一种多微粒群协同进化算法

李菲菲[1],姚 坤[1],刘希玉[2]
LI Fei-fei[1],YAO Kun[1],LIU Xi-yu[2]

1.山东师范大学 信息科学与工程学院,济南 250014
2.山东师范大学 管理学院,济南 250014
1.School of Information Science & Engineering,Shandong Normal University,Ji'nan 250014,China
2.School of Management,Shandong Normal University,Ji'nan 250014,China
E-mail:lff_yk@163.com

LI Fei-fei,YAO Kun,LIU Xi-yu.Multi-particle swarm co-evolution algorithm.Computer Engineering and Applications,2007,43(22):44-46.

Abstract: Illumined by phenomenon of co-evolution in nature,particle swarm optimization is combined with co-evolution,a multi-particle swarm co-evolution algorithm is presented.In the process of evolution,particle not only exchanges information with other particles in its sub-swarm but also is influenced by other sub-swarms.By doing experiments on three benchmark functions,the results show that the algorithm avoids to trap into local optimum in a certain extent and improves the precision of convergence.
Key words: particle swarm optimization;co-evolution;multi-particle swarm co-evolution

摘 要:受自然界共生现象的启发,将微粒群算法和协同进化相结合,提出了一种多微粒群协同进化算法。进化过程中,粒子不仅要与本子群的其他微粒交换信息,还要受其他子群体的影响。通过对三个标准函数优化的实验结果表明,此算法在一定程度上避免了陷入局部极值点并且提高了收敛精度。
关键词:微粒群算法;协同进化;多微粒群协同进化算法
文章编号:1002-8331(2007)22-0044-03 **文献标识码**:A **中图分类号**:TP301

1 研究背景

微粒群算法最早是在1995年由美国社会心理学家James Kennedy和电气工程师Russell Eberhart共同提出的[1]。微粒群算法因其在迭代初期的有效性及实现简单等特点在很多领域中得到了广泛地应用,但是在寻找近似问题的最优解的效率和精度方面还是遇到了很大的困难。这表明,微粒群算法损失了粒子的多样性-所有的粒子被迄今为止找到的最好的微粒而吸引。很多研究指出了微粒群算法的缺陷并提出了改进的方法,比如增加粒子的多样性、引入进化选择机制以及速率更新方程中的可调参数。

受协同进化遗传算法的启发,Shi和Krohling[2]提出了一种基于两个PSO协同进化的求解极大极小值的算法。Asmara、Krohling和Hoffmann[3]在此算法的基础上又提出了一种"accelerated co-evolutionary PSO",提高了算法的收敛速度并将此算法应用到机器手臂计算机扭距控制中。Ashraf M.Abdelbar、Sherif Ragab和Sara Mitri[4]提出了将协同进化微粒群算法应用到古埃及的一种对弈的棋盘游戏中。Mohammed El-Abd和Mohamed Kamel[5]在他们的论文中指出了两个相互合作的微粒群之间如何进行信息交换。J.J.Liang和P.N.Suganthan[6]提出了一种动态多微粒群优化算法。Krohling和Hoffmann等[7]提出了用协同进化微粒群算法来解决带约束的极大极小值问题。在国内,很多学者也对协同进化微粒群算法进行了研究。Ben Niu,Yunlong Zhu和Xiaoxian He[8,9]提出了多群体合作微粒群算法,并将其用在了为动态系统构建模糊模式上。李爱国[10]提出了一种两层结构的多粒子群协同优化算法,底层用多个粒子群相互独立地搜索解空间以扩大搜索范围,上层用1个粒子群追逐当前全局最优解以加快算法收敛。Hong-yun Meng、Xiao-hua Zhang和San-yang Liu[11]提出将协同进化微粒群算法用到求解多目标优化问题中,此算法针对多目标问题设计了协同进化算子、竞争变异算子和新的选择机制,通过共享和交换群体间的信息从而收缩搜索区域、维持了种群的多样性,避免了陷入最优。王俊年、申群太等[12,13]提出了将多种群协同进化微粒群算法应用到神经网络自适应噪声消除系统和径向基神经网络设计中。

虽然协同进化微粒群算法在一定程度上得到了发展,但是在多种群协同的机制上有待于改善,在微粒群算法和遗传算法的结合上还有待于进一步研究。本文在已有文献的基础上提出了一种多微粒群协同进化算法,实验表明此算法在一定程度上

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.6037405);"泰山学者"建设工程专项经费。
作者简介:李菲菲(1981-),硕士研究生,研究方向为群体智能、数字图像处理;姚坤(1981-),硕士研究生,研究方向为群体智能、创新概念设计;刘希玉(1964-),博士,教授,博士生导师,"泰山学者",研究方向为人工神经网络与进化计算及其应用等。

避免了陷入局部极值点并且提高了收敛精度。

2　微粒群算法

设 D 在维搜索空间中，有 m 个没有重量和体积的粒子组成的一个种群，这些粒子在搜索空间中以一定的速度飞行，飞行速度由个体的飞行经验和群体的飞行经验进行动态调整。其中第 i 个粒子的位置为 $X_i=(x_{i1},x_{i2},\cdots,x_{in})$，飞行速度为 $V_i=(v_{i1},v_{i2},\cdots,v_{in})$，$P_i=(p_{i1},p_{i2},\cdots,p_{in})$为粒子 i 所经历的最好位置，也就是微粒 i 所经历过的具有最好适应值的位置，称为个体最好位置。$P_g=(p_{g1},p_{g2},\cdots,p_{gn})$代表整个微粒群找到的最优位置。将 X_i 代入要优化的问题中计算适应值，粒子状态更新方程为：

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}(t)[p_{ij}(t)-x_{ij}(t)]+c_2r_{2j}(t)[p_{gj}(t)-x_{ij}(t)] \quad (1)$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \quad (2)$$

式中 w 称为惯性权重，$i=1,2,\cdots,m$ 表示粒子个数，$j=1,2,\cdots,D$ 表示搜索空间维数，c_1、c_2 为学习系数，r_1、r_2 是之间的随机数。

3　多微粒群协同进化算法

多微粒群协同进化(MPSC)算法采用一个 master 种群和若干个 slave 种群，每个 slave 种群独立进化，每次迭代结束后将最好的微粒复制到 master 种群，然后 master 种群开始进化，找出最好的微粒，将此微粒信息反馈给各 slave 种群，然后各 slave 种群据此信息修改微粒的速度继续进化。各 slave 种群通过 master 种群进行信息交流，各种群信息交流过程如图 1 所示。各 slave 种群进化过程采用文献[14]中的 GA 和 PSO 混合算法-GSO 算法，GSO 算法的流程如图 2 所示。

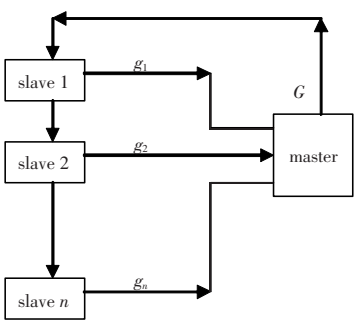

图 1　多微粒群协同进化算法信息交流示意图

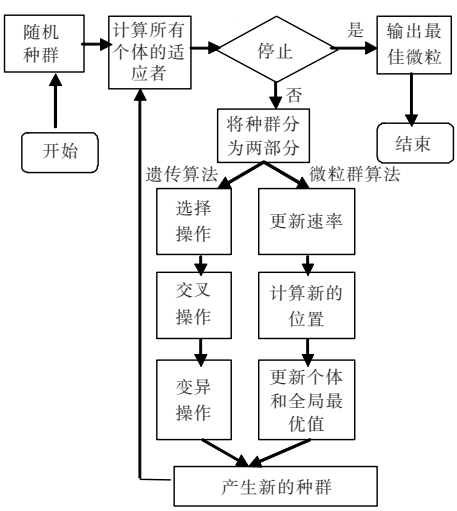

图 2　GSO 算法流程图

对 GSO 算法作一些改进，微粒的速率和位置更新公式为：

$$v_j^i(t+1)=\chi\times(w\times v_j^i(t)+r_1c_1(p_j^i-x_j^i(t))+r_2c_2(g_i-x_j^i(t))+r_3c_3(G-x_j^i(t))) \quad (3)$$

$$x_j^i(t+1)=x_j^i(t)+v_j^i(t+1) \quad (4)$$

$v_j^i(t+1)$为第 i 个 slave 种群的第 j 个微粒在第 $t+1$ 代的速度；

$x_j^i(t)$为第 i 个 slave 种群的第 j 个微粒在第 t 代的位置；

w 为惯性权重。随着进化过程，根据如下公式调整惯性权重：

$$w=w_{\max}-\frac{w_{\max}-w_{\min}}{iter_{\max}}\times iter \quad (5)$$

$$\chi=\frac{2}{(2-\varphi-\sqrt{\varphi^2-4\varphi})},\varphi=c_1+c_2,\varphi>4 \quad (6)$$

χ 为收缩因子；

r_1、r_2、r_3 为在[0,1]之间的随机数；

c_1、c_2、c_3 为学习系数。

p_j^i为第 i 个 slave 种群的第 j 个微粒的个体历史最好位置；

g_i 为第 i 个 slave 种群的全局最好位置；

G 为 master 种群的全局最好位置。

交叉操作：采用单点交叉，将每个粒子的当前位置从维数的中间交换重组，例如粒子的维数为 10，粒子一 6 维-10 维的值等于粒子二 6 维-10 维的值，粒子二 1 维-5 维的值等于粒子一 1 维-5 维的值，其余各项值保持不变。

变异操作：随机生成变异粒子的当前位置，变异粒子其余的值保持不变。

master 种群采用 PSO 算法，其速度和位置更新公式为：

$$v_i^M(\partial)=\chi\times(w\times v_i^M+c_1r_1(g_i^M-x_i^M)+c_2r_2(G-x_i^M)) \quad (7)$$

$$x_i^M(\partial)=x_i^M+v_i^M(\partial) \quad (8)$$

$v_i^M(\partial)$为 master 种群中微粒 i 更新后的速度；

$x_i^M(\partial)$为 master 种群中微粒 i 更新后的位置；

g_i^M 为当前 master 种群中最好微粒的位置；

G 为上次迭代结束后 master 种群中最好微粒的位置。

master 种群的速度和位置更新结束后，要计算每个微粒的适应值，找出最好的微粒。如果最好的微粒的位置差于 G，则继续保留 G。否则，将 G 更新为最好微粒的位置。

多微粒群协同进化算法流程

```
Begin
  初始化每个 slave 群体中的每个微粒;
  for i=1 to n
    计算每个微粒的适应值找出最好的微粒,
    将最好的微粒复制到 master 种群;
  endfor
  对 master 种群中的微粒找出最好的微粒 G;
  while(G 不满足条件或未达到迭代次数)
    for i=1 to n
      每个 slave 种群按概率分为两部分;
      其中一部分进行交叉、变异操作;
      另一部分按照公式(3)和(4)更新速度和位置;
      计算每个微粒的适应值找出最好的微粒;
      更新 p_j^i 和 g_i;
      将最好的微粒复制到 master 种群;
    endfor
```

表 2 MPSC 算法运行 20 次的平均最好适应值对比(平均值±标准差)

Functions	PSO	FPSO	HPSO	PSCO	MPSC
$f_1(x)$	317.78±55.16	190.70±57.33	176.83±40.66	100.60±54.47	7.466 8±1.3195 3
$f_2(x)$	24.92±4.90	22.33±2.71	11.99±1.58	10.635 8±2.58	3.376 2±2.415 2
$f_3(x)$	0.21±0.044	0.025 7±0.218	0.047±0.023	0.007±0.014	0.005 3±0.011 5

对 master 种群中的每个微粒根据公式(7)
和(8)更新速度和位置,计算适应值更新 G;
endwhile
End

4 仿真实验

本文比较 MPSC 算法与模糊 PSO(FPSO)、杂交 PSO(HPSO)以及 PSCO 等算法[10]对 Rosenbrock 函数(f_1),Rastrigrin 函数(f_2)和 Griewank 函数(f_3)等 3 种基准函数的优化精度。

Rosenbrock 函数又称为香蕉函数,其最优解为 $x^*=(1,1,\cdots,1)$极小值为 $f(x^*)=0$。

$$f_1(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2),-100\leqslant x_i\leqslant 100$$

Rastrigrin 函数是一个多峰函数,有很多正弦凸起的局部极小点,它的全局极小点在 $x^*=(0,0,\cdots,0)$处,全局极小值 $f(x^*)=0$。

$$f_2(x)=\sum_{i=1}^{n}((x_i^2-10\cos(2\pi x_i))+10),-10\leqslant x_i\leqslant 10$$

Griewank 函数也是一个多峰函数,它的全局极小点在 $x^*=(0,0,\cdots,0)$处取得,全局极小值 $f(x^*)=0$。

$$f_3(x)=\frac{1}{4\,000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1,-600\leqslant x_i\leqslant 600$$

MPSC 算法用.NET2003 实现,进行实验时,每个微粒设为 10 维,取微粒群子群体数 S=6;每子群中粒子数 m=20,相应主群体中粒子数 m=6;MPSC 算法的每次迭代过程中每个子群中的粒子按照概率 0.5 被分成两个群体:一个是 PSO 群体按照公式(3)和(4)更新速度和位置进行 PSO 操作,另一个是 GA 群体进行 GA 操作;主群体进行 PSO 操作时按照公式(7)和(8)更新粒子速度和位置,其惯性权重也是随迭代次数从 0.7 递减至 0.3,学习系数 c_{master} 的值和子群体的一样;具体参数设置请参看表 1。实验经典 PSO 算法时,取 m=20;$c_1=c_2$=2.0;w=0.7;Vmax=100(对于 $f_1(x)$),Vmax=10(对于 $f_2(x)$),Vmax=600(对于 $f_3(x)$)。

表 1 MPSC 算法优化标准函数的参数设置

	最大速度	w	c_1	c_2	c_3	c_{master}	交叉概率	变异概率	迭代次数
$f_1(x)$	20	0.7 递减至 0.3	2.05	2.05	2.05	2.05	0.5	0.1	10 000
$f_2(x)$	2	0.7 递减至 0.3	2.05	2.05	2.05	2.05	0.2	0.1	10 000
$f_3(x)$	200	0.7 递减至 0.3	2.05	2.05	2.05	2.05	0.7	0.1	10 000

MPSC 算法运行 20 次取平均最好适应值与其他算法进行比较,如表 2 所示实验结果,MPSC 算法相对于前几种算法显示了更好的优化性能。特别对于函数 $f_2(x)$,当 MPSC 算法的迭代次数在 30 000 次以上时,每次运行基本都能得到全局最优值。

5 结论与展望

本文在已有文献的基础上将微粒群算法与协同进化算法结合,集中两个算法的优点,提出了一种多微粒群协同进化算法,通过对三个标准函数优化的结果表明该算法在一定程度上加快了收敛速度、避免了陷入局部极值点的问题。下一步要做的工作有:

(1)因为此算法的交叉操作的交叉点是固定的,是否交叉点随机产生会产生更好的结果还有待于进一步研究。

(2)因为此算法的参数较多,各项参数的不同设置会对运行结果产生什么影响,会不会产生更好的结果还有待于进一步研究。(收稿日期:2007 年 3 月)

参考文献:

[1] Kennedy J,Eberhart R C.Particle swarm optimization[C]//Proc IEEE Int Conf on Neural Networks IV.Pscataway,NJ:IEEE Service Center,1995:1942-1948.

[2] Shi Y,Krohling R.Co-evolutionary particle swarm optimization to solving min-max problems [C]//Proc IEEE Congress on Evolutionary Computation,Honolulu,Hawaii,May 2002:1682-1687.

[3] Asmara A,Krohling R A,Hoffmann F.Parameter tuning of a computed-torque controller for a 5 degree of freedom robot arm using co-evolutionary particle swarm optimization[C]//Proceedings of 2005 IEEE Conference on Swarm Intelligence Symposium,2005:162-168.

[4] Abdelbar A M,Ragab S,Mitri S.Applying co-evolutionary particle swarm optimization to the egyptian board game seega[C]//Proceedings of 2004 IEEE International Joint Conference on Neural Networks,2004:244-248.

[5] Mohammed El-Abd,Mohamed Kamel.Information exchange in multiple cooperating swarms[C]//Proceedings of the 2005 Conference on Genetic and Evolutionary Computation,2005:269-270.

[6] Liang J J,Suganthan P N.Dynamic multi-swarm particle swarm optimizer[C]//Proceedings of 2005 IEEE Conference on Swarm Intelligence Symposium,2005:124-129.

[7] Krohling R A,Hoffmann F,Leandro dos,et al.Co-evolutionary particle swarm optimization for min-max problems using Gaussian distribution[C]//Proceedings of CEC2004 Congress on Evolutionary Computation,2004:959-964.

[8] Niu Ben,Zhu Yun-long,He Xiao-xian.Multi-population cooperative particle swarm optimization[C]//LNAI 3630:ECAL 2005,2005:874-883.

[9] Niu Ben,Zhu Yun-long,He Xiao-xian.Construction of fuzzy models for dynamic systems using multi-population cooperative particle swarm optimizer[C]//LNAI 3613:FSKD 2005,2005:987-1000.

[10] 李爱国.多粒子群协同优化算法[J].复旦学报:自然科学版,2004,43(5):923-925.

[11] Meng Hong-yun,Zhang Xiao-hua,Liu San-yang.A co-evolutionary particle swarm optimization-based method for multiobjective optimization[C]//LNAI 3809:AI 2005,2005:349-359.

[12] 王俊年,申群太,沈洪远,等.基于多种群协同进化微粒群算法的径向基神经网络设计[J].控制理论与应用,2006,23(2):251-255.

[13] 王俊年,申群太,沈洪远,等.基于协同进化微粒群算法的神经网络自适应噪声消除系统[J].计算机工程与应用,2005,41(13):20-23.

[14] Grimaldi E A,Gandelli A,Grimaccia F,et al.A new hybrid technique for the optimization of large-domain electromagnetic problems[C]//Proceedings of URSI GA 2005,2005.