

## 基于稳定策略的粒子群优化算法

魏 波<sup>1,2</sup> 李元香<sup>1,2</sup> 徐 星<sup>3</sup> 申鼎才<sup>1,2</sup>

(武汉大学计算机学院 武汉 430072)<sup>1</sup> (武汉大学软件工程国家重点实验室 武汉 430072)<sup>2</sup>

(景德镇陶瓷学院信息工程学院 景德镇 333403)<sup>3</sup>

**摘 要** 为了解决传统粒子群算法易陷入局部最优解的问题,在借鉴生物学中“进化稳定策略”的基础上,对传统粒子群算法进行了改进,提出了基于稳定策略的粒子群算法。该算法的核心在于,通过稳定参数的设定,使种群中较优的一部分个体按照标准粒子群算法进行寻优,而对种群中其余部分的个体进行随机突变,以达到快速扩大搜索空间、稳定种群中个体多样性的目的。实验结果表明,该算法有效地避免了基本粒子群算法早熟现象的发生,提高了PSO对全局最优解的搜索能力和收敛速度。

**关键词** 演化稳定策略,粒子群算法,早熟收敛

**中图法分类号** TP311 **文献标识码** A

### Particle Swarm Optimization Algorithm Based on Stable Strategy

WEI Bo<sup>1,2</sup> LI Yuan-xiang<sup>1,2</sup> XU Xing<sup>3</sup> SHEN Ding-cai<sup>1,2</sup>

(Computer School, Wuhan University, Wuhan 430072, China)<sup>1</sup>

(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)<sup>2</sup>

(College of Information and Engineering, Jingdezhen Ceramic Institute, Jingdezhen 333403, China)<sup>3</sup>

**Abstract** An improved particle swarm optimization algorithm based on the evolutionarily stable strategy was proposed to avoid the problem of local optimum. The key to this algorithm lies in searching an optimum solution for optimal individuals of the population according to the standard particle swarm algorithm by setting a stable parameter, while the rest part of population is mutated randomly. Therefore, the operator can keep the number of the best individuals at a stable level when it enlarges the search space. The performance of this algorithm shows that this algorithm can effectively avoid the premature convergence problem. Moreover, this algorithm improves the ability of searching an optimum solution and increases the convergent speed.

**Keywords** Evolutionarily stable strategy, Particle swarm algorithm, Premature convergence

## 1 引言

粒子群优化算法<sup>[1]</sup> (Particle Swarm Optimization, PSO)是由Eberhart和Kennedy于1995年提出的,它是一种基于群智能的全局优化进化算法,是对简单生物社会系统的模拟。它不仅可以解决静态优化问题,而且可以解决现实中普遍存在的动态优化问题<sup>[2-4]</sup>。粒子群算法最早源于对鸟群觅食行为的研究。研究者发现鸟群在飞行过程中经常会有规律地改变方向、散开、聚集,其个体行为简单,但其整体却能够完成复杂的行为。通过对类似生物群体行为的研究,发现生物群体中存在着一种社会信息共享机制,它为群体的进化提供了一种优势,这也是粒子群算法形成的基础。

粒子群算法和演化算法<sup>[5]</sup>、模拟退火算法以及蚁群算法等其它元启发式算法一样存在着早熟收敛的问题,早熟收敛现象极大地影响了算法的稳定性。鉴于这些元启发式算法本身都是计算机学科和其它学科交叉研究的成果,所以为了避

免算法出现过早收敛的现象,提高算法的性能,一些学者通过将粒子群算法与物理、数学和生物等其它学科进行交叉研究,或者与其它计算智能方法相融合,以突破其自身局限,提高算法总体性能<sup>[6]</sup>。李元香于2002年提出了一种新的演化算法:动力学演化算法(Dynamical Evolutionary Algorithm, DEA)<sup>[7]</sup>。该算法结合统计物理学和动力学的相关规律,引入了动量和活动量两个概念,提出了一种新的选择机制,使所有的粒子在演化过程中都运动起来,系统中每一个粒子都有机会参与演化,从而有效地保持了种群多样性。文献[8]结合物理学中的自组织临界(Self-Organized Criticality, SOC)提出了一种扩展的SOC-PSO。文献[9-11]分别借鉴遗传算法思想提出杂交PSO算法、具有选择机制的PSO算法和具有变异算子的PSO算法,提高了标准PSO算法的精度。

为了保持种群多样性,避免算法过早陷入局部最优解,同时加快其收敛速度,本文提出了一种基于稳定策略的粒子群优化算法(Stable Strategy Particle Swarm Optimization Algo-

收稿日期:2011-05-03 返修日期:2011-07-01 本文受国家自然科学基金(61070009)资助。

魏 波(1983-),男,博士生,主要研究方向为智能计算与演化计算, E-mail: weibo@whu.edu.cn; 李元香(1962-),男,教授,博士生导师,主要研究方向为演化计算与并行计算。

rithm, SSPSO)。该算法借鉴“进化上的稳定策略”思想,对标准粒子群优化算法进行了改进。本文第2节介绍标准粒子群优化算法;第3节提出 SSPSO 算法;最后为实验数据和分析。

## 2 标准粒子群优化算法

在标准 PSO 算法中,每个优化问题的解被看成是搜索空间的一个“粒子”,算法初始化为一群随机粒子(随机解),然后通过迭代找到最优解。在每次的迭代中,粒子通过跟踪两个“极值”来更新自己。在找到这两个最优值后,粒子通过下面公式来更新自己的速度和位置。

$$v_{id}^{t+1} = \omega v_{id}^t + R_1 c_1 (P_{id} - x_{id}^t) + R_2 c_2 (P_{gd} - x_{id}^t) \quad (1)$$

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

在式(1)、式(2)中,  $i = 1, 2, 3, \dots, M$ ,  $M$  是该群体中粒子的总数。

$v_{id}^t$ : 粒子  $i$  第  $t$  次迭代后速度矢量的第  $d$  维分量;

$x_{id}^t$ : 粒子  $i$  第  $t$  次迭代后的位置矢量的第  $d$  维分量;

$P_{id}$ : 粒子  $i$  最好位置的第  $d$  维分量;

$P_{gd}$ : 种群当前位置的第  $d$  维分量;

$R_1$  和  $R_2$  是服从  $U(0,1)$  分布的随机数;

$c_1$  和  $c_2$  是学习因子,通常取  $c_1 = c_2 = 2$ 。

在每一维,粒子都有一个最大限制速度  $V_{\max}$ ,如果某一维的速度超过设定的  $V_{\max}$ ,那么这一维的速度就被限定为  $V_{\max}$ 。从社会学的角度来看,式(1)的第一部分称为记忆项,表示过去的速度对现在的影响;公式第二部分称为自身认知项,是从当前点指向粒子自身最好点的矢量,表示粒子的动作来源于自己经验部分;公式的第三部分称为群体认知项,是从当前点指向种群最好点的矢量,反映了粒子间的协同合作和知识共享。粒子就是通过自己的经验和同伴中最好的经验来决定下一步的运动。

## 3 进化上的稳定策略粒子群优化算法

### 3.1 进化上的稳定模型简介

进化上的稳定策略<sup>[12]</sup> (Evolutionarily Stable Strategy, ESS) 是 Smith 提出的基本概念。稳定策略是指种群中大部分成员所采用的策略,而这种策略的好处是其它策略所不能比的。也就是说,个体的行为应该遵守群体的约定,一种稳态模型一旦确定,就会稳定下来。任何偏离该模型的个体行为将受到自然选择的惩罚。

该策略通过设置稳定参数,对多余的优秀个体实施不同于一般变异算子的突变操作,使得每一代种群中优秀个体的数目维持在一个稳定值,既减轻了选择机制对低适应值个体所造成的“生存压力”,保持了种群的多样性,避免了过早收敛;同时又扩大了搜索空间,起到了通常的变异达不到的快速逃离局部最优解误区的作用,提高了收敛速度。这里将该策略应用于粒子群优化算法中,提出了稳定策略的粒子群优化算法 (Stable Strategy Particle Swarm Optimization Algorithm, SSPSO)。该算法提出了一种新的搜索机制,该机制不仅保留了基本粒子群优化算法的优点,还提高了 PSO 算法的收敛速度,而且也有效避免了算法的早熟现象,并能快速达到问题的全局最优值。

### 3.2 一种基于稳定策略的粒子群优化算法(SSPSO)及其实现

在文献[11]中,借鉴遗传算法思想提出杂交 PSO 算法,提高了标准粒子群优化算法的精度。然而,其采取的策略是对最优个体进行变异,即对最优个体的少量性质的改变。变异后产生的新个体仍然保留了母体中的大量性质。这种变异在当前种群正处于最优解临近区域而逐步向最优解逼近中起着重大作用。这是因为,也许最优解正好期待着当前种群中某一个体在某一性质上的变异。可是这种变异也有其不利于寻找全局最优解的方面,如果最优个体已经陷入局部最优区域,那么这种变异也改变不了这种总体的趋势。

为了改变这种局部最优的状态,提出了一种 SSPSO 算法。该算法在种群进行寻优的过程中,使一定比例的且靠近最优解的个体处于稳定状态,而对其余个体进行突变。这种突变在保持标准 PSO 算法中最优个体搜索趋势的同时,增大了种群的搜索空间,增加了种群的多样性,提高了种群搜索全局最优解的能力,避免了早熟收敛。

算法 SSPSO 的流程如下所示:

Step1 初始化一群粒子(群体规模为  $M$ ),包括位置、速度和稳定参数  $k$ ;

Step2 评价每个粒子的适应值,找到种群当前最优解;

Step3 按粒子的适应值优劣将其进行排序;

Step4 如果算法收敛准则满足或达到最大迭代次数,执行 Step8,否则执行 Step5;

Step5 排序序号在前  $M * k$  的粒子,根据式(1)、式(2)更新其速度和位置;

Step6 排序序号从  $M * k + 1$  到  $M$  的粒子,使其速度和位置按初始化方式进行随机突变(其中  $k$  的取值范围为 0 到 1.0);

Step7 未满足结束条件则转入 Step2;

Step8 输出算法所找到的全局最优,算法结束。

## 4 实验仿真与结果

在本文试验中,使用了 5 个经典的测试函数,其中 4 个测试函数有全局最小值,这些函数经常被国内外很多学者用来测试函数优化算法的性能和可靠性。在算法测试过程中,种群大小为 100,最大迭代次数设置为 1000,稳定策略参数  $k$  的取值范围为  $(0,1)$ ,根据文献[12]取值为  $1/6$ ,参数  $\omega$  采用线性递减的策略从 0.96 递减到 0.4,  $V_{\max}$  取每维搜索空间宽度的一半。算法 SSPSO 在所有的测试用例上都运行 50 次。

测试函数如下:

$$F_1 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, -2.048 \leq x_i \leq 2.048$$

$$F_2 = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^4}, -100 \leq x_i \leq 100$$

$$F_3 = (x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0], -100 < x_i < 100$$

$$F_4 = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2, -100 < x_i < 100$$

$$F_5 = \sum_{i=0}^{n-1} x_i^2, -5.12 \leq x_i \leq 5.12$$

为了深入地研究 PSO 算法和 SSPSO 算法,用 MATLAB 对 5 个测试函数优化问题进行了仿真。在相同迭代次数下,

两种算法最优个体的适应值变化情况如图 1—图 5 所示。

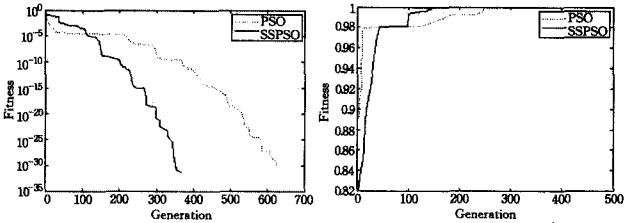


图 1 在  $F_1$  上两种算法的收敛曲线

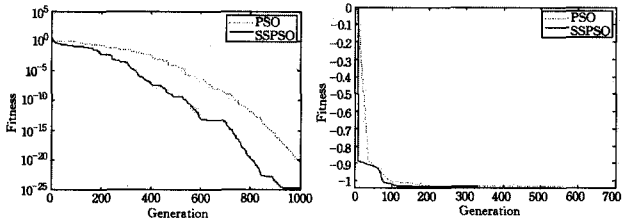


图 2 在  $F_2$  上两种算法的收敛曲线

从仿真实验可知,在分别对测试函数  $F_1$ 、 $F_2$ 、 $F_4$  进行仿真中,标准 PSO 算法分别在第 629、389、564 次迭代时全局收

敛,而 SSPSO 算法分别在第 371、185、330 次迭代时就已经全局收敛,即 SSPSO 算法的收敛的迭代次数几乎只用了标准 PSO 算法的一半。同样,对测试函数  $F_3$  进行仿真中,虽然 PSO 算法和 SSPSO 算法在最大迭代次数(1000 次)时都还没有收敛到全局最优值 0,但此时 PSO 得到的最优值为  $4.02724\text{e-}021$ ,而 SSPSO 得到的最优值为  $1.67162\text{e-}025$ 。当测试函数为  $F_5$  时,实验中  $n$  的取值为 10,即函数的维数为 10,仿真效果如图 5 所示。

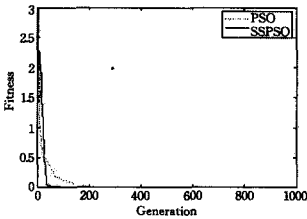


图 5 在  $F_5$  上两种算法的收敛曲线

同样,对测试函数  $F_1$ 、 $F_2$ 、 $F_3$ 、 $F_4$ 、 $F_5$  进行 50 次运行情况的统计分析,从表 1 中可以看出,SSPSO 算法的收敛性明显优于 PSO 算法的收敛性,且运用 SSPSO 算法得出的平均最优值、最差值以及达到最优值的比例均优于运用算法 PSO 得出的结果。

表 1 算法 PSO 与算法 SSPSO 在  $F_1$ 、 $F_2$ 、 $F_3$ 、 $F_4$ 、 $F_5$  上的运行结果比较

| No.   | PSO     |                        |            |                         | SSPSO                  |            |                        |
|-------|---------|------------------------|------------|-------------------------|------------------------|------------|------------------------|
|       | 最优值     | 最差值                    | 达到最优比例 (%) | 平均最优值                   | 最差值                    | 达到最优比例 (%) | 平均最优值                  |
| $F_1$ | 0       | $2.05 \times 10^{-21}$ | 99         | $1.27 \times 10^{-23}$  | $7.54 \times 10^{-26}$ | 100        | $5.64 \times 10^{-28}$ |
| $F_2$ | 1.00    | 0.9903                 | 95         | 1                       | 1.00                   | 99         | 1.00                   |
| $F_3$ | 0       | $2.05 \times 10^{-19}$ | 98         | $4.024 \times 10^{-21}$ | $1.67 \times 10^{-20}$ | 100        | $3.04 \times 10^{-25}$ |
| $F_4$ | -1.0316 | -0.09210               | 95         | -1.0295                 | -1.0316                | 99         | 1.0316                 |
| $F_5$ | 0       | $3.35 \times 10^{-10}$ | 95         | $5.33 \times 10^{-12}$  | $4.42 \times 10^{-12}$ | 97         | $3.04 \times 10^{-15}$ |

**结束语** 本文提出的 SSPSO 算法借鉴了生物学中的“进化上的稳定策略”,在向当前最优值逼近的同时,检验种群的全局最优特性,通过突变算子的引入,使整个种群保持在最有可能成功状态,加快了整个算法向全局最优值逼近的速度。采用这种策略,在遵循标准 PSO 算法基本寻优规则的同时,使一定比例的且排序靠近最优解的个体处于稳定状态,而对其余个体进行突变。这种突变在保持标准 PSO 算法中最佳个体搜索趋势的同时,增大了种群的搜索空间,增加了种群的多样性,提高了种群搜索全局最优解的能力,避免了早熟收敛。这种处理方式较好地平衡了局部“搜索”和“开拓”新领域之间的关系。实验数据表明,本文提出的算法较大地改善了标准粒子群优化算法的性能,具有更稳健的搜索能力和更高的收敛精度。我们下一步的研究重点是稳定参数的最优化研究及 SSPSO 算法与其它改进 PSO 算法的比较。

### 参考文献

[1] Kennedy J, Eberhart R. Particle Swarm Optimization[C] // Proceedings of the IEEE International Conference on Neural Network. Perth, Australia, USA; IEEE Press, 1995: 1942-1948

[2] Esquivel S C, Coello C A. Hybrid particle swarm optimizer for a class of dynamic fitness landscape[J]. Eng. Optim., 2006, 38 (8): 873-888

[3] Greeff M, Engelbrecht A P. Solving dynamic multi-objective problems with vector evaluated particle swarm optimization[C] // Proc. Congr. Evol. Comput. 2008: 2922-2929

[4] Hu J, Zeng J, Tan Y. A diversity-guided particle swarm opti-

zer for dynamic environments[M]. Bio-Inspired Computational Intelligence and Applications, Berlin, Germany; Springer-Verlag, 2007: 239-247

[5] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版, 1998

[6] 徐星, 李元香, 姜大志, 等. 一种基于分子动理论的改进粒子群优化算法[J]. 系统仿真学报, 2009, 21(7): 1904-1907

[7] Li Y X, Zou X F, Kang L S, et al. A new dynamical evolutionary algorithm from statistical mechanics[J]. Computer Science and Technology (S1000-9000), 2001, 18(3): 361-368

[8] Lovbjerg M, Krink T. Extending Particle Swarm Optimisers with Self-organized Criticality[C] // Proceedings of the IEEE International Conference on Evolutionary Computation. Hawaii, USA; IEEE Press, 2002: 1588-1593

[9] Angeline P J. Evolutionary Optimization versus Particle Swarm Optimization; Philosophy and Performance Differences [C] // Proceedings of the Seventh Annual Conference on Evolutionary Programming. Berlin, Germany; Springer-Verlag, 1998: 601-610

[10] Angeline P J. Using Selection to Improve Particle Swarm Optimization [C] // Proceedings of the IEEE International Conference on Evolutionary Computation. Anchorage, Alaska, USA; IEEE Press, 1998: 84-89

[11] Stacey A, Jancic M, Grundy I. Particle swarm optimization with mutation[C] // Proceedings of IEEE Congress on Evolutionary Computation. Canbella, Australia, USA; IEEE Press, 2003: 1425-1430

[12] Dawkins R. The Selfish Gene[M]. Oxford University Press, Reprinted, 1997