

基于 t 分布变异的进化规划

周方俊, 王向军, 张 民

(海军工程大学电气与信息工程学院, 湖北武汉 430033)

摘 要: 参考已提出的各种变异算子, 提出了基于 t 分布变异的进化规划. 在该算法中, 使用 t 分布将高斯分布和柯西分布衔接起来, 改变自由度 n 可以调整其变异性能. 通过仿真得到了 t 算子随 n 和 σ 变化的一般规律, 初步总结了进化代数、变异尺度和求解精度三者之间的关系. 在此基础上, 给出了一个基于 t 算子的自适应性进化规划算法, 该算法将高斯变异和柯西变异的优点结合起来, 仿真表明算法具有良好的性能.

关键词: 进化规划; 算子; t 分布; t 算子

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2008) 04-0667-05

Evolutionary Programming Using Mutations Based on the t Probability Distribution

ZHOU Fang-jun, WANG Xiang-jun, ZHANG Min

(Electrical and Information Engineering College, Naval University of Engineering, Wuhan, Hubei 430033, China)

Abstract: Consulted some previous mutations, this paper proposed an evolutionary programming (EP) using mutations based on the t probability distribution (t EP). The t probability distribution can connect with Gaussian and Cauchy probability distribution. Its variance is changeable by adjusting degree of freedom n . Rules of t mutations according to the change of n and standard deviation are analyzed by simulations, and the relationship among number of generations, standard deviation and minimum fitness are briefly summarized. Then an EP using adaptive t mutations (AtEP) is put forward at the end. The AtEP uses several t mutations with different n at the same time, so it can integrate the advantages of Gaussian and Cauchy mutation. And simulation evidence shows the algorithm is better than classical EP algorithm.

Key words: evolutionary programming; mutation; t probability distribution; t mutation

1 引言

进化算法 (Evolutionary Algorithms, EA) 借鉴生物界自然选择法则、遗传机制, 模拟生物进化过程以解决函数优化、神经网络结构优化、模式识别与系统辨识等问题^[1-3]. 进化规划 (Evolutionary programming, EP) 是 EA 中的代表算法之一, 由美国 L. J. Fogel 提出, D. B. Fogel 等作了改进, 逐渐得到广泛的应用. EP 的进化操作主要依赖突变^[4], 直接利用实数编码, 表述问题更加自然, 搜索过程比较平稳, 适用于 TSP、神经网络学习、模式识别等问题^[5-7]. 本文参考 EP 常用的几个变异算子, 提出基于 t 分布的变异算子. 通过改变自由度 n , 算子变异幅度可在高斯和柯西算子之间变化, 局部开发和全局探索能力两者权重可变. 对典型算例的仿真证明, 该算子在进化计算中有良好的灵活性.

2 进化算子

2.1 高斯算子和柯西算子

高斯算子的使用起源于进化策略 (Evolution Strategies, ES)^[8,9], 并被推广到 EP 等其他算法中^[10]. 标准高斯分布密度 $N(0, 1)$ 的期望为 0, 方差为 1. 受快速模拟退火算法的启发, Yao 等用柯西算子代替高斯算子, 以产生更大的变异^[11]. 标准柯西分布概率密度 $C(0, 1)$ 的期望不存在, 方差无限大. 文献^[12] 认为使用柯西算子的 FEP 较传统 EP 有一定的优点.

标准高斯分布和柯西分布的方差相差相当悬殊, 态上表现为高斯算子局部开发能力较好, 柯西算子全局探索能力较好. 为综合利用两者优点, 陆续提出了平均变异、混合变异等改进方法, 尝试将两者优点结合起来, 提高算法的灵活性和求解效果^[12-14]. 如文献^[15] 将高

斯和柯西算子卷积结合得到平均算子;而文献[16]指出,实际高维计算中平均算子和柯西算子基本一样,不具备类似于高斯算子的局部搜索能力。

2.2 Lévy 算子

Lévy(莱维)分布的概率密度函数为: $L_{\alpha, \nu}(y) = \frac{1}{\pi} \int_0^{\infty} e^{-yq} \cos(qy) dq, y \in \mathbf{R}$. 在 $\nu = 1$ 下, $\alpha = 1$ 得到 $C(0, 1)$, $\alpha = 2$ 对应于 $N(0, 2)$. 文献[17]将 Lévy 分布引入到 EP 中,通过改变 α 值来获得不同的变异算子. 通过仿真,LEP 收敛效果好于 GEP. 但编制 Lévy 算子是较为困难的. 文献[18]提出的生成服从 Lévy 分布的随机数算法要求随机数数量不能太少. 文献[17]提出的方法可以产生服从 $0.8 < \alpha < 1.95$ 的 Lévy 分布随机数,但仍然不能产生高斯分布。

3 t 分布

t 分布又称学生分布,由英国戈塞特(Gosset W S)提出,费希尔(Fisher R A)给出严格推导,开创了小样本统计理论的先河^[19,20]. t 分布含有参数自由度 n , 方差均大于 1; $n \leq 2$ 时,方差不存在; $n > 2$ 时,方差为 $\frac{n}{n-2}$. $t(n \rightarrow \infty) \rightarrow N(0, 1)$, 一般 $n \geq 30$ 两者偏离可以忽略; $t(n=1) = C(0, 1)$. 即标准高斯分布和柯西分布是 t 分布的两个边界特例分布。

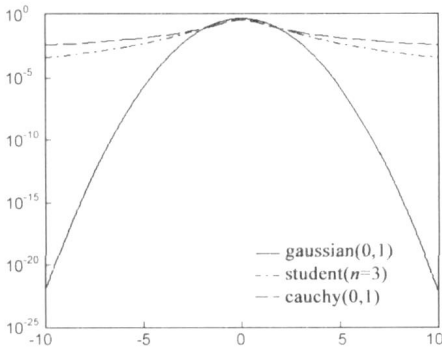


图 1 高斯分布、t 分布和柯西分布密度函数曲线

4 t 算子

伽玛(Gamma)函数 $\Gamma(n)$ 在定义域内是连续线性取值, $t(n)$ 函数在数学上也是 n 的连续函数. 将 n 在定义域内拓展到正实数,构造出 t 算子并引入到 EA 中. t 算子可以衔接高斯算子和柯西算子,改变 n 值可以获得更多不同变异幅度的算子. 不同于 Lévy 算子,只要 n 取到很大, t 算子可以非常逼近高斯算子。

5 基于 t 分布的进化规划算法

标准 t 分布的方差与自由度 n 有关, t 算子的变异能力需要使用 σ 和 n 两个参数来衡量. σ 越大, n 越小,

算子变异能力越大. 构造基于 t 分布变异的进化规划(Evolutionary Programming using mutations based on the t probability distribution, tEP), 选取若干 n 值,通过对一些测试函数的极值运算来了解 t 算子的特点及从高斯变异向柯西变异过渡的演变情况。

5.1 实验设定

算法流程如下:

- (1) 参数初始化. 设置种群个体数目 N (N 取为偶数), 随机竞争个体数目 q , 最大进化代数 K_{max} , 选取 t 算子自由度 n , 选取变异尺度 σ , 设置进化终止条件;
- (2) 种群初始化, 在问题的可行解空间中随机产生 N 个个体作为初始种群 X_0 ;
- (3) 终止进化判断. 满足给定条件则停止进化, 输出计算结果; 否则转步(4);
- (4) 变异产生后代. 个体按照如下方式进行变异, 且各维算子相互独立;

$$x_{m+1}^j = x_m^j + \sigma_m \cdot T_j(n) \quad (1)$$

- (5) 重组 x_{m+1}^j, x_m^j 组成临时种群 X^+ , 计算 X^+ 中个体的适应度, 进行随机 q 竞争, 选取排在前 N 位的个体组成新的种群 X_m ;
- (6) $k = k + 1$, 转步(3).

约束函数设为:

if $x_m >$ 定义域上限, then $x_m = x_m^-$ (定义域上限- 定义域下限) $\cdot Z$; Z 为整数.

if $x_m <$ 定义域下限, then $x_m = x_m^+$ (定义域上限- 定义域下限) $\cdot Z$; 将数值映射到定义域内.

参数选取: $N = 40, q = 0.85N, K_{max} = 2000, n = 1, 1.5, 2, 2.5, 3, 30, 50$ 补充高斯变异($n \rightarrow \infty$); 达到最大代数停止进化. 为体现进化与 n 的关系, 令 $\sigma_m = \sigma_0$ 为若干定值. 选取如下目标测试函数, 维数分别取为 30、30、30、30、2.

$$f1: F(x) = \sum_{i=1}^N x_{i0}^2, x_i \in [-1000, 100] \quad (2)$$

$$f2: F(x) = \sum_{i=1}^N \left[\sum_{j=1}^N x_j^2 \right], x_i \in [-1000, 100] \quad (3)$$

$$f3: F(x) = \left| \sum_{i=1}^N x_i \cdot \sin(\sqrt{|x_i|}) \right|, x_i \in [-500, 500] \quad (4)$$

$$f4: F(x) = \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i) + 10), x_i \in [-512, 512] \quad (5)$$

$$f5: F(x) = -20 \exp \left[-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \right]$$

$$- \exp\left[\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)\right] + 20) + e,$$

$$x_i \in [-32, 32] \quad (6)$$

$$f_6: F(x) = 4x_1^2 - 21x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

$$x_i \in [-5, 5] \quad (7)$$

5.2 试验结果

多次独立试验后将各函数的进化计算过程作图. 其中横坐标表示进化代数, 纵坐标表示种群平均适应度函数值. 在选定的测试函数中, f_1 和 f_2 只有一个极值, $f_3 \sim f_5$ 具有非常多的局部极值, f_6 具有不多的几个局部极值.

对于 f_6 等低维仿真计算, 各 t 算子都能够以较快的速度和精度收敛到最优值附近. 重点对其他高维计算进行仿真分析, 以对 f_4 仿真为例, 清晰地比较出高斯变异到柯西变异的过渡演化. 由图 2, 较大 n 值的 t 算

子与高斯算子性能较接近, 这与数理统计中 N 较大时 $t_\alpha(n) \approx u_\alpha$ 吻合; $n = 1 \sim 2$ 的 t 算子方差均不存在, 但高 n 值时方差是低 n 值时的低阶无穷小, 各求解曲线相互距离较大. n 越小, 算子性能差别越大, 最终限制到柯西算子; n 越大, t 算子性能过渡趋缓, 最终限定到高斯算子, 求解曲线愈加接近. 取较小的 n 可获得介于高斯和柯西中间的变异.

设进化曲线在零进化代数位置为首部, 直观地将进化曲线体现的普遍规律描述如下.

纵向比较 σ . 随着 σ 增大, 各算子的函数进化曲线尾部自右上方逐渐向左下方垂落. 存在与 x 轴平行的某条目标求解精度基准线, 各曲线靠近基准线后反弹, 精度变差. 且精度基准线随 σ 的增大而升高.

横向比较 n . 随着 n 增大, 各算子进化曲线尾部依次垂落, n 小的进化曲线最先下垂, 进入收敛; 各曲线靠近基准线后依次反弹, n 小的进化曲线先反弹且上升最高.

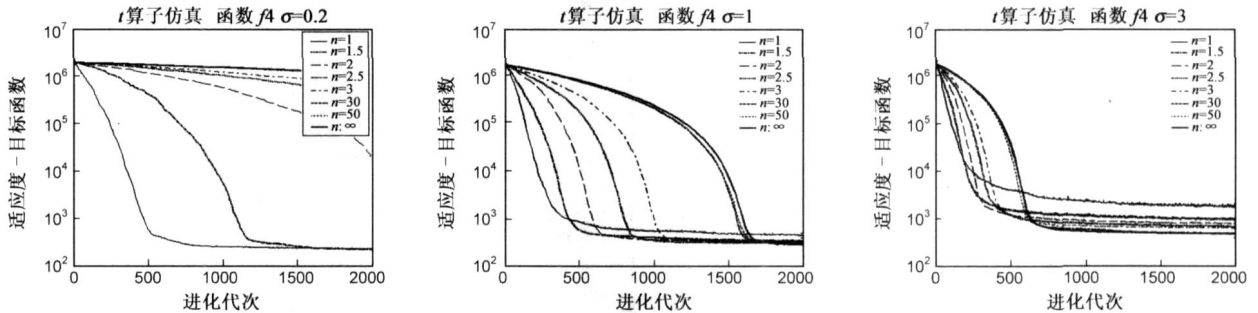


图 2 函数 f_4 的 tEP 仿真进化过程

5.3 仿真实验机理分析

在确定的维数及 K_{max} 下, 针对采用定值 σ 、 n 的 tEP , 考察其种群的平均函数适应度变化.

(1) 存在 σ_{min} , 当 $\sigma < \sigma_{min}$ 时, 变异步长太小, 所有算子都无法使个体向最优解靠近, 是“零”收敛变异.

(2) $\sigma > \sigma_{min}$. 由于 t_1 算子容易大变异, 算法克服小步长 σ 的约束开始靠近最优解. 在有限的进化代数内, 当 $\sigma = \sigma_{K_{max}}$, t_1 算子在进化末期达到最好的求解精度. 在 $\sigma_{min} < \sigma < \sigma_{K_{max}}$, 其他各 t_n 算子依照 n 增大顺序依次克服小 σ 的约束逐渐向最优解靠近.

(3) $\sigma > \sigma_{K_{max}}$. 由于 σ 较大, 种群中个体与全局极值间的距离不能进一步缩小, 整个算法可实现的最佳求解精度变小. t_1 算子的综合变异能力太大, 求解精度最先变差, 相邻的 t 算子相对继续寻优, 结果 t_1 算子的进化效果反而差于 $t_{1+\Delta n}$ 算子. σ 再增大, 各 t_n 的求解精度陆续差于 $t_{n+\Delta n}$ 算子.

(4) $\sigma \geq \sigma_{(\infty)}$ 后, t_∞ 算子方差最小, 求解精度相对最高, 各算子的收敛效果按照 n 减小顺序依次变差.

如图 3, 如果将 σ 看作反馈控制系统中的输入, 每次变异为一次脉冲, 测试函数为被控对象, 求解精度为

系统输出, 变异算子为校正装置中的 PID 控制器. 则:

(1) 自由度小的 t 算子具有较强的微分作用, 对较小的 σ 脉冲产生的变化较微弱的输入响应速度快, 产生较明显的控制输出, 求解精度提高得快, 算子有快速性. 在较大的 σ 脉冲产生的变化较强烈的输入下, 微分作用较强, 系统的超调量增大, 系统的动态求解过程不平稳, 误差大, 求解精度不高.

(2) 自由度大的 t 算子具有较强的积分作用, 对 σ 脉冲产生的变化响应与前者正相反, 算子有稳定性.

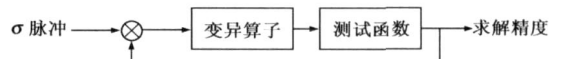


图 3 t 算子类比 PID 控制的原理框图

文献[21]将不收敛情况区分为两种不收敛, 在 t 算子仿真进化中表现为在较小的 σ 下, n 大易使种群形成第一类不收敛, 反之会实现较佳的寻优精度, 说明第一类不收敛受 n 影响较大; 在较大的 σ 下, 不论 n 大小, 种群的求解精度都不是很高, 种群均会形成第二类不收敛, 说明第二类不收敛受 σ 影响较大.

可以看出, 自由度 n 与 σ 均会影响 t 算子的进化效果, 与所设定的进化代数、 σ 、种群规模、维数及函数性质等有关, 没有某一确定的 t 算子在算法中能获得最

好的进化效果. t 算子的两个极限: 柯西算子和高斯算子不是在所有的进化过程中均能表现出最好的性能.

由图 4 对设定的进化代数, 求解精度随 σ 增大先变好后变差, 对应一个 σ_{\min} , 小于此值, 无论进化多少代, 都不能开始收敛, 超过 $\sigma_{k_{\max}}$ 后, 求解精度开始变差; 对设定的 σ , 进化代数越多, 最终求解精度越高; 对进化代数增加, 能获得的最佳求解精度相应的 $\sigma_{k_{\max}}$ 越小.

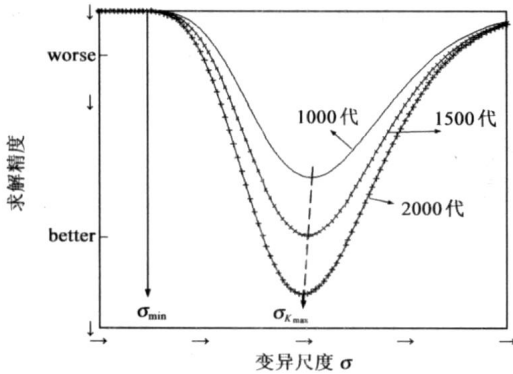


图 4 变异尺度、进化代数和求解精度关系

6 基于 t 算子的自适应性进化规划算法

借鉴文献[17]的 ALEP 方法, 构造基于 t 算子的自

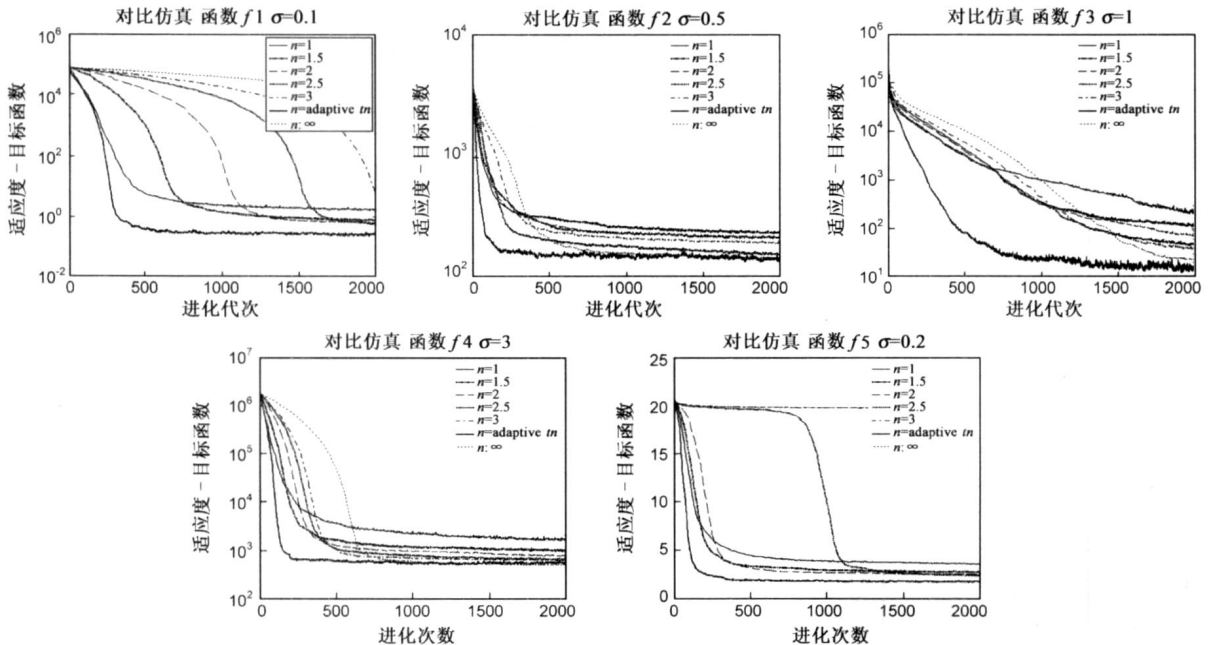


图 5 各函数的 AtEP 进化过程

7 结论

本文提出基于 t 分布变异的进化规划, 通过对典型测试函数的仿真分析, 初步了解 t 算子随自由度 n 和变异尺度 σ 值变化体现的一般变异规律, 并总结了变异尺度、进化代数和求解精度三者关系. t 算子能实现高斯算子和柯西算子之间平滑过渡, 且统一了后两者

适应性算法(A t EP), 并检验其求解效果. 各子算子在同一父体下依照自身的变异特性生成不同的新个体, 由算法动态选择较优的个体作为子代, 进入下一轮进化. 反复依此操作, 以取得较单一算子算法更好的效果. 选用的算子有 $t_{1}, t_{1.5}, t_{2}, t_{2.5}, t_{3}, t_{20+k}$ (k 为进化代数), 各算子产生不同的个体, 依据适应度择优选择作为子代, 进入下轮进化. 与与普通算法作比较, 文献[17]提出自适应算法按比例缩减种群规模, 但前提是种群规模仍然较大(100). 这里按照变异后待 q 选择个体规模等量原则来确定算法的初始种群规模(80/7).

从图 5 可以看出, 与单一 t 算子表现的进化规律不同, A t EP 基本保持在最佳的收敛状态. 在进化初期向极值周围收敛很快, 在 t EP 需要较长的进化周期才能获得的精度, A t EP 在较短的进化周期就可以获得, 提高了进化的快速性. 在进化后期, t EP 最佳求解精度仅对应于某个 n , A t EP 均能获得较好的结果, 具有较强的鲁棒性.

相同条件下, 通过设置多个自由度, A t EP 的表现至少不比 t EP 差, 同样验证了在目标问题未明确的前提下, 在算法中保持多个自由度或尺度是较为合适的, 类似的情况在文献[17]已进行了较深入的讨论.

的表达形式, 使用方便、容易^[22]. 结果表明, 选择合适的 n 与 σ 来构造自适应变异算法, 可以获得优于高斯变异和柯西变异的整体优化效果.

参考文献:

[1] Yao X, Liu Y. A new evolutionary system for evolving artificial neural networks[J]. IEEE Transactions on Neural Networks,

- 1997, 8(3): 694– 713.
- [2] Sebald A V, Schlenzig J. Minimax design of neural net controllers for highly uncertain plants[J]. IEEE Transactions on Neural Networks, 1994, 5(1): 73– 82.
- [3] Bornholdt S, Graundenz D. General symmetric neural networks and structure design by genetic algorithms[J]. IEEE Transactions on Neural Networks, 1992, 5(5): 327– 334.
- [4] 云庆夏. 进化算法[M]. 北京: 冶金工业出版社, 2000. 5.
- [5] Fogel L J. Evolutionary programming in perspective: the top down view[A]. Zurada J M, Mark II R J, Robinson C J eds. Computational Intelligence: Imitating Life [C]. New York: IEEE Press, 1994. 135– 141.
- [6] Fogel D B, Stayton L C. On the effectiveness of crossover in simulated evolutionary optimization[J]. Biosystems, 1994, 32(2): 171– 182.
- [7] Angeline P J, Saunders G M, Pollack J B. An evolutionary algorithm that constructs recurrent neural networks[J]. IEEE Transactions on Neural Networks, 1994, 5(1): 54– 65.
- [8] 玄光男, 程润伟, 著. 于歆杰, 周根贵, 译. 遗传算法与工程优化[M]. 北京: 清华大学出版社, 2003. 26.
- [9] Bäck T, H Schwefel. An overview of evolution algorithms for parameter optimizations[J]. Evolutionary Computation, 1993, 1(1): 1– 23.
- [10] Fogel D B. An introduction to simulated evolutionary optimization[J]. IEEE Transactions on Neural Network, 1994, 5(1): 3– 14.
- [11] Yao X, Liu Y. Fast evolutionary programming[A]. In: Evolutionary Programming V. Proceedings of the 5th Annual Conference on Evolutionary Programming[C]. Cambridge, MA: MIT Press, 1996. 441– 451
- [12] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82– 102.
- [13] Yao X, Liu Y, Lin G. An analysis of evolutionary algorithms based on neighborhood and step size[A]. Evolutionary Programming VI[C]. Lecture Notes in Computer Science, vol. 1213, Berlin: Springer, 1997. 297– 307.
- [14] Fogel D B. Applying evolutionary programming to selected traveling salesman problem[J]. Cybernetics and Systems, 1993, 24(1): 27– 36.
- [15] Gehlhaar D K, Fogel D B. Two new mutation operators for enhanced search and optimization in evolutionary programming[A]. B Biosacchi, J Bezdak, D B Fogel, eds. Proceedings of SPIE[C]. Piscataway, NJ: IEEE Press, 1996, 3165: 260– 269.
- [16] 林丹, 李敏强, 寇纪松. 进化规划和进化策略中变异算子的若干研究[J]. 天津大学学报, 2000. 9, 33(5): 627– 630.
- [17] Lee C, Yao X. Evolutionary programming using mutations based on Lévy probability distribution[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(1): 1– 13.
- [18] R Mantegna. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic process[J]. Phys Rev E, 1994, 49(5): 4677– 4683.
- [19] 姜礼平, 等. 工程数学[M]. 湖北: 湖北科学技术出版社, 2000.
- [20] 崔智超, 王青建. 数理统计学源流及应用[J]. 大连教育学院学报, 2005, 2(2): 53– 55.
- [21] 王向军, 嵇斗, 张民. 一种多群竞争进化算法[J]. 电子学报, 2004, 32(11): 1824– 1828.
- [22] 王沫然, 等. MATLAB 与科学计算(第二版)[M]. 北京: 电子工业出版社, 2003. 9.

作者简介:



周方俊 男, 1979 年 12 月生于山东邹城, 2001 年在武汉海军工程大学获得工学学士学位, 2005 年至今在海军工程大学电气与信息学院硕士研究生在读. 研究方向为电力电子技术、电路故障诊断、进化计算.

E mail: thereisfancy@tom.com



王向军 男, 1973 年 11 月生于河北省滦南县, 分别于 1996、1999、2004 年在海军工程大学获得工学学士、硕士和博士学位. 现为海军工程大学电气与信息学院副教授, 研究方向为电力电子技术、进化计算、神经网络.