

文章编号: 1001-0920(2008)10-1135-04

适应性粒子群寻优算法

罗辞勇, 陈民铀

(重庆大学 a. 电气工程学院, b. 输变电设备及系统安全与新技术国家重点实验室, 重庆 400044)

摘要: 社会性的群体寻优是秩序与混沌之间的平衡, 适应性微粒群寻优算法 (APSO) 是在标准 PSO 上添加反映适应性的随机项, 并引入小概率因子, 使微粒飞行到粒子群的中心, 平衡秩序和随机两个行为. APSO 算法的本质是在有序的决策中始终引入随机的、不可预测的决定, 从而使得寻优的决策尽可能模拟社会性群体寻优的复杂行为. 典型复杂函数优化的仿真结果表明, APSO 算法具有较好的稳定性.

关键词: 粒子群算法; 适应性; 随机

中图分类号: TP18 **文献标识码:** A

Adaptive particle swarm optimization algorithm

LUO Ci-yong, CHEN Mir-you

(a. School of Electrical Engineering, b. State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing 400044, China. Correspondent: LUO Ci-yong, E-mail: luociyong@cqu.edu.cn)

Abstract: The social behavior of swarm is a balance between complete order and total chaos. Therefore, the level of randomness in the group is an important factor. Firstly, the concept of "chaos" is introduced to the particle swarm optimization (PSO) by adding a durative random item representing principle of adaptability of swarm intelligence. Then, particles with small probability will fly to the center of the swarm, which is introduced to balance the order and random behaviour. The essence of adaptive particle swarm optimization (APSO) is that the inscrutable decision on the rational behavior is introduced to order decision, and the complex behaviour of social swarm is simulated. Experimental simulations show that the proposed method can improve the stability of convergence effectively.

Key words: PSO; Adaptive; Random

1 引言

PSO (Particle swarm optimization) 优化算法是由 Kennedy 等^[1]提出的一种基于种群搜索的自适应进化计算技术. 算法是受到飞鸟和鱼类活动的规律性启发, 用组织社会行为代替了进化算法的自然选择机制, 通过种群间个体的协作来实现对问题最优解的搜索.

与其他全局优化算法 (如遗传算法) 一样, 粒子群优化算法同样存在早熟收敛现象, 尤其是在较复杂的多峰搜索问题中^[2]. 另外, PSO 算法在种群初始化和微粒飞行过程中均具有随机性, 使得算法的执行结果也具有随机性^[3].

Millonas^[4]研究了群体在迁徙过程中的合作智能, Kennedy 等^[1]提出的 PSO 算法受到了 Millonas

研究成果的启发. 考察标准 PSO 算法, 存在着进一步模拟社会行为的可能性. Millonas 指出, 社会性的群体寻优应该是在秩序与混沌之间的平衡^[4]. 基于该想法, 本文提出了适应性粒子群优化算法, 在标准 PSO 上引入了反映适应性的随机项. 为了分析适应性粒子群优化算法的性能, 采用 4 个 Benchmark 函数进行测试, 并与标准 PSO, HPSO, DPSO 和 AEPSON 算法进行对比分析. 结果显示, 适应性粒子群优化算法具有较好的收敛稳定性.

2 粒子群优化算法及其改进算法

Millonas^[4]研究群体在迁徙过程中的合作智能, 提出了 5 个规律: 1) 接近规律, 种群能够做基本的空间和时间计算; 2) 质量规律, 种群不仅应该考虑空间和时间的花费, 也要考虑食物的质量和所在地

收稿日期: 2007-10-22; 修回日期: 2007-12-19.

基金项目: 重庆市自然科学基金项目 (CSTC2006BB2238).

作者简介: 罗辞勇 (1973—), 男, 安徽灵璧人, 讲师, 博士, 从事智能控制、阻抗成像的研究; 陈民铀 (1954—), 男, 重庆人, 教授, 博士生导师, 从事智能控制、数据建模等研究.

的安全;3)多样性规律,种群不应支配它所有的资源沿着过度狭窄的线路,应该分配它的资源选择多种模式来应对环境波动的突然变化;4)稳定性规律,种群在环境发生波动时,不应轻易改变自己的行为模式,由于这些改变需要能量消耗,可能不会对投资产生一个合理的回报;5)适应性规律,当改变行为习惯的报酬可能值得在能量上投资时,种群应该做出转变的决定.最好的决策是在完全秩序和彻底混沌中间寻找平衡,所以在种群中存在一定水平的随机性是很重要的.当然随机噪声会带来多样性,若它太大将会破坏合作的行为习惯.

Millonas 指出,很多复杂适应系统的行为可能会符合这些规则.有趣的是,这些规则同很多好的经济决定规律是类似的:时间就是金钱;宁可求稳,以免事后后悔;一鸟在手,胜于二鸟在林(最好满足于现有的,避免因贪心而失去一切);不要把你的鸡蛋放在一个篮子里面;为未来投资.

Kennedy 等人于 1995 年提出 PSO 的最初版本,其后 Shi 等人引入了惯性权重对算法进行了改进^[5],形成了当前的标准版本^[6].在每次迭代过程中,粒子的位置式确定如下:

$$v_i(t) = wv_i(t-1) + c_1 r_1 (p_i - x_i(t-1)) + c_2 r_2 (p_g - x_i(t-1)), \quad (1)$$

$$x_i(t) = v_i(t) + x_i(t-1). \quad (2)$$

式中: x_i 为粒子位置; v_i 为粒子飞行速度; w 为惯性权重; c_1 和 c_2 为加速度常数,取正常数; r_1 和 r_2 是在 $[0,1]$ 范围变化的随机数; p_i 为粒子在历史中的最好位置; p_g 为整个群体中的最好位置.

Kennedy 等人阐述了算法均满足 Millonas 提出的体现群体合作智能的 5 个规律.求解问题属于 n 维空间,并且花费一连串时间进行迭代^[1].种群飞向质量因子 p_i 和 p_g ,在 p_i 与 p_g 之间的随机分布保证了多样性.只有当 p_g 改变时,种群才改变自身状态,符合稳定性规则;当 p_g 改变时,种群也随之改变,符合自适应规律.但是 PSO 算法存在一个固有的缺陷,其粒子总是向全局最优、个体最优飞行,经过一定次数的迭代后,可能发生全局最优、个体最优和所有粒子都相同的情况.这时按式(1)和(2)迭代,所有粒子的速度为 0,粒子的位置不再发生变化,即发生早熟收敛.当发生早熟收敛后,显然对于自适应群体的 5 个规律全部不满足.

为了避免早熟收敛,一些学者提出了通过控制种群多样性来提高算法性能.提出了带选择的 PSO,具有高斯变异的 PSO,具有繁殖和子种群的 PSO 等改进算法;考虑到微粒间合理高效的相互作用关系,提出了簇分析 PSO,劳动分工 PSO 和协同

PSO 等;还有的学者提出了进化 PSO,多阶段 PSO 等^[7].

Xie 在 2002 年提出耗散型 PSO (DPSO)^[8],通过对微粒速度或位置引入一个小概率随机变异操作来增强种群多样性,使得该算法能够有效地进行全局搜索.但过大的变异率在增加种群多样性的同时也将导致种群发生混乱,使种群不能进行精确的局部搜索,延缓算法的收敛速度. Ratnaweera 等人提出了自适应层次 PSO (HPSO)^[9],在自组织算法的基础上给出了一种变异操作的自适应参数选择方式. HPSO 算法消除了速度公式的惯性部分,发生变异的条件是微粒速度为 0,使微粒不能快速、有效地逃出局部极小点.赫然等人在 2005 年提出自适应逃逸 PSO (AEP SO)^[13],当微粒飞行速度过小时,通过逃逸运动使微粒能够有效地进行全局和局部搜索,具有较快的收敛速度,但该算法存在不稳定性.

目前对 PSO 的改进使得 PSO 算法越来越复杂. Kennedy 等^[11]指出,为什么社会行为如此普遍存在于动物王国,因为它是优化的,解决工程优化问题比较好的途径就是模拟社会行为.仿效自然而不控制它,让算法看起来简单而不是复杂.大自然已经提供了第一流的信息处理技术.本文注意到 DPSO, HPSO 和 AEP SO 在确保种群多样性所进行的有价值的工作,这几种算法在引入随机行为产生多样性时都有各自发生的条件.本研究试图在更大的范围内引入随机行为,来进一步模拟社会型的群体行为,获取简单、具有稳定性的寻优算法.

3 适应性粒子群寻优算法

考察标准 PSO 算法.式(1)的第 1 部分为微粒先前的速度,第 2 部分为认知部分,表示微粒本身的思考;第 3 部分为社会部分,表示微粒间的信息共享与相互合作.这 3 部分都体现社会行为中的秩序,尽管在系数上给出各种随机变化,但只是改变遵守秩序的规则.最终的结果导致把鸡蛋放在同一个篮子里,没有为未来投资,缺乏多样性.哥伦布等探险家的行为体现出社会行为的另一个方面,作为遵守规则和秩序的反面,冒险、偏执、狂妄、嫉妒、逆反等心理和其他非理性行为同时支配社会行为,保持种群高度的多样性,以开拓更广阔的空间.社会性的群体寻优应该是在遵守秩序和随机非理性行为之间的平衡,文献[10]提出的自适应惯性权 PSO 算法中引入的随机因子和动量加速操作,实际上展现了这种随机非理性行为对保持优化解的多样性和克服早熟收敛所起的作用.为此,本文在标准 PSO 公式中,持续引入一个体现随机行为的项,形成了适应性粒子群寻优算法 (APSO).在每次迭代过程中,速度由

下式确定:

$$v_i(t) = wv_i(t-1) + c_1 r_1(p_i - x_i(t-1)) + c_2 r_2(p_g - x_i(t-1)) + (1 - 2r_3) R_i \quad (3)$$

式中: r_3 是在 $[0, 1]$ 范围变化的随机数; 为混沌常数, 取正值; R_i 为各维的搜索长度; $(1 - 2r_3) R_i$ 体现一种不可预测的随机行为, 命名为随机项. 式(3)体现了对速度的决策存在着正反面, 前 3 项体现秩序, 最后 1 项体现随机非理性. 将式(3)与式(1)相比, 可以看出即使出现发生全局最优、个体最优和粒子都相同的情况, 也能保证粒子进行运动, 并保持一种强劲的搜索欲望, 克服了早熟收敛问题. 即使已经找到了全局最优解, 随机项的存在依然促使粒子再度进行寻优.

控制随机决策行为的强弱, 过大会淹没遵守秩序的意识, 过小会降低种群多样性和全局搜索能力^[4]. 参考惯性权重采用随代数下降的策略可获得较好的寻优结果^[3,6]. 在算法中, 按式(4)的规律下降, 表示在寻优初期种群有强烈的随机行为, 而在寻优后期种群有较强的秩序, 有

$$w = \max(1 - \text{iter}/\text{maxitr})^2 \quad (4)$$

式中: \max 是混沌因子最大值, iter 为迭代次数, maxitr 为最大迭代代数.

Ratnaweera 认为, 速度惯性量的作用是作为自我认知和社会认知之间的平衡^[9]. 那么如何平衡秩序和随机这两个相反的行为呢? 向中心靠拢, 或称为中庸之道, 也是社会型群体智能的一种. 在研究中, 一般按照式(2)和(3)来确定粒子的位置, 较少让粒子飞行到群体的中心, 描述粒子的行为如下:

$$x_{id} = \frac{1}{N} \sum_{j=1}^N x_{jd} / N, \quad r < c_p \quad (5)$$

式中: c_p 是飞向中心的概率, N 代表种群规模. 实验表明式(5)的引入将会提高算法的收敛深度.

4 性能验证

为分析 APSO 算法的收敛稳定性和全局搜索性能, 本文选取 4 个 Benchmark 优化问题进行分析, 表 1^[3,8,9] 显示了这些 Benchmark 函数的定义和搜索

表 1 Benchmark 测试函数

函数名称	数学公式	搜索空间
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$(-100, 100)$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$(-50, 50)$
Rastrigrin	$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] + 10J$	$(-10, 10)$
Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	$(-300, 300)$

空间.

选择标准 PSO, DPSO, HPSO, AEPSO, 同 APSO 进行性能比较. 所有实验的维数 $d = 30$, 种群规模 $N = 60$, 最大迭代次数 $N_t = 4000$, 函数评价次数为 $N * N_t = 24000$, 最大速度为 0.5, 收敛因子为 0.01, 循环次数为 200. 表 2 显示了各个算法的实验参数设计. PSO, DPSO, HPSO, AEPSO 选用文献 [3, 8, 9] 中的参数. 测试结果见表 3.

对于 Sphere 函数, 几种算法均得到了 100% 的收敛概率, DPSO 和 AEPSO 体现出快速的收敛能力, 但是 APSO 体现出收敛的稳定性, 最优解、平均解和最劣解在数值之间是最为接近的.

对于 Rosenbrock 函数, AEPSO 取得较好的收敛概率和收敛的深度, 但收敛结果不稳定. APSO 的收敛概率和收敛深度性能同 DPSO 和 HPSO 的性能是类似的. 值得注意的是在最劣解当中, APSO 的值是最小的.

对于 Rastrigrin 函数, AEPSO 和 DPSO 取得了较好的收敛概率和收敛深度, 但也体现出 AEPSO 的不稳定性, AEPSO 存在收敛到 0 的概率, 但平均解 $3.98e - 002$ 没有小于收敛因子, 其解是不可预测的. APSO 虽然没有收敛, 但是最优解、平均解和最劣解在数值之间是最为接近的.

在 30 维多模态 Griewank 函数上, APSO 全局搜索能力优于标准 PSO 及另几种改进算法, 产生较高的收敛概率和最好的平均解. AEPSO 在 Griewank 函数上却存在较低的收敛概率和极大的不稳定性, 该结果同文献 [3] 的结论是一致的. Griewank 是一个复杂的多模态函数, 各维之间显著

表 2 微粒群算法参数设置

算法	w	c1	c2	其他
PSO	$[0.95 - 0.4]$	1.4	1.4	---
DPSO	0.7	1.4	1.4	$C_v = 0.001, C_t = 0$
HPSO	0.0	$[2.5 - 0.5]$	$[0.5 - 2.5]$	$v = 0.05$
AEPSO	0.7	1.4	1.4	$K_1 = 10, K_2 = 10, T_d = 1.0e - 10$
APSO	$[0.95 - 0.35]$	$[2.5 - 0.5]$	$[0.5 - 2.5]$	$\max = 0.001, c_p = 0.05$

表3 Benchmark 测试函数对比测试结果

函数	算法	最优解	平均解	最劣解	最快收敛代数	平均收敛代数	收敛概率 / %
f_1	PSO	2.98e - 037	1.12e - 022	5.27e - 021	966	1039	100
	DPSO	7.59e - 015	8.67e - 014	5.74e - 013	210	238	100
	HPSO	3.42e - 028	2.84e - 025	1.87e - 024	592	658	100
	AEPSO	7.28e - 032	1.78e - 029	1.03e - 028	140	232	100
	APSO	2.63e - 017	3.56e - 017	4.43e - 017	992	1050	100
f_2	PSO	4.16e + 000	4.85e + 002	1.00e + 004	0	0	0
	DPSO	1.05e - 001	2.51e + 001	7.62e + 001	0	0	0
	HPSO	1.51e - 001	2.06e + 001	8.10e + 001	0	0	0
	AEPSO	2.18e - 004	8.90e + 000	6.91e + 001	1880	3845	16
	APSO	4.11e + 000	3.22e + 001	6.17e + 001	0	0	0
f_3	PSO	2.29e + 001	3.62e + 001	5.07e + 001	0	0	0
	DPSO	4.02e - 008	4.82e - 001	2.99e + 000	1674	3281	62
	HPSO	2.98e + 000	9.79e + 000	2.29e + 001	0	0	0
	AEPSO	0.00e + 000	3.98e - 002	9.95e - 001	1352	2446	96
	APSO	1.19e + 001	2.89e + 001	5.27e + 001	0	0	0
f_4	PSO	0.00e + 000	2.37e - 001	2.25e + 001	1262	2416	52
	DPSO	4.22e - 015	1.35e - 002	7.38e - 002	198	1999	54
	HPSO	1.11e - 016	7.58e - 003	9.53e - 002	648	1921	65
	AEPSO	0.00e + 000	3.84e - 001	8.88e - 001	220	3650	10
	APSO	9.60e - 004	5.07e - 003	3.24e - 002	2086	2778	90

相关,某一维位置的变化不能提供微粒的适应值,只有各维同时发生变化时才有可能提高微粒的适应值.DPSO 是进行小概率下的变异,HPSO 是在某维粒子速度为 0 时进行变异,AEPSO 是在某维速度小于 T_d 时进行逃逸,均不能做到各维同时变化,因此这几种算法逃出极小点的概率较小^[3].APSO 能够做到在各维同时发生随机变化,使得 APSO 在 Griewank 函数上发生了较高的收敛概率,并获得了最好的平均解.

5 结 论

在标准 PSO 中,粒子总是向个体最优和群体最优的方向飞行,尽管在飞行过程中存在着随机性,但其决策行为的本身是有序的、理性的,最终粒子将聚集在最优粒子位置.本文继续考察动物学的社会行为,根据 Millonas 的观点,提出一种新颖的改进算法适应性微粒群算法.

DPSO, HPSO 和 AEPSO 均通过随机操作来增强多样性.DPSO 进行极小概率下的变异,HPSO 在 DPSO 算法的基础上,将发生变异的行为调整为速度为 0 时,AEPSO 在 DPSO 和 HPSO 算法基础上将发生变异的行为调整为小于某一个不断下降的阈值^[3,8,9].本文的 APSO 与其不同之处在于:没有先决条件的,持续地在有序决策的基础上叠加随机非理性的行为.

适应性微粒群算法的本质是在有序的决策中始终引入随机的、不可预测的决定,从而使寻优的决策尽可能模拟社会性群体寻优的复杂行为.改进的算法虽然简单,但并没有在所有的测试函数上均取得

良好的收敛深度,却在稳定性方面取得了较好的结果,在秩序和随机行为共同支配的群体行为中随机性反而降低,稳定性提高.在计算速度不是主要矛盾时,算法的稳定性也越来越受到关注^[3].

算法收敛深度不够高的问题需要进一步解决,但出发点依然是进一步模拟社会型群体寻优行为.在算法模型上,有很多社会寻优行为值得参考,如对非理性决策行为导致不好的结果有适当的惩罚机制,优胜劣汰等.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [2] 吕振肃,侯志荣.自适应变异的粒子群优化算法[J].电子学报,2004,32(3):416-420.
(Lv Z S, Hou Z R. Particle swarm optimization with adaptive mutation[J]. Acta Electronica Sinica, 2004, 32(3): 416-420.)
- [3] 赫然,王永吉,王青,等.一种改进的自适应逃逸微粒群算法及实验分析[J].软件学报,2005,16(12):2036-2044.
(He R, Wang Y J, Wang Q, et al. An improved particle swarm optimization based on self-adaptive escape velocity[J]. J of Software, 2005, 16(12): 2036-2044.)
- [4] Make M Millonas. Swarms, phase transtions and collective intelligence [M]. Massachusetts: Addison Wesley Reading, 1994.

(下转第 1144 页)

成的计算误差.应用基于修正因子的模糊 PID 控制器在升力鳍系统中,使模糊控制器的量化因子、决策因子能够通过模糊规则来进行在线修正,改善了升力鳍系统的动态性能和稳态精度,使系统可以在各种海况下都能够取得良好的减摇效果.

参考文献(References)

- [1] 金鸿章,李国斌.船舶特种装置控制系统[M].北京:国防工业出版社,1995:93-120.
(Jin H Z, Li G B. Control systems of marine equipments [M]. Beijing: National Defence Industry Press, 1995: 93-120.)
- [2] 金鸿章,姚绪良.船舶控制理论[M].哈尔滨:哈尔滨工程大学出版社,2001.
(Jin H Z, Yao X L. Ship control principle [M]. Harbin: Harbin Engineering University Press, 2001.)
- [3] Zhang H P, Yao X L, Chen F, et al. Research on lift feedback fin stabilizer and its nonlinear controller[C]. SICE Annual Conf. Fukui: Fukui University, 2003.
- [4] 金鸿章,王科俊,吉明.智能技术在船舶减摇鳍系统中的应用[M].北京:国防工业出版社,2003.
(Jin H Z, Wang K J, Ji M. The application of intelligent technology used in fin stabilizers[J]. Beijing: National Defence Industry Press, 2003.)
- [5] Surendran S, Lee S K, Kim S Y. Studies on an algorithm to control the roll motion using active fins[J]. Ocean Engineering, 2007, 34(3/4): 542-551.
- [6] 李洪兴.变论域自适应模糊控制器[J].中国科学, 1999, 29(1): 32-42.
(Li H X. Adaptive fuzzy logic controllers with variable universe[J]. Science of China, 1999, 29(1): 32-42.)
- [7] 李洪兴.非线性系统的变论域稳定自适应模糊控制器[J].中国科学, 2000, 32(2): 211-223.
(Li H X. Adaptive fuzzy logic controllers of non-linear system with variable universe [J]. Science of China, 2000, 32(2): 211-223.)
- [8] 王亚英,张春慨,邵惠鹤.变论域知识约简算法[J].上海交通大学学报, 2002, 36(4): 566-569.
(Wang Y Y, Zhang C K, Shao H H. Variable universal knowledge reduction algorithm [J]. J of Shanghai Jiaotong University, 2002, 36(4): 566-569.)
- [9] Guzelkaya M, Eksin I, Yesil E. Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer [J]. Engineering Applications of Artificial Intelligence, 2003, 16(3): 227-236.
- [10] Amin Haj-Ali, Hao Ying. Structural analysis of fuzzy sets in relation to nonlinear PID control with variable gains[J]. Automatica, 2004, 40(9): 1551-1559.
- [5] Shi Y, Eberhart R. A modified particle swarm optimizer [C]. IEEE World Congress on Computation Intelligence. Anchorage, 1998: 69-73.
- [6] 谢晓锋,张文俊,杨之廉.微粒群算法综述[J].控制与决策, 2003, 18(2): 129-134.
(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-134.)
- [7] 汪镭,康琦,吴启迪.基于形式化描述的逻辑分层延迟 PSO 算法及应用[J].控制与决策, 2005, 20(6): 611-615.
(Wang L, Kang Q, Wu Q D. Logical layered-and-delayed PSO algorithm and application based on formalization description [J]. Control and Decision, 2005, 20(6): 611-615.)
- [8] Xie X E, Zhang W J, Yang Z L. A dissipative particle swarm optimization [C]. Proc IEEE Congress on Evolutionary Computation. Honolulu, 2002, 2: 1456-1461.
- [9] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
- [10] Mahfouf M, Chen M, Linkens D A. Multi-objective optimal design of alloy steels using adaptive weighted particle swarm optimisation [C]. Proc of Parallel Problem Solving from Nature-PPSN. Birmingham, 2004: 762-771.

(上接第 1138 页)