

## 多尺度量子谐振子优化算法的并行性研究

黄焱<sup>1</sup>, 王鹏<sup>2</sup>, 程琨<sup>3</sup>, 刘峰<sup>4</sup>

(1. 淮阴师范学院计算机科学与技术学院, 江苏 淮安 223300; 2. 西南民族大学计算机科学与技术学院, 四川 成都 610225;  
3. 中国科学院成都计算机应用研究所, 四川 成都 610041; 4. 成都信息工程大学并行计算实验室, 四川 成都 610225)

**摘要:** 多尺度量子谐振子优化算法(MQHOA, multi-scale quantum harmonic oscillator algorithm)是一种利用量子谐振子波函数构造的新的智能算法, 采样运算是 MQHOA 算法的基本运算单元和主要运算量, 采样运算的独立性赋予 MQHOA 算法内在并行性。通过对 MQHOA 算法群体参数和采样参数进行实验, 确定算法的并行粒度并提出多尺度量子谐振子并行算法 (MQHOA-P, multi-scale quantum harmonic oscillator parallel algorithm)。在由 10 个计算节点构成的集群上对 6 种标准测试函数进行实验, 通过改变计算节点数、函数维数和采样参数测试 MQHOA-P 算法的加速比, 实验结果表明, MQHOA-P 算法具有良好的加速比和扩展性, 可以在大规模集群中部署、运行。

**关键词:** 多尺度量子谐振子优化算法; 算法并行性; 加速比; 并行粒度; 函数优化

中图分类号: TP393

文献标识码: A

## Parallelism of multi-scale quantum harmonic oscillator algorithm

HUANG Yan<sup>1</sup>, WANG Peng<sup>2</sup>, CHENG Kun<sup>3</sup>, LIU Feng<sup>4</sup>

(1. School of Computer Science and Technology, Huaiyin Normal University, Huaian 223300, China;  
2. School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610225, China;  
3. Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China;  
4. Parallel Computing Lab, Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract:** MQHOA was a novel intelligent algorithm constructed by quantum harmonic oscillator's wave function. Sampling was the basic operation and main computational burden of MQHOA. The independence of sampling operation constructs MAHOA's parallelism. Parallel granularity was obtained by experiments of group parameter and sampling parameter, and MQHOA-P was proposed. Experiments were done in a cluster of ten nodes on six standard test functions. By changing node number, function dimension and sampling parameter, experiments of MQHOA-P's speed-up ratio were done. The experimental results show the good performance of MQHOA-P's speed-up ratio and expansibility. MQHOA-P can be deployed and run on multiple nodes in a large-scale cluster.

**Key words:** MQHOA, algorithm parallelization, speedup, parallel granularity, functional optimization

### 1 引言

MQHOA 算法的基本思想是利用函数优化问题与量子谐振子从高能态向基态收敛过程的相似性, 结合高斯二进信息采样方法, 将函数优化问题的求解过程分为量子谐振子 (QHO, quantum harmonic

oscillator) 收敛和多尺度收敛 (M 收敛) 2 个互相嵌套的收敛过程, 通过 QHO 收敛实现在当前尺度对搜索空间的逐步收缩定位, 通过 M 收敛逐渐提高搜索精度, 使目标函数在指定精度找到全局最优解<sup>[1]</sup>。

MQHOA 算法在求解函数优化<sup>[2]</sup>、组合优化<sup>[3]</sup>等问题时具有良好的性能, 并可以求解聚类中心点<sup>[4]</sup>

收稿日期: 2016-01-27; 修回日期: 2016-06-27

通信作者: 王鹏, wp002005@163.com

基金项目: 国家自然科学基金资助项目 (No.60702075); 模式识别与智能信息处理四川省高校重点实验室开放基金资助项目 (No.MSSB-2015-9)

**Foundation Items:** The National Natural Science Foundation of China (No.60702075), Sichuan Key Laboratory Open Foundation of Pattern Recognition and Intelligent Information Processing (No.MSSB-2015-9)

等应用问题。文献[1]介绍了 MQHOA 算法的基础模型，并对算法的收敛性进行研究；文献[5]通过研究 MQHOA 算法与经典谐振子和量子谐振子物理模型之间的对应关系对 MQHOA 算法的物理模型进行分析，介绍 MQHOA 算法中的量子隧道效应和测不准原理；文献[6]将 MQHOA 算法用于求解整数非线性规划问题，与 QPSO 算法进行对比，验证 MQHOA 算法的性能；文献[7]通过将聚类数据集投影到基于划分的网格，将聚类中心点问题转化为离散函数优化问题进行求解，使用 MQHOA 算法可以准确地找到聚类中心点位置，并且聚类数据集越大，找到的采样中心点位置越准确，文献[8]提出了具有能级稳定过程的 MQHOA 算法。

采样运算是 MQHOA 算法的基本运算单元，其运算量是 MQHOA 算法主要的运算量。采样运算的独立性赋予 MQHOA 算法内在的并行性。当求解高维函数优化问题时，随着函数维数的增加，函数解空间呈指数级增加，求解目标函数最优解所需的迭代次数也大幅增长，利用 MQHOA 算法的并行性可以有效地降低算法的运行时间。

本文通过对 MQHOA 算法的采样参数和群体参数进行实验分析，分析 MQHOA 算法的 3 种并行粒度，提出 MQHOA-P 算法，该算法的核心思想是每次迭代运算时主节点将本次迭代的采样中心点和当前尺度发送到从节点，从节点以接收到的采样中心点位置和尺度进行采样运算并在从节点内更新最优解位置，然后各从节点向主节点发起通信，将更新后的最优解位置汇总到主节点，完成一次迭代运算。通过对 6 种测试函数在不同的函数维数、运行节点数的实验分析，MQHOA 算法具有较强的并行性和扩展性。

## 2 MQHOA 算法流程与运行时间

### 2.1 MQHOA 算法流程

MQHOA 算法的基本流程如图 1 所示。

1) 初始化：初始化群体参数  $k$ 、采样参数  $m$ 、采样范围  $S$ 、初始尺度  $\sigma_0$  和最小尺度  $\sigma_{\min}$ ，并在采样范围内随机生成  $k$  个初始采样位置  $\{\vec{X}_0^1, \vec{X}_0^2, \dots, \vec{X}_0^k\}$ 。算法运行过程中将不断地更新这些采样位置，保留  $k$  个最优解。

2) 迭代运算：假设当前为第  $p$  次迭代，尺度为  $\sigma_p$ ， $k$  个当前最优解位置为  $\{\vec{X}_p^1, \vec{X}_p^2, \dots, \vec{X}_p^k\}$ ，以每

个采样位置  $\vec{X}_p^i$  为中心，分别按照高斯分布  $N(\vec{X}_p^i, \sigma_p^2)$  生成  $m$  个新的采样位置  $\{\vec{X}_p^{i1}, \vec{X}_p^{i2}, \dots, \vec{X}_p^{im}\}$ ，将  $km$  个新采样位置的目标函数值分别与当前的  $k$  个最优解位置的目标函数值进行比较，如果出现更优解则替换原采样位置，以上操作为一次迭代运算，完成本次迭代运算后的最优解位置记为  $\{\vec{X}_{p+1}^1, \vec{X}_{p+1}^2, \dots, \vec{X}_{p+1}^k\}$ 。

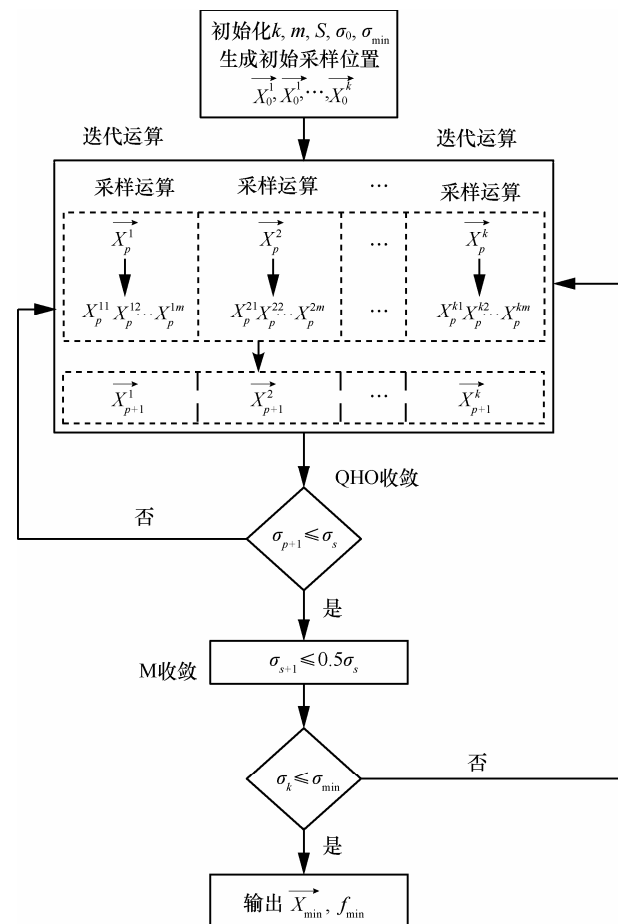


图 1 MQHOA 算法流程

以一个采样位置为中心根据采样参数和当前尺度生成  $m$  个新的高斯采样位置并对目标函数值进行比较是一次采样运算。采样运算是 MQHOA 算法的基本运算单元。采样运算仅取决于采样位置和当前尺度，采样运算之间没有任何关联关系，采样运算的独立性是 MQHOA 算法并行性的基础。

3) QHO 收敛：计算当前  $k$  个采样位置的标准差  $\sigma_{p+1}$ ，与算法当前的尺度  $\sigma_s$  进行比较。如果  $\sigma_{p+1} > \sigma_s$ ，则说明当前采样中心点尚未在当前尺度收敛，返回 2) 继续进行迭代运算；如果  $\sigma_{p+1} \leq \sigma_s$ ，则算法已在当前尺度收敛，进行 M 收敛过程。

4) M 收敛: 如果  $\sigma_{p+1} > \sigma_{\min}$ , 则将当前尺度减半  $\sigma_{s+1} = 0.5\sigma_s$ , 返回 2) 依次进行迭代运算和 QHO 迭代; 如果  $\sigma_{p+1} \leq \sigma_{\min}$ , 则说明当前的采样点的聚集程度已经满足算法计算精度的要求, 结束算法的收敛过程。

5) 输出结果: 将当前  $k$  个采样位置中的最优位置和目标函数值作为结果输出。

### 2.2 MQHOA 算法的运行时间分析

根据如图 1 所示的 MQHOA 算法流程, 可将算法划分为初始化、迭代运算、尺度判断、尺度收缩、精度判断和结果输出 6 个部分, 其编号、名称、单位运行时间和运算次数如表 1 所示。其中,  $A_1$ 、 $A_3$ 、 $A_4$ 、 $A_5$ 、 $A_6$  的单位运行时间为常数,  $A_2$  的单位运行时间由算法的群体参数  $k$  和采样参数  $m$  决定;  $A_1$ 、 $A_4$ 、 $A_5$ 、 $A_6$  的运算次数为常数,  $A_2$ 、 $A_3$  的运算次数为算法的迭代次数  $a_1$ 。

每次迭代运算  $A_2$  包含  $k$  次独立的采样运算  $B_1$ , 每次采样运算  $B_1$  包含  $m$  次独立的单位采样  $B_2$ , 单位采样的单位运行时间为  $D_2$ , 由此可得  $A_2$  的单位运行时间为  $a_1kmD_2$ 。

表 1 MQHOA 算法的运行时间

编号	名称	单位运行时间	运算次数
$A_1$	初始化	$C_1$	1
$A_2$	迭代运算	$C_2$	$a_1$
$A_3$	尺度判断	$C_3$	$a_1$
$A_4$	尺度收缩	$C_4$	$a_2$
$A_5$	精度判断	$C_5$	$a_3$
$A_6$	输出结果	$C_6$	1
$B_1$	采样运算	$D_1$	$a_1k$
$B_2$	单位采样	$D_2$	$a_1km$
$B_3$	节点通信	$D_3$	—

MQHOA 算法的总运行时间为

$$C_{\text{total}} = C_1 + a_1kmD_2 + a_1C_3 + a_2C_4 + a_3C_5 + C_6 \quad (1)$$

设  $C = C_1 + a_2C_4 + a_3C_5 + C_6$ ,  $C$  为常数, 则式 (1) 可简化为

$$T_0 = a_1kmD_2 + a_1C_3 + C \quad (2)$$

迭代运算占总运行时间的比例为

$$\frac{A_2}{T_0} = \frac{a_1kmD_2}{a_1kmD_2 + a_1C_3 + C} \quad (3)$$

MQHOA 算法运行时间由算法迭代次数  $a_1$ 、群

体参数  $k$  和采样参数  $m$  共同决定。MQHOA 算法的迭代次数取决于目标函数的复杂程度, 随着目标函数维度的增加, 函数解空间的数量呈指数级增长, 算法的迭代次数  $a_1$  显著增长, 为了快速、准确地获取全局最优解需要加大  $k$  和  $m$  的数值, 采样参数的运行时间会相应地增加, 由式 (3) 可知, 迭代运算的占比增大, 因此迭代运算是 MQHOA 算法最主要的运行时间。

## 3 MQHOA 算法的并行化特性

### 3.1 MQHOA 算法的并行方法

算法的并行方法取决于算法逻辑单元之间的依赖关系以及可并行化单元的内部结构<sup>[9]</sup>。对于 MQHOA 算法而言, 其迭代运算单元间的串行依赖关系和互相独立的采样运算单元的可并行特性决定了算法的并行方法。

设当前迭代为 MQHOA 算法的第  $p$  次迭代, 第  $p+1$  次迭代运算的采样中心点位置  $\{\overrightarrow{X_{p+1}^1}, \overrightarrow{X_{p+1}^2}, \dots, \overrightarrow{X_{p+1}^k}\}$  是第  $p$  次迭代运算更新的最优解位置; 第  $p+1$  次迭代运算的高斯采样尺度取决于  $\sigma_{p+1}$  与  $\sigma_s$  的比较结果, 如果  $\sigma_{p+1} \leq \sigma_s$ , 则采样尺度减半, 否则采样尺度不变; 算法是否执行完毕取决于  $\sigma_{p+1}$  与  $\sigma_{\min}$  的比较结果, 如果  $\sigma_{p+1} \leq \sigma_{\min}$ , 则算法终止运行。因此下一次迭代运算的采样中心位置、采样尺度和是否继续迭代均取决于当前迭代运算的结果, MQHOA 算法的迭代运算必须串行执行。

MQHOA 算法的一次迭代运算包含  $k$  次采样运算, 在每个采样中心点位置以当前尺度生成  $m$  个新解位置, 并与当前最优解位置进行比较, 采样运算之间互相独立, 可以在多个计算节点上并行运行。

MQHOA 算法的运行过程是由迭代运算的串行运行构成, 由于迭代运算是 MQHOA 算法的主要运算量, 其内含的采样运算是可并行化的, 可以通过将采样中心位置发送到多个并行计算节点, 使采样运算并行运行, 实现 MQHOA 算法的并行化。根据算法结构确定并行粒度是 MQHOA 算法并行化的核心。

### 3.2 MQHOA 算法的 3 种并行粒度

设 MQHOA 算法运行在由一个主节点和  $n$  个从节点构成的并行计算集群上。划分到从节点运行的任务规模是 MQHOA 算法的并行粒度, 直接影响算法的并行性和并行开销。根据图 1 所示的 MQHOA

算法流程，MQHOA 算法在迭代运算、尺度判断和精度判断时需要将上次迭代运算的结果进行汇总，因此 MQHOA 算法有以下 3 种并行粒度。

1) 粒度 1

当 MQHOA 算法按照粒度 1 进行并行化时，从节点仅完成本次迭代的并行运算，其并行化框架如图 2 所示。

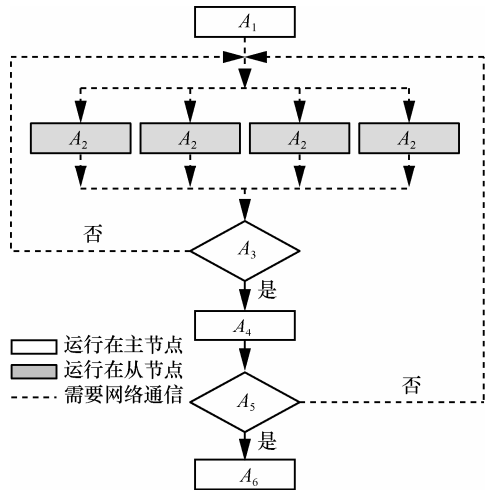


图 2 采用并行粒度 1 的并行化框架

与串行程序相比，这种方法将一次迭代运算分发到多个从节点并行计算，其采样中心点的生成方式、数量和收敛条件与串行程序完全相同。将采样中心点位置均分并按照粒度 1 进行并行化后，每个从节点的运算量是相同的，当从节点的计算性能相同时，每次迭代从节点的运行时间相等。

2) 粒度 2

当 MQHOA 算法按照粒度 2 进行并行化时，从节点完成当前尺度的并行运算，其并行化框架如图 3 所示。与串行程序相比，这种方法在当前尺度收敛时使用的采样中心点数量较少，导致当前尺度收敛的迭代次数增加、当前尺度正确收敛的概率降低，大幅增加算法的运行时间，降低获取全局最优解的概率。由于从节点独立完成当前尺度的收敛，每个从节点的迭代运算次数不确定，导致各个从节点的运算时间存在差异，MQHOA 算法的并行化时间由运算时间最长的节点决定。

3) 粒度 3

当 MQHOA 算法按照粒度 3 进行并行化时，从节点完成所有尺度的并行运算，其并行化框架如图 4 所示。与串行程序相比，这种方法将采样中心点分发到多个从节点独立运行 MQHOA 算法，运行完后

对结果进行汇总，因此同样会增加从节点的迭代次数并降低获取全局最优解的概率。从节点独立地完成所有尺度的收敛的运算时间存在差异，MQHOA 算法并行化的时间由运算时间最长的节点决定。

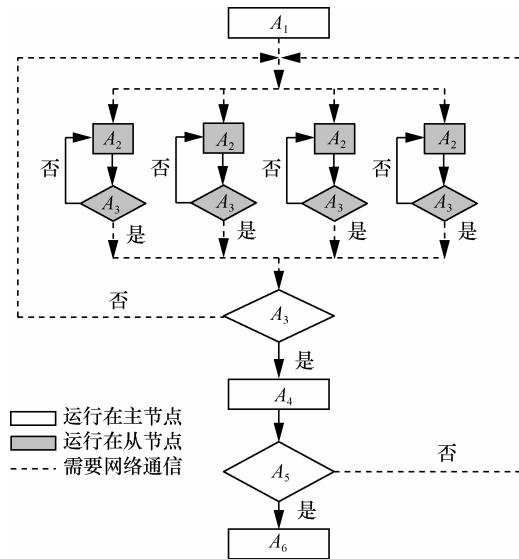


图 3 采用并行粒度 2 的并行化框架

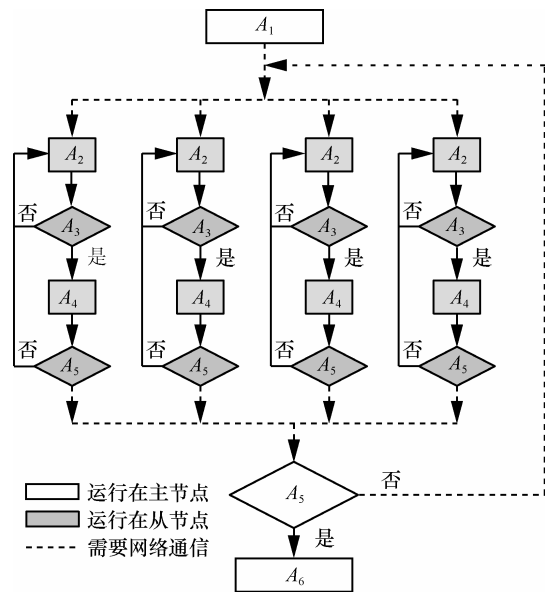


图 4 采用并行粒度 3 的并行化框架

3.3 根据采样参数选择 MQHOA 算法的并行粒度

采用粒度 2 和粒度 3 将采样中心分发到从节点独立地进行迭代运算，与采用粒度 1 进行并行化相比，每个从节点上运行的采样中心点数较少。MQHOA 算法在不同采样参数  $k$  下的性能决定了算法并行粒度的选择方法。

将 MQHOA 算法的参数  $m$  依次取 10、20、30、40、50、60、70、80、90、100、150、200、250、

300、500，参数  $k$  依次取 10、20、30、40、50、60、70、200、500、1 000，形成 150 个实验组合，对 3 维的 Griewank 函数在  $(-100,100)$  的解空间进行实验，每个实验组合分别进行 10 次重复实验，测试算法迭代次数如文献[1]中表 1 所示。

可以看出，当采用粒度 2、粒度 3 进行算法并行化时，从节点上的采样中心数为串程序的  $\frac{1}{n}$ ，群体参数  $k$  较小，MQHOA 算法落入局部最优区域的概率加大，算法求解的准确性下降；与此同时，从节点上的迭代次数大幅增加，算法运行时间大幅增加。因此，本文采用粒度 1 对 MQHOA 算法进行并行化研究。

#### 4 MQHOA-P 算法的运行流程

根据 MQHOA 算法流程和算法并行性分析，本节使用粒度 1 对 MQHOA 算法进行并行化，提出 MQHOA-P 算法，算法的并行化框架如图 2 所示，其并行化流程的详细说明如下。

1) 初始化：在主节点完成参数  $k$ 、 $m$ 、 $S$ 、 $\sigma_0$ 、 $\sigma_{\min}$  的初始化运算，在采样范围内随机生成  $k$  个初始采样位置  $\{\overrightarrow{X}_0^1, \overrightarrow{X}_0^2, \dots, \overrightarrow{X}_0^k\}$ ，将当前的采样位置平均分为  $n$  份，分别发送到  $n$  个从节点。

2) 迭代运算：每个从节点按照主节点发送的采样位置和当前尺度进行迭代运算，将更新后的当前最优解位置发送到主节点。

3) 主节点将从节点发送来的  $k$  个采样位置进行汇总，计算其标准差  $\sigma_{p+1}$ ，与算法当前的尺度  $\sigma_s$  进行比较。如果  $\sigma_{p+1} > \sigma_s$ ，则说明当前采样中心点尚未在当前尺度收敛，返回步骤 2) 继续在从节点上进行迭代运算；如果  $\sigma_{p+1} \leq \sigma_s$ ，则算法已在当前尺度收敛，进行 M 收敛过程。

4) M 收敛：如果  $\sigma_{p+1} > \sigma_{\min}$ ，则通知从节点将当前尺度减半  $\sigma_{s+1} = 0.5\sigma_s$ ，返回步骤 2)；如果  $\sigma_{p+1} \leq \sigma_{\min}$ ，则说明当前的采样点的聚集程度已经满足算法计算精度的要求，结束算法的收敛过程。

5) 输出结果：主节点将当前  $k$  个采样位置中的最优位置和目标函数值作为结果输出。

#### 5 MQHOA-P 算法的并行化性能分析

MQHOA-P 算法的初始化、尺度判断、尺度收缩、精度判断和输出结果均在主节点运行，从节点仅执行

迭代运算，算法的收敛条件与串行程序完全相同。

MQHOA-P 算法的主从节点间的通信包含 3 个部分。

1) 算法初始化后主节点向从节点发起一次通信，将初始采样点和算法参数发送到从节点。

2) 从节点完成采样运算后向主节点发起一次通信，将更新后的最优解位置发送到主节点，共进行  $a_1$  次通信。

3) 主节点接收到从节点发送来的最优解位置后向从节点发起通信，将下一次迭代的尺度发送至从节点，共进行  $a_1 - 1$  次通信。

MQHOA-P 算法的运行时间为

$$T_1 = \frac{a_1 km D_2}{n} + a_1 C_3 + C + 2a_1 D_3 \quad (4)$$

通信时间占比为

$$\frac{B_3}{T_1} = \frac{2a_1 D_3}{\frac{a_1 km D_2}{n} + a_1 C_3 + C + 2a_1 D_3} \quad (5)$$

当算法迭代次数不断增长时，通信时间占比趋于常数，即  $\lim_{a_1 \rightarrow \infty} \frac{B_3}{T_1} = \frac{2D_3}{\frac{km D_2}{n} + C_3 + 2D_3}$  为常数。

MQHOA-P 算法的加速比为

$$S = \frac{T_0}{T_1} = \frac{a_1 km D_2 + a_1 C_3 + C}{\frac{a_1 km D_2}{n} + a_1 C_3 + C + 2a_1 D_3} \quad (6)$$

当从节点数增加时，加速度比增加，最大加速比为

$$\lim_{n \rightarrow \infty} S = \frac{a_1 km D_2 + a_1 C_3 + C}{a_1 C_3 + C + 2a_1 D_3} \quad (7)$$

由于迭代运算是 MQHOA-P 算法的主要运算量，当迭代次数  $a_1$  不断增长时，加速比不断增加并趋于稳定。

### 6 MQHOA-P 算法的实验验证

#### 6.1 实验环境

本文在由同构计算节点构成的并行计算集群中进行实验，所有节点的配置为 i3-3220(3.3 GHz) 的 CPU，4 GB 内存，操作系统为 CentOS 6.5，将 MPICH 3.1.3 部署在集群中构建并行计算环境。

#### 6.2 测试函数

本文采用 6 个典型的高维测试函数对 MQHOA-P 算法的并行性进行实验分析，测试函数的函数表达式、定义域、最优解位置和全局最小值如表 2 所示。

表 2 测试函数

函数表达式	定义域	最优解位置	全局最小值	单机执行时间/s
$f_1(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$-600 \leq x_i \leq 600, i = 1, \dots, n$	(0.000 000, 0.000 000)	0.000 000	351.965 333
$f_2(x) = \sum_{i=1}^n \frac{x_i^2}{40\,000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, n$	(0.000 000, 0.000 000)	0.000 000	6.427 333
$f_3(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} [(y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1))] + (y_n - 1)^2 (1 + 10 \sin^2(2\pi y_n))$ , $y_i = 1 + \frac{x_i - 1}{4}$	$-10 \leq x_i \leq 10, i = 1, \dots, n$	(-1.306 853, -1.424 845)	-176.137 578	0.523 667
$f_4(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	$-5 \leq x_i \leq 10, i = 1, \dots, n$	(0.000 000, 0.000 000)	0.000 000	0.400 667
$f_5(x) = \sum_{i=1}^n ix_i^2$	$-10 \leq x_i \leq 10, i = 1, \dots, n$	(0.000 000, 0.000 000)	0.000 000	0.355 000
$f_6(x) = \sum_{i=1}^n x_i^2$	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, n$	(0.000 000, 0.000 000)	0.000 000	0.350 000

为了测试 MQHOA-P 算法的加速比，首先使用 MQHOA 算法在单机环境中求解 6 个测试函数的全局最优解， $k = 80$ ， $m = 200$ ，函数维数为 6，每个测试函数重复 30 次实验计算单机平均执行时间，实验数据如表 2 的最右列所示。

### 6.3 MQHOA-P 算法加速比实验

#### 1) 计算节点数对 MQHOA-P 算法加速比的影响

并行算法包含可并行化和不可并行化 2 种成分的运算量，并行化后的 MQHOA-P 算法将可并行化部分分别发送到各个子节点运行，子节点仍需运行一部分不可并行化的运算量。

将计算节点数由 2 个增加到 10 个，使用 MQHOA-P 算法求解测试函数  $f_1 \sim f_6$  的全局最优解， $k = 80, m = 200$ ，每个测试函数重复 30 次实验计算平均运算时间，与表 2 中的单机执行时间进行比较，计算加速比，实验结果如表 3 所示。从表 3

可以看出随着计算节点数的增加，MQHOA-P 算法加速比首先出现增加趋势，然后逐渐趋于稳定。这个变化趋势与 MQHOA-P 算法的加速比表达式 (6) 相符合。由于目标函数  $f_1 \sim f_6$  的平均迭代次数不同，各目标函数加速比的稳定值存在一定的差异。

#### 2) 函数维数对 MQHOA-P 算法加速比的影响

随着测试函数维数的增长，函数解空间的数量呈指数级增长，算法迭代运算的次数也不断增长。在由 1 个主节点和 4 个计算节点构成的并行计算集群上使用 MQHOA-P 算法求解测试函数  $f_3 \sim f_6$  的全局最优解，函数维数由 3 维逐渐增加到 15 维，每个测试函数重复 30 次实验计算平均运算时间，求解算法加速比，实验数据如表 4 所示。从表 4 可以看出，随着函数维数的增长，算法加速比逐渐增长。这是因为随着函数维数的增长，算法迭代次数不断增加，算法可并行化部分的运算量比例增加，使算法加速比出现增长。

表 3 计算节点数对 MQHOA-P 算法加速比的影响

节点数	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
2	1.406 597 68	1.713 275	1.959 493	1.944 946	1.924 994	1.901 553
3	2.285 842 56	2.082 543	2.832 287	2.793 387	2.755 718	2.901 710
4	3.263 174 93	3.147 938	3.695 499	3.568 996	2.743 106	3.580 870
5	3.442 467 67	3.483 784	4.415 762	4.345 221	4.373 072	4.319 634
6	3.463 230 28	4.224 470	5.613 564 1	5.578 059	4.747 294	5.434 101
7	3.726 798 11	4.386 515	6.138 731 7	6.172 003	5.335 081	6.002 582
8	3.408 043 96	4.185 896	6.082 875 1	6.053 333	5.877 689	6.278 014
9	3.393 770 63	4.479 221	6.461 573 0	6.420 042	6.276 037	7.126 693
10	3.410 369 47	4.590 488	6.430 997 5	6.800 474	6.728 722	7.102 575

表 4 函数维数对 MQHOA-P 算法加速比的影响

维数	$f_3$	$f_4$	$f_5$	$f_6$
3	2.880 649	2.711 77	3.530 218	2.665 889
4	2.976 71	2.859 233	3.134 647	2.866 137
5	2.881 833	2.961 212	3.297 564	3.009 759
6	3.129 49	2.951 289	2.566 698	3.039 434
7	3.128 264	3.103 655	3.456 517	3.149 114
8	3.164 913	3.060 944	3.373 018	2.866 696
10	3.344 578	3.170 757	3.660 677	3.250 646
15	3.526 671	3.608 939	3.914 432	3.396 572

### 3) 采样参数 $m$ 对 MQHOA-P 算法加速比的影响

采样运算是算法最基本的运算单元，采样运算单元的运算量取决于参数  $m$ 。参数  $k$  依次取 80、100、150、200、250、300，参数  $m$  依次取 200、300、400、500、600，形成 30 个实验组合，在 5 个计算节点对 6 维函数  $f_5$  进行实验，测试算法加速比，每个组合重复 30 次实验，实验数据如表 5 所示。从表 5 可以看出，随着参数  $m$  的增长，算法加速比逐步增长，这反映出随着采样参数  $m$  的增长，采样运算单元的运算量相应地增长，算法可并行化部分的运算量比例增加，算法加速比增长。

表 5 参数  $m$  对 MQHOA-P 算法加速比的影响

$m$	$k=80$	$k=100$	$k=150$	$k=200$	$k=250$	$k=300$
200	3.450 087	3.065 583	2.792 84	3.026 471	3.051 19	3.394 296
300	3.442 412	3.621 03	3.667 513	3.198 545	3.192 759	3.448 752
400	3.508 801	3.943 25	3.809 55	3.591 404	3.392 792	3.771 928
500	3.421 67	4.062 604	3.777 593	3.746 629	3.535 984	3.379 141
600	3.945 592	4.080 465	4.115 259	3.793 128	3.669 308	3.510 004

## 7 结束语

本文通过对群体参数和采样参数进行实验分析确定 MQHOA 算法的并行粒度，提出并行化的 MQHOA-P 算法。通过在 10 个计算节点构成的集群上对 6 种标准测试函数进行实验，测试了计算节点数、函数维数和采样参数与算法加速比的关系，实验结果表明，MQHOA 算法具有很好的加速比和可扩展性，适用于求解高维函数优化问题。

### 参考文献:

[1] 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.

[2] ZABARAS N, YANG G Z. A functional optimization formulation and implementation of an inverse natural convection problem[J]. Computer Methods in Applied Mechanics and Engineering, 1997, 144(3): 245-274.

[3] PAPANITRIOU C H, STEIGLITZ K. Combinatorial optimization: algorithms and complexity[M]. Courier Corporation, 1998.

[4] LAI J Z C, HUANG T J, LIAW Y C. A fast k-means clustering algorithm using cluster center displacement[J]. Pattern Recognition, 2009, 42(11): 2551-2556.

[5] 王鹏, 黄焱. 多尺度量子谐振子优化算法物理模型[J]. 计算机研究与探索, 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 & Technology, 2015, 9(10): 1271-1280.

[6] 袁亚男, 王鹏, 刘峰. 多尺度量子谐振子算法性能分析[J]. 计算机应用, 2015, 35(6): 1600-1604.

YUAN Y N, WANG P, LIU F. Performance analysis of multi-scale quantum harmonic oscillator algorithm[J]. Journal of Computer Applications, 2015, 35(6): 1600-1604.

[7] 燕京京. 基于量子谐振子模型的聚类中心选取算法[D]. 成都: 中国科学院成都计算机应用研究所, 2015.

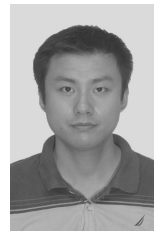
YAN J J. Clustering center selecting algorithm based on quantum harmonic oscillator model[D]. Chengdu: Chengdu Institute of Computer Application, 2015.

[8] 王鹏, 黄焱. 具有能级稳定过程的 MQHOA 优化算法[J]. 通信学报, 2016, 37(7): 79-86.

WANG P, HUANG Y. MQHOA algorithm with energy level stabilizing process[J]. Journal on Communications, 2016, 37(7): 79-86.

[9] GRAMA A, GUPTA A, KARYPIS G. Introduction to parallel computing: design and analysis of algorithms[M]. Redwood City, CA: Benjamin/Cummings Publishing Company, 1994.

### 作者简介:



黄焱 (1982-), 男, 江苏泗阳人, 博士, 淮阴师范学院讲师, 主要研究方向为智能算法、最优化理论等。

王鹏 (1975-), 男, 四川乐山人, 西南民族大学教授、博士生导师, 主要研究方向为智能算法、高性能计算等。

程琨 (1987-), 女, 山东新泰人, 中国科学院成都计算机应用研究所博士生, 主要研究方向为智能算法。

刘峰 (1987-), 男, 河南郑州人, 成都信息工程大学硕士生, 主要研究方向为智能算法。