基于数论方法的微粒群算法*)

米利波[1] 谭立伟[2]
(重庆文理学院教育技术中心 重庆永川 402160)[1]
(重庆文理学院数学与计算机科学系 重庆永川 402168)[2]

摘 要 为了改进微粒群优化算法的结果,用数论网格法初始化微粒群的初始位置,对位于当前全局最优点的微粒重新初始化,最后用一种爬行算法求精微粒群寻优的结果。实验表明,改进后的算法能克服标准微粒群算法的困难,获得更好的结果。

关键词 数论网格法,微粒群算法,最优化

Particle Swarm Optimization Based on Number-theoretic Method

MI Li-Bo[1] TAN Li-Wei[2]
(Center of Education Technology, Chongqing University of Arts and Sciences, Yongchuan Chongqing 402160)[1]
(Department of Mathematics and Computer Science, Chongqing University of Arts and Sciences, Yongchuan Chongqing 402168)[2]

Abstract An improved particle swarm optimization algorithm is proposed. Number-theoretic method is used to initialize the particles' position. The particle locates on the current global minimum is re-initialized during the algorithm. In the end the result of PSO is fined by a method named creeping algorithm. The experiments manifest that the improved algorithm can get the better results.

Keywords Number-theoretic method, Particle swarm optimization, Optimizing

各种函数最优化方法的研究是目前研究的热点,新的优化算法被不断地提出并改良,微粒群算法(Particle Swarm Optimization, PSO)就是其中的杰出代表。该算法于 1995 年由 James Kennedy 和 Russell Eberhart 在 Frank Heppner 的生物群体模型的基础上提出[1]。由于该算法的良好表现,大量的科技工作者对该算法进行了该进,如 Shi 和 Eberhart 为 PSO 算法的速度项引入了惯性权重 ω[2,3],并进一步提出了模糊自适应 PSO 算法[4]。

最优化方法可以分为确定型算法和随机型算法,PSO 算法就是随机型算法的一种,确定型方法的代表有数论网格法等。钟良、钟守楠和章晓燕结合数论方法和遗传算法提出了基于数论的总体优化随机搜索算法,效果较好[5];刘飞等将数论网格法应用于极大似然估计,具有较高的精度和效率[6]。

PSO 算法应用中暴露出的一个重要的缺点是初始粒子群的数量和分布对最终结果有较大的影响,特别是对多峰函数寻优的情况[7]。本文通过应用数论方法来确定粒子群的分布,并在微粒群寻优结束后用类似于梯度下降的爬行算法改进最终结果,最后把改进后的 PSO 算法应用于一些多峰函数寻优实验,取得了较好的效果。

1 算法的基本原理

1.1 带收缩因子的 PSO 算法

不失一般性,不妨假设最优化问题如下:

$\min \sigma = f(X)$

s. t. $X \in S = \{X \mid g_j(X) \leqslant 0, j=1,2,...,m\}$

其中 $\sigma = f(X)$ 称为目标函数,$g_j(X)$ 称为约束函数,S 为约束域。下边的讨论中就把目标函数 $f(X)$ 作为适应度函数。

设

$X_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ 为微粒 i 的当前位置;

$V_i = (v_{i1}, v_{i2}, \cdots, v_{in})$ 为微粒 i 的当前飞行速度;

$P_i = (p_{i1}, p_{i2}, \cdots, p_{in})$ 为微粒 i 到目前为止所经历过的最好位置(即适应度最佳的位置);

$G = (g_1, g_2, \cdots, g_n)$ 为到目前为止,所有微粒所经历过的最好位置。

基本微粒群算法的进化方程为[8]:

$$v_{ij}(t+1) = \chi(v_{ij}(t) + c_1 r_{1j}(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(g_j - x_{ij}(t)))$$

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

其中,下标"i"表示微粒 i,"j"表示微粒的第 j 维,"t"表示第 t 代,$c1$ 和 $c2$ 为加速因子,其值用于控制当前微粒向自已所经历的最优点和全局当前最优点的飞行速度,χ 称为收缩因子,用于保证算法收敛。

1.2 数论网格法

数论方法(Number theoretic method, NTM)是数论与近似分析交叉的产物,其实质是在 s 维的单位立方体中找到一个均匀分布的点集,其均匀性可以用偏差的大小来度量。该随机点集常常称为数论网格(NT-net)[9]。

令 $(n; h_1, h_2, \cdots, h_s)$ 为一个整数矢量,满足 $1 \leqslant h_i < n$,且当 $i \neq j$ 时,$h_i \neq h_j$,$s < n$ 及最大公约数 $(n, h_i) = 1, i = 1, ..., s$。令

$qk_i \equiv kh_i \pmod n + 1$,

$x_{ki} = (2q_{ki} - 1)/2n$,其中,$k = 1, 2, \cdots, n, i = 1, 2, \cdots, s$。

则称集合 $P_n = \{(x_{k1}, x_{k2}, \cdots, x_{ks}), k = 1, 2, \cdots, n\}$ 为生成矢量 $(n; h_1, h_2, \cdots, h_s)$ 的格子点集,若 P_n 在所有可能的生成矢量

*)基金项目:重庆市教育委员会科学技术研究项目(KJ061203)、重庆市教育委员会科学技术研究项目(KJ061201)资助。米利波 硕士生,实验师,主要研究方向:计算机网络;谭立伟 硕士,讲师,主要研究方向:计算智能。

具有最小偏差，则称 P_n 为最优格子点集(Good lattice point set, GLP 集合)。其中生成矢量值的确定方法参考文[9]。

1.3 爬行求精算法

PSO 算法的一个困难之处在于在算法运行的后期，微粒会在全局最优点附近摆动，却不能达到该全局最优点，此时，用梯度下降法是可以获得该最优点的。但是梯度下降法有一个问题，就是需要对目标函数求偏导数以获得梯度向量，对于有的函数求导数并不是一件很容易的事情。考虑到此时微粒离将要收敛到的最优值点已经很接近了，因此采用了一种不需要求导数的爬行算法进行求精。

其思路是考虑微粒目前所在位置的四周，以一个极小的步长向能获得更优值的方向爬行，这又有两种选择：向最优值变化最快的方向靠近和向最优值更优的所有方向爬行。为了找到可以爬行的方向，需要向微粒当前位置的四周进行探测，若假设空间是 n 维，则需要探测的方向数目是 2^n，当维数非常多的时候，可能会产生较大的时间开销。

假设当前找到的全局最优值点是 $X_G=(x_{g1}, x_{g2}, \cdots, x_{gn})$，给定一个很小的爬行步长 $step$，对 G 周围的 2^n 个方向分别进行下述运算。对每一维使用下述公式之一获得点 $X=(x_1, x_2, \cdots, x_n)$：

$x_i=\mathrm{random}(x_{gi},\ x_{gi}+step)$

或者 $x_i=\mathrm{random}(x_{g_i}-step,\ x_{gi})$，

其中 $i=1,2,\cdots,n$，$\mathrm{random}(a,b)$ 表示在 a 与 b 之间取随机数。

若 $f(X_G)<f(X)$，则换下一个方向继续探测，若 $f(X_G)>f(X)$，则用 X 更新 X_G。重复这个过程，直到不能找到更好的点。

1.4 改进后的 PSO 算法

PSO 算法有几个困难的地方，其一是算法的精度可能会受到微粒数量和微粒分布的影响；其二是在算法运行的后期，微粒会在希望收缩到的极值点附近摆动，却难以达到该极值点。本文对第一个问题的解决方法是首先用数论网格法来确定微粒的分布，然后在进化过程中重新初始化位于全局最优值点的微粒；对第二个问题的解决方案是采用近似于梯度下降的爬行算法来使得算法最终收敛于全局最优点。

具体的计算步骤如下：

step 1：初始化生成矢量；

step 2：应用 GLP 集合的生成方法产生 n 个均匀分布的微粒 $X_i, i=1,\cdots,n$；

step 3：计算各微粒适应值 $f(X_i)$，初始化各微粒当前最优值点 P_i 和当前全局最优值点 G；

step 4：初始化各微粒初始速度以及速度方程参数；

step 5：用带收缩因子的 PSO 算法进行计算，获得新的微粒位置；

step 6：计算每个微粒的适应值 $f(X_i)$；

step 7：更新各微粒 P_i 和 G 值；

step 8：重新随机初始化位于当前全局最优点的微粒；

step 9：判断 G 未发生变化的迭代次数，若达到迭代次数阈值 $G_{\max}$，转 step 10；否则转 step 6；

step 10：以很小的步长探测 G 点四周，寻找适应值更优的点，若找到则更新 G；

step 11：重复进行 step 10 直到找不到更好点；

step 12：算法结束，$f(G)$ 就是所求的最优值。

2 实验结果分析

在配置 Intel Core 2 Extreme QX6600 2.4G Hz 的处理器和 1G 内存的微型机上，利用 Visual C++ 6.0 编程环境进行了仿真实验。为了更好地对比结果，论文选择了标准 PSO 算法和遗传算法进行实验，并将结果与改进后的 PSO 算法进行对比。

实验选择了 Goldstein-Price 函数、J. D. Schaffer 函数和一个四维函数进行计算和分析。

Goldstein-Price 函数为：

$$\min f_1(X)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+16x_1x_2+3x_2^2)]*[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$$

其中 $-2\leqslant x_1\leqslant 2, -2\leqslant x_2\leqslant 2$。

J. D. Schaffer 函数为：

$$\min f_2(X)=\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001(x_1^2+x_2^2)]^2}-0.5, -100\leqslant x_1, x_2\leqslant 100$$

四维函数为：

$$\min f_3(X)=(x_1-\frac{3}{11})^2+(x_2-\frac{6}{13})^2+(x_3-\frac{12}{23})^4+(x_4-\frac{8}{37})^6$$

s.t. $0\leqslant x_i\leqslant 1, i=1,2,3,4$

表 1 遗传算法、标准 PSO 和改进 PSO 寻优结果对比

	遗传算法	标准 PSO 算法	改进 PSO 算法
f_1	0	0	0
f_2	−2.064324442	−0.633690342	−1.0007002032
f_3	$6.3533098e^{-5}$	$7.4300233e^{-10}$	$1.2134534e^{-16}$

从表 1 中的实验结果可以看出，对于局部最优点较少的函数，三种算法的效果都比较理想，但对于局部最优点很多或者搜索空间维数较高的场合，基于数论方法的 PSO 算法有较大的优势。

结束语 作为群体智能的代表性方法之一，微粒群方法受到了广泛的重视，它易于实现，效果良好。本文把数论网格法和梯度下降法与微粒群算法结合起来，试图克服微粒群分布对寻优结果的影响，解决算法后期微粒在最优值附近摆动的问题。实验证明，改进后的算法达到了预期的目的。

参考文献

1 Kennedy J, Eberhart R C. Particle Swarm Optimization[C]. In: Proc. IEEE International Conference on Neural Networks. IV Piscataway, NJ: IEEE Service Center, 1995. 1942～1948

2 Shi Y, Eberhart R C. Particle Swarm Optimization: developments, applications and resources[C]. In: Proc. Congress on Evolutionary Computation 2001 NJ: Piscataway. IEEE Press, 2001. 81～86

3 Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. In: IEEE World Congress on Computational Intelligence, 1998. 69～73

4 Shi Y, Eberhart R C. Fuzzy Adaptive Particle Swarm Optimization [C]. In: Proc. Congress on Evolutionary Compution, 2001. 101～106

5 钟良，钟守楠，章晓燕. 基于数论的总体优化随机搜索算法[J]. 数学杂志，2006, 26(1): 75～82

6 Liu Fei, Dou Yi-fang, Zhang Wei-hua. 数论网格法在极大似然估计中的应用[J]. 系统仿真学报，2006, 18(9): 2534～2536

7 张晓清，张建科，方敏. 多峰搜索的动态微粒群算法[J]. 计算机应用，2005, 25(11): 2668～ 2670

8 Clerc M. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In: Proc. of CEC 1999, 1999, 1: 1951～1957

9 Fang K T, Wang Y. Number-theoretic Methods in Statistics [M]. Landon: Chapman & Hall, 1993. 111