

文章编号: 1001-0920(2009)02-0196-06

一种全局收敛的 PSO 算法及其收敛分析

高 浩¹, 冷文浩², 须文波¹

(1. 江南大学 信息工程学院, 江苏 无锡 214122; 2. 中国船舶科学研究中心, 无锡 214082)

摘 要: 在分析了速度因子对微粒群算法影响的基础上, 针对以往算法的弱点, 提出了一种基于 Gaussian 变异全局收敛的粒子群算法. 该算法使用全局变异因子使粒子具有了良好的全局搜索能力, 并证明了它能以概率 1 收敛到全局最优解. 同时使用了局部变异因子, 使算法在局部搜索过程中具有较高的搜索精度. 典型函数优化的仿真结果表明, 该算法具有寻优能力强、搜索精度高、稳定性好等优点, 适合于工程应用中的函数优化问题.

关键词: 粒子群; Gaussian; 收敛性; 变异; 蒙特卡罗模拟

中图分类号: TP18 **文献标识码:** A

A global convergence algorithm of particle swarm optimization and its convergence analysis

GAO Hao¹, LENG Wen-hao², XU Wen-bo¹

(1. School of Information Technology, Jiangnan University, Wuxi 214122, China; 2. China Ship Scientific Research Center, Wuxi 214082, China. Correspondent: GAO Hao, E-mail: gaohao_wuxi@yahoo.com)

Abstract: Based on the analysis of the mutation operator having impacts on the particle swarm optimization, a global Gaussian particle swarm optimization (GGPSO) is proposed to overcome the problem of previous algorithm. A global mutation operator is used to make the particle have excellent ability of search in a global scope. Meanwhile this algorithm can converge to the global optimization solution with probability one. Furthermore, for improving the searching ability in local area, the algorithm uses the local mutation operator to make the algorithm behave well in local searching. Experiment simulations show that the proposed algorithm has powerful optimizing ability, good stability and higher optimizing precision, so it can be applied in optimization problems.

Key words: Particle swarm; Gaussian; Convergence; Mutation; Monte Carlo simulation

1 引 言

由 Eberhart 等^[1]提出的粒子群优化算法 (PSO) 是一种基于种群搜索的自适应进化计算技术, 它源于对鸟群和鱼群群体觅食运动行为的模拟. 与其他生物进化算法类似, PSO 算法是一种基于迭代过程的优化方法. PSO 作为一种并行优化算法, 可用于解决大量非线性、不可微和多峰值的复杂问题优化. 目前在函数优化、神经网络训练、工业系统优化及控制、游戏设计^[2]等领域均取得了非常好的效果.

目前, PSO 算法作为一种崭新的随机搜索算法, 依然存在着早熟收敛以及局部搜索能力不强这两个难题. 为了避免早熟收敛, 一些研究者提出了通过控制种群多样性来提高算法性能的方法, 一方面

通过解决微粒间的小生境分割^[4]、种群随机多代初始化等思想, 给出了不同的策略来增强种群多样性, 使算法不至于过早陷入局部最小; 另一方面, 通过引入进化算法的“变异”等操作来改善种群的多样性也是一种十分有效的方法, 文献[5, 6]在标准的 PSO 速度和位置更新的基础上添加了 Gaussian 等变异参数, 以达到脱离局部最优点的目的 (GPSO). 但这几种方法只是在原有的粒子位置上作简单的变异, 并不能克服粒子群早熟收敛的弱点. 文献[7]在自组织算法的基础上, 给出了一种变异操作随时间变化的自适应层次 PSO 算法, 以进一步提高搜索性能, 并给出了适合变异操作的自适应参数选择方式. 但 HPSO 算法消除了速度公式中的惯性部分, 且其发生变异的条件是微粒速度为 0, 使微粒不能迅速、有

收稿日期: 2008-01-01; 修回日期: 2008-03-23.

基金项目: 国家自然科学基金项目 (60703106, 60474030).

作者简介: 高浩 (1976—), 男, 安徽蚌埠人, 博士生, 从事进化计算的研究; 须文波 (1946—), 男, 江苏无锡人, 教授, 博士生导师, 从事人工智能、计算机控制技术等研究.

效地逃出局部最小点.文献[8]提出了一种逃逸变异方法(AEPSO),当粒子的速度下降到某个程度上时,自动调整速度大小重新恢复搜索能力.但其局部搜索能力却因为频繁的变异操作而减弱,并且该算法受初始速度值以及搜索区间设置影响比较大,造成该算法不稳定.这些算法虽然使用了变异操作,但都没有很好地解决全局搜索与局部搜索平衡这一问题.

本文在分析了粒子群算法收敛性的基础上,针对粒子群优化算法的早熟收敛以及后期局部搜索能力弱的缺点,结合当前改进 PSO 算法的特性,引入了 Gaussian 变异因子的改进策略,提出了全局 Gaussian 粒子群优化(GGPSO)算法并给出了收敛性分析.该算法当每一维速度达到一定的阈值时,首先给予一个较大的速度值以保证全局搜索性能,同时对该维上达到该阈值的粒子给予较小的变异值以保证局部搜索能力.以几种典型复杂函数为测试算例,验证了该算法高效可行并克服了以前算法的种种缺点.

2 粒子群算法收敛性分析

考虑全局优化问题

$$(P) \min \{ f(x) : x \in R^n \},$$

$$f: R^n \rightarrow R^l,$$

则问题(P)的多个可行解的一个集合称为一个种群,种群中的每个元素(可行解)称为一个粒子,微粒的个数称为种群规模.用 n 维向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$ 来表示第 i 个粒子的位置,用 $V_i = (v_{i1}, v_{i2}, \dots, v_{in})^T$ 来表示第 i 个微粒的速度.微粒在搜索空间飞行过程中,它自身经历的最佳位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{in})^T$,也称为 p_{pbest} ,所有微粒经历过的最佳位置用索引号 g 表示,即 p_{gbest} .因此,微粒在每一代中的速度和计算函数的评价函数的位置可通过如下两个公式计算:

$$v_{id}(t+1) = \tilde{v}_{id}(t) + c_1 \text{rand}_1 (P_{pbest} - x_{id}) + c_2 \text{rand}_2 (P_{gbest} - x_{id}), \quad (1)$$

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

文献[3]虽然从位置角度对单个粒子运动轨迹进行了细致的分析,并给出了粒子群不能克服早熟收敛的结论,但该方法并没给出粒子速度的详细描述,而速度值是粒子运动过程中最重要的变异因素.同时,该文献在进化过程中将学习因子 $c_i \text{rand}_i$ 认为是常量,没有反映出对于速度的影响,因此有必要对速度变化进行详细的考察.对于标准 PSO 而言,每一维上的数据都是独立的,因此可以只分析单维粒子的变化,省去下标 d .根据式(1)和(2)可以得出

$$v(t+1) = \tilde{v}(t) + \omega_1 (P_{pbest} - x(t)) + \omega_2 (P_{gbest} - x(t)), \quad (3)$$

$$x(t) = v(t) + x(t-1), \quad (4)$$

$$\tilde{v}(t) = \frac{1}{\omega_3 + \omega_4} (v(t) - \tilde{v}(t-1) - \omega_3 P_{pbest} - \omega_4 P_{gbest}). \quad (5)$$

其中:式(3)和(5)中的 $\omega_i = c_i \text{rand}_i, i = 1, 2, 3, 4$.把式(4), (5)代入(3),得到

$$v(t+1) = (\omega_1 - s + q)v(t) - \omega_2 v(t-1) - (\omega_3 P_{pbest} - \omega_4 P_{gbest}). \quad (6)$$

式中

$$s = \omega_1 + \omega_2, \quad q = \frac{\omega_1 + \omega_2}{\omega_3 + \omega_4},$$

$$= \frac{\omega_1 + \omega_2 - \omega_3 - \omega_4}{\omega_3 + \omega_4}.$$

考察式(6)可以看出,第1项和第2项作为一个整体可称之为速度记忆项,是对过去速度值的记忆;而第3项可称之为速度学习项,它是 PSO 算法保证搜索性以及学习历史经验的关键所在.因为当学习项为 0 时,记忆项为保证粒子群收敛性而随之为 0,否则 $\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} (x(t-1) + v(t+1))$ 不存在.如果在这个过程中速度记忆项未能使粒子发现函数最优解,则粒子停止进化^[3].因此可以看出对于 $v(t)$ 来说,学习项对于跳出局部最小是关键因素.本文将重点考察学习项对于收敛性的影响.

对于学习因子 ω 而言,根据对称性可以计算出它的取值范围 $(-1.48, 1.48)$,均值为 0,但由于它包含 4 维变量,直接计算概率密度分布非常不方便,因此本文根据蒙特卡罗方法^[8]画出它的分布密度直方图.分析图 1 可以看出,学习因子 ω 主要集中在 $(-1, 1)$ 之间,结合式(6)可以看出学习项变化主要集中在 P_{pbest} 和 P_{gbest} 之间,当粒子陷入局部最优解时难以跳出,因此可以考虑给粒子以较大的速度值使之跳出局部最优.同时可以看出,学习项的因子的分布图和 Gaussian 分布图(见图 2)非常相似,因此用 Gaussian 分布来对学习项因子进行拟合以保持 PSO 算法的搜索特性.

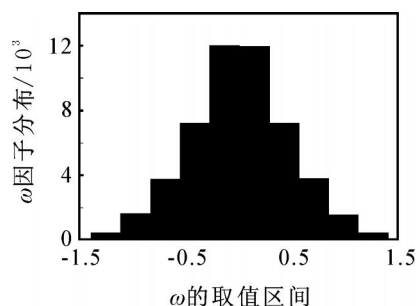


图 1 因子密度蒙特卡罗模拟直方图

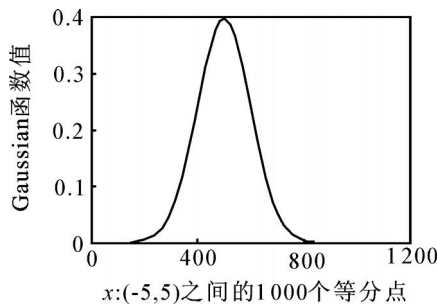


图2 Gaussian分布概率密度函数

3 基于 Gaussian 变异的 PSO 算法

3.1 引入 Gaussian 变异的粒子群算法

函数优化常采用状态发生器方式,即^[6]

$$x = x + v.$$

式中: v 为扰动幅度参数, v 为随机扰动. 随机扰动可服从 Gaussian 和均匀等分布, 不同的分布机制对算法的搜索行为和性能均有影响. 根据对式(6)的分析结果, 粒子群速度学习项因子和 Gaussian 分布比较类似, 因此使用的是 Gaussian 变异因子. 以原点为中心的 Gaussian 分布的概率密度函数为

$$f(v) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right), \quad (7)$$

式中: σ^2 为 Gaussian 分布的方差, σ^2 越小分布越集中在中轴线附近, 反之越远. 对应 $\sigma^2 = 1$ 时的函数曲线如图 2 所示.

根据对粒子群算法的分析可以看出, 当速度的值不发生变化时, 粒子群便丧失了搜索能力. 因此, 可通过控制微粒速度的方式来避免粒子群丧失搜索能力的弱点.

在粒子群迭代初期, 种群的多样性比较好, 但随着迭代过程的深入, 多样性随之下降, 因此在迭代初始时应给予粒子以较小的速度变异, 而到迭代后期给予粒子速度以较大的速度变异. 根据这种特点并结合 Gaussian 分布的性质, 本文引入了全局 Gaussian 变异因子以克服早熟收敛现象, 速度的变异具体依靠 v_{id}/v_{max} 的调节来实现, 同时更新粒子历史最优解的值以提高粒子的全局搜索能力. 过程描述如下:

$$\begin{aligned} & \text{if } k < 5 \text{ then} \\ & \quad v_{id}(t+1) = v_{max} (f(v_{id}(t)/v_{max})), \quad (8) \\ & \quad P_{\text{best}} = x_{id}(t) + v_{id}(t+1). \end{aligned}$$

在粒子群迭代过程中, 为了保证粒子群在每一维上的搜索精度, 克服因为全局变异操作而带来的局部搜索能力下降的缺点, 使用了局部 Gaussian 变异因子. 具体描述如下:

$$\text{if } k \geq 5 \text{ then}$$

$$x_{id} = x_{id} (1 + \text{sign}(f(x_{id}/x_{max}))). \quad (9)$$

为了给粒子较大的速度以脱离局部收敛区域, 式(7)中 Gaussian 分布的方差系数的取值为 0.16, 从而覆盖了整个的速度值. 如果 $k > 10$, 则 $T_{id} = T_d/10, k = 0$; 否则 $T_{id} = T_d, k = k + 1$. k 为计算每一维上粒子达到阈值 T_d 的次数, T_d 初始值为 v_{max} , $\text{sign} = \begin{cases} 1, & \text{rand} > 0.5; \\ -1, & \text{else.} \end{cases}$

另外, 局部变异因子的引入是为了提高种群的多样性, 因此它变异后的粒子值应直接参与迭代过程; 全局变异因子是为了克服单维空间上的早熟收敛现象, 因此其余维上的粒子值仍按照标准 PSO 算法公式计算. 同时粒子群速度在未下降到速度阈值之前, 依然使用标准粒子群公式进行计算.

3.2 GGPSO 算法收敛性分析

假设 1 1) 问题(P)的可行域为 R^n 中的有界闭区域; 2) 目标函数 $f(x)$ 是区域上的连续函数.

定义 1 设 $\{x_n\} (n = 1, 2, \dots)$ 为概率空间上定义的随机序列, 若存在随机变数 ϵ , 对 $\forall \epsilon > 0$, 若 $P\left\{\bigcap_{n=1}^{\infty} \{ |x_k - x_n| \geq \epsilon \}\right\} = 0$, 则称该随机序列以概率 1 收敛于 x^* .

定理 1 设 $x_{id} = v_{id}$, 则 $x_{id} \sim N(0, \sigma_{id}^2)$.

证明 粒子速度值在迭代过程中分别使用了公式(1), (8), (9) 作为位置变换公式. 式(1)为正常迭代时 GGPSO 的位置变化公式. 在进化过程中, 式(8)给予速度值下降到一定数值的粒子以较大的冲量, 使得 v 保持非零变量, 且 $\text{rand}_{1,2} \sim N(0, 1)$, 因此满足定理. 当速度值下降到规定阈值时, GGPSO 使用式(8)和(9)作为位置变化公式, 且两式中的变异因子符合 $N(0, \sigma^2)$ 分布. 因此定理得证.

引理 1 设问题(P)中的目标函数和可行域满足假设 1, 且 $x_{id} \sim N(0, \sigma_{id}^2)$. 若记 $x(t)$ 为状态 S_i 时, $x(t+1)$ 为状态 S_j 的概率为 $q_{ij} (i, j = 0, 1)$, 则有: 1) 对于任意状态为 S_0 的点集 $x(t)$, $q_{00} = 1$; 2) 对于任意状态为 S_1 的点集 $x(t)$, 存在常数 $c \in (0, 1)$, 使得 $q_{11} < c$. 其中: S_0 表示对于任给 $\epsilon > 0$, 由 GGPSO 算法产生的点至少有一个点满足 $D_0 = \{x \mid |f(x) - f^*| < \epsilon\}$; 状态 S_1 表示由本算法产生的点全部属于 $D_1 = \mathbb{R}^n \setminus D_0$. 详细证明参见文献[9].

定理 2 设 $\{x(m)\}$ 是由 GGPSO 算法产生的种群序列, 其中 $x^*(m) = x(m)$ 为第 m 代种群中的最优点, 即 $x^*(m) = \arg \min_u f(x_i(m))$, 如果问题(P)中的目标函数和可行域满足假设 1, 则有

$P\{\lim_m f(x^*(m)) = f^*\} = 1$ 成立,即种群序列以概率为 1 收敛于全局最优解.

证明 对于 $\forall \epsilon > 0$, 令

$$p_k = P\{|f(x^*(k)) - f^*| > \epsilon\},$$

有

$$p_k = \begin{cases} 0, & \exists T \quad (0, \dots, k), x^*(T) \in D_0; \\ \bar{p}_k, & x^*(m) \notin D_0, m = 0, 1, \dots, k. \end{cases}$$

由引理 1, $\bar{p}_k = \prod_{i=1}^k q_i$, $c^k = \frac{c}{1-c} < 1$, 及波雷尔 - 坎特里定理^[9] 知

$$P\{\lim_{n \rightarrow \infty} \prod_{k=1}^n [|f(x^*(k)) - f^*| > \epsilon]\} = 0,$$

因此根据定义 1 可知结论成立.

4 对比实验

4.1 实验设计

为了分析 EPSO 算法的总体性能,本文设计了 4 类测试实验:1) PSO 优化实验;2) GPSO 优化实验,变异率^[5] 设置为最佳数值 0.5, Gaussian 方差 设置为 0.1;3) AEPSO 优化实验;4) GGPSO 优化实验对比分析,引入 6 个 Benchmark 优化问题来进行分析,同时根据文献[10, 11] 对测试函数和测试标准的规定,函数形式、搜索范围以及评价函数标准设置见表 1,这些函数的理论极值均为 0. 所有实验的种群规模均为 20,速度最大值 $V_{max} = X_{max}$. 由于待优化函数的维数越高,自变量范围不对称,优化过程的难度就越大. 为了便于比较和突出改进算法的性能,本文的目标函数维数分别设置为 30 和 100 维(其中 Shaffer 函数为 2 维),自变量初始化范围不对称. 每个函数独立运行 10 次,每次运行 3000 代. 每个函数的惯性权重 \bar{w} 均设置为 0.7, $c_1 = c_2 = 1.4$, T_d 初始值设为 V_{max} . 实验结果中的数值“0”表示 Matlab 精度值 $1e-308$ (当 Matlab 小于这个值时显示为 0). 性能评估采用如下方法:1) 固定进化迭代次数,评估

算法收敛精度,其中比较参数为平均最优值(Mean Best),标准方差(St. Dev),最小值(Min);2) 固定收敛精度目标值,评估算法达到该目标所需要的平均耗费时间;3) 与文献报道的性能进行比较.

4.2 实验结果及分析

对于单模函数来说,由于使用了非对称初始化粒子群,显然加大了搜索难度,同时虽然 GPSO 与 AEPSO 同样使用了变异因子,但 GPSO 变异的条件仅是对当前位置进行局部变异,因此局部搜索能力并没有太大的减弱,在单模函数 f_3 上取得了较好的成绩. 而 AEPSO 算法由于其频繁地在每一维上使用单一的变异因子,影响了它的局部搜索能力及收敛速度,在单模函数上没有取得较好的效果. 而 PSO 和 GPSO 没有良好的全局搜索能力,一旦陷入到局部最小点时很难摆脱这个局面,因此在单模态函数 f_1, f_2 上效果也不是太好. GGPSO 相对于其他比较算法而言,在保证能够跳出局部最小点的情况下,依靠着局部变异依然在单模态函数上取得了很好的效果. 括号内的数字表明 GGPSO 达到最优解占总运行次数的百分比.

在多模态函数实验比较中,Rastrigrin 函数是一个典型的测试算法的全局搜索性能的函数. 从表 3 可以看出,GGPSO 由于使用了全局 Gaussian 变异因子,使其随着迭代次数的增加跳出局部最小点的可能也增大,从而有着更好的全局搜索能力. 而 AEPSO 依靠其全局搜索能力在这个函数上也取得了较好的成绩,但由于它的变异因子是随机产生的,使其并不能很有效地跳出局部最小点,没有 GGPSO 的效果好. 对于 Griewank 函数,它考察的是局部与全局搜索能力的平衡, AEPSO 由于频繁的全局变异操作使其不能很好地进行局部搜索,因此效果较差. 而 PSO 以及 GPSO 则相反,在这个函数上取得了比较好的成绩. 在函数 Shaffer 上,GGPSO, AEPSO 依

表 1 Benchmark 函数

Name and code	Formula	Rang [x_{min}, x_{max}]	X_{max}	Criterion
Sphere f_1	$\sum_{i=1}^n x_i^2$	(-50, 50)	100	0.01
Schwefel f_2	$10^6 x_1^2 + \sum_{i=2}^n (x_i^2)^2$	(-50, 50)	100	0.01
Rosenbrock f_3	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(15, 30)	100	100
Rastrigrin f_4	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	(-2.56, 5.12)	10	100
Griewank f_5	$\frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	(-300, 600)	600	0.05
Shaffer f_6	$0.5 + \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1 + 0.01(x^2 + y^2))^2}$	(-30, 100)	100	0.00001

注: $f_1 \sim f_3$ 为单模态函数, $f_4 \sim f_6$ 为多模态函数.

表2 GGPSO与其他算法在单模态 Benchmark 问题上的比较

Function	Algorithm	Mean Best	St. Dev.	Min
Sphere	PSO	16000	8.000E+3	9.4543E-46
	GPSO	8000	7.4833E+3	4.1563E-46
	AEPSO	64000	1.0198e+4	50000
	GGPSO	0	0	0
Schwefel	PSO	46000	3.9294E+4	10000
	GPSO	84000	2.7276E+4	50000
	AEPSO	1.54E+5	2.5768E+4	120000
	GGPSO	0	0	0
Rosenbrock	PSO	58.561	114.653	60.2791
	GPSO	32.327	21.4258	15.0187
	AEPSO	200.26	120.46	86.6772
	GGPSO	26.075	33.3034	3.3693

表3 GGPSO与其他算法在多模态 Benchmark 问题上的比较

Function	Algorithm	Mean Best	St. Dev.	Min
Rastrigrin	PSO	75.366	20.813	63.451
	GPSO	97.705	41.1607	66.7148
	AEPSO	15.272	4.5755	8.0679
	GGPSO	0.68165	0.7740	0(40%)
Griewank	PSO	50.38	47.1096	1.443E-15
	GPSO	88.034	50.359	0.0246
	AEPSO	157.79	82.65	63.91
	GGPSO	0.00958	0.0100	0(40%)
Shaffer	PSO	0.0097	0	0.0097
	GPSO	0.0038864	0.0048	0
	AEPSO	0	0	0
	GGPSO	0	0	0

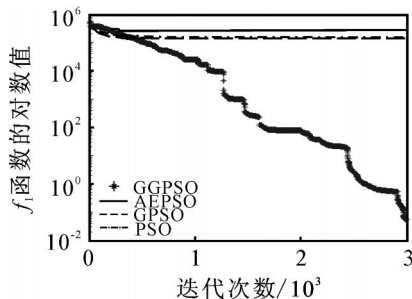


图3 100维 Sphere 函数迭代比较图

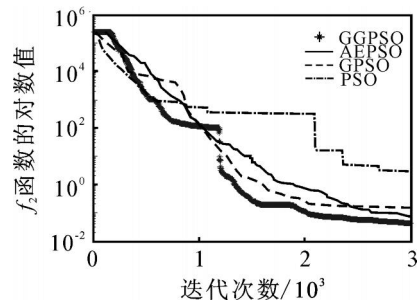


图4 100维 Rosenbrock 函数迭代比较图

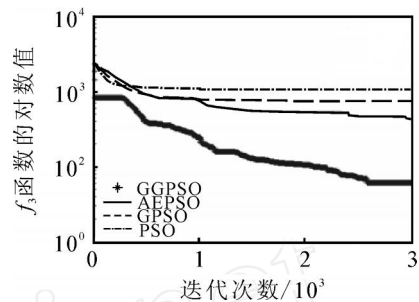


图5 100维 Rastrigrin 函数的迭代比较图

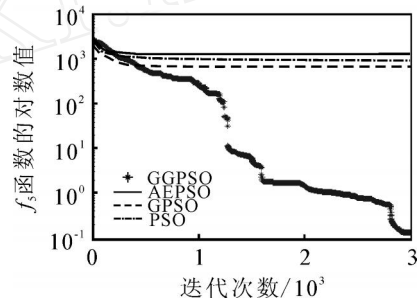


图6 100维 Griewank 函数迭代比较图

靠着良好的全局搜索能力获得了较好的效果,而GPSO凭借着局部变异,也达到了较高的精度。

大量的数据表明,GGPSO的寻优能力无论在低维函数优化还是高维函数优化问题上的寻优能力都比AEPSO,GPSO和PSO算法要好.特别值得注意的是,AEPSO,GPSO以及PSO算法的寻优能力随着维数的增加明显减弱,而GGPSO算法的寻优能力以及收敛速度却随着维数的增加并没有减弱.从图1~图4中可以清晰地得出结论,在AEPSO,GPSO以及PSO基本丧失了寻优能力的情况下,GGPSO的寻优能力没有受到维数的影响.对于Rosenbrock这个函数值梯度下降的单模函数来说,由于

表4 30维函数且固定进化精度下的比较

算法	时 耗					
	Sphere	Schwefel	Rosenbrock	Rastrigrin	Griewank	Shaffer
PSO	*	*	1.4188	0.59375	*	0.1375 *
GPSO	*	*	3.4125	0.64063	*	2.0938
AEPSO	*	*	2.3506	1.5938	*	0.078125
GGPSO	1.3063	1.7125	1.2719	0.45188	1.20313	0.046875

AEPSO, GPSO 含有变异因子而使其达到了较好的效果. 同时根据表 4 的结果可以看出, GGPSO 的算法在固定搜索精度的同时收敛速度依旧最优, 因此算法并没有单纯地依靠牺牲时间来提高精度, 在速度以及精度之间达到了一个良好的结合.

5 结 论

本文借鉴生物界大量存在的变异现象, 针对粒子群算法中的早熟收敛以及局部搜索能力差的问题, 提出了一种全新的粒子群算法 (GGPSO). 该算法继承了 PSO 算法简洁以及收敛速度快的优点, 同时在保证全局寻优的基础上, 很大程度地提高了搜索精度和收敛速度, 达到了其他比较算法无法达到的好成绩. 在进一步的工作当中, 主要研究如何更好地协调局部与全局搜索能力. 另外, 更为高效简洁的算法是研究的主要目标.

参考文献(References)

[1] Kennedy J, Eberhart R C. Particle swarm optimization [C]. IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.

[2] Kalyan Veeramachneni, Lisa O, Ganapathi K. Probabilistically driven particle swarms for optimization of multi valued discrete problems: Design an analysis [C]. Proc of IEEE Swarm Intelligence Symposium (SIS). Honolulu, 2007: 141-149.

[3] Van den Bergh F. An analysis of particle swarm optimizers[D]. Pretoria: University of Pretoria, 2001.

[4] Engelbrecht A P, Masiye B S, Pampara G. Niching ability of basic particle swarm optimization algorithms [C]. Proc of IEEE Swarm Intelligence Symposium

(SIS). Pasadena, 2005: 397-400.

[5] Paul S Andrews. An investigation into mutation operators for particle swarm optimization [J]. IEEE Congress on Evolutionary Computation, Vancouver, 2006, 16(21): 1044-1051.

[6] Higashi N, Iba H. Particle swarm optimization with Gaussian mutation [C]. Proc of IEEE Swarm Intelligence Symposium(SIS), 2003: 71-79.

[7] HU X H, Eberhart R C. Adaptive particle swarm optimization: Detection and response to dynamic system [C]. Proc of the IEEE Conf on Evolutionary Computation. Honolulu, 2002: 1666-1670.

[8] 赫然, 王永吉, 王清, 等. 一种改进的自适应逃逸微粒群算法及试验分析 [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.

[9] 郭崇慧, 唐焕文. 演化策略的全局收敛性 [J]. 计算数学, 2001, 23(1): 105-110.
(Guo C H, Tang H W. Global convergence properties of evolution strategies [J]. Mathematica Numerica Sinica, 2001, 23(1): 105-110.)

[10] Rui Mendes. Population topologies and their influence in particle swarm performance [D]. Escola de Engenharia: University of Minho, 2004.

[11] Daniel Bratton, James Kennedy. Defining a standard for particle swarm optimization [C]. Proc of IEEE Swarm Intelligence Symposium (SIS). Honolulu, 2007: 120-127.

下 期 要 目

欠驱动水面船舶运动控制研究综述 郭 晨, 等

直觉模糊集隶属度与非隶属度函数的确定方法研究 邢清华, 刘付显

电磁轴承增益调度多目标控制器设计 尚安利, 等

对角 CARIMA 模型多变量自适应约束广义预测控制..... 李奇安, 李 平

基于一阶混杂 Petri 网的柔性制造系统粗糙控制..... 廖伟志, 等

基于 K-means 聚类和数据场理论的复杂网络社团结构探寻 高忠科, 金宁德

基于自学习模糊 Petri 网的知识化制造系统采购预测..... 孟宪刚, 严洪森

基于直觉模糊粗糙集的属性约简 路艳丽, 等

路网辅助的车载 INS/ GPS 组合导航方法..... 徐田来, 等

一种面向效用值偏好信息的大群体决策方法 徐选华, 等