

基于高斯变异的量子粒子群算法

石永生^{1,2} 陈家琪¹

(1. 上海理工大学光电信息与计算机工程学院, 上海 200093 2. 江苏信息职业技术学院计算机工程系, 江苏 无锡 214101)

摘 要 粒子群算法相对于其他优化算法来说有着较强的寻优能力以及收敛速度快等特点,但是在多峰值函数优化中,基本粒子群算法存在着早熟收敛现象。针对粒子群算法易于陷入局部最小的弱点,提出了一种基于高斯变异的量子粒子群算法。该算法使粒子同时具有良好的全局搜索能力以及快速收敛能力。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;高斯;变异;全局搜索;收敛速度

中图分类号 TP18 **文献标识码** A

An Improved Quantum-behaved Particle Swarm Optimization Based on Gaussian Mutation

SHI Yong-sheng^{1,2}, CHEN Jia-qi¹

(1. Department of Computer Technology, University of Shanghai for Science and Technology, Shanghai 200093;

2. Department of Computer, Jiangsu College of Information Technology, Wuxi 214101, China)

Abstract Comparing with the other optimization algorithms, Particle swarm optimization (PSO) has comparable or even superior search performance for many optimization problems with faster and more stable convergence rates. But it can't guarantee to find the global optima in the search space especially in multimodal function. For conquering the shortcoming of particle swarm optimization, a novel quantum-behaved particle swarm optimization based Gaussian mutation (GQPSO) is introduced in this paper. The improved algorithm not only has more powerful global searching ability but also does 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 optimization; Gaussian mutation; global searching; convergence rate

由 Eberhart 和 Kennedy 等于 1995^[1]年提出的粒子群优化算法(Particle swarm optimization, 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,可以用于解决大量非线性、不可微和多峰值的复杂问题优化。目前在电路及滤波器设计、神经网络训练、机器人控制^[2-4]等领域均取得了非常好的效果。

由于粒子群算法具有收敛速度快、计算简单、易于实现等优点,因此,当它提出以后就引起了进化计算领域学者们的广泛关注。但由于粒子群算法在对多模函数求解时出现了早熟收敛的现象,许多学者对它进行

了改进。文献[5]中指出具有惯性因子的粒子群算法不仅有效的提高了原始 PSO 算法的收敛速度,而且在收敛精度上也优于原始 PSO。因此现在对粒子群算法的研究大都是基于这个版本,在本文中我们称作基本 PSO 算法。量子-粒子群算法^[6](QPSO)相对于标准粒子群算法来说,由于粒子随时具有跳出当前局部点的可能性,因此算法具有了更强的全局搜索能力。但 QPSO 在迭代后期由于粒子的聚集性,算法依然存在着早熟收敛的趋势。

本文首先针对 QPSO 算法的缺点,提出了基于高斯变异的量子粒子群算法(GQPSO)。该策略针对群体历史最优解能够有效地引导粒子群前进的方向,对群体历史最优解有效地运用了高斯变异因子。由于该策

收稿日期:2010-09-17

基金项目:中国博士后科学基金(20090460323)

作者简介:石永生(1970-),男,江苏无锡人,讲师,硕士,主要研究方向为人工智能、图像分割;陈家琪(1957-),男,吉林长春人,教授,博导,主要研究方向为人工智能、计算机网络。

略在保证粒子有效的进行局部搜索的前提下,同时加大了允许粒子对整个搜索空间进行搜索的可能性,从而保证改进以后的量子粒子群算法能够更为快速有效的进行全局搜索。

1 基本粒子群算法介绍

全局优化问题(p)的多个可行解的一个集合称为一个种群(swarm)。种群中的每个元素(可行解)称为一个粒子(particle),微粒的个数称为种群规模(size)。用n维向量 X_i 来表示第i个粒子的位置,用 V_{id} 来表示第i个微粒第d维的速度。粒子在迭代过程中,它自身经历的最佳位置为 $P_{pi}=(P_{pi1}, P_{pi2}, \dots, P_{pin})^T$,微粒群所经历过的最佳位置用 P_g 表示。因此,微粒在每一次迭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:

$$v_{id}(t+1)=wv_{id}(t)+c_1rand_1*(p_{pid}-x_{id}(t))+c_2rand_2*(p_{gid}-x_{id}(t)) \quad (1)$$

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

其中 $i=1, 2, \dots, m$ $d=1, 2, \dots, n$ w 称为惯性因子。根据文献[5]设置为大小随着迭代次数的减少由0.9下降到0.4; $rand_1$ 和 $rand_2$ 是服从 $U(0, 1)$ 分布的随机数,学习因子 C_1 和 C_2 为非负常数, $C_1=C_2=2$, $V_{id}=[-V_{max}, V_{max}]$, V_{max} 是由用户设定的速度上限。

2 基于高斯变异的 QPSO 算法

文献[6]从量子力学的角度出发提出了一种新的 PSO 算法模型。该模型以 DELTA 势阱为基础,提出了量子粒子群算法。通过蒙特卡罗随机模拟的方式得到粒子的位置方程为:

$$X(t)=P \pm \frac{L}{2} \ln(1/u) \quad (3)$$

其中 u 为 $[0, 1]$ 范围内变化的随机数, L 被定义为

$$L(t+1)=2*\beta * |mbest-x(t)| \quad (4)$$

其中

$$mbest = \sum_{i=1}^M P_i / M \\ = \left(\sum_{i=1}^M P_{i1} / M, \sum_{i=1}^M P_{i2} / M, \dots, \sum_{i=1}^M P_{iD} / M \right) \quad (5)$$

$$P_d = \varphi * p_{best} + (1 - \varphi) p_{gbest} \quad (6)$$

式中 β 称为收缩扩张因子。第 T 次迭代时取为 $\beta = 0.5 * (MAXTIME - T) / MAXTIME$, M 为粒子的数目。 p_i 为第 i 个粒子的历史最优解,可以由(6)式求得, φ 为 $[0, 1]$ 范围内变化的随机数。

根据(4)式可以看出,在 QPSO 算法中,只有一个参数 β 决定粒子的位置,同标准粒子群算法相比较更少, QPSO 算法更易于控制。另外由于 QPSO 算法中的粒子能够以一定的概率出现在整个搜索空间中,甚至是一个远离收敛点 P 的点,因此相对于标准粒子群算法来说具有着更强的全局搜索能力,图 1 为高斯概率密度分布图。

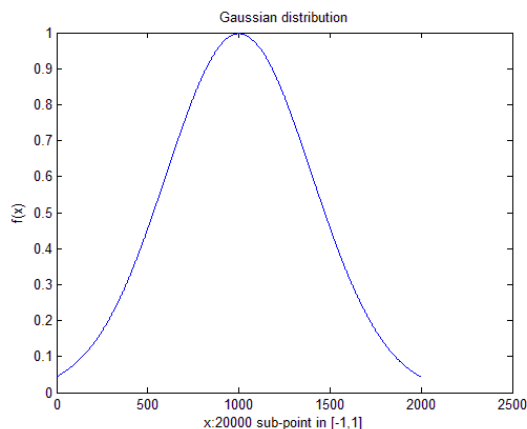


图 1 高斯概率密度分布图

基于高斯变异因子的粒子群算法主要是借鉴遗传算法中的变异因子来改善粒子群算法的全局搜索能力。以原点为中心的 Gaussian:

$$f(v) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{v^2}{2\sigma^2}\right], -\infty < v < \infty$$

其中为了覆盖整个解空间, Gaussian 的参数分别设置为 $\sigma=0.16$ 。

$$gbest1 = gbest + \text{cauchy}(t) * x_{max} \quad (7)$$

其中 t 是在 $[-1, 1]$ 内的随机数。算法迭代过程描述为:

Step1: 设置最大进化代数, 随机初始化每个粒子的位置, 初始化 p_p, p_g 。

Step2: repeat:

Step3: for each particle $j \in [1 \cdots D]$

Step4: 利用(4) (5)式计算每个粒子的适应度值, 并移动粒子到新的位置上。

if $f(X_i) < f(P_i)$

$P_i = X_i$

end

if $f(P_i) < f(P_g)$

$P_g = P_i$

end

end for

Step5: for $i \leq 5$

对 $gbest$ 使用(7)式进行变异并比较两者结果获得最优值。

end for

Step6: until 结束条件成立

3 对比试验

3.1 实验设计

本文设计了 4 类测试实验以分析提出的 GQPSO 的总体性能：

- (1)PSO 优化试验；
- (2)GPSO 优化试验 ,参数设置参考文献[7]；
- (3)QPSO[6]；

(4)GQPSO 同时引入 6 个 Benchmark 优化问题来进行优化分析。根据文献[8]~[9]对测试函数和测试标准的规定 ,函数形式以及搜索范围的设置见表 1。

表 1 BENCHMARK 函数

| code | Formula | Rang | V _{max} |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|----------------|------------------|
| f ₁ | $\sum_{i=1}^n x_i^2$ | [- 100 ,100] | 100 |
| f ₂ | $\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | [- 30 ,30] | 10 |
| f ₃ | $\sum_{i=1}^n i(x_i)^2$ | [- 100 ,100] | 100 |
| f ₄ | $\sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$ | [- 5.12 ,5.12] | 10 |
| f ₅ | $-20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^n \cos2\pi x_i\right] + 20 + e$ | [- 32 ,32] | 32 |
| f ₆ | $0.5 + \frac{\sin(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}$ | [- 100 ,100] | 100 |

本文测试的粒子群数目设置为 20。基本 PSO 中使用的 v_{max} 设置为 x_{max}。另外针对 Matlab 语言的特点 ,使用 rand(‘state’ ,sum(i*30)) 语句以保证每种算法运行时对应的 rand 函数种子相同（其中 i 表示在算法中的当前运行次数）,每一种算法独立运行 50 次。

测试函数中 f₁~f₅ 是单模函数 ,对于单模函数来说 ,算法的收敛速度比实验结果更值得关注 ,因此本文在列出比较结果的同时也列出比较函数在单模函数上的收敛比较图 ,对于 f₄~f₅ 这种高维的多模函数 ,实验结果的精度比收敛速度更值得关注 ,因此在本文中列出了各种比较算法解的平均值和标准方差。对于以上两种函数本文分别测试各种比较算法在 10 ,20 以及 30 维下的运行情况。f₆ 是具有较少局部最优点的低维函数 ,用以测试比较算法在低维函数上的运行情况。所有测试函数的理论最优解都为 0。

3.2 实验结果及分析

对于单模函数来说 ,由于 GQPSO 针对 gbest 使用了 Gaussian 变异因子 ,因此能够在当前历史最优解附近进行有效的搜索。根据表 2 的结果可以看出 ,在单模函数上相对于其他比较函数来说 ,GQPSO 都获得了最优的成绩。另外 ,图 2 的结果论证了该算法在收敛速度上都优于比较函数。对于复杂的多模函数 ,由于 GQPSO 相对于其他比较算法来说粒子具有更强的跳出局部最小点的能力 ,同时由于 gbest 能够提供给粒子群更为有效的进化方向 ,因此相对于其他算法来说 ,GQPSO 更为彻底的搜索了解空间。根据实验结果无论是全局搜索能力还是稳定性 ,GQPSO 都获得了最好的成绩。对于低维函数的优化能力参考表 3 的结果 ,GQPSO 依然获得了最好的结果。

表 2 GQPSO 及比较算法在单模函数的测试结果

| Alg. | f ₁ | | | f ₂ | | | f ₃ | | |
|-------|--------------------------|--------------------------|-------------------------|-------------------|--------------------|-------------------|--------------------------|--------------------------|-------------------------|
| | 10 | 20 | 30 | 10 | 20 | 30 | 10 | 20 | 30 |
| | 1000 | 2000 | 3000 | 1000 | 2000 | 3000 | 1000 | 2000 | 3000 |
| PSO | 1.67e- 20 (8.1e- 20) | 1.8e- 16 (5.2e- 16) | 1.4e- 13 (2.7e- 13) | 203.45 (405.7) | 132.05 (175.63) | 5786.3 (18391) | 8.17e- 20 (2.6e- 19) | 3.9e- 16 (1.4e- 15) | 1.35e- 12 (2.6e- 12) |
| GPSO | 6.4e- 21 (1.2e- 20) | 3.8e- 16 (1.1e- 15) | 6.5e- 14 (1.1e- 13) | 62.5 (224.68) | 99.922 (109.1) | 0.171 0.919 | 2.1e- 20 (6.5e- 20) | 2.98e- 16 (7.9e- 16) | 7.68e- 13 (1.7e- 12) |
| QPSO | 2.82e- 44 (7.63e- 44) | 1.35e- 30 (1.76e- 29) | 9.74e- 23 (2.2e- 22) | 6.456 (6.78) | 85.49 (81.3) | 62.94 (79.1) | 2.91e- 33 (1.26e- 33) | 6.49e- 18 (2.8e- 17) | 2.45e- 13 (9.5e- 13) |
| GQPSO | 1.03e- 45 (4.15e- 45) | 4.08e- 32 (2.6e- 31) | 2.86e- 24 (7.8e- 24) | 0.4751 (0.42) | 1.34 (4.12) | 0.48 (0.69) | 1.21e- 43 (2.82e- 43) | 9.68e- 20 (3.95e- 19) | 4.32e- 16 (4.1e- 16) |

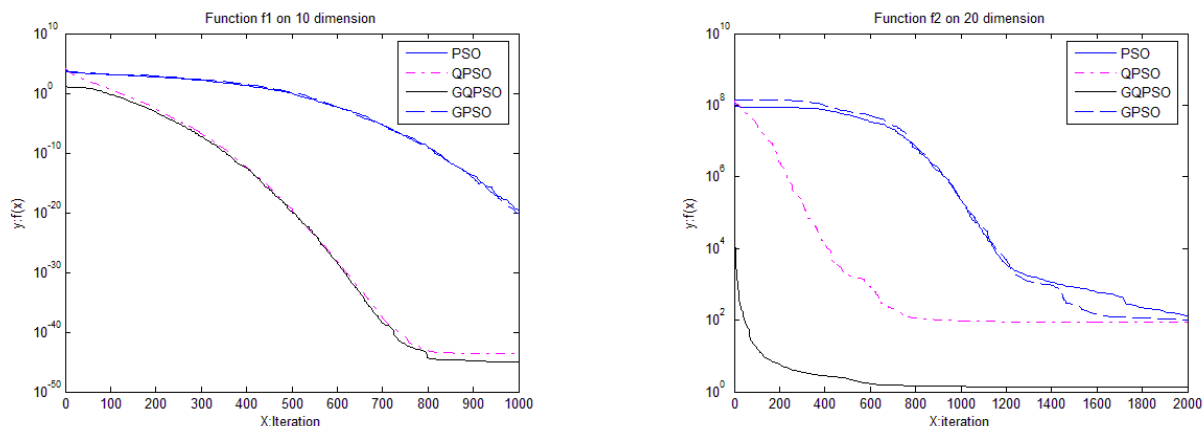
图 2 f_1 f_2 函数迭代比较图

表 3 WPSO 及比较算法在多模函数的测试结果

| Alg. | f_4 | | | f_5 | | | f_6 |
|-------|------------------|-----------------|-----------------|-----------------------|-----------------------|-----------------------|--------------------|
| | 10 | 20 | 30 | 10 | 20 | 30 | 2 |
| | 1000 | 2000 | 3000 | 1000 | 2000 | 3000 | 1000 |
| PSO | 5.936 (2.4) | 20.73 (4.4) | 43.6 (10.03) | 8.7e-11 (2.8e-10) | 9.3e-9 (2.5e-8) | 2.2e-7 (3.7e-7) | 0.0029 (0.004) |
| GPSO | 5.7714 (2.73) | 20.43 (7.26) | 43.8 (9.6) | 8.11e-11 (2.7e-10) | 1.8e-8 (5.3e-8) | 2.4e-7 (3.86e-7) | 0.0032 (0.005) |
| QPSO | 4.4 (3.25) | 13.86 (5.94) | 25.19 (5.93) | 3.37e-15 (1.4e-15) | 5.94e-14 (7.3e-14) | 4.14e-12 (7.5e-12) | 0.0032 (0.005) |
| GQPSO | 1.42 (0.494) | 10.12 (3.66) | 20.48 (2.94) | 3.19e-15 (1.2e-15) | 1.93e-14 (1.5e-14) | 9.37e-13 (1.7e-12) | 0.0004 (0.0013) |

4 结论

针对粒子群算法无法保证搜索到全局最优的缺点和搜索精度低的问题,提出了一种基于 Gaussian 变异的量子粒子群算法(GQPSO)。典型函数优化的仿真结果表明,该算法不仅可有效的避免标准 PSO 算法的早熟收敛,而且具有寻优能力强、搜索精度高、稳定性好等优点。同比其他使用变异因子的粒子群算法,GQPSO 表现出极强的全局搜索能力和快速的收敛速度,接下来该算法的应用和推广将是研究的重点。

参考文献:

- [1] J. Kennedy and R. C. Eberhart. Particle swarm optimization[C], In Proc IEEE Int. Conf Neural Networks, IV, Perth: IEEE Press, 1995.1942- 1948.
- [2] Jehad I. Ababneh, Mohammad H. Bataineh. Linear phase FIR filter design using particle swarm optimization and genetic algorithms [C]. Digital Signal Processing, 2008, vol. 18 (4), pp. 657- 668.
- [3] 张长胜, 孙吉贵, 欧阳丹彤, 等.求解车间调度问题的自适应混合粒子群算法[J].计算机学报, 2009, 32(11): 2137- 2146.
- [4] Saska, M., Macas, M., Preucil, L., Lhotska, L.. Robot Path Planning using particle swarm optimization of Ferguson Splines [C]. IEEE Con. Emerging Tech. and Factory Auto., 2006, vol. 20 (22), pp. 833- 839.
- [5] Daniel Bratton, James Kennedy. Defining a standard for Particle Swarm Optimization [A]. Proceedings of IEEE Swarm Intelligence Symposium (SIS), Honolulu, USA. 2007: 120- 127.
- [6] SUN J, XU WB. Particle Swarm Optimization with Particle Having Quantum Behavior [A]. IEEE Congress .Evolutionary Computation. 2004.325- 331.
- [7] Andrews. P. S. An Investigation into Mutation Operators for Particle Swarm Optimization [C]. IEEE Cong. On Evolutionary Computation, pp. 1044- 1051, Sep. 2006.
- [8] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky. A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems, Journal of Global Optimization [J]. pp. 635- 672, Nov. 2005.
- [9] Rui Mendes. Population Topologies and Their Influence in Particle Swarm Performance [D]. Escola de Engenharia, University of Minho, 2004.