2.3 算法步骤

(1) 初始化设置粒子群的规模、惯性权值、加速系数和最大允许迭代次数、适应值误差及粒子的初始位置和初始速度等.

(2) 根据目标函数来评价各粒子的适应值.

(3) 子群运算：根据距离划分子群，然后在每个子群中进行每个粒子的适应值的比较和更新，如果粒子值优于全局最佳值 g_{best} 的适应值，则 g_{best} 设置为新位置.

(4) 若满足停止条件(其适应值误差达到设定的适应值误差限或者迭代次数超过最大允许迭代次数)，搜索停止后输出搜索结果；否则，利用速度进化方程进行进化，修改更新各粒子，并返回(2).

3 实验结果

3.1 测试函数的选取

为了测试基于子群体粒子群算法在多峰函数问题上的寻优能力和收敛速度，本文选取 6 个典型的多峰值函数作为测试函数[7]，用以上所描述的算法求解这 6 个函数，寻找其全局最优解.

(1) $f_1(x)=-(x_1^2+2x_2^2-0.4\cos(3\pi x_1)-0.6\cdot\cos(4\pi x_2)),\ x_1,x_2\in[-10,10]$.

函数 $f_1(x)$ 是多峰函数，但只有 1 个全局最大点 $[0,0]$，最大值为 1.

(2) $$f_2(x)=0.5-\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{\left(1+0.001(x_1^2+x_2^2)\right)^2},$$
$x_1,x_2\in[-10,10]$.

函数 $f_2(x)$ 有无数个局部极大点，但只有 1 个 $[0,0]$ 为全局最大点，最大值为 1. 此函数的最大峰周围有一圈脊，它们的取值为 0.990 284，因此优化过程中很容易停滞在这些局部极大点.

(3) $$f_3(x)=\left(\frac{a}{b+x_1^2+x_2^2}\right)^2+\left(x_1^2+x_2^2\right)^2,$$
$x_1,x_2\in[-5.12,5.12]$

在函数 $f_3(x)$ 中，$a=3.0, b=0.05, \max f(0,0)=3\,600$，4 个局部极值为 $(-5.12,5.12)$, $(-5.12,-5.12)$, $(5.12,-5.12)$ 和 $(5.12,5.12)$，函数值则为 2 748.78，这是一类全局最优解被次优解所包围，隔断了模式的重组过程，使得搜索长期陷入局部极值点.

(4) $$f_4(x)=0.5+\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2},$$
$x_1,x_2\in[-10,10]$.

函数 $f_4(x)$ 是 schaffer 函数求最小值，函数在 $(x_1,x_2)=(0,0)$ 处有 1 个全局最小值 0.

(5) $f_5(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3,$

$x_1, x_2 \in [-50, 50]$.

(6) $f_6(x) = x_1^2 + 2x_2^2 - 0.3\cos 3\pi x_1 - 0.4 \cdot \cos 3\pi x_2 + 0.7, x_1, x_2 \in [-10, 10]$.

函数 $f_5(x)$ 和 $f_6(x)$ 是 2 个 Bohachevsky 函数. 函数有几个局部极小点,这 2 个函数都在原点 $[0, 0]$ 处有全局极小值为 0.

3.2 实验结果

实验参数选择 c_1 和 c_2，实验取值为 1.8，w 的取值以 0.95～0.4 逐次递减. 取粒子数为 20，运行 500 代，分子群数为 4，每个测试用例均在相同条件下独立运行 50 次，记录其最好结果、最差结果、最小迭代数、最大迭代数及计算出平均结果. 为比较本文提出的 IPSO 算法的有效性，与 PSO 标准算法的结果进行了比较，实验结果见表 1，而 IPSO 算法不同子群数的结果见表 2.

3.3 结果分析

(1) 由表 1 可得改进的 IPSO 粒子群算法比标准 PSO 算法在搜索最优值能力和算法的稳定性方面都要好，改进的 IPSO 粒子群算法在最好结果上更接近于标准值. 其中对于测试函数 $f_2(x)$, $f_3(x)$, $f_4(x)$ 而言，本文算法找到的最好结果的成功次数要比标准 PSO 算法找到结果的成功次数多很多，且平均结果都比标准 PSO 算法好. IPSO 对于标准 PSO 算法，特别是对于测试函数 $f_2(x)$, $f_3(x)$, $f_4(x)$ 而言，本文算法不容易陷入局部极值，而标准 PSO 算法则较易就陷入局部极值中. 根据表 1 可以很清楚地看到在标准 PSO 算法中，运行 50 次，经过 500 次的迭代中 $f_2(x)$ 的极值陷入局部极值 0.990 284，$f_3(x)$ 的极值陷入局部极值 2 748.763 2，$f_4(x)$ 的极值陷入局部极值 0.009 76.

(2) 由表 2 可看到，当粒子数确定时，在一定范围内将粒子群划分的子种群数多比划分子种群数少的得到的结果要好一些. 如 $N = 4$ 的比 $N = 2$ 的求得的极值要更准确，而且迭代次数也少. 但是划分的子种群数目较大时，如 $N = 8$，得到的结果与 $N = 4$ 是差不多，个别函数的平均迭代次数比 $N = 4$ 时大. 可见，当粒子数确定时，子种群划分的数量也不是越多越好.

表 1　IPSO 算法与标准 PSO 实验结果比较

函数	最优值	最小迭代数	最大迭代数	最好结果		平均结果		成功次数	
				IPSO	标准 PSO	IPSO	标准 PSO	IPSO	标准 PSO
$f_1(x)$	max＝1	82	218	0.999 998	0.999 971	0.999 951	0.999 939	50	50
$f_2(x)$	max＝1	48	148	0.999 999	0.999 999	0.999 959	0.990 284	50	17
$f_3(x)$	max＝3 600	138	287	3 600.000 0	3 599.999 954	3 599.999 960	2 748.763 2	50	2
$f_4(x)$	min＝0	64	205	0.000 001	0.000 052	0.000 050	0.009 760	50	9
$f_5(x)$	min＝0	128	235	0.000 001	0.000 012	0.000 049	0.000 058	50	50
$f_6(x)$	min＝0	85	226	0.000 002	0.000 007	0.000 048	0.000 054	50	50

表 2　IPSO 算法不同子群数的结果

函数	最优值	最好结果			平均迭代数		
		N=2	N=4	N=8	N=2	N=4	N=8
$f_1(x)$	max＝1	0.999 994	0.999 998	0.999 998	185	175	175
$f_2(x)$	max＝1	0.999 992	0.999 999	0.999 999	203	113	106
$f_3(x)$	max＝3 600	3 599.999 993	3 600.000 000	3 599.999 998	281	225	227
$f_4(x)$	min＝0	0.000 005	0.000 001	0.000 001	246	148	152
$f_5(x)$	min＝0	0.000 003	0.000 001	0.000 002	207	197	199
$f_6(x)$	min＝0	0.000 003	0.000 002	0.000 001	175	166	165

注：N 代表不同的子群数.

4 结论

本文使用子群体的概念，利用多个引导粒子将其邻近个体聚集形成多子种群，并引导种群中的粒子飞行，从而更有效地测试到搜索空间中孤立的、狭长的可行解区域，并且通过自适应惯性因子的改变，使得算法能够在粒子飞行的过程中将有效改变的方向得到加强，从而更快速地进入可行解区域，使得群体向最优解逼近.

本文改进的粒子群算法避免了粒子群算法的早期收敛，维护了群体的多样性，并提高了全局搜索能力. 典型多峰值函数的求解实验表明：所有的测试函数都能搜索到有效峰值，并且具有良好的收敛稳定性和很高的精度值. 因此，引入子群体的粒子群算法有更强的全局搜索能力和更高的收敛速度，能够高效地寻找到全局最优值，是一种寻优能力、效率和可靠性更高的优化算法，其综合性能比基本 PSO 算法有显著提高.

参考文献：

[1] Kennedy J, Eberhart R C. Particle swarm optimization [C]//Proceedings of Conference on Neural Networks, IEEE Press, 1995:1 942-1 948.

[2] Fogel D B. An introduction to simulated evolutionary optimization[J]. IEEE Transaction Neural Network, 1994, 5(1):3-14.

[3] Yisu J, Knowles J, Hongmei L, et al. The landscape adaptive particle swarm optimizer[J]. Applied Soft Computing, 2008, 8(1):1-7.

[4] 汪定伟, 王俊伟, 王洪峰, 等. 智能优化方法[M]. 北京: 高等教育出版社, 2007.

[5] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]//Proceeding of The IEEE Congress on Evolutionary Computation, IEEE Press, 1999:1 945-1 950.

[6] 王俊伟, 汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报, 2005, 20(2):194-198.

[7] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transaction on Evolutionary Computation, 2004, 8(3):240-255.

Study on an Improved Particle Swarm Optimization Algorithm for Solving Multi-peak Function

JIANG Bao-chuan[1], HU Jun-ming[2]

(1.Faculty of Information Science and Technology, Ningbo University, Ningbo 315211, China;
2.Jinyun Country Office of the Communist Party of China, Lishui 321400, China)

Abstract: In application of particle swarm optimization algorithm, the solution is usually misled to local minima resulting in compromised searching efficiency. To cope this drawback, the division of subpopulation and self-adaptive inertia weight are incorporated to PSO. The particle swarm is divided by subpopulation into subgroups close to the center of the swarm with the best fitness value judged by the similarity of particle swarm. Test of the improved algorithm using six benchmark functions indicates that the algorithm is capable of locating the global optimal value with less time-consumption and higher efficiency.

Key words: particle swarm optimization; multi-peak function; subgroup

CLC number: TP301.6 **Document code:** A

（责任编辑 章践立）