

新的进化计算算法——粒子群优化算法

李炳宇 萧蕴诗

(同济大学电子与信息工程学院 上海200092)

New Evolution Algorithm——Particle Swarm Optimization

LI Bing-Yu XIAO Yun-Shi

(School of Electronic and Information Engineering, Tongji University, Shanghai 200092)

Abstract Particle Swarm Optimization (PSO), rooting from simulation of swarm of bird, is a new branch of Evolution Algorithms based on Swarm Intelligence. Concept of PSO, which can be described with only several lines of codes, is more easily understood and realized than some other optimization algorithms. PSO has been successfully applied to much engineering. Firstly, this paper depicts natural explanation about PSO, secondly, introduces its basic theory and several development versions of PSO, and presents some applications of PSO. At last, a brief conclusion and further research direction are given.

Keywords Particle swarm optimization, Evolution algorithm, Swarm intelligence

1. 引言

大自然始终是开启人类智慧的老师。“师法自然”，人类受到社会系统、物理系统、生物系统等运行机制启发，建立和发展起一个个研究工具和手段来解决和攻克研究过程中遇到的困难。典型的有遗传算法(Genetic Algorithm, GA)、人工神经网络(Artificial Neural Networks, ANN)、进化规划(Genetic Programming, EP)、蚁群优化算法(Ant Colony Optimization, ACO)等。这些算法已经被广泛应用于工程，有许多很成功的实例。而在计算智能(Computational Intelligence, CI)领域中，有两种基于群体智能(Swarm Intelligence, SI)^[1,2]的算法：蚁群优化算法和粒子群优化算法。前者是对蚂蚁群落搜索食物行为的模拟，已经成功运用在很多组合优化问题上；后者就是本文将要介绍的粒子群优化算法。

粒子群优化算法(Particle Swarm Optimization, PSO)是由Eberhart博士和Kennedy博士发明的一种进化计算技术^[3,4]，是进化计算领域中的一个新的分支，其源于鸟群和鱼群群体运动行为的研究。

与基于达尔文“适者生存，优胜劣汰”进化思想的遗传算法不同的是，粒子群优化算法是通过个体之间的协作来寻找最优解的。文[3]引用了生物社会学家E. O. Wilson关于生物群体的一段话，“至少在理论上，一个生物群体中的一员可以从这个群体中所有其他成员以往在寻找食物过程中积累的经验中获得好处。只要食物源不可预知地分布于不同地方，这种协作带来的优势可能变成决定性的，超过群体中个体之间对食物竞争所带来的劣势”^[5]。这段话的意思是说生物群体中信息共享会产生进化优势，这也正是粒子群优化算法的基本思想。

PSO算法概念简单，容易实现，其代码只有短短的几行，和其他优化算法相比，这也是它的优点之一。短短几年里，PSO算法已经获得了很大的发展，并已经在一些领域得到应用。国内对PSO算法的研究刚刚开始引起人们的注意。这篇文章的目的是抛砖引玉，使读者能够了解粒子群优化算法的特点和目前的研究情况，并展开进一步的研究。

2. PSO 的算法原理

自然界中一些生物的行为特征呈现群体特征，可以用简单的几条规则将这种群体行为(Swarm Behavior)在计算机中建模，实际上就是在计算机中用简单的几条规则来建立个体的运动模型，但这个群体的行为可能很复杂。例如，Reynolds使用了下列三个规则作为简单的行为规则^[6]：

- 1)向背离最近的同伴的方向运动
- 2)向目的运动
- 3)向群体的中心运动

在这个群体中每个个体的运动都遵循这三条规则，通过这个模型来模拟整个群体的运动。PSO算法的基本概念也是如此，每个粒子(Particle)的运动可用几条规则来描述，因此PSO算法简单，容易实现，越来越多地引起人们的注意。

2.1 基本原理

PSO算法是基于群体的，最初是为了在二维空间图形化模拟鸟群优美而不可预测的运动^[3]。PSO算法从这种模型中得到启示并用于解决优化问题。后来将其推广到N维空间，每个优化问题的解就是搜索空间中的一只鸟，在搜索空间中以一定的速度飞行，这个速度根据它本身的飞行经验和同伴的飞行经验来动态调整。鸟被抽象为没有质量和体积的微粒(点)，第*i*个粒子在N维空间里的位置表示为矢量 $X_i = (x_1, x_2, \dots, x_N)$ ，飞行速度表示为矢量 $V_i = (v_1, v_2, \dots, v_N)$ 。每个粒子都有一个由被优化的函数决定的适应值(Fitness Value)，并且知道自己到目前为止发现的最好位置(*pbest*)和现在的位置 X_i 。这个可以看作是粒子自己的飞行经验。除此之外，每个粒子还知道到目前为止整个群体中所有粒子发现的最好位置(*gbest*)(*gbest*是在*pbest*中的最好值)。这个可以看作是粒子的同伴的经验。每个粒子使用下列信息改变自己的当前位置：1)当前位置；2)当前速度；3)当前位置与自己最好位置之间的距离；4)当前位置与群体最好位置之间的距离。

PSO和上文的几种优化算法类似，是一种基于迭代的优化工具。对于第*k*次迭代，每个粒子是按照下式进行变化的：

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times \text{rand}() \times (p_{id} - x_{id}^k) + c_2 \times \text{rand}() \times$$

$$(p_{gd} - x_{id}^k) \quad (1)$$

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

在式(1)、(2)中, $i=1, 2, \dots, M$, M 是该群体中粒子的总数; v_{id}^k : 第 k 次迭代粒子 i 飞行速度矢量的第 d 维分量; x_{id}^k : 第 k 次迭代粒子 i 位置矢量的第 d 维分量; p_{id} : 粒子 i 个体最好位置 $pbest$ 的第 d 维分量; p_{gd} : 群体最好位置 $gbest$ 的第 d 维分量; c_1, c_2 : 权重因子; $rand()$: 随机函数, 产生 $[0, 1]$ 的随机数; w : 惯性权重函数。

公式(1)主要通过三部分来计算粒子 i 新的速度: 粒子 i 前一时刻的速度, 粒子 i 当前位置与自己最好位置之间的距离, 粒子 i 当前位置与群体最好位置之间的距离。粒子 i 通过公式(2)计算新位置的坐标。通过式(1)、(2)粒子 i 决定下一步的运动位置。在图1中, 以两维空间为例描述了粒子根据公式(1)、(2)从位置 X^k 到 X^{k+1} 移动的原理。

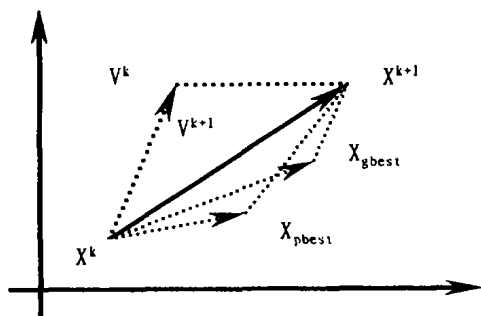


图1 粒子移动原理图

如果从社会学的角度来看公式(1), 其中第二部分(粒子 i 当前位置与自己最好位置之间的距离)为“认知(cognition)”部分^[7], 表示粒子的动作来源于自己经验的部分; 第三部分(粒子 i 当前位置与群体最好位置之间的距离)为“社会(social)”部分, 表示粒子的动作来源于群体中其它粒子经验的部分, 表现为知识的共享和合作。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。这与人类的决策何其相似, 人们通常也是通过综合自身已有的信息和从外界得到的信息来作出决策的。

2.2 参数分析

PSO 参数包括: 惯性权重函数 w , 权重因子 c_1, c_2 , 最大迭代次数 $iter_{max}$ 和群体规模 M 。通常权重函数 w 由下式来确定:

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \quad (3)$$

w_{max}, w_{min} 分别是 w 的最大值和最小值; $iter, iter_{max}$, 分别是当前迭代次数和最大迭代次数。通常权重函数 w 对于 PSO 算法的收敛性起到很大的作用。通过调整 w 的大小来控制以前速度对当前速度的影响, 使其是兼顾全局搜索与局部搜索的一个折中。因为 w 大, 则速度 V 就大, 有利于粒子搜索更大的空间, 可能发现新的解域; 而 w 小, 则速度 V 就小, 有利于在当前解空间里挖掘更好的解。因此, 在迭代开始时设 $w = w_{max}$, w 在迭代过程中逐步减小, 直到 $w = w_{min}$ 。这样使 PSO 更好控制探索(exploration)与开发(exploitation), 在开始优化时搜索较大的解空间, 得到合适的种子, 然后在后期逐渐收缩到较好的区域进行更精细的搜索以加快收敛速度。

权重因子 c_1, c_2 是调整粒子的自身经验与社会(群体)经验在其运动中所起的作用的权重。如果 $c_1 = 0$, 则粒子没有自身经验, 只有“社会(social-only)经验”^[7], 它的收敛速度可能较快, 但在处理较复杂的问题时, 容易陷入局部最优点。如果

$c_2 = 0$, 则粒子没有群体共享信息, 只有“自身经验”^[7], 因为个体间没有交互, 一个规模为 M 的群体等价于运行了 M 个单个微粒的运行, 所以得到解的几率非常小。

2.3 基本算法

PSO 算法的流程如下:

1) 初始化所有粒子(群体规模为 M)。在允许范围内随机设置粒子的初始位置和速度。每个粒子的 $pbest$ 设为初始位置, $pbest$ 中的最好值设为 $gbest$ 。

2) 评价每个粒子的适应值。计算每个粒子的目标函数。如果优于 $pbest$, 则 $pbest$ 被当前位置替换。如果所有粒子 $pbest$ 中有优于 $gbest$ 的, 则重新设置 $gbest$ 的索引号。

3) 根据公式(1)、(2)、(3)调整粒子的位置和速度。

4) 检查终止条件。如果达到最大迭代次数或者最好解停滞不再变化, 就终止迭代; 否则回到步骤(2)。

3. PSO 算法的发展

从 PSO 的基本算法发展出许多不同的版本, 这些版本是对基本 PSO 算法的改进或者是在某一方面应用的延伸。

3.1 离散二进制 PSO 算法^[8]

PSO 的基本算法最初是用来对连续函数进行优化的。而实际中许多问题是组合优化问题, 因此, Kennedy 和 Eberhart 博士在基本 PSO 算法的基础上发展了离散二进制 PSO 算法^[8]来解决这一类问题。

离散二进制 PSO 算法与基本 PSO 算法的主要区别在于运动方程(公式(1)、(2)), 离散二进制 PSO 算法的运动方程如下:

$$v_{id}^{k+1} = v_{id}^k + rand() \times (p_{id} - x_{id}^k) + rand() \times (p_{gd} - x_{id}^k) \quad (4)$$

$$\text{if } \rho_{id}^{k+1} < sig(v_{id}^{k+1}) \text{ then } x_{id}^{k+1} = 1; \text{ else } x_{id}^{k+1} = 0 \quad (5)$$

式(5)中, $sig(v_{id}^{k+1}) = \frac{1}{1 + \exp(-v_{id}^{k+1})}$ 是 sigmoid 函数; $\rho_{id}^{k+1} \in [0, 1]$ 是随机矢量 ρ^{k+1} 的第 d 维分量。粒子的位置只有(0, 1)两种状态, 而速度 v 与某一概率的阈值相关, 速度值越大, 则粒子位置取1的可能性越大, 反之越小。

在式(4)中没有权重函数 w 对速度进行调节, 为防止速度 V 过大, 原文设置了 V_{max} , 使函数 $sig(V)$ 不会过于接近0或1, 以保证一定的概率使算法从一种状态跃迁到另一种状态。

从上文可以看出, 除了式(4)、(5)不同, 离散二进制的 PSO 算法与基本 PSO 算法几乎一样。

3.2 混合 PSO 算法^[9]

混合 PSO 算法(Hybrid PSO, HPSO), 是将基本 PSO 算法和选择机制相结合而得到的。基本 PSO 算法的搜索过程很大程度上依赖 $pbest$ 和 $gbest$, 它的搜索区域受到 $pbest$ 和 $gbest$ 的限制。在通常的进化算法中, 选择机制用来选择相对较好的区域和淘汰较差的区域, 可以更合理地分配有限的资源。

混合 PSO 算法的选择机制与其它进化算法十分相似, 如遗传算法。混合 PSO 算法计算每个个体基于当前位置的适应值, 并将这些适应值进行排序, 然后将群体中一半适应值差的个体的当前位置和速度替换为另一半好的个体的当前位置和速度, 但保留每个个体的最好位置 $pbest$ 。因此, 群体搜索集中到相对较优的空间, 但还受到个体自身以前最好位置的影响。

混合 PSO 算法的流程如下:

1) 初始化所有粒子

2) 评价每个粒子的适应值

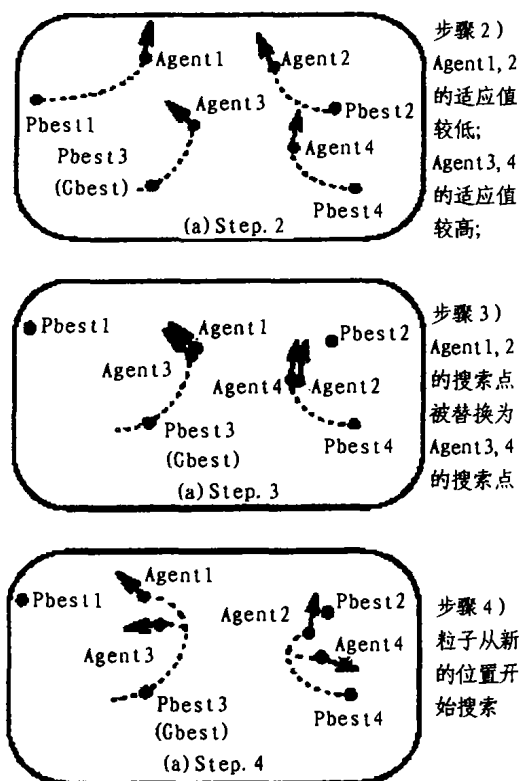


图2 混合 PSO 算法的流程图解

- 3) 使用适应值对粒子进行选择
- 4) 调整粒子的搜索位置, 粒子从新的位置开始搜索
- 5) 检查终止条件。如果达到最大迭代次数或者最好解停滞不再变化, 就终止迭代; 否则回到步骤(2)。

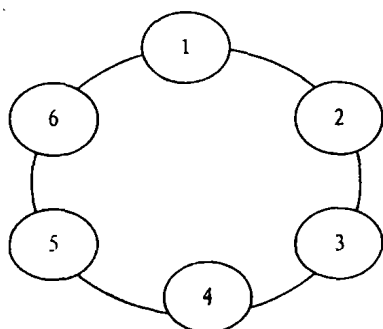


图3(a) 圆形“Lbest”模型

PSO 算法也被成功地应用于电力系统领域^[13~16]。这里主要涉及到带有约束条件的, 使用不同版本的 PSO 算法相结合用来决定对连续和离散的控制变量的控制策略的问题。

PSO 算法和其它进化计算算法类似, 能用于求解大多数优化问题。比如多元函数的优化问题, 包括带约束的优化问题^[17]。此外, PSO 还在动态问题^[18]和多目标优化问题^[19]中得到应用。

目前, 对 PSO 算法的研究才刚刚起步, 相对其它比较成熟的进化算法, 应用的范围小。但随着研究的进一步深入, PSO 算法将会应用到越来越多的领域中, 如电路设计、分类、作业调度、机器人路径规划等等。

结论与进一步的工作 PSO 算法是一个新的基于群体智能的进化算法, 其研究刚刚开始, 远没有像遗传算法和模拟退火算法那样形成系统的分析方法和一定的数学基础, 有许

混合 PSO 算法的流程如图2(a)、(b)、(c)。

3.3 Lbest 模型^[10]

Kennedy 和 Eberhart 博士称以上的 PSO 算法为“gbest”模型, 他们也发展了“Lbest”模型^[10]。在“Lbest”模型中粒子所能得到的信息只有它自己和最近的一组邻居的最好位置(lbest), 而不是整个群体的最好位置(gbest)。

“Lbest”模型主要有两种形式: 圆形和车轮形, 如图3(a)、(b)。

在圆形“Lbest”模型中, 个体只与其相邻的两个个体通信; 而在车轮形“Lbest”模型中, 个体之间的通信必须通过中间的个体。“Lbest”模型中个体的移动规则与“gbest”模型相似, 用 lbest 替换 gbest 就可以了。

4. PSO 的应用

已经有一些利用 PSO 算法代替反向传播算法来训练神经网络的论文。研究表明 PSO 算法是一种很有潜力的神经网络算法。PSO 算法速度比较快而且可以得到比较好的结果, 还没有遗传算法碰到的问题^[12]。PSO 算法被用于选择 BP 神经网络的权值^[10]。文[10]使用 BP 神经网络的输入层、隐含层和输出层分别有 5 个、3 个和 1 个神经元。他们使用标准 BP 算法训练, 需要大约 3.5 小时, 而使用 PSO 算法以后, 训练时间降为 2.2 分钟。此外, 用 PSO 算法进化神经网络的一个成功例子是用于分析人的颤抖。对人的颤抖的诊断, 包括帕金森(Parkinson)病和原发性震颤(Essential Tremor), 是一个非常具有挑战性的领域^[11]。PSO 算法被应用于进化一个神经网络, 这个进化神经网络能快速和准确地辨别正常颤抖和病态颤抖。

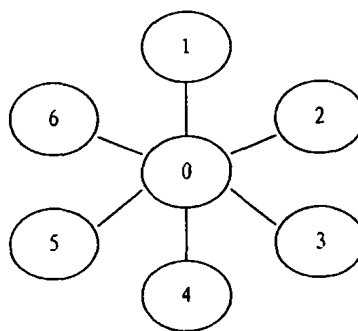


图3(b) 车轮形“Lbest”模型

多问题还需要进一步研究:

1) 适用范围。PSO 算法应用得最成功的是在进化神经网络方面。其它的一些应用许多还停留在研究阶段。显然, PSO 算法不会仅仅局限于目前的这些领域。如果将 PSO 算法引入机器学习、自动控制等领域, 将大大地促进算法的研究与发展。

2) 参数的选择与设计。PSO 算法中参数的选择依赖于具体问题, 设计合适的参数需要经过多次试验。研究如何选择和设计参数, 使其减少对具体问题的依赖, 也将大大促进 PSO 算法的发展和应用。

3) 算法的基础理论研究。PSO 算法的有效性在一些实例和数值实验中得到证明, 但算法性能的优劣需要理论上的保证。是否能够得到像模拟退火算法那样的结论, 如何指导参数的选择和设计, 如何设计适应值函数, 如何提高算法在解空间

里搜索的效率,算法收敛性以及算法模型本身的研究都需要在理论上进行更深入的探索,仅仅依靠试验方法是不够的。

随着 PSO 算法和一些相关领域学科的发展,PSO 算法一定会大显身手并有新的突破。本文希望读者能够对 PSO 算法原理、特点和应用范围有初步的认识,从深度和广度上进一步研究和发展 PSO 算法。

参考文献

- Kennedy J, Eberhart R. Swarm Intelligence. Morgan Kaufmann Publishers, 2001
- Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. Oxford Press, 1999
- Kennedy J, Eberhart R. Particle Swarm Optimization [A]. In: Proc. IEEE Int. Conf. on Neural Networks [C], 1995. 1942~1948
- Eberhart R, Kennedy J. A New Optimizer Using Particle Swarm Theory [A]. In: Proc. 6th Int. Symposium on Micro Machine and Human Science [C], 1995. 39~43
- Wilson E O. Sociobiology: The News Synthesis. Belknap Press, Cambridge, MA, 1975
- Reynolds C. Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics, 1987, 21(4): 25~34
- Kennedy J. The particle swarm: social adaptation of knowledge [A]. In: Proc. IEEE Int. Conf. on Evolutionary Computation [C], 1997. 303~308
- Kennedy J, Eberhart R. A discrete binary version of the particle swarm optimization algorithm. In: Proc. of the 1997 Conf. on Systems, Man, and Cybernetics (SMC'97), 1997. 4104~4109
- Angeline P. Using Selection to Improve Particle Swarm Optimization. In: Proc. of IEEE Intl. Conf. on Evolutionary Computation (ICEC'98), Anchorage, May 1998
- Eberhart R C, Kennedy J. Swarm Intelligence. Morgan Kaufmann, San Diego. 2001. ISBN 1-55860-595-9
- Eberhart R, Hu Xiaohui. Human tremor analysis using particle swarm optimization [A]. In: Proc. of the Congress on Evolutionary Computation [C], 1999. 1927~1930
- Hu Xiaohui. 粒子群优化算法介绍. <http://icdweb.cc.purdue.edu/~hux>
- Fukuyama Y, et al. A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment. IEEE Transaction on Power Systems, 2000, 15(4)
- Fukuyama Y, Yoshida H. A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems. In: Proc. of Congress on Evolutionary Computation (CEC2001), Seoul, Korea. Piscataway, NJ: IEEE Service Center, 2001
- Naka S, Genji T, Yura T, Fukuyama Y. Practical Distribution State Estimation Using Hybrid Particle Swarm Optimization. In: Proc. of IEEE Power Engineering Society Winter Meeting, Columbus, Ohio, USA, 2001
- Yoshida H, Kawata K, Fukuyama Y, Nakanishi Y. A particle swarm optimization for reactive power and voltage control considering voltage stability. In: G. L. Torres, A. P. Alves da Silva, eds. Proc. of Intl. Conf. on Intelligent System Application to Power Systems (ISAP'99), Rio de Janeiro, Brazil, 1999. 117~121
- Ray T, Liew K M. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problems [A]. In: Proc. IEEE Int. Conf. on Evolutionary Computation [C], 2001. 75~80
- Eberhart R, Shi Yuhui. Tracking and optimizing dynamic systems with particle swarms [A]. In: Proc. IEEE Int. Conf. on Evolutionary Computation [C], 2001. 94~100
- Parsopoulos K E, Vrahatis M N. Particle Swarm Optimization Method in Multiobjective Problems. In: Proc. of the 2002 ACM Symposium on Applied Computing (SAC 2002), pp. 603~607, ISBN: 1-58113-445-2
- Fukuyama-Yoshikazu. Fundamentals of Particle Swarm Optimization Techniques. <http://homepage2.nifty.com/fukuyama-yoshikazu/researches.htm>

怎样编写中、英文摘要

摘要是科技论文的重要组成部分,是一种以提供文献内容梗概为目的,不加评论和补充解释,简明、确切地记述文献重要内容的短文。其基本要素包括研究的目的、方法、结果和结论。摘要可大致分为 3 种类型:报道性摘要、指示性摘要、报道-指示性摘要。

报道性摘要 是指明一次文献的主题范围及内容梗概的简明摘要,相当于简介。报道性摘要一般用来反映科技论文的目的、方法或定量的信息,充分反映该研究的创新之处。

指示性摘要 是指明一次文献的论题及取得的成果的性质和水平的摘要,其目的是使读者对该研究的主题的主要内容(即作者做了什么工作)有一个轮廓性的了解。

报道-指示性摘要 这种摘要界于上述两者之间,以报道性摘要的形式表述一次文献中信息价值较高的部分,而以指示性摘要的形式表述其余部分。

1. 中文摘要一般不宜超过 200~300 字,外文摘要不宜超过 250 个实词。如遇特殊需要字数可以略多(为了扩大国际影响,英文摘要尽量写长一些,行文须符合英语习惯,不必与中文摘要一一对应)。

2. 除了实在无变通办法可用以外,摘要中不用图、表、化学结构式、非公知公用的符号和术语。

3. 英文时态以简练为佳,常用一般现在时、一般过去时。

4. 摘要中过去多用第三人称 This paper... 等开头,现在倾向于采用更简洁的被动语态或原形动词开头。

5. 行文时最好不用第一人称,以方便文摘刊物的编辑刊用。

总之,好的摘要既能使读者了解论文的主要内容,又能为科技情报人员和计算机检索提供方便。可以这样说,摘要质量的高低,直接影响着论文的被录用情况和期刊的知名度。