

# 区间长度可变的反向混沌优化算法

傅文渊<sup>1,2,3,4</sup>, 李国刚<sup>1,3</sup>, 王燕琼<sup>1</sup>

(1. 华侨大学信息科学与工程学院, 福建厦门 361002; 2. 中山大学电子与信息工程学院, 广东广州 510006;  
3. 厦门市专用电路系统重点实验室, 福建厦门 361008 ; 4. 福建省电机控制与系统优化调度工程技术研究中心, 福建厦门 361002)

**摘 要:** 针对现有算法在大空间和高维度寻优存在效率较低的问题, 提出一种区间长度可变的反向混沌优化算法, 并证明了该算法以概率 1 收敛于全局最优解. 算法采用区间长度可变的反向优化策略, 利用反向优化方法增大算法进化过程的多样性, 使优化的变量区间不断减小. 同时, 提出基于 Fuch 混沌映射的反向混沌优化策略增大算法逃逸局部极值的能力, 以及两级优化策略提高算法执行后期的寻优精度. 通过 22 个基准函数测试结果表明, 本文提出的算法与改进的混沌优化算法以及其他智能优化算法相比, 其搜索的综合性能要优于其他算法.

**关键词:** 混沌优化; 区间长度可变; 反向优化; 混沌映射

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2019)01-0113-09

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2019.01.015

## Opposite Based Chaos Optimization Algorithm with Variable Interval Length

FU Wen-yuan<sup>1,2,3,4</sup>, LI Guo-gang<sup>1,3</sup>, WANG Yan-qiong<sup>1</sup>

(1. College of Information Science and Engineering, Huaqiao University, Xiamen, Fujian 361002, China;  
2. School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, Guangdong 510006, China;  
3. Xiamen Key Laboratory of ASIC System, Xiamen, Fujian 361008, China;  
4. Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen, Fujian 361002, China)

**Abstract:** To improve search efficiency in large space and high dimension for the algorithm, an opposite based chaos optimization algorithm (VILOC) with variable interval length is proposed, which is verified to converge global optimum solution with probability one. Meanwhile, an opposite optimization approach to increase the diversity of the algorithm is also introduced, which gives rise to decrease of the optimized variable interval. In the implementation procedure of VILOC, an anti-chaotic optimization strategy based on Fuch chaotic map is accommodated to escape the local extremum, and the two-stage optimization strategy to increase the convergence precision. The comparisons are carried out through experiments and the numerical results demonstrate that the proposed algorithm is superior to other improved chaos optimization algorithms and intelligent optimization algorithms.

**Key words:** chaos optimization; variable interval length; opposite optimization; chaotic map

## 1 引言

混沌理论是 20 世纪物理学三大重大发明之一. 作为一门新兴学科, 其研究的理论成果已成功应用于信息科学、计算机科学、最优化理论、物理学、经济学和气象学等多个领域<sup>[1-6]</sup>. 混沌是非线性系统中普遍存在的现象, 它体现为貌似随机的不确定运动, 其动力学行为具有不确定性、不可预测、不可重复等特点. 混沌优化算

法 (Chaos Optimization Algorithm, COA) 的核心是利用混沌的不确定性和遍历性产生相应的混沌映射, 将该映射作为随机数发生器生成扰动因子并根据贪婪法则进行全局寻优.

混沌优化算法的研究主要存在 2 个局限性<sup>[7-10]</sup>: (1) 混沌优化在处理大优化空间问题的算法执行效率较低. 优化空间越大, 算法达到相同的收敛精度需要的迭代次数也相应增大. 然而混沌优化过程中单纯增大

迭代次数并不能提高搜索的效率,甚至会造成寻优集中于局部最优值附近;(2)混沌优化求解高维度优化问题时,由于混沌映射的概率分布具有唯一性,因此单纯提高混沌优化的混沌变量个数并不能显著提高优化效率.虽然传统混沌优化算法在一定范围内具有遍历性,但由于高维度优化问题复杂度高,使得优化过程中某些状态需要较长时间才能遍历完全.若全局最优解处于这些状态,那么算法需要更长的寻优时间,其搜索速度和收敛精度将极大降低.因此优化问题的维度越高,混沌优化算法的寻优效率越低.

目前关于混沌优化算法(COA)的研究主要集中于低维度COA的寻优效率<sup>[11,12]</sup>.为提高COA在高维度优化问题的寻优效率,国内外学者将混沌优化算法与其他算法融合,借鉴混沌优化良好的全局开采性能取得了较好的成果.文献[13]将Lozi混沌映射作为混沌发生器驱动差分进化算法(Differential Evolution, DE)算法的变异和交叉过程的变异参数,提高了算法的执行效率.文献[14]将混沌优化与遗传算法(Genetic Algorithm, GA)结合,利用聚类方法跟踪全局最优值轨迹,同时运用混沌优化方法避免算法陷入局部最优.文献[15]提出一种自适应惯性因子调节方法平衡算法勘探和开采能力.同时将混沌映射与粒子群优化算法(Particle Swarm Optimization, PSO)结合共同搜索,取得了较为理想的寻优效果.文献[16]在文献[15]的基础上利用线性逐点混沌映射取代Logistic混沌映射,解决了COA中收敛速度较低的问题.另外,COA与梯度优化相结合,借助于梯度算法良好的开采能力,大幅度提高算法执行后期的收敛精度<sup>[17-19]</sup>.

然而,由于实际工程中优化问题往往不能精确获得变量的优化空间,因此相比于高维度优化问题,研究大优化空间具有更强的工程实用性.迄今为止还未有文献研究大优化空间的COA.基于此,本文提出一种区间长度可变的反向混沌优化算法(Variable Interval Length Opposite based Chaos, VILOC),该算法利用无限折叠Fuch混沌映射增大随机搜索能力、两级优化策略提高算法寻优精度、反向优化策略增大逃逸局部极值的能力以及区间长度可变的反向优化策略减小优化空间.随着迭代次数的增加,优化的变量区间不断减小,算法不断趋向于最优解方向,最终实现全局寻优.该算法克服了COA大优化空间和高维度优化问题时易收敛于局部极小值点和运算效率较低的缺点,具有较强的鲁棒性.

## 2 区间长度可变的反向混沌优化算法 (VILOC)

### 2.1 区间长度可变的反向优化策略

本文提出的VILOC算法由区间长度可变的反向优

化策略、反向混沌优化策略和两级搜索策略组成.它解决了传统混沌优化算法在处理大优化空间,高维度优化问题时存在搜索效率低,易陷入局部最优的局限性.区间长度可变的反向优化策略是利用当前最优解的反向解增大算法进化过程的多样性,通过不断细化优化变量的区间长度,以达到快速逼近全局最优解的目的.

**定义1** 反向解:设 $x \in [a, b]$ ,且 $x, a, b \in R$ ,定义 $x$ 的反向解为 $\hat{x} = a + b - x$ .

假设优化问题的空间维度为 $D$ ,其中 $i \in [1, D]$ ,具体操作步骤如下:

(a) 设置优化问题的变量区间为 $[a^{(i)}, b^{(i)}]$ ,当前最优位置 $\mathbf{X}_{\min}$ 的第 $i$ 维度为 $x_{\min}^{(i)}$ ,同时设置更新后的区间为 $[c_k^{(i)}, d_k^{(i)}]$ ,其中 $[c_k^{(i)}, d_k^{(i)}] \equiv [a^{(i)}, b^{(i)}]$ ,  $k = 1$ .

(b) 设置区间长度变化因子 $\delta^{(i)}$ ,按照以下关系生成新的搜索区间 $[a_{k+1}^{(i)}, b_{k+1}^{(i)}]$ ,其中 $r$ 为均匀分布随机数,

$$a_{k+1}^{(i)} = x_{\min}^{(i)} + \delta^{(i)} \cdot r \cdot (a_k^{(i)} - b_k^{(i)}),$$

$$b_{k+1}^{(i)} = (1 - \delta^{(i)} \cdot r) \cdot b_k^{(i)} + \delta^{(i)} \cdot r \cdot a_k^{(i)}.$$

(c) 计算当前第 $i$ 维度最优位置 $x_{\min}^{(i)}$ 的反向最优位置 $\hat{x}_{\min}^{(i)} = a_k^{(i)} + b_k^{(i)} - x_{\min}^{(i)}$ .根据函数适应度值判断,若反向最优位置 $\hat{\mathbf{X}}_{\min}$ 优于 $\mathbf{X}_{\min}$ ,则按照数学关系

$$\hat{a}_{k+1}^{(i)} = (1 - \delta^{(i)} \cdot r) \cdot a_k^{(i)} + \delta^{(i)} \cdot r \cdot b_k^{(i)} \text{ 和}$$

$$\hat{b}_{k+1}^{(i)} = (1 - \delta^{(i)} \cdot r) \cdot b_k^{(i)} + (1 - \delta^{(i)} \cdot r) \cdot a_k^{(i)} - x_{\min}^{(i)}$$

生成反向搜索区间 $[\hat{a}_{k+1}^{(i)}, \hat{b}_{k+1}^{(i)}]$ ,否则算法按照区间 $[a_{k+1}^{(i)}, b_{k+1}^{(i)}]$ 执行寻优操作.

(d) 按照更新后的区间 $[c_{k+1}^{(i)}, d_{k+1}^{(i)}]$ 进行搜索,其中

$$c_{k+1}^{(i)} = \frac{1}{2}(\hat{a}_{k+1}^{(i)} + a_{k+1}^{(i)})d_{k+1}^{(i)} = \frac{1}{2}(b_{k+1}^{(i)} + b_{k+1}^{(i)}).$$

(e) 重复执行步骤(a)~(d),直至搜索到最优的变量区间.

### 2.2 基于Fuch映射的反向混沌优化策略

本文的混沌映射采用文献[20]提出的一种新型无限折叠Fuch映射.它的表达式为 $u_{k+1} = \cos(1/u_k^2)$ ,其中 $u_k \neq 0, k \in Z$ .该映射与经典的Logistic、Chebyshev等映射相比具有更强的混沌性,更灵敏的初值特性.同时,当 $u_{k+1} > 0$ 时,该映射由初始状态产生的混沌轨道分离程度随时间增加而急剧增大,具有更好的遍历特性.

**定义2** Fuch映射的混沌向量:设优化问题的空间维度为 $D$ ,Fuch映射的混沌变量为 $u_k^{(i)}$ ,其中 $i \in [1, D]$ , $k$ 为混沌迭代次数.令Fuch映射第 $k$ 次迭代的混沌向量为 $\mathbf{U}_k = (u_k^{(1)}, u_k^{(2)}, \dots, u_k^{(i)}, \dots, u_k^{(D)})$ ,则第 $k+1$ 次迭代的混沌向量为 $\mathbf{U}_{k+1} = (u_{k+1}^{(1)}, \dots, u_{k+1}^{(D)}) = (\cos(1/u_k^{(1)^2}), \dots, \cos(1/u_k^{(D)^2}))$ .

混沌优化的基本思想是将优化变量通过混沌迭代发生器产生混沌序列进行迭代,然后混沌序列又逆向

映射到优化变量区间内,确保优化过程自变量的取值范围不改变.为克服混沌优化存在的大空间、高维度寻优效率不高的缺点以及提高混沌变量在优化空间内的遍历特性,本文提出一种反向混沌优化策略.该策略的执行过程包含三个步骤:首先将混沌变量  $u_k^{(i)}$  映射到优化变量  $x^{(i)}$  的区间  $[a^{(i)}, b^{(i)}]$ , 其中  $i \in [1, D]$ , 然后利用混沌变量在优化空间内搜索得到当前的最优解  $\mathbf{X}_{\min}$ . 其次,将当前最优解  $\mathbf{X}_{\min}$  执行更新操作  $\tilde{\mathbf{X}}_{\min} = \mathbf{X}_{\min} \cdot |\mathbf{U}_k|$ , 评价该优化解  $\tilde{\mathbf{X}}_{\min}$  是否优于当前最优解  $\mathbf{X}_{\min}$ . 如果优于当前最优解  $\mathbf{X}_{\min}$ , 则更新最优解为  $\tilde{\mathbf{X}}_{\min}$ , 否则不更新当前的最优解  $\mathbf{X}_{\min}$ . 最后,计算当前最优解  $\mathbf{X}_{\min}$  的反向解  $\hat{\mathbf{X}}_{\min}$  并评价该反向解  $\hat{\mathbf{X}}_{\min}$  是否优于当前最优解  $\mathbf{X}_{\min}$ , 如果优于当前最优解,则更新最优解,同时将当前混沌向量  $\mathbf{U}_k$  变更为  $1 - |\mathbf{U}_k|$ . 否则不更新当前的最优解  $\mathbf{X}_{\min}$  和混沌向量  $\mathbf{U}_k$ .

### 2.3 两级搜索策略

本文对优化解的扰动评价采用两级搜索策略.首先在优化空间内采用 Fuch 映射并结合反向混沌优化策略遍历当前最优解.其次,以当前最优解为中心,利用反向混沌优化策略附加微小随机混沌扰动,进行精细搜索以获得全局最优解.

首先,第一级搜索是针对基本混沌优化算法的精细搜索进行改进.基本混沌优化算法精细搜索的数学表达式如式(1)所示,

$$\bar{\mathbf{X}} = \mathbf{X}_{\min} + \beta \cdot \mathbf{U}_k \quad (1)$$

其中  $\mathbf{X}_{\min}$  为当前最优解,  $\mathbf{U}_k$  为第  $k$  次迭代的混沌向量.这种搜索方式存在  $\beta$  取值不易确定和搜索方向单一的问题.因此本文将其改进为式(2), (3)和(4),

$$\beta_k = (1 - \theta_k) \cdot \bar{\mathbf{U}}_k + \theta_k \cdot \mathbf{U}_k \quad (2)$$

$$\theta_k = 1 - \frac{1}{k} \quad (3)$$

$$\bar{\mathbf{X}} = \mathbf{X}_{\min} + \eta \cdot \beta_k \cdot \mathbf{U}_k \quad (4)$$

其中  $\bar{\mathbf{U}}_k$  为当前搜索的最优 Fuch 映射向量,  $\mathbf{U}_k$  为第  $k$  次迭代的混沌向量,  $\eta$  为高斯分布的随机数,  $\beta_k = (\beta_k^{(1)}, \dots, \beta_k^{(i)}, \dots, \beta_k^{(D)})$ .  $\beta_k$  和  $\theta_k$  随迭代次数自适应变化.改进后的算法开始搜索时,  $\theta_k$  较小, 最优 Fuch 混沌向量  $\bar{\mathbf{U}}_k$  所占权重较大, 因此  $\bar{\mathbf{X}}$  沿当前搜索的最优方向进化.当迭代次数的不断增加时,  $\theta_k$  逐渐增大, 混沌映射向量  $\mathbf{U}_k$  所占权重逐渐变大, 此时算法更易跳出局部极值点而搜索到全局最优解.另外,  $\eta$  为高斯分布的随机数, 具有重尾效应, 因而能有效提高收敛速度.

其次,第二级搜索是以当前最优解  $\mathbf{X}_{\min}$  为中心, 利用反向混沌优化策略附加微小随机混沌扰动进行精细搜索.具体的数学表达式为式(5)和(6),

$$\beta_k = (1 - \theta_k) \cdot (1 - |\bar{\mathbf{U}}_k|) + \theta_k \cdot \mathbf{U}_k \quad (5)$$

$$\bar{\alpha} x^{(i)} = a^{(i)} + b^{(i)} - x_{\min}^{(i)} - \eta \cdot \beta_k \cdot u_k^{(i)} \quad (6)$$

其中,  $a^{(i)}$  和  $b^{(i)}$  为优化变量  $x^{(i)}$  的取值区间  $[a^{(i)}, b^{(i)}]$ ,  $i \in [1, D]$ ,  $\bar{\alpha} x^{(i)}$  为当前搜索的反向最优解的第  $i$  维度值,  $x_{\min}^{(i)}$  为当前搜索的反向最优解的第  $i$  维度值.

### 3 VILOC 算法实现

综上所述,本文提出的 VILOC 算法具体执行步骤如下:

(1) 算法初始化. 随机产生一组初始向量作为初始解  $\mathbf{X} = (x^{(1)}, x^{(2)}, \dots, x^{(i)}, \dots, x^{(D)})$ ,  $i \in [1, D]$ , 计算函数适应度值  $f(\mathbf{X})$ , 同时假设当前最优解为  $\mathbf{X}_{\min} = \mathbf{X}$ . 设置 VILOC 算法总迭代次数  $M$  以及初始迭代次数  $k = 1$ 、混沌粗搜索迭代次数  $M_1$ 、细搜索迭代次数  $M_2$ 、混沌粗搜索和细搜索初始迭代次数分别为  $k_1 = 1$  和  $k_2 = 1$ ;

(2) 将  $\mathbf{X}$  通过 Fuch 混沌映射到优化空间  $[\mathbf{a}, \mathbf{b}]$ , 成为前混沌优化向量  $\mathbf{X}_p$ . 若  $\mathbf{X}_p$  优于当前最优解, 则  $\mathbf{X}_{\min} = \mathbf{X}_p$ , 同时  $\mathbf{U}_k = \mathbf{X}$ ;

(3) 对当前最优解  $\mathbf{X}_{\min}$  执行更新操作  $\tilde{\mathbf{X}}_{\min} = \mathbf{X}_{\min} \cdot |\mathbf{U}_k|$ , 根据函数适应度值判断, 若优于当前最优解, 则更新为  $\mathbf{X}_{\min} = \tilde{\mathbf{X}}_{\min}$ ;

(4) 计算当前最优解  $\mathbf{X}_{\min}$  的反向解  $\hat{\mathbf{X}}_{\min}$ , 若优于当前最优解, 则更新为  $\mathbf{X}_{\min} = \hat{\mathbf{X}}_{\min}$ , 同时将当前混沌最优向量  $\mathbf{U}_k$  变更为  $1 - |\mathbf{U}_k|$ . 否则不更新当前的最优解  $\mathbf{X}_{\min}$  和混沌变量  $\mathbf{U}_k$ .  $k_1 = k_1 + 1$ , 重复执行步骤(2) ~ 步骤(4), 直至粗搜索迭代次数为  $M_1$ ;

(5) 执行一级搜索, 将当前最优混沌向量  $\mathbf{U}_k$  作为混沌迭代的初始值生成相应的混沌迭代向量, 按照式(2) ~ (4) 计算  $\bar{\mathbf{X}}_{\min}$ , 若优于当前最优解, 则更新为  $\mathbf{X}_{\min} = \bar{\mathbf{X}}_{\min}$ , 并且将当前最优混沌向量更新为  $\mathbf{U}_k = 1 - |\mathbf{U}_k|$ ;

(6) 执行二级搜索, 根据式(5)和(6)计算  $\alpha \bar{\mathbf{X}}$ , 若优于当前最优解, 则更新为  $\mathbf{X}_{\min} = \alpha \bar{\mathbf{X}}$ , 同时将当前最优混沌向量更新为  $\mathbf{U}_k$ ;

(7)  $k_2 = k_2 + 1$ , 重复执行步骤(5)和(6), 判断迭代次数是否大于  $M_2$ , 若大于则停止搜索, 输出当前最优值  $\mathbf{X}_{\min}$ , 否则转入步骤(5)继续执行;

(8) 根据区间长度可变的反向优化策略, 将优化变量区间  $[a_k^{(i)}, b_k^{(i)}]$  变更为  $[c_{k+1}^{(i)}, d_{k+1}^{(i)}]$ , 为确保  $[c_{k+1}^{(i)}, d_{k+1}^{(i)}]$  的取值范围不超过  $[a^{(i)}, b^{(i)}]$ , 因此做如下处理: 如果  $c_{k+1}^{(i)} < a^{(i)}$ , 则  $c_{k+1}^{(i)} = a^{(i)}$ . 如果  $d_{k+1}^{(i)} > b^{(i)}$ , 则  $d_{k+1}^{(i)} = b^{(i)}$ ;

(9)  $k = k + 1$ , 重复执行步骤(2) ~ 步骤(8), 直至迭代次数  $k = M$  时, 算法寻优结束, 输出最优解  $\mathbf{X}_{\min}$ .

本文的优化算法中, 区间长度变化因子  $\delta^{(i)}$  的取值范围在  $(0, 1)$ . 两级搜索策略中的  $\eta$  取值为  $[0.01, 0.9]$

且服从高斯分布.  $\eta$  越小, 则反向优化解  $\bar{oX}$  遍历的区间范围越大, 逼近全局最优解的概率越大, 算法更易跳出局部极值点搜索到全局最优解, 因而能显著提高搜索全局最优解的寻优速度.

#### 4 VILOC 算法收敛性分析

为了对 VILOC 算法进行收敛性分析, 引入相关定义和假设.

**定义 3**<sup>[21]</sup> 设  $\zeta_m (m = 1, 2, \dots)$  为概率空间上定义的随机序列, 且随机序列  $\{\zeta_m\}$  依概率收敛于随机变量  $\zeta$ , 若  $P\{\bigcup_{m=1}^{+\infty} \cup_{l \geq m} (|\zeta_l - \zeta| \geq \varepsilon)\} = 0$ , 则称随机序列  $\{\zeta_m\}$  以概率为 1 收敛于随机变量  $\zeta$ .

**假设 1**<sup>[22]</sup> 若  $f(Q(z, \zeta)) \leq f(z)$ , 且  $\zeta \in S$ , 则  $f(Q(z, \zeta)) \leq f(\zeta)$ . 其中  $Q$  为优化问题的解函数,  $\zeta$  为概率空间  $(R^D, V, \Psi)$  产生的随机向量,  $f(x): R^D \rightarrow R, S \subset R^D, V$  为  $R^D$  上的一个  $\sigma$  代数,  $\Psi$  为  $V$  上的概率测度.

**假设 2**<sup>[22]</sup> 对于  $S$  的任意 Borel 子集  $A$ , 若其测度  $\rho(A) > 0$ , 则有  $\prod_{k=0}^{+\infty} (1 - \Psi_k(A)) = 0$ . 其中,  $\Psi_k(A)$  是第  $k$  代进化算法  $Q$  在  $A$  上的概率测度.

**定理 1**<sup>[22]</sup> 假设  $f$  为可测函数,  $S$  为  $R^D$  上任意 Borel 可测子集  $A, \{x_k\}, (k = 1, 2, \dots)$  为概率空间上算法产生的随机解, 则当满足假设 1 和 2 时, 算法以概率 1 收敛.

**定理 2** 假设 VILOC 算法优化问题的目标函数  $f$  是可测的, 并且解空间  $S$  为可测子集, 那么算法以概率 1 收敛于全局最优解.

**证明** 首先定义函数  $Q$  为

$$Q(X_{\min,k}, X_k) = \begin{cases} X_{\min,k} & \text{if } f(X_{\min,k}) \leq f(g(X_k)) \\ X_k & \text{if } f(X_{\min,k}) > f(g(X_k)) \end{cases} \quad (7)$$

其中,  $g(X_k) = X_{k+1} = X_k \cdot \chi_{M_2}(X_{\min}) + X_{\min} \cdot \chi_{M_1}(X_{\min}) + \bar{X}_{\min} \cdot \chi_{M_1}(\bar{X}_{\min}) + \hat{X}_{\min} \cdot \chi_{M_1}(\hat{X}_{\min}) + \bar{oX} \cdot \chi_{M_2}(\bar{oX}) + \bar{X}_{\min} \cdot \chi_{M_2}(\bar{X}_{\min})$ ,  $\chi_{M_1}(\cdot) = \chi_{k \in (-\infty, M_1)}(\cdot)$ ,  $\chi_{M_2}(\cdot) = \chi_{k \in (M_2, +\infty)}(\cdot)$ ,  $\chi_{M_1}(\cdot)$  和  $\chi_{M_2}(\cdot)$  均表征指示函数. 联立式(4)、(5)和(6)推得 VILOC 算法满足假设 1. 其次, 由概率测度论可知 VILOC 算法属于下降算法, 其执行过程的解满足  $f(X_{\min,k}) \geq f(X_{\min,k+1})$ , 因此  $f(X_{\min,M_1}) \geq f(X_{\min,M_2}) \geq f(X_{\min,M})$ . 设  $L$  表示全局最优解  $X_{\min,M}$  的邻域, 则  $f(X_{\min,k}) \geq f(X_{\min,M})$ , VILOC 算法产生的解序列  $(X_{\min,k}, f(X_{\min,k}))$  落入  $L$  的事件集合为  $T_k$ . 由区间长度可变的反向优化策略可知在算法执行过程中优化变量寻优空间  $[c_{k+1}^{(i)}, d_{k+1}^{(i)}]$  不断缩小, 可以推得 VILOC 算法执行过程中的概率测度单调不减, 所以

$\prod_{k=0}^{+\infty} (1 - \Psi_k(A)) = 0$ . 由此可知 VILOC 算法满足假设 2. 根据定理 1, VILOC 算法能以概率 1 收敛于全局最优解.

### 5 仿真实验分析

#### 5.1 测试函数及设置

高效的全局优化算法必须具备较高的收敛精度和跳出局部最优解的能力. 为了全面验证提出的 VILOC 算法性能, 本文采用 22 个基准函数测试<sup>[10]</sup>, 涵盖单模、多模、高维度和大优化空间的特点, 其中  $F18 \sim F22$  分别为文献[10]的  $f20, f23, f24, f27$  和  $f43$ . 它们分为 2 组, 分别具有不同的函数特性. 第 1 组是高维单模函数, 记为  $U$ , 分别为  $F1, F3 \sim F4, F8 \sim F9, F18$ . 第 2 组为高维多模函数, 记为  $M$ , 分别为  $F2, F5 \sim F7, F10 \sim F17, F19 \sim F22$ . 第 1 组高维单模函数主要测试算法的收敛精度, 侧重考察优化算法局部收敛性能. 第 2 组高维多模函数主要测试算法的全局寻优能力. 该组测试函数的优化空间拥有大量局部极值点, 因而算法在有限的函数评价次数内搜索到全局最优值的难度极高. 通过测试这 2 组基准函数能够全面验证 VILOC 算法的先进性和有效性.

#### 5.2 VILOC 算法仿真结果

VILOC 算法设置粗搜索次数  $M_1 = 5$ , 细搜索次数  $M_2 = 15$ , 函数评价次数为  $F_c = 3 \times 10^4$ . 限于篇幅, 选取 30 维和 50 维基准测试函数  $F1 \sim F4, F6, F9, F12$  和  $F15$  独立运行 30 次, 其寻优函数适应度值  $F_{fit}$  的箱型对比结果如图 1 所示.

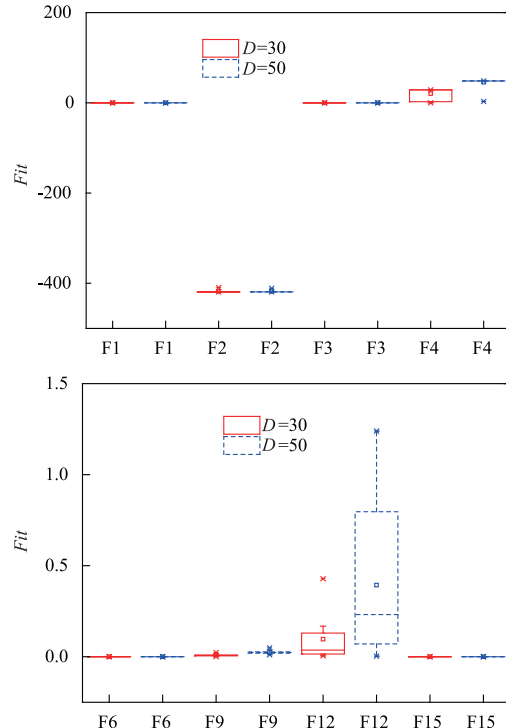


图1 30维和50维部分测试函数运行30次箱型对比

由图 1 可知, VILOC 算法对 30 维和 50 维基准函数

测试都能搜索到全局最优解,并且当空间维度增加时算法寻优值除了  $F4$  和  $F12$  以外基本保持不变,具有很好的稳定性.从图中示出  $F4$  的最优值波动较大,主要归因于  $F4$  的函数变化关系呈现特殊的马鞍形状.当搜索接近全局最优解时,函数适应度数值急剧变化,寻优难度极大.

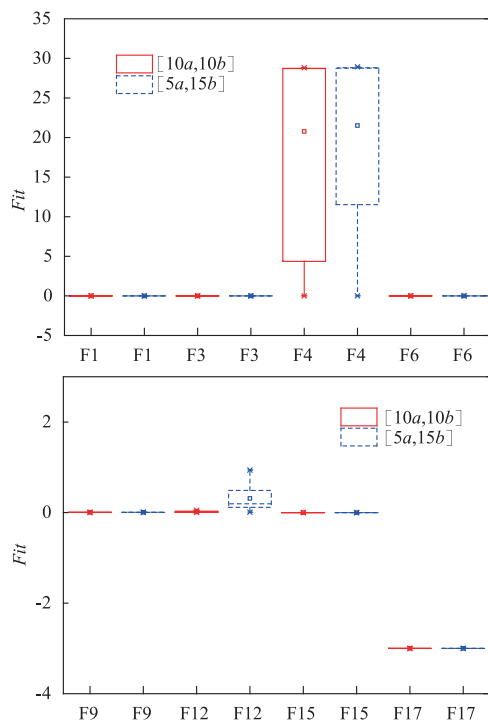


图2 优化空间增大时30维部分测试函数运行30次箱型对比

在实际工程应用中,优化问题往往不能精确获得变量的优化空间,因此优化算法需要具备大优化空间搜索能力.为了验证 VILOC 算法在大优化空间变化时算法的寻优性能,本文采用增大基准测试函数的变量区间长度并与原变量区间长度进行对比分析.为确保区间长度变化的多样性,将优化区间的变化划分为两组:一组为原变量区间的10倍,即 $[10a, 10b]$ ,另一组为原变量区间的下限扩大5倍、上限扩大15倍,即 $[5a, 15b]$ .图2为优化变量空间增大时部分30维测试函数独立运行30次寻优函数适应度值  $Fit$  的箱型对比结果.从图中示出,优化变量的区间变化时,VILOC 算法仍然能够具有较好的全局寻优能力,与优化变量区间未变化前的优化能力相当,验证了算法有效性和先进性.

### 5.3 与其他改进算法比较

为进一步验证 VILOC 算法的优化能力和该算法的可行性,将该算法与其他改进的混沌优化算法(CFA<sup>[23]</sup>、CTLA<sup>[24]</sup>)和当前主流优化算法(DE算法<sup>[25]</sup>、PSO<sup>[26]</sup>算法)进行对比.为保证比较公平,对每一种算法都取相同的函数评价次数进行优化.本节实验选取优化问题的测试函数维度为  $D = 50$ ,函数评价次数为  $F_c = 5 \times 10^4$ ,STD

为标准偏差,Mean 为优化均值.其他改进算法具体参数设置如下:CFA 算法的随机化参数  $\alpha = 0.2, \beta_0 = 1$ .DE 算法的交叉概率  $P_c = 0.5$ ,缩放因子  $F = 0.5$ .PSO 算法的学习因子  $c_1 = c_2 = 2$ ,惯性权重  $w \in [0.2, 0.9]$ .

在算法性能比较时,由于各个算法结构及参数不同,单纯的采用算法进化过程的迭代次数评价会存在局限性,因此本文采用函数评价次数作为衡量各种算法的标准,即在相同的函数评价次数时,评估各个算法优化效果.表1为5种优化算法50维性能比较结果.从表中看出,不论是寻优精度还是寻优稳定度,VILOC 算法综合性能最好,CTLA 性能次之,CFA 和 DE 算法有近似性能,仅在个别函数优化上有区别.VILOC 算法优于其他改进算法,证明了本文算法的先进性和有效性. $F2$  是一个具有大量局部最优的基准测试函数,传统优化算法很难处理.其他改进算法在优化该问题时效果均不理想.本文的 VILOC 算法具有较强的混沌搜索及反向寻优逃离局部陷阱,因而能够以较少的迭代次数搜索到全局最优,具体寻优过程如图3(b)所示.观察表1,除了  $F4, F12, F19$  和  $F20$  以外,VILOC 算法的标准方差都是最小的,这表明 VILOC 算法对于绝大部分基准测试函数的稳定性也是最好的.由图3所示,优化  $F1, F2, F5, F7, F10$ , 和  $F14$  时,VILOC 算法具有最佳的寻优精度,并且其收敛速度远远大于其他优化算法.虽然 CTLA 也能  $F5$  和  $F10$  搜索到全局最优以及 CFA 能在  $F14$  搜寻到最优值,但从收敛曲线看出,VILOC 算法的收敛速度要远大于 CTLA 和 CFA.在  $F8$  的优化过程中,CTLA 的表现最好,其收敛速度最快,明显优于其他4种算法.但是在  $F2$  的优化过程中,CTLA 的收敛速度和收敛精度弱于 PSO 算法、DE 算法、CFA 和 VILOC 算法.表明 CTLA 在优化包含大量局部最优解的问题中的寻优效率较低,全局搜索性能不足.CFA 在  $F11$  和  $F14$  中搜索速度和收敛精度最好,但在优化其他基准测试函数时,该算法的优化均值和标准方差弱于 VILOC 算法和 CTLA,搜索后期算法搜索能力不足.VILOC 算法在  $F8$  和  $F11$  上虽然获得较好解,但却未能提高解的质量.VILOC 算法采用区间长度可变的反向优化策略和反向混沌优化策略,利用当前最优解的反向解增大算法进化过程的多样性,通过不断细化优化变量的区间长度,以达到快速逼近全局最优解的目的.而算法在  $F8$  和  $F11$  上未能改善优化解的质量,主要原因是  $F8$  和  $F11$  的全局最优解在初始化搜索空间之外,导致区间长度可变的反向优化策略和反向混沌优化策略无法引导算法进行有效的局部搜索,因此影响优化解的质量.虽然 VILOC 算法在  $F8$  和  $F11$  上收敛能力分别弱于 CFA 和 CTLA,但在其他测试函数中它的求解能力还是最好的,在收敛速度和精度上都获得了最佳结果.

表 1 5 种优化算法 50 维性能比较

$F$		CFA	CTLA	DE	PSO	VILOC
$F_1$	Mean	1.53E-05	3.94E-85	2.90E-06	2.53E-04	0
	STD	1.41E-06	2.75E-85	3.73E-06	3.03E-04	0
$F_2$	Mean	-2.09E+02	-1.78E+02	-1.93E+02	-2.48E+02	-418.0743
	STD	2.10E+01	4.81E+01	3.01E+01	2.43E+01	1.1120
$F_3$	Mean	7.75E-04	4.44E-15	2.79E-01	3.07E-01	8.8818E-016
	STD	1.85E-05	0	4.50E-01	6.20E-01	0
$F_4$	Mean	1.21E+02	4.60E+01	1.82E+02	1.35E+02	21.4321
	STD	1.43E+02	1.00E+00	9.37E+01	5.67E+01	10.3254
$F_5$	Mean	6.39E+01	0	3.29E+02	8.13E+01	0
	STD	1.39E+01	0	3.21E+01	1.75E+01	0
$F_6$	Mean	1.01E-03	0	1.18E-02	9.55E-03	0
	STD	3.12E-03	0	1.96E-02	1.13E-02	0
$F_7$	Mean	3.94E-02	1.89E-03	1.11E-01	1.40E-01	1.0934E-005
	STD	1.32E-02	5.22E-04	5.17E-02	3.43E-02	1.0020E-005
$F_8$	Mean	1.93E+02	1.72E-34	3.60E+02	3.55E+01	1.5274E-015
	STD	6.17E+01	2.31E-34	8.20E+01	9.52E+00	2.2411E-013
$F_9$	Mean	1.10E+00	0	1.17E+01	9.33E-01	7.9390E-003
	STD	9.94E-01	0	2.06E+01	1.08E+00	4.1734E-003
$F_{10}$	Mean	-4.87E+01	-49	-3.15E+01	-4.88E+01	-49
	STD	9.85E-01	0	6.36E+00	7.90E-01	0
$F_{11}$	Mean	2.54E-08	1.50E-02	1.19E-01	7.77E-02	5.5450E-004
	STD	3.20E-09	1.60E-02	3.14E-01	2.56E-01	2.1697E-004
$F_{12}$	Mean	1.01E-06	4.92	1.10E-03	8.81E-03	6.1471E-002
	STD	1.06E-06	1.75E-03	3.47E-03	1.94E-02	7.1508E-002
$F_{13}$	Mean	-67.1360	-67.1468	-74.0342	-66.7000	-6.8138E+001
	STD	1.76	1.67	1.36	2.01	1.5472E+000
$F_{14}$	Mean	2.8325E-2	1.05E+04	3.71E+03	0.31	2.7133E-2
	STD	3.3521E-2	1.71E+04	9.03E+03	0.52	5.1247E-2
$F_{15}$	Mean	5.34E-21	0	4.37E-20	5.43E-18	0
	STD	1.14E-21	0	1.28E-19	2.24E-17	0
$F_{16}$	Mean	7.87E-02	5.37E-44	1.21E-03	5.94E-04	0
	STD	7.13E-02	1.78E-44	1.19E-03	6.26E-04	0
$F_{17}$	Mean	-4.7042	-5	-4.38	-4.0467	-5
	STD	2.87E-01	0	3.66E-01	2.73E-01	0
$F_{18}$	Mean	2.28E-03	4.88E-43	3.03E-04	1.99E-03	0
	STD	7.75E-05	3.04E-43	2.51E-04	2.35E-03	0
$F_{19}$	Mean	0	-1.65E-026	0	-2.13E-01	-3.8234E-001
	STD	0	5.22E-26	0	4.30E-01	5.2350E-001
$F_{20}$	Mean	1.09	6.67E-01	4.39	4.44	4.9448E-001
	STD	6.46E-01	1.79E-10	3.55	3.39	1.9312E-002
$F_{21}$	Mean	2.71E-132	0	0	2.11E-87	0
	STD	5.72E-132	0	0	1.16E-86	0
$F_{22}$	Mean	-3.35	-47.84	-1.14E-02	-2.53E-01	-49
	STD	1.57	2.68	1.35E-03	3.05E-02	0

综上所述,对比其他改进算法,VILOC 算法收敛速度明显优于其他算法,求解能力较高,表现了最好的搜索性能。

## 6 结论

本文提出的 VILOC 算法解决了现有的优化算法在大空间和高维度优化问题的寻优效率较低的问题,并用数学理论证明了该算法以概率 1 收敛于全局最优

解. 该算法利用无限折叠 Fuch 混沌映射增大随机搜索能力、两级优化策略增大算法寻优精度、反向优化策略增大逃逸局部极值的能力以及区间长度可变的反向优化策略减小优化空间. 随着迭代次数的增加,优化变量的区间不断减小,算法不断趋向于最优解方向,最终实现全局优化. 利用 24 个基准函数进行数值实验,测试结果表明,VILOC 算法可以显著减少优化过程中函数迭

代次数,具有搜索速度快、稳定性高和易于实现的特

点,证明了该算法的工程实用性和高效性.

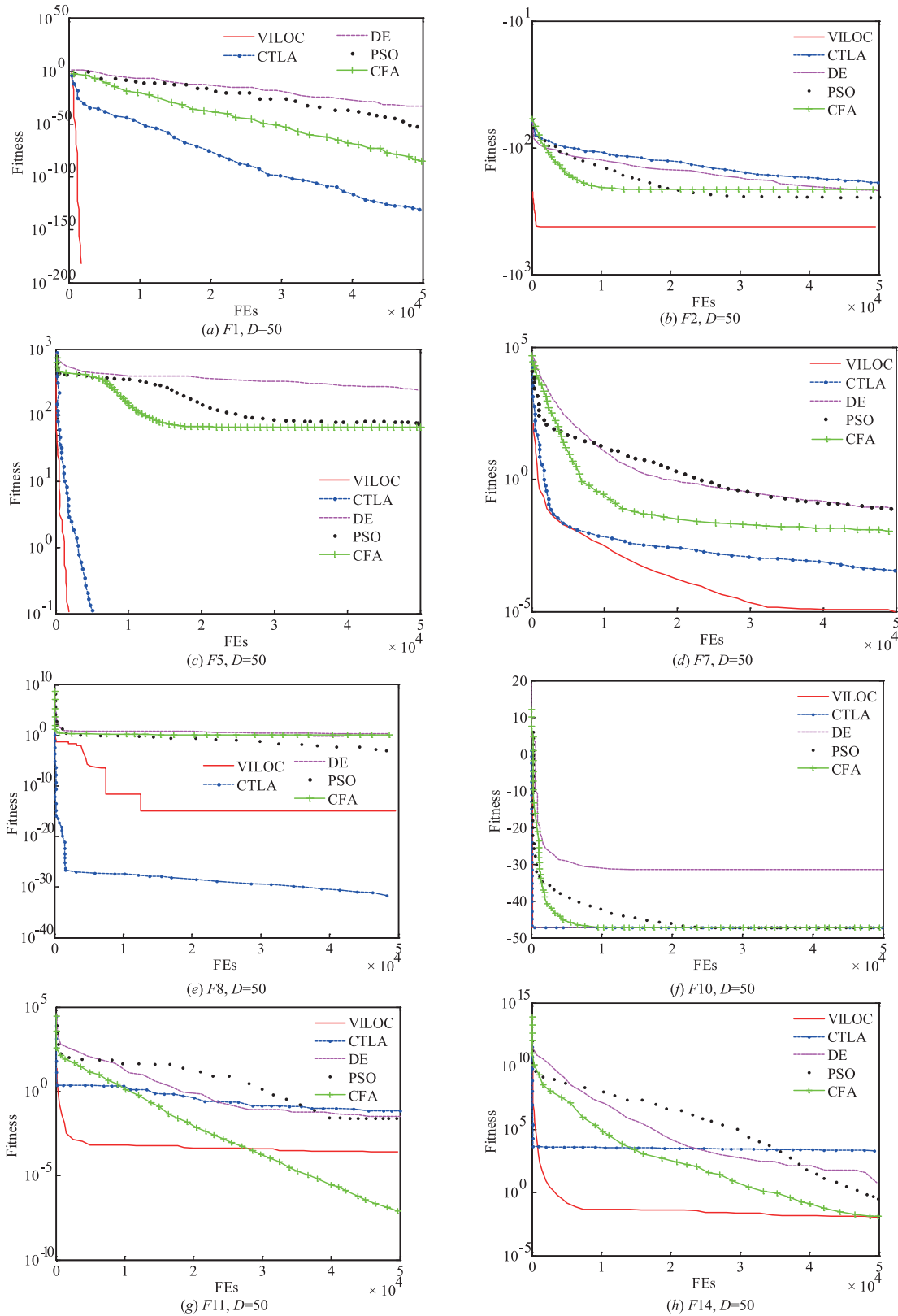


图3 VILOC与其他改进算法在部分函数的适应度值收敛特性

## 参考文献

- [1] Kumar D, Budman H. Applications of polynomial chaos expansions in optimization and control of bioreactors based on dynamic metabolic flux balance models[J]. *Chemical Engineering Science*, 2017, 167(10): 18–28.
- [2] Günther S, Gauger N R, Wang Q. A framework for simultaneous aerodynamic design optimization in the presence of chaos[J]. *Journal of Computational Physics*, 2016, 328(1): 387–398.
- [3] Zhang Y, Zhao Y, Fu X, et al. A feature extraction method of the particle swarm optimization algorithm based on adaptive inertia weight and chaos optimization for Brillouin scattering spectra[J]. *Optics Communications*, 2016, 376(1): 56–66.
- [4] Sengül Dogan. A new data hiding method based on chaos embedded genetic algorithm for color image[J]. *Artificial Intelligence Review*, 2016, 46(1): 129–143.
- [5] Keshtegar B. Stability iterative method for structural reliability analysis using a chaotic conjugate map[J]. *Nonlinear Dynamics*, 2016, 84(4): 2161–2174.
- [6] Wang Y, Jin S, Penugonda S, et al. Variability analysis of crosstalk among differential vias using polynomial chaos and response surface methods[J]. *IEEE Transactions on Electromagnetic Compatibility*, 2017, 59(4): 1–11.
- [7] Arora J S, Elwakeil O A, Chahande A I, et al. Global optimization methods for engineering applications: A review[J]. *Structural & Multidisciplinary Optimization*, 1995, 9(3): 137–159.
- [8] Saha S, Mukherjee V. A novel chaos-integrated symbiotic organisms search algorithm for global optimization[J]. *Soft Computing*, 2017, 13(4): 1–20.
- [9] Lu Z, Shieh L S, Chen G R. On robust control of uncertain chaotic systems: a sliding-mode synthesis via chaotic optimization[J]. *Chaos Solitons & Fractals*, 2003, 18(4): 819–827.
- [10] Aslimani N, Ellaia R. A new hybrid algorithm combining a new chaos optimization approach with gradient descent for high dimensional optimization problems[J]. *Computational & Applied Mathematics*, 2017, 4: 1–29.
- [11] Hamaizia T, Hamri N E. Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor[J]. *Applied Mathematics & Computation*, 2012, 219(1): 188–196.
- [12] Yang D X, Li G, Cheng G D. On the efficiency of chaos optimization algorithms for global optimization[J]. *Chaos Solitons & Fractals*, 2007, 34(5): 1366–1375.
- [13] Senkerik R, Davendra D, Zelinka I, et al. Chaos driven differential evolution with Lozi map in the task of chemical reactor optimization[A]. *International Conference on Artificial Intelligence and Soft Computing [C]*. Berlin Heidelberg: Springer, 2013. 56–66.
- [14] Cheng M Y, Huang K Y. Genetic algorithm-based chaos clustering approach for nonlinear optimization[J]. *Journal of Marine Science & Technology Taiwan*, 2010, 18(3): 435–441.
- [15] Liu B, Wang L, Jin Y H, et al. Improved particle swarm optimization combined with chaos[J]. *Chaos Solitons & Fractals*, 2005, 25(5): 1261–1271.
- [16] Xiang T, Liao X, Wong K W. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map[J]. *Applied Mathematics & Computation*, 2007, 190(2): 1637–1645.
- [17] Gudla P K, Ganguli R. An automated hybrid genetic-conjugate gradient algorithm for multimodal optimization problems[J]. *Applied Mathematics & Computation*, 2005, 167(2): 1457–1474.
- [18] Moumen S E, Ellaia R, Aboulaich R. A new hybrid method for solving global optimization problem[J]. *Applied Mathematics & Computation*, 2011, 218(7): 3265–3276.
- [19] Barradell T. The chaos theory of careers[J]. *Journal of Employment Counseling*, 2011, 48(4): 163–166.
- [20] 傅文渊, 凌朝东. 自适应折叠混沌优化方法[J]. *西安交通大学学报(自然科学版)*, 2013, 47(2): 33–38.
- Fu W Y, Ling C D. An adaptive iterative chaos optimization method[J]. *Journal of Xi'an Jiaotong University*, 2013, 47(2): 33–38. (in Chinese)
- [21] Keener R W. Probability and measure[J]. *Journal of the American Statistical Association*, 1979, 91(433): 1–24.
- [22] Solis F J, Wets R J B. Minimization by random search techniques[J]. *Mathematics of Operations Research*, 1981, 6(1): 19–30.
- [23] Su S, Su Y, Xu M. Comparisons of firefly algorithm with chaotic maps[J]. *Computer Modeling & New Technologies*, 2014, 18(12): 326–332.
- [24] Farah A, Guesmi T, Abdallah H H, et al. A novel chaotic teaching-learning-based optimization algorithm for multi-machine power system stabilizers design problem[J]. *Electr Power Energy Syst*, 2016, 77: 197–209.
- [25] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 398–417.
- [26] Tanweer M R, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm[J]. *Information Sciences*, 2015, 294(10): 182–202.

## 作者简介



**傅文渊** 男,1982年9月出生,福建邵武人,讲师,主要从事智能信号优化与智能学习控制、电路与系统设计及嵌入式系统设计等方面的研究工作.

E-mail:fwy@hqu.edu.cn



**李国刚** 男,1973年12月出生,福建邵武人,博士,副教授.主要从事信息安全及其应用,电路与系统设计等方面的研究工作.

E-mail:lgg@hqu.edu.cn



**王燕琼** 女,1973年1月出生,福建泉州人,实验师.主要从事单片机及其应用,EDA设计等方面的研究工作.