

文章编号: 1001-0920(2004)12-1364-04

基于多元最优信息规划的微粒群优化算法

汪 镭, 康 琦, 吴启迪

(同济大学 电子与信息工程学院, 上海 200092)

摘 要: 在改进型微粒群优化算法的基础上, 考虑到运算初期避免陷入局部寻优的需要, 将多元最优信息分布状态知识引入对微粒群的总体运动规划, 构造了适于求解高维多峰函数寻优问题的微粒群算法。仿真结果显示, 该方法比改进型微粒群算法具有更好的总体寻优特性。

关键词: 多元最优规划; 微粒群优化算法; 高维多峰函数寻优

中图分类号: TP18 **文献标识码:** A

Multi-optimum information programming based particle swarm optimization algorithm

WANG Lei, KANG Qi, Wu Qi-di

(Electronics and Information Institute, Tongji University, Shanghai 200092, China Correspondent: WANG Lei, E-mail: wanglei@mail.tongji.edu.cn)

Abstract: The knowledge of multi-optimum distribution state is introduced into general programming of the particle swarm movement to avoid falling into local optimums at the original stage of the computation. The algorithm is improved based on the modified particle algorithm and used to optimize the multi-dimensional and multi-optimum function. Simulation results show that the general convergence character of the derived algorithm is better than the results derived based on the modified algorithm.

Key words: multi-optimum programming; particle swarm optimization algorithm; multi-dimensional and multi-modal function optimization

1 引 言

微粒群优化(Particle Swarm Optimization)算法是由 Kennedy 等^[1]于 1995 年开发的一种演化计算技术, 其基本思想来源于对鸟群简化模型的研究及行为模拟。在这种优化算法中, 寻优空间中的每个微粒没有质量和体积, 其运动速度受到自身和群体的历史运动状态信息的影响, 以自身和群体的历史最优位置来对微粒当前的运动方向和运动速度加以影响, 较好地协调了微粒本身和群体运动之间的关

系

自该算法提出后, 由于其概念简明、实现方便, 在短期内迅速得到了国际演化计算研究领域的认可^[2,3], 并在约束优化、电能控制^[4]及病理分析^[5]等领域得到了有效应用。同时, 算法本身也在原有基础上, 考虑到其自适应性^[6]、参数优化^[7]、邻域特征^[8,9]及与其他智能优化算法的结合^[10,11]等因素, 作了许多改进和实验仿真^[12~18], 得到了较为丰富和系统的结果。

收稿日期: 2004-02-16; 修回日期: 2004-05-17

基金项目: 国家自然科学基金资助项目(70271035; 60104004); 国家 973 子项目(2002CB312202); 上海市启明星计划项目(03QG14053)。

作者简介: 汪镭(1970—), 男, 江苏无锡人, 副教授, 从事智能自动化系统等研究; 吴启迪(1947—), 女, 浙江永嘉人, 教育部副部长, 教授, 博士生导师, 从事智能自动化理论与应用、系统工程、计算机集成制造系统等研究。

本文在改进型微粒群算法^[19]的基础上, 考虑到高维多峰函数寻优的需要, 将多元最优信息分布状态分析的思路引入微粒群的运动规划, 并以最优和次最优分布信息为例, 作了高维 Rosenbrock 函数求解初期的微粒群总体运动规划, 加速了微粒群在复杂寻优空间中的寻优进程, 保证了算法的总体收敛特征. 仿真结果表明, 该算法与传统算法相比其性能有了很大程度的提高.

2 传统及改进型微粒群算法描述

在传统型微粒群算法中, 每个微粒在 D 维空间中的坐标位置可表示为 $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$, 微粒 $i (i = 1 \sim N)$ 在第 $d (d = 1 \sim D)$ 维子空间中的飞行速度 v_{id} 按下式进行调整:

$$v_{id} = v_{id} + c_1 \text{rand}_1() (p_{id} - x_{id}) + c_2 \text{rand}_2() (p_{gd} - x_{id}); \quad (1a)$$

$$v_{id} = \begin{cases} v_{\max}, & v_{id} > v_{\max}; \\ -v_{\max}, & v_{id} < -v_{\max}. \end{cases} \quad (1b)$$

其中: p_{id} 是整个微粒群的历史最佳位置记录, 它与当前微粒的位置之差用于改变当前微粒向群体最优值运动的增量分量, 此增量还需进行一定程度的随机化 (运用 $\text{rand}_1()$ 随机发生器); p_{gd} 是当前微粒的历史最佳位置记录, 它与当前微粒的位置之差也用于该微粒的方向性随机运动设定 ($\text{rand}_2()$ 亦为随机发生器); c_1 和 c_2 为常数.

由上式可知, 微粒的运动速度增量与其历史飞行经验和群体飞行经验相关, 并受最大飞行速度的限制. 这种运动模式可用于各类寻优问题的求解.

在微粒群算法的改进模式中, 其飞行速度按下式决定:

$$v_{id} = w v_{id} + c_1 \text{rand}_1() (p_{id} - x_{id}) + c_2 \text{rand}_2() (p_{gd} - x_{id}); \quad (2a)$$

$$v_{id} = \begin{cases} (1 - (t/T)^k) v_{\max}, & v_{id} > (1 - (t/T)^k) v_{\max}; \\ - (1 - (t/T)^k) v_{\max}, & v_{id} < - (1 - (t/T)^k) v_{\max}. \end{cases} \quad (2b)$$

在飞行模式 (2) 中, 对上次飞行速度的记忆分量要乘以一个加权系数 w , 且 w 随运行时间的延续也要进行线性变化, 即

$$w = K_1 + (K_2 - K_1)t/T. \quad (3)$$

其中: T 为总循环次数; t 为当前计算所处的循环次数; K_1 和 K_2 为常数, 表明 w 变化的左右边界, 本文取 $K_1 = 0.4$, $K_2 = 0.9$, 这是许多文献中所采用的最佳参数值.

另外, 从式 (2) 还可看出, 其飞行速度的限幅值随着时间的进行而逐步缩小, 从而逐步限制了微粒群的飞行范围. 在本文计算中, K 取相关文献中所获的最优值 0.05.

3 基于多元最优信息规划的微粒群算法

在上述基本算法及改进型算法中, 微粒的运动考虑了群体和个体最优值对自身速度的影响, 有效地引导了微粒群体的运动, 因此在许多寻优问题求解中获得了有效的应用. 但这中间运动模式必然受微粒群初始分布特征的较大影响, 尤其是高维复杂多峰函数的求解, 在某些初始分布状态下, 这种运动模式可能导致微粒群运动受到初始分布中局部最优值的过度影响, 从而陷入局部最优.

为解决上述问题, 在微粒群算法寻优的初期, 可适当地多考虑一些整个群体的多元最优值分布特征. 除当前微粒的历史最优值外, 仅利用群体最优值这一参量来对微粒运动产生影响显然不够, 尤其是在高维多峰函数较大范围内的寻优问题求解更是如此. 这时, 可以引入 M 个群体最优值来对当前微粒的运动进行多元规划, 微粒的运动速度可由下式决定:

$$v_{id} = w v_{id} + \sum_{k=1}^M c_k \text{rand}_k() (p_{k,id} - x_{id}). \quad (4)$$

其中: $p_{k,id}$ 表示在整个群体中排列第 k 的最优值, c_k 和 $\text{rand}_k()$ 分别表示与其相匹配的加权系数和随机量. 在这种运动模式中, 虽然速度计算中增加了一定程度的多元最优值比较和规划求和分量, 但其避免陷入局部最优的能力却大大增强了. 当然, 如果微粒群的总体已进入最优值的较近邻域内, 这种计算模式反而可能导致收敛速度的减缓. 这时, 可采用传统型或传统改进型微粒群算法.

为进一步加快微粒群在寻优初期的总体收敛速度, 跳出局部最优, 还可根据寻优函数的总体分布特征等, 先将微粒每一维坐标的初始分布范围分为 N 个区间, 并依次给定序列号, 在此基础上得到微粒的运动速度.

$$v_{id} = w v_{id} + \sum_{k=1}^M c_k \text{rand}_k() (G_{k,id} - G_{id}) \text{abs}(S_{k,id} - S_{id}). \quad (5)$$

其中: $S_{k,id}$ 表示排列第 k 的最优值所处的第 d 维坐标区间号, S_{id} 表示当前微粒 i 的第 d 维坐标所处的区间号, $\text{abs}()$ 为绝对值函数, G_{id} 表示当前微粒所处区间重心坐标的第 d 维分量, $G_{k,id}$ 表示当前微粒所处循环中排列第 k 的最优值所处的第 d 维区间的重心.

坐标 这样,与整个微粒群中前 M 个最优值相距越远的微粒,其运动速度将越快,且其运动方向的规划完全取决于微粒所在区间和这 M 个最优值所处区间的重心位置,而这些计算在算法初始化期间便可完成,从而将大大降低实际循环中的计算量

4 高维多峰函数的寻优计算结果及仿真

仿真计算中,在函数寻优初期,采用了基于二元最优规划的步长增量计算模式,即方程(4)中 $M = 2$,方程中 $p_{1,id}$ 和 $p_{2,id}$ 即为整个群体当前的最优和次最优值

算法运行初期采用下式计算速度:

$$v_{id} = w v_{id} + \text{rand}_1() c_1 (G_{1,id} - G_{id}) \text{abs}(S_{1,id} - S_{id}) + \text{rand}_2() c_2 (G_{2,id} - G_{id}) \text{abs}(S_{2,id} - S_{id}). \quad (6)$$

本文对在同样循环数(1 000)下各算法在相同的运行次数(100)后能达到最优值设定邻域内的微粒占群体总量的平均比例作了比较

仿真是在 Windows XP 环境下进行的,程序用 Visual Basic 6 编写 仿真计算中所采用的寻优检验函数为一典型的高维多峰 Rosenbrock 函数,即

$$f(x) = \sum_{i=1}^n [100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2], \quad (7)$$

此函数的全局最优值 0 在 $x_i = 1$ 时取得

仿真计算中针对 10 维和 20 维 Rosenbrock 函数进行了寻优求解 其中:微粒群的初始分布邻域设定为 $x_i \in [-100, 100]$, v_{\max} 设为 100, $c_1 = 10, c_2 = 2, k_1 = 0.4, k_2 = 0.9$,群体规模为 64, x_i 的分布邻域均分为 10 个 当各微粒处于相同或相邻区域时,即当各微粒间距小于 40 时,采用传统改进算法

在各微粒处于设定邻域之外和之内时,所采取的 v_{id} 最大速度限制常数的比例取为 $S = 5$ 统计微粒群在各算法下在有限循环次数内收敛至 $[-15, +15]$ 范围中的微粒比例,所得结果如表 1 和表 2 所示

表 1 20 维 Rosenbrock 函数优化结果

算 法	收敛至要求范围内的微粒比例/%
传统改进算法	21.22
本文方法($c_1 = 10, c_2 = 2$)	55.75
本文方法($c_1 = 5, c_2 = 2$)	64.66

图 1~ 图 3 为 20 维 Rosenbrock 函数在不同算法下得到的典型最优值进化曲线

表 2 10 维 Rosenbrock 函数优化结果

算 法	收敛至要求范围内的微粒比例/%
传统改进算法	67.27
本文方法($c_1 = 10, c_2 = 2$)	88.94
本文方法($c_1 = 5, c_2 = 2$)	89.97

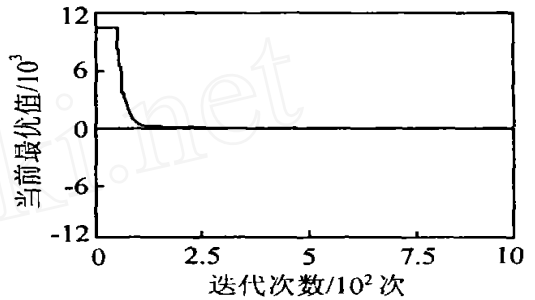


图 1 20 维 Rosenbrock 函数寻优结果动态($c_1 = 5, c_2 = 2$)

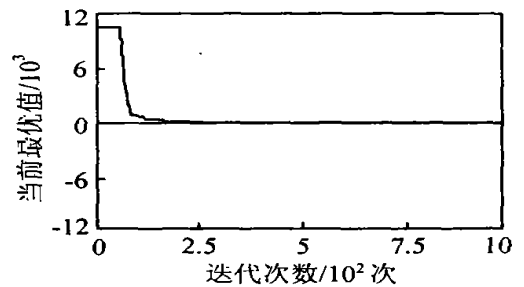


图 2 20 维 Rosenbrock 函数寻优结果动态($c_1 = 10, c_2 = 2$)

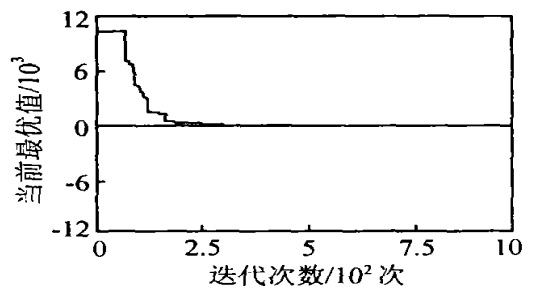


图 3 20 维 Rosenbrock 函数寻优结果动态(传统改进算法)

5 仿真结果分析

以上仿真结果充分证明了本文算法的优越性 总体而言,各算法在函数维数增加时,其性能均有所降低,但传统方法性能的降低幅度更大,其原因在于传统算法对于高维、多峰函数求解的不适应性,而采用本文算法,由于在运算初期就考虑了微粒求解多元最优值的总体分布特征,并且其步幅系数采取与

当前微粒和多元最优值所处区间差别成比例的模式,这就必然使得处于较差寻优区间内的微粒能以较大步幅尽快跳出局部最优

另外,因为运算初期的运动规划矢量采用的是基于各事先划分区间重心坐标的比较值,所以在具体的运算迭代过程中,可以根据运算初期的计算结果直接调用。在这种运算模式下,因多元规划而导致的计算量的增加和寻优速度的降低,可通过运算初期步幅的增加进行适当的抵消;同时,由于运算后期进入设定收敛域之后,其步幅又以一定的倍数减小,必然能够较好地协调计算速度与精度之间的关系,达到良好的收敛效果

仿真曲线可形象地反映算法的收敛性特征。总体上看,本文算法收敛速度和收敛过程的平稳性要优于传统的改进型算法(尤其是当 $c_1/c_2 = 2.5$ 时),这说明多元规划系数之间相对比例大小的设置问题是值得研究的。一般而言,各最优值在群体最优值集合中所处的位置越靠前,其系数应越大;相邻最优值之间的规划系数比例越大,排列靠前的多元最优值对微粒运动的影响也越大,从而系统越容易加速收敛,但同时也越容易陷入局部最优。这在本文的高维多峰函数寻优结果中得到了很好的验证。因为本文实验函数的局部最优较多,寻优难度较大,所以上级最优值与下级最优值之间规划系数的较高比例值反而不如较低比例值的寻优效果好。这在 10 维和 20 维 Rosenbrock 函数优化问题求解过程中均得到了很好的验证。当然,将此方法用于其他类型的函数寻优,其多元最优值的规划系数要根据所给函数的具体特征加以调整,以达到最佳的加速收敛效果

6 结 语

本文针对高维多峰函数的寻优问题,将微粒群算法在多元最优值规划的基础上进行步长设计,达到了很好的加速收敛效果。仿真结果证明了该算法的有效性。今后将开展的工作包括:根据所求解问题的大致种类进一步分析多元最优值各加权系数的设置问题,以及微粒分布模式对求解结果的影响问题等;针对更多典型的复杂寻优函数给出相应合理的 PSO 算法。另外,可进一步探讨多种具有智能特征的多元最优规划模式在微粒群算法寻优中的应用,针对各类复杂寻优函数,进一步改进算法的收敛模式和动稳态性能

参考文献(References):

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. *Proc IEEE Int Conf on Neural Networks* [C]. Piscataway: IEEE Service Center, 1995. 1942-1948
- [2] Eberhart R C, Shi Y. Particle swarm optimization: Developments, applications and resources [A]. *Proc Congress on Evolutionary Computation* [C]. Seoul, Piscataway: IEEE Service Center, 2001. (b): 81-86
- [3] 谢晓峰, 张文俊. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-133
(Xie X F, Zhang W J. Review of particle swarm optimization[J]. *Control and Decision*, 2003, 18(2): 129-133)
- [4] Fukuyama Y, Yoshida H. A particle swarm optimization for reactive power and voltage control in electric power systems [A]. *Proc Congress on Evolutionary Computation* [C]. Seoul, Piscataway: IEEE Service Center, 2001. 1232-1239
- [5] Eberhart R C, Hu X. Human tremor analysis using particle swarm optimization [A]. *Proc Congress on Evolutionary Computation* [C]. Washington, Piscataway: IEEE Service Center, 1999. 1927-1930
- [6] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization [A]. *Proc Congress on Evolutionary Computation* [C]. Washington, Piscataway: IEEE Service Center, 1999. 1951-1957
- [7] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization [A]. *Proc Congress on Evolutionary Computation* [C]. San Diego, 2000. 84-88
- [8] Kennedy J. Small worlds and megaminds: Effects of neighborhood topology on particle swarm performance [A]. *Proc Congress on Evolutionary Computation* [C]. Piscataway: IEEE Service Center, 1999. 1931-1938
- [9] Suganthan P N. Particle swarm optimizer with neighborhood operator [A]. *Proc of the 1999 Congress on Evolutionary Computation* [C]. Piscataway: IEEE Service Center, 1999. 1958-1962
- [10] He Z, Wei C, Yang L, et al. Extracting rules from fuzzy neural network by particle swarm optimization [A]. *Proc IEEE Int Conf on Evolutionary Computation* [C]. Anchorage, 1998. 74-77
- [11] Shi Y, Eberhart R. Fuzzy adaptive particle swarm optimization [A]. *Proc Congress on Evolutionary Computation* [C]. Seoul, Piscataway: IEEE Service Center, 2001. 101-106

(下转第 1394 页)

$$2x^T P \dot{y}u \quad x^T P x + (P \dot{y}u)^T P^{-1} P \dot{y}u = x^T P x + u^T \dot{y} P^T \dot{y} u. \quad (26)$$

则

$$2x^T P \dot{y}u \quad P \quad \bar{x}^2 + P \quad \dot{y}^2 u^2. \quad (27)$$

将式(25)和(27)代入(24)得

$$\dot{V}(\bar{x}) \quad (-\mu + 2 P + L_\sigma^2 P \quad w^2) \bar{x}^2 + P \quad \dot{y}^2 u^2.$$

定义

$$\alpha(r) = -(-\mu + 2 P + L_\sigma^2 P \quad w^2) r^2, \quad \Psi(r) = P \quad r^2 r^2,$$

对 $r \in R$, 有

$$\dot{V}(\bar{x}) \quad \Psi(\dot{y}u) - \alpha(\bar{x}).$$

若 $\alpha(\cdot)$ 和 $\Psi(\cdot)$ 为 K 函数, 则 $V(\bar{x})$ 为一个 ISS-Lyapunov 函数. 由于 $\Psi(\cdot)$ 总是满足该条件的, 当 $-\mu + 2 P + L_\sigma^2 P \quad w^2 < 0$ 时, 系统(19)是输入-状态稳定的, 即

$$w^2 < \frac{\mu - 2 P}{L_\sigma^2 P}, \quad P < \frac{\mu}{2}.$$

5 结论

本文针对非自治的动态神经网络系统, 深入分析了平衡点的存在性、唯一性和全局渐近稳定性, 给出了系统输入-状态稳定的充分条件, 构建了 ISS-Lyapunov 函数并采用该函数确保系统的全局渐近

稳定性. 该方法只需满足全局 Lipschitz 条件, 比已有方法的条件放宽了^[6], 因此更有利于神经计算和设计全局稳定的神经网络, 在理论和工程应用中具有重要的意义.

参考文献(References):

[1] Zhang J Y, Dai H Y, Wu P B. Global stability analysis in Hopfield neural networks[J]. *Control Theory and Applications*, 2003, 20(2): 180-184

[2] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. *Proc of National Academy of Science*, 1982, 79(8): 2554-2558

[3] Hopfield J J. Neurons with graded response have collective computation properties like those of two state neurons[J]. *Proc of National Academy of Science*, 1984, 81(10): 3088-3092

[4] Michel A N, Farrel J A, Porod W. Qualitative analysis of neural networks[J]. *IEEE Trans on Circuit and Systems*, 1989, 36(2): 229-243

[5] Guez A, Protopopescu V, Barhen J. On the stability, storage capacity and design of nonlinear continuous neural networks[J]. *IEEE Trans on System, Man, and Cybernetics*, 1988, 18(1): 80-87.

[6] Juang J C. Stability analysis of Hopfield-type neural networks[J]. *IEEE Trans on Neural Networks*, 1999, 10(6): 1366-1374

(上接第 1367 页)

[12] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[A]. *Proc of the 6th Int Symposium on Micro Machine and Human Science* [C]. Nagoya, Piscataway: IEEE Service Center, 1995. 39-43

[13] Eberhart R C, Shi Y. Tracking and optimizing dynamic systems with particle swarms[A]. *Proc Congress on Evolutionary Computation* [C]. Seoul, Piscataway: IEEE Service Center, 2001. (a): 99-100

[14] Fan H Y. A modification to particle swarm optimization algorithm[J]. *Engineering Computations*, 2002, 19(8): 970-989

[15] Kennedy J. The particle swarm: Social adaptation of knowledge[A]. *Proc Int Conf on Evolutionary Computation* [C]. Indianapolis, Piscataway: IEEE Service Center, 1997. 303-308

[16] Kennedy J. Methods of agreement: Inference among

the elements[A]. *Proc 1998 Int Symposium on Intelligent Control* [C]. Piscataway: IEEE Service Center, 1998. 883-887.

[17] Naka S, Grenji T, et al. Practical distribution state estimation using hybrid particle swarm optimization[A]. *Proc of IEEE PES Winter Meeting* [C]. Columbus, 2001. 1-6

[18] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[A]. *Proc of the 1999 Congress on Evolutionary Computation* [C]. Piscataway: IEEE Service Center, 1999. 1945-1950

[19] Shi Y, Eberhart R C. A modified particle swarm optimizer[A]. *Proc of the IEEE Int Conf on Evolutionary Computation* [C]. Piscataway: IEEE Press 1998. (b): 69-73