

一种带逆反的粒子群算法^{*})

张晓明¹ 王儒敬^{1,2}

(中国科学院合肥智能机械研究所 合肥 230031)¹ (中国科学技术大学 合肥 230026)²

摘要 粒子群算法(PSO)是一种典型的基于群体智能的优化算法,但其在速度较小时,容易陷入局部最优解;本文提出一种带逆反的粒子群算法(PSORTP),并对其全局收敛性进行了理论分析,证明该算法能够以概率 1 收敛于全局最优解,最后以典型的函数优化问题的仿真实验及与经典方法的 PSO 的对比,验证了 PSORTP 的有效性。

关键词 群集智能,粒子群算法,逆向思维粒子,全局最优性

Particle Swarm Optimization Algorithm with Reverse Thinking Particles

ZHANG Xiao-Ming¹ WANG Ru-Jing^{1,2}

(Institute of Intelligent Machines (IIM) of Chinese Academy of Science, Hefei 230031)¹

(University of Science and technology of China, Hefei 230031)²

Abstract The particle swarm optimization algorithm is a kind of intelligent optimization algorithm. This algorithm is prone to be fettered by the local optimization solution when the particle's velocity is small. This paper presents a novel particle swarm optimization algorithm named particle swarm optimization with reverse thinking particles which is guaranteed to converge to the global optimization solution with probability one. And we also make the global convergence analysis. Finally, two function optimizations are simulated to show that the PSORTP is better and more efficient than the PSO with inertia weights.

Keywords Swarm intelligence, Particle swarm optimization, Reverse thinking particle, Global optimality

1 引言

通过对生物群体的观察和研究发现,生物群体中个体间的合作与竞争等复杂行为产生的群集智能^[1],往往能对某些优化问题提供高效的解决方法,由此出现了许多著名的群集算法,PSO 便是其中的代表性方法之一^[2]。

PSO 采用速度-位移搜索模型,每个粒子随机初始化表示为一个可能的解,解的优劣程度由适应度函数决定。速度向量 V_i 决定粒子在搜索空间单位迭代次数的位移,适应度函数根据优化的具体问题决定。粒子的位移向量为 X_i ,每迭代一次,粒子通过动态跟踪两个极值来更新其速度和位移,第一个极值是粒子从初始到到当前迭代次数搜索产生的最优解,称为个体极值,用向量 P_i 表示,第二个是粒子种群目前的最优解,称为全局极值,用向量 G 表示^[3]。粒子更新其速度和位移的公式为:

$$V_i = V_i + c_1 * rand() * (P_i - X_i) + c_2 * rand() * (G - X_i) \quad (1)$$

$$X_{i+1} = X_i + V_{i+1} \quad (2)$$

其中 $rand()$ 为均匀分布在(0,1)区间的随机数, c_1 和 c_2 是学习因子,一般取 2。粒子在解空间中不断跟踪个体极值与全局极值进行搜索,直到达到规定的迭代次数或满足规定的误差标准为止。粒子在空间中搜索的速度不能超过算法规定的最大速度 V_{max} 。 V_{max} 设得较大,粒子的全局搜索能力就较大;反之,粒子的全局搜索能力会较小,局部搜索能力则加强^[4]。

本文分析了经典的粒子群算法,并在此基础上提出了一种新的粒子群算法。该算法能够使粒子在 V_{max} 很小的情况

下,同样具有很强的全局搜索能力,并且保证以概率 1 找到全局最优解。最后,利用测试函数进行了数值试验,验证了该算法的有效性。

2 逆反粒子群算法(PSORTP)

2.1 PSO 算法的改进

在基本 PSO 算法中,粒子的飞行速度是一个很重要的因素,它取决于粒子当前的位置,个体最好位置以及群体最优位置,而且它还不能超过算法设置的最大速度 V_{max} 。设置较大的 V_{max} 就可能使粒子拥有较大的活动范围,有可能摆脱局部极值的束缚;但是这样却造成局部寻优能力下降,往往很难获得高精度的解。 V_{max} 设置得较小,则会使粒子易于陷入局部极值,无法找到全局最优解。

为此,我们提出逆向思维粒子的概念。逆向思维粒子的提出是受现实生活中的逆向思维心理效应的启发。逆向思维实质上是在顺向思维基础上的双向思维。这种思维方法倡导对思维对象不要仅仅限于某个单一的角度、方面去思考,要善于从同时存在、同时起作用的两个(多个)方面去思考,尤其是要善于、敢于从相反的方面多问几个为什么,或者对某种传统的,甚至是公认的看法,从反面提出质疑,“唱反调”,以达到突破和创新的目的。一个群体中,具有逆向思维的个体往往会在一个群体中表现突出,他们往往能够取得意想不到的成功,使得整个群体受益^[5]。

本文提出了 PSORTP 算法,加入逆向思维粒子,使得在粒子速度很小时,算法仍具有很强的全局搜索能力。逆向思维粒子的特点是不朝当前粒子的最优结果和当前全局最优结

^{*})基金项目:国家自然科学基金重点项目(69835010)。张晓明 硕士研究生,研究领域为智能计算。王儒敬 研究员,研究领域为智能计算,知识发现。

果的加权方向移动,而是朝相反的方向移动。它们随机在粒子群中选取,其位移公式为:

$$X'_i(t+1) = X_i(t) - c_3 * rand() * V_i(t) \quad (3)$$

其中 c_3 是逆向思维因子,在我们的实验中取 2; $rand()$ 为均匀分布在 $(0,1)$ 区间的随机数。

可以在图 1 中看到逆向思维粒子和正常粒子轨迹的不同。

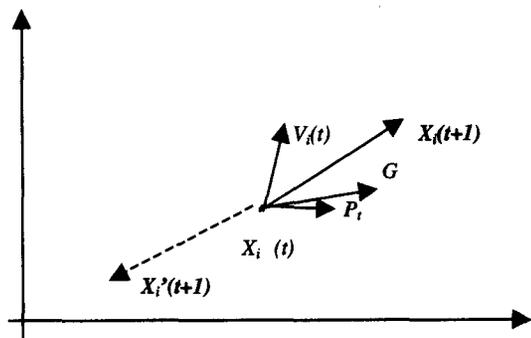


图 1 逆向思维粒子($X'_i(t+1)$)与正常粒子($X_i(t+1)$)运动轨迹对比图

算法执行时,正常粒子按照基本 PSO 算法运行,逆向思维粒子则按照上述运行。只要 G 是正常粒子获得的,逆向思维粒子就始终采取逆向思维行为;当某个逆向思维粒子找到优于 G 的值时,该粒子就更改为正常粒子,然后从正常粒子群中随机选取一个替代该粒子。如此进行寻优,直到达到规定的循环次数或最小误差数。PSORTP 算法的流程图如下。

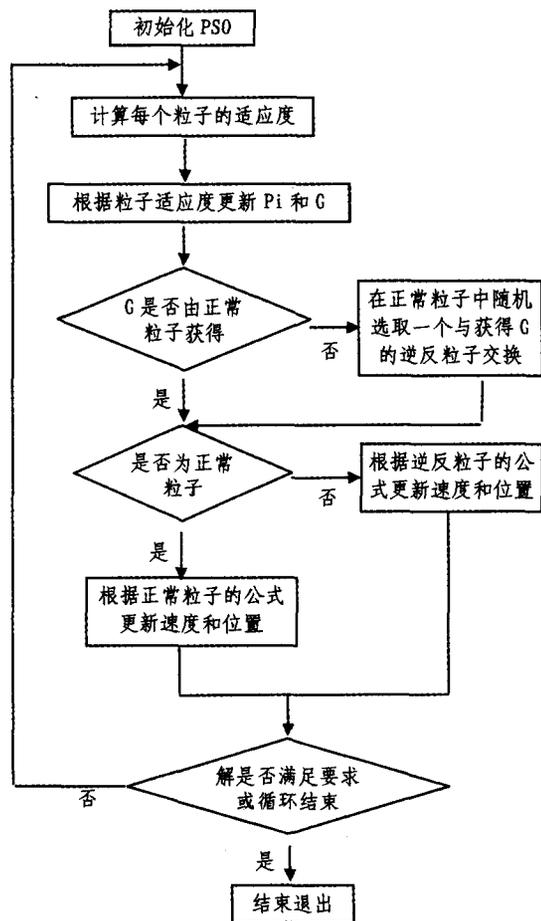


图 2 PSORTP 算法流程图

PSORTP 算法在运行过程中总有粒子朝着和大家认定的最优方向相反的方向进行搜索,由于逆向思维粒子的选择是随机的,因此其运动的方向也是随机的,这就保持了群体的

活力,增强了全局寻优的能力。

2.2 PSORTP 算法的全局最优性分析

PSORTP 算法是一种随机优化算法,关于随机优化算法以概率 1 收敛于全局最优解的条件, Solis 和 Wets 对其进行了证明^[6-8],其主要结论如下:

假设 1 若 $f(D(z, \zeta)) \leq f(z)$, $\zeta \in S$, 则 $f(D(z, \zeta)) \leq f(z)$ 。其中, D 为产生问题解的函数, ζ 为从概率空间 (R^n, B, μ_k) 产生的随机向量, f 为目标函数, S 为 R^n 的子集, 表示问题的约束空间。 μ_k 为 B 上的概率度量, B 为 R^n 子集的 σ 域。

假设 2 若对 S 的任意 Borel 子集 A , 有 $\nu(A) > 0$, 则

$$\prod_{k=0}^{\infty} (1 - \mu_k(A)) = 0,$$

其中, $\nu(A)$ 为子集 A 的 n 维闭包, $\mu_k(A)$ 为由 μ_k 产生 A 的概率。

定理 设 f 为一可测函数, S 为 R^n 的一可测子集, $\{z_k\}_{k=0}^{\infty}$ 为随机算法产生的解序列, 则当满足假设 1 和假设 2 时, 有

$$\lim_{k \rightarrow +\infty} P[z_k \in R_c] = 1,$$

其中, R_c 为全局最优点的集合。

由上述结论可知: 对于 PSORTP 算法, 只要能够满足假设 1 与假设 2, 就可以保证以概率 1 收敛于全局最优解。下面是我们对 PSORTP 的分析:

PSORTP 返回的结果是第 t 代的最优位置 G , $X(t)$ 为第 t 代的某个粒子的位置, 因此, 我们令假设 1 中的函数 D 为:

$$D(G, X(t)) = \begin{cases} G, & f(G) \leq f(X(t)), \\ X(t), & f(G) > f(X(t)), \end{cases} \quad (4)$$

可以看出, 假设 1 是满足的。

对于假设 2, 只须证明规模为 S 的粒子群的样本空间的并集包含 S , 即

$$\bigcup_{i=1}^j M_{i,t} \supseteq S,$$

其中, $M_{i,t}$ 是 t 代时粒子 i 的样本空间的支撑集, 所以, 也就是要证明粒子群的样本空间的支撑集的并集必须包含 S 。

以下为针对于此的分析: 在 PSORTP 算法中, 对于正常粒子, 设其支撑集的并集为 A ; 对于逆向思维粒子, 设其支撑集的并集为 B 。由于正常粒子与逆向思维粒子的活动方式正好相反, 但又同一问题空间中, 而且逆向思维粒子的选择是随机的, 同时当最优位置出现在逆向思维粒子中时, 还会有粒子的交换现象, 因此存在自然数 N , 使得代数 $t > N$ 时,

$$A \cup B \supseteq S.$$

定义 S 的 Borel 子集 $C = M_{i,t}$, $M_{i,t}$ 为第 i 个粒子在第 t 代时的样本空间的支撑集, 则有 $\nu(C) > 0$, $\mu_k[C] = \sum_{i=0}^k \mu_{i,t}[C] = 1$, 故

$$\prod_{t=0}^{\infty} (1 - \mu_t[C]) = 0,$$

所以假设 2 成立。

综上所述, 由定理可得:

$$\lim_{t \rightarrow +\infty} P[z_t \in R_c] = 1,$$

即 PSORTP 算法以概率 1 收敛于全局最优解。

3 实验与分析 (Experiments and analyses)

实验程序的编制采用 java 语言, 利用 JCreator 开发平台。实验函数采用 Rosenbrock, Rastrigin 和 Shaffer's F6 三个标准测试函数。

函数 Rosenbrock, Rastrigin 和 Shaffer's F6 的最小值求

解问题都是 GA-难度大的问题,用它们来验证和比较算法的性能是比较合适的。

3.1 函数介绍

Rosenbrock 函数的一般形式为:

$$\min f(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], x_i \in [-10, 10] \quad (5)$$

该函数具有一个全局极小点: $A=(1,1,1,\dots,1)$, 最小值 $f(A)=0$ 。

其形状如图 2(左)所示

Rastrigin 函数的一般形式为:

$$\min f(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10], x_i \in [-5.12, 5.12]$$

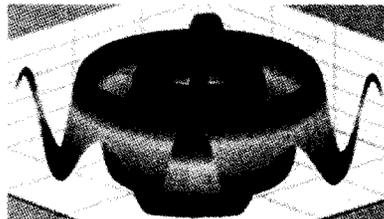
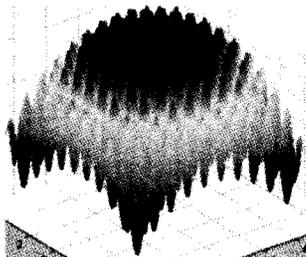
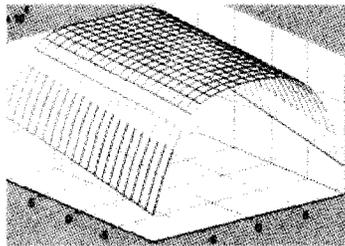


图 3 函数图像

12] (6)

该函数具有一个全局极小点: $A=(1,1,1,\dots,1)$, 最小值 $f(A)=0$ 。

其形状如图 2(中)所示。

Shaffer's F6 函数的一般形式为:

$$\min f(x,y) = 0.5 + \frac{(\sin \sqrt{x^2+y^2}) - 0.5}{(1.0 + 0.001(x^2+y^2))^2}, x_i \in [-100, 100] \quad (7)$$

该函数具有一个全局极小点: $A=(0,0)$, 最小值 $f(A)=0$ 。

其形状如图 2(右)所示。

3.2 实验结果

实验结果如表 1 所示。

表 1 实验结果

函数名	方法	粒子个数	P_{max}	逆反比例	结果
Rosenbrock	PSO	20	10	0	32.978
Rosenbrock	PSORTP	20	10	10%	30.216
Rosenbrock	PSORTP	20	10	30%	30.182
Rosenbrock	PSORTP	20	10	50%	30.149
Rosenbrock	PSO	100	10	0	28.871
Rosenbrock	PSORTP	100	10	10%	27.493
Rosenbrock	PSORTP	100	10	30%	26.839
Rosenbrock	PSORTP	100	10	50%	27.611
Rastrigin	PSO	20	5.12	0	37.264
Rastrigin	PSORTP	20	5.12	10%	37.087
Rastrigin	PSORTP	20	5.12	30%	37.202
Rastrigin	PSORTP	20	5.12	50%	35.568
Rastrigin	PSO	100	5.12	0	32.472
Rastrigin	PSORTP	100	5.12	10%	30.947
Rastrigin	PSORTP	100	5.12	30%	23.456
Rastrigin	PSORTP	100	5.12	50%	28.753
Shaffer's F6	PSO	20	100	0	0.000323
Shaffer's F6	PSORTP	20	100	10%	0.000142
Shaffer's F6	PSORTP	20	100	30%	0.000227
Shaffer's F6	PSORTP	20	100	50%	0.000259
Shaffer's F6	PSO	100	100	0	0.00012
Shaffer's F6	PSORTP	100	100	10%	0.0
Shaffer's F6	PSORTP	100	100	30%	0.0
Shaffer's F6	PSORTP	100	100	50%	0.0

说明:

1. PSO:经典的 PSO 算法;
2. 每种方法运行的循环次数为 2000;
3. Rosenbrock 函数和 Rastrigin 函数都是 30 维的,是 2

维的;

4. 针对 Rosenbrock 函数,粒子的最大速度为 0.5,粒子的初始位置域为 $[-10,10]$;
5. 针对 Rastrigin 函数,粒子的最大速度为 0.5,粒子的初

始位置域为[-5.12, 5.12];

6. 针对 Shaffer's F6 函数,粒子的最大速度为 2.0,粒子的初始位置域为[-100,100];

7. 逆反比例:指逆向思维粒子数在总群体中占的比例;

8. 结果的得出是针对每个函数每种算法每种参数各运行 50 次,对它们取平均值。

表 2 Maurice Clerc 和 James Kennedy 的最优实验结果

函数名	最优结果
Rosenbrock	39.118488
Rastrigin	46.4689
Shaffer's F6	0.000247

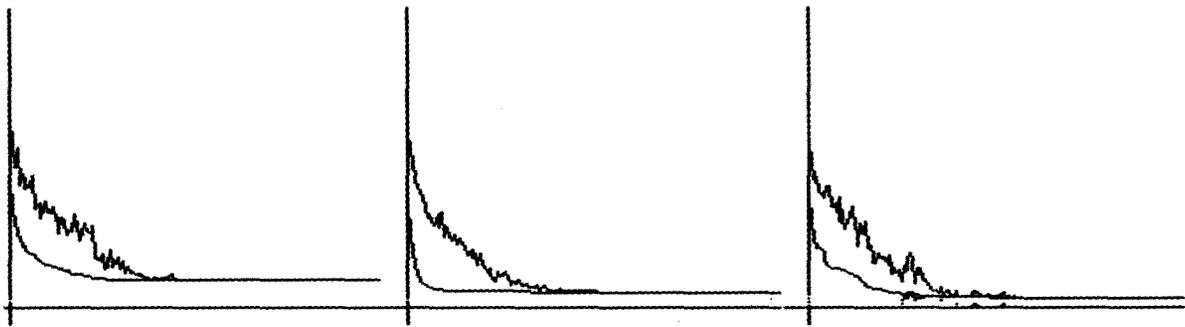


图 4 运行结果曲线示意

3.3 结果分析

由上面的结果可以看出 PSORTP 算法的性能好于经典的 PSO 算法,同时也明显优于 Maurice Clerc 和 James Kennedy 的最优实验结果。当粒子数较少时(20 个),PSORTP 与经典方法的 PSO 算法的性能相差不多,尤其是当逆反比例很小时,最优值通常是由具有 50%逆反比例的 PSORTP 获得。当粒子数较大时(100 个),PSORTP 的优势就较为明显了,尤其是逆反比例为 30%时,对于三个函数的优化都取得了很好的效果,结果要比经典方法的 PSO 算法的结果好许多。

结束语 本文对 PSO 算法进行了改进,提出了一种能够保证全局收敛的 PSO 算法—PSORTP 算法。该算法引入逆向思维粒子,与传统 PSO 算法相比,提高了算法全局寻优的能力;在理论上,我们也对它做了分析,证明它是以概率 1 收敛到全局最优解的。最后通过实验和与其它方法的对比,表明了本方法的有效性。

在逆向思维粒子的选取上,本文采用的是固定比例选取的方式,这样就不能保证在任何情况下都能选择最佳比例,如果能让算法自动确定逆向思维粒子的最佳比例,算法的性能将会变得更好。因此,下一步,我们将会把研究的重点放在对

Maurice Clerc 和 James Kennedy 在文[9]中的实验中也用到了上面提到的函数,在他们的实验中,粒子的群体规模是 20。他们得出的最好结果如表 2 所示。

我们选取计算 Rastrigin 最小值的运行结果曲线,作为运算结果示意图的比较(图 4 所示),其中最左边的是经典方法的 PSO,运算结果为 32.1132,中间的是逆反比例为 10%的 PSORTP,运算结果为 30.8226,最右边的是逆反比例为 30%的 PSORTP,运算结果为 23.1431(图中上面的曲线是群体最优值 G,下面的曲线是粒子的平均适应度值(函数值)。运算参数:粒子数:100;循环次数:2000;初始值 $x_i \in [-5.12, 5.12]$; $V_{max} : 0.5$)。

逆向思维粒子比例的自适应确定上,进一步提高算法的性能。

参考文献

- 1 Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behaviour [J]. Nature, 2000, 406(6): 39~42
- 2 Kennedy J, Eberhart R C. Particle swarm optimisation [A]. In: Proc. of the IEEE International Conference on Neural Networks [C]. Piscataway, NJ, USA: IEEE, 1995. 1942~1948
- 3 李宁,付国江,等. 粒子群优化算法的发展与展望. 武汉理工大学学报, 27(2): 26~29
- 4 柯晶,钱积新,乔谊正. 一种改进粒子群优化算法[J]. 电路与系统学报, 2003, 8(5): 87~91
- 5 李沛君. 逆向思维在企业经营创新中的应用. 重庆商学院学报, 2002(1)
- 6 Cui Zhihua, Zeng Jianchao. A Guaranteed Global Convergence Particle Swarm Optimizer. RSTC' 2004 [C]. Sweden, 2004. 762~767
- 7 Solis F, Wets R. Minimization by random search techniques. Mathematics of Operations Research, 1998, 6(1): 19~30
- 8 http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-099Fall2003/594A2FDC-A9B1-4336-AFDC-E2298F3C0DC4/0/ses5_solis_wets.pdf
- 9 Clerc M, Kennedy J. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space[J]. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, 2002, 6(1): 58~73

(上接第 110 页)

- 9 Tzeng S F, Hwang M S, Yang C Y. An improvement of nonreputable threshold proxy signature scheme with known signers. Computer and Security, 2004, 23: 174~178
- 10 Tan Z W, Liu Z J. Cryptanalysis of threshold proxy signature scheme. MM Research Preprints, Dec. 2004(23): 226~233
- 11 Bellare M, Rogaway P. The exact security of digital signature—How to sign with RSA and Rabin. In: Proc. Advance in Cryptology- Eurocrypt'96, 1996, 399~416
- 12 卢建朱,陈火炎,沈建军. 高效的不可否认的门限代理签名新方案. 计算机工程与应用, 2005, 31: 14~17

附录

PSS(Probabilistic Signature Scheme)签名方案

PSS[k_0, k_1]是由公开参数 k_0, k_1 刻画的签名方案,满足 $k_0 + k_1 \leq k - 1$ 。其中, k 为 N_0 的特长, $k = 1024$, 下设 $k_0 = k_1$

= 128。(MD5 的输出为 128 比特)

算法中的 Hash 函数: $h': s: \{0, 1\}^* \rightarrow \{0, 1\}^{k_1}; g: \{0, 1\}^{k_0} \rightarrow \{0, 1\}^{k-k_1-1}; g_1: \text{输入 } b \in \{0, 1\}^{k_0}, \text{输出 } g(b) \text{ 的前 } k_0 \text{ 比特}; g_2: \text{输入 } b \in \{0, 1\}^{k_0}, \text{输出 } g(b) \text{ 的余下 } k-k_0-k_1-1 \text{ 比特}。$

签名时: $l \leftarrow \{0, 1\}^{k_0}; b \leftarrow h'(s \parallel l); l^* \leftarrow g_1(b) \oplus l; M^* \leftarrow g_2(b) \oplus s; m' \leftarrow 0 \parallel b \parallel l^* \parallel M^*。$

验证时: 将 m' 写成 $a \parallel b \parallel l^* \parallel M^*$ 的形式; $l \leftarrow l^* \oplus g_1(b); s \leftarrow M^* \oplus g_2(b)$ 判断 $h'(s \parallel l) = b, a = 0$; 若成立, 输出 s 。

注: 上述 PSS 消息处理方案由 n 个代理者在系统建立时产生, 可以作为一个整体用程序编入黑盒子, 并植入一个随机数产生器, 每签一次名, 随机数发生器产生一个随机数 l , 输入一个 s , 结果输出一个 m' 。此方案的安全性是基于 PSA 模的分解和安全 Hash 函数。