

噪声环境下基于蒲丰距离的依概率多峰优化算法

王霞^{1,2} 王耀民¹ 施心陵¹ 高莲¹ 李鹏¹

摘要 针对噪声环境下求解多个极值点的问题, 本文提出了噪声环境下基于蒲丰距离的依概率多峰优化算法 (Probabilistic multimodal optimization algorithm based on the Buffon distance, PMB). 算法依据蒲丰投针原理提出噪声下的蒲丰距离和极值分辨率概念, 理论推导证明了二者与算法峰值检测率符合依概率关系. 在全局范围内依据蒲丰距离划分搜索空间, 可以使 PMB 算法保持较好的搜索多样性. 在局部范围内利用改进的斐波那契法进行探索, 减少了算法陷入噪声引起的局部最优的概率. 基于 34 个测试函数, 从依概率特性验证、寻优结果影响因素分析、多极值点寻优和多维函数寻优四个角度进行实验. 证明了蒲丰距离与算法的峰值检测率符合所推导的依概率关系. 对比噪声环境下的改进蝙蝠算法和粒子群算法, PMB 算法在噪声环境中可以依定概率更精确地定位多峰函数的更多极值点, 从而证明了 PMB 算法原理的正确性和噪声条件下全局寻优的依概率性能, 具有理论意义和实用价值.

关键词 噪声环境, 多峰优化, 蒲丰距离, 依概率

引用格式 王霞, 王耀民, 施心陵, 高莲, 李鹏. 噪声环境下基于蒲丰距离的依概率多峰优化算法. 自动化学报, 2021, 47(11): 2691-2714

DOI 10.16383/j.aas.c190474

Probabilistic Multimodal Optimization Algorithm Based on the Buffon Distance in Noisy Environment

WANG Xia^{1,2} WANG Yao-Min¹ SHI Xin-Ling¹ GAO Lian¹ LI Peng¹

Abstract To solve the problem of multiple extremum points optimization in noisy environment, a probabilistic multimodal optimization algorithm based on the Buffon distance (PMB) is proposed in this paper. Based on the principle of Buffon needles, the concepts of Buffon distance and resolution of extreme value under noisy environment are put forward. The theoretical derivation proves that the relationship between peak detection rate of PMB algorithm and Buffon distance conforms to a probabilistic relation. In global scope, the search space is divided by Buffon distance for diversity maintenance. The improved Fibonacci method is used in local search to reduce the probability of falling into the local optimum caused by noise. Based on 34 test functions, experiments are carried out from four aspects, including probabilistic property verification, analysis of influencing factors of optimization results, multiple extremum points optimization and multidimensional optimization. It is proved that the relationship of Buffon distance and peak detection rate of PMB algorithm is in line with the deduced probabilistic relationship. Compared with the improved bat algorithm and particle swarm optimization (PSO), PMB algorithm can locate extremum points of multimodal function by a steady probability in noise environment, and gain more extremum points accurately. Thus, the correctness of PMB algorithm principle and probabilistic performance of global optimization under noise condition are proved, which has theoretical and practical significance.

Key words Noisy environment, multimodal optimization, Buffon distance, probabilistic

Citation Wang Xia, Wang Yao-Min, Shi Xin-Ling, Gao Lian, Li Peng. Probabilistic multimodal optimization algorithm based on the Buffon distance in noisy environment. *Acta Automatica Sinica*, 2021, 47(11): 2691-2714

收稿日期 2019-07-01 录用日期 2019-11-16

Manuscript received July 1, 2019; accepted November 16, 2019

国家自然科学基金 (61763046, 61762093, 61662089, 61761048), 云南省第十七批中青年学术和技术带头人资助项目 (2014HB019), 云南省高校科技创新团队支持计划资助, 云南省教育厅科学研究基金 (2017ZZX229), 云南省应用基础研究重点项目 (2018FA036) 资助

Supported by National Natural Science Foundation of China (61763046, 61762093, 61662089, 61761048), The 17th Batches of Young and Middle-aged Leaders in Academic and Technical Reserved Talents Project of Yunnan Province (2014HB019), The Program for Innovative Research Team (in Science and Technology) in University of Yunnan Province, Yunnan Provincial Department of Education Science Research Fund Project (2017ZZX229),

实际优化问题往往存在着多个全局最优解及其他有价值的局部最优解, 我们不仅需要找到全局最优解, 还需要找到局部最优解进行辅助决策, 这类

and Key Projects of Applied Basic Research of Yunnan Province (2018FA036)

本文责任编辑 穆朝絮

Recommended by Associate Editor MU Chao-Xu

1. 云南大学信息学院 昆明 650500 2. 云南民族大学电气信息工程学院 昆明 650500

1. School of Information Science and Engineering, Yunnan University, Kunming 650500 2. School of Electrical and Information Engineering, Yunnan Minzu University, Kunming 650500

优化问题称为多峰搜索 (Multi-peak searching) 或多峰优化 (Multimodal optimization) 问题^[1]. 此外, 实际优化问题总是受到各种随机噪声的影响. 在噪声环境下求解多峰优化问题不仅要求算法能在噪声干扰下寻找真实的极值点位置, 而且希望算法能提供备选的多个优化方案. 噪声的分布往往符合一定的概率关系, 因此噪声环境下的多峰优化问题是一个依概率求解极值点的问题. 其中包含的噪声环境下的寻优问题^[2] (Noisy optimization) 和多峰优化问题在科学和工程领域中有理论和实际意义. 噪声优化算法的研究可以为机器人^[3-5]、智能控制与决策^[6-9]、信号识别与测量^[10-11]、智能制造^[12] 等存在噪声干扰的工程优化问题提供优化解决方案.

在噪声优化算法的依概率问题研究方面, 一些学者研究了噪声优化算法的概率模型, 如 Nakama^[13] 和李军华等^[14] 研究了噪声环境下遗传算法的 Markov 模型. Ma 等^[15] 使用 Markov 模型分析了噪声对生物地理优化算法 (Biogeography-based optimization, BBO) 的影响. Beyer 等^[16] 对自适应进化策略 (Self-adaptive evolution strategy, ES) 的鲁棒优化模型进行了稳态分析. 但是算法求解优化结果的依概率关系尚无相关研究. 也有一些学者在研究中涉及到了优化算法的依概率问题, 但都是针对无噪声环境下的优化算法, 例如蚁群优化算法^[17] (Ant colony optimization, ACO) 基于信息素指导依概率为子代个体构建解, 从而将信息素的分布理解为一种特殊的概率分布. 李宝磊等^[18] 证明了所提出的多元优化算法以概率 1 收敛于全局最优解. 董易等^[19] 证明了所提出的斐波那契树优化算法以概率 1 收敛于全局最优解. 针对噪声环境下多峰优化算法的依概率问题尚缺乏相关研究. 对于噪声优化算法的依概率特性研究非常重要, 它可以评估算法搜索优化问题全部极值点的能力, 衡量算法的优化效率, 同时为算法性能的改进和提升指明方向.

在噪声环境下多极值点寻优问题的研究方面, 元启发式算法作为解决确定性优化问题的一种主流方法, 其在噪声环境中的优化性能受到了越来越多的关注. 李军华、黎明等^[14, 20-22] 分析了噪声对遗传算法的影响以及正态随机噪声对适应度评价的影响机理, 提出了多次评价一次采样的动态适应度评价方法. 许多学者通过将假设检验 (Hypothesis test, HT)、最优计算量分配 (Optimal computing budget allocation, OCBA) 技术、反向学习法、自动学习机 (Learning automata, LAs)、重采样、聚类算法、加权搜索中心等多种策略引入粒子群优化算法 (Particle swarm optimization, PSO) 中^[23-29], 以提高算

法的鲁棒性. 这些优化算法能够在噪声环境中找到接近全局最优的极值点, 但都没有求解其他极值点. 在多峰优化问题的研究中, 许多多峰函数优化算法^[1, 30-31] 能搜索到多峰函数的多个极值点. 但大部分算法只针对确定性环境, 并未考虑噪声环境下的多峰优化问题. Jamil 等^[32] 对蝙蝠算法 (Bat algorithm) 进行了改进, 使其适用于无噪声环境和有噪声环境, 将其应用到多峰函数优化问题中, 可以找到多峰函数的多个全局极值点. 然而遗憾的是, 文中并未讨论多个局部极值点的情况, 且极值点的坐标并未明确给出. 尽可能多的搜索局部极值点能为生产实际提供更多样化的优化方案. 此外, 为了提供明确的优化方案, 实验结果应给出极值点的适应度值和对应的解, 以便为实际应用提供明确的决策变量^[33].

蒲丰投针问题是一个几何概率问题. 它是法国科学家蒲丰 (Buffon) 在 1777 年的论著《或然性算术试验》中提出的. 蒲丰投针原理反映了几何概率的特征及处理方法, 衍生出了蒙特卡洛 (Monte-Carlo) 方法. 其概率规律在探矿、近似计算中得到广泛应用, 但在优化算法中的应用尚无相关研究. 在噪声环境中通过随机方式探索多峰函数的极值点与蒲丰投针问题具有相似的概率解释. 本文将蒲丰推导得到的定概率原理应用于噪声环境下的多极值点寻优问题, 从概率的角度出发, 提出了噪声环境下基于蒲丰距离的依概率多峰优化算法 (Probabilistic multimodal optimization algorithm based on the Button distance, PMB), 推导证明了蒲丰距离和极值分辨率共同依概率决定算法的峰值检测率. 依据蒲丰距离划分全局搜索空间, 并在局部范围内利用改进的斐波那契法进行局部寻优, 针对多维函数, 采取多级划分策略, 从而提高求解精度和算法效率. 通过算法依概率收敛特性实验, 验证了 PMB 算法在噪声环境下能够依固定概率找到多个极值点; 通过设置不同的噪声强度与蒲丰距离进行实验, 分析了噪声强度与蒲丰距离对算法求解结果的影响; 通过对比噪声环境下的改进蝙蝠算法, 表明了 PMB 算法具有很好的多极值点寻优特性和定位精准性; 通过对比 PSO 算法在多维函数上的表现, 证明了 PMB 算法在噪声环境下对多维函数寻优的有效性.

第 1 节对噪声环境下的多峰优化问题进行数学描述, 并说明评价指标. 第 2 节详细描述了本文提出的 PMB 算法的基本原理和理论推导. 第 3 节是论文的仿真实验与结果分析, 用以验证算法的有效性. 第 4 节给出结论和将来的研究方向.

1 问题描述及评价指标

1.1 问题描述

在受到加性噪声干扰的情况下, 多峰函数的适应度值可描述为:

$$f_{\sigma}(\mathbf{x}) = f(\mathbf{x}) + \xi \quad (1)$$

式中, $f_{\sigma}(\mathbf{x})$ 是受到噪声干扰后的函数值, $f(\mathbf{x})$ 是未受噪声干扰的函数值. 设 \mathbf{R} 为实数集, $S \subseteq \mathbf{R}$ 称为搜索空间, 即变量的可行解范围, $\mathbf{x} = (x_1, x_2, \dots, x_d) \in S$ 为 d 维实变量. $x_i \in [a_i, b_i], i = 1, 2, \dots, d$, 其中 a_i 和 b_i 分别是第 i 维变量的取值上边界和下边界. 由于高斯型噪声在自然和工程系统中普遍存在^[34], 因此考虑 ξ 为服从高斯分布的随机变量, $\xi \sim N(\mu, \sigma^2)$, 噪声的强度由方差 σ^2 决定. 以极小值优化为例, 设 $\mathbf{x}^* \in S$, 若存在 $\delta > 0$ 为任意独立的无穷小值, 使得

$$f(\mathbf{x}^*) = \min_{\mathbf{x} \in S \text{ 且 } \|\mathbf{x} - \mathbf{x}^*\| < \delta} f(\mathbf{x}) \quad (2)$$

则称 \mathbf{x}^* 为 $f(\mathbf{x})$ 在 S 上的局部最优解, $f(\mathbf{x}^*)$ 为局部最优值. 噪声环境下多峰函数优化问题可描述为: 在一定强度的噪声干扰下, 根据受噪声干扰的函数值 $f_{\sigma}(\mathbf{x})$, 尽可能多的定位 $f(\mathbf{x})$ 的多个极值点, 即找到 $f(\mathbf{x})$ 的多个最优值及其最优解.

为全文描述方便, 给出极值深井和极值分辨率的定义:

假设 d 维函数 $f(\mathbf{x})$ 包含 m 个极小值点 ($m \geq 1$), 其中第 i 个极小值点为 $M_i(\mathbf{x}_{M_i}, f(\mathbf{x}_{M_i}))$, 其中, $\mathbf{x}_{M_i} = (x_{M_i1}, x_{M_i2}, \dots, x_{M_id})$. 在第 k 维坐标上, 与 M_i 左右相邻的两个极大值点分别为 $B_k(\mathbf{x}_B, f(\mathbf{x}_B))$ 和 $C_k(\mathbf{x}_C, f(\mathbf{x}_C))$. B_k 和 C_k 的第 k 维坐标分别为 x_{Bk} 和 x_{Ck} .

定义 1. 在第 k 维坐标上, 由 $B_k(\mathbf{x}_B, f(\mathbf{x}_B))$ 和 $C_k(\mathbf{x}_C, f(\mathbf{x}_C))$ 及函数 $f(\mathbf{x})$ 在该维平面上投影的曲线所确定的范围形如开口向上的深井, 称此范围为该极值点在第 k 维坐标平面上的极值深井.

定义 2. $B_k(\mathbf{x}_B, f(\mathbf{x}_B))$ 和 $C_k(\mathbf{x}_C, f(\mathbf{x}_C))$ 的第 k 维坐标之差的绝对值记为 $\lambda_{M_i k} = |x_{Bk} - x_{Ck}|$, 称为极小值点 M_i 在第 k 维坐标平面上的极值深井的井口宽度. $\lambda_{M_i} = \min_{1 < k < d} \lambda_{M_i k}$ 为极小值点 M_i 在各维坐标平面上的极值深井的最小井口宽度, 称为该极小值点的极值分辨率. 则 m 个极值点中任意极小值点的极值分辨率的最小值

$$\lambda_{\min} = \min_{1 \leq i \leq m} \lambda_{M_i} \quad (3)$$

称为该函数的极值分辨率.

以一维函数为例, 如图 1 所示, 极值点 M_1 的极

值分辨率 λ_{M_1} 小于极值点 M_2 的极值分辨率 λ_{M_2} . 当决策变量受到外界不确定因素产生的噪声影响时, 函数适应度值总是不可避免地产生波动, 如果产生的波动较小, 则认为解的抗噪声性能较强, 如果目标函数波动的很厉害则说明解的抗噪声性能较差. 从图 1 可以看出点 M_1 和 M_2 都是函数的局部极小值点, 其函数适应度值相差不大. 假设决策变量受到干扰产生相同程度偏移, 点 M_2 的适应度值的变化将比点 M_1 的变化更小, 从而点 M_2 的抗噪声性能优于点 M_1 .

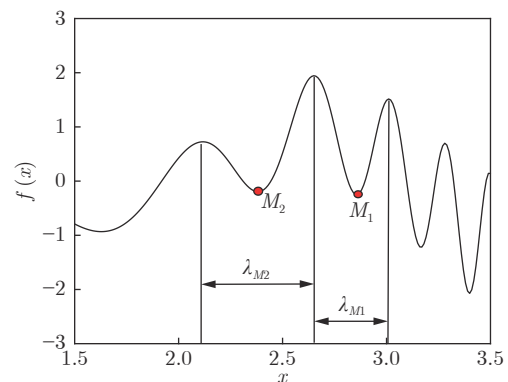


图 1 极值分辨率

Fig.1 Extreme value resolution

从上面的例子和分析可以看出, 极值分辨率越低, 即该极值点所在深井的井口宽度越小, 则该极值点的抗噪声性能越差. 工程应用时要根据实际要求的误差允许范围确定极值分辨率. 例如, 误差允许范围是 0.05, 若极值分辨率小于 0.05, 通过优化算法将找出井口宽度小于 0.05 的极值点, 这样的点在受到噪声干扰时, 若决策变量的偏移量达到 0.05, 虽然没有超过误差允许范围, 但适应度值已经由极小值偏移到了极大值, 偏移程度较严重, 极有可能对工程设计造成不良后果. 因此, 选取的极值分辨率需大于误差允许范围.

1.2 评价指标

优化问题是工程领域普遍存在且需要解决的问题^[35], 涉及到的研究领域繁多, 除了优化方法的依概率问题、噪声优化问题和多峰优化问题, 还包括全局优化问题^[36-37]、局部收敛精度提高问题^[37]、高维优化问题^[38]、算法收敛速度问题^[39]、多目标优化问题^[40-41]、多目标鲁棒优化问题^[42]等. 针对不同的研究领域, 都有相应的评价指标. 本文主要研究噪声环境下的多峰优化问题, 在这类问题的优化求解中, 不仅要尽可能多地求出优化问题的全局极值点和局部极值点, 而且要求出每个极值点的 d 维变量取值,

因此定义噪声环境下的峰值检测率、峰值误差和定位误差,并将它们作为评价噪声环境下算法性能指标。

定义 3. 假设 d 维函数 $f(\mathbf{x})$ 包含 m 个极值点 ($m \geq 1$), 算法求得 m^* 个极值点, 则

$$\eta = \frac{m^*}{m} \times 100\% \quad (4)$$

称为算法求解极值点的峰值检测率。

定义 4. 假设算法求得的噪声环境下的极值点适应度值为 $f_\sigma(\mathbf{x}_\sigma^*)$, 极值点坐标为 $\mathbf{x}_\sigma^* = (x_{\sigma 1}^*, x_{\sigma 2}^*, \dots, x_{\sigma d}^*)$, 无噪声情况下的极值点适应度值为 $f(\mathbf{x}^*)$, 极值点坐标为 $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_d^*)$, 则

$$P_f = |f_\sigma(\mathbf{x}_\sigma^*) - f(\mathbf{x}^*)| \quad (5)$$

称为该算法求解极值点的峰值误差。

$$P_l = |\mathbf{x}_\sigma^* - \mathbf{x}^*| = \sqrt{\sum_{i=1}^d (x_{\sigma i}^* - x_i^*)^2} \quad (6)$$

称为该算法求解极值点的定位误差。

2 噪声环境下基于蒲丰距离的依概率多峰优化算法

2.1 算法基本原理

在噪声环境下的多峰优化求解问题中, 噪声干扰使得函数波形发生波动, 由此引入若干局部极值点, 如图 2 中的实心点。噪声环境下多峰优化的目的是要找到函数的真实极值点位置, 即图 2 中的空心点。为此, 可以根据工程需求对搜索空间进行合理均匀划分, 在划分的多个局部区域内进行单极值点搜索, 从而保留下多个极值点, 最后利用合适的同峰判别方法区分异峰极值点, 合并同峰极值点。

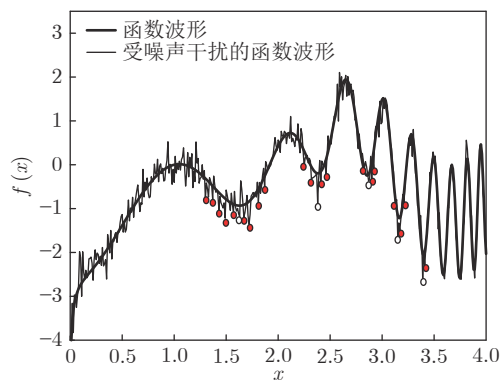


图 2 函数波形与极值点

Fig. 2 Function waveform and extremum points

在此过程中, 需要解决两个问题: 1) 划分搜索空间的依据; 2) 局部区域搜索时如何尽可能地避开实心点 (噪声引起的局部最优) 的干扰。

本文从概率的角度出发, 将蒲丰投针问题中的几何概率规律引入解空间划分问题, 提出了噪声环境下基于蒲丰距离的依概率多峰优化算法。首先基于蒲丰投针原理提出蒲丰距离, 推导并证明了蒲丰距离与峰值检测率的关系。在此基础上, 依据蒲丰距离对全局区域进行探索, 得到多个探索点。在多个探索点之间进行局部寻优, 可以保留下多个极值点。其次, 在局部寻优阶段, 为减小算法陷入噪声引起的局部最优的发生概率, 利用区域伸缩准则改进斐波那契法, 提高了算法在真实最优解邻域附近的收敛能力, 削弱了噪声的影响。

2.2 蒲丰投针原理与蒲丰距离

18 世纪, 法国数学家蒲丰 (Buffon, 1707~1788) 提出的“投针问题”, 记载于蒲丰 1777 年出版的著作中: “在平面上画有一组间距为 a 的平行线, 将一根长度为 l ($l \leq a$) 的针任意掷在这个平面上, 求此针与平行线中任一条相交的概率。”如图 3 所示。蒲丰最早设计了投针试验。这一方法的步骤是:

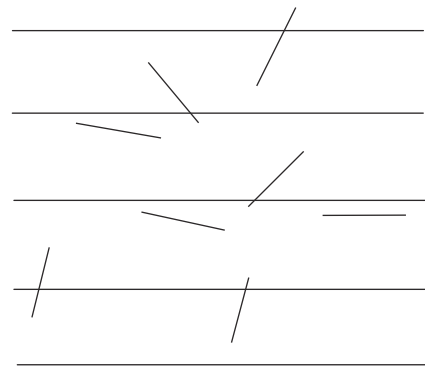


图 3 蒲丰投针实验设计图

Fig. 3 Experimental design of Buffon needles

- 1) 在白纸上画数条间距为 a 的平行线。
- 2) 取长度为 l ($l \leq a$) 的针, 随机地向画有平行直线的纸上掷 n 次, 观察针与直线相交的次数。
- 3) 计算针与直线相交的概率。

由于向桌面投针是随机的, 所以用二维随机变量 (X, Φ) 来确定它在桌上的具体位置。设 X 表示针的中点到平行线的距离, Φ 表示针与平行线的夹角, 则当 $X < l/2 \sin \Phi$ 时, 针与直线相交。且 X 在 $(0, a)$ 范围内服从均匀分布, Φ 在 $(0, \pi/2)$ 范围内服从均匀分布, X 和 Φ 相互独立, 由此可以写出 (X, Φ) 的概率密度函数

$$f(x, \varphi) = \begin{cases} \frac{4}{\pi a}, & 0 < x < \frac{a}{2}, 0 < \varphi < \frac{\pi}{2} \\ 0, & \text{其他} \end{cases} \quad (7)$$

则针与直线相交的概率为

$$P \left\{ X < \frac{l}{2} \sin \Phi \right\} = \iint_{X < \frac{l}{2} \sin \Phi} f(x, \varphi) dx d\varphi = \int_0^{\frac{\pi}{2}} \int_0^{\frac{l}{2} \sin \phi} \frac{4}{\pi a} dx d\varphi = \frac{2l}{\pi a} \quad (8)$$

若对多峰函数的求解空间进行等间隔采样, 每对相邻的采样点可以确定一个局部范围. 若采样间隔设置合理, 使得在此局部范围内函数只包含唯一一个局部极值点, 则在此范围内进行单极值寻优即可找出函数的局部极值点. 通过采样得到的多对相邻采样点, 可以找到函数的多个局部极值点. 为实现等间隔采样, 将求解域在每一维上以固定长度划分, 划分的等间隔平行线与函数的若干个交点即为采样点, 相邻采样点在第 k ($1 < k < d$) 维上的距离看作蒲丰投针问题中的随机扔置的针, 而等间隔的采样平行线看作蒲丰投针问题中的间距为 a 的平行线, 则有如下定理.

定理 1. 设 a 为采样平行线间距, 以函数的极值分辨度为固定针长 l , 若 $2a \geq l \geq a$, 则针同时与两条平行线相交的概率为

$$P_1 = \frac{l-a}{a} \quad (9)$$

若 $l \leq a$, 则针同时与两条平行线相交的概率为 $P_2 = 0$.

证明. 以二维平面为例, 用二维随机变量 (X, Φ) 来确定针在平面上的具体位置. 设 X 表示针的端到平行线的距离, 如图 4 所示. X 在 $(0, a)$ 范围内服从均匀分布, 由此可以写出 X 的概率密度函数

$$f(x) = \begin{cases} \frac{1}{a}, & 0 < x < a \\ 0, & \text{其他} \end{cases} \quad (10)$$

Φ 表示针与平行线的夹角, 根据极值分辨度的定义, 可知针与平行线始终为垂直方向, $\Phi = \pi/2$, 故当 $0 < X < (l-a)$ 时, 针同时与两条直线相交, 其概率为

$$P_1 = P\{0 < X < (l-a)\} = \int_0^{l-a} f(x) dx = \frac{l-a}{a} \quad (11)$$

若 $l \leq a$, 则针同时与两条平行线相交的概率为 $P_2 = 0$. \square

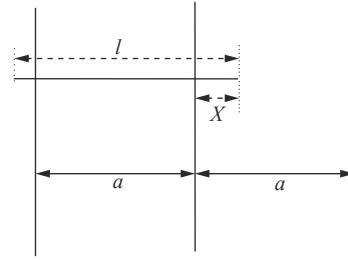


图 4 式 (9) 证明示意图
Fig.4 Diagram to prove (9)

定理 2. 设 a 为采样平行线间距, 以函数的极值分辨度为固定针长 l , 若 $2a \geq l \geq a$, 则针只与一条平行线相交的概率为

$$P_3 = \frac{2a-l}{a} \quad (12)$$

若 $l \leq a$, 则针只与一条平行线相交的概率为

$$P_4 = \frac{l}{a} \quad (13)$$

若 $l \geq 2a$, 则针只与一条平行线相交的概率为 $P_5 = 0$.

证明. 以二维平面为例, 用二维随机变量 (X, Φ) 来确定针在平面上的具体位置. 设 X 表示针的端到平行线的距离, 在 $(0, a)$ 范围内服从均匀分布, 由此可以写出 X 的概率密度函数为式 (10) 所示. Φ 表示针与平行线的夹角, 因此根据极值分辨度的定义, 可知针与平行线始终为垂直方向, $\Phi = \pi/2$. 若 $2a \geq l \geq a$, 当 $l-a < X < a$ 时, 针只与一条直线相交, 其概率为

$$P_3 = P\{l-a < X < a\} = \int_{l-a}^a f(x) dx = \frac{2a-l}{a} \quad (14)$$

若 $l \leq a$, 当 $0 < X < l$ 时, 针只与一条直线相交, 其概率为

$$P_4 = P\{0 < X < l\} = \int_0^l f(x) dx = \frac{l}{a} \quad (15)$$

若 $l \geq 2a$, 则针与一条平行线相交的概率为 $P_5 = 0$. \square

定理 3. 在噪声环境下, 对于由两条平行线确定的局部范围内的极值点, 若针同时与两条平行线相交, 则找到该极值点的概率为 $P_6 = 1 - P_\xi$, 若针只与其中一条平行线相交, 则找到该极值点的概率为 $P_7 = 1 - P_\gamma - P_\xi$.

证明. 根据算法的基本原理和极值分辨度的定义, 若针同时与两条平行线相交, 说明由两条平行线确定的两个采样点包含在局部极值点所在的深井

范围内,即两个采样点确定的范围内只包含唯一一个极值点,在此局部范围内进行单极值点寻优,无噪声环境下算法将以概率 1 找到该极值点,在噪声环境下,该区域内将出现若干噪声引起的极值点,若以概率 P_ξ 出现函数值低于该极值点函数值的点,则算法将保留函数值更低的点,故找到该极值点的概率为 $P_6 = 1 - P_\xi$.

若针只与一条平行线相交,则在三条平行线确定的范围内,除去此针长 l 的确定的范围,其余 $(2a - l)$ 的范围内,若以概率 P_γ 出现函数值更低的极值点,或以概率 P_ξ 出现函数值更低的噪声波动点,则算法将留取函数值更低的点,故 $P_7 = 1 - P_\gamma - P_\xi$.

综上可得如下推论:

推论 1. 设 a 为采样平行线间距,称为蒲丰距离,以函数的极值分辨度为固定针长 l ,若 $2a \geq l \geq a$,则算法找到极值分辨度最小的极值点的概率为

$$P_{2a \geq l \geq a} = P_1 \cdot P_6 + P_3 \cdot P_7 = \frac{l - a}{a} \cdot (1 - P_\xi) + \frac{2a - l}{a} \cdot (1 - P_\gamma - P_\xi) = 1 - P_\xi - \frac{2a - l}{a} \cdot P_\gamma \quad (16)$$

若 $l < a$,则算法找到极值分辨度最小的极值点的概率为

$$P_{l < a} = P_2 \cdot P_6 + P_4 \cdot P_7 = \frac{l}{a} \cdot (1 - P_\gamma - P_\xi) \quad (17)$$

以一维函数为例,图 5 (a) 和 5 (b) 所示分别是 $2a \geq l \geq a$ 和 $l < a$ 时的蒲丰距离与针长的示意图.

对比式 (16) 和 (17) 可以看出,在函数的极值分辨度 l 既定的情况下,蒲丰距离 a 越小,算法找到所有极值点的概率也就越大,反之则遗漏的极值点将会越多.然而, a 取值过小则会带来数据量急剧增加,延长了算法的运算时间,因此,要根据工程需要来设定 a 的值,例如,当 $a = (1/2)l$ 时,算法找到极值分辨度最小的极值点的概率为 $1 - P_\xi$,仅受噪声引起的局部极值点的影响,若在无噪声环境下,则此概率为 1,即当蒲丰距离为针长的一半时,算法以概率 1 找到极值分辨度最小的极值点,这一结论与采样定理相符.

推论 2. 设 a 为采样平行线间距,以函数的极值分辨度为固定针长 l .对于全局极值点,由于 $P_\gamma = 0$,故

$$P_{2a \geq l \geq a} = 1 - P_\xi \quad (18)$$

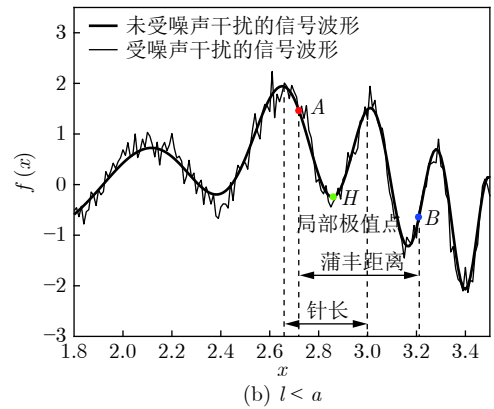
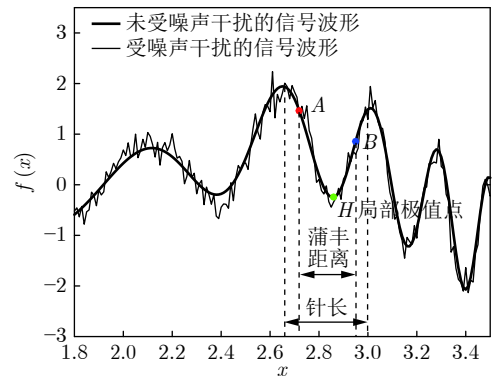


图 5 蒲丰距离与针长的关系示意图

Fig.5 Diagram of the relationship between Buffon distance and needle length

$$P_{l < a} = (l/a) \cdot (1 - P_\xi) \quad (19)$$

对比式 (16) 与 (18) 以及 (17) 与 (19) 可知,算法找到全局极值点的概率大于找到局部极值点的概率,因此算法具有较好的全局极值点优化性能.

2.3 改进的斐波那契法

对于无约束优化问题,斐波那契法是求解一维单峰函数的最优策略^[43].将斐波那契法的变量维数从一维扩展到多维,可以解决多维函数优化问题^[43].斐波那契策略通过按比例压缩搜索区间令区间内的试探点不断逼近最优解.斐波那契法中需要用到斐波那契数列,其通项递推公式为:

$$\begin{cases} F_1 = F_2 = 1 \\ F_k = F_{k-1} + F_{k-2}, k > 2 \end{cases} \quad (20)$$

斐波那契数列中相邻两项之比形成的数列收敛于黄金分割数^[44],即

$$\lim_{i \rightarrow \infty} \frac{F_k}{F_{k+1}} = 0.6180339887499 \dots \quad (21)$$

PMB 算法在进行局部搜索时,为使算法能够跳出噪声引入的局部最优,算法的搜索范围不能只

局限于初始的局部范围, 而应适当扩展, 探索周围邻域内是否有更优点. 如有更优点, 则搜索区域应该适当扩展, 向更优点偏移. 可以利用试探点引导区域向优化点方向移动. 同时, 为使算法收敛, 搜索区域还需进行压缩. 随着迭代次数的增加, 区域最终将收敛于可接受的精度范围内. 为此提出区域伸缩准则, 定义如下:

对于由任意两点 $A(\mathbf{x}_A, f(\mathbf{x}_A))$ 和 $B(\mathbf{x}_B, f(\mathbf{x}_B))$ 确定的区域边界, 若 $f(\mathbf{x}_A) < f(\mathbf{x}_B)$, 则选取点 $A(\mathbf{x}_A, f(\mathbf{x}_A))$ 为中心, 将区域边界以 α 为比例扩展到点 $C(\mathbf{x}_C, f(\mathbf{x}_C))$, 其中,

$$\alpha = \frac{F_k}{F_{k+1}} \quad (22)$$

$$\mathbf{x}_C = \mathbf{x}_A - (1 - \alpha) \cdot (\mathbf{x}_B - \mathbf{x}_A) \quad (23)$$

点 $C(\mathbf{x}_C, f(\mathbf{x}_C))$ 用来试探新扩展的区域中是否存在更优的点, 然后再以比例 β 对区间进行压缩, 压缩率 β 根据对扩展点 $C(\mathbf{x}_C, f(\mathbf{x}_C))$ 的试探情况进行设置. 若 $f(\mathbf{x}_A) > f(\mathbf{x}_B)$, 则选取点 $B(\mathbf{x}_B, f(\mathbf{x}_B))$ 为中心, 做类似扩展和压缩操作.

利用区域伸缩准则改进斐波那契法, 得到算法步骤如下:

步骤 1. 选定初始区间 $[a_1, b_1]$ 及求解精度 $\varepsilon > 0$, 令 $k = 1$, 若 $f(a_1) > f(b_1)$ 转步骤 2, 否则转步骤 3;

步骤 2. 计算

$$\begin{aligned} \lambda_k &= a_k + \frac{F_k}{F_{k+1}} \cdot |b_k - a_k| \\ \mu_k &= 2b_k - \lambda_k \end{aligned} \quad (24)$$

若 $f(\mu_k) < f(a_k)$ 时转步骤 4, 否则转步骤 5;

步骤 3. 计算

$$\begin{aligned} \mu_k &= b_k - \frac{F_k}{F_{k+1}} \cdot |b_k - a_k| \\ \lambda_k &= 2a_k - \mu_k \end{aligned} \quad (25)$$

若 $f(\lambda_k) < f(b_k)$ 时转步骤 4, 否则转步骤 6;

步骤 4. 置

$$\begin{cases} a_{k+1} = \lambda_k \\ b_{k+1} = \mu_k \end{cases} \quad (26)$$

若 $|b_{k+1} - a_{k+1}| \leq \varepsilon$, 转步骤 7, 否则令 $k = k + 1$, 转步骤 2;

步骤 5. 置

$$\begin{cases} a_{k+1} = a_k \\ b_{k+1} = \lambda_k \end{cases} \quad (27)$$

若 $|b_{k+1} - a_{k+1}| \leq \varepsilon$, 转步骤 7, 否则令 $k = k + 1$, 转步骤 2;

步骤 6. 置

$$\begin{cases} a_{k+1} = \mu_k \\ b_{k+1} = b_k \end{cases} \quad (28)$$

若 $|b_{k+1} - a_{k+1}| \leq \varepsilon$, 转步骤 7, 否则令 $k = k + 1$, 转步骤 2;

步骤 7. 停止计算, 取极小值点的坐标为 $1/2 |b_{k+1} - a_{k+1}|$.

由算法执行步骤可以看出, 初始搜索区间由 $[a_1, b_1]$ 界定, 但后续搜索范围并不仅仅局限于 $[a_1, b_1]$, 而是向周围邻域进行了适当的扩展, 探索周围是否还存在更优的点. 虽然每次都对区间进行了扩展, 但是每次计算结束时都根据探索的结果对区间进行压缩, 根据选取的扩展点不同, 第 k 次计算后的区域压缩率为 β_1 或 β_2 ,

$$\beta_1 = \frac{|\mu_k - \lambda_k|}{|b_k - a_k|} = 2 \left(1 - \frac{F_k}{F_{k+1}} \right) \quad (29)$$

$$\beta_2 = \frac{|b_k - \mu_k|}{|b_k - a_k|} = \frac{|\lambda_k - a_k|}{|b_k - a_k|} = \frac{F_k}{F_{k+1}} \quad (30)$$

由斐波那契数列的特点, 可知

$$\begin{aligned} \lim_{k \rightarrow \infty} \beta_1 &= \lim_{k \rightarrow \infty} \frac{|\mu_k - \lambda_k|}{|b_k - a_k|} = \\ &= \lim_{k \rightarrow \infty} 2 \left(1 - \frac{F_k}{F_{k+1}} \right) = \\ &= 2 \times (1 - 0.618) = 0.764 \end{aligned} \quad (31)$$

$$\begin{aligned} \lim_{k \rightarrow \infty} \beta_2 &= \lim_{k \rightarrow \infty} \frac{|b_k - \mu_k|}{|b_k - a_k|} = \lim_{k \rightarrow \infty} \frac{|\lambda_k - a_k|}{|b_k - a_k|} = \\ &= \lim_{k \rightarrow \infty} \frac{F_k}{F_{k+1}} = 0.618 \end{aligned} \quad (32)$$

可见, 算法的搜索区间在逐渐缩小, 最后可以收敛到设定的精度范围内.

为了对比改进前的斐波那契法与改进后的斐波那契法, 以受噪声干扰的一维函数为例, 图 6 的 9 个子图展示了改进的斐波那契法的区域伸缩过程, 图 7 的 4 个子图展示了改进前的斐波那契法的区域收缩过程. 其中实线波形是受噪声干扰的函数波形, 点 ‘.’ 表示区域边界点 a_i 和 b_i , 点 ‘*’ 代表试探点 λ_i 和 μ_i . 图 6(a) 表示局部搜索的初始区间由 $[a_1, b_1] = [2.6700, 3.0300]$ 确定. 由于 $f(a_1) > f(b_1)$, 根据式 (24) 可以得到式 (33):

$$\left\{ \begin{aligned} \lambda_1 &= a_1 + \frac{F_k}{F_{k+1}} \times |b_1 - a_1| = \\ &2.67 + \frac{1}{2} \times (3.03 - 2.67) = 2.85 \\ f(\lambda_1) &= -0.3019 \\ \mu_1 &= 2 \cdot b_1 - \lambda_1 = 2 \times 3.03 - 2.85 = \\ &3.21 \\ f(\mu_1) &= -0.7351 \end{aligned} \right. \quad (33)$$

由于 $f(\mu_1) < f(a_1)$, 根据式 (26) 可以得到新的压缩后的搜索区间 $[\lambda_1, \mu_1]$, 也就是图 6 (b) 中的搜索区间 $[a_2, b_2]$. 从图 6 (a) 中可以看出, 初始搜索区间先由试探点 μ_1 进行扩展, 再由试探点 λ_1 进行压. 在后续的 6 (b)~6 (i) 子图中按照改进的斐波那契

法的区域伸缩准则执行类似操作. 图 7 则按照斐波那契法的区域压缩规则进行迭代, 此处不再赘述.

通过比较图 6 中的 9 个子图, 我们可以观察到改进后的斐波那契法的搜索区间可以扩展到具有更优解的邻域, 而不是初始时的局部范围. 而从图 6 可以看到, 传统的斐波那契法的搜索区间始终限制在初始的局部范围内. 此外, 从图 7 中可以看出, 斐波那契法在第 4 次迭代时区间长度就压缩到了 0.0758, 而改进后的斐波那契法在第 7 次迭代时区间长度才压缩到 0.0704, 说明引入了区域伸缩准则的斐波那契法每次迭代都遵循先扩展再压缩的规则, 因此区域压缩和算法收敛的速度比改进前的斐波那契法下降了, 但是改进后的算法可以搜索到初

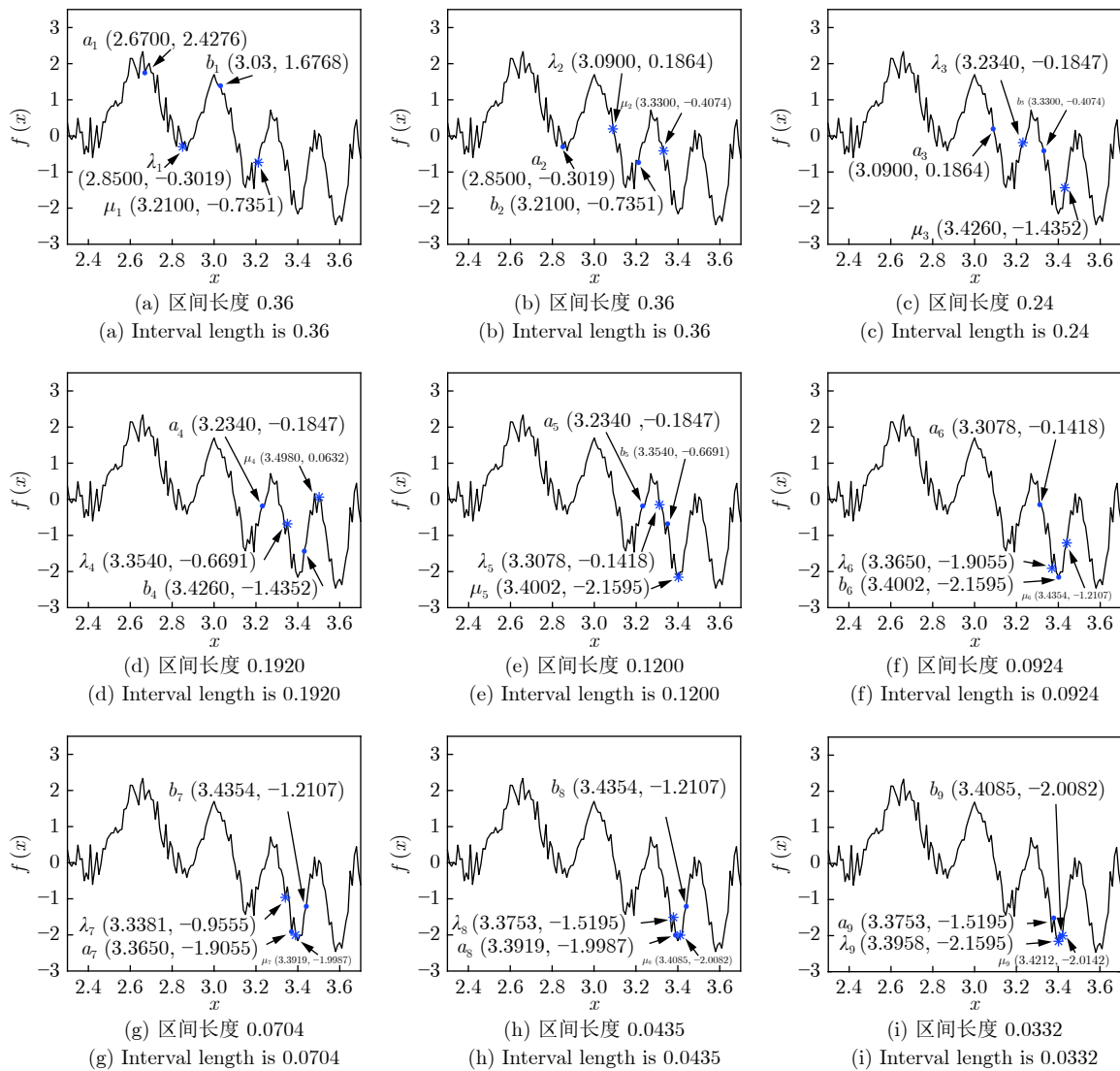


图 6 改进的斐波那契法的区域伸缩过程

Fig. 6 The process of regional scaling of the improved Fibonacci method

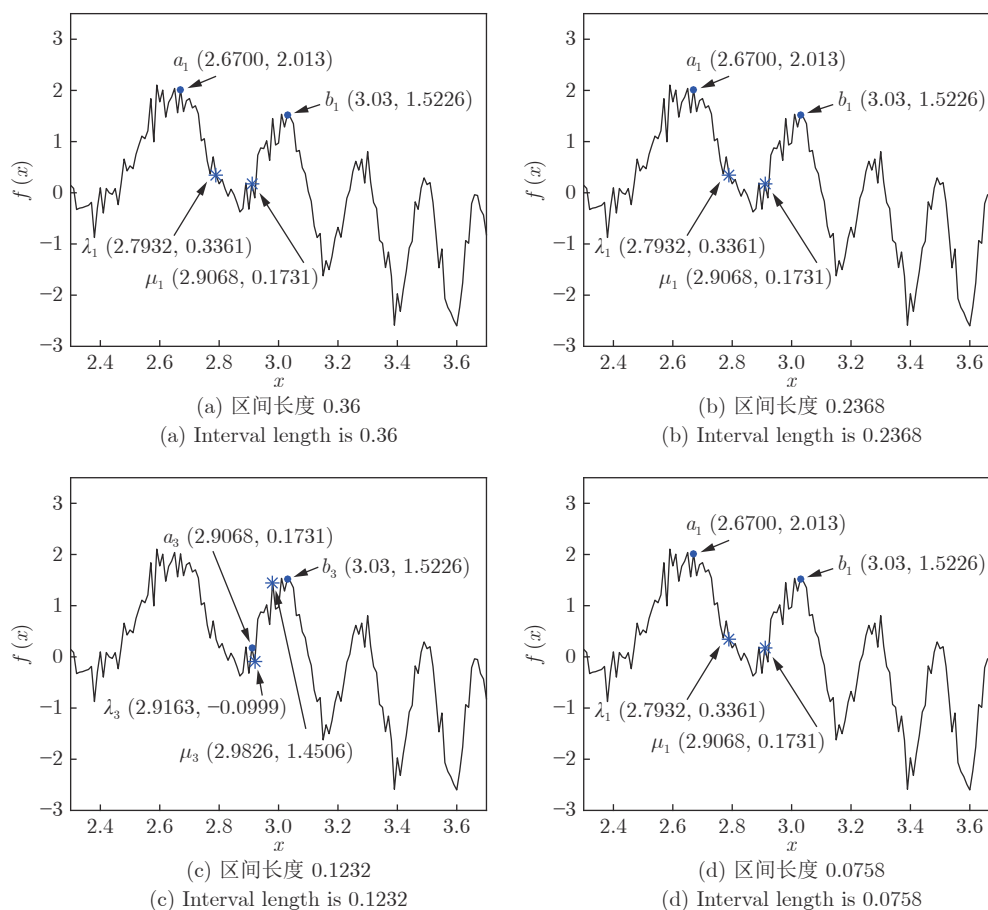


图 7 斐波那契法的区域收缩过程

Fig. 7 The process of regional shrink of the Fibonacci method

始区间附近邻域范围内的极小值点, 提高了算法在函数真实最优解邻域附近的收敛能力, 避免其陷入噪声引入的局部最优.

2.4 PMB 算法流程

步骤 1. 初始化设置. 根据实际需求的极值分辨率确定蒲丰距离 a 和求解精度 ε .

步骤 2. 在全局范围内以蒲丰距离 a 进行等间隔采样, 得到采样点集合 $S = \{A_1, A_2, \dots, A_k\}$.

步骤 3. 以采样点集合 S 中的相邻点作为初始区间的两个端点, 在局部范围内利用区域伸缩的斐波那契法进行单极值点寻优, 可以得到 $k-1$ 个极值点 $\{E_1, E_2, \dots, E_{k-1}\}$.

步骤 4. 利用改进型插点法^[45]对得到的极值点进行同峰判断.

步骤 5. 利用归并方法^[1]将同峰的极值点进行合并, 并取这些点中的最大(最小)值最为最终解.

2.5 PMB 算法的多级划分策略

为了提高 PMB 算法的求解精度, 同时也为了

提高算法在多维函数上的优化效率, 可以采用多级划分策略. 对于 N 级划分策略, 首先依据蒲丰距离对全局区域进行划分, 得到多个探索点, 在多个探索点之间进行局部寻优, 可以保留下第一级的多个极值点. 对第一级的极值点的极值大小进行比较后, 选取一定数量的较优极值点, 对它们所在的区域仍然依据蒲丰距离进行下一级划分, 得到多个探索点, 在多个探索点之间再进行局部寻优, 从而得到第二级的多个极值点. 依此类推, 直到划分到第 N 级. 对第 N 级得到的极值点, 利用改进型插点法^[45]进行同峰判断, 再利用归并方法^[1]将同峰的极值点进行合并, 并取这些点中的最大(最小)值最为最终解.

以图 8 (a) 所示的二维平面为例, 首先在每一维上进行等距离划分, 可以得到若干交点, 即为第一级的采样点, 如图 8 (a) 中的实心交点所示. 每一维上划分的间隔距离可以相同, 也可以不同, 一般情况下设置为每一维上的划分间隔都是相同的, 特殊情况除外. 如图 8 (a) 所示, 虽然两个维度上的范围尺度不同, 但不影响对它们进行等距离的划分. 然后每个实心交点与其相邻的点之间进行局部寻

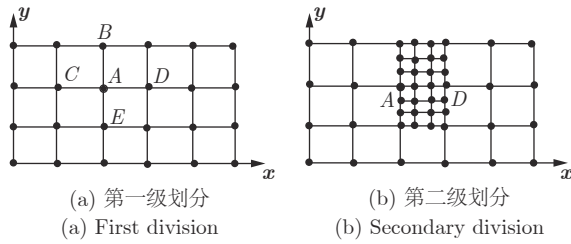


图 8 PMB 算法的多级划分策略

Fig.8 The multilevel partition strategy of PMB algorithm

优, 得到若干第一级极值点. 以图 8 (a) 中点 A 为例, 它将分别与点 B、C、D、E 之间进行局部寻优. 在第二级划分时, 若选中了 A 和 D 局部寻优得到的极值点, 则将对 A 和 D 两点所在的公共区域进行第二级划分, 得到第二级的采样点, 如图 8 (b) 所示. 可以证明, 若 n 维变量空间的每一维划分为 m 段区间, 则在每一维上可以得到 $m + 1$ 个采样点. 所有采样点经局部寻优后将得到 $m \times (m + 1) \times 2^{(2 \times n - 3)}$ 个极值点. 以三维变量空间为例, 假设每一维的长度域为 $[-100, 100]$, 第一级若取蒲丰距离 $a_1 = 5$, 则每一维可得到 $m_1 = 40$ 段区间和 41 个采样点. 所有采样点经局部寻优后将得到 $m_1 \times (m_1 + 1) \times 2^{(2 \times n - 3)} = 40 \times 41 \times 2^3 = 13\,120$ 个第一级极值点. 从第一级极值点中选出部分较优极值点, 对它们所在的区域依据蒲丰距离 $a_2 = 0.1$ 进行下一级划分, 对于其中一个子区域而言, 其每一维的长度域为 5, 按照 $a_2 = 0.1$ 划分, 每一维可以得到 $m_2 = 50$ 段区间和 51 个采样点, 从而每个区间可以得到 $m_2 \times (m_2 + 1) \times 2^{(2 \times n - 3)} = 50 \times 51 \times 2^3 = 20\,400$ 个第二级极值点, 之后执行类似操作, 直到达到指定的划分等级.

从上述过程可以看出, 初始时每一维的长度域为 200, 经过第一级划分后, 每一维的长度域为 5,

经过第二级划分后, 每一维的长度域为 0.1, 若第二级只选取 1 个最优区域进行划分, 则需要局部寻优的次数为 $13\,120 + 20\,400 = 33\,520$. 若不采取多级划分策略, 直接对初始时长度域以 $a_2 = 0.1$ 的蒲丰距离进行划分, 则 $m = 2\,000$, 需要局部寻优的次数为 $m \times (m + 1) \times 2^{(2 \times n - 3)} = 2\,000 \times 2\,001 \times 2^3 = 32\,016\,000$. 可见采取多级划分策略可以省去很多计算时间, 这在多维函数的优化问题上优势更为明显. 此外, 多级划分策略不仅在同级区域内横向可并行执行, 不同级区域也可纵向并行执行, 计算过程不会相互干涉.

3 实验仿真与分析

3.1 实验设计与测试函数

为了验证本文所提出的 PMB 算法的寻优结果与蒲丰距离之间的依概率关系, 研究噪声和蒲丰距离对 PMB 算法寻优结果的影响, 验证 PMB 算法在噪声环境下的多极值点寻优能力, 以及研究 PMB 算法在噪声环境下对多维函数寻优的有效性, 本文选用两组基准测试函数研究 PMB 算法的上述性能. 第一组测试函数来自文献 [1] 和文献 [32], 如表 1 所示. 这 6 个测试函数都是多峰函数, 不仅有一到多个全局极值点, 还有若干局部极值点, 其中, 除 f_2 , f_3 和 f_4 只有唯一一个全局极值点, 其他函数均有多个全局极值点. 第二组测试函数是文献 [46] 中给出的 CEC2013 推荐基准函数, 如表 2 所示. 其中, $f_1 \sim f_5$ 是单峰函数, $f_6 \sim f_{20}$ 是多峰函数, $f_{21} \sim f_{28}$ 是组合函数. CEC2013 被视为黑盒问题, 并且不允许使用问题的显式方程[46].

3.2 PMB 算法依概率收敛特性验证

通过第 3.2 节的理论推导可知, PMB 算法的寻

表 1 测试函数集 1 (TF1)
Table 1 Test function set 1 (TF1)

| 函数 | 名称 | 定义域 | 全局最优值f (与之对应的最优解 x_1, x_2) | 表达式 |
|-------|----------------|-------------|---------------------------------------|--|
| f_1 | | $[-1, 1]$ | 3.5326 ($\pm 0.8781, \pm 0.8781$) | $f_1 = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$ |
| f_2 | Levy 05 | $[-10, 10]$ | -176.1375 (-1.3068, -1.4248) | $f_2 = \sum_{i=1}^5 i \cos[(i-1)x_1 + i] \times \sum_{j=1}^5 j \cos[(j+1)x_2 + j] + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2$ |
| f_3 | Michalewics | $[0, \pi]$ | -1.8013 (2.2031, 1.5709) | $f_3 = -\sum_{i=1}^{20} \sin(x_i) \left(\sin\left(x_i \times \frac{\pi}{i}\right)\right)^{20}$ |
| f_4 | Periodic | $[0, \pi]$ | 0.9 (0,0) | $f_4 = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{-(x_1^2 + x_2^2)}$ |
| f_5 | Carrom Table | $[-10, 10]$ | -24.1568 ($\pm 9.6466, 9.6466$) | $f_5 = -\left[\cos(x_1) \cos(x_2) e^{\left 1 - \sqrt{x_1^2 + x_2^2} / \pi\right } \right]^2 / 30$ |
| f_6 | Holder Table 2 | $[-10, 10]$ | -19.2085 ($\pm 8.0550, \pm 9.6646$) | $f_6 = -\left \sin(x_1) \cos(x_2) e^{\left 1 - \sqrt{x_1^2 + x_2^2} / \pi\right } \right $ |

表 2 测试函数集 2 (TF2) (维度为 5)
Table 2 Test function set 2 (TF2) (Dimension is 5)

| | 函数 | 全局最优值 f_i^* | 函数名称 |
|----------|----------|--|---|
| 单峰函数 | f_1 | -1 400 | Sphere function |
| | f_2 | -1 300 | Rotated high conditioned elliptic function |
| | f_3 | -1 200 | Rotated bent cigar function |
| | f_4 | -1 100 | Rotated discus function |
| | f_5 | -1 000 | Different powers function |
| 多峰函数 | f_6 | -900 | Rotated Rosenbrock's function |
| | f_7 | -800 | Rotated Schaffers F7 function |
| | f_8 | -700 | Rotated Ackley's function |
| | f_9 | -600 | Rotated Weierstrass function |
| | f_{10} | -500 | Rotated Griewank's function |
| | f_{11} | -400 | Rastrigin's function |
| | f_{12} | -300 | Rotated Rastrigin's function |
| | f_{13} | -200 | Non-continuous rotated Rastrigin's function |
| | f_{14} | -100 | Schwefel's function |
| | f_{15} | 100 | Rotated Schwefel's function |
| | f_{16} | 200 | Rotated Katsuura function |
| | f_{17} | 300 | Lunacek bi_Rastrigin function |
| | f_{18} | 400 | Rotated lunacek bi_Rastrigin function |
| f_{19} | 500 | Expanded Griewank's plus Rosenbrock's function | |
| f_{20} | 600 | Expanded Scaffer's F6 function | |
| 组合函数 | f_{21} | 700 | Composition function 1 ($n = 5$, Rotated) |
| | f_{22} | 800 | Composition function 2 ($n = 3$, Unrotated) |
| | f_{23} | 900 | Composition function 3 ($n = 3$, Rotated) |
| | f_{24} | 1 000 | Composition function 4 ($n = 3$, Rotated) |
| | f_{25} | 1 100 | Composition function 5 ($n = 3$, Rotated) |
| | f_{26} | 1 200 | Composition function 6 ($n = 5$, Rotated) |
| | f_{27} | 1 300 | Composition function 7 ($n = 5$, Rotated) |
| | f_{28} | 1 400 | Composition function 8 ($n = 5$, Rotated) |

优结果与蒲丰距离之间存在依概率关系. 当蒲丰距离 a 与针长 l 满足 $2a \geq l \geq a$ 时, 算法找到极值分辨率最小的极值点的概率满足式 (16), 当 $l \leq a$ 时, 概率关系满足式 (17). 为了验证上述依概率关系, 在本节实验中分别取两种不同的蒲丰距离 $l \leq a$ 和 $2a \geq l \geq a$, 观察 PMB 算法的寻优结果并进行讨论. 为了确定针长, 参考文献 [1] 所得到的结果及本文定义 2 可知, TF1 中的 f_1 的极值分辨率大约为 0.38, 即针长 $l \approx 0.38$. 分别取蒲丰距离 $a = 0.5$ (即 $l \leq a$) 和 $a = 0.2$ (即 $2a \geq l \geq a$) 两种情况, 求解精度 $\varepsilon = 0.01$, 验证本文所提 PMB 算法在无噪声环境和不同噪声环境下对 TF1 中的 f_1 中的所有极值点求解结果的依概率特性.

参照文献 [32] 中的噪声环境设置参数, 分别取噪声方差 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$, 重复独立实验

50 次、100 次和 200 次, PMB 算法求得的优化数值结果列于表 3 中. 表中 σ^2 表示噪声方差, a 表示蒲丰距离, Num_mean 表示算法平均搜索到的极值点数, Num_min 和 Num_max 分别表示算法在重复实验中搜索到的极值点个数的最小值和最大值, η_{mean} 表示平均峰值检测率, η_{global} 表示全局极值点检测率. 从表 3 中可以看出:

当蒲丰距离 $a = 0.2$ 时, 即满足 $2a \geq l \geq a$ 时, 可得出如下结论:

1) 在 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$ 的噪声环境下, 重复独立实验 50 次、100 次和 200 次, PMB 算法都能找到所有极值点, η_{mean} 为 100 % 以上. 重复独立实验中算法每次最少都能确保找到全部 36 个极值点.

2) $\sigma^2 = 0.01$ 时算法最多能找到 37 个极值点,

表 3 针对 TF1 中的 f_1 取不同蒲丰距离时 PMB 算法在不同噪声环境下的优化结果
 Table 3 Optimization results of PMB algorithm for f_1 of TF1 in different noise environments with different Buffon distances

| | σ^2 | a | Num_mean | Num_max | Num_min | η_{mean} | η_{global} |
|-------|------------|-----|----------|---------|---------|----------------------|------------------------|
| 50 次 | 0.01 | 0.5 | 29.62 | 33 | 26 | 82.28 % | 100.00 % |
| | | 0.2 | 36.04 | 37 | 36 | 100.11 % | 100.00 % |
| | 0.05 | 0.5 | 31.32 | 35 | 27 | 87.00 % | 100.00 % |
| | | 0.2 | 36.26 | 38 | 36 | 100.72 % | 100.00 % |
| 100 次 | 0.01 | 0.5 | 29.78 | 33 | 27 | 82.72 % | 100.00 % |
| | | 0.2 | 36.02 | 37 | 36 | 100.06 % | 100.00 % |
| | 0.05 | 0.5 | 32.08 | 36 | 28 | 89.11 % | 100.00 % |
| | | 0.2 | 36.37 | 38 | 36 | 101.03 % | 100.00 % |
| 200 次 | 0.01 | 0.5 | 29.715 | 34 | 23 | 82.54 % | 100.00 % |
| | | 0.2 | 36.02 | 37 | 36 | 100.06 % | 100.00 % |
| | 0.05 | 0.5 | 32 | 35 | 26 | 88.89 % | 100.00 % |
| | | 0.2 | 36.4 | 40 | 36 | 101.11 % | 100.00 % |

$\sigma^2 = 0.05$ 算法最多能找到 38 至 40 个极值点, 且 $\sigma^2 = 0.05$ 时的 η_{mean} 略高于 $\sigma^2 = 0.01$ 时的 η_{mean} , 出现了 η_{mean} 的值超出 100 % 的情况. 这是由于噪声的干扰引入了众多的局部极值, 使得算法找到的极值点数超出了原函数的极值点数. 可以看出, 随着噪声强度的增大, 这种干扰越明显, 噪声引入的局部极值点数越多.

3) 在 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$ 的噪声环境下, 重复独立实验 50 次、100 次和 200 次, PMB 算法都能找到所有全局极值点, η_{global} 都能达到 100 %.

当蒲丰距离 $a = 0.5$ 时, 即满足 $l \leq a$ 时, 有如下结论:

1) 在 $\sigma^2 = 0.01$ 的噪声环境下, 重复独立实验 50 次、100 次和 200 次, PMB 算法的 η_{mean} 都是 82 % 左右.

2) 在 $\sigma^2 = 0.05$ 的噪声环境下, 重复独立实验 50 次、100 次和 200 次, PMB 算法的 η_{mean} 都为 88 % 左右.

3) 由于噪声干扰会引入局部极值点, 因此当噪声强度增强时, PMB 算法的 η_{mean} 有所增加.

4) 在 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$ 的噪声环境下, 重复独立实验 50 次、100 次和 200 次, PMB 算法都能找到所有全局极值点, η_{global} 都能达到 100 %.

综上所述:

1) 在相同的噪声环境下, PMB 算法的平均峰值检测率是一个固定概率, 取决于算法的蒲丰距离 a . 对于 TF1 中的 f_1 , 当 $2a \geq l \geq a$ 时, 平均峰值检测率可达到 100 %, 当 $l \leq a$ 时, 平均峰值检测率约为 80 % 以上. 这一结果印证了推论 1, 即算法依固定概率找到函数的多个极值点.

2) 在不同噪声环境下取不同蒲丰距离时, PMB 算法都能找到 TF1 中的 f_1 的所有全局极值点. 这一结果印证了推论 2, 即算法找到全局极值点的概率高于找到局部极值点的概率. 表明 PMB 算法具有较好的全局极值点优化性能.

3) 噪声强度对峰值检测率有一定程度的影响. 由于噪声干扰会引入局部极值点, 因此当噪声强度增强时, PMB 算法的平均峰值检测率也会有所增加, 其中包含了少数几个噪声引入的局部极值点.

3.3 噪声强度与蒲丰距离对 PMB 算法寻优结果的影响

首先定量分析噪声强度和蒲丰距离对 PMB 算法求解结果的影响. 分别取 $a = 0.2$ 和 $a = 0.5$, 求解精度 $\varepsilon = 0.01$, 参照文献 [32], 分别设置 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$, PMB 算法对 TF1 中的 f_1 的一次寻优结果分别绘于图 9~图 14 中, 其中, 图 11~图 14 的 (a) 图呈现的是算法在同峰判断前找到的极值点, (b) 图呈现的是算法经同峰判断后确定的极值点.

从图 9~图 14 中可以看出, 算法求得的极值点个数受蒲丰距离的影响较为明显. 蒲丰距离越小, PMB 算法找到的极值点越多, $a = 0.2$ 时可以找到全部极值点, $a = 0.5$ 时, PMB 算法不能找到全部极值点, 个别局部极值点会被遗漏, 但仍然可以找到所有全局极值点.

为进行定性分析, 以全局极值点为例, 将 PMB 算法求解 TF1 中的 f_1 得到的全局极值点的数值结果分别列于表 4 和表 5 中.

从表 4 和表 5 中可以看到, 算法求解的定位误差和峰值误差同时受到蒲丰距离与噪声强度的影

响. 在蒲丰距离相同的情况下, 随着噪声强度的增加, 全局极值点的峰值误差 P_f 和定位误差 P_l 也随之增加. 在噪声强度相同的情况下, 蒲丰距离越小, 全局极值点的定位误差 P_l 越小, 定位精度越好.

除了全局极值点, PMB 算法还能求得多个局部极值点. 取 $a = 0.2$, PMB 算法在 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.09$ 的噪声环境中的 100 次独立重复实验结果列于表 6 中.

从表 6 中可以看出, 当 $a = 0.2$ 时 PMB 算法可以找到 TF1 中的 f_1 的全部极值点. 对比文献 [1] 的数值结果, 可以验证 PMB 算法在噪声环境下定位所有极值点的正确性. 依据文献 [1] 的数值结果计算得到 PMB 算法定位每个极值点的定位误差, 绘于图 15 中.

从图 15 中可以看出, 随着噪声强度的增加, 各个极值点的定位误差都随之增加. 综上可知:

1) 蒲丰距离的大小会对 PMB 算法找到极值点

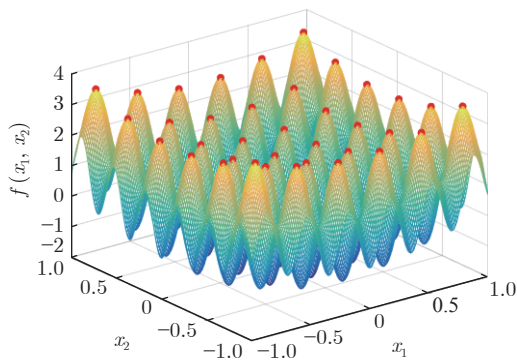


图 9 $a = 0.2$ 时无噪声环境下 PMB 算法对 TF1 中的 f_1 的寻优结果

Fig.9 Optimization results for f_1 of TF1 by PMB algorithm in noiseless environment with $a = 0.2$

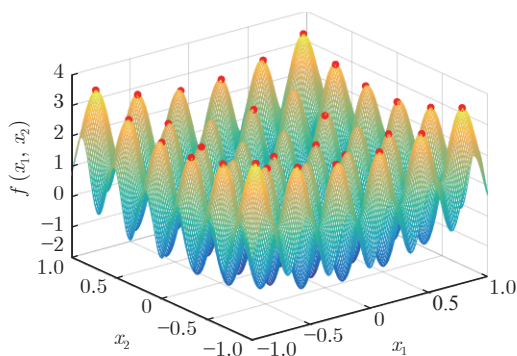
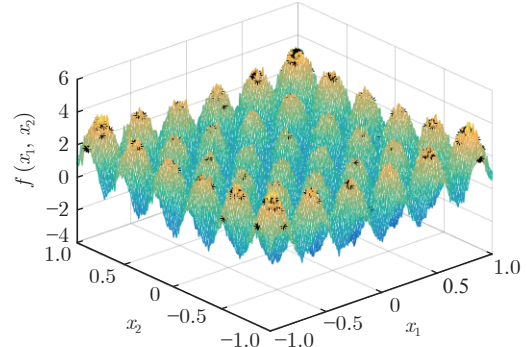
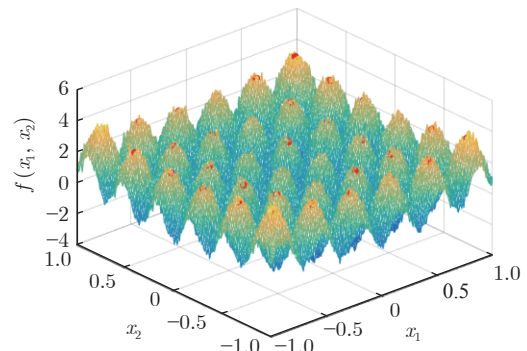


图 10 $a = 0.5$ 时无噪声环境下 PMB 算法对 TF1 中的 f_1 的寻优结果

Fig.10 Optimization results for f_1 of TF1 by PMB algorithm in noiseless environment with $a = 0.5$



(a) 同峰判断前找到的极值点
(a) The extremum points found before the same-peak judgment



(b) 经同峰判断后确定的极值点
(b) The extremum points determined by the same-peak judgment

图 11 $\sigma^2 = 0.05$, $a = 0.5$ 时 PMB 算法对 TF1 中的 f_1 的寻优结果

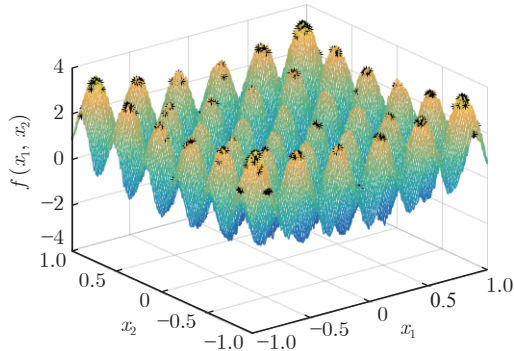
Fig.11 Optimization results for f_1 of TF1 by PMB algorithm in noise environment with $\sigma^2 = 0.05$ and $a = 0.5$

的个数以及求解极值点的定位精度产生主要影响. 蒲丰距离越小, 算法的全极值点寻优性能就越好, 遗漏的极值点越少, 峰值检测率越高, 同时定位准确度越高.

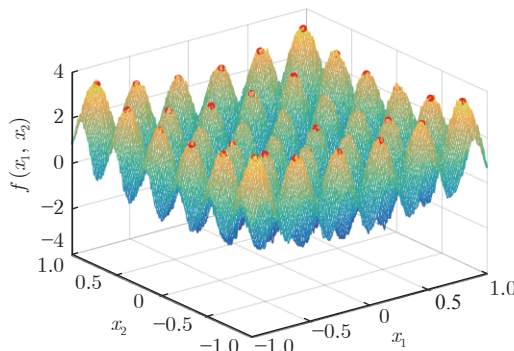
2) 噪声强度也会对 PMB 算法的峰值检测率和求解极值点的定位精度产生影响. 噪声强度的增加使得极值点的峰值误差 P_f 和定位误差 P_l 都随之增加, 同时会引入更多的局部最优, 使得峰值检测率增加.

3.4 PMB 算法在不同噪声环境下的多极值点寻优能力

本节实验选用文献 [32] 中的 5 个测试函数, 即表 1 中 TF1 函数 $f_2 \sim f_6$, 检验本文所提 PMB 算法在不同噪声环境下对不同函数的多极值点寻优能力, 并与文献 [32] 提出的改进蝙蝠算法 (Improved bat algorithm, IBA) 进行对比分析.



(a) 同峰判断前找到的极值点
(a) The extremum points found before the same-peak judgment



(b) 经同峰判断后确定的极值点
(b) The extremum points determined by the same-peak judgment

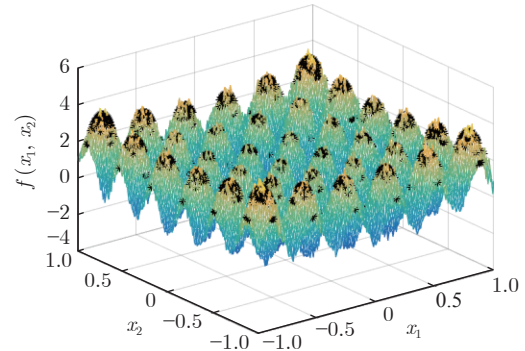
图 12 $\sigma^2 = 0.01, a = 0.5$ 时 PMB 算法对 TF1 中的 f_1 的寻优结果

Fig. 12 Optimization results for f_1 of TF1 by PMB algorithm in noise environment with $\sigma^2 = 0.01$ and $a = 0.5$

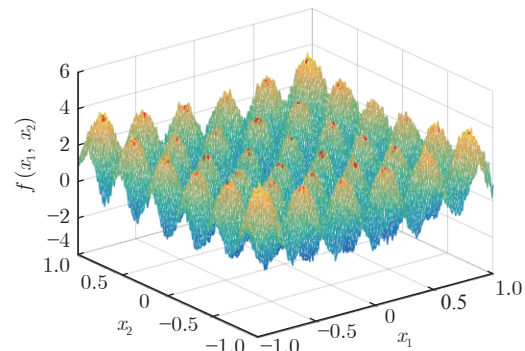
参照文献 [32] 的噪声环境设置, 分别在 $\sigma^2 = 0.01$ 、 $\sigma^2 = 0.05$ 、 $\sigma^2 = 0.07$ 和 $\sigma^2 = 0.09$ 的噪声环境中利用 PMB 算法对 TF1 函数 $f_2 \sim f_6$ 进行优化求解, 重复独立实验 100 次, 求解精度 $\varepsilon = 0.01$, 优化数值结果列于表 7 中. 表中 Mean 表示 100 次重复独立实验得到的全局极值点的适应度平均值, MAPE 表示适应度平均值相对百分比误差.

从表 7 中看到, 全局极值点的适应度值受噪声的影响非常明显, 随着噪声强度的增加, MAPE 随之增大. 从 TF1 中 f_2 的寻优结果上可以看到, 本文所提出的 PMB 算法可以得到比 IBA 算法更低的 MAPE, 而在其他函数上, PMB 算法的 MAPE 高于 IBA 算法, 但依然呈现出平均相对百分比误差随着噪声强度的增加而增大的规律.

尽管在 TF1 中 $f_3 \sim f_6$ 上 PMB 算法得到的 MAPE 更高, 但是在噪声环境下衡量算法的优化性能应该更关注算法的定位误差指标. 在无噪声环境



(a) 同峰判断前找到的极值点
(a) The extremum points found before the same-peak judgment

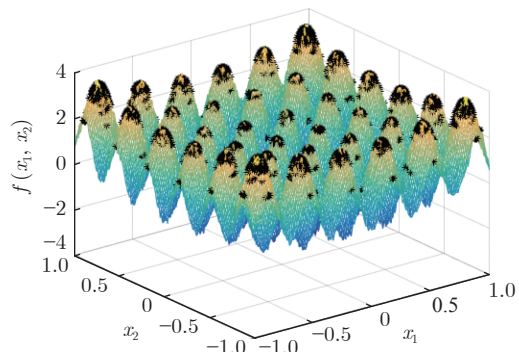


(b) 经同峰判断后确定的极值点
(b) The extremum points determined by the same-peak judgment

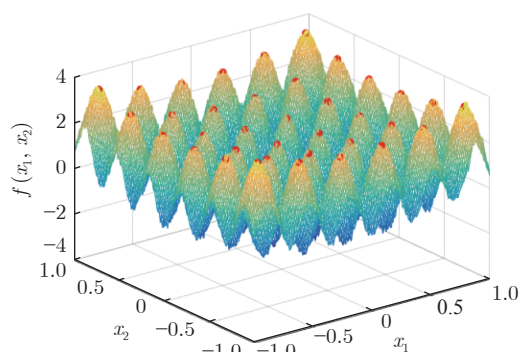
图 13 $\sigma^2 = 0.05, a = 0.2$ 时 PMB 算法对 TF1 中的 f_1 的寻优结果

Fig. 13 Optimization results for f_1 of TF1 by PMB algorithm in noise environment with $\sigma^2 = 0.05$ and $a = 0.2$

下, 衡量算法的寻优性能往往通过考察算法求得的全局极值点的适应度值与真实适应度值的误差来得到, 例如, 函数 f_2 在无噪声情况下的全局极值点是 -176.1375 ($-1.3068, -1.4248$), 如果算法找到的极值点的适应度值近似为 -176.1375 , 则通常认为算法已经找到了全局极值点, 同时也证明了算法的全局优化性能较好. 而在噪声环境下, 点 $(-1.3068, -1.4248)$ 处的函数值将发生变化, 偏离原来的函数值 -176.1375 , 因此, 衡量算法在噪声环境下优化求解性能的指标不应是找到函数值为 -176.1375 的点, 这样的点极有可能已经不是原函数的全局极值点, 算法优化求解的目的应该是力图找到 $(-1.3068, -1.4248)$ 这个点, 因为它才是原函数的全局极值点位置. 因此, 在噪声环境下, 求解极值点的坐标显得尤为重要, 它是判断算法是否能够找到全局极值点的有力依据, 而定位误差则反映了算法定位全局极值点的精确度.



(a) 同峰判断前找到的极值点
(a) The extremum points found before the same-peak judgment



(b) 经同峰判断后确定的极值点
(b) The extremum points determined by the same-peak judgment

图 14 $\sigma^2 = 0.01, a = 0.2$ 时 PMB 算法对 TF1 中的 f_1 的寻优结果

Fig. 14 Optimization results for f_1 of TF1 by PMB algorithm in noise environment with $\sigma^2 = 0.01$ and $a = 0.2$

PMB 算法在不同噪声环境下求解 TF1 中 $f_2 \sim f_5$ 得到的全局极值点的定位误差分别列于表 8 和表 9 中.

从表 8 和表 9 中可以看到, PMB 算法求出了每个极值点对应的坐标值, 明确了全局极值点的位置, 可以直观地与无噪声时的真实极值点位置做对

比, 从而证明 PMB 算法在噪声环境下定位全局极值点的准确性. 而文献 [32] 只给出了 IBA 算法求解全局极值点的 P_l 结果, 并未给出具体的坐标值.

同时, 对比两种算法得到的 P_l 值可以看到, 除了 4 个极值点 (f_2 在 $\sigma^2 = 0.09$ 时、 f_4 在 $\sigma^2 = 0.01$ 和 $\sigma^2 = 0.05$ 时、 f_6 的第 3 个极值点在 $\sigma^2 = 0.09$ 时) 外, PMB 算法在不同噪声环境下求解 TF1 中 $f_2 \sim f_5$ 得到的全局极值点的定位误差均小于 IBA 算法得到的结果. 即 PMB 算法在不同噪声环境下对 TF1 中 $f_2 \sim f_5$ 的全局极值点定位比 IBA 算法更为精确.

此外, 文献 [32] 的 IBA 算法只给出了 TF1 中 $f_2 \sim f_5$ 的全局极值点求解结果, 对其他局部极值点并未讨论. 而 PMB 算法运行一次即可得到全部极值点和多个局部极值点. 以 TF1 中 f_2 为例, 取蒲丰距离 $a = 1$ 时, PMB 算法运行一次得到的优化结果如图 16 所示.

从图 16 中可以看出, TF1 的 f_2 是一个多峰函数, 其在 $\sigma^2 = 0.05$ 的噪声环境下出现了无穷多的噪声峰值点, PMB 算法有效的找到了它的 1 个全局极值点及 170 个局部极值点. 综上可知:

- 1) PMB 算法在噪声环境下可以一次性得到极值点的适应度值和对应坐标, 可以为生产实际提供明确的决策变量.
- 2) PMB 算法求解的极值点位置精度总体高于 IBA 算法.
- 3) PMB 算法不仅能够求得噪声环境下的全局极值点位置, 还能得到多个局部极值点的位置, 为生产实际提供更多的备选方案.

3.5 PMB 算法在 CEC2013 测试函数上与 PSO 算法的对比

本节实验选用文献 [46] 中给出的 CEC2013 推荐基准函数, 即表 2 中的函数 $f_1 \sim f_{28}$, 检验 PMB 算法在噪声环境下对多维函数寻优的有效性, 并与 PSO 算法进行对比分析. 为了检验 PMB 算法和

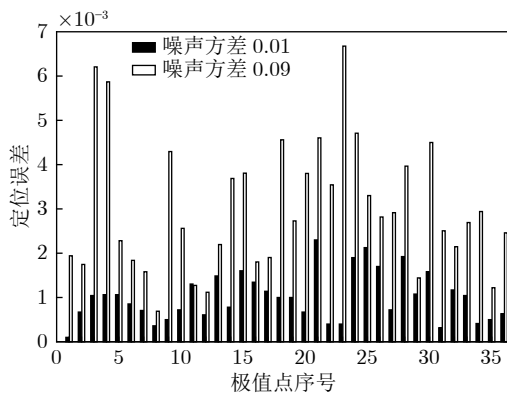
表 4 $a = 0.2$ 时 PMB 算法针对 TF1 中的 f_1 在不同噪声环境下的全局极值点优化数值结果

Table 4 Numerical results of global extremum points for f_1 of TF1 searched by PMB algorithm with $a = 0.2$ under different noise environments

| 极值点序号 | P-MQHOA ^[1] 无噪声 | | | PMB | | | | | | | | | |
|-------|----------------------------|---------|--------|-------------------|---------|--------|-----------------------|-----------------------|-------------------|---------|--------|-----------------------|-----------------------|
| | | | | $\sigma^2 = 0.01$ | | | | | $\sigma^2 = 0.05$ | | | | |
| | x_1 | x_2 | f | x_1 | x_2 | f | P_f | P_l | x_1 | x_2 | f | P_f | P_l |
| 1 | -0.8781 | -0.8781 | 3.5326 | -0.8781 | -0.878 | 3.8057 | 2.73×10^{-1} | 1.42×10^{-1} | -0.8776 | -0.8773 | 4.1956 | 6.63×10^{-1} | 9.37×10^{-1} |
| 2 | -0.8781 | 0.8781 | 3.5326 | -0.8789 | 0.8784 | 3.8027 | 2.70×10^{-1} | 7.86×10^{-1} | -0.8788 | 0.8782 | 4.1974 | 6.65×10^{-1} | 8.19×10^{-1} |
| 3 | 0.8781 | -0.8781 | 3.5326 | 0.8782 | -0.8778 | 3.8086 | 2.76×10^{-1} | 3.64×10^{-1} | 0.8788 | -0.8784 | 4.2048 | 6.72×10^{-1} | 7.36×10^{-1} |
| 4 | 0.8781 | 0.8781 | 3.5326 | 0.8783 | 0.8775 | 3.8049 | 2.72×10^{-1} | 6.60×10^{-1} | 0.8774 | 0.8765 | 4.2050 | 6.72×10^{-1} | 1.79×10^{-3} |

表 5 $a = 0.5$ 时 PMB 算法针对 TF1 中的 f_1 在不同噪声环境下的全局极值点优化数值结果Table 5 Numerical results of global extremum points for f_1 of TF1 searched by PMB algorithm with $a = 0.5$ under different noise environments

| 极值点 序号 | P-MQHOA ^[1] 无噪声 | | | PMB | | | | | | | | | |
|-----------|----------------------------|---------|--------|-------------------|---------|--------|-----------------------|-----------------------|-------------------|---------|--------|-----------------------|-----------------------|
| | x_1 | x_2 | f | $\sigma^2 = 0.01$ | | | | | $\sigma^2 = 0.05$ | | | | |
| | | | | x_1 | x_2 | f | P_f | P_l | x_1 | x_2 | f | P_f | P_l |
| 1 | -0.8781 | -0.8781 | 3.5326 | -0.8760 | -0.8780 | 3.6923 | 1.60×10^{-1} | 2.07×10^{-3} | -0.8805 | -0.8775 | 3.9568 | 4.24×10^{-1} | 2.46×10^{-3} |
| 2 | -0.8781 | 0.8781 | 3.5326 | -0.8771 | 0.8782 | 3.6953 | 1.63×10^{-1} | 1.03×10^{-3} | -0.8770 | 0.8803 | 3.9510 | 4.18×10^{-1} | 2.50×10^{-3} |
| 3 | 0.8781 | -0.8781 | 3.5326 | 0.8775 | -0.8794 | 3.6847 | 1.52×10^{-1} | 1.05×10^{-3} | 0.8772 | -0.8788 | 3.9467 | 4.14×10^{-1} | 1.14×10^{-3} |
| 4 | 0.8781 | 0.8781 | 3.5326 | 0.8788 | 0.8784 | 3.6917 | 1.59×10^{-1} | 8.18×10^{-4} | 0.8775 | 0.8756 | 3.9697 | 4.37×10^{-1} | 2.52×10^{-3} |

图 15 $a = 0.2$, PMB 算法在不同噪声环境下对 TF1 中的 f_1 求解所有极值点的定位误差Fig. 15 Positioning errors of total-extremum points for f_1 of TF1 searched by PMB algorithm with $a = 0.2$ under different noise environments

PSO 算法在噪声环境中寻优的准确性, 首先给出了无噪声环境下的优化结果, 然后与噪声环境下的优化结果进行了比较. 实验中改进的斐波那契法的求解精度 ε 设置为 1×10^{-5} , 针对 CEC2013 的 5 维测试函数采用 5 级划分策略.

1) 无噪声环境下的比较

PMB 算法和 PSO 算法在无噪声环境下对 TF2 测试函数进行寻优, 获得的优化结果列于表 10 中, 其中 t 是完成一次实验的时间.

从表 10 中可以得出以下结论:

a) 从函数值的角度来看, PMB 对 24 个测试函数得到的函数最小值更接近这些函数的全局最小值, 精度高于 PSO. PMB 得到的这 24 个测试函数的数值结果已在表中用粗体标示出. PSO 仅在 4 个函数上得到的极值比 PMB 更接近函数的全局最小值, 在其他 24 个函数上陷入了局部最优. 且 PSO 在单峰函数 f_2 、多峰函数 $f_3 \sim f_{15}$ 和组合函数 $f_{21} \sim f_{28}$ 上得到的极值点的适应度值偏离函数的全局最小值较大, 在表中已用下划线标示出. 其中, 在函数

f_2 上的偏离值为 691.6966. 这说明 PSO 算法很容易陷入局部最优, 而 PMB 算法具有很好的全局寻优性能, 不容易陷入局部最优.

b) 对比两个算法运行的时间 t 可以看出, 为了跳出局部最优和保留多个极值点, PMB 算法需要花费比 PSO 算法更多的计算时间.

2) $\sigma^2 = 0.01$ 的噪声环境下比较

PMB 算法和 PSO 算法在 $\sigma^2 = 0.01$ 的噪声环境下对 TF2 测试函数进行寻优, 获得的优化结果分别列于表 11 和表 12 中.

对比表 11 和表 12 中可以得出以下结论:

a) 通过比较 P_l 的数值结果, 可以看出 PMB 在 25 个测试函数上得到的极值点的 P_l 小于 PSO, 这 25 个数值结果已在表 11 中用粗体标示出. 这表明 PMB 在噪声环境和无噪声环境下获得的极点位置偏差很小, 说明 PMB 算法具有良好的稳定性和抗噪性能. 然而, PSO 在噪声环境中获得的结果与在无噪声环境中获得的结果差距较大, 其中在 20 个测试函数上的 P_l 值超过了 10, 在表 11 中已经用下划线标示出. 这表明 PSO 的稳定性差, 这使得它易受噪声干扰, 抗噪性能较差.

b) 对比两个算法的运行时间 t 可以看出, 在噪声环境中, PMB 需要花费更多的时间来跳出由噪声引起的局部最优. PSO 的收敛速度在噪声的影响下也会减慢, 但往往会收敛到噪声引起的局部最优.

3) 多极值点寻优结果

PMB 算法可以在噪声环境中实现多极值点优化. 对于多峰函数 $f_6, f_7, f_8, f_{16}, f_{19}$ 和 f_{20} , PMB 算法得到的多个极值点的数值结果列于表 13 中. 由于篇幅有限, 每个函数只列出 5 个极值点的数值结果.

从表 13 中可以看出, PMB 算法可以在无噪声环境和噪声环境中找到多峰函数的多个极值点, 说明 PMB 算法可以在噪声环境中实现多维、多峰函数的多极值点优化.

表 6 P-MQHOA^[1] 算法与 $a = 0.2$ 时的 PMB 算法针对 TF1 中的 f_1 在不同环境下的所有极值点求解结果对比
 Table 6 Comparison of total-extremum points for f_1 of TF1 searched by P-MQHOA^[1] algorithm and PMB algorithm with $a = 0.2$ under different noise environments

| 极值点序号 | P-MQHOA ^[1] 无噪声 | | | PMB | | | | | |
|-------|----------------------------|---------|---------------|-------------------|---------|---------------|-------------------|---------|---------------|
| | | | | $\sigma^2 = 0.01$ | | | $\sigma^2 = 0.09$ | | |
| | x_1 | x_2 | $f(x_1, x_2)$ | x_1 | x_2 | $f(x_1, x_2)$ | x_1 | x_2 | $f(x_1, x_2)$ |
| 1 | -0.8781 | -0.8781 | 3.5326 | -0.8781 | -0.8780 | 3.8057 | -0.8762 | -0.8777 | 4.4627 |
| 2 | -0.8781 | -0.5269 | 3.0421 | -0.8778 | -0.5263 | 3.3069 | -0.8774 | -0.5285 | 3.9294 |
| 3 | -0.8781 | -0.1756 | 2.7969 | -0.8784 | -0.1746 | 3.0482 | -0.8778 | -0.1818 | 3.6818 |
| 4 | -0.8781 | 0.1756 | 2.7969 | -0.8789 | 0.1763 | 3.0605 | -0.8790 | 0.1814 | 3.7123 |
| 5 | -0.8781 | 0.5269 | 3.0421 | -0.8788 | 0.5277 | 3.3100 | -0.8761 | 0.5258 | 3.9341 |
| 6 | -0.8781 | 0.8781 | 3.5326 | -0.8789 | 0.8784 | 3.8027 | -0.8798 | 0.8774 | 4.4562 |
| 7 | -0.5269 | -0.8781 | 3.0421 | -0.5276 | -0.8780 | 3.3047 | -0.5264 | -0.8766 | 3.9330 |
| 8 | -0.5269 | -0.5269 | 2.5517 | -0.5266 | -0.5267 | 3.4294 | -0.52621 | -0.5269 | 2.7918 |
| 9 | -0.5269 | -0.1756 | 2.3065 | -0.5264 | -0.1756 | 2.5465 | -0.5278 | -0.1798 | 3.1998 |
| 10 | -0.5269 | 0.1756 | 2.3065 | -0.5263 | 0.1760 | 2.5499 | -0.5289 | 0.1772 | 3.1915 |
| 11 | -0.5269 | 0.5269 | 2.5517 | -0.5268 | 0.5256 | 2.7924 | -0.526 | 0.5278 | 3.4282 |
| 12 | -0.5269 | 0.8781 | 3.0421 | -0.5268 | 0.8775 | 3.3045 | -0.5271 | 0.8770 | 3.9228 |
| 13 | -0.1756 | -0.8781 | 2.7969 | -0.1770 | -0.8786 | 3.0600 | -0.1745 | -0.8800 | 3.6858 |
| 14 | -0.1756 | -0.5269 | 2.3065 | -0.1761 | -0.5275 | 2.5472 | -0.1792 | -0.5261 | 3.1804 |
| 15 | -0.1756 | -0.1756 | 2.0613 | -0.1772 | -0.1755 | 2.2932 | -0.1775 | -0.1789 | 2.9170 |
| 16 | -0.1756 | 0.1756 | 2.0613 | -0.1765 | 0.1746 | 2.2879 | -0.1755 | 0.1774 | 2.9344 |
| 17 | -0.1756 | 0.5269 | 2.3065 | -0.1763 | 0.5278 | 2.5418 | -0.1757 | 0.5250 | 3.1700 |
| 18 | -0.1756 | 0.8781 | 2.7969 | -0.1756 | 0.8791 | 3.0500 | -0.1784 | 0.8817 | 3.6725 |
| 19 | 0.1756 | -0.8781 | 2.7969 | 0.1762 | -0.8773 | 3.0574 | 0.1769 | -0.8757 | 3.6846 |
| 20 | 0.1756 | -0.5269 | 2.3065 | 0.1762 | -0.5266 | 2.5532 | 0.1794 | -0.5270 | 3.1708 |
| 21 | 0.1756 | -0.1756 | 2.0613 | 0.1756 | -0.1779 | 2.2964 | 0.1802 | -0.1758 | 2.9292 |
| 22 | 0.1756 | 0.1756 | 2.0613 | 0.1756 | 0.1752 | 2.2921 | 0.1790 | 0.1766 | 2.9235 |
| 23 | 0.1756 | 0.5269 | 2.3065 | 0.1752 | 0.5269 | 2.5441 | 0.1820 | 0.5250 | 3.1637 |
| 24 | 0.1756 | 0.8781 | 2.7969 | 0.1756 | 0.8800 | 3.0573 | 0.1803 | 0.8784 | 3.6873 |
| 25 | 0.5269 | -0.8781 | 3.0421 | 0.5253 | -0.8767 | 3.2985 | 0.5250 | -0.8808 | 3.9639 |
| 26 | 0.5269 | -0.5269 | 2.5517 | 0.5254 | -0.5261 | 2.8010 | 0.5256 | -0.5244 | 3.4443 |
| 27 | 0.5269 | -0.1756 | 2.3065 | 0.5275 | -0.1752 | 2.5432 | 0.5266 | -0.1785 | 3.1973 |
| 28 | 0.5269 | 0.1756 | 2.3065 | 0.5266 | 0.1775 | 2.5502 | 0.5291 | 0.1789 | 3.1769 |
| 29 | 0.5269 | 0.5269 | 2.5517 | 0.5273 | 0.5259 | 2.7941 | 0.5277 | 0.5257 | 3.4395 |
| 30 | 0.5269 | 0.8781 | 3.0421 | 0.5254 | 0.8776 | 3.3128 | 0.5242 | 0.8817 | 3.9135 |
| 31 | 0.8781 | -0.8781 | 3.5326 | 0.8782 | -0.8778 | 3.8086 | 0.8793 | -0.8759 | 4.4479 |
| 32 | 0.8781 | -0.5269 | 3.0421 | 0.8792 | -0.5265 | 3.3119 | 0.8771 | -0.5288 | 3.9199 |
| 33 | 0.8781 | -0.1756 | 2.7969 | 0.8778 | -0.1766 | 3.0594 | 0.8774 | -0.1782 | 3.7063 |
| 34 | 0.8781 | 0.1756 | 2.7969 | 0.8780 | 0.1760 | 3.0549 | 0.8772 | 0.1784 | 3.6981 |
| 35 | 0.8781 | 0.5269 | 3.0421 | 0.8784 | 0.5265 | 3.3001 | 0.8788 | 0.5279 | 3.3046 |
| 36 | 0.8781 | 0.8781 | 3.5326 | 0.8783 | 0.8775 | 3.8049 | 0.8803 | 0.8792 | 4.4365 |

综上所述:

a) PMB 算法可以在无噪声和噪声环境中准确定位测试函数的真正全局极值点, 反映了 PMB 算法良好的稳定性和抗噪性能. PSO 算法在无噪声环

境中容易陷入局部最优, 而在噪声环境中容易陷入由噪声引起的局部最优. 因此, PSO 算法的稳定性和抗噪性能较差.

b) PSO 算法运行一次只能获得一个极值点,

表 7 PMB 算法对 TF1 中的 $f_2 \sim f_6$ 在不同噪声环境下的全局极值点优化数值结果
 Table 7 Numerical results of global extremum points for function $f_2 \sim f_6$ of TF1 searched by PMB algorithm under different noise environments

| 函数 | 全局极值 | 指标 | IBA ^[32] | | | | PMB | | | |
|-------|-----------|------|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| | | | $\sigma^2 = 0.01$ | $\sigma^2 = 0.05$ | $\sigma^2 = 0.07$ | $\sigma^2 = 0.09$ | $\sigma^2 = 0.01$ | $\sigma^2 = 0.05$ | $\sigma^2 = 0.07$ | $\sigma^2 = 0.09$ |
| f_2 | -176.1375 | Mean | -176.1830 | -176.3646 | -176.5443 | -176.5442 | -176.1607 | -176.0738 | -176.0689 | -175.9722 |
| | | MAPE | 0.0258 | 0.1289 | 0.1799 | 0.2309 | 0.0132 | 0.0362 | 0.0389 | 0.0938 |
| f_3 | -1.8013 | Mean | -1.8468 | -2.0283 | -2.1206 | -2.2095 | -2.0007 | -2.3048 | -2.4433 | -2.5351 |
| | | MAPE | 2.5281 | 12.6022 | 17.7255 | 22.6617 | 11.0675 | 27.9516 | 35.6393 | 40.7363 |
| f_4 | 0.9 | Mean | 0.8556 | 0.6838 | 0.6053 | 0.5330 | 0.6534 | 0.2620 | 0.1312 | 0.0529 |
| | | MAPE | 4.9308 | 24.0261 | 32.7394 | 46.5648 | 27.4009 | 70.8921 | 85.4263 | 94.1181 |
| f_5 | -24.1568 | Mean | -24.2016 | -24.3806 | -24.4665 | -24.5567 | -24.2903 | -24.538 | -24.5977 | -24.6518 |
| | | MAPE | 0.1856 | 0.9266 | 1.2822 | 1.6556 | 0.5526 | 1.5781 | 1.8253 | 2.0490 |
| f_6 | -19.2085 | Mean | -19.253 | -19.4329 | -19.5236 | -19.6095 | -19.3724 | -19.6408 | -19.7391 | -19.8169 |
| | | MAPE | 0.2319 | 1.1683 | 1.6403 | 2.0875 | 0.8533 | 2.2504 | 2.7625 | 3.1673 |

表 8 PMB 算法对 TF1 中 $f_2 \sim f_4$ 在不同噪声环境下得到的全局极值点的定位误差
 Table 8 Positioning errors of global extremum points for function $f_2 \sim f_4$ of TF1 searched by PMB algorithm under different noise environments

| 函数 | σ^2 | IBA ^[32] | PMB | | |
|-------|------------|---------------------|---------|---------|---------------|
| | | P_l | x_1 | x_2 | P_l |
| f_2 | 0.01 | 0.00082301 | -1.3062 | -1.4246 | 0.0007 |
| | 0.05 | 0.0019 | -1.3064 | -1.4241 | 0.0009 |
| | 0.07 | 0.0022 | -1.3079 | -1.4244 | 0.0011 |
| | 0.09 | 0.0023 | -1.3041 | -1.4252 | 0.0027 |
| f_3 | 0.01 | 0.0540 | 2.2088 | 1.5727 | 0.0059 |
| | 0.05 | 0.0132 | 2.1913 | 1.5723 | 0.0119 |
| | 0.07 | 0.0160 | 2.1882 | 1.5664 | 0.0156 |
| | 0.09 | 0.0201 | 2.1833 | 1.5710 | 0.0198 |
| f_4 | 0.01 | 0.0282 | 0.0611 | 0.0698 | 0.0927 |
| | 0.05 | 0.0830 | 0.0836 | 0.0623 | 0.1042 |
| | 0.07 | 1.1067 | 0.0820 | 0.0655 | 0.1049 |
| | 0.09 | 2.2026 | 0.0579 | 0.0934 | 0.1099 |

表 9 PMB 算法对 TF1 中的 $f_5 \sim f_6$ 在不同噪声环境下的全局极值点优化数值结果
 Table 9 Numerical results of global extremum points for function $f_5 \sim f_6$ of TF1 searched by PMB algorithm under different noise environments

| 函数 | 算法 | IBA ^[32] | | | | PMB | | | | |
|-------|------------|---------------------------|---------------------|---------------|---------------------|-------------------|---------------------|---------------|---------------------|---------------|
| | σ^2 | 0.01 | 0.05 | 0.07 | 0.09 | 0.01 | 0.05 | 0.07 | 0.09 | |
| | 极值点序号 | P_l | | | | $(x_1, x_2), P_l$ | | | | |
| f_5 | 1 | 0.00480.01510.01790.0206 | (9.6453, -9.6439), | 0.0030 | (9.6498, -9.6444), | 0.0039 | (9.6398, -9.6475), | 0.0068 | (9.6559, -9.6482), | 0.0094 |
| | 2 | 0.00570.0162 0.016 0.0158 | (-9.6457, 9.6444), | 0.0023 | (-9.6473, 9.6442), | 0.0025 | (-9.6520, 9.6450), | 0.0057 | (-9.6473, 9.6401), | 0.0065 |
| | 3 | 0.006 0.01120.01490.0153 | (-9.6454, -9.6455), | 0.0016 | (-9.6460, -9.6413), | 0.0053 | (-9.6427, -9.6406), | 0.0072 | (-9.6407, -9.6419), | 0.0076 |
| | 4 | 0.00440.01330.01510.0178 | (9.6476, 9.6468), | 0.0010 | (9.6477, 9.6528), | 0.0063 | (9.6493, 9.6532), | 0.0071 | (9.6434, 9.6390), | 0.0082 |
| f_6 | 1 | 0.01070.02150.02380.0291 | (8.0518, -9.6675), | 0.0043 | (8.0521, -9.6712), | 0.0072 | (8.0462, -9.6588), | 0.0106 | (8.0525, -9.6534), | 0.0115 |
| | 2 | 0.00940.02070.02620.0289 | (-8.0541, 9.6626), | 0.0022 | (-8.0532, 9.6629), | 0.0025 | (-8.0561, 9.6614), | 0.0034 | (-8.0586, 9.6567), | 0.0087 |
| | 3 | 0.00880.01990.02420.0243 | (-8.0513, -9.6634), | 0.0039 | (-8.0541, -9.6590), | 0.0057 | (-8.0595, -9.6698), | 0.0069 | (-8.0602, -9.6566), | 0.0096 |
| | 4 | 0.00810.0241 0.021 0.0238 | (8.0529, 9.6611), | 0.0041 | (8.0503, 9.6688), | 0.0063 | (8.0547, 9.6757), | 0.0111 | (8.0642, 9.6743), | 0.0134 |

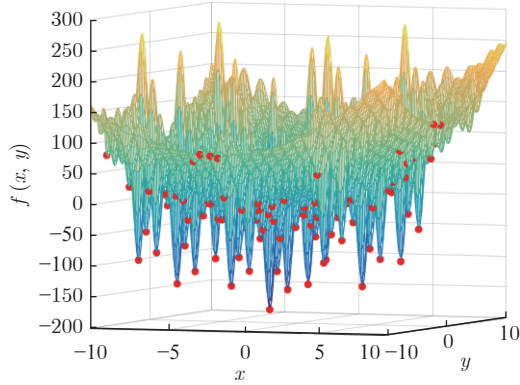


图 16 $a = 1, \sigma^2 = 0.05$, PMB 算法对 TF1 中 f_2 的寻优结果

Fig.16 Optimization results for f_2 of TF1 by PMB algorithm in noise environment with $\sigma^2 = 0.05$ and $a = 1$

而 PMB 算法运行一次可以获得多个极值点. 表明

PMB 算法可以在噪声环境下找到多维、多峰函数的多个全局和局部极值点.

c) 为了跳出局部最优和保留多个极值点, PMB 算法需要花费比 PSO 算法更多的计算时间. PSO 的收敛速度在噪声的影响下也会减慢, 但它更容易收敛到局部极值点.

4 结论

针对噪声环境下的多极值点寻优, 本文采用一种定概率的方法来解决这类随机优化问题. 从蒲丰投针的定概率原理出发, 提出了噪声环境下基于蒲丰距离的依概率多峰优化算法 (PMB). 理论推导证明了蒲丰距离和极值分辨率依概率影响算法的峰值检测率. 验证了在不同的噪声条件下, PMB 算法依概率收敛到函数的多个极值点, 且算法找到全局极值点概率高于找到局部极值点的概率; 验证了蒲丰距离影响 PMB 算法的全极值点寻优性能和求解极

表 10 PMB 算法和 PSO 算法对 TF2 在无噪声环境下得到的全局极值点

Table 10 Global extremum points of TF2 obtained by PMB algorithm and PSO algorithm in noiseless environment

| TF2 | Opti- mum | PSO | | | | | | | PMB | | | | | | |
|----------|--------------|------------------|----------|----------|----------|----------|----------|---------|-------------------|----------|---------|----------|---------|----------|-----------|
| | | f | x_1 | x_2 | x_3 | x_4 | x_5 | $t(s)$ | f | x_1 | x_2 | x_3 | x_4 | x_5 | $t(s)$ |
| f_1 | -1 400 | -1400.5153 | -22.003 | 11.6561 | -36.1216 | 69.3583 | -37.5101 | 1.2932 | -1399.9755 | -22.1013 | 11.5532 | -35.9107 | 69.3869 | -37.6351 | 4.024 |
| f_2 | -1 300 | <u>-608.3034</u> | -23.2765 | 15.9243 | -18.1464 | 70.6333 | -35.3268 | 1.586 | -1284.0072 | -22.1909 | 10.6542 | -37.981 | 69.5946 | -37.5306 | 13.084 |
| f_3 | -1 200 | -1191.9927 | -90.9013 | 11.4835 | -35.8577 | 68.8415 | -40.1122 | 1.3798 | -1199.4865 | -32.8088 | 11.5433 | -35.9866 | 69.289 | -38.0021 | 30.5866 |
| f_4 | -1 100 | -1100.2332 | -22.0721 | 11.3036 | -35.9249 | 69.1684 | -37.6371 | 1.4573 | -1098.4189 | -22.0954 | 11.9348 | -36.1631 | 68.5683 | -37.4961 | 52.0969 |
| f_5 | -1 000 | -1000.4002 | -21.9647 | 11.3823 | -36.0354 | 69.0788 | -37.36 | 1.326 | -999.9999 | -21.9848 | 11.5549 | -36.0044 | 69.3447 | -37.6114 | 73.5208 |
| f_6 | -900 | -900.1464 | -48.0332 | 16.1924 | -31.6003 | 85.6123 | -64.8171 | 1.2566 | -900.0000 | -22.0309 | 11.5686 | -36.0034 | 69.4094 | -37.6539 | 6.7166 |
| f_7 | -800 | -800.314 | -47.6006 | 11.5431 | -35.9756 | 69.3023 | -37.8813 | 1.9918 | -799.9792 | -22.0032 | 11.5635 | -36.004 | 69.3839 | -37.5808 | 24.2937 |
| f_8 | -700 | <u>-680.2271</u> | -89.9164 | 95.3877 | 55.5365 | -63.3428 | -20.1823 | 1.6314 | -699.9191 | -22.2758 | 11.5436 | -36.0022 | 69.3631 | -37.6026 | 44.2769 |
| f_9 | -600 | -600.3439 | 75.0307 | 21.2084 | -65.4178 | -90.6598 | 25.5987 | 10.7053 | -599.9286 | -22.0921 | 11.5053 | -35.998 | 69.3653 | -37.6013 | 137.0404 |
| f_{10} | -500 | -500.3287 | -22.7446 | 11.9779 | -35.0425 | 69.3475 | -37.3164 | 1.6048 | -499.8497 | -21.6232 | 11.8265 | -36.1124 | 69.0713 | -37.5719 | 164.8766 |
| f_{11} | -400 | -400.4644 | -21.9891 | 11.5276 | -36.014 | 69.3475 | -37.5829 | 1.8479 | -399.54 | -21.6252 | 11.9614 | -35.9928 | 69.484 | -37.3817 | 203.6023 |
| f_{12} | -300 | -299.4175 | -19.0411 | 5.6211 | -50.5124 | 72.7959 | -42.8024 | 1.9517 | -299.2154 | -22.224 | 11.6704 | -35.3449 | 68.6809 | -37.0768 | 244.3073 |
| f_{13} | -200 | -199.3823 | -31.2247 | 9.7199 | -34.7315 | 78.1449 | -45.3504 | 1.8811 | -199.7728 | -22.0876 | 11.6491 | -35.5865 | 69.2094 | -37.2746 | 288.9675 |
| f_{14} | -100 | -100.0015 | -6.2047 | 11.5466 | -27.1106 | 69.3496 | -32.6014 | 1.5652 | -99.8081 | -22.0000 | 11.4699 | -36.0043 | 69.3659 | -37.622 | 25.5247 |
| f_{15} | 100 | <u>156.5509</u> | 49.6019 | 63.3034 | -72.268 | -3.7613 | -79.9879 | 1.6973 | 110.4386 | -21.8498 | 11.9445 | -35.3707 | 69.3721 | -37.4593 | 45.1034 |
| f_{16} | 200 | 199.8538 | -57.0813 | 58.5504 | -31.0707 | 26.9993 | -38.2225 | 3.9518 | 200.4084 | -22.0509 | 11.5686 | -36.0034 | 69.4094 | -37.6539 | 31.5485 |
| f_{17} | 300 | 304.7026 | 8.0018 | -16.5642 | -5.9498 | 38.2561 | -8.1092 | 1.5401 | 300.3612 | -22.0466 | 11.5575 | -37.5698 | 69.3452 | -37.6059 | 114.3228 |
| f_{18} | 400 | 405.3374 | 8.8635 | -22.2516 | -5.391 | 38.826 | -10.5181 | 1.6256 | 401.5735 | -22.269 | 12.4883 | -38.1444 | 68.9858 | -37.9558 | 144.0145 |
| f_{19} | 500 | 499.8491 | -36.1124 | -7.1141 | -60.5984 | 57.0958 | -63.211 | 1.4022 | 500.0023 | -22.1995 | 11.4599 | -36.2687 | 68.9998 | -37.9225 | 175.7896 |
| f_{20} | 600 | 599.7283 | -43.4837 | 14.2779 | -40.0212 | 72.9036 | -43.5544 | 1.4423 | 600.0068 | -22.0109 | 11.5786 | -36.0034 | 69.4014 | -37.6519 | 214.3663 |
| f_{21} | 700 | <u>999.5585</u> | 74.8671 | 7.56 | 60.3875 | 15.7259 | 31.6655 | 1.8884 | 709.2098 | -22.1135 | 11.7078 | -36.2284 | 69.2862 | -37.5341 | 259.2839 |
| f_{22} | 800 | <u>899.5949</u> | -48.5424 | 53.7715 | 13.7283 | 69.8239 | -18.6229 | 2.6328 | 821.3446 | -22.0219 | 12.2169 | -36.155 | 69.2986 | -37.4481 | 338.1721 |
| f_{23} | 900 | <u>1040.6959</u> | -47.8633 | 59.6935 | 18.944 | 63.6605 | -16.7946 | 2.5248 | 926.4912 | -22.2323 | 11.3133 | -36.2082 | 69.665 | -37.8375 | 400.5942 |
| f_{24} | 1 000 | <u>1104.3212</u> | -43.5817 | 62.9002 | 24.6363 | 69.2317 | -19.6422 | 11.4319 | 1005.5802 | -22.2796 | 11.6315 | -36.0045 | 69.656 | -37.0404 | 536.6315 |
| f_{25} | 1 100 | <u>1199.5464</u> | -48.5541 | 53.8377 | 13.7864 | 69.8039 | -18.5822 | 11.4093 | 1103.9827 | -22.2144 | 11.3766 | -35.7601 | 69.5939 | -37.2318 | 674.0186 |
| f_{26} | 1 200 | <u>1230.9793</u> | -45.3432 | 36.5543 | -61.2965 | 73.0248 | -28.7484 | 11.8107 | 1201.2061 | -21.9038 | 11.8136 | -36.0593 | 69.1101 | -37.8049 | 824.5081 |
| f_{27} | 1 300 | <u>1643.8584</u> | 39.9937 | 64.0666 | 74.3787 | 82.5407 | 88.9271 | 11.4554 | 1316.2217 | -22.0002 | 11.5006 | -36.0034 | 69.4000 | -37.6006 | 977.0531 |
| f_{28} | 1 400 | <u>1699.6997</u> | 74.858 | 7.5684 | 60.4032 | 15.7134 | 31.6678 | 2.8537 | 1404.1964 | -21.9492 | 11.4562 | -36.0421 | 69.5111 | -37.5802 | 1061.4819 |

表 11 PSO 算法对 TF2 在 $\sigma^2 = 0.01$ 的噪声环境下得到的全局极值点
 Table 11 Global extremum points of TF2 obtained by PSO algorithm in noise environment of $\sigma^2 = 0.01$

| TF2 | Optimum | f | x_1 | x_2 | x_3 | x_4 | x_5 | t (s) | P_f | P_t |
|----------|---------|-------------|----------|----------|----------|----------|----------|---------|----------|-----------------|
| f_1 | -1 400 | -1 400.4781 | -22.0184 | 11.4865 | -35.9794 | 69.3581 | -37.6173 | 1.404 | 0.0372 | 0.2464 |
| f_2 | -1 300 | -930.7041 | -20.245 | 15.2674 | -30.6074 | 67.6185 | -38.8701 | 1.4961 | 322.4007 | <u>13.6581</u> |
| f_3 | -1 200 | -1 199.0062 | -50.622 | 11.5252 | -35.947 | 69.152 | -38.649 | 1.2854 | 7.0135 | <u>40.3071</u> |
| f_4 | -1 100 | -1 100.2794 | -21.9225 | 11.64 | -36.0456 | 69.5782 | -37.4029 | 1.4142 | 0.0462 | 0.6106 |
| f_5 | -1 000 | -1 000.3961 | -21.994 | 11.5758 | -36.1201 | 69.1453 | -37.9101 | 1.2586 | 0.0041 | 0.5937 |
| f_6 | -900 | -900.3562 | -31.8677 | 13.59 | -34.0754 | 76.8813 | -49.9714 | 1.218 | 0.2098 | <u>23.8924</u> |
| f_7 | -800 | -800.3509 | -22.0771 | 11.5961 | -35.183 | 69.3788 | -37.6198 | 1.8579 | 0.0369 | <u>25.5374</u> |
| f_8 | -700 | -680.3089 | -71.3566 | -56.4316 | 57.2977 | -28.3447 | -56.5965 | 1.615 | 0.0817 | <u>161.0824</u> |
| f_9 | -600 | -599.2936 | 49.9955 | -78.2402 | 0.1786 | 32.7815 | 56.5152 | 10.4697 | 1.0503 | <u>176.1058</u> |
| f_{10} | -500 | -500.2789 | -22.3318 | 11.3539 | -34.8006 | 70.2354 | -36.8967 | 1.5857 | 0.0497 | 1.2581 |
| f_{11} | -400 | -400.4319 | -21.9215 | 11.6111 | -36.0095 | 69.3882 | -37.6099 | 1.822 | 0.0325 | 0.1181 |
| f_{12} | -300 | -300.4235 | -21.962 | 11.6181 | -35.9548 | 69.2314 | -37.4736 | 1.8597 | 1.006 | <u>17.2487</u> |
| f_{13} | -200 | -195.7745 | -16.5439 | 0.7856 | -63.1612 | 78.2956 | -49.8104 | 1.8291 | 3.6078 | <u>33.5188</u> |
| f_{14} | -100 | -76.8796 | 93.8116 | 23.4253 | -27.1196 | 69.3665 | -37.609 | 1.5651 | 23.1219 | <u>100.8436</u> |
| f_{15} | 100 | 162.9651 | 63.763 | 73.9758 | -59.6899 | 6.8698 | -69.5651 | 1.6225 | 6.4142 | <u>26.3496</u> |
| f_{16} | 200 | 200.0371 | -77.2249 | 85.4895 | -99.6791 | 9.5552 | 0.5678 | 3.677 | 0.1834 | <u>87.4504</u> |
| f_{17} | 300 | 305.138 | 8.0174 | -22.2004 | -9.134 | 41.8141 | -7.1107 | 1.5252 | 0.4354 | 7.4541 |
| f_{18} | 400 | 406.2551 | 10.8862 | -24.2858 | -0.1323 | 39.2559 | -7.5586 | 1.5572 | 0.9177 | 6.6953 |
| f_{19} | 500 | 499.7671 | -26.9034 | 2.4016 | -45.2641 | 65.0955 | -40.2783 | 1.3468 | 0.082 | <u>31.6291</u> |
| f_{20} | 600 | 599.6336 | -71.294 | 14.1417 | -38.0582 | 69.6662 | -29.0219 | 1.4203 | 0.0947 | <u>31.6063</u> |
| f_{21} | 700 | 799.646 | -48.5365 | 53.7646 | 13.7222 | 69.8263 | -18.6266 | 1.8519 | 199.9126 | <u>158.1048</u> |
| f_{22} | 800 | 1 016.1259 | 95.0903 | -70.2239 | -51.7248 | 71.7396 | -52.0955 | 2.4272 | 116.5309 | <u>203.5029</u> |
| f_{23} | 900 | 1 066.8577 | -61.6704 | 86.9732 | 42.3647 | 41.522 | -18.9232 | 2.4921 | 26.1619 | <u>44.4746</u> |
| f_{24} | 1 000 | 1 111.4644 | -61.8977 | 82.0776 | 36.3207 | 76.7673 | 1.768 | 11.6881 | 7.1432 | <u>36.8097</u> |
| f_{25} | 1 100 | 1 199.6718 | -48.5853 | 53.7148 | 13.7168 | 69.7682 | -18.5628 | 11.7015 | 0.1254 | 0.1502 |
| f_{26} | 1 200 | 1 302.6409 | -71.6478 | 58.3538 | 19.0587 | 68.5734 | -24.1115 | 12.4385 | 71.6616 | <u>87.5525</u> |
| f_{27} | 1 300 | 1 599.7342 | 74.8682 | 7.5586 | 60.3873 | 15.7236 | 31.6624 | 12.1106 | 44.1241 | <u>111.1257</u> |
| f_{28} | 1 400 | 1 499.6147 | -48.5389 | 53.7644 | 13.7202 | 69.8289 | -18.6285 | 2.8779 | 200.085 | <u>158.1087</u> |

表 12 PMB 算法对 TF2 在 $\sigma^2 = 0.01$ 的噪声环境下得到的全局极值点
 Table 12 Global extremum points of TF2 obtained by PMB algorithm in noise environment of $\sigma^2 = 0.01$

| TF2 | Optimum | f | x_1 | x_2 | x_3 | x_4 | x_5 | t (s) | P_f | P_t |
|----------|---------|-------------|----------|---------|----------|---------|----------|----------|----------|---------------|
| f_1 | -1 400 | -1 400.0699 | -21.8287 | 11.8038 | -36.3299 | 69.5330 | -37.6233 | 12.7743 | 0.0943 | 0.5782 |
| f_2 | -1 300 | -608.3307 | -23.1175 | 11.4652 | -34.0409 | 70.2874 | -36.6119 | 39.0930 | 675.6766 | 4.2854 |
| f_3 | -1 200 | -1 017.2976 | -32.7028 | 11.5533 | -35.9896 | 69.2990 | -38.0010 | 68.3085 | 182.1889 | 0.1070 |
| f_4 | -1 100 | -1 054.7452 | -22.0054 | 11.9548 | -36.1731 | 68.5083 | -37.3461 | 102.1388 | 43.6737 | 0.1863 |
| f_5 | -1 000 | -1 000.2789 | -21.9308 | 11.4701 | -36.3134 | 69.5171 | -37.2096 | 135.1712 | 0.2791 | 0.5448 |
| f_6 | -900 | -900.0006 | -22.1408 | 11.3658 | -36.1004 | 69.2084 | -37.6539 | 12.8520 | 0.0006 | 0.3210 |
| f_7 | -800 | -800.0919 | -21.9418 | 11.6135 | -35.8100 | 69.3635 | -37.6909 | 40.6402 | 0.1127 | 0.2376 |
| f_8 | -700 | -699.7605 | -22.1688 | 11.4203 | -36.5141 | 69.3235 | -37.5792 | 69.8338 | 0.1586 | 0.5393 |
| f_9 | -600 | -600.1833 | -21.8968 | 12.9478 | -32.0218 | 69.5171 | -37.6985 | 215.7291 | 0.2547 | 4.2381 |
| f_{10} | -500 | -499.9168 | -21.3850 | 11.1694 | -35.9969 | 69.3487 | -37.4039 | 251.9397 | 0.0671 | 0.7791 |
| f_{11} | -400 | -400.0491 | -21.9828 | 11.4866 | -35.7514 | 69.3762 | -37.4913 | 306.9611 | 0.5091 | 0.6597 |
| f_{12} | -300 | -299.9505 | -21.8073 | 11.5243 | -35.7096 | 69.1440 | -37.0312 | 358.0750 | 0.7351 | 0.7379 |
| f_{13} | -200 | -200.0731 | -21.8323 | 11.3919 | -36.2855 | 69.2937 | -37.4883 | 412.7622 | 0.3003 | 0.8202 |

表 12 PMB 算法对 TF2 在 $\sigma^2 = 0.01$ 的噪声环境下得到的全局极值点 (续表)

Table 12 Global extremum points of TF2 obtained by PMB algorithm in noise environment of $\sigma^2 = 0.01$ (continued)

| TF2 | Optimum | f | x_1 | x_2 | x_3 | x_4 | x_5 | $t(s)$ | P_f | P_l |
|----------|---------|------------|----------|---------|----------|---------|----------|------------|---------|---------------|
| f_{14} | -100 | -99.4329 | -22.1030 | 11.5419 | -36.1002 | 69.3419 | -37.6012 | 523.8452 | 0.3752 | 0.1612 |
| f_{15} | 100 | 104.4172 | -21.7108 | 11.3782 | -36.1444 | 69.3134 | -37.7273 | 58.0593 | 6.0214 | 1.0070 |
| f_{16} | 200 | 200.8154 | -22.1002 | 11.4383 | -36.1730 | 69.4102 | -37.6045 | 126.4925 | 0.4069 | 0.2250 |
| f_{17} | 300 | 301.8527 | -22.1000 | 11.6151 | -37.4783 | 69.4012 | -37.5859 | 366.6641 | 1.4915 | 0.1345 |
| f_{18} | 400 | 404.3400 | -22.2810 | 12.4912 | -38.0133 | 68.8766 | -37.9101 | 341.7595 | 2.7665 | 0.1771 |
| f_{19} | 500 | 500.1091 | -22.1404 | 11.2668 | -36.2592 | 68.9192 | -37.9133 | 318.0817 | 0.1068 | 0.2178 |
| f_{20} | 600 | 599.9330 | -22.0115 | 11.5642 | -35.9012 | 69.4102 | -37.6413 | 501.6370 | 0.0739 | 0.1041 |
| f_{21} | 700 | 704.7231 | -22.1409 | 11.5481 | -35.9884 | 69.3233 | -37.5984 | 799.2632 | 4.4867 | 0.2989 |
| f_{22} | 800 | 843.9745 | -22.0029 | 11.7082 | -36.0243 | 69.3881 | -38.1587 | 1 284.7595 | 22.6299 | 0.8884 |
| f_{23} | 900 | 923.2544 | -21.9358 | 11.4092 | -36.9182 | 69.2899 | -37.7927 | 1 405.1109 | 3.2368 | 0.8625 |
| f_{24} | 1 000 | 1 002.7871 | -22.0554 | 11.0213 | -36.4846 | 69.5489 | -37.5213 | 1 645.1116 | 2.7931 | 0.9464 |
| f_{25} | 1 100 | 1 100.8277 | -21.8449 | 11.4803 | -35.9915 | 69.3188 | -37.5424 | 1 886.3396 | 3.1550 | 0.6107 |
| f_{26} | 1 200 | 1 215.0429 | -21.8701 | 11.7852 | -37.0089 | 69.1101 | -37.1271 | 2 148.5756 | 13.8368 | 1.1675 |
| f_{27} | 1 300 | 1 379.5470 | -22.2113 | 11.7141 | -36.6134 | 69.4000 | -37.2465 | 2 412.4953 | 63.3253 | 0.7665 |
| f_{28} | 1 400 | 1 401.3007 | -21.9727 | 11.5578 | -36.0055 | 69.3242 | -37.6351 | 2 555.0831 | 2.8957 | 0.2240 |

值点的定位精度, 噪声强度影响算法的峰值检测率和定位精度; 对比噪声环境下的改进蝙蝠算法, PMB 算法具有更好的极值点定位精度和较好的多

极值点寻优特性; 对比 PSO 算法在 5 维 CEC2013 测试函数集上的表现, PMB 算法具有更好的多维函数寻优能力、抗噪声性能和多极值点寻优性能.

表 13 PMB 算法对 TF2 中部分函数在无噪声环境和 $\sigma^2 = 0.01$ 的噪声环境下的多极值点优化结果

Table 13 Optimization results of multiple extremum points for some functions of TF2 obtained by PMB algorithm in noiseless environment and noise environment of $\sigma^2 = 0.01$

| TF2 | Noiseless environment | | | | | $\sigma^2 = 0.01$ | | | | | P_f | P_l | | |
|----------|-----------------------|----------|----------|----------|----------|-------------------|-----------|----------|----------|----------|----------|----------|--------|--------|
| | f | x_1 | x_2 | x_3 | x_4 | x_5 | f | x_1 | x_2 | x_3 | | | x_4 | x_5 |
| f_6 | -900.0000 | -22.0309 | 11.5686 | -36.0034 | 69.4094 | -37.6539 | -900.0006 | -22.1408 | 11.3658 | -36.1004 | 69.2084 | -37.6539 | 0.0006 | 0.3210 |
| | -896.0639 | -33.8126 | 44.7393 | 54.6931 | 77.1452 | -50.0368 | -896.0683 | -34.4983 | 44.9367 | 54.8740 | 77.5853 | -50.9570 | 0.0044 | 1.2579 |
| | -899.9794 | -28.6913 | 12.8088 | -35.1466 | 74.6026 | -46.3440 | -900.2057 | -28.8650 | 12.9076 | -35.1465 | 74.7262 | -46.3359 | 0.2264 | 0.2351 |
| | -899.9959 | -19.0610 | 10.9852 | -36.4042 | 66.8717 | -33.4840 | -899.8525 | -19.1580 | 11.2422 | -36.3101 | 66.9289 | -32.6954 | 0.1433 | 0.8423 |
| | -899.9987 | -20.6166 | 11.3082 | -36.2410 | 68.2617 | -35.8258 | -899.9635 | -20.4386 | 11.2333 | -36.0719 | 68.0558 | -35.3752 | 0.0353 | 0.5580 |
| f_7 | -799.9792 | -22.0032 | 11.5635 | -36.0040 | 69.3839 | -37.5808 | -800.0919 | -21.9418 | 11.6135 | -35.8100 | 69.3635 | -37.6909 | 0.1127 | 0.2376 |
| | -799.8756 | -28.3647 | 11.5956 | -35.8878 | 69.3717 | -37.6959 | -800.0568 | -28.2097 | 11.5979 | -35.9973 | 69.4689 | -37.5283 | 0.1812 | 0.2712 |
| | -799.7114 | -40.2000 | 11.5118 | -35.9099 | 69.3766 | -37.8352 | -799.8792 | -40.1102 | 11.5317 | -35.9960 | 69.3786 | -37.8348 | 0.1678 | 0.1259 |
| | -799.6562 | -56.9976 | 11.5324 | -35.9848 | 69.1641 | -38.1372 | -799.8114 | -55.1614 | 11.5224 | -35.9948 | 69.1645 | -38.1372 | 0.1552 | 1.8363 |
| | -799.5697 | -66.1517 | 11.6099 | -35.9001 | 69.4026 | -37.9670 | -799.6939 | -65.2517 | 11.6094 | -35.9078 | 69.3432 | -37.9666 | 0.1242 | 0.9020 |
| f_8 | -699.9191 | -22.2758 | 11.5436 | -36.0022 | 69.3631 | -37.6026 | -699.7605 | -22.1688 | 11.4203 | -36.5141 | 69.3235 | -37.5792 | 0.1586 | 0.5393 |
| | -699.9042 | -24.1426 | 11.5593 | -36.0070 | 69.3609 | -37.3899 | -699.7605 | -24.1231 | 11.5323 | -35.9885 | 69.3697 | -37.3910 | 0.1437 | 0.0391 |
| | -699.6190 | -45.4600 | 11.5691 | -36.0014 | 84.1102 | -20.9700 | -699.7605 | -45.4382 | 11.5681 | -36.0127 | 84.1234 | -20.9690 | 0.1415 | 0.0279 |
| | -699.1928 | -36.3628 | 11.5985 | -35.9784 | 69.3447 | -36.1163 | -699.7605 | -36.4531 | 11.5900 | -35.9804 | 69.3459 | -36.0956 | 0.5676 | 0.0931 |
| | -699.4293 | -30.7647 | 11.5746 | -35.9425 | 69.3885 | -37.6095 | -699.7605 | -30.7591 | 11.5895 | -35.9425 | 69.4115 | -37.5944 | 0.3312 | 0.0318 |
| f_{16} | 200.4084 | -22.0509 | 11.5686 | -36.0034 | 69.4094 | -37.6539 | 200.8154 | -22.1002 | 11.4383 | -36.1730 | 69.4102 | -37.6045 | 0.4069 | 0.2250 |
| | 200.0320 | -52.1005 | -61.6883 | -32.3610 | -46.5936 | 1.0443 | 200.3819 | -52.1021 | -61.7593 | -32.3582 | -46.5732 | 1.1053 | 0.3499 | 0.0958 |
| | 200.0353 | -33.1330 | -38.1129 | 2.5053 | -47.4949 | 37.2437 | 200.1492 | -33.1432 | -38.1214 | 2.4901 | -47.4899 | 37.2348 | 0.1139 | 0.0226 |
| | 200.0370 | -25.6295 | 23.1822 | -3.7438 | -74.5210 | -22.2897 | 200.1477 | -25.6000 | 23.1812 | -3.7542 | -74.5312 | -22.2982 | 0.1107 | 0.0340 |
| | 200.0511 | 31.4036 | 49.5090 | 46.7470 | -62.4386 | -15.8403 | 200.0922 | 31.4125 | 49.4947 | 46.7173 | -62.4472 | -15.8504 | 0.0411 | 0.0366 |

表 13 PMB 算法对 TF2 中部分函数在无噪声环境和 $\sigma^2 = 0.01$ 的噪声环境下的多极值点优化结果 (续表)Table 13 Optimization results of multiple extremum points for some functions of TF2 obtained by PMB algorithm in noiseless environment and noise environment of $\sigma^2 = 0.01$ (continued)

| TF2 | Noiseless environment | | | | | | $\sigma^2 = 0.01$ | | | | | | | |
|----------|-----------------------|-----------|----------|----------|---------|----------|-------------------|-----------|----------|----------|---------|----------|--------|--------|
| | f | x_1 | x_2 | x_3 | x_4 | x_5 | f | x_1 | x_2 | x_3 | x_4 | x_5 | P_f | P_l |
| f_{19} | 500.0023 | -22.1995 | 11.4599 | -36.2687 | 68.9998 | -37.9225 | 500.1091 | -22.1404 | 11.2668 | -36.2592 | 68.9192 | -37.9133 | 0.1068 | 0.2178 |
| | 500.0611 | -25.0001 | 6.8588 | -39.2834 | 63.8559 | -42.1153 | 500.1292 | -25.0018 | 6.8592 | -39.2723 | 63.8337 | -42.1010 | 0.0681 | 0.0287 |
| | 500.0907 | -46.4426 | -3.0700 | -49.8269 | 56.0704 | -50.5283 | 500.0979 | -46.4534 | -3.0612 | -49.8173 | 56.1213 | -50.5448 | 0.0072 | 0.0561 |
| | 500.1281 | -31.3099 | 4.2889 | -42.9418 | 58.0552 | -46.9003 | 500.1475 | -31.3063 | 4.2901 | -42.9317 | 58.0661 | -46.9169 | 0.0194 | 0.0226 |
| | 500.1030 | -34.7069 | -10.5994 | -60.2413 | 54.6492 | -60.9458 | 500.1952 | -34.7672 | -10.5528 | -60.2362 | 54.6518 | -60.9347 | 0.0922 | 0.0772 |
| f_{20} | 600.0068 | -22.0109 | 11.5786 | -36.0034 | 69.4014 | -37.6519 | 599.9330 | -22.0115 | 11.5642 | -35.9012 | 69.4102 | -37.6413 | 0.0739 | 0.1041 |
| | 600.0586 | -100.0000 | 11.0005 | -37.8204 | 70.8742 | -25.9073 | 599.9468 | -100.0000 | 11.1015 | -37.8113 | 70.8513 | -25.8984 | 0.1118 | 0.1043 |
| | 600.0591 | -41.9112 | 12.0157 | -38.3612 | 72.3213 | -39.5142 | 600.0655 | -41.8397 | 12.0277 | -38.3525 | 72.3357 | -39.5054 | 0.0063 | 0.0749 |
| | 600.0671 | -25.7643 | 9.0700 | -27.3033 | 69.5256 | -36.3012 | 600.0348 | -25.7528 | 9.0594 | -27.3270 | 69.5188 | -36.3140 | 0.0323 | 0.0319 |
| | 600.1270 | -99.9565 | 11.5712 | -36.7345 | 62.4002 | -37.5823 | 600.2360 | -100.0000 | 11.5834 | -36.7231 | 62.4028 | -37.5947 | 0.1089 | 0.0483 |

从而使得 PMB 算法能在生产实际中提供更准确的决策变量和更多的优化方案. 噪声环境下的依概率优化算法的理论和应用研究刚刚起步, 作者后续工作将进一步研究提高 PMB 算法对高维问题的处理效率.

References

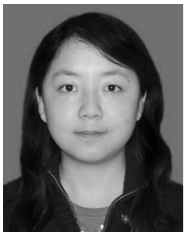
- Lu Zhi-Jun, An Jun-Xiu, Wang Peng. Partition-based MQHOA for multimodal optimization. *Acta Automatica Sinica*, 2016, **42**(2): 235-245
(陆志君, 安俊秀, 王鹏. 基于划分的多尺度量子谐振子算法多峰优化. *自动化学报*, 2016, **42**(2): 235-245)
- Rakshit P, Konar A. Realization of learning induced self-adaptive sampling in noisy optimization. *Applied Soft Computing*, 2018, **69**: 288-315
- Rakshit P, Konar A, Das S, Jain L C, Nagar A K. Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2014, **44**(7): 922-937
- Zhang Yong, Gong Dun-Wei, Hu Ying, Zhang Jian-Hua. A PSO-based multi-robot search method for odor source in indoor environment with noise. *Acta Electronica Sinica*, 2014, **42**(1): 70-76
(张勇, 巩敦卫, 胡滢, 张建化. 室内噪声环境下气味源的多机器人微粒群搜索方法. *电子学报*, 2014, **42**(1): 70-76)
- Ariizumi R, Tesch M, Kato K, Choset H, Matsuno F. Multiobjective optimization based on expensive robotic experiments under heteroscedastic noise. *IEEE Transactions on Robotics*, 2017, **33**(2): 468-483
- Hong J H, Ryu K R. Simulation-based multimodal optimization of decoy system design using an archived noise-tolerant genetic algorithm. *Engineering Applications of Artificial Intelligence*, 2017, **65**: 230-239
- Yun R, Singh V P, Dong Z C. Long-term stochastic reservoir operation using a noisy genetic algorithm. *Water Resources Management*, 2010, **24**(12): 3159-3172
- de Lope J, Maravall D. Multi-objective dynamic optimization for automatic parallel parking. In: Proceedings of the 10th International Conference on Computer Aided Systems Theory. Las Palmas de Gran Canaria, Spain: Springer, 2005. 513-518
- Hing J, Hart K, Goodman A. Towards autonomous weapons movement on an aircraft carrier: Autonomous swarm parking. In: Proceedings of the 20th International Conference on Human Interface and the Management of Information. Information in Applications and Services. Las Vegas, NV, USA: Springer, 2018. 403-418
- Chang Y, Hao Y, Li C W. Phase dependent and independent frequency identification of weak signals based on duffing oscillator via particle swarm optimization. *Circuits, Systems, and Signal Processing*, 2014, **33**(1): 223-239
- Pereira L A M, Papa J P, Coelho A L V, Lima C A M, Pereira D R, de Albuquerque V H C. Automatic identification of epileptic EEG signals through binary magnetic optimization algorithms. *Neural Computing and Applications*, 2019, **31**(2): 1317-1329
- Costa D M D, Paula T I, Silva P A P, Paiva A P. Normal boundary intersection method based on principal components and Taguchi's signal-to-noise ratio applied to the multiobjective optimization of 12L14 free machining steel turning process. *The International Journal of Advanced Manufacturing Technology*, 2016, **87**(1): 825-834
- Nakama T. Theoretical analysis of genetic algorithms in noisy environments based on a Markov model. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. Atlanta, GA, USA: ACM, 2008. 1001-1008
- Li Jun-Hua, Li Ming. An analysis on convergence and convergence rate estimate of genetic algorithms in noisy environments. *Acta Electronica Sinica*, 2011, **39**(8): 1898-1902
(李军华, 黎明. 噪声环境下遗传算法的收敛性和收敛速度估计. *电子学报*, 2011, **39**(8): 1898-1902)
- Ma H P, Fei M R, Simon D, Chen Z X. Biogeography-based optimization in noisy environments. *Transactions of the Institute of Measurement and Control*, 2015, **37**(2): 190-204
- Beyer H G, Sendhoff B. Toward a steady-state analysis of an evolution strategy on a robust optimization problem with noise-induced multimodality. *IEEE Transactions on Evolutionary Computation*, 2017, **21**(4): 629-643
- Zhan Z H, Zhang J, Li Y, Shi Y H. Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2011, **15**(6): 832-847
- Li Bao-Lei, Shi Xin-Ling, Gou Chang-Xing, Lv Dan-Ju, An Zhen-Zhou, Zhang Yu-Feng. Multivariate optimization al-

- gorithm and its convergence analysis. *Acta Automatica Sinica*, 2015, **41**(5): 949–959
(李宝磊, 施心陵, 苟常兴, 吕丹桔, 安镇宙, 张榆锋. 多元优化算法及其收敛性分析. 自动化学报, 2015, **41**(5): 949–959)
- 19 Dong Yi, Lv Dan-Ju, Wang Xia, Wang Yao-Min, Li Peng, Shi Xin-Ling. A global randomness-based probability convergence analysis of Fibonacci tree optimization algorithm. *Control and Decision*, 2018, **33**(3): 439–446
(董易, 吕丹桔, 王霞, 王耀民, 李鹏, 施心陵. 斐波那契树优化算法全局随机性概率收敛分析. 控制与决策, 2018, **33**(3): 439–446)
- 20 Li Jun-Hua, Li Ming. Genetic algorithm for multi-modal function optimization in noisy environments. *Acta Electronica Sinica*, 2012, **40**(2): 327–330
(李军华, 黎明. 噪声环境下多模态函数优化的遗传算法. 电子学报, 2012, **40**(2): 327–330)
- 21 Li Ming, Li Jun-Hua. Performance evaluation of Genetic algorithm in noisy environments. *Acta Electronica Sinica*, 2010, **38**(9): 2090–2094
(黎明, 李军华. 噪声环境下遗传算法的性能评价. 电子学报, 2010, **38**(9): 2090–2094)
- 22 Li Jun-Hua, Li Ming, Chen Hao, Wu Jia-Ju. Dynamic fitness evaluation of Genetic algorithms in normal random noisy environments. *Acta Electronica Sinica*, 2019, **47**(3): 649–656
(李军华, 黎明, 陈昊, 伍家驹. 正态随机噪声环境下遗传算法的动态适应度评价. 电子学报, 2019, **47**(3): 649–656)
- 23 Pan H, Wang L, Liu B. Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation*, 2006, **181**(2): 908–919
- 24 Han L, He X S. A novel opposition-based particle swarm optimization for noisy problems. In: Proceedings of the 3rd International Conference on Natural Computation (ICNC 2007). Haikou, China: IEEE, 2007. 624–629
- 25 Zhang J Q, Xu L W, Li J, et al. Integrating Particle Swarm Optimization with Learning Automata to solve optimization problems in noisy environment. In: Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). San Diego, CA, USA: IEEE, 2014. 1432–1437
- 26 Rada-Vilela J, Johnston M, Zhang M J. Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems. *Swarm and Evolutionary Computation*, 2014, **17**: 37–59
- 27 Taghiyeh S, Xu J. A new particle swarm optimization algorithm for noisy optimization problems. *Swarm Intelligence*, 2016, **10**(3): 161–192
- 28 Zhang S, Xu J, Lee L H, Chew E P, Wong W P, Chen C H. Optimal computing budget allocation for Particle Swarm Optimization in stochastic optimization. *IEEE Transactions on Evolutionary Computation*, 2017, **21**(2): 206–219
- 29 Zhang J Q, Zhu X X, Wang Y H, Zhou M C. Dual-environmental particle swarm optimizer in noisy and noise-free environments. *IEEE Transactions on Cybernetics*, 2019, **49**(6): 2011–2021
- 30 Xiao Hui-Hui, Wan Chang-Xuan, Duan Yan-Ming, Tan Qian-Lin. Flower pollination algorithm based on gravity search mechanism. *Acta Automatica Sinica*, 2017, **43**(4): 576–594
(肖辉辉, 万常选, 段艳明, 谭黔林. 基于引力搜索机制的花朵授粉算法. 自动化学报, 2017, **43**(4): 576–594)
- 31 Dong Yi, Shi Xin-Ling, Wang Xia, Wang Yao-Min, Lv Dan-Ju, Zhang Song-Hai, et al. On accessibility of Fibonacci tree optimization algorithm for global optima of multi-modal functions. *Acta Automatica Sinica*, 2018, **44**(9): 1679–1689
(董易, 施心陵, 王霞, 王耀民, 吕丹桔, 张松海, 等. 斐波那契树优化算法求解多峰函数全局最优解的可达性分析. 自动化学报, 2018, **44**(9): 1679–1689)
- 32 Jamil M, Zepernick H J, Yang X S. Multimodal function optimization using an improved bat algorithm in noise-free and noisy environments. *Nature-Inspired Computing and Optimization*. Springer, 2017. 29–49
- 33 Rao R V, Kalyankar V D. Multi-pass turning process parameter optimization using teaching-learning-based optimization algorithm. *Scientia Iranica*, 2013, **20**(3): 967–974
- 34 Goldberg D E, Deb K, Clark J H. Genetic algorithms, noise, and the sizing of populations. *Complex System*, 1992, **6**: 333–362
- 35 Liu Man-Dan. Research and analysis of a novel heuristic algorithm: Five-elements cycle optimization algorithm. *Acta Automatica Sinica*, 2020, **46**(5): 957–970
(刘漫丹. 一种新的启发式优化算法——五行环优化算法研究与分析. 自动化学报, 2020, **46**(5): 957–970)
- 36 Cai X J, Gao X Z, Xue Y. Improved bat algorithm with optimal forage strategy and random disturbance strategy. *International Journal of Bio-Inspired Computation*, 2016, **8**(4): 205–214
- 37 Long Wen, Wu Tie-Bin, Tang Ming-Zhu, Xu Ming, Cai Shao-Hong. Grey wolf optimizer algorithm based on lens imaging learning strategy. *Acta Automatica Sinica*, 2020, **46**(10): 2148–2164
(龙文, 伍铁斌, 唐明珠, 徐明, 蔡绍洪. 基于透镜成像学习策略的灰狼优化算法. 自动化学报, 2020, **46**(10): 2148–2164)
- 38 Xu Mao-Xin, Zhang Xiao-Shun, Yu Tao. Transfer bees optimizer and its application on reactive power optimization. *Acta Automatica Sinica*, 2017, **43**(1): 83–93
(徐茂鑫, 张孝顺, 余涛. 迁移蜂群优化算法及其在无功优化中的应用. 自动化学报, 2017, **43**(1): 83–93)
- 39 Ji Xia, Yao Shen, Zhao Peng. Relative neighborhood and pruning strategy optimized density peaks clustering algorithm. *Acta Automatica Sinica*, 2020, **46**(3): 562–575
(纪霞, 姚晟, 赵鹏. 相对邻域与剪枝策略优化的密度峰值聚类算法. 自动化学报, 2020, **46**(3): 562–575)
- 40 Yu Wei-Wei, Xie Cheng-Wang, Bi Ying-Zhou, Xia Xue-Wen, Li Xiong, Ren Ke-Yan, et al. Many-objective particle swarm optimization based on adaptive fuzzy dominance. *Acta Automatica Sinica*, 2018, **44**(12): 2278–2289
(余伟伟, 谢承旺, 闭应洲, 夏学文, 李雄, 任柯燕, 等. 一种基于自适应模糊支配的高维多目标粒子群算法. 自动化学报, 2018, **44**(12): 2278–2289)
- 41 Cui Z H, Zhang J J, Wang Y C, Cao Y, Cai X J, Zhang W S, et al. A pigeon-inspired optimization algorithm for many-objective optimization problems. *Science China Information Sciences*, 2019, **62**(7): 70212
- 42 Chen Mei-Rong, Guo Yi-Nan, Gong Dun-Wei, Yang Zhen. A novel dynamic multi-objective robust evolutionary optimization method. *Acta Automatica Sinica*, 2017, **43**(11): 2014–2032
(陈美蓉, 郭一楠, 巩敦卫, 杨振. 一类新型动态多目标鲁棒进化优化方法. 自动化学报, 2017, **43**(11): 2014–2032)
- 43 Etmianiesfahani A, Ghanbarzadeh A, Marashi Z. Fibonacci indicator algorithm: A novel tool for complex optimization problems. *Engineering Applications of Artificial Intelligence*, 2018, **74**: 1–9
- 44 Adam M, Assimakis N, Farina A. Golden section, Fibonacci sequence and the time invariant Kalman and Lainiotis filters. *Applied Mathematics and Computation*, 2015, **250**: 817–831
- 45 Wang Fang-Fang. A Study on Multi-Swarm Particle Swarm Optimization for Unimodal and Multi-Modal Function Optimization [Master thesis], Nanjing Agricultural University, China,

2014

(王芳芳. 面向单模和多模函数优化的多子群粒子群算法研究 [硕士学位论文], 南京农业大学, 中国, 2014)

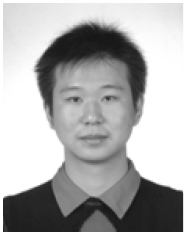
- 46 Liang J J, Qu B Y, Suganthan P N, Hernández-Díaz A G. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Technical Report, Nanyang Technological University, Singapore, 2013.



王 霞 云南大学博士研究生. 主要研究方向为智能优化算法.

E-mail: wangxiacsu@163.com

(**WANG Xia** Ph. D. candidate at Yunnan University. Her main research interest is intelligent optimization algorithm.)



王耀民 云南大学博士研究生. 主要研究方向为 SDN, 数据中心和智能优化算法.

E-mail: 18988081898@189.cn

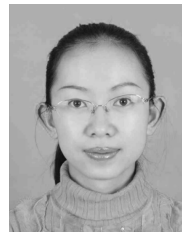
(**WANG Yao-Min** Ph. D. candidate at Yunnan University. His research interest covers SDN, data center and intelligent optimization algorithm.)



施心陵 云南大学信息学院教授. 主要研究方向为智能优化算法, 自适应信号处理与信息系统, 医学电子学. 本文通信作者.

E-mail: xlshi@ynu.edu.cn

(**SHI Xin-Ling** Professor at the School of Information Science and Engineering, Yunnan University. His research interest covers intelligent optimization algorithm, adaptive signal processing and information system and medical electronics. Corresponding author of this paper.)



高 莲 云南大学信息学院讲师. 主要研究方向为生物医学信号处理.

E-mail: ylbgl123@sina.com

(**GAO Lian** Lecturer at the School of Information Science and Engineering, Yunnan University. Her main research interest is biomedical signal processing.)



李 鹏 云南大学信息学院副教授. 主要研究方向为输变电系统安全诊断、预警与维护决策研究.

E-mail: lipeng@ynu.edu.cn

(**LI Peng** Associate professor at the School of Information Science and Engineering, Yunnan University. His research interest covers safety diagnosis, early warning and maintenance decision of power transmission and transformation system.)