

量子启发式优化算法的尺度动态调速机制

穆磊^{1,2}, 王鹏¹

(1. 西南民族大学计算机科学与工程学院, 四川成都 610041; 2. 电子科技大学计算机科学与工程学院, 四川成都 611731)

摘要: 尺度在量子启发谐振子优化算法中起着重要作用, 反映了解空间中搜索探针的分辨率. 当前研究中以固定速度调整尺度并未合理利用尺度资源. 此外, 候选解可能会因高斯采样的聚集效应而陷入边界. 本文提出了一种在一定程度上反映了适应度利用效率的指标, 称为适应度进化利用率. 在此基础上, 本文提出了一种具有尺度动态调速机制和边界映射反弹策略的量子启发式优化方法. 该算法通过与尺度调整因子相关的适应度进化利用率动态调节尺度调整速度, 通过2种不同的边界映射反弹策略增加可行解的多样性. 将本算法与多种流行优化算法在基准测试函数集上进行对比实验, 采用了一种带有动态可接受误差的成功率评估机制保证公平性, 实验结果表明该算法具有较强的竞争性.

关键词: 速度调节; 优化算法; 尺度调整; 量子行为

中图分类号: TP301.6

文献标识码: A

文章编号: 0372-2112(2022)07-1664-10

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20201305

Speed Regulation of Scale Adjustment in Quantum-Inspired Optimization Algorithm

MU Lei^{1,2}, WANG Peng¹

(1. School of Computer Science and Engineering, Southwest Minzu University, Chengdu, Sichuan 610041, China;

2. School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China)

Abstract: The scale plays an important role in the multiscale quantum-inspired harmonic oscillator algorithm, and it reflects the resolution of searching probes in solution space. The fixed-rate speed of scale adjustment in the current researches leads to unreasonable use of the resources on a specific scale. Besides, candidate solutions may be trapped in the boundary for the aggregation effect of Gaussian sampling. A metric called fitness evolution ratio, which reflects the fitness utilization efficiency to a certain extent, is proposed in this paper. On this basis, a quantum-inspired optimization algorithm is put forward with speed regulation of scale adjustment and boundary mapping rebounding strategy. The algorithm dynamically adjusts the scale adjustment speed through the fitness evolution ratio related to the scale adjustment factor. Besides, it implements two different boundary mapping rebounding strategies to increase the diversity of candidate solutions. Comparison experiments are conducted on the benchmark functions with a variety of compared algorithms. For a fair comparison, an evaluation mechanism of success rate with dynamic acceptable errors is utilized. The results show considerable competitiveness for our scheme.

Key words: speed regulation; optimization algorithm; scale adjustment; quantum behavior

1 引言

优化问题是一种选择特定方案以在特定条件下实现最佳目标的方法, 被广泛用于军事、工程、管理等领域, 是人工智能和其他技术领域的基石^[1].

多尺度量子启发式谐振子优化算法(Multiscale Quantum-inspired Harmonic Oscillator Algorithm, MQHOA)

是一种受量子物理过程启发的优化算法^[2], 源于量子退火(Quantum Annealing, QA)派生的量子优化理论^[3], 其理论体系和应用领域都得到了广泛的关注, 在数据聚类^[4]、任务分配^[5]、功率自适应控制^[6]等方面都有应用.

王鹏等人^[2]定义了MQHOA算法波函数, 提出了算

法基本物理模型. 随后, 零点能量和量子隧穿效应的分析极大地发展了该算法框架^[7]. 文献^[8]证明了波函数定理和尺度减半策略的有效性, 通过量子算符从理论上解释了多尺度的必要性. 随后, 一些工作补充了算法理论体系^[9]. 应用方面, 基于 MQHOA 的具有多级分辨率的多模优化分区算法显示出良好的性能^[10]. MQHOA 还被用于选择聚类中心, 通过波函数的概率变化将高能级降低到低能级来找到最佳的聚类和聚类中心^[4]. 韩虎等人^[11]将云计算任务调度方案与 MQHOA 的采样位置建立关联, 在 CloudSim 平台上实施并获得了优于比较算法的调度方案.

尺度因素是大多数智能优化算法的重要组成部分, 用于调整算法在解空间的搜索粒度. 现有大量 MQHOA 的研究工作使用尺度减半策略, 即尺度除以 2 作为下一个尺度值. 该方法可以实现不同粒度的搜索, 但缺乏灵活性, 未考虑实际进化过程中不同的算法状态具有不同的搜索粒度要求, 无法实现自适应调整. 另外, 由于高斯分布的聚集效应, 高斯采样生成的候选解会聚集在采样中心区域附近. 如果当前采样中心位于定义域的边界, 可能导致采样点因聚集效应以大概率集中在边界附近, 使得算法陷入早熟收敛. 这也是本文要解决的问题.

本文的主要贡献如下: (1) 将适应度进化利用率作为尺度动态调速的依据, 通过尺度调节因子动态改变尺度调节速度, 以达到探索与开发的平衡; (2) 根据映射点、采样中心点和生成点之间的位置关系, 定义了 2 种不同的边界映射策略, 测试了不同尺度调节因子参数组合对性能的影响; (3) 采用一种动态可接受误差的评估机制, 通过数据自身特征来计算可接受误差, 以此计算成功率, 评估算法稳定性; (4) 将改进算法与 MQHOA 及对比算法在基准测试函数上进行了比较, 实验结果表明改进算法有效地提高了 MQHOA 的性能.

2 多尺度量子启发式谐振子优化算法

MQHOA 的理论基础的核心思想是优化空间中的搜索过程与量子空间中的粒子运动具有相似性^[12]. 根据 QA 派生的量子优化理论^[3], 当量子系统收敛到基态时可以得到目标函数的极值^[13,14]. 因此, 优化问题可以理解为在有约束势阱的量子系统中寻找基态.

2.1 量子运动与优化问题的映射

量子系统中, 波函数随时间和空间变化的方程被称为薛定谔方程. 其中, 定态薛定谔方程描述了粒子在势阱中受不含时 Hamiltonian 约束的运动, 它可以看作从量子力学的角度来描述优化问题的工具, 形式如下^[15]:

$$E\psi(x) = \left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \right) \psi(x) \quad (1)$$

其中, \hbar 表示简化的普朗克常数, m 表示粒子质量, 状态 $\psi(x)$ 的能量特征值表示为 E . 波函数 $\psi(x)$ 的二次方对应于粒子的概率分布, 可以被认为与最优解的概率分布相关^[12-14].

理论上, 可以通过求解薛定谔方程得到基态波函数, 然后得到最优解的概率分布. 与量子计算机相比, 经典计算机实际处理的优化问题通常涉及非常大的 Hilbert 空间, 薛定谔方程的推演在这种条件下是不可行的. MQHOA 将复杂目标函数 $f(x)$ 的全局优化问题转换为求解不同尺度下谐振子势的基态波函数问题, 使整个理论系统具有可操作性, 原理如图 1 所示.

从图 1 可以看出, MQHOA 采用势阱等效和谐振子势能近似, 使薛定谔方程可以近似求解. 定态薛定谔方程变为

$$E\psi(x) = \left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2} kx^2 \right) \psi(x) \quad (2)$$

其中, k 是弹性系数.

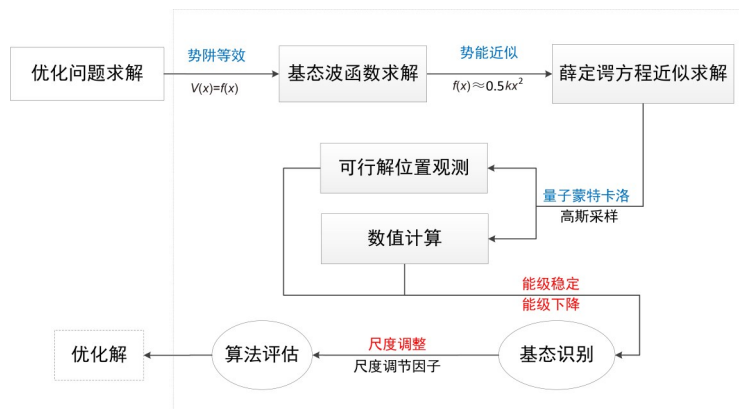


图 1 MQHOA 基本原理框图

2.2 MQHOA 算法流程

MQHOA 的具体实现框架由 3 部分组成,即能级稳定、能级下降和尺度调整,具体流程如图 2 所示.

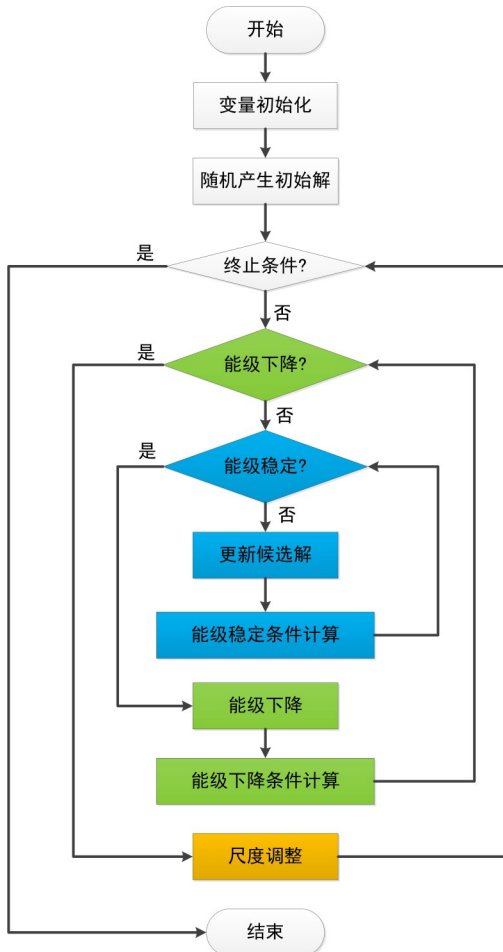


图 2 MQHOA 算法流程图

从图 2 可以看出,能级稳定是算法的基本进程,达到能级稳定条件后将转入能级下降阶段,依此类推,直到在该尺度下达到基态.尺度调整后,在新搜索粒度下重复该循环过程.通过这样的机制,全局优化问题转换为多尺度的基态收敛问题.

关于 MQHOA 的详细理论推导和实现细节可以参考文献[14].

3 尺度调整速度调节的改进方法

本节定义了尺度调节因子以及进化效率的评估指标,改进了边界映射策略以提高算法性能,随后,详细阐述了 MQHOA 中尺度调整速度调节的改进算法.

3.1 术语

定义 1 f_s 定义为尺度调节因子,用作尺度调节的控制量.

$$\sigma'_s = \frac{\sigma_s}{f_s} \quad (3)$$

其中, σ'_s 和 σ_s 分别表示调整前后的尺度.显然,调整式(3)中的尺度调节因子 f_s 可以控制尺度变化的趋势和速度.

定义 2 适应度进化利用率 r_{fe} 定义为下式中适应度变化与进化次数变化的比率:

$$r_{fe} = \frac{f'_{\min} - f_{\min}}{n'_{evo} - n_{evo}} \quad (4)$$

其中, f'_{\min} 和 f_{\min} 是更改前后的适应度值, n'_{evo} 和 n_{evo} 是它们分别对应的进化次数.在某种程度上,该指标描述了在适应度改变时平均每次进化的收益,它可以用来评估进化效能,为尺度调整提供依据,以获得最大的利用效率.

3.2 边界反弹映射策略

进化算法中一般使用边界映射策略处理候选解超出定义域的情况. MQHOA 中采用直接截断策略.本方法将越界点根据其临近边界点的位置关系,反弹回定义域特定区域.根据映射点、采样中心点和越界点的位置,可以定义 2 种不同策略.

在定义之前给出以下符号定义:越界采样点位置为 x_b , 采样中心位置为 x_c , 边界反弹映射策略生成的点位置是 x_g . 定义域上界为 u_b , 下界为 l_b . 2 种边界反弹映射策略定义如下.

定义 3 生成点和越界采样点在采样中心的同一侧,称之为正向反弹策略.在这种情况下, $(x_b - x_c) \times (x_g - x_c) \geq 0$. 设 α 是均匀分布在 $(0, 1)$ 上的随机数,则可以根据以下公式生成新点:

$$x = \begin{cases} x_c + \alpha \times (u_b - x_c), & x_b \geq u_b \\ l_b + \alpha \times (x_c - l_b), & x_b < l_b \end{cases} \quad (5)$$

定义 4 生成点和越界采样点在采样中心的另一侧,称之为负向反弹策略.在这种情况下, $(x_b - x_c) \times (x_g - x_c) < 0$, 可以根据以下公式生成新点:

$$x_g = \begin{cases} l_b + \alpha \times (x_c - l_b), & x_b \geq u_b \\ x_c + \alpha \times (u_b - x_c), & x_b < l_b \end{cases} \quad (6)$$

图 3 是 2 种反弹映射策略的示例.图 3 中,浅色点表示采样中心点,蓝色点是越界采样点,超出了定义域的范围.正向反弹策略将点映射到采样中心的同侧,如图 3(a)所示的绿色点.负向反弹策略将点映射到采样中心的相反一侧,表示为绿色点,如图 3(b)所示.

3.3 尺度调整的速度调节

进化算法的探索 and 开发存在均衡问题.如果使用

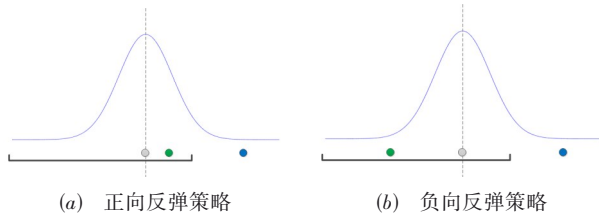


图3 边界映射策略

固定尺度调节因子 f_s 调整尺度,则可能会丢失某些尺度上的测量值,也有可能某个尺度浪费过多资源. MQHOA 中尺度减半策略对应的尺度调节因子是 2,该策略以此作为速度调整变化的基准.

本方法通过尺度调节因子来改变尺度调整的速度. 当适应度值变小时,获得适应度值并记录对应的进化数,通过式(4)来计算适应度进化利用率. 一旦适应度进化利用率发生变化,尺度调整速度将动态调节.

如果新适应度进化利用率 r'_{fe} 小于原有适应度进化利用率 r_{fe} ,则表明进化平均收益在下降. 此时,尺度应迅速下降,以加强新尺度下的探索,尺度调整采用加速因子 f_a ,其值应大于 2. 如果 r'_{fe} 大于 r_{fe} ,则表明进化平均收益在增加. 此时,尺度应缓慢下降,以促进该尺度附近的开发,尺度调整采用减速因子 f_d ,其值应小于 2.

定义 \mathbf{a} 为尺度调整因子的控制向量,尺度调整基向量为 \mathbf{f} ,则尺度调整因子 f_s 可以表示为 \mathbf{a} 和 \mathbf{f} 的内积:

$$f_s = \mathbf{a} \cdot \mathbf{f} = \mathbf{a}^T \mathbf{f} \quad (7)$$

其中

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f_a \\ 2 \\ f_d \end{bmatrix} \quad (8)$$

定义符号函数 $\text{sgn}(\cdot)$ 为

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (9)$$

则本算法尺度调整因子的控制向量为

$$\mathbf{a} = \begin{bmatrix} 1 - \text{sgn}(\text{sgn}(r'_{fe} - r_{fe}) + 1) \\ 1 - \text{abs}(\text{sgn}(r'_{fe} - r_{fe})) \\ 1 + \text{sgn}(\text{sgn}(r'_{fe} - r_{fe}) - 1) \end{bmatrix} \quad (10)$$

显然

$$\begin{cases} a_i \in \{0, 1\}, & i \in \{1, 2, 3\} \\ \sum_{i=1}^n a_i = 1, & n = 3 \end{cases} \quad (11)$$

3.4 算法实现框架

本文提出的方法在 MQHOA 的框架基础上进行改进,其伪代码如算法 1 所示.

算法 1 尺度动态调速的量子优化方法

输入: $k, [d_{\min}, d_{\max}]$

输出: $x^{\text{best}}, f(x^{\text{best}})$

变量初始化;

在区间 $[d_{\min}, d_{\max}]$ 随机产生 $x^i, (i=1, \dots, k)$;

WHILE 未达到终止条件 DO

 WHILE $\sigma_t > \sigma_s(t)$ DO

 WHILE $\Delta\sigma_t > \sigma_s(t)$ DO

 FOR 给定任意 $x(i, j, t)$ DO

 高斯分布产生候选解 $x'(i, j, t) \sim N(x(i, j, t), \sigma_s^2)$

 执行边界映射策略

 IF $f(x(i, j, t+1)) < F_{\min}$ THEN

$F_{\min} \leftarrow f(x(i, j, t+1))$

$x_{\min} \leftarrow x(i, j, t+1)$

 END IF

 计算所有 $x(i, j, t+1)$ 的标准差 σ_{t+1}

$\Delta\sigma_{t+1} \leftarrow |\sigma_{t+1} - \sigma_t|$

 END FOR

 END WHILE

$x^{\max}(t) \leftarrow \arg \max f(x(t))$

$x^{\text{mean}}(t) \leftarrow x^{\text{mean}}(t)$

$f(x^{\max}(t)) \leftarrow f(x^{\text{mean}}(t))$

 END WHILE

 尺度调整的速度调节: $f_s = \mathbf{a} \cdot \mathbf{f}$

$\sigma_s(t+1) = \sigma_s(t) f_s$

END WHILE

4 数值模拟讨论

PSR 和 NSR 分别表示具有正向和负向反弹策略的尺度动态调速方法,与以下优化算法进行综合比较: MQHOA, 精简烟花算法 (Bare Bone FireWorks Algorithm, BBFWA)^[16], 量子行为粒子群优化算法 (Quantum Particle Swarm Optimization, QPSO)^[12] 和标准粒子群优化算法 2011 (Standard Particle Swarm Optimization 2011, SPSO2011)^[17].

4.1 基准测试函数

基准测试函数用来验证不同优化算法的总体性能,它们都被表述为最小化问题,并根据其重要物理特性和形状相似性进行分组,如表 1 所示.

这些基准测试函数分为 2 类. 函数 $f_1 \sim f_6$ 是多模函数,它们都具有许多局部最小值,可以测试优化算法跳出局部最优值的能力. 函数 $f_7 \sim f_{14}$ 是单模函数,它们只有一个全局最小值,具有不同的形状,例如碗形、谷形等,主要测试算法对不同问题的快速收敛能力.

4.2 实验参数设置

本文针对每个基准测试函数的 30 维问题进行了实验,最大进化次数设为 10 000×维度,每个实验重复运

表 1 基准测试函数

函数名称	函数表达式	维度	搜索空间	全局最优位置
Griewank	$f_1 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	n	$[-100, 100]$	$f(0, \dots, 0)=0$
Rastrigin	$f_2 = 10n + \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)]$	n	$[-5.12, 5.12]$	$f(0, \dots, 0)=0$
Ackley	$f_3 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	n	$[-32.77, 32.77]$	$f(0, \dots, 0)=0$
Levy	$f_4 = \sin^2(\pi \omega_1) + \sum_{i=1}^{n-1} (\omega_i - 1)^2 [1 + 10 \sin^2(\pi \omega_i + 1)] + (\omega_n - 1)^2 [1 + \sin^2(2\pi \omega_n)]$, where $\omega_i = 1 + \frac{x_i - 1}{4}$, for all $i=1, \dots, n$	n	$[-10, 10]$	$f(1, \dots, 1)=0$
Alpine	$f_5 = \sum_{i=0}^{n-1} x_i \sin(x_i) + 0.1x_i $	n	$[0, 10]$	$f(0, \dots, 0)=0$
Schwefel	$f_6 = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	n	$[-500, 500]$	$f(420.97\dots)=0$
Sphere	$f_7 = \sum_{i=1}^n x_i^2$	n	$[-5.12, 5.12]$	$f(0, \dots, 0)=0$
Sum Squares	$f_8 = \sum_{i=0}^{n-1} ix^2$	n	$[-10, 10]$	$f(0, \dots, 0)=0$
Rotated Hyper-Ellipsoid	$f_9 = \sum_{i=1}^n \sum_{j=1}^i x_j^2$	n	$[-65.54, 65.54]$	$f(0, \dots, 0)=0$
Ellipsoidal	$f_{10} = \sum_{i=1}^n (x_i - i)^2$	n	$[-100, 100]$	$f(1, 2, \dots, n)=0$
Sum of Different Power	$f_{11} = \sum_{i=1}^n x_i ^{i+1}$	n	$[-100, 100]$	$f(0, \dots, 0)=0$
Zakharov	$f_{12} = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	n	$[5, 10]$	$f(0, \dots, 0)=0$
Rosenbrock	$f_{13} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	n	$[5, 10]$	$f(1, \dots, 1)=0$
Dixon-Price	$f_{14} = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	n	$[-10, 10]$	$f\left(2^{\frac{2^i-2}{2^i}}\right)=0$

行 51 次。所有比较算法的种群规模均设置为 40 以进行公平比较。每种算法中的参数设置均基于相关文献。BBFWA 中的 C_a 和 C_r 分别设置为 1.2 和 0.9, 在 QPSO 中考虑了线性递减的收缩膨胀, 其中 a_1 和 a_2 分别设置为 1 和 0.5。SPSO2011 的参数选自文献[17]。在 MQHOA 及其改进算法中, k 表示探针或个体的数量, 设置为 40。所有数值模拟都在 8 GB 内存的 Intel i7 6700 CPU (3.4 GHz) 和 Windows 10 的计算机上通过 64 位版本的 MATLAB R2018a 实现。

4.3 尺度调节因子参数分析

在比较算法之前, 应先确定合适的尺度调节因子 f_s 。本文使用不同取值下尺度调节因子来求解 30 维问题基准测试函数的实验, 统计分析参数组合对两种边界

映射策略的影响。

为了便于计算, 限制了参数组合的数量, 采用网格搜索确认参数组合。其中, f_a 从 2.1 变化至 2.9, 步长为 0.2; f_d 从 1.1 变化至 1.9, 步长为 0.2。在这种情况下, 2 种边界映射策略改进的 MQHOA 算法总计有 5×5 种参数组合。针对每个基准测试函数进行 51 次运行, 计算每种参数组合对应的误差。所有实验运行完成后, 统计 2 种改进算法的不同参数组合中, 每种参数组合下误差不高于原始 MQHOA 的函数数量。随后, 固定尺度调节因子组合中的一个参数值, 然后将另一参数所有对应结果求平均值。由于函数数量为整数, 向下取整直接保留整数部分。

2 种边界映射策略的结果以不同的颜色区分, 如图 4 所示, 其中虚线是趋势线。

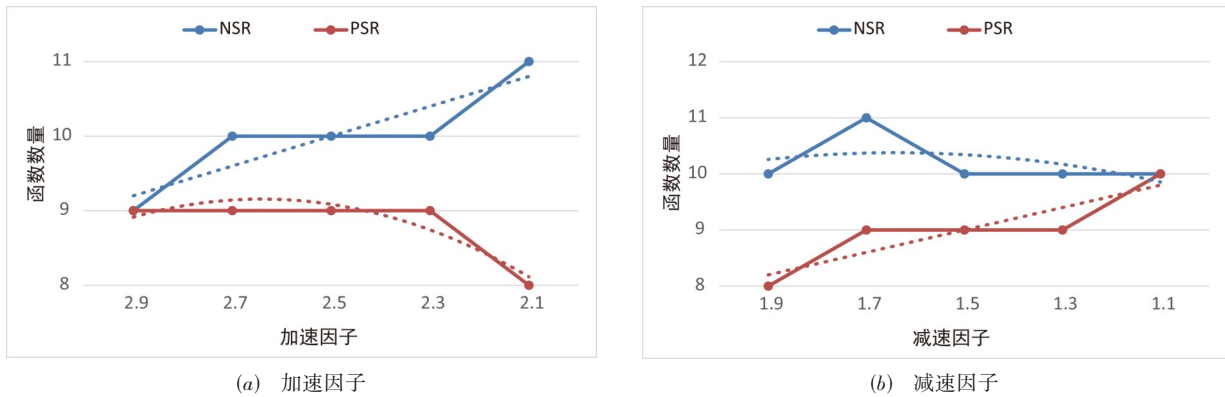


图4 尺度调节因子统计结果

如图4所示,根据尺度调节因子的定义以及基准参考值的选取不难看出,当尺度调节因子与2越接近的时候,加速或减速越慢;当尺度调节因子与2差距越大时,加速或减速越快.

对于负向反弹策略,当 f_a 较大时,仅在较少数量的测试函数上实现性能提升. f_d 较小时也会发生相同的情况.数据表明,大幅度的加速和减速不利于负向反弹策略的性能提升.负向反弹策略位置调整相对较大,快速调整尺度会导致特定尺度下的搜索不足.

正向反弹策略显示出相反的趋势.当 f_a 较大时,在大量测试函数上实现性能提升. f_a 的值与实现性能改进的函数数量负相关.结果表明,在正向反弹策略中,尺度调整应相对较快地加速和减速.该映射策略将点映射到越界点附近的定义域内,如果速度调节相对较慢,则会在特定尺度上消耗资源,从而影响多样性和进化利用率,使性能受到影响.

4.4 与其他方案的比较和讨论

将2种改进算法与一些常见优化方案在基准测试函数的30维问题中进行比较,评估不同优化算法的误差和成功率.在NSR中 f_a 和 f_d 设置为2.1和1.7,在PSR中 f_a 和 f_d 设置为2.3和1.1.这些参数值参考第4.3节中的实验结果进行设置.

4.4.1 误差分析

不同优化算法的误差结果显示在表2中.每个单元格中的数据均由3部分组成:平均值、标准差(第一个括号中)以及该误差在所有算法中的排名顺序(最后一个括号中的数字).

为了更清楚地观察整体性能,将每个算法误差均值的排名按照单模函数、多模函数和所有函数分类进行平均.

图5显示了所有平均误差的平均排名(Average Ranking, AR).根据中央极限定理, $AR \sim N((k+1)/2, (k^2-1)/12n)$,其中 k 是对比算法数量, n 是测试函数数量^[18].

单模函数,多模函数以及所有函数上AR的先验标准差(Prior Standard Deviation, PSD)为0.6,0.69和0.46.

如图5所示,BBFWA在单模函数上排名第一,其次是NSR和SPSO2011.NSR在多模函数中排名第一,紧随其后的是QPSO和SPSO2011.在所有函数上,NSR排名第一,BBFWA排名第二,之后是SPSO2011.虽然BBFWA在 $f_7, f_8, f_9, f_{12}, f_{14}$ 等函数中排名优势显著,但在某些函数上排名太低.NSR由于在不同函数上的均衡排名而胜出.

图6显示了部分基准函数上不同算法平均误差的统计信息.其中MQHOA和BBFWA分别用M和BBF表示.

根据箱形图的分析结果,NSR在除 f_2 之外的所有基准测试函数上均具有良好的稳定性.NSR在 $f_1, f_3, f_4, f_7, f_9, f_{10}, f_{12}, f_{13}$ 上表现出稳定性和良好性能之间的平衡.在 f_5 和 f_{11} 上,NSR在性能和稳定性方面的表现并不是最好的,但也排在前列.

4.4.2 成功率分析

成功率 r_s 可用于衡量算法的可靠性,它被定义为达到可接受误差范围内的运行次数占运行总次数的比例,可以按下式计算:

$$r_s = \frac{n_{\text{accept}}}{n_{\text{total}}} \quad (12)$$

其中, n_{accept} 和 n_{total} 分别表示达到可接受误差范围内的运行次数和运行总次数.对于每个函数,计算所有算法误差均值的平均数,将其作为可接受误差 e_{accept} :

$$e_{\text{accept}} = \frac{\sum_{i=1}^{n_{\text{alg}}} e_{\text{mean}}(i)}{n_{\text{alg}}} \quad (13)$$

其中, n_{alg} 表示对比算法总数, $e_{\text{mean}}(i)$ 表示第 i 个算法的误差均值.

成功率结果显示在表3中,括号中的数字表示其在测试函数上的排名.

表 2 基准测试函数上的误差结果

No.	SPSO2011	MQHOA	NSR	PSR	BBFWA	QPSO
1	7.48E-03(7.73E-03) (3)	1.02E-02(3.98E-02) (4)	2.22E-03(4.65E-03) (1)	4.25E-03(6.72E-03) (2)	1.08E-02(9.11E-03) (5)	4.31E-01(2.28E-01) (6)
2	2.35E+01(6.62E+00) (1)	1.32E+02(6.91E+01) (5)	1.16E+02(7.83E+01) (3)	1.21E+02(7.28E+01) (4)	1.14E+02(3.15E+01) (2)	1.63E+02(2.00E+01) (6)
3	8.89E-01(7.77E-01) (5)	1.96E-01(4.70E-01) (4)	5.05E-02(2.63E-01) (2)	1.37E-01(4.77E-01) (3)	1.10E+00(1.05E+00) (6)	4.33E-06(6.08E-06) (1)
4	6.89E-01(6.04E-01) (5)	7.11E-02(2.67E-01) (4)	3.69E-02(6.25E-02) (2)	4.08E-02(1.11E-01) (3)	1.84E+01(1.05E+01) (6)	7.57E-10(1.94E-09) (1)
5	6.78E+00(5.82E+00) (5)	6.68E+00(1.07E+00) (4)	3.37E+00(6.45E+00) (2)	1.31E+01(2.95E+00) (6)	4.52E+00(2.18E+00) (3)	5.89E-16(1.77E-15) (1)
6	4.62E+03(5.34E+02) (1)	7.84E+03(2.96E+02) (5)	7.95E+03(2.55E+02) (6)	7.13E+03(2.60E+02) (4)	5.26E+03(6.36E+02) (2)	6.05E+03(7.66E+02) (3)
7	9.33E-101(5.77E-102)(2)	4.83E-84(7.24E-84) (5)	6.59E-86(8.80E-86) (4)	4.03E-88(7.11E-88) (3)	2.53E-107(1.55E-106)(1)	1.64E-12(3.36E-12) (6)
8	7.68E-30(1.83E-29) (2)	1.37E-05(4.35E-05) (6)	5.88E-06(8.74E-06) (4)	8.03E-06(1.62E-05) (5)	2.47E-55(1.40E-54) (1)	1.41E-11(1.94E-11) (3)
9	9.24E-101(5.34E-102)(2)	3.44E-79(4.62E-79) (5)	6.35E-81(1.69E-80) (4)	2.72E-83(3.04E-83) (3)	1.27E-103(5.92E-103)(1)	1.06E-07(1.92E-07) (6)
10	7.08E-29(1.21E-29) (5)	0.00E+00(0.00E+00) (1)	0.00E+00(0.00E+00) (1)	0.00E+00(0.00E+00) (1)	1.22E-29(1.53E-29) (4)	9.33E-10(3.78E-09) (6)
11	1.73E-09(1.11E-09) (5)	1.68E-09(1.42E-09) (4)	7.20E-10(7.23E-10) (2)	1.66E-09(1.49E-09) (3)	1.84E-09(4.63E-10) (6)	4.83E-23(2.69E-22) (1)
12	1.14E-37(2.78E-37) (2)	4.38E-14(2.38E-14) (4)	3.99E-14(3.18E-14) (3)	6.62E-14(5.18E-14) (5)	9.29E-83(5.03E-82) (1)	6.42E+00(2.47E+00) (6)
13	2.17E+01(2.56E+01) (5)	4.85E-01(3.21E-01) (3)	4.78E-01(2.59E-01) (2)	4.31E-01(3.28E-01) (1)	1.27E+01(1.14E+01) (4)	3.54E+01(2.26E+01) (6)
14	6.67E-01(3.46E-08) (2)	6.79E-01(3.89E-02) (6)	6.69E-01(7.55E-03) (4)	6.71E-01(1.98E-02) (5)	6.67E-01(5.48E-10) (1)	6.67E-01(6.07E-05) (3)

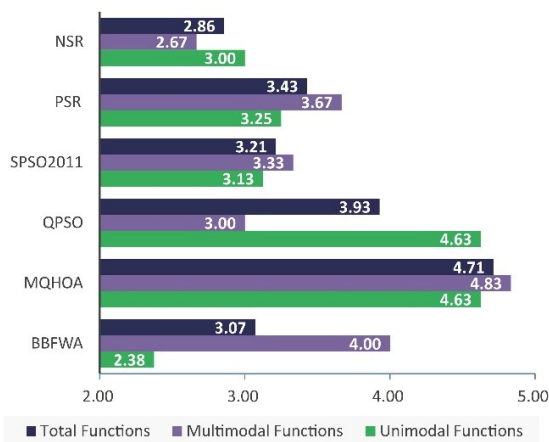


图 5 基准测试函数上的误差均值平均排名

如表 3 所示, SPSO2011 在所有算法中的 10 个函数上表现最佳. NSR 再次显示出其均衡性, 它在 7 个函数上表现最佳, 在 3 个函数上均获得第二名, 在 11 个函数

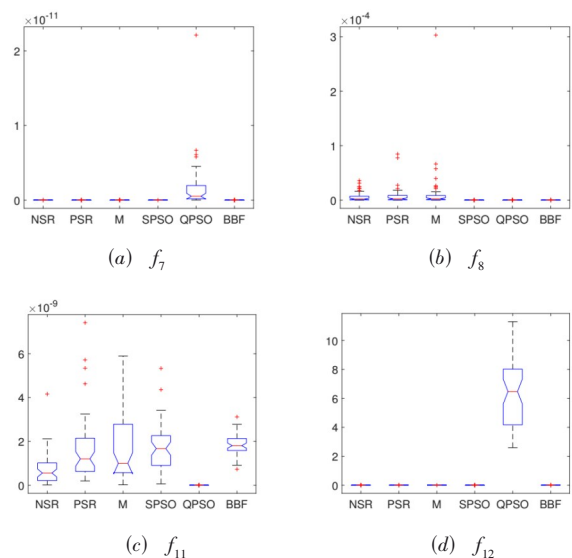


图 6 部分基准测试函数误差均值箱形图

表3 基准测试函数成功率

No.	SPSO2011	MQHOA	NSR	PSR	BBFWA	QPSO	e_{accept}
1	1.00(1)	0.98(5)	1.00(1)	1.00(1)	1.00(1)	0.10(6)	7.77E-02
2	1.00(1)	0.22(5)	0.31(3)	0.27(4)	0.47(2)	0.00(6)	1.11E+02
3	0.39(6)	0.84(4)	0.96(2)	0.92(3)	0.43(5)	1.00(1)	3.95E-01
4	1.00(1)	1.00(1)	1.00(1)	1.00(1)	0.00(6)	1.00(1)	3.21E+00
5	0.53(4)	0.20(5)	0.92(2)	0.00(6)	0.67(3)	1.00(1)	5.75E+00
6	1.00(1)	0.00(5)	0.00(5)	0.02(4)	0.96(2)	0.69(3)	6.47E+03
7	1.00(1)	1.00(1)	1.00(1)	1.00(1)	1.00(1)	0.35(6)	2.73E-13
8	1.00(1)	0.61(5)	0.67(4)	0.61(5)	1.00(1)	1.00(1)	4.61E-06
9	1.00(1)	1.00(1)	1.00(1)	1.00(1)	1.00(1)	0.35(6)	1.76E-08
10	1.00(1)	1.00(1)	1.00(1)	1.00(1)	1.00(1)	0.59(6)	1.55E-10
11	0.37(5)	0.57(3)	0.82(2)	0.53(4)	0.10(6)	1.00(1)	1.27E-09
12	1.00(1)	1.00(1)	1.00(1)	1.00(1)	1.00(1)	0.00(6)	1.07E+00
13	0.41(5)	1.00(1)	1.00(1)	1.00(1)	0.73(4)	0.00(6)	1.19E+01

上获得高于0.8的成功率,这是所有算法中的最佳结果. SPSO2011在10个函数上的成功率都是1,但在其他函数上的成功率小于0.6. BBFWA在8个函数上的成功率均大于0.8. 总体而言,NSR在更多函数上获得了更高的成功率.

各个算法按照不同的基准测试函数进行成功率排名,并分别按照单模函数、多模函数以及所有函数的类别分别计算成功率的平均排名,结果显示在图7中.

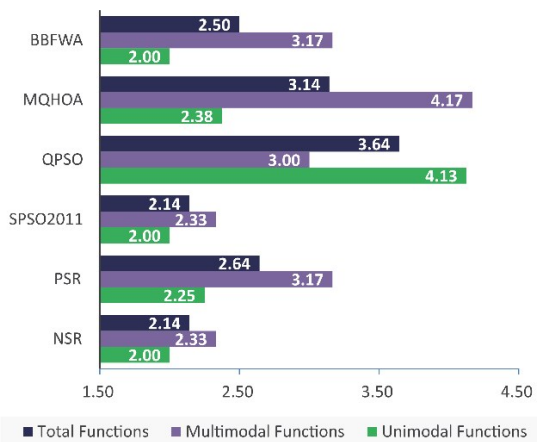


图7 基准测试函数上成功率平均排名

该图显示了NSR和SPSO2011在单模函数、多模函数和所有函数上成功率的平均排名均优于其他方案. NSR在不同的测试函数上表现出更好的均衡性. BBFWA由于在多模函数上性能不佳而在所有函数上获得第三名.

4.4.3 计算复杂度

算法1中生成候选解的时间复杂度为 $O(n_{probe}n_{dim})$,其中, n_{probe} 和 n_{dim} 分别表示探针数量和问题维度. 当探

针数量固定时,此时间复杂度随问题的维度线性增加,即时间复杂度为 $O(n_{dim})$. 此外,尺度调整速度调节的时间复杂度不会随问题的维度而变化,即时间复杂度为 $O(1)$. 通过分析整个算法,可以看到其他部分的时间复杂度最多为 $O(n_{dim})$. 从以上分析可以得出,在不考虑目标函数影响的情况下,本文提出的算法总体上具有线性时间复杂度.

将每个算法在相同的基准测试函数上执行51次,然后计算该基准测试函数上算法的平均执行时间,结果显示在表4中,括号中的数字表示其在基准测试函数上的排名.

表4 基准测试函数平均执行时间(单位:s)

No.	SPSO2011	MQHOA	NSR	PSR	BBFWA	QPSO
1	4.64(4)	17.06(6)	1.59(2)	16.02(5)	0.73(1)	4.51(3)
2	5.87(6)	2.87(4)	1.32(2)	1.34(3)	0.73(1)	4.42(5)
3	5.85(6)	3.03(3)	1.45(2)	4.47(5)	0.76(1)	4.32(4)
4	9.53(4)	15.50(6)	3.79(2)	13.54(5)	3.19(1)	8.50(3)
5	5.57(6)	2.83(4)	1.48(2)	2.44(3)	1.02(1)	4.22(5)
6	5.72(6)	2.85(4)	2.51(2)	2.81(3)	0.86(1)	4.57(5)
7	2.62(3)	30.32(5)	2.31(2)	35.73(6)	0.89(1)	4.09(4)
8	5.10(6)	2.63(4)	1.36(3)	1.35(2)	0.87(1)	4.14(5)
9	3.09(3)	4.09(5)	2.58(2)	3.47(4)	1.30(1)	4.90(6)
10	5.70(6)	1.72(4)	1.07(3)	1.04(2)	0.70(1)	4.22(5)
11	6.50(6)	3.79(4)	2.49(3)	2.47(2)	2.09(1)	5.85(5)
12	5.07(6)	2.88(4)	1.55(2)	1.60(3)	0.88(1)	4.34(5)
13	4.75(3)	7.21(4)	81.79(6)	78.94(5)	0.86(1)	4.23(2)
14	4.70(6)	2.62(4)	1.35(2)	1.44(3)	0.88(1)	4.26(5)

从表4中可以看出,BBFWA的平均执行时间在各种基准测试函数中显示出绝对的优势. NSR在大多数基准测试函数中排名第二,在3个基准测试函数中排名

第三. 因此, NSR 的排名仍然不错. PSR 在平均执行时间上也表现良好, 它在 14 个函数的 8 个函数中排名前三.

各个算法的平均执行时间排名按照单模函数、多模函数和所有函数上的分类计算平均值, 结果如图 8 所示.

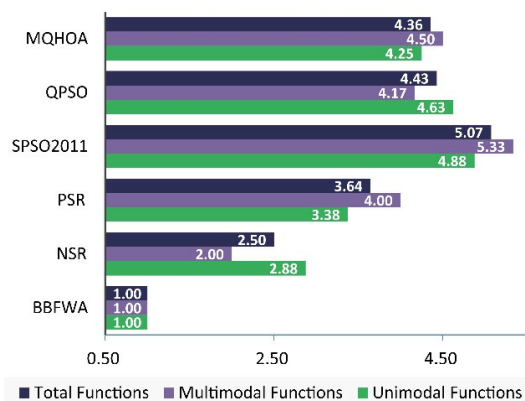


图 8 基准测试函数执行时间均值平均排名

从图 8 可以看出, BBFWA 在所有函数中均排名第一, 其次是 NSR 和 PSR. MQHOA 和 QPSO 的平均执行时间平均排名接近, 而 SPSO2011 排名最后. 对于每种算法, 不同类型函数之间的排名得分差异很小. 在不同类型函数中, 所有算法的排名顺序都没有改变.

4.4.4 讨论

这些对比算法具有不同的特性. SPSO2011 对原始粒子群优化有两大改进, 即自适应随机拓扑和旋转不变性. 这些措施带来了良好的全局搜索能力, 并阻止了早熟现象的产生. BBFWA 的结构非常简单. 它使用动态爆炸半径进行搜索, 并采用贪婪策略来更新候选解. QPSO 通过压缩-扩展系数来调整算法的收敛过程, 从而可以在全局搜索和局部搜索之间取得平衡. 其候选解的采样函数是对应于 δ 势阱的双指数分布. 该采样分布的累积效应强于高斯分布, 因此局部搜索能力强于 MQHOA. MQHOA 利用高斯分布的波函数生成候选解, 并通过调整尺度以不同的分辨率搜索解空间. 本文所提出的算法主要在边界映射策略和尺度调整速度调节方面改进了 MQHOA.

边界映射策略可以使算法具有更大的概率在定义域内有效搜索. 负向反弹策略将越界点反弹到采样中心的另一侧并远离采样中心, 具有更强的量子隧穿作用, 并探索了更广阔的区域, 获得了更好的多样性. 通过分析 r_{fe} 的变化, 动态选择 f_a 和 f_d 以调节尺度调整的速度. 负向反弹策略将点出界映射到远离采样点的区域, 在该区域采样的可能性很小. 应该缓慢减小尺度, 以微调搜索分辨率进行开发. 较小的 f_a 和较大的 f_d 可以获得更好的结果. 正向反弹策略将外部越界点映射到采样点附近, 该区域被采样的可能性更高. 如果速度调节相对较慢, 则可能会消耗过多的搜索资源, 从而影响 r_{fe} 和

候选解的多样性, 导致性能下降. 因此, 应迅速进行尺度调整, 这可以通过较大的 f_a 和较小的 f_d 来实现. 第 4.3 节中的实验结果验证了上述分析.

为了更直观地观察各算法排名对比情况, 把不同算法在各个评估指标上的整体排名进行综合统计分析, 如表 5 所示.

表 5 不同算法的平均综合排名

	精度排名	成功率排名	执行时间排名	平均综合排名
NSR	1	1	2	1.33
PSR	4	4	3	3.67
SPSO2011	3	1	6	3.33
QPSO	5	6	5	5.33
MQHOA	6	5	4	5.00
BBFWA	2	3	1	2.00

可以看出, 在所有算法中, NSR 在平均误差方面排名最高, NSR 和 SPSO2011 在成功率方面排名最高. BBFWA 在平均执行时间方面有很大的优势, 其次是 NSR. 在多个维度的综合评价中, NSR 获得了最高的平均综合排名. 从性能、效率和稳定性的总体平衡来看, 排名结果显示了 NSR 的优势. 总体而言, NSR 与对比算法相比具有较强的竞争力.

5 结论

本研究的重点是改善尺度资源的利用效率和解的多样性, 以提高算法性能. 具体而言, 采用适应度进化利用率表征特定尺度下的资源利用效率, 用作动态调节尺度调整速度的标准. 提出了 2 种不同的边界映射策略来提高多样性. 通过使用不同参数组合的实验, 统计分析尺度调节因子对算法的影响, 实验结果表明 2 种改进算法通过适当的速度调节可以提升 MQHOA 的性能.

为了评估所提出方法的整体性能, 与 MQHOA 以及其他流行算法在基准测试函数上针对 30 维问题进行了实验比较. 计算平均误差、成功率和平均执行时间并分别进行排名. 实验结果表明, 尺度动态速度调节可以通过平衡探索和开发来提高 MQHOA 的性能, 边界映射策略对性能改进有积极作用, 负向反弹策略会带来更大的竞争力.

本文提出的算法仍然存在一些局限性. 首先, 改进算法的平均执行时间仍有提升空间. 此外, 增加了算法中的参数数量, 参数的各种组合会带来不同的效果, 因此有必要分析参数组合以获得更好的性能. 另外, 尺度反映了优化算法搜索粒度, 多种尺度调整策略和规则值得进一步研究. 这些问题都是未来的研究重点.

参考文献

- [1] JIANG M, HUANG Z Q, QIU L M, et al. Transfer learning-based dynamic multiobjective optimization algorithms

- [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(4): 501-514
- [2] 王鹏, 黄焱, 任超, 等. 多尺度量子谐振子高维函数全局优化算法[J]. 电子学报, 2013, 41(12): 2468-2473.
WANG P, HUANG Y, REN C, et al. Multi-scale quantum harmonic oscillator for high-dimensional function global optimization algorithm[J]. Acta Electronica Sinica, 2013, 41(12): 2468-2473. (in Chinese)
- [3] FINNILA A B, GOMEZ M A, SEBENIK C, et al. Quantum annealing: A new method for minimizing multidimensional functions[J]. Chemical Physics Letters, 1994, 219(5/6): 343-348.
- [4] 燕京京, 王鹏, 范家兵, 等. 基于量子谐振子模型的聚类中心选取算法[J]. 电子学报, 2016, 44(2): 405-412.
YAN J J, WANG P, FAN J B, et al. Clustering center selecting algorithm based on quantum harmonic oscillator model[J]. Acta Electronica Sinica, 2016, 44(2): 405-412. (in Chinese)
- [5] MU L, QU X M, WANG P. Application of multi-scale quantum harmonic oscillator algorithm for multifactor task allocation problem in WSANs[C]//2017 13th IEEE International Conference on Control & Automation. Ohrid, Macedonia: IEEE, 2017: 1004-1009.
- [6] 史忠亚, 吴华, 沈文迪, 等. 考虑射频隐身雷达功率自适应管控方法[J]. 现代雷达, 2017, 39(10): 6-10.
SHI Z Y, WU H, SHEN W D, et al. A method of radar power self-adapting control considering radio frequency stealth[J]. Modern Radar, 2017, 39(10): 6-10. (in Chinese)
- [7] 王鹏, 黄焱. 多尺度量子谐振子优化算法物理模型[J]. 计算机科学与探索, 2015, 9(10): 1271-1280.
WANG P, HUANG Y. Physical model of multi-scale quantum harmonic oscillator optimization algorithm[J]. Journal of Frontiers of Computer Science and Technology, 2015, 9(10): 1271-1280. (in Chinese)
- [8] 王鹏, 黄焱, 袁亚男, 等. 多尺度量子谐振子算法的收敛特性[J]. 电子学报, 2016, 44(8): 1988-1993.
WANG P, HUANG Y, YUAN Y N, et al. Convergence characteristics of multi-scale quantum harmonic oscillator algorithm[J]. Acta Electronica Sinica, 2016, 44(8): 1988-1993. (in Chinese)
- [9] 黄焱, 王鹏, 程琨, 等. 多尺度量子谐振子优化算法的并行性研究[J]. 通信学报, 2016, 37(9): 68-74.
HUANG Y, WANG P, CHENG K, et al. Parallelism of multi-scale quantum harmonic oscillator algorithm[J]. Journal on Communications, 2016, 37(9): 68-74. (in Chinese)
- [10] 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[J]. 自动化学报, 2016, 42(2): 235-245.
LU Z J, AN J X, WANG P. Partition-based MQHOA for multimodal optimization[J]. Acta Automatica Sinica, 2016, 42(2): 235-245. (in Chinese)
- [11] 韩虎, 王鹏, 程琨, 等. 基于多尺度量子谐振子算法的云计算任务调度[J]. 计算机应用, 2017, 37(7): 1888-1892.
HAN H, WANG P, CHENG K, et al. Task scheduling algorithm for cloud computing based on multi-scale quantum harmonic oscillator algorithm[J]. Journal of Computer Applications, 2017, 37(7): 1888-1892. (in Chinese)
- [12] SUN J, FANG W, WU X J, et al. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection[J]. Evolutionary Computation, 2012, 20(3): 349-393.
- [13] BROOKE J, BITKO D, ROSENBAUM T F, et al. Quantum annealing of a disordered magnet[J]. Science, 1999, 284(5415): 779-781.
- [14] WANG P, YE X G, LI B, et al. Multi-scale quantum harmonic oscillator algorithm for global numerical optimization[J]. Applied Soft Computing, 2018, 69: 655-670.
- [15] SHANKAR R. Principles of Quantum Mechanics[M]. 2nd ed. New York, NY, USA: Springer, 1994.
- [16] LI J Z, TAN Y. The bare bones fireworks algorithm: A minimalist global optimizer[J]. Applied Soft Computing, 2018, 62: 454-462.
- [17] MAURICIO Zambrano-Bigiarini, MAURICE Clerc, RODRIGO Rojas. Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements[C]//2013 IEEE Congress on Evolutionary Computation. Cancun, Mexico: IEEE, 2013: 2337-2344.
- [18] LI J Z, TAN Y. Loser-out tournament-based fireworks algorithm for multimodal function optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(5): 679-691.

作者简介



穆 磊 男, 1982年2月出生于河南省三门峡市. 2011年在北京理工大学获得工学博士学位, 现为西南民族大学讲师. 研究方向为群体智能、物联网技术.

E-mail: truemoller@126.com



王 鹏(通讯作者) 男, 1975年8月出生于四川省乐山市. 现为西南民族大学教授、中科院成都计算所博士生导师. 研究方向为智能算法、云计算、并行计算.

E-mail: wp002005@163.com