粒子群算法的改进研究

覃建波，陆安山

(钦州学院 物理与材料科学学院，广西 钦州 535000)

[摘 要] 粒子群算法具有在优化过程中需要调整的参数不多，结构简单，收敛速度快等特点。在分析其他改进粒子群算法的基础上,提出了一种新的粒子群协同优化算法。通过测试函数测试表明,新的粒子群协同优化算法明显提高了算法的收敛性能。

[关键词] 粒子群算法; 测试函数; 收敛性能

[中图分类号] TP30 [文献标识码] A [文章编号] 1673-8314(2011)06-0020-05

粒子群优化算法(Particle Swarm Ontimization,PSO)源于对鸟类觅食过程中迁徙和聚集的模拟,是一种基于群智能的自适应随机优化算法。PSO算法原理简单,参数少、进化初期收敛速度快、易于实现,一经提出就引起了众多学者的极大关注,并得到了迅速的发展。目前已经被广泛应用于目标函数优化、组合优化、图像处理、信号处理、决策调度、神经网络训练等许多领域。

与其他群智能算法相比较，PSO算法的优势在于在优化过程中需要调整的参数不多，结构简单，收敛速度快。由于PSO算法易早熟收敛,局部寻优能力较差,许多学者针对基本PSO算法存在的问题提出了多种改进算法。Angeline[1]将选择算子引入PSO算法,选择每次迭代后的较好粒子复制到下一代,以保证每次迭代的粒子群都具有较好的性能; Higash等人[2]提出了自己的变异算法,希望通过引入变异算子跳出局部极值点的吸引,从而提高算法的全局搜索能力,得到较高的搜索成功率; 吕振肃等人[3]则提出了一种新的基于群体适应度方差自适应变异的粒子群优化算法。一些学者还采用模拟退火策略来改进PSO算法。曾建潮等人[4]则提出利用模拟退火算法和禁忌算法等策略在群体中产生随机微粒以保证算法的全局收敛性; 高鹰等[5]将模拟退火思想引入到具有杂交和高斯变异的PSO算法中,给出了一种基于模拟退火的PSO算法。

本文通过分析粒子群算法存在的缺陷,提出了一种改进的粒子群协同优化算法,并用测试函数测试算法的收敛性能。

1 标准粒子群算法[6](PSO)

PSO算法与其他演化算法相似,也是基于群体的[7-8],根据对环境的适应度将群体中的个体移动到好的区域。在粒子群优化算法中,群体中的每个个体都有一个速度来反映位置的改变,粒子根据速度在搜索空间运动。而且每个个体都有一个记忆单元记下它曾经到达过的最优位置。整个寻优过程就是个体根据自己先前到过的最优位置和其领域中其他个体到达的最优位置来改变自己的位置和速度,从而趋向全局最优值的聚集加速过程。在粒子群算法中,问题的解被看作D维搜索空间中的一个没有体积的粒子,在搜索空间中以一定的速度飞行。所有粒子都有一个被优化的目标函数决定的适应度,每个粒子都有一个速度决定他们飞行的方向和距离。PSO算法初始化为一群随机粒子,粒子根据对个体和群体的飞行经验的综合分析来动态调整自己的速度,在解空间中搜索,通过迭代找到最优解。在每次迭代中,粒子通过跟踪两个极值来更新自己:一个是粒子自身目前所找到的最优解 p_{best};另一个是整个群

[收稿日期] 2011-04-28
[基金项目] 钦州学院校级科研项目:基于群智能算法的机器人控制研究(2010XJKY-03A)。
[作者简介] 覃建波(1983-),男,广西来宾人,钦州学院物理与材料科学学院教师,硕士。

体目前找到的最优值 g_{best}。因此，PSO 的进化方程为：

$$v_i(t+1) = w * v_i(t) + c1 * r1 * (p_{best} - x_i(t)) + c2 * r2 * (g_{best} - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

其中，p_{best} 表示第 i 个微粒所经历过的最好位置，g_{best} 表示该群体中所有微粒所经历过的最好位置，g_{best}，p_{best} 在进化过程中根据其适应值而不断更新；w 为惯性常量，w 使粒子具有扩大搜索范围的能力；$c1$，$c2$ 为常数（常取值为 $c1 = c2 = 2$），$r1$，$r2 \in [0,1]$ 为均匀分布的随机数。

2　粒子群算法的改进——粒子群协同优化算法（简称 PSCO）

标准粒子群算法的分析：

首先，从算法的进化方程看，标准 PSO 算法依靠的是群体之间的合作和竞争，粒子本身没有变异机制，因而单个粒子一旦受某个局部极值约束后本身很难跳出局部极值的约束，此时需要借助其它粒子的成功发现来跳出局部极值。可见，PSO 算法的寻优能力主要来自于粒子之间的相互作用和相互影响。如果从算法中去除粒子之间的相互作用和相互影响，则 PSO 算法的寻优能力就变得非常有限。

其次，通过参考其他改进粒子群算法不难发现，各种改进算法都是着眼于如何更有效地用一个粒子群在解空间中搜索最优解。但是分析算法的进化方程（1）和（2）式发现，粒子们在搜索时，总是追逐当前全局最优点，和自己迄今搜索到的最优点，因此粒子们的速度很快降到接近于 0，导致粒子们陷入局部极小而无法摆脱，这种现象被称为粒子群的“趋同性”[9]。这种“趋同性”限制了粒子的搜索范围。要想扩大搜索范围，就要增加粒子群的粒子数，减弱粒子对整个粒子群当前搜索到的全局最优点的追逐。增加粒子数将导致算法计算复杂度增高，而且研究表明通过增加粒子数所获得的效果并不明显[10]，而减弱粒子对全局最优点的追逐又存在算法不易收敛的缺点。因此，根据此种现象本文提出粒子群协同优化（PSCO）算法。

PSCO 算法的基本思想：利用 S 个独立的粒子群进行协同优化。其中前 $S-1$ 个粒子群根据本粒子群迄今搜索到的最优点来修正群中粒子的速度，而第 S 个粒子群则是根据全部粒子群迄今搜索到的最优点修正群中粒子的速度。其中前 $S-1$ 个粒子群称为底层粒子群，第 S 个粒子群称为顶层粒子群。这样既利用底层粒子群的独立搜索来保证寻优搜索过程可以在搜索空间中的较大范围内进行，又利用顶层粒子群追逐当前全局最优点来保证算法的收敛性，从而兼顾优化过程的精度和效率。该粒子群协同优化算法与一般的粒子群协同优化算法不同，这种算法并不要求每个粒子群的粒子数相等，但是要求粒子群的粒子状态更新策略中应有不同的粒子状态更新策略。三种不同的粒子状态更新策略如下：

第一，常规的粒子状态更新策略，如式（1）和（2）所示；

第二，鉴于 PSO 陷入局部极小时，粒子速度接近 0，导致粒子位置保持不变。故在基本 PSO 中增加扰动因子的策略：如果迄今搜索到的全局最优适应值连续 n 步迭代没有更新，则重置粒子的速度。n 是自然数，称为扰动因子；扰动策略的思想是当粒子群陷入局部极小点时，重置粒子的速度（或者位置），强迫粒子跳出局部极小点（与遗传算法中的变异思想类似），从而增强粒子的搜索趋势。在基本 PSO 算法中引入扰动策略，可以进一步改善 PSO 算法的性能。

第三，根据粒子的“趋同性”，粒子们在搜索时，总是追逐当前全局最优点，和自己迄今搜索到的最优点。如果当前全局最优点是全局最优值，那么粒子总是在追逐全局极值；如果当前全局最优点是局部最优值，那么粒子总是在追逐局部极值，从而很有可能错过全局最优极值。因此，为了尽可能搜索到全局最优，故在迭代中前期使粒子在搜索时只追逐自己迄今搜索到的最优点，从而扩大了粒子的全局搜索能力，在后期则使用常规的粒子状态更新策略，以达到全局收敛。

设粒子群算法迭代过程中每一代群体中所有个体为 $P = [P_1 \quad P_2 \quad \cdots \quad P_m]^T$，$m$ 为群体规模，即粒子个数。PSCO 寻优的具体步骤如下：

（1）随机产生 $S-1$ 个群体的初始参数，每个初始参数 $P^{(0)} = [P_1^{(0)} \quad P_2^{(0)} \quad \cdots \quad P_m^{(0)}]^T$ 的速度 $v^{(0)} = [v_1^{(0)} \quad v_2^{(0)} \quad \cdots \quad v_m^{(0)}]^T$ 和位置 $x^{(0)} = [x_1^{(0)} \quad x_2^{(0)} \quad \cdots \quad x_m^{(0)}]^T$，其中每个参数的位置和速度向量中第 i 分量分别在区间 $[X_{\min i}, X_{\max i}]$ 和 $[V_{\min i}, V_{\max i}]$ 中选取。此区间可根据实际要解决的问题或已有的知识和经验给出，以缩小搜索范围。

（2）计算底层 $S-1$ 个群体中每个粒子的适应值。

（3） 对底层的每个群体的每个微粒，将其适应值与其经历过的最好位置 p_{best} 作比较，如果较好，则将其作为当前微粒的最好位置 p_{best}；对每个微粒，将其适应值与此子群体所经历的最好位置 g_{best} 作比较，如果较好，则重新设置 g_{best}。同时比较各个群体的最好位置得到全局最好位置 q_{best}，重新设置各群体的最好位置。

（4） 在 $S-1$ 个粒子群中选择两个粒子群。其中，一个粒子群中的粒子采用粒子状态更新策略中第二种策略即增加扰动因子的策略来变化粒子的位置和速度；另一个粒子群体中的粒子采用粒子状态更新策略中第三种策略来改变粒子的位置和速度。其它群体粒子根据方程（1）和（2）变化粒子的速度和位置；

（5） 满足结束条件（达到预设最大迭代数 $G\max$）即迭代次数 $g>G\max$ 时，粒子群算法的搜寻过程结束，此时对应于最小适应值（或最大适应值）的参数既是搜索到的最优结果；否则转到步骤（2）继续搜索。

3 粒子群协同优化算法（PSCO）收敛性能分析

标准遗传算法（Genetic Algorithm，简称 GA）是一种优秀的智能优化算法，算法比较成熟，而且已经在一些实际问题的解决方面发挥了重要的作用，实践已经表明遗传算法具有很好的处理约束，跳出局部最优，最终得到全局最优解的能力，全局搜索能力较强[11]。

本文为了检验粒子群协同优化算法（PSCO）的性能，采用遗传算法和它进行比较，同时也和改进前的标准粒子群算法进行比较。通过三个测试函数来比较它们的性能。三个测试函数[12]如下：

测试函数（1）：$f1 = 100*\ (x^2 - y)^2 + (1 - x)^2, x, y \in [-2.048, 2.048]$

测试函数（2）：$f2 = x*\sin(4\pi x) - y*\sin(4\pi y + \pi + 1), x, y \in [-1, 2]$

测试函数（3）：$f3 = (3/(0.05 + x^2 + y^2))^2 + (x^2 + y^2)^2, x, y \in [-5.12, 5.12]$

其中，$f1$、$f2$ 和 $f3$ 的最大极值分别为 3905、3.3099 和 3600。

测试函数说明：测试函数 $f1$ 属于比较典型的病态函数问题，常规的算法容易陷入局部极值点，不易获得全局最优解；测试函数 $f2$ 峰值较多，且函数值相差较大，常规的算法不易获得全局最优解；测试函数 $f3$ 属于典型的大海捞针问题，该问题的一个全局最优解被四个局部最优解所包围，绝大部分经典算法都会落入四个局部极值点上而找不到最优解。对于以上三种测试函数，如果某种算法能够快速准确地找出全局最优解，则说明该算法稳定可靠，具有很好的函数优化性能。

图 1、图 2 和图 3 分别为三种算法优化测试函数 $f2$ 所得结果（限于篇幅，其他两种测试函数结果图未给出），从图中可以发现，三种算法都能找到最优解，表明这三种算法都具有很好的收敛性能，但是从图 1 发现，遗传算法部分个体并没有收敛到最优极值点，而图 2 和图 3 中粒子群算法和改进粒子群算法所有粒子都收敛到最优极值点。为了进一步从不同侧面来衡量算法性能：如算法个体都收敛到极值点，说明算法收敛性能好；算法找到全局最优解，说明算法全局寻优性能好，分别采用三种不同算法来优化测试函数 $f1$、$f2$ 和 $f3$ 各 1000 次，统计结果如表 1 所示。

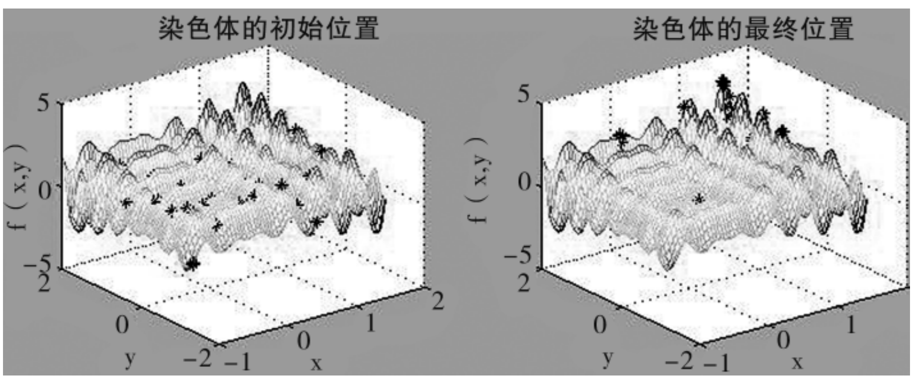

图1 GA 优化 $f2$

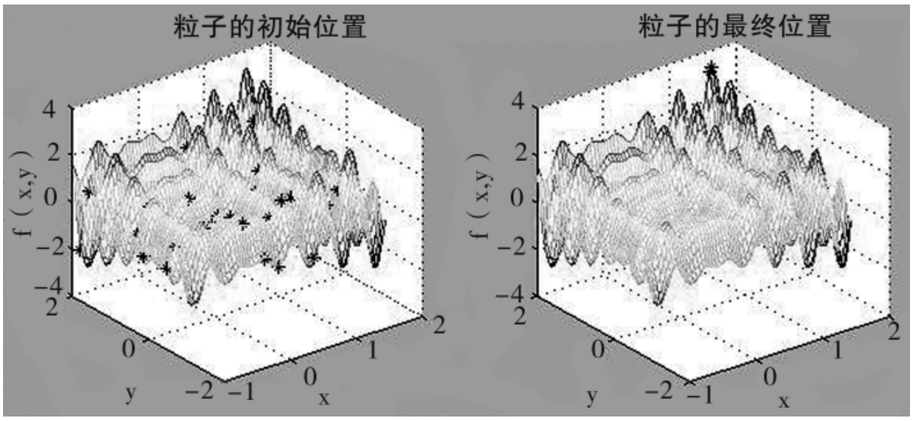

图2　PSO 优化 $f2$

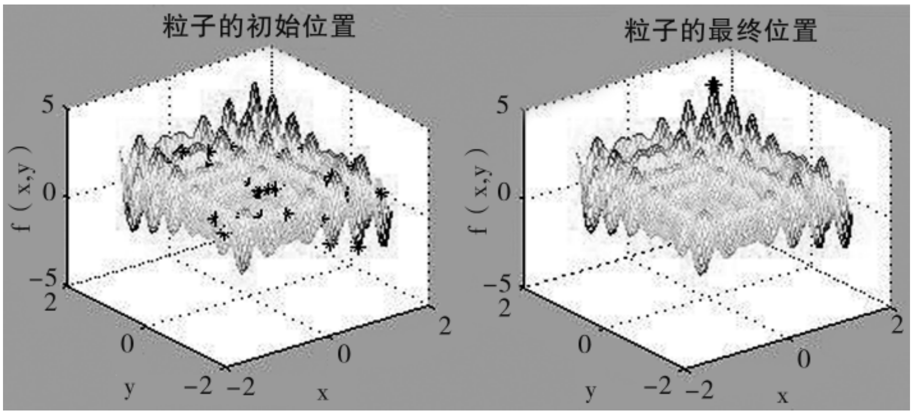

图3　PSCO 优化 $f2$

表1　GA、PSO 和 PSCO 算法优化 $f1$、$f2$ 和 $f3$ 的数据统计

函数	算法	迭代次数	平均收敛率	平均收敛代数
$f1$	GA	50	48.6%	17.9115
$f1$	PSO	50	54.7%	2.8757
$f1$	PSCO	50	100%	5.0731
$f2$	GA	1000	57.1%	126.1018
$f2$	PSO	1000	99.2%	11.3323
$f2$	PSCO	1000	100%	10.0733
$f3$	GA	1000	51.3%	25.9701
$f3$	PSO	1000	5.80%	3.0384
$f3$	PSCO	1000	100%	26.4811

表1中迭代次数是每一次运行算法迭代搜索的次数。当算法搜索达到最大搜索次数时所获得的最优值和实际的最优值之差与实际的最优值的比小于0.1%时算是收敛。平均收敛代数是所有收敛代数的平均值。

由表1可知，类似 $f1$ 这样的病态函数的寻优，PSCO 算法能够很轻易找到最优值，而 GA 和 PSO 算法则易陷入局部最优；而 $f2$ 为多峰值，且函数值相差较大的函数，PSCO 和 PSO 比 GA 算法收敛性能更好。在测试函数 $f3$ 中，通过平均收敛率及平均收敛代数发现，PSO 极易陷入局部极值，由(1)(2)式可知这主要是由于标准 PSO 算法中过多的强调 g_{best} 的影响的结果，g_{best} 的影响使算法加速收敛但是却限制了粒子扩大搜索范围进行搜索的趋势，而 PSCO 则能很好的避免了标准 PSO 算法存在的极易陷入局部最优的缺陷，仿真结果也表明了 PSCO 算法比 GA 算法和标准 PSO 算法具有更优的搜索性能。

5　结论

通过测试函数试验表明，改进的粒子群协同优化算法(PSCO)比标准 PSO 算法和标准的遗传算法(GA)具有更好的寻优性能。

参 考 文 献

[1] Angeline P J. Using Selection to Improve particle Swarm Optimization [A]. proceedings of the 1999 Congresson Evolutionary Com-

Putation [C]. Piscataway, NJ: IEEE Press, 1999: 84-89.

[2] Higashi N, Iba H. Particle SwannOptimization with Gaussian Mutation [A]. Proceedings of the 2003 Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Press, 2003: 72-79.

[3] 吕振肃,侯志荣. 自适应变异的粒子群优化算法 [J]. 电子学报, 2004, 32(3): 416-420.

[4] 曾建潮,崔志华. 一种保证全局收敛的微粒群算法 [J]. 计算机研究和发展, 2004, 41(8): 1333-1338.

[5] 高鹰,谢胜利. 基于模拟退火的粒子群优化算法 [J]. 计算机工程与应用, Jan. 2004: 47-49.

[6] M Clerc, J Kennedy. The particle swarm: Explosion, stability and convergence in a multidimensional complex space. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.

[7] L. Garces. Parameter Adaptation for the Speed Controlled Static AC Drive with Squirrel-Cage Induction Motor [J]. IEEE-IAS Annual Meeting Conference Record, 1979: 843-850.

[8] M. Velez-Reyes, K. Minami & G. C. Verghese. Recursive speed and parameter estimation for induction machines [C]. IEEE/IAS Annual Meeting Conf. Record, San Diego, 1989: 607-611.

[9] 覃建波,黄开胜,等. 改进粒子群算法在永磁同步电动机参数识别中的应用 [J]. 微特电机, 2009, 9: 16-19.

[10] 黄友锐. 智能优化算法及其应用 [M]. 北京: 国防工业出版社, 2008.

Research on Improving Particle Swarm Optimization Algorithm

QIN Jian-Bo, LU An-shan

(*College of Physics and Materials Science*, *Qinzhou University*, *Qinzhou* 535000, *China*)

Abstract: Particle Swarm Optimization Algorithm is structure-simple, requires less adjustment in parameters, and fast convergence rate in optimizing process. This paper introduces an improving particle swarm optimization algorithm based on the summarization of other particle swarm algorithm. Through testing functions test, it shows that the particle swarm optimization algorithm improves the convergence performance obviously.

Key words: Particle swarm optimization (PSO); testing functions; convergence performance

[责任编辑 江元杪]

(上接第4页)

Week Function Mathematics Softwares Applied to Higher Mathematics Experiments

GUOLI Ren

(*College of Mathematics and Computer Science*, *Qinzhou University*, *Qinzhou* 535000, *China*)

Abstract: Compared with Matlab, Mathematica or Maple, Geometer's SketchPAD and Excel are Week Function Mathematics Software, which can be easily contented and popularized while applied in the Higher Mathematics Experiments. This article explores applications of the frequently used functions of Excel in Higher Mathematics Experiments.

Key words: Week Function Mathematics Tool Software; Excel; Mathematics Experiments

[责任编辑 江元杪]