

基于种群熵的多粒子群协同优化^{*}

胡成玉^{1,2}, 吴湘宁², 王永骥¹

(1. 华中科技大学 控制科学与工程系, 武汉 430074; 2. 中国地质大学 计算机学院, 武汉 430074)

摘 要: 提出了一种基于种群熵的多粒子群协同优化算法, 通过引入熵对种群粒子的分布性进行度量, 然后利用它来引导在多种群协同演化中粒子迁徙的时间和方向, 从而保持粒子在寻优过程中的多样性和快速性。通过四个典型测试函数的仿真说明了该算法具有摆脱局部极值能力和较高的收敛速度。

关键词: 种群熵; 粒子群优化; 协同

中图分类号: TP301.6 文献标志码: A 文章编号: 1001-3695(2008)12-3593-03

Co-evolutionary particle swarm optimization based on population entropy

HU Cheng-yu^{1,2}, WU Xiang-ning², WANG Yong-ji¹

(1. Dept. of Control Science & Engineering, Huazhong University of Science & Technology, Wuhan 430074, China; 2. School of Computer, China University of Geosciences, Wuhan 430074, China)

Abstract: This paper applied a modified CPSO based on population entropy in the context of ECPSO. The entropy was used to measure the diversity of the whole population and then guided the particles how to migrate. The ECPSO was tested on some benchmark optimization problems and the results show a superior performance compared with the standard PSO and CPSO.

Key words: population entropy; particle swarm optimization(PSO); co-evolutionary

粒子群优化(PSO)算法作为一种新的优化算法, 由于收敛速度快、参数设置少, 近年来受到众多学者的重视^[1,2]。它常被用于解决大量非线性、不光滑和多峰值的复杂问题优化, 现已广泛应用于许多科学和工程领域。但是, 人们在实际应用中发现, 对某些多峰、平坦函数优化问题, PSO在进化过程中种群多样性损失过快, 容易导致其陷入局部极值, 引起算法过早收敛。针对这一问题, 很多学者提出了改进的方法, 如增加粒子的多样性、引入进化选择机制^[3,4]以及引入空间邻域^[5]等。

另外, 由于微粒群算法是基于同种群体内信息共享的假设而提出的, 它反映了在一个种群中个体之间的合作关系。在自然界的生态系统中, 很多物种通过与其他物种相互作用来提高自己的生存能力。受此启发, Shi等人^[6]提出了一种基于两个种群协同进化求解极大极小值的算法。Asmara等人^[7]在此基础上又提出了一种改进算法ACPSO(accelerated co-evolutionary PSO), 提高了算法的收敛速度并将此算法应用到机器人手臂计算机扭矩控制中。El-Abol等人^[7]指出了两个相互合作的微粒群之间如何进行信息交换。在国内, 很多学者也对协同进化微粒群算法进行了研究。Niu等人^[8]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^[9]提出了一种两层结构的多粒子群协同优化算法。王俊年等人^[10,11]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。以上改进算法都存在一些问题, 即在多种群中粒子的迁移机制上主要依靠经验。

虽然协同微粒群进化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, 特别是在多种群合作协同

的过程中, 粒子通过何种方式迁移才能保持种群的多样性和收敛的快速性, 还需要进一步的研究。本文利用群熵的概念对粒子多样性进行量化, 从而在时间和方法上对多种群协同中的信息流动进行引导, 不仅保持了种群的多样性、还保持种群了收敛速度。

1 标准粒子群优化算法

Shi等人^[12]在研究基本粒子群算法收敛性的过程中, 发现在速度项加上一个权重 ω 对整个算法非常重要。于是在1998年的IEEE国际进化计算学术会议上发表了论文, 首次在速度进化方程中引入惯性权重, 即

$$V_t = \omega \times V_{t-1} + c_1 \times \text{rand} \times (P\text{Best} - P_t) + c_2 \times \text{rand} \times (g\text{Best} - p_t) \tag{1}$$

$$P_{t+1} = P_t + V_t \tag{2}$$

以上公式常被认为是粒子群算法标准版本。式中, ω 称为惯性权重, 它使微粒保持运动惯性, 使其有扩展搜索空间的趋势, 有能力探索新的区域。引入惯性权重 ω 可清除基本PSO算法对 $V_{\max}$ 的需要, 因为 ω 本身具有维护全局和局部搜索能力的平衡作用。这样, 当 $V_{\max}$ 增加时, 可通过减小 ω 来达到平衡搜索, 而 ω 的减小可使得所需的迭代次数变小。因此, 可以将 $V_{\max}$ 固定, 只对 ω 进行调节。通常的做法是: 采用自适应调整 ω 的策略, 即随着迭代的进行, 而线性减小 ω 的值。使得算法在迭代初期探索能力较强, 可以不断搜索新的区域; 然后开发能力逐渐增强, 使算法在可能最优解周围精细搜索。也有学者提出了一种用模糊规则动态调整 ω 的方法^[13]。

收稿日期: 2008-01-28; 修回日期: 2008-04-18 基金项目: 国家自然科学基金资助项目(60674105); 中国地质大学(武汉)优秀青年教师计划资助项目(CUGQNL0821)

作者简介: 胡成玉(1978-), 男, 湖北枣阳人, 博士研究生, 主要研究方向为演化计算、智能控制(huchengyu@cug.edu.cn); 吴湘宁(1972-), 男, 副教授, 博士研究生, 主要研究方向为计算机体系结构; 王永骥(1955-), 男, 教授, 博士, 主要研究方向为神经网络、智能计算。

2 基于群熵的多粒子群协同优化算法

2.1 群熵的定义

定义 在一种群所有适应值构成的集合 R 中, 存在真子集 $A_1, A_2, \dots, A_n$, 若 $A_1 \cup A_2 \cup \dots \cup A_n = R, A_1 \cap A_2 \cap \dots \cap A_n = \emptyset$, 集合 $A_1, A_2, \dots, A_n$ 中元素的个数分别是 $s_1, s_2, \dots, s_n$, 则定义该种群的熵为

$$E = - \sum_{i=1}^n p_i \lg(p_i) \tag{3}$$

其中
$$p_i = s_i / \sum_{i=1}^n s_i \tag{4}$$

由定义可知, 当种群中所有粒子的适应值都相同时, 熵取最小值 $E = 0$; 当种群中粒子的适应值分布越大, 则粒子在解空间越具有多样性和勘探能力。在多种群协同的过程中, 可以通过种群的熵度量种群中各粒子的分布状况, 从而决定需要迁移的粒子及其转移方向。

2.2 多粒子群协同

在生物界中不仅存在达尔文的“适者生存, 优胜劣汰”的自然进化规律, 同时还存在着多个个体或多个物种通过相互之间的合作而共同进化的自然规律。多粒子协同进化算法正是源于这种思想。在遗传算法中, 个体之间仅存在着竞争关系, 通过个体之间的竞争而淘汰差的个体, 保留优秀个体, 从而使得解群体不断逼近最优解。在协同进化算法中, 个体之间不仅存在竞争关系, 同时也存在相互合作、相互促进的关系, 各个子群体之间通过适应度的关联而共同进化。

多种群的协同方式分为竞争型协同进化和合作型协同进化, 在本文中采用后者。合作型协同进化是指某个群体中个体的与其他群体中个体通过一系列的合作, 并根据其对目标问题解决的贡献度来决定进化方向。

利用合作型协同进化求解问题时, 一般通过适应值共享实现, 在粒子迁移机制上对粒子的选择通常有如下几种方式: a) 选择最好的个体。选取当前各种群中的最好个体作为代表, 但在有些情况下, 这种策略显得太贪婪。b) 随机选择。从每个种群中随机选取个体的代表。c) 选择平均个体。显然, 合作方式的选择对于问题解向量的优劣有很大的影响, 以上这些方式较多地依赖于经验和具体问题。

2.3 基于群熵的多粒子群协同函数优化

在本文提出的 ECPSO 算法中, 利用两个种群进行协同演化, 而在选择迁移的粒子及迁移的方向上, 主要依据种群的熵进行度量。主要策略是: 熵值较大的种群中, 适应值较大一部分粒子向熵值较小的种群迁徙, 随机替换相同数目的粒子; 而在熵值较小的种群中, 最优的粒子向熵值较大种群移动, 替换其最差粒子。通过这种迁移策略, 保证了种群始终能够保持一定的多样性, 也保证了种群全体的快速性。算法流程如图 1 所示。

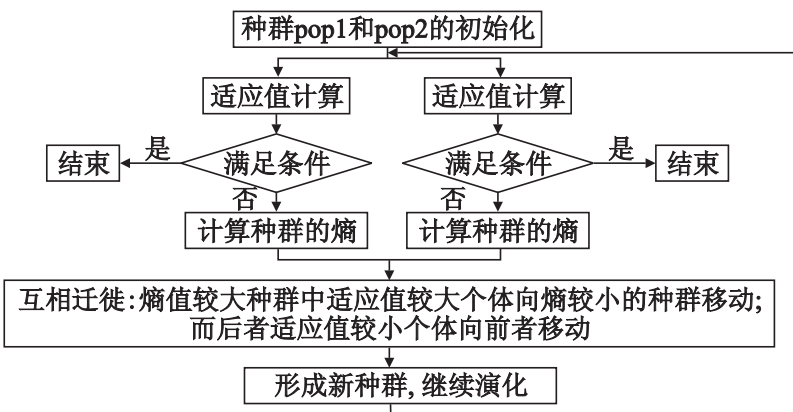


图1 基于种群熵多粒子协同算法流程图

3 仿真及结果分析

3.1 实验设计

为了测试本文提出的基于种群熵的协同粒子群(ECPSO) 算法, 本文把本算法和协同粒子群算法(CPSO) 及标准的粒子群算法(SPSO) 相比较, 实验选用四个常用于优化算法比较的基准函数。

$$F_1 = \sum_{i=1}^{10} x_i^2, x \in [-10, 10] \tag{5}$$

$$F_2 = 0.5 + (\sin \sqrt{x^2 + y^2} - 0.5) / [1 + a(x^2 + y^2)]^2, x, y \in [-10, 10] \tag{6}$$

$$F_3 = 20 + x^2 - 10 \cos(2\pi x) + y^2 - 10 \cos(2\pi y), x, y \in [-10, 10] \tag{7}$$

$$F_4 = [\sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)] / n, x \in [-5, 5] \tag{8}$$

四个优化函数非常具有代表性: F_1 是一个球面函数, 只有一个全局最优解 0, 分布在(0, $\dots$, 0) 处; F_2 函数是二维 Schaffer's f6 函数, 全局最优解为 0, 分布在(0, 0) 处, 特点是全局极小值被一群局部极小值包围, 很难跳出局部极小值; F_3 是 Rastigrin 典型函数, 全局最优解为 0, 分布在(0, 0) 处, 它是一个典型的欺骗函数, 其特点是在极小值附近, 存在多个局部极大和极小, 局部极大值和局部极小值如树状林立, 很难跳出局部最优; F_4 函数比较平坦, 所以在寻找极小值的过程中, 一般的算法收敛速度比较慢。图 2 ~4 分别是函数 F_2 、 F_3 和 F_4 的仿真曲面。

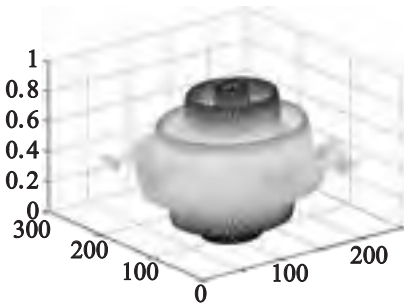


图2 函数 F_2 三维曲面

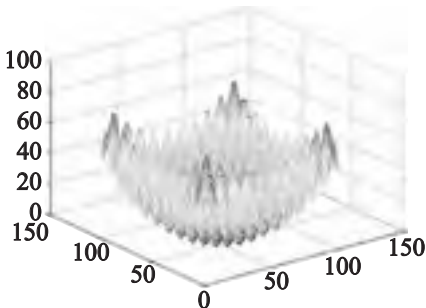


图3 函数 F_3 三维曲面

在本文中, 标准版本 SPSO 算法、基于贪婪机制的协同粒子群算法(CPSO)、基于群熵的协同粒子群算法(ECPSO) 具体参数设置为: 种群粒子为 30, 进化世代数为 300, 收敛的退出条件是精度达到 0.000 1 或者达到进化代数, $c_1 = c_2 = 2, \omega = 0.8$, $V_{\max}$ 为函数边界的 0.1 倍。最终对 20 次平均适应值进行评测, 以及对寻到极值的成功率进行统计。

3.2 种群熵值的计算

种群的熵值不能预知, 针对不同的优化函数及其边界, 在不同的优化代数内, 种群的熵值是变化的, 是一个逐步趋近于 0 的过程。因此, 使用如下方法对种群的熵进行估计: 首先估计解空间 S 及划分区域数 M 。解空间用 $[z_{\min}, z_{\max}]$ 来度量, 设算法运行中迄今为止所发现的最小和最大适应值分别为 $\min$ 和 $\max$, 则通过 $z_{\min} = \alpha \times \min, z_{\max} = \beta \times \max$ 对区间进行估计, 一般 $\alpha < 1, \beta > 1$ 。然后把区间 $[z_{\min}, z_{\max}]$ 等分或者非线性分成 M 个子区间, 在演化每一代计算各个区间粒子的个数。最后根据式(1) (2) 计算种群在每一代的熵值。

3.3 实验结果及分析

实验结果如图 5 ~8 所示, 分别表示待优化函数 F_1 、 F_2 、 F_3 和 F_4 在演化代数内的平均适应值曲线。SPSO、CPSO 和 ECPSO 算法均运行 20 次取平均最好适应值与其他算法进行比较, 结果如表 1 所示。

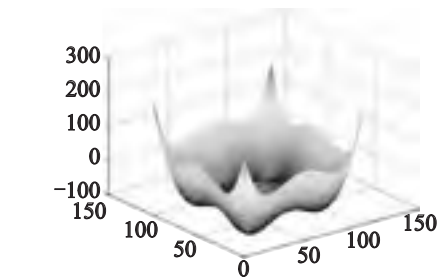


图4 函数 F_4 三维曲面

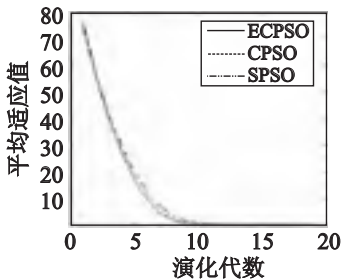


图5 函数 F_1 适应值曲线

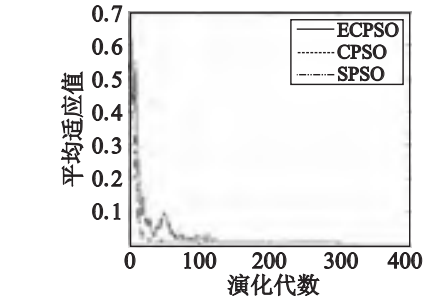


图6 函数 F_2 适应值曲线

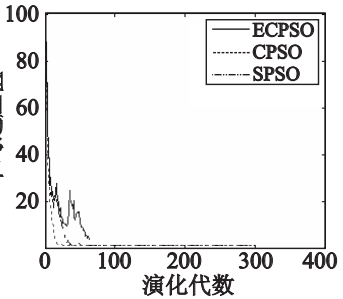


图7 函数 F_3 适应值曲线

表1 各算法运算的 20 次平均结果

函数	演化代数	最小适应值	成功率
表1 各算法运算的 20 次平均结果			
函数	演化代数	最小适应值	成功率 / %
F_1	SPSO	19	0.000 068 100
	CPSO	15	0.000 081 100
	ECPSO	16	0.000 068 100
F_2	SPSO	300	0.009 563 0
	CPSO	300	0.008 532 0
	ECPSO	234	0.000 091 35
F_3	SPSO	300	0.994 96 0
	CPSO	300	0.836 954 0
	ECPSO	72	0.000 096 100
F_4	SPSO	300	- 78.332 0
	CPSO	300	- 78.332 0
	ECPSO	300	- 78.331 0

对于函数 F_1 , 从图 5 和表 1 可以看到, 三种方法的寻优性能相差不多, ECPSO 稍微占优, CPSO 次之, SPSO 最差, 但是由于球面函数只有一个全局极值, 没有局部极值的干扰, 三种方法都可以收敛到极值。

对于函数 F_2 , CPSO 和 SPSO 虽然在收敛速度上占有优势, 但是很快收敛到局部极值。因为在全局极值附近有无数局部极值环绕。实验表明, 在 20 次寻优过程中, 收敛到最优值的几率为 0。而 ECPSO 在收敛过程中, 由于粒子是基于熵值互相移动, 适应值略有波动, 最终能够跳出局部极值, 找到最优解。虽然 ECPSO 不是每次都能收敛到全局极值, 在 20 次寻优中有 7 次可以跳出局部极值, 找到最优解, 成功率达到 35%。

对于函数 F_3 , CPSO 和 SPSO 相比, 收敛速度略占优势, 但是两者都很快收敛到局部极值。因为在全局极值附近有无数局部极值林立。实验表明, 在 20 次寻优过程中, 收敛到最优值的几率也为 0。而 ECPSO 在前期收敛速度稍慢, 但是经过平均 66 代演化, 均可以收敛到全局极值, 在 20 次收敛到全局的极值成功率为 100%。

对于函数 F_4 , 由于函数过于平坦, 三种算法在最优极值的寻找都有困难, 速度上相差也不多, 说明此类平坦函数三种算法效果均不太好, 还需要进一步的改进。

通过分析可以得知: CPSO 的迁移机制是采用基于最优值共享的方法, 这种方式比较贪婪, 使得种群收敛速度变快, 而容易陷入局部最优; 而在 ECPSO 中, 迁移之前首先计算每个种群的熵值, 根据熵的大小进行迁移, 这样不仅使得收敛的速度基本不变, 而种群的多样性可以保持。从计算复杂性的角度考虑, SPSO 算法较为简单, 而 CPSO 和 ECPSO 较为复杂, 但是 ECPSO 与 CPSO 相比难易程度相差无几, 性能却提高很多。

这里通过对函数 F_3 在寻优过程中两个种群在协同时熵值的变化曲线(图 9) 可以看出, 两个种群熵值开始都从 3 左右向 0.5 下降, 并且在下降过程中, 熵值交替变化, 说明两个种群通过粒子的迁移, 种群的多样性呈现交替变化, 从而证明了算法的有效性。

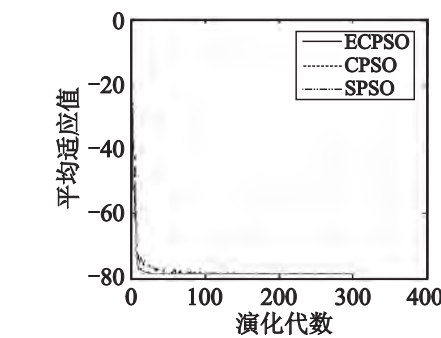


图8 函数 F_4 适应值曲线

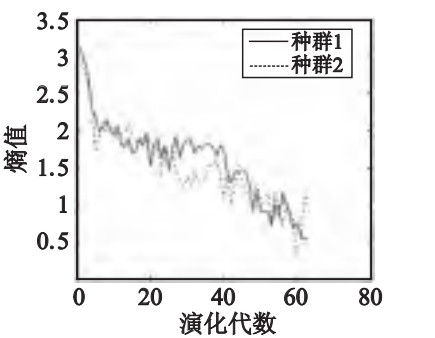


图9 函数 F_3 两个种群熵值变化曲线

4 结束语

本文采用种群的熵值对种群中粒子分布情况进行量化, 从而利用熵值引导两个种群在协同过程中粒子的迁移时机和方向。用 SPSO、CPSO 和本文提出的 ECPSO 算法对四个标准测试函数的优化表明, ECPSO 对大部分函数具有快速收敛性和跳出局部极值的能力, 对于平坦的函数还需要研究新的迁移机制, 才能保证收敛到全局极值, 这也是本文后续的研究内容。

参考文献:

- [1] ENNEDY J, EBERHART R C. Particle swarm optimization[C] // Proc of IEEE International Conference on Neural Networks. 1995.
- [2] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C] //Proc of the 6th International Symposium on Micro Machine and Human Science. Nagoya: [s. n.], 1995.
- [3] ANGELINE P J. Using selection to improve particle swarm optimization[C] //Proc of IEEE International Conference on Evolutionary Computation. Anchorage: [s. n.], 1998: 84-89.
- [4] LOVBJERG M, RASMUSSEN T K, KRINK T. Hybrid particle swarm optimizer with breeding and subpopulations[C] //Proc of the 3rd Genetic and Evolutionary Computation Conference. 2001.
- [5] SUGANTHAN P N. Particle swarm optimizer with neighbourhood operator[C] //Proc of Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1958-1962.
- [6] SHI Y, KROHLING R. Co-evolutionary particle swarm optimization to solving min-max problems[C] //Proc of IEEE Congress on Evolutionary Computation. Honolulu, Hawaii: [s. n.], 2002: 1682-1687.
- [7] EL-ABD M, KAMEL M. Information exchange in multiple cooperating swarms[C] //Proc of Conference on Genetic and Evolutionary Computation. 2005: 269-270.
- [8] NIU Ben, ZHU Yun-long, HE Xiao-xian. Multi-population cooperative particle swarm optimization[C] //Proc of the 8th Conference on ECAL. 2005: 874-883.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] SHI Y, EBERHART R C. A modified particle swarm optimizer [C] //Proc of IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998: 69-73.
- [13] SHI Y, EBERHART R C. Fuzzy adaptive particle swarm optimization [C] //Proc of IEEE Conference on Evolutionary Computation. 2001: 101-110.