

# 基于量子自由粒子模型的优化算法框架

王 鹏<sup>1</sup>, 杨云亭<sup>2,3</sup>

(1. 西南民族大学计算机科学与技术学院, 四川成都 610225; 2. 中国科学院成都计算机应用研究所, 四川成都 610041;  
3. 中国科学院大学, 北京 100049)

**摘 要:** 基于量子系统下的自由粒子模型, 提出了多尺度自由粒子优化算法 (Multi-scale Free Particle Optimization Algorithm, MFPOA), 并在物理模型的基础之上研究了该算法的内部机制. 通过类比量子系统和优化系统, 将优化问题的求解过程转化成粒子在微观系统下的运动过程. 通过在 MATLAB 仿真平台上对自由粒子优化算法的参数设置进行了研究, 并分析了与同类搜索机制的算法的区别. 最后通过实验得出, MFPOA 更适合求解单模简单函数, 求解复杂多模函数需要更多的迭代次数.

**关键词:** 自由粒子; 谐振子; 均匀分布采样; 高斯采样; 波函数

**中图分类号:** TP301.6      **文献标识码:** A      **文章编号:** 0372-2112 (2020)07-1348-07

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2020.07.013

## Optimization Algorithm Framework Based on Quantum Free Particle Model

WANG Peng<sup>1</sup>, YANG Yun-ting<sup>2,3</sup>

(1. School of Computer Science and Technology, Southwest Minzu University, Chengdu, Sichuan 610225, China;  
2. Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu, Sichuan 610041, China;  
3. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Based on the free particle model of quantum system, the Multi-scale Free Particle Optimization Algorithm (MFPOA) is proposed, and the internal mechanism of the algorithm is studied on the basis of the physical model. Through analogy between quantum system and optimization system, the solving process of optimization problem is transformed into the motion process of particles under the microscopic system. The parameter setting of free particle optimization is studied on MATLAB simulation platform, and the differences between the algorithm and similar search mechanism are analyzed. Finally, experiments show that MFPOA is more suitable for solving single-mode functions, and more iterations are needed to solve complex multi-mode functions.

**Key words:** free particle; harmonic oscillator; uniformly distributed sampling; Gaussian sampling; wave function

### 1 引言

最优化领域对社会生产起着至关重要的作用, 优化算法的研究也一直受到学者的关注. 随着量子理论的发展, 优化算法也受到了量子理论的启发, 并使用量子机制改进算法. 量子退火算法<sup>[1]</sup>在模拟退火算法<sup>[2]</sup>的基础之上加入了量子能级跃迁过程中的隧道效应, 改善了算法跳出局部最优的能力. 量子进化算法<sup>[3]</sup>在进化算法<sup>[4]</sup>中加入量子比特编码和量子旋转门策略对个体进行演化, 增强了算法的求解效率. 量

子粒子群算法<sup>[5]</sup>在粒子群优化算法<sup>[6]</sup>中引入量子空间中的不确定性, 使用波函数的概率解释指导粒子的搜索, 改进了算法的求解能力. 多尺度量子谐振子优化算法 (Multi-Scale Quantum Harmonic Oscillator Optimization Algorithm, MQHOA)<sup>[7]</sup>同样引入量子力学中的不确定性, 并将这种不确定性具体到微观系统下的谐振子的空间概率分布, 通过概率分布在可行解空间中进行采样搜索, 实现优化问题求解的过程. 同时, 复杂的算法结构使算法的适用性和可理解性下降, 因此骨架模型的研究也为算法的优化提供了方向. 2003年,

Kennedy 提出粒子群算法的简化模型骨架粒子群算法<sup>[8]</sup>,改进粒子群算法<sup>[9]</sup>中产生新粒子的速度公式,增强了粒子从全局搜索到局部搜索的能力. 2009 年, Omran 等人提出骨架差分进化算法<sup>[10]</sup>,该算法借鉴骨架粒子群算法中粒子更新方法,改善了差分算法<sup>[11]</sup>中参数敏感的问题. 2013 年,有学者提出一种新的高斯骨干差分进化算法<sup>[12]</sup>,其借鉴骨架粒子群算法中的高斯过程产生粒子的方法,通过自适应调整策略选择交叉概率,在较高的交叉概率下,使用高斯变异产生新个体进行搜索. 2017 年,在烟花算法<sup>[13]</sup>的基础上提出了骨干烟花算法(Bare Bones Fireworks Algorithm, BBFWA)<sup>[14]</sup>,采用了均匀分布的采样方式和搜索空间的动态调整的策略,减少了算法的控制参数,使得算法结构更加简单高效.

本文提出了一种基于量子系统的骨架优化算法,多尺度自由粒子优化算法(Multi-scale Free Particle Optimization Algorithm, MFPOA). MFPOA 结合量子理论中的不确定原理和骨架模型的特点,将优化问题解的形式以一种概率分布表示,通过概率分布采样和尺度迭代逐渐逼近函数的全局最优解.

## 2 MFPOA 算法的物理模型

将优化算法的求解过程中随机搜索点类比量子系统中的粒子,搜索点的搜索策略类比粒子在空间位置中的移动,通过量子系统和优化算法的映射关系,实现对优化问题的求解. 粒子在量子系统下的移动可以由波函数描述,波函数可以通过薛定谔方程进行求解.

在量子系统中薛定谔方程写作如下形式:

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

其中,  $\hbar = h/(2\pi)$ ,  $h$  为普朗克常数,  $m$  为粒子的质量,  $V(x)$  为粒子的势能函数,  $\psi(x)$  为粒子的波函数,  $E$  为粒子的能量. 在势阱约束下,通过薛定谔方程即可求解粒子的波函数. 在优化问题中,将目标函数作为势阱约束,带入薛定谔方程可以求得优化函数约束下的波函数,依据波函数在量子系统中的概率意义指导优化算法的搜索过程. 一般情况下,在复杂势阱约束下很难通过薛定谔方程准确的求出对应的波函数,因此本研究采用目标函数泰勒展开式的近似形式来简化薛定谔方程中势阱函数.

将优化的目标函数作为薛定谔方程式中的势阱函数,进而得到:

$$\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + f(x)\right)\psi(x) = E\psi(x) \quad (2)$$

采用目标函数在最优值位置处的泰勒展开式的零阶近似式,可以求出相应的波函数. 由于零阶近似项为

常数,为简化求解,设常数为 0,将这一条件带入薛定谔方程转化为求解粒子 0 势阱约束下的波函数,此时的薛定谔方程转化为如下形式:

$$\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\right)\psi(x) = E\psi(x) \quad (3)$$

通过式(3),解二阶微分方程可以求解出对应的波函数形式为

$$\psi(x) = C_1 e^{i\sqrt{\frac{2mE}{\hbar^2}}x} + C_2 e^{-i\sqrt{\frac{2mE}{\hbar^2}}x} \quad (4)$$

其中  $C_1$  和  $C_2$  是任意实常数. 根据波函数的概率解释,计算出粒子在空间位置  $x$  处出现的概率如下所示:

$$|\psi(x)|^2 = C_1^2 + C_2^2 + C_1^* C_2 + C_1 C_2^* \quad (5)$$

由式(5)可知,波函数表示的概率含义是一个常数,即粒子以同等概率出现在任意位置. 同时物理中波函数的模的平方满足归一化的条件,即  $\iiint |\psi|^2 dV = 1$ .

在自由粒子的物理模型下,将实际中的优化问题转化为量子系统下自由粒子的运动形式,根据波函数的概率统计意义模拟粒子的存在位置.

## 3 MFPOA 的算法框架

根据自由粒子波函数的物理模型,得出算法的基本流程. 优化问题在多尺度自由粒子优化算法下的求解过程分为两步:(1)在同一个尺度上进行精细搜索,实现该尺度下搜索区域的聚焦;(2)在不同尺度下进行充分采样,实现函数多区域信息的获得. 在一定程度上平衡了算法开发和勘探的能力.

多尺度自由粒子优化算法通过尺度的迭代操作实现搜索区域聚焦,一次迭代的基本过程为:随机生成  $k$  个采样中心,以这  $k$  个点为中心分别进行当前尺度内的均匀分布采样,并更新采样中心的位置. 将  $k$  个采样中心中的最优函数值作为算法所能求解到的最优值,算法在当前尺度下最优值不变的次数达到预定的次数时,进行尺度收缩,并重复上述步骤. 直到尺度缩小到指定的精度,算法结束. 优化算法基本工作流程的伪代码如下.

Begin

- 1: Initialize  $k, \max, \min, \delta_{\min}, \delta = \max - \min, c$
- 2: Randomly generate  $x_i (i = 1, 2, \dots, k) \in [\min, \max]$
- 3: Calculate current best value  $f(x)^{best} = \min(f(x))$
- 4: Do
- 5:  $\forall x_i$ , generate  $x'_i \sim x_i + U(-\frac{\delta}{2}, \frac{\delta}{2})$
- 6:  $\forall x_i$  and  $x'_i$ , if  $f(x'_i) < f(x_i)$  then  $x_i = x'_i$
- 7: Update the worst solution  $x^{worst} = x^{\min}$
- 8: If numbers  $(\min(f(x)) = f(x)^{best}) > c$
- 9:  $\delta: \delta \leftarrow \delta/2$

10: End  
 11: Update  $f(x)^{best}$ :  $f(x)^{best} \leftarrow \min(f(x))$   
 12: While ( $\delta_{min} \leq \delta$ )  
 13: Output  $x^{best}$  and  $f(x^{best})$   
 End

其中  $k$  为采样参数,  $c$  为判定当前尺度下稳定状态的参数.  $[\min, \max]$  为目标函数在搜索空间中每一维的定义域,  $\delta_{min}$  为最小的搜索区域,  $\delta$  为搜索过程中的尺度. 初

始的尺度设定为搜索区间的大小,  $k$  个采样点在当前尺度下的搜索区间内随机生成. 尺度迭代过程是自由粒子优化算法的主要过程, 通过不断的迭代实现算法从大尺度向小尺度方向的收敛.

#### 4 仿真实验与分析

本文中采用的测试函数如表 1 所示, 其中  $f5 \sim f10$  为多模函数, 其余的均为单模函数.

表 1 测试函数描述

函数代号	函数名	函数表达式
$f1$	Sphere function	$f(x) = \sum_{i=1}^d x_i^2$
$f2$	Quadric function	$f(x) = \sum_{i=1}^d \left( \sum_{j=1}^i x_j \right)^2$
$f3$	Sum squares function	$f(x) = \sum_{i=1}^d ix_i^2$
$f4$	Zakharov function	$f(x) = \sum_{i=1}^d x_i^2 + \left( \sum_{i=1}^d 0.5ix_i \right)^2 + \left( \sum_{i=1}^d 0.5ix_i \right)^4$
$f5$	Levy function	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)]$ $w_i = 1 + \frac{x_i - 1}{4}, i = 1, 2, \dots, d$
$f6$	Ackley function	$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1)$ $a = 20, b = 0.2, c = 2\pi$
$f7$	Griewank function	$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$f8$	Alpine function	$f(x) = \sum_{i=1}^n  x_i \sin(x_i) + 0.1x_i $
$f9$	Rastrigin function	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$
$f10$	Rosenbrock function	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$f11$	Dixon and price function	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2$
$f12$	Rotated Hyper-ellipsoid function	$f(x) = \sum_{i=1}^d \sum_{j=1}^i x_j^2$
$f13$	Ellipsoidal function	$f(x) = \sum_{i=1}^d (x_i - i)^2$
$f14$	Sum of different power	$f(x) = \sum_{i=1}^d  x_i ^{i+1}$

#### 4.1 采样个数的影响

采样个数代表了算法求解过程中较优解区域的数目. 算法首先通过随机生成  $k$  个采样点, 然后再分别以每个采样点为中心进行均匀分布采样, 生成新的点, 来探测该种群区域内能取得更优值的点. 本小节测试了

算法在稳定判据  $c$  值为 100 次在不同的 10 维函数的性能. 采样点个数分别设置为 1, 5, 10, 30, 实验数据如表 2 所示. 成功率是 30 次重复实验中求解精度达到  $1E-5$  的实验次数占比, 求解精度是 30 次实验中找到最优值的平均值.

表 2 不同采样个数  $k$  下成功率和求解精度变化

	成功率(%)				求解精度			
	$k=1$	$k=5$	$k=10$	$k=30$	$k=1$	$k=5$	$k=10$	$k=30$
$f_1$	100.00	100.00	100.00	100.00	$2.26E-13$	$1.28E-13$	$9.87E-14$	$7.3E-14$
$f_2$	100.00	100.00	100.00	100.00	$3.98E-13$	$2.05E-13$	$1.22E-13$	$1.84E-13$
$f_3$	100.00	100.00	100.00	100.00	$1.53E-12$	$9.13E-13$	$8.92E-13$	$5.2E-13$
$f_4$	100.00	100.00	100.00	100.00	$3.67E-13$	$2.99E-13$	$1.42E-13$	$1.33E-13$
$f_5$	6.67	33.33	66.67	96.67	$2.85E-13$	$1.64E-13$	$8.68E-14$	$1.11E-13$
$f_6$	46.67	96.67	100.00	100.00	$6.72E-07$	$4.35E-07$	$4.73E-07$	$3.96E-07$
$f_7$	0.00	0.00	0.00	6.67	$1.97E-02$	$9.86E-03$	$7.39E-03$	$3.73E-14$
$f_8$	0.00	0.00	6.67	13.33	$6.52E-03$	$1.68E-03$	$1.32E-07$	$1.00E-07$
$f_9$	0.00	0.00	0.00	0.00	$1.79E+01$	$5.96E+00$	$5.96E+00$	$5.96E+00$
$f_{10}$	90.00	100.00	100.00	93.33	$1.26E-07$	$1.23E-07$	$1.22E-07$	$1.29E-07$
$f_{11}$	0.00	16.67	23.33	83.33	$6.67E-01$	$3.84E-11$	$3.63E-11$	$2.94E-11$
$f_{12}$	100.00	100.00	100.00	100.00	$1.14E-11$	$1.11E-11$	$7.63E-12$	$7.41E-14$
$f_{13}$	100.00	100.00	100.00	100.00	$1.85E-13$	$1.9E-13$	$1.69E-13$	$7.29E-14$
$f_{14}$	100.00	100.00	100.00	100.00	$4.97E-10$	$2.84E-10$	$1.54E-10$	$2.37E-10$

成功率数据显示,  $f_1, f_2, f_3, f_4, f_6, f_{10}, f_{12}, f_{13}, f_{14}$  函数的在采样点个数比较小的时候就达到了 100% 的成功率, 这说明  $k$  参数对这些函数的影响较小, 而且这些函数多为单模函数; 对  $f_5, f_7, f_8$  和  $f_{11}$  函数, 成功率随  $k$  值增大有增加的趋势;  $k$  值在算法求解  $f_9$  时, 并没有体现出成功率增加的趋势, 说明算法在较复杂的函数求解上更难跳出局部最优. 最小值精度数据显示,  $k$  值的增加可以带来求解精度的增加, 但精度增加的程度并不是很大. 通过增加粒子方式在带来更多的优化信息, 同时也会存在性能上升的上限.  $k$  值增加很大不仅不能帮助算法求解更加精确的解, 反而会使算法在迭代过程中函数的进化次数加大, 进而增加算法的求解时间.

#### 4.2 稳态判据值的影响

稳态判据  $c$  在算法中控制尺度进行缩减, 以实现尺度的收敛.  $c$  值是尺度迭代的最大次数, 表示在每一个尺度下搜索到的最优值不发生变化的次数. 本小节测试了采样个数为 30 的算法求解 10 维函数的性能,  $c$  值分别设置为 2, 10, 30, 100, 实验数据如表 3. 实验数据表

明算法求解的精度在随着稳定判据  $c$  值的增加而提高. 当在稳定判据设置为 2 时, 算法求解性能较差. 此时随机均匀分布采样在没有详尽搜索的情况下, 缩减尺度, 会使算法早熟, 陷入局部最优. 相对使用尺度稳定判据更大的  $c$  值, 可以在一定程度上削减了这种偶然性的影响, 实现当前尺度下的充分搜索. 当算法的稳定判据设置的过大, 求解精度不会提升很大, 反而会带来很多次无效搜索.

成功率实验数据中, 同样显示算法在 30 次重复实验中求解成功率随着  $c$  值的增加而增大, 成功率越高, 说明算法越稳定. 当  $c$  值达到 30 时, 单模函数的求解成功率是 100%, 多模函数  $f_7 \sim f_9$  的成功率却很差. 当尺度的稳定判据值为 100 次, 大部分函数达到 77% 以上的成功率, 但部分复杂的多模函数效果仍然较差. 针对具体的优化问题, 可以通过设置更多次的独立实验, 选择求解结果中的最优值作为优化问题的解.  $c$  值越大, 稳定判据的约束力越强, 当前搜索区间内的搜索到的最好值在最大概率程度上会被保证, 相应的函数求解时间也会变长.

表3 算法不同  $c$  值的设置对求解优化函数的成功率和最优值精度的影响

	成功率(%)				求解精度			
	$c=2$	$c=10$	$c=30$	$c=100$	$c=2$	$c=10$	$c=30$	$c=100$
$f_1$	0.00	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	$3.00\text{E}+00$	$3.08\text{E}-13$	$1.37\text{E}-13$	<b><math>1.20\text{E}-13</math></b>
$f_2$	0.00	30.00	<b>100.00</b>	<b>100.00</b>	$1.23\text{E}+01$	$2.30\text{E}-09$	$1.17\text{E}-12$	<b><math>1.61\text{E}-13</math></b>
$f_3$	0.00	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	$3.16\text{E}+01$	$1.17\text{E}-11$	$9.82\text{E}-13$	<b><math>6.50\text{E}-13</math></b>
$f_4$	0.00	87.00	<b>100.00</b>	<b>100.00</b>	$3.54\text{E}+01$	$2.23\text{E}-11$	$5.50\text{E}-13$	<b><math>2.40\text{E}-13</math></b>
$f_5$	0.00	3.00	63.00	97.00	$1.51\text{E}+00$	$3.65\text{E}-10$	$3.45\text{E}-13$	<b><math>1.06\text{E}-13</math></b>
$f_6$	0.00	47.00	<b>100.00</b>	<b>100.00</b>	$4.89\text{E}+00$	$8.38\text{E}-07$	$5.44\text{E}-07$	<b><math>3.38\text{E}-07</math></b>
$f_7$	0.00	27.00	20.00	3.00	$1.87\text{E}-01$	$1.02\text{E}-13$	$5.65\text{E}-14$	<b><math>7.47\text{E}-14</math></b>
$f_8$	0.00	0.00	3.00	7.00	$2.31\text{E}+00$	$1.25\text{E}-01$	$7.27\text{E}-07$	<b><math>1.31\text{E}-07</math></b>
$f_9$	0.00	0.00	0.00	0.00	$4.36\text{E}+01$	$7.96\text{E}+00$	<b><math>3.98\text{E}+00</math></b>	$4.97\text{E}+00$
$f_{10}$	0.00	0.00	93.00	<b>100.00</b>	$4.96\text{E}+02$	$8.17\text{E}-02$	$8.20\text{E}-07$	<b><math>1.32\text{E}-07</math></b>
$f_{11}$	0.00	0.00	47.00	<b>77.00</b>	$8.33\text{E}+01$	$7.70\text{E}-04$	$1.10\text{E}-10$	<b><math>3.02\text{E}-11</math></b>
$f_{12}$	0.00	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	$2.81\text{E}+02$	$2.59\text{E}-11$	$9.86\text{E}-12$	<b><math>5.68\text{E}-12</math></b>
$f_{13}$	0.00	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	$1.16\text{E}+01$	$3.44\text{E}-13$	$2.34\text{E}-13$	<b><math>1.05\text{E}-13</math></b>
$f_{14}$	0.00	67.00	<b>100.00</b>	<b>100.00</b>	$2.71\text{E}+01$	$2.02\text{E}-06$	$4.44\text{E}-09$	<b><math>2.12\text{E}-10</math></b>

### 4.3 算法收敛性分析

本节的算法收敛性分析借助算法在求解优化问题时粒子分布的可视化过程展示算法的收敛过程. 在测试函数中选择了 Ackley 展示算法在求解过程中粒子的收敛过程, 如图 1 所示. 其中上面的图为粒子在二维的

优化函数中的图示, 采样点分别在初始搜索尺度(实验中的搜索区间为  $[-10, 10]$ ), 尺度进化 2 次, 4 次, 8 次后的 30 个采样点的分布. 从图中可以明显的看到粒子随着尺度的迭代, 会慢慢的向最小值收敛.

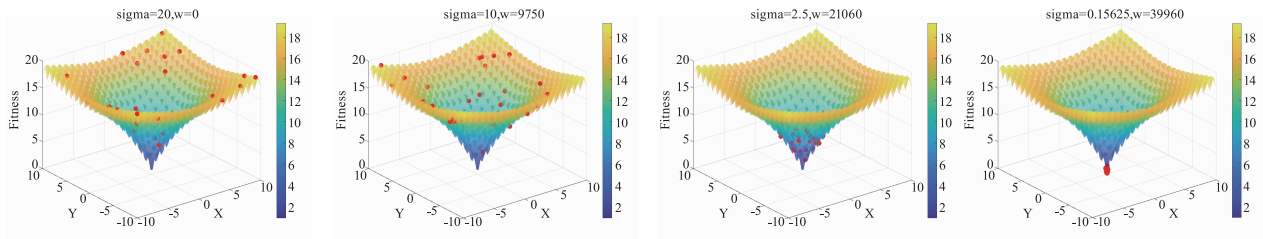


图1 Ackley函数的算法收敛过程立体图

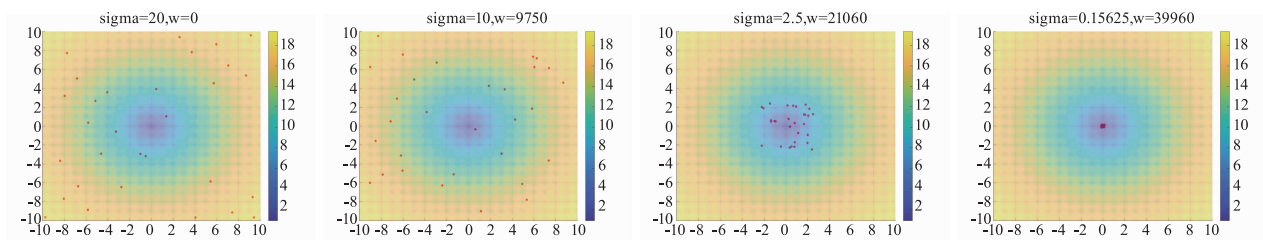


图2 Ackley函数的算法收敛过程俯视图

## 5 拓展实验

MFPOA 模拟量子自由粒子的运动形式在当前搜索区域进行均匀分布采样实现粒子位置的更新, MQHOA 在谐振子的启发下, 使用高斯采样进行粒子位置的更新. 二者不同的采样方式使得算法性能有一定的区别,

同时 MFPOA 通过尺度迭代进行算法收敛, MQHOA 通过能级稳定, 谐振子迭代和尺度迭代达到最后算法的收敛. 同样基于均匀分布采样方式的 BBFWA 使用单粒子在动态变化的搜索空间中进行搜索, 实现优化问题的寻优. 本节针对三种算法分别测试了函数在 2, 10, 30, 60 维(dim) 下三种算法的成功率, 如表 4.

表 4 MFPOA(A)、MQHOA(B)和 BBFWA(C)算法在不同维度的成功率(%)对比

	dim = 2			dim = 10			dim = 30			dim = 60		
	A	B	C	A	B	C	A	B	C	A	B	C
<i>f</i> 1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	<b>0.00</b>	<b>40.00</b>	<b>100.00</b>
<i>f</i> 2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
<i>f</i> 3	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	<b>100.00</b>	<b>50.00</b>	<b>0.00</b>
<i>f</i> 4	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	<b>100.00</b>	<b>100.00</b>	<b>0.00</b>
<i>f</i> 5	100.00	100.00	100.00	<b>36.67</b>	<b>100.00</b>	<b>66.67</b>	<b>0.00</b>	<b>73.33</b>	<b>0.00</b>	0.00	0.00	0.00
<i>f</i> 6	100.00	100.00	100.00	100.00	100.00	96.67	<b>80.00</b>	<b>96.67</b>	<b>76.67</b>	<b>0.00</b>	<b>13.33</b>	<b>0.00</b>
<i>f</i> 7	<b>100.00</b>	<b>100.00</b>	<b>86.67</b>	<b>16.67</b>	<b>100.00</b>	<b>0.00</b>	<b>100.00</b>	<b>100.00</b>	<b>3.33</b>	<b>100.00</b>	<b>100.00</b>	<b>50.00</b>
<i>f</i> 8	100.00	100.00	100.00	<b>20.00</b>	<b>6.67</b>	<b>3.33</b>	0.00	0.00	0.00	0.00	0.00	0.00
<i>f</i> 9	100.00	100.00	100.00	<b>0.00</b>	<b>30.00</b>	<b>0.00</b>	0.00	0.00	0.00	0.00	0.00	0.00
<i>f</i> 10	100.00	100.00	100.00	<b>100.00</b>	<b>96.67</b>	<b>0.00</b>	<b>3.33</b>	<b>0.00</b>	<b>0.00</b>	0.00	0.00	0.00
<i>f</i> 11	100.00	100.00	100.00	<b>70.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	0.00	0.00	0.00	0.00	0.00
<i>f</i> 12	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00
<i>f</i> 13	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>f</i> 14	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

表中粗体数据是有差异数据,表中的数据表明,随着优化函数的维度增加,三种算法求解的成功率均有下降.与MQHOA的采样方式相比,自由粒子尺度范围内的位置概率均等,谐振子尺度范围内位置概率有中心集中的趋势,理论上在相同尺度下自由粒子的均匀采样更容易跳出局部最优,因为在边缘区域均匀分布的概率高于高斯分布的概率.但在部分多模函数中成功率要低于MQHOA,一定程度上说明了使用复杂函数的0阶近似不够准确.与BBFWA相比,多数情况下,MFPOA比BBFWA的成功率高.说明自由粒子算法在大部分情况下的跳出局部最优的能力更强.相比BBFWA中的单粒子搜索方式和尺度根据一次的采样信息进行动态调整,MFPOA的多粒子搜索方式一定程度上保证了算法对全局信息的掌握,同时尺度次数据保证了在某个尺度下的充分搜索.

## 6 总结

本文通过量子自由粒子模型研究了量子算法的骨架模型.自由粒子模型的骨架特点主要体现在三个方面,第一,算法结构中包含一个尺度迭代过程;第二,算法的采样方式采用随机均匀分布;第三,算法的可调参数少;第四,算法的可解释性强,算法通过尺度控制均匀分布的搜索,在大区域下搜索不到更好解时进行尺度收缩.通过与烟花算法的骨架模型BBFWA进行对比,我们发现尺度判据的改变虽然可以加速算法的收敛,但不能给高维函数的求解带来性能上的提升.MFPOA与MQHOA都是通过势阱等效,泰勒展开,将优

化问题转化为量子系统下的求解波函数问题,二者有一定的相似性,但采样方式区别明显,求解性能也不尽相同.

未来的工作中还需要进一步在算法物理模型的启发下,深入研究优化算法在其他势阱条件下的性能和联系.

## 参考文献

- [1] Kadowaki T, Nishimori H. Quantum annealing in the transverse Ising model[J]. Physical Review E, 1998, 58(5): 5355-5363.
- [2] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [3] 王凌. 量子进化算法研究进展[J]. 控制与决策, 2008, 23(12): 1321-1326.  
WANG Ling. Advances in quantum-inspired evolutionary algorithms[J]. Control and Decision, 2008, 23(12): 1321-1326. (in Chinese)
- [4] 姚新, 陈国良, 徐惠敏, 等. 进化算法研究进展[J]. 计算机学报, 1995, 18(9): 694-706.  
YAO Xin, CHEN Guo-liang, XU Hui-min, et al. Research progress of evolutionary algorithm[J]. Chinese Journal of Computers, 1995, 18(9): 694-706. (in Chinese)
- [5] 方伟, 孙俊, 谢振平, 等. 量子粒子群优化算法的收敛性分析及控制参数研究[J]. 物理学报, 2010, 59(6): 3686-3694.  
FANG Wei, SUN Jun, XIE Zhen-ping, et al. Convergence

- analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter [J]. Acta Physica Sinica, 2010, 59(6): 3686 – 3694. (in Chinese)
- [6] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection [J]. Information Processing Letters, 2003, 85(6): 317 – 325.
- [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 Peng, HUANG Yan. Physical model of multi-scale quantum harmonic oscillator optimization algorithm [J]. Journal of Frontiers of Computer Science & Technology, 2015, 9(10): 1271 – 1280. (in Chinese)
- [8] Kennedy J. Bare bones particle swarms [A]. Proceedings of the 2003 IEEE Swarm Intelligence Symposium [C]. IEEE, 2003. 80 – 87.
- [9] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58 – 73.
- [10] Omran M G H, Engelbrecht A P, Salman A. Bare bones differential evolution [J]. European Journal of Operational Research, 2009, 196(1): 128 – 139.
- [11] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4 – 31.
- [12] Wang H, Rahnamayan S, Sun H, et al. Gaussian bare-bones differential evolution [J]. IEEE Transactions on Cybernetics, 2013, 43(2): 634 – 647.
- [13] Li J, Tan Y. The bare bones fireworks algorithm: A minimalist global optimizer [J]. Applied Soft Computing, 2018, 62: 454 – 462.
- [14] Tan Y, Zhu Y. Fireworks algorithm for optimization [A]. International conference in swarm intelligence [C]. Berlin, Heidelberg: Springer, 2010. 355 – 364.

#### 作者简介



王 鹏 (通信作者) 男, 1975 年 8 月出生, 四川乐山人. 现为西南民族大学教授、中科院成都计算所博士生导师, 研究方向为智能优化算法、云计算、并行计算.  
E-mail: wp002005@163.com



杨云亭 女, 1994 年 9 月出生, 河北定州人. 2017 年毕业于黑龙江工程学院计算机科学与技术学院, 后进入中国科学院成都计算机应用研究所攻读硕士学位. 研究方向为智能优化算法.  
E-mail: 1424127847@qq.com