

具有追尾行为的自适应变异粒子群算法

刘炳全¹,黄崇超²

LIU Bing-quan¹,HUANG Chong-chao²

1.渭南师范学院 数学与信息科学系,陕西 渭南 714000

2.武汉大学 数学与统计学院,武汉 430072

1.Department of Mathematics and Information Science,Weinan Teachers University,Weinan,Shaanxi 714000,China

2.School of Mathematics & Statistics,Wuhan University,Wuhan 430072,China

E-mail:lbingquan@sohu.com

LIU Bing-quan,HUANG Chong-chao.Adaptive mutation particle swarm optimization algorithm with following behavior. Computer Engineering and Applications,2008,44(30):74-76.

Abstract: In order to avoid the premature convergence and shock of traditional particle swarm optimization algorithm,a novel adaptive mutation algorithm with following behavior is proposed.If particles' density in the visual region added around optimal particle is higher than the given criteria, their individual extreme value points will be adaptively mutated in given probability. Then consider whether they are updated when comparing with current ones.The mutation operation can't be stopped until it reaches the given mutation step,particles fly away the visual region or update the global extreme value point.The algorithm not only improves searching ability but also reduces computational cost.Simulation with testing function demonstrates that the novel method is effective and efficient.

Key words: Particle Swarm Optimization(PSO);following behavior;adaptive operator;mutation operator

摘要:针对基本粒子群算法容易早熟及算法震荡问题,提出了一种具有追尾行为的自适应变异粒子群算法,在最优粒子周围添加一个可视区域,如果可视区域内的粒子浓度超过给定标准,则对区域内粒子的个体极值点以一定概率进行自适应变异操作,通过与当前状态比较决定是否更新极值点,变异操作直至粒子离开可视区域、更新了全局极值点或者达到给定变异步数为止。算法增大了搜索能力,而且避免了多余的运算,减少了计算量。通过测试函数仿真验证,结果表明新算法不仅确保收敛、改善了收敛速度,而且有效避免了算法震荡。

关键词:粒子群算法;追尾行为;自适应算子;变异算子

DOI:10.3778/j.issn.1002-8331.2008.30.022 **文章编号:**1002-8331(2008)30-0074-03 **文献标识码:**A **中图分类号:**TP301

粒子群优化算法(Particle Swarm Optimization,PSO)是一种简单、快速、容易实现、基于迭代的群体智能优化算法,是Kennedy和Eberhart与1995年提出的^[1]。PSO起源于对一个简化社会模型的仿真,它和人工生命理论以及鸟类或鱼类的群集现象有十分明显的联系,是群集智能的代表算法之一。从社会认知学角度,PSO应用了如下简单道理:即群体中的每个个体都可以从邻近个体的出发和以往经验中受益,其理论基础主要的基本要素为:刺激的评价、与邻近的比较和对邻近领域的模仿。

近年来,粒子群算法作为一种解决复杂而困难优化问题的方法得到了深入的研究^[2-4]和应用^[5-6],总体来说,PSO算法主要特点为:(1)PSO算法是一种全局优化算法,不需要问题的梯度信息,可以从不同方向进行搜索;(2)算法简单,收敛速度快;(3)算法既可用于连续型非线性问题,也可应用到离散问题和

混合式非线性系统。但PSO算法还存在易陷入局部最优、算法后期易于震荡等缺点,针对这类问题,很多学者进行了研究^[7-8],他们的共同点是,在迭代进程中,按一定的概率对每个粒子进行变异操作,从而对粒子跳出局部最优起到了较好的效果。但是,在迭代进程中对所有粒子进行变异,固然可以增大算法的搜索能力,但却增加了多余的运算,大大增加了算法的计算量,而且没有研究算法后期的震荡问题。基于如上原因,本文提出了一种具有追尾行为的自适应变异粒子群算法,在最优粒子周围添加一个可视区域,如果可视区域内的粒子浓度超过一定标准,则对区域内的粒子的个体极值点以一定概率进行自适应变异操作,通过与当前状态的极值比较决定是否更新极值点,变异操作直至粒子离开可视区域、更新了全局极值点或者达到给定变异步数为止。新算法不仅确保算法收敛、防止早熟,而且改善了收敛速度,有效避免了算法震荡。通过测试函数进行仿真

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.70771079);渭南师范学院人才专项科研基金项目(No.07YkZ007)。

作者简介:刘炳全(1980-),男,讲师,研究方向:系统优化建模与算法;黄崇超(1957-),男,博士,教授,博士生导师,研究方向:系统优化建模与算法。

收稿日期:2008-03-31 **修回日期:**2008-06-16

验证, 结果表明新算法是有效的。

1 PSO 算法原理

令 $x_i=(x_{i1}, x_{i2}, \dots, x_{in})$: 粒子 i 的当前位置; $v_i=(v_{i1}, v_{i2}, \dots, v_{in})$: 粒子 i 的当前飞行速度; $P_i=(P_{i1}, P_{i2}, \dots, P_{in})$: 迄今为止粒子 i 所经历的最好位置, 称为个体极值点。对于最小化问题, 目标函数值越小, 对应的适应值越好。

设群体中的粒子数为 M , 群体中所有粒子经历过的最好位置为 $P_g(t)$, 称为全局极值点, 则:

$$P_g(t) \in (P_1(t), P_2(t), \dots, P_M(t))$$

$$f(P_g(t)) = \min\{f(P_1(t)), f(P_2(t)), \dots, f(P_M(t))\}$$

其中, f 为适应值函数。粒子根据如下公式来更新自己的速度和新位置:

$$v_{ij}(t+1) = \omega(t)v_{ij}(t) + c_1r_{1i}(P_{ij}(t) - x_{ij}(t)) + c_2r_{2i}(P_{gj}(t) - x_{ij}(t)) \quad (1)$$

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

其中, $i=1, 2, \dots, M, c_1, c_2$ 为加速常数, 通常在 0~2 间取值, r_1, r_2 为 [0, 1] 间随机数。 c_1 调节粒子飞向自身最好位置的步长, c_2 调节粒子向全局最好位置飞行的步长。 $\omega(t)$ 称为惯性因子, t 为当前迭代次数, T 为最大迭代次数。为了减少在迭代过程中粒子离开搜索空间的可能性, v_{ij}, x_{ij} 通常限定于一定范围: $v_{ij} \in [-v_{max}^j, v_{max}^j], x_{ij} \in [-x_{max}^j, x_{max}^j]$, 迭代中若位置和速度超过边界范围则取边界值。

2 粒子群算法的改进

2.1 正态变异算子

针对 PSO 算法在迭代过程中容易出现早熟情况, 本文对基本 PSO 算法添加正态变异算子^[5], 利用概率为 p_m 的正态变异因子对粒子个体极值点进行自适应变异操作, 步骤如下: 群体的每个粒子是由一个扰动向量 $\sigma=(\sigma_1, \sigma_2, \dots, \sigma_n)$ 和一个个体极值向量 $P_i=(P_{i1}, P_{i2}, \dots, P_{in})$ 组成, 如果 (σ, P_i) 是被选个体, 新个体 $(\bar{\sigma}, \bar{P}_i)$ 按下式变异:

$$\bar{\sigma}_i = \sigma_i \cdot \exp(N_i(0, \Delta\sigma)) \quad (3)$$

$$\bar{P}_{ij} = P_{ij} + \kappa \cdot N(0, \bar{\sigma}_i) \quad j=1, 2, \dots, \text{int}(p_m \cdot n) \quad (4)$$

其中: $\Delta\sigma$ 称为二级步长控制参数, $N_i(0, \Delta\sigma)$ 是相互独立的均值为 0, 方差为 $\Delta\sigma$ 的符合正态分布的随机数, κ 为增量系数。

2.2 自适应惯性因子

在粒子群算法中, 惯性算子决定了粒子先前速度对当前速度的影响程度, 从而起到平衡算法全局搜索和局部搜索能力的作用。可以用来控制算法的开发和探索能力, 较大的惯性因子将使粒子具有较大的速度, 从而有较强的探索能力; 较小的惯性因子将使粒子具有较强的开发能力。可以说算法的执行效果很大程度上取决于惯性因子的选取。常选择的两种自适应惯性因子为:

$$\omega(t) = \omega_{max} \exp(-\alpha \frac{t}{T}) \quad (5)$$

$$\omega(t) = \omega_{max} - t \cdot \frac{\omega_{max} - \omega_{min}}{T} \quad (6)$$

α 为控制参数, 取 $\alpha=1, \omega_{max}=0.9, \omega_{min}=0.4$ 。

2.3 速度压缩因子

粒子群算法中粒子通过跟踪两个“极值”来更新自己的位置, 算法结构简单, 运行速度较快。但是, 算法在解空间搜索时,

有时会出现粒子在一个极值点的附近来回“振荡”现象, 通过调整学习因子和惯性权重因子也无法完全避免, 而且这个极值点可能就是局部最优解。这是因为当粒子距离最优解较近时, 粒子速度下降较慢而导致的粒子“飞过”最优位置而产生“振荡”, 针对这种现象, 本文在公式(2)添加自适应压缩因子^[6]: $\beta=1+k\cos(\frac{t}{T}\pi)$, 令

$$x_{ij}(t+1) = x_{ij}(t) + \beta v_{ij}(t+1) \quad (7)$$

k 为调节因子。

2.4 追尾行为

在迭代进程中对每个粒子进行变异, 虽然可以增大算法的搜索能力, 但却增加了多余的运算, 从而增加了算法的计算量, 为此模拟人工鱼的追尾行为^[10]引进一个可视区域 $\Phi=\{x_i \neq P_g | d(x_i, P_g) \leq \text{visual}\}$, 其中 visual 称为可视距离, $d(x_i, P_g) = \|x_i - P_g\|$ 表示粒子 i 与全局极值点 P_g 的距离, 令 m_Φ 表示 Φ 内的粒子个数, δ 为 Φ 内的粒子的允许最大浓度, 如果 $\frac{m_\Phi}{M} \leq \delta$, 表示可视区域内粒子还不算拥挤, 可以允许其它粒子继续向全局极值点靠拢, 一旦 $\frac{m_\Phi}{M} > \delta$, 表明当前全局极值点附近粒子已经足够多, 为避免早熟对 Φ 内的粒子进行变异操作。

2.5 算法的迭代步骤

步骤 1 初始步

(1) 初始化 $M, \kappa, \delta, k, \text{visual}, \sigma, \Delta\sigma$, 置 $t=1$;

(2) 随机初始化所有粒子并进行评价;

(3) 将各粒子初始位置向量 $x_i(t)$ 作为个体极值点 P_i , 并计算出群体全局极值点 P_g , 设定最大迭代次数 T 。

步骤 2 迭代步

(1) 对第 i 个粒子按迭代式(5)(1)(7)或(6)(1)(7)计算 v_{ij}, x_{ij} 并评价所有粒子;

(2) 如果 $f(x_i(t+1)) < f(P_i)$, 置 $P_i = x_i(t+1)$;

(3) 如果 $f(x_i(t+1)) < f(P_g)$, 置 $P_g = x_i(t+1)$ 。

步骤 3 变异步

(1) 如果 $\frac{m_\Phi}{M} > \delta$, 对 Φ 内所有粒子的个体极值点按公式(3)

(4) 进行自适应变异操作并评价所有粒子;

(2) 如果 $f(x_i(t+1)) < f(P_i)$, 置 $P_i = x_i(t+1)$;

(3) 如果 $f(\bar{P}_i) < f(P_g)$, 置 $P_g = \bar{P}_i$;

(4) 当粒子离开可视区域、更新了全局极值点或者达到给定变异步数时, 步骤 3 结束。

步骤 4 停止步

如果满足收敛标准或者 $t+1 \geq T$, 算法结束; 否则, 置 $t=t+1$ 转步骤 2。

从步骤 3 可以发现: (1) 变异操作只有满足一定条件时才可以进行, 这样不但可以保证粒子的多样性, 防止早熟, 而且可以避免多余的运算; (2) 可视区域内的粒子可以进行不止一次变异, 增强了算法的搜索能力。由于算法中的变异条件与人工鱼的追尾行为类似, 因此称本算法为具有追尾行为的自适应变异粒子群算法。

3 仿真实验

为了分析算法的有效性, 采用以下函数进行仿真研究:

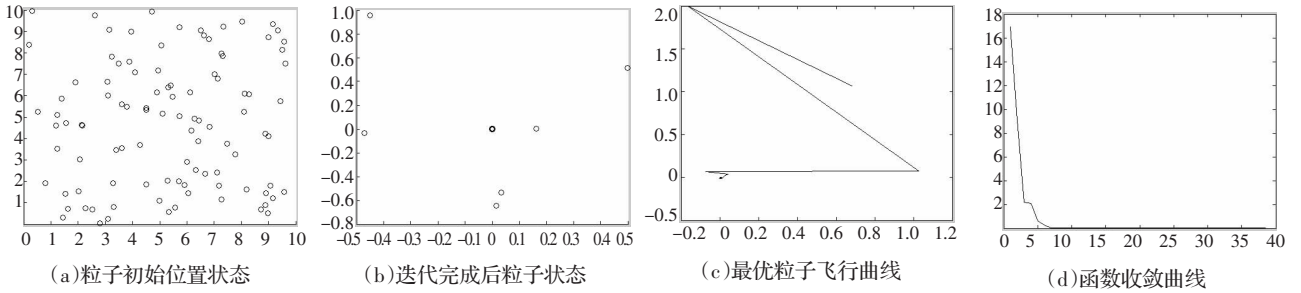


图1 Rastrigrin 函数改进粒子群算法仿真图

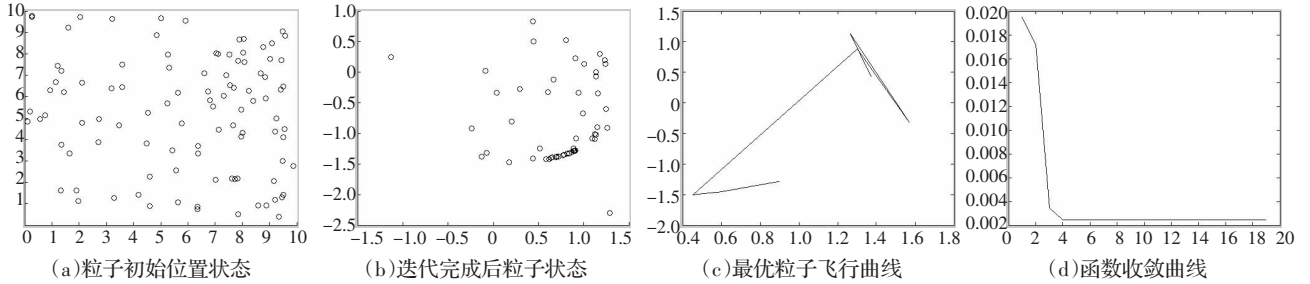


图2 Schaffer 函数改进粒子群算法仿真图

表1 Rastrigrin 函数、Schaffer 函数改进粒子群算法仿真结果

	Rastrigrin 函数仿真结果			Schaffer 函数仿真结果		
	x_1	x_2	$f_1(x)$	x_1	x_2	$f_1(x)$
1	-0.122 388 657 962 07 e-08	-0.046 098 026 840 75 e-08	0	0.900 613 046 888 30	-1.285 061 056 969 63	0.002 455 858 171 50
2	0.045 499 448 140 40 e-08	0.111 063 915 019 09 e-08	0	0.893 512 485 816 83	1.290 008 230 525 26	0.002 455 858 171 50
3	0.144 322 610 624 59 e-08	0.042 749 998 171 38 e-08	0	-1.325 287 151 059 25	0.840 297 426 095 02	0.002 455 858 171 50
4	-0.136 101 504 650 54 e-08	0.115 023 216 188 57 e-08	0	1.555 915 894 165 27	0.203 989 051 239 38	0.002 455 858 171 50
5	-0.012 710 106 608 53 e-08	-0.134 676 428 224 50 e-08	0	0.669 986 279 693 48	1.419 015 219 628 41	0.002 455 858 171 50
6	0.296 939 525 315 95 e-09	0.038 194 672 050 29 e-09	0	-0.674 684 538 039 89	-1.416 787 414 487 70	0.002 455 858 171 50
7	0.013 614 906 164 55 e-09	-0.803 100 247 255 41 e-09	0	-1.073 863 507 477 19	1.144 247 767 343 92	0.002 455 858 171 50
8	-0.119 924 202 696 43 e-08	0.121 317 754 463 47 e-08	0	-1.231 111 495 144 26	0.973 062 325 593 31	0.002 455 858 171 50

(1) Rastrigrin 函数:

$$f_1(x) = \sum_{i=1}^2 (x_i^2 - 10 \cos(2\pi x_i)) + 10$$

(2) Schaffer 函数:

$$f_2(x) = 0.5 - \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}$$

其中算法种群数为 $T=100$, $c_1=c_2=2$, 搜索区间为 $[-10, 10]$; 变异概率 $p_m=0.5$; 变异增量系数 $\kappa=5$; 浓度 $\delta=0.6$; 震荡调节因子 $k=0.6$; 可视距离 $visual=0.5$; 二级步长控制参数 $\Delta\sigma=0.5$, 扰动向量 $\sigma=[1, 1]$ 。图1、图2给出了 Rastrigrin 函数和 Schaffer 函数的仿真轨迹, 表1给出了对每个测试函数重复运行8次的仿真结果。

4 结束语

利用具有追尾行为的自适应变异粒子群算法, 针对 Rastrigrin 函数和 Schaffer 函数在 Matlab 环境下进行仿真实验, 可以发现: (1) 算法具有较快的收敛速度 (Rastrigrin 函数迭代7次, Schaffer 函数迭代4次既可达到全局极值点) 和强大的全局搜索能力; (2) 算法可以有效降低“震荡”问题; (3) 能够有效地避免早熟收敛问题, 为求解一些复杂的工程优化问题 (交通工程、电力系统) 提供了一种新方法。

参考文献:

- [1] Kenedy J, Eberhart R, Shi Y. Swarm intelligence [M]. San Francisco, CA, USA: Morgan Kaufmann Publisher, 2001: 287-300.
- [2] 李宁, 孙德宝, 岑翼刚, 等. 带变异算子的粒子群优化算法[J]. 计算机工程与应用, 2004, 40(17): 12-14.
- [3] 王丽, 王晓凯. 一种非线性改变惯性权重的粒子群算法[J]. 计算机工程与应用, 2007, 43(4): 47-48.
- [4] Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers & Operations Research (S0305-0548), 2005, 3(33): 857-871.
- [5] 付绍昌, 黄辉先, 肖业伟, 等. 自适应变异粒子群算法在交通控制中的应用[J]. 系统仿真学报, 2007, 19(7): 1562-1564.
- [6] 刘新颖, 王曙红, 邱捷. 改进粒子群算法及其在超导电缆参数优化中的应用[J]. 西安交通大学学报, 2007, 41(2): 219-222.
- [7] 黄辉先, 陈资滨. 一种改进的粒子群优化算法[J]. 系统仿真学报, 2007, 19(21): 4922-4925.
- [8] vanden Bergh F, Engelbrecht A. A study of particle swarm optimization particle trajectories[J]. Information Sciences, 2006, 176: 937-971.
- [9] 张建科, 刘三阳, 张晓清. 飞行时间自适应调整的粒子群算法[J]. 计算机应用, 2006, 26(10): 2513-2515.
- [10] 李晓磊, 路飞, 田国会, 等. 组合优化问题的人工鱼群算法应用[J]. 山东大学学报: 工学版, 2004, 34(5): 64-67.