

# 微粒群优化算法研究现状及其进展

杨 燕<sup>1</sup>, 靳 蕃<sup>1</sup>, Kamel M<sup>2</sup>

(1. 西南交通大学计算机与通信工程学院, 成都 610031; 2. University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1)

**摘 要:** 对进化计算中引起广泛兴趣的微粒群优化 (PSO) 算法的研究现状进行了考察, 介绍了一些最新研究进展, 包括: 杂交 PSO、基于邻域算子的 PSO 和基于不同搜索方向的 PSO, 并简要介绍了 PSO 在求解复杂优化问题如多目标优化和带约束优化中的优势。最后给出了一些应用实例, 讨论了将来可能的研究内容。

**关键词:** 微粒群优化; 进化计算; 群集智能

## Research and Development of Particle Swarm Optimization

YANG Yan<sup>1</sup>, JIN Fan<sup>1</sup>, Kamel M<sup>2</sup>

(1. School of Computer and Communication Engineering, Southwest Jiaotong University, Chengdu 610031;

2. Department of System Design Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada)

**[Abstract]** This paper reviews the research of particle swarm optimization (PSO), which is an interesting branch in evolutionary computation. Some latest research results are presented: hybrid PSO, PSO with neighborhood operator, and PSO based on different search directions. The paper also describes the advantages of PSO in solving complicated optimization problems such as multi-objective particle swarm optimization and constrained optimization. Finally a few applications in engineering are introduced, and the future research issues are discussed.

**[Key words]** Particle swarm optimization; Evolutionary computation; Swarm intelligence

在计算智能领域有两种基于群集智能的算法: 蚁群算法和微粒群优化算法 (particle swarm optimization, PSO)<sup>[1]</sup>, PSO 算法 1995 年由 Kennedy 和 Eberhart 率先提出<sup>[2]</sup>, 它借鉴了鸟群或鱼群捕食过程的社会行为, 是一种有别于遗传算法 (genetic algorithm, GA) 的并行进化计算技术。经过短短几年时间的发展, PSO 已广泛应用于函数优化、人工神经网络训练、模糊系统控制等领域, 成为目前进化计算研究的一个新热点<sup>[3,4]</sup>。

### 1 PSO 基本算法

PSO 是一种基于群体的随机优化技术。与其它基于群体的进化算法相比, 它们均初始化为一组随机解, 通过迭代搜寻最优解。不同的是: 进化计算遵循适者生存原则, 而 PSO 模拟社会。PSO 将每一个可能产生的解表述为群中的一个微粒, 每个微粒都具有自己的位置向量和速度向量, 以及一个由目标函数决定的适应度。所有微粒在搜索空间中以一定的速度飞行, 通过追随当前搜索到的最优值来寻找全局最优。

假设在一个 D 维目标搜索空间有 m 个微粒, 每个微粒的位置表示一个潜在的解。第 i 个微粒的位置向量  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , 位置变化率—速度向量  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , 它经历过的最好位置 (有最好适应度) 记为  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ , 也称为个体极值  $p_{best}$ 。整个微粒群迄今为止搜索到的最优位置记为  $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ , 也称为全局极值  $g_{best}$ 。每一次迭代, 微粒的速度由个体极值和全局极值来更新<sup>[5]</sup>:

$$V_{i,t+1} = w \cdot V_{i,t} + c_1 \cdot r_1 \cdot (P_i - X_i) + c_2 \cdot r_2 \cdot (P_g - X_i) \quad (1)$$

其中  $V_{i,t+1}$  为更新后的微粒速度;  $V_{i,t}$  为当前微粒速度;  $w$  为惯性权重, 表示当前速度对下一代速度的影响权重;  $X_i$  为当前微粒位置;  $r_1$  和  $r_2$  为范围在  $[0, 1]$  内变化的随机数;  $c_1$  和  $c_2$  为加速常数, 表示向  $p_{best}$  和  $g_{best}$  靠近的速度变化, 取值越大, 则微粒移向  $p_{best}$  和  $g_{best}$  的加速度越大。微粒的速度  $V$  被一个最大速度  $V_{max}$  所限制, 以免微粒跑到可行解的范围之外进行搜索。

每次迭代中, 每个微粒的位置由速度向量加上位置向量

来修改:

$$X_{i,t+1} = X_{i,t} + V_i \quad (2)$$

其中  $X_{i,t+1}$  为更新后的微粒位置。

PSO 基本算法流程可简单描述为<sup>[6]</sup>: 初始化一群微粒, 即每个微粒被随机赋予初始位置  $X_i$  (视为潜在解) 和初始速度  $V_i$ , 计算其适应度  $f$ 。在每一次迭代中, 微粒的速度和位置根据公式 (1) 和 (2) 更新, 并评价其适应度。当一个微粒发现一个好于它以前所经历的位置, 则将此坐标记入向量  $P_i$  中,  $P_i$  与该微粒当前位置向量之差被随机加入到下一代速度向量, 致使下一代微粒围绕此点震荡搜索, 见公式 (1) 中第 2 项, 此项亦称为“认知”分量。微粒群中迄今为止搜索到的最优位置  $P_g$  与该微粒当前位置向量之加权差也被随机加入到速度向量, 以调整下一代速度, 见公式 (1) 中第 3 项, 此项亦称为“社会”分量。这两项调整使微粒围绕两个最优值展开搜索。

### 2 PSO 研究进展

自 Kennedy 和 Eberhart 提出 PSO 算法<sup>[2]</sup>, 吸引了不少研究者。Shi 等引入惯性权重形成了目前的 PSO 基本算法<sup>[5]</sup>, 并对 PSO 模型中参数选择作了详尽阐述<sup>[7]</sup>。Kennedy<sup>[8]</sup> 及 Suganthan<sup>[9]</sup> 分别从邻域算子角度分析了 PSO 算法性能, Clerc 等<sup>[6]</sup> 给出了 PSO 算法较完整的理论分析, 文献<sup>[10]</sup> 比较了 PSO 与标准 GA 差别。上述研究表明, PSO 在迭代早期性能优异, 但在几个实际优化问题中当逼近最优解时表现不近人意。针对 PSO 的不足, 下面介绍几种最新推出的改进算法。

#### 2.1 杂交 (Hybrid) PSO

杂交 PSO 组合了进化计算与 PSO 的思想。Angeline 利用

**作者简介:** 杨 燕 (1964—), 女, 副教授、在职博士生, 主研方向: 计算智能, 群集智能, 数据挖掘; 靳 蕃, 教授、博导; Kamel M, 教授、博士

**收稿日期:** 2003-09-17

**E-mail:** cileny@tom.com

PSO机理和通常为进化计算所用的自然选择机理如GA,最早提出了一种杂交PSO——HS (hybrid swarm)。其原理采用了进化计算中锦标赛选择方法:每个微粒将其当前位置上的适应度与其它 $k$ 个微粒的适应度比较,记下最差的一个得分,然后整个微粒群以分值高低排队。在此过程中,不考虑微粒的个体极值 $p_{best}$ 。群体排序完成后,用群体中当前位置和速度最好的一半替换群体中差的另一半,而不修改微粒的个体极值 $p_{best}$ 。这样,每一次迭代后,一半微粒将移动到搜索空间中相对较优的位置,这些移动的微粒仍保留它们的个体极值,以便用于下一代位置更新。此HS方法加入选择后具有更强的搜索能力。

Lovbjerg等进一步提出了基于繁殖和子群的杂交PSO模型。他们在繁殖过程中引入了繁殖概率来代替适应度。每次迭代时,依据繁殖概率在微粒群中选取一定数量的微粒放入一个池中,池中微粒随机两两进行繁殖,产生数目相同的后代微粒,用后代微粒代替父母微粒,使种群数目保持不变。后代微粒的位置由父母位置的算术“交叉”得

$$child_1(X_i) = p \cdot parent_1(X_i) + (1-p) \cdot parent_2(X_i) \quad (3)$$

$$child_2(X_i) = p \cdot parent_2(X_i) + (1-p) \cdot parent_1(X_i) \quad (4)$$

其中 $p$ 是均匀分布在0-1之间的随机数。后代的速度向量由父母速度向量之和归一化后得到:

$$child_1(V_i) = \frac{parent_1(V_i) + parent_2(V_i)}{|parent_1(V_i) + parent_2(V_i)|} |parent_1(V_i)| \quad (5)$$

$$child_2(V_i) = \frac{parent_1(V_i) + parent_2(V_i)}{|parent_1(V_i) + parent_2(V_i)|} |parent_2(V_i)| \quad (6)$$

搜索空间中位置的算术“交叉”操作是标准GA中最常用的“交叉”方法之一,这里使用“交叉”操作的原因是后代微粒可受益于父母双方,增强搜索能力。例如,父母微粒分别处于不同局部最优峰值,则其后代能跳出局部最优,更快寻找到全局最优。基于子群的PSO将微粒群划分成一组子粒群,每一个子粒群有自己的最优解。繁殖操作可在同一子群内部进行,也可在不同子群之间进行。

最近, Higashi和Iba提出了用进化计算中高斯“变异”算子与PSO杂交的思想。该方法在传统的速度与位置更新规则中融入高斯“变异”,以避免陷入局部极小。在GA中常用的“变异”操作被应用到PSO中:微粒群中的每个微粒在搜索区域内移动到另一位置时,不象原来只依据一个先验概率,而不受其它微粒的影响,而是由于高斯“变异”具有一定的模糊性。“变异”的计算公式为

$$mut(x) = x \cdot (1 + gaussian(\sigma)) \quad (7)$$

根据实验经验,其中 $\sigma$ 取值为搜索空间中一维长度的十分之一。搜索中,微粒依先验概率选择,它们的位置由具有高斯分布的概率决定。搜索开始时,搜索范围较宽,后来随着高斯变异率的逐渐减小,搜索效率得到改善。

## 2.2 基于邻域算子(Neighborhood)的PSO

对PSO中邻域算子的研究发现<sup>[8,9]</sup>:邻域算子能改进PSO性能,因为它能保持微粒群的多样性,避免过早收敛。Suganthan<sup>[9]</sup>引入一个可变的邻域算子来改进标准PSO的性能。在优化的初始阶段,一个微粒的邻域就是它本身。随着迭代次数增多,邻域逐渐变大,直至包括所有微粒。换句话说,PSO算法中的全局极值 $g_{best}$ 被一个局部邻域逐渐增加的

局部极值 $l_{best}$ 所代替。与此同时,PSO算法中随机漫步和惯性权重的大小也逐渐调整,以使在优化的最后阶段进行更好的搜索。

最近Peer等用不同的邻域拓扑来研究保证收敛PSO(GCPSO)的性能。GCPSO用来保证当减少微粒群多样性而使算法更快收敛时的局部收敛性。将两种邻域拓扑结构: $l_{best}$ 拓扑和Von Neumann拓扑应用到GCPSO,并在不同的基准函数上测试结果显示,它们比标准PSO的 $g_{best}$ 拓扑有更优的性能。

Hu等将一种动态邻域PSO—DNPSO应用到多目标优化问题,并用扩展存储区存储全局Pareto最优解来减少计算时间。实验表明,改进的DNPSO算法比原DNPSO算法和其它多目标优化技术更有效。

## 2.3 基于不同搜索方向的PSO

文献[6]的理论分析指出:PSO中的微粒位置以减幅正弦波震荡,直至收敛于它们的个体极值和全局极值之间的点。如果一个微粒在这震荡期间访问的某些点比它自己历史上最好位置的适应度好,则此微粒继续运动,一般收敛于至今发现的全局最好位置,所有微粒有类似性质,快速收敛于问题的解。但是,如果问题的全局最优位置不是位于初始微粒位置与一个局部最优位置之间,则这种收敛性质将阻碍搜寻全局最优解,许多微粒因为在同样方向(已知的局部最优)上搜索而浪费时间。也许让不同微粒在其它可能搜索方向上搜寻,效果更好。Al-Kazemi和Mohan提出的多阶段PSO在此方面进行了探索。算法将微粒按不同阶段的临时搜索目标分组,这些临时目标允许微粒向着或背着它自己或全局至今发现的最好位置移动。从更高层次来看,该算法能进行多目标搜索,避免微粒陷入局部最优。

Peram等从微粒受其它微粒影响角度探索另一种搜索方向的变化,提出了基于适应度/距离比(Fitness-Distance-Ratio, FDR)的PSO算法。在该算法中,微粒向附近微粒中适应度较高的位置移动,而不是向至今搜索到的全局极值处移动。在标准PSO算法的速度更新公式中,“认知”和“社会”分量表示微粒从自己的经验和最成功微粒的经验中学习,FDR-PSO在此公式中新增加了一项,表示微粒也从有较高适应度的邻域微粒的经验中学习。即在每一维速度更新中,每次还需选取一个有较高适应度的邻域微粒,这可通过最大化适应度之差与一维距离之比来确定。换句话说,微粒 $i$ 的第 $d$ 维速度可通过一个称之为 $n_{best}$ 的邻域微粒修改,设此邻域微粒迄今为止搜索到的最优位置为 $P_i$ ,则最大化下式:

$$\frac{Fitness(P_i) - Fitness(X_i)}{|P_{id} - X_{id}|} \quad (8)$$

即可确定 $n_{best}$ 。上式称为适应度/距离比(FDR),算法因此而得名。事实证明,这种在每一维速度更新中根据FDR选取不同邻域微粒的方法比在所有速度维只选取一个邻域微粒更有效。

## 3 PSO应用实例

PSO之所以具有吸引力的原因之一是它只有很少的参数需要调整,而且算法只需要作细微调整即可广泛应用于不同场合。PSO最直接的应用是解决大多数优化问题,包括多目标优化和带约束优化问题。解决多目标优化问题的方法有:动态邻域策略、用Pareto统治概念决定微粒飞行方向、以及

(下转第9页)

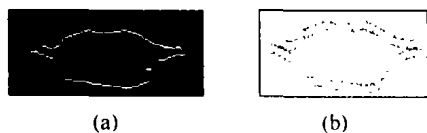


图2 Snake图像能量及外力场示意图

### 3 实验结果

本文的嘴部特征提取中各参数分别为  $\alpha = 1$ 、 $\beta = 5$ 、 $\gamma = 1$ 、 $T = D/10$  ( $D$ 为两嘴角之间的距离)，迭代次数为20次。图3中给出了一些嘴部分割的例子。可以看出本文提出的方法在增强弱边界以及防止边缘泄漏上取得了比较好的效果，且对胡须和嘴的开闭不敏感。

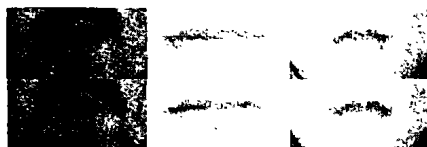


图3 嘴部特征提取的例子

## 4 结论

本文首先指出了利用Snake模型分割人脸特征可能出现的问题,并给出了通过手工定位分两步进行Snake初始化的方法,得到位置接近形状近似的初始化轮廓。引入了Snake图像能量区域的方法来构造外力场,并将初始化轮廓看作模

板,使曲线在目标边缘缺失处基本保持模板形状不变。所得到的最终轮廓准确自然,克服了可变形模板存在人工痕迹、不能准确刻画目标轮廓的缺点。同时能够准确定位初始化轮廓,减少了迭代次数,提高了收敛速度。但本文采用的方法有许多预处理过程,算法的整体速度可能变慢。在今后的工作中,将进一步改进初始化方法,达到能够快速、自动地分割图像的目标。

## 参考文献

- 1 Radeva P, Marti E. Facial Features Segmentation by Model-based Snake. Int. Conf. on Comp. Anal. and Image Processing, Prague, 1995
- 2 Xu Chengyang, Prince J L. Snakes, Shapes and Gradient Vector Flow. IEEE Trans. Image Processing, 1998,7(3)
- 3 Looney C G. Pattern Recognition Using Neural Network. Oxford University Press, Chap. 10, 1997
- 4 Vincent L. Minimal Path Algorithm for the Robust Detection of Linear Features in Gray Images. *Mathematical Morphology and Applications to Image and Signal Proc.* Kluwer Academic Publications, 1998:331
- 5 Leymarie F, Levine M D. Fast Raster Scan Distance Propagation on the Discrete Rectangular Lattice. CVGIP: IMAGE UNDERSTANDING, 1992,55(1):84-94
- 6 Park J, Keller J M. Snakes on the Watershed. IEEE Trans. Pattern Analysis and Machine Intelligence, 2001,23(10):1201-1205

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

(上接第4页)

用Sigma方法寻求微粒的最佳局部向导等。Parsopolos等采用一个非固定的多阶段任务惩罚函数和PSO将约束优化问题转换为非约束优化问题。Ray等用Pareto排列方案处理约束优化。文献[11]用保留可行性策略来处理约束优化,并在压力容器设计、焊接梁设计及压缩弹簧重量最小化等实际工程项目中得到验证。

另一个更为广泛的应用是进化人工神经网络。PSO不仅用于训练网络的权重，而且进化网络的结构<sup>[1]</sup>。PSO已成功应用于进化一个诊断人颤抖的神经网络。Xiao等用PSO进化自组织映射(Self-Organizing Maps, SOM)网络的权重，提出杂交SOM/PSO算法进行基因聚类。算法的第一步用SOM聚类数据集，得到一组权重；第二步用此权重初始化PSO，再由PSO优化聚类结果。

PSO最近在任务分配、虚拟仪器参数校准、交通事故检测、温度模型优化、以及粗轧宽展控制模型优化等方面均有报道。PSO还可用于一种基于Internet的新型软件工具-推荐系统,掌握用户个人喜好,在其选择网站时提出取舍建议。

## 4 结论

现阶段对PSO的研究还缺乏完善的理论分析,数学基础相对薄弱。这也是今后需深入研究的重点之一。此外,笔者认为,以下几个方面值得研究者关注:

- 基于生物/社会原理的微粒群算法
- 微粒群的自组织（或突现）性质
- PSO算法的性能评价
- 微粒群算法的收敛性能分析
- 用神经网络、模糊系统等杂交算法
- PSO的应用

## 参考文献

- 1 Kennedy J, Eberhart R C, Shi Y. *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001
- 2 Kennedy J, Eberhart R C. Particle Swarm Optimization. Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995: 1942-1948
- 3 谢晓峰, 张文俊, 杨之廉. 微粒群算法综述. 控制与决策, 2003, 18(2): 129
- 4 李爱国, 覃征, 鲍复民等. 粒子群优化算法. 计算机工程与应用, 2002, 21(1)
- 5 Shi Y, Eberhart R C. A Modified Particle Swarm Optimization. Proc. IEEE International Conference on Evolutionary Computation, Anchorage, 1998: 69-73
- 6 Clerc M, Kennedy J. The Particle Swarm Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Trans. on Evolutionary Computation, 2002, 6(1): 58
- 7 Shi Y, Eberhart R C. Parameter Selection in Particle Swarm Optimization. Evolutionary Programming VII, Lecture Notes in Computer Science 1447, Springer, 1998, 591-600
- 8 Kennedy J. Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance. Proc. of the IEEE Congress of Evolutionary Computation, 1999, 3: 1938
- 9 Suganthan P N. Particle Swarm Optimiser with Neighborhood Operator. Proc. of the IEEE Congress of Evolutionary Computation, 1999, 3: 1958
- 10 Eberhart R C, Shi Y. Comparison Between Genetic Algorithms and Particle Swarm Optimization. Evolutionary Programming VII, Lecture Notes in Computer Science 1447, Springer, 1998, 611-616
- 11 Hu X, Eberhart R C, Shi Y. Engineering Optimization With Particle Swarm. IEEE Swarm Intelligence Symposium, Indianapolis, USA, 2003: 53-57