
基于协方差矩阵的多尺度量子谐振子算法

陆志君¹, 王鹏², 安俊秀¹

¹ (成都信息工程大学 并行实验室, 成都 610225)

² (西南民族大学 计算机科学与技术学院, 成都 610225)
(bingjungg@126.com)

Multi-scale quantum harmonic oscillator algorithm based on Covariance Matrix

Lu Zhijun¹, Wang Peng², An Junxiu¹

¹(Parallel Laboratory, Chengdu University of Information and Technology, Chengdu 610225)

²(Computer Science and Technology, Southwest University for Nationalities, Chengdu 610225)

Abstract For global optimization problems, the multi-scale quantum harmonic oscillator algorithm (MQHOA) is a novel optimization algorithm based on the wave function of one-dimensional quantum harmonic oscillator. In addition to its simpleness on setting parameters and superior optimization speed, it can obtain the high accuracy solution for the low dimensional function. But for high dimension problems, MQHOA's scales can't be regulated dynamically which leads to convergence difficulties. In order to solve the problems from the process of quantum harmonic oscillator's convergence and multi-scale convergence, this paper is proposed to improve MQHOA. It's inspired by estimation of multivariate normal algorithm(EMNA) to improve the method of generating covariance matrix and iteration step is used to speed up update rate of the new covariance matrix. According to the core part of original MQHOA, a new sampling distribution function was constructed. It accelerates the convergence rate comparing with the original one, which performs more excellently in high dimension space. In addition, this paper also divided 18 evaluation functions into 3 groups to compare with 4 classic optimization algorithms in the 10 and 30 dimensions. Experimental results show the MQHOA based on covariance matrix is comparable, and even better than these 4 algorithms with regard to many functions.

Key word Quantum harmonic oscillator; global optimization; multivariate normal; covariance matrix; dynamic adjustment

摘要 针对全局优化问题, 多尺度量子谐振子算法(MQHOA)是一种基于一维量子谐振子波函数原理提出的新优化算法, 其参数设置简单, 收敛速度快, 对低维函数能够得到高精度的解, 但对于高维复杂问题来说由于算法尺度不能动态调节而难以收敛, 对此本文提出基于协方差矩阵的 MQHOA. 它从多元正态分布评估算法(EMNA)中获得启发, 改进协方差矩阵的生成方式, 增加迭代步长来加快新协方差矩阵的更新速度, 以此解决原 MQHOA 在量子谐振子收敛过程和多尺度收敛过程中存在的问题. 本文根据原 MQHOA 核心部分重新构造了采样点分布函数, 与原 MQHOA 相比该算法收敛速度加快, 并可以解决原 MQHOA 不能解决的高维复杂问题. 此外本文还组织了 18 个评价函数分成 3 组, 与其他 4 个经典优化算法在 10 维和 30 维作对比, 实验表明基于协方差矩阵的 MQHOA 算法与其他 4 种经典优化算法具有可比性, 甚至对很多函数来说要优于这 4 种算法.

关键词 量子谐振子; 全局优化; 多元正态分布; 协方差矩阵; 动态调节

收稿日期: 2016-06-06

基金项目: 国家自然科学基金项目(60702075); 国家社会科学基金项目(12XSH019)

This work was supported by the National Natural Science Foundation of China(60702075) and the Social Science Foundation of China (12XSH019).

通信作者: 安俊秀 (anjunxiu@cuit.edu.cn)

全局优化问题在工业设计、应用科学、分子生物学等应用科学中应用广泛,许多优化问题不能只靠数学分析解决,因而不得不通过数值优化算法解决.此外很多全局优化问题是不可分解的、有噪声数值的和基于仿真的,因此基于梯度的方法不能找出此类问题中的最小值,而对于全局优化问题的主要挑战是算法容易陷入局部最优点,并且随着维度的增大问题变得明显,因此有许多学者研究发现了许多不依赖目标函数梯度计算的可靠随机智能优化算法,这些算法以基于群体的优化算法为代表,具体分为进化算法和群智能算法两类.进化算法包括遗传算法(Genetic Algorithms, GAs)^[1],进化策略(Evolution Strategy, ES)^[2],进化规划(Evolutionary Programming, EP)^[3],差分进化(Differential Evolution, DE)^[4]等,基于智能粒子群的算法主要包括粒子群优化算法(Particle Swarm Optimization, PSO)^[5],人工蜂群算法(Artificial Bee Colony, ABC)^[6],蚁群算法(Ant Colony Optimization, ACO)^[7],蝙蝠算法(Bat Algorithm, BA)^[8]等.

近年来,基于量子算法的智能优化算法被提出^[9],即多尺度量子谐振子算法(Multi-scale Quantum Harmonic Oscillator Algorithm, MQHOA).该算法数学原理简单,设置方便,扩展性强.比如文献[10]用MQHOA来处理多峰优化问题,有很好的查全率和准确率.文献[11]把MQHOA与模拟退火算法(Simulated Annealing, SA)、量子退火算法(Quantum Annealing, QA)、量子粒子群算法(Quantum-behaved Particle Swarm Optimization, QPSO)作了对比,相比之下更具有稳定性和更好的计算精度.但在高维度下,相比于传统智能优化算法来说MQHOA对复杂函数不容易收敛,比如文献[9]中对Griewank函数在6维时已超过一万次迭代.为了解决高维复杂问题难以收敛的情况,本文主要工作在于改进原MQHOA算法性能.

本文第一节先介绍MQHOA算法核心原理,然后改造了EMNA协方差矩阵^[12,13]的生成方式,并在此基础上利用旋转矩阵、迭代步长等新概念重新组织MQHOA算法核心部分,解决了原算法在QHO收敛和M收敛存在的问题.第二节介绍了基于协方差矩阵的MQHOA算法步骤以及参数选取.第三节除了通过实验比较基于协方差矩阵的MQHOA算法与原算法性能区别,还组织3组共18个评价函数与选取的4个经典优化算法进行不同维度的对比,而这四个经典优化算法分别代表优化领域的两类,即群智能和进化策略.最后是总结和未来的研究方向.

1. 基于协方差矩阵的MQHOA模型

1.1 MQHOA算法

多尺度量子谐振子算法是近年提出的一种用于全局优化的算法,其近似模拟分子运动规律,利用微粒动态振动时在振子平衡位置达到势能最低点来解决全局优化问题.在理论中利用谐振子物理模型及量子谐振子波函数的概率解释构造算法核心,并利用量子隧道效应保证全局最优区域采样数目足够大以避免陷入局部最优位置.通过与其他算法进行对比,实验结果进一步说明了其具有更好的计算速度和计算精度^[9].

其核心是利用量子谐振子的薛定谔方程的解得到如下概率密度函数:

$$|\psi_n(x)|^2 = \frac{1}{2^n n!} \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{2}} e^{-\frac{\alpha^2 x^2}{2}} |H_n(\alpha x)|^2. \quad (1)$$

其中: ψ 表示波函数, m 表示粒子的广义质量, ω 表示粒子的振动频率, \hbar 为普朗克常量, H 为哈密顿算符, n 表示能级数.式(1)随着能级的减少量子谐振子从高能态到低能态逐渐收敛,直到收敛到基能态: $\psi_0(x) = \frac{\sqrt{\alpha}}{\pi^{1/4}} e^{-\alpha^2(x-x_0)^2/2}$,因此定义MQHOA算法在尺度 σ_s 下的波函数为 k 个以 k_i 为中心的高斯概率密度函数的迭加:

$$\psi_{QHO}(x) = \sum_{i=1}^k \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(x-k_i)^2}{2\sigma_s^2}}. \quad (2)$$

$\psi_{QHO}(x)$ 近似代表了目标函数以 k 个最优解为中心的目标函数在可行域上的概率分布,通过引入多尺度优化函数二进信息采样模型,把高斯函数作为二进小波尺度函数按照不同尺度和不同精度要求进行聚焦搜索,可以保证信息的不遗漏采样,从而获得全局最优解.

MQHOA算法的过程如下:

初始尺度向量 δ_s 的各维度值代表解空间在该维度上的跨度大小,并在解空间随机初始 k 个中心点,每个点根据多元正态分布以 δ_s 为标准差向量各生成 m 个采样点,从这 $k \times m$ 个采样点中选出最好的 k 个点并根据这些点计算新的标准差向量,如果新的标准差向量各维度上的值均比原 δ_s 相应维度上的值小,那么新的尺度向量 $\delta_s = \frac{1}{2}\delta_s$.重复上述过程直到 δ_s 各维上的最小值小于收敛精度为止.

文献[10]把上述收敛过程划分为两个过程,即在同一尺度向量 δ_s 下的量子谐振子收敛过程(Quantum Harmonic Oscillator, QHO收敛),和多尺度收敛过程(Multi-scale, M收敛).前者可以理解是横向收敛过程,后者是纵向收敛过程.QHO收敛过程实现对搜索空间的逐步收缩定位,M收敛过程实现对采样精度的逐步提高.

但研究发现, MQHOA虽然在低维下收敛速度快, 结果精确, 但当维度升高(大于10维)后就很难收敛. 其原因在于QHO收敛过程中随机迭代算法所固有的“无记忆性”, 不能将迭代过程中的优良个体有效保留, 另一方面M收敛过程直接采用尺度减半的固定方式会导致在某些重要维度上搜索的细节缺失, 而这些重要维度对函数值的影响至关重要, 不应该与其他维度上的搜索同等对待. 下面, 本文将根据协方差矩阵分别对这两个过程加以改进以增强算法性能.

2.2 基于协方差的MQHOA原理

为了使描述便于理解, 先给出以下概念:

如果随机向量 $\mathbf{X} = (X_1, \dots, X_n)'$ 的概率分布密度函数有如下形式: $f(x_1, \dots, x_n) = f(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |\mathbf{C}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})' \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m}) \right\}$. 其中 $\mathbf{x} = (x_1, \dots, x_n)'$, \mathbf{C} 为 $n \times n$ 正定矩阵, $\mathbf{m} = (m_1, \dots, m_n)'$ 为常向量, 则称 \mathbf{X} 服从 n 元正态分布, 记作 $\mathbf{X} \sim N_n(\mathbf{m}, \mathbf{C})$. \mathbf{X} 的均值向量 $E\mathbf{X} = \mathbf{m}$, 协方差矩阵 $\text{cov}(\mathbf{X}) = \mathbf{C}$. 对于任意常数 $c \geq 0$, \mathbf{X} 的概率分布密度在曲面 $(\mathbf{x} - \mathbf{m})' \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m}) = c$ 上, 称为 \mathbf{X} 的等密度曲面, 其在 R^n 空间为以 μ 为中心的超椭圆.

根据线性代数正定矩阵概念可以推出如下定理:

定理1 设 \mathbf{C}_s 是来自正态总体 $N_n(\mathbf{m}, \mathbf{C})$ 的样本协方差矩阵, 样本数为 λ , \mathbf{C}_s 为正定矩阵的充要条件为 $\lambda > n$.

文中默认所有的样本协方差矩阵都满足定理1. 既然 \mathbf{C} 正定, 可以通过矩阵正交分解得到规范正交基 $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$, 特征向量记作 \mathbf{D}^2 , 其对应的特征值 $d_1^2, \dots, d_n^2 > 0$, 其正交分解可表示成:

$$\mathbf{C} = \mathbf{B} \mathbf{D}^2 \mathbf{B}^T. \quad (3)$$

同样我们定义 \mathbf{C} 的均方根

$$\mathbf{C}^{\frac{1}{2}} = \mathbf{B} \mathbf{D} \mathbf{B}^T = \mathbf{B} \text{diag}(d_1, \dots, d_n) \mathbf{B}^T, \quad d_1, \dots, d_n > 0. \quad (4)$$

也很容易推出

$$\mathbf{C}^{-\frac{1}{2}} = \mathbf{B} \mathbf{D}^{-1} \mathbf{B}^T = \mathbf{B} \text{diag} \left(\frac{1}{d_1}, \dots, \frac{1}{d_n} \right) \mathbf{B}^T, \quad d_1, \dots, d_n > 0. \quad (5)$$

多元正态分布 $N(\mathbf{m}, \mathbf{C})$ 可以写成如下形式:

$$\begin{aligned} N(\mathbf{m}, \mathbf{C}) &\sim \mathbf{m} + N(0, \mathbf{C}) \\ &\sim \mathbf{m} + \mathbf{C}^{\frac{1}{2}} N(0, \mathbf{I}) \\ &\sim \mathbf{m} + \mathbf{B} \mathbf{D} \mathbf{B}^T N(0, \mathbf{I}) \\ &\sim \mathbf{m} + \mathbf{B} \mathbf{D} N(0, \mathbf{B}^T \mathbf{I} \mathbf{B}) \\ &\sim \mathbf{m} + \mathbf{B} N(0, \mathbf{D}^2). \end{aligned} \quad (6)$$

其中 \mathbf{I} 为单位矩阵. 式(6)的几何意义描述了等密度超椭圆面的形态, 其中心点为 \mathbf{m} , \mathbf{D} 表示这个超椭圆面的轴长, 对于二元正态分布对应椭圆长

轴和短轴, \mathbf{B} 表示超椭圆面的旋转矩阵. 对于二元正态分布对应椭圆逆时针旋转大小, 但对于高维超椭圆面旋转变得更加复杂, 它们不再由一个角度来刻画, 并可能影响多个平面子空间.

2.2.1 改进的QHO过程 原QHO过程的采样搜索点根据多元正态分布生成, λ 个采样搜索点均满足:

$$\begin{aligned} \mathbf{x}_k &\sim N(\mathbf{m}, \mathbf{D}^2) \\ &\sim \mathbf{m} + N(0, \mathbf{D}^2), k = 1, \dots, \lambda. \end{aligned} \quad (7)$$

其中 \mathbf{D} 是对角矩阵, $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, $\mathbf{m} = (m_1, m_2, \dots, m_n)$ 为样本均值向量, n 为采样搜索点的维度, σ_i 为标准差, m_i 为各维度均值, $i = 1, \dots, n$. 当 \mathbf{D} 对角线上值相等时, 其等密度曲面为高维球面, 当不相等时, 其等密度曲面为高维椭圆球面.

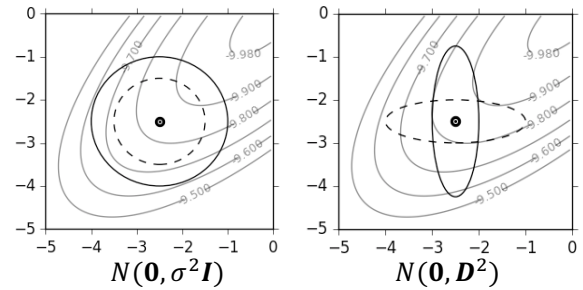


图1 4种不同高斯分布等密度曲线
Fig.1 Lines of equal density of four different normal distributions

图1灰色曲线为某二维椭圆面 $z = f_{\text{ell}}(x, y)$ 在一定范围内的等高线, 左图中的圆为等密度曲面投影, 其中 \mathbf{I} 为单位矩阵, 黑点为均值坐标, 其采样点沿黑点向周围均匀扩散. 右图的两条椭圆曲线分别是不同等密度曲面投影, 两者的 \mathbf{D} 矩阵对角线上值均相异, 其采样点沿坐标轴分布. 原MQHOA初始化 \mathbf{D} 后, QHO收敛过程依赖于 \mathbf{D} , 在没有到达QHO收敛条件的情况下 \mathbf{D} 始终不变, 在各维度上没有动态尺度变化.

多元相关性的分布估计算法(Estimation of distribution algorithms, EDA)在连续域内有一种类似QHO收敛过程的EMNA(Estimation of multivariate normal algorithm)算法, 其同样采用了多元正态模型来表示采样搜索点的概率分布, 在进化过程中每一代采用最大似然估计方法, 估计多元正态分布的均值向量和协方差矩阵^[12-13]. 其协方差矩阵可以表示如下:

$$\mathbf{C}_{\text{EMNA}_{\text{global}}} = \frac{1}{u} \sum_{i=1}^u (\mathbf{x}_{i:\lambda} - \mathbf{m})(\mathbf{x}_{i:\lambda} - \mathbf{m})^T. \quad (8)$$

其中 u 表示从 λ 个采样搜索点中选取的适应度值最好的点个数, $\mathbf{x}_{i:\lambda}$ 表示从 λ 个采样搜索点中选取的适应度值排在第 i 位的点, $\mathbf{m} = \frac{1}{u} \sum_{i=1}^u \mathbf{x}_{i:\lambda}$ 为均值向量. $\mathbf{C}_{\text{EMNA}_{\text{global}}}$ 是对变量总体的一致性估计, 文献[14]采用协方差矩阵 $\frac{\mu}{\mu-1} \mathbf{C}_{\text{EMNA}_{\text{global}}}$ 增加了对 \mathbf{C} 的无偏估计, 除了权重略微不同外在各个维度上

的分布完全一致. 利用 EMNA 在二维线性函数 $f_l = -\sum_{i=1}^2 x_i$ 上计算最小值, 演化过程如图 2 所示. 图 2 中的散点为采样点, 采样点个数 $\lambda=100$, 选择评估点数 $\mu = 30$, ‘+’ 为采样点均值向量坐标, 曲线为协方差分布等密度曲面在 $f_l = 0$ 处的投影. 左图为初始化采样点状态图, 此时 $\mathbf{C}_{EMNA_{global}} = \mathbf{I}$. 中图通过对这 100 个采样点对应的评价函数值排序, 选择

最好的 30 个点来评估下一轮的 $\mathbf{C}_{EMNA_{global}}$ 和均值向量 \mathbf{m} . 右图以评估产生的 \mathbf{m} 与 $\mathbf{C}_{EMNA_{global}}$, 利用式(8)生成新的 100 个采样点, 依次循环直到满足收敛条件. 虽然迭代过程中采样均值点不断靠近最小点, 但从图中可见在梯度方向上采样均值中心点的移动步长不断减

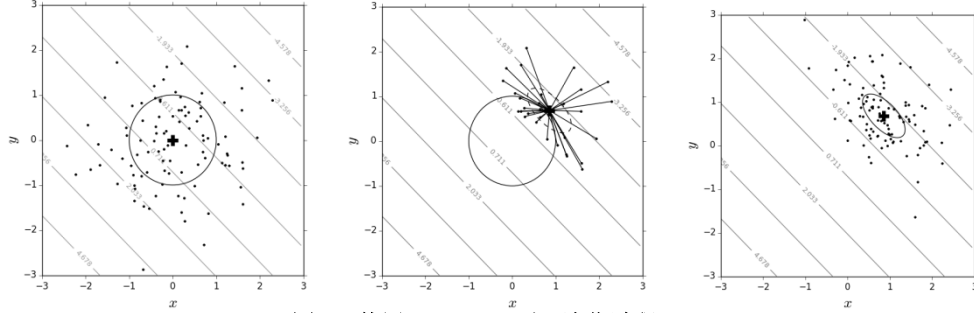


图 2. 使用 $\mathbf{C}_{EMNA_{global}}$ 对 f_l 演化过程

Fig.2 evolution process of f_l by using $\mathbf{C}_{EMNA_{global}}$

小, 这容易导致其变得过早收敛. 对此本文提出新的协方差矩阵 \mathbf{C}_{qho} 以及均值向量的生成模型:

$$\mathbf{C}_{qho} = \sum_{i=1}^{\mu} \omega_i (\mathbf{x}_{i:\lambda} - \mathbf{m}_{old})(\mathbf{x}_{i:\lambda} - \mathbf{m}_{old})^T. \quad (9)$$

$$\mathbf{m} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}. \quad (10)$$

$$\sum_{i=1}^{\mu} \omega_i = 1, \omega_1 \geq \omega_2 \geq \dots \geq \omega_{\mu} > 0. \quad (11)$$

比较式(8)与式(9), 其最大不同点在于均值向量的选择, 式(8)估计协方差使用当前所选样本点的均值向量, 式(9)使用了上一代种群样本的均值向量, 这就解决了原 QHO 过程的“无记忆性”问题,

这种选择还大大增加了在梯度方向上的方差值. 此外如式(10) 式(11)所示, 均值向量和协方差矩阵的计算方法多了加权方式, 增加了种群差异性, 比如可以选择典型的简单排序编码法设置权重为 $\omega_i \propto u - i + 1, i = 1 \dots \mu$. 特别当 $\omega_i = \frac{1}{\mu}$ 时, 式(8)与式(9)除了均值向量选择的不同外其他都相同. 为了与 EMNA 作对比, 取 $\omega_i = \frac{1}{\mu}, i = 1 \dots \mu$, 重复图 2 算法过程:

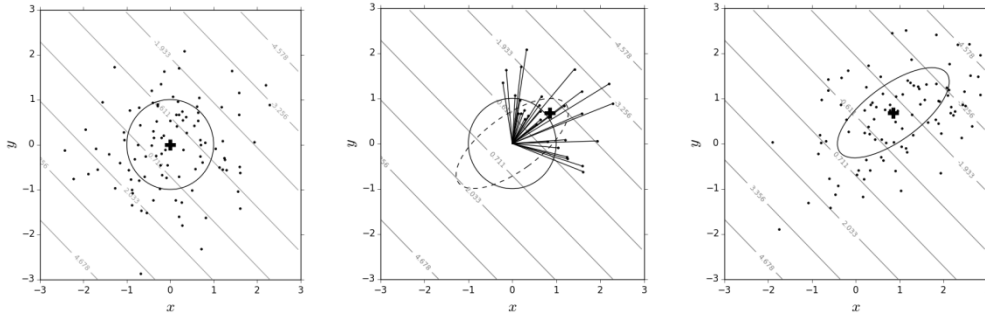


图 3. 使用 \mathbf{C}_{qho} 对 f_l 演化过程

Fig.3 evolution process of f_l by using $\mathbf{C}_{EMNA_{global}}$

图 3 中左图与图 2 一致, 中图以原始均值向量构造协方差矩阵, 椭圆曲线即等密度曲面在 $f_l = 0$ 处的投影, 右图重新以最新的均值向量构造协方差矩阵, 相当于把中图椭圆的中心点平移到最新的均值向量中去. 从图中可见在梯度方向上的方差增加了, 更容易找到最好点.

2.2.2 改进的 M 收敛过程 由于原 M 收敛过程是固定衰减, 没有动态衰减机制, 容易丢失重要维度的采样信息, 因此这里增加迭代步长:

$$p = \|\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T(\mathbf{m} - \mathbf{m}_{old})\|. \quad (12)$$

式中 \mathbf{B} 是 \mathbf{C} 的正交矩阵, 根据前面描述的几何意义可知, \mathbf{B}^T 把向量空间变换到坐标轴上, \mathbf{D}^{-1} 使坐标轴变化尺度相同, \mathbf{B} 重新把向量从坐标轴变换到其原本的向量空间中去, 确保差向量的向量空间与原多元正态分布的超椭圆等密度曲面的向量空间一致. 此外, $\mathbf{m} - \mathbf{m}_{old}$ 越小 p 越小, 也就是下次迭代在梯度方向上的步长越小, 同样 $\mathbf{m} - \mathbf{m}_{old}$ 越大 p 越大, 下次迭代在梯度方向上的步长越大. 这样当前两代的均值中心点接近时会进一步进行精确搜索, 而当两代均值中心点较远时, 相比上一代搜索范围将扩大, 这样也可以减少迭代次

数并加快算法收敛. 但快速收敛也就会提高算法的不稳定性, 因此改进算法需要提供较多采样点以保证算法稳定性. 根据式(6), 式(12)也可写成

$$p = \left\| C^{-\frac{1}{2}}(\mathbf{m} - \mathbf{m}_{old}) \right\|. \quad (13)$$

综合以上公式, 可以总结出基于协方差矩阵的 MQHOA 算法的采样点分布:

$$\begin{aligned} \mathbf{x}_k &\sim \mathbf{m} + pN(\mathbf{0}, \mathbf{C}_{qho}) \\ &\sim \mathbf{m} + \left\| C^{-\frac{1}{2}}(\mathbf{m} - \mathbf{m}_{old}) \right\| N(\mathbf{0}, \mathbf{C}_{qho}) \\ &\sim \mathbf{m} + \left\| \mathbf{B} \mathbf{D}^{-1} \mathbf{B}^T (\mathbf{m} - \mathbf{m}_{old}) \right\| \mathbf{B} N(\mathbf{0}, \mathbf{D}^2) \\ k &= 1, \dots, \lambda. \end{aligned} \quad (14)$$

3 改进 MQHOA 算法描述

根据 2.2 算法改进描述, 改进的 MQHOA 整个算法流程可以描述如下:

- Step 1** 输入样本数 λ , 评估点比例 α , 评估点数 $\mu = \alpha\lambda$, 以及对应的权重向量 $\omega = [\omega_1, \dots, \omega_u]$, 初始样本均值向量 \mathbf{m} , 收敛精度 σ_{crit} , 初始迭代步长 p , 初始标准差向量 σ , 由 σ 构成初始方差矩阵 \mathbf{C} 的对角线.
- Step 2** 依据式(14)生成 λ 个样本点.
- Step 3** 根据这些点的评价函数值选取 μ 个最好的点作为评估点, 依据式(9) 式(10)产生新的协方差矩阵 \mathbf{C} 和均值向量 \mathbf{m} , 依据式(12)产生新的迭代步长 p .
- Step 4** 对于 $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, 如果 $\min_{1 \leq i \leq n} d_i > \sigma_{crit}$, 返回 Step2.

Step 5 对最终 λ 个采样点对应的评价值排序, 取最好的一个点作为最终解.

一般设置初始迭代步长 $p = 1$, $\sigma_{crit} = 1E - 6$, $\sigma = \mathbf{MAX} - \mathbf{MIN}$, $[\mathbf{MIN}, \mathbf{MAX}]$ 为目标函数定义域边界范围, ω 满足式(10), 在实际工程应用中设置 $\omega_i \propto \ln \frac{u+1}{i}$, $i = 1 \dots \mu$ 是较好的选择, 但相对来说以上参数并不是关键参数, 这些参数相对较自由且不会妨碍算法收敛. 选择 $\lambda > n$, 为了保证算法鲁棒性并且能在多峰函数中有更强的搜索能力, 可以增大 λ 的值, 同时对于单峰等简单问题可以适当减少 λ 的值或者置 $\alpha = \frac{1}{4}$, 这样将使算法快速收敛.

4 实验

4.1 评测函数与参数设置

15 个函数被用来测试改进 MQHOA 算法性能. 其中 $f_1 \sim f_4$ 为单峰函数, 一般来说, 当算法在 Sphere 函数上容易收敛时同样在 Sum Square 函数和 Zakharov 函数上也较容易收敛. $f_5 \sim f_{11}$ 为 7 个多峰函数, 而 $f_{12} \sim f_{18}$ 为这七个多峰函数的旋转版本, 即 $f(\mathbf{y}) = f(\mathbf{M}\mathbf{x})$. 其中 \mathbf{M} 为旋转矩阵, 只要 \mathbf{x} 中某一维度上的值发生变化, \mathbf{y} 所有维度上的值也将变化, 避免 $f_5 \sim f_{11}$ 中一些函数的自变量过于独立而可以通过单独优化的方式找到最优解 (如 Griewank 函数自变量相互独立, n 维 Griewank 函数只需要 n 次一维优化即可). 本文中的旋转矩阵 \mathbf{M} 根据 Salomon 的方法产生^[18], 所有函数最优值以及搜索范围等都在表 1 列出并分组, 其中 \mathbf{x}^* 代表理论最优解.

表 1 评测函数

Table 1 Benchmark Functions

F	$f(\mathbf{x}^*)$	搜索范围	函数名称	函数分组
f_1	0	$[-100, 100]^n$	Sphere ^[5]	A
f_2	0	$[-100, 100]^n$	Quadric ^[5]	A
f_3	0	$[-10, 10]^n$	Sum Square ^[15]	A
f_4	0	$[-10, 10]^n$	Zakharov ^[15]	A
f_5	0	$[-2.048, 2.048]^n$	Rosenbrock ^[5]	B
f_6	0	$[-32.768, 32.768]^n$	Ackley ^[5]	B
f_7	0	$[-600, 600]^n$	Griewank ^[5]	B
f_8	0	$[-0.5, 0.5]^n$	Weierstrass ^[16]	B
f_9	0	$[-5.12, 5.12]^n$	Rastrigin ^[5]	B
f_{10}	0	$[-10, 10]^n$	Levy ^[17]	B
f_{11}	0	$[-500, 500]^n$	Schwefel ^[5]	B
f_{12}	0	$[-2.048, 2.048]^n$	Roated Rosenbrock ^[16]	C
f_{13}	0	$[-32.768, 32.768]^n$	Roated Ackley ^[16]	C
f_{14}	0	$[-600, 600]^n$	Rotated Griewank ^[16]	C
f_{15}	0	$[-0.5, 0.5]^n$	Rotated Weierstrass ^[16]	C
f_{16}	0	$[-5.12, 5.12]^n$	Rotated Rastrigin ^[16]	C
f_{17}	0	$[-10, 10]^n$	Rotated Levy ^[17]	C
f_{18}	0	$[-500, 500]^n$	Rotated Schwefel ^[16]	C

影响智能算法测试结果的往往是采样点数, 不考虑时间的情况下, 一般采样点数越大结果越

好. 对基于协方差矩阵的 MQHOA 算法, 当采样点数固定时影响算法性能的主要参数是评估点比例

α , α 较小时评估样本点差异性少, 不能反映种群特征. 当 α 较大时种群与上一代差异性减少, 容易早熟而局部收敛. 在样本数为 100 的情况下通过 50 次独立重复实验测试 α 值对算法性能的影响, 经过比较实验, 采用两个对 α 值最敏感的函数进行测试, 即 30 维的 Griewank 函数和 Rastrigin 函数, 计算出的最小值误差在 0.01 内视为命中, 测试结果如图 3. 通过比较发现, μ 取 0.3~0.35 命中概率较大,

考虑到 α 越大迭代次数越小, 决定算法消耗总时间的主要是算法迭代次数, 因此实验设置 $\alpha = 0.35$.

除此之外, 实验中的所有算法均以 python 实现, 版本为 3.4.5, 实验主机为双核 Celeron2.8GHZ, 4 GB RAM, Windows 10. 为减少实验统计数据错误率, 每个函数重复 30 次测试.

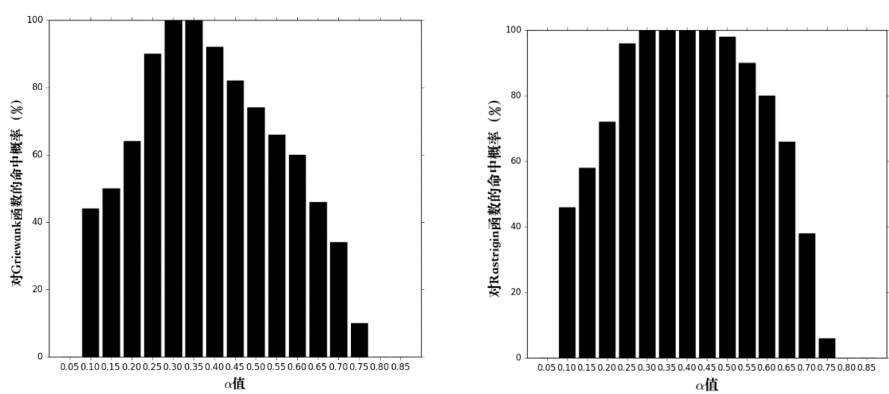


图 3. μ 对算法性能影响
Fig.3 μ 's effect on algorithm's performance

4.2 基于协方差矩阵的 MQHOA 算法与原算法性能比较

基于协方差矩阵的 MQHOA 算法目的是克服原算法高维难以收敛或者收敛缓慢问题, 这一节将两算法进行对比, 根据文献[10]中六个单峰和多峰函数来评估各算法迭代次数. 算法参数设置与文献[10]一致, 其定义域边界范围为 $[-10, 10]$, 样本数 $\lambda = 200$, $\sigma_{crit} = 1E - 6$, 初始迭代步长 $p = 1$, 初始中心点在定义域内随机选择. 重复 10 次最后汇总算法迭代次数并取平均值记录在表 1, 其中空白记录表示数值大于 20000, 也就是需要花费非常长的迭代时间. 表中记录行上面数值是原 MQHOA 算法迭代次数, 下面数值为基于协方差矩阵的 MQHOA 算法迭代次数.

表 1 基于协方差矩阵的 MQHOA 与原算法迭代次数比较
Table 1 Comparison improved MQHOA's iteration with the original one

函数名称	函数维度							
	3	4	5	6	7	8	10	15
Griewank	68.7	550.8	5150.9	15828.5				
	28.4	39.6	44.8	60.1	75.5	101.9	356.2	2132.1
Rastrigin	15	41.1	126	2466.6	3234.7	16928.3		
	24.6	27.0	35.7	47.2	56.7	98.4	282.2	1987.6
Levy	11	15.1	20.7	25.0	33.9	47.7	126.2	
	20.1	25.1	31.5	37.1	46.8	55.7	80.4	208.4
Zakharov	11.7	18.2	25.2	31.8	47.7	71.9	205.3	
	18.1	22.4	27.5	32.1	41.2	46.0	70.5	163.4
Sum Square	11.1	16.8	22.7	39.5	80.3	192.8	3398.3	
	18.5	22.4	26.5	24.1	38.4	46.7	72.41	169.4
Sphere	10.6	14.5	18.9	22.6	28.5	35	63.1	1101.6

22.1	27.8	36.3	43.5	56.9	65.2	70.4	398.4
------	------	------	------	------	------	------	-------

从表 1 可见, 基于协方差矩阵的 MQHOA 在性能上要远超原 MQHOA, 尤其是在高维部分和复杂函数部分, 而原 MQHOA 虽然在低维简单函数能够取得暂时的优势, 但算法总体而言随着维度增长总迭代次数也呈几何增长, 尤其对复杂多峰函数在维度较高情况下算法很难收敛. 对此特用 Griewank 函数的比较 对这两个算法单独分析, 如下图 4 所示.

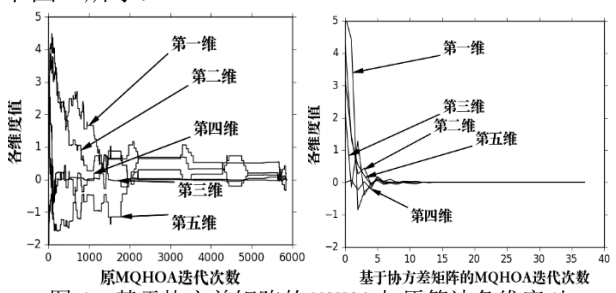


图 4. 基于协方差矩阵的 MQHOA 与原算法各维度对 Griewank 函数收敛情况

Fig.4 MQHOA based on covariance matrix and the original's convergence situation about all dimension of Griewank

左图中, 原 MQHOA 由于各维尺度不能动态调节, 在未达到 QHO 收敛条件的情况下, 尺度固定不变, 故收敛情况呈阶梯走势. 当算法搜索到最优解附近时 (左图中迭代次数约为 5700), 也因为尺度较大而使下次迭代过程中错过最优解, 而当算法搜索到最优解附近且尺度也很小时, 收敛

速度非常快（左图最右端所示）。右图中，函数各维度平滑收敛，在第 10 次迭代过程中基本锁定了最优解。通过对比可知，基于协方差矩阵的 MQHOA 算法解决了原算法尺度不能动态调节问题，并且加快了收敛速度。

4.3 与其他算法比较

我们用表 1 中的 18 个评测函数测试基于协方差矩阵的 MQHOA 算法性能，为更好评估此算法，实验选择一些经典的优化算法作为对比，这些算法分为进化和群智能两方面，有粒子群优化算法 (Particle Swarm Optimization, PSO)、快速进化策略算法 (Fast Evolution Strategies, FES)、人工蜂群算法 (Artificial Bee Colony, ABC)、差分进化算法 (Differential Evolution, DE)。这些算法对非线性多维度空间有很好的搜索能力，广泛应用于各种研究领域，在实际工程问题中也有很多应用^[19]。为使以上各算法在问题域内发挥较好的搜索能力，除了粒子种群数目和适应度函数迭代次数 (Number of Fitness Function Evaluations, NFFEs) 自行设置外，其他参数按照经典参数设置如下：

PSO：认知参数和社会参数，即文献[5]中的 ϕ_1 和 ϕ_2 ，是可以用来改变个体和群体之间权重的

向量，设置这两个参数均为 1.8。惯性权重是用来决定上一代粒子速率对下一代粒子速率的影响，以上均根据文献[5]设置。

FES：FES 算法以柯西变异算子代替经典进化策略算法中的高斯变异算子，与原算法相比有更快的收敛速度，其柯西分布尺度参数初始为 0.5，根据文献[20]设置。

DE：在 DE 中，F 是影响两个解之间的差分变化的实数常量，设置为 0.6。交叉率是用来控制种群的多样性，设置为 0.9，以上均根据文献[21]设置。

ABC：ABC 中除了食物源数量 SN，还有个参数叫食物源限制 Limit，当采到的食物源数量超过 Limit 部分就会被丢弃，采用文献[21]中的设置。

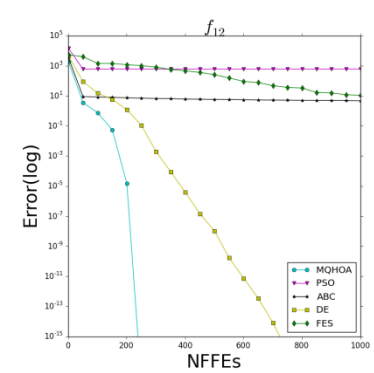
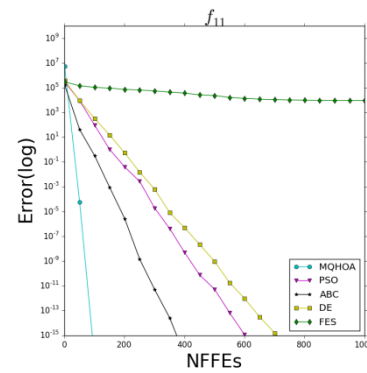
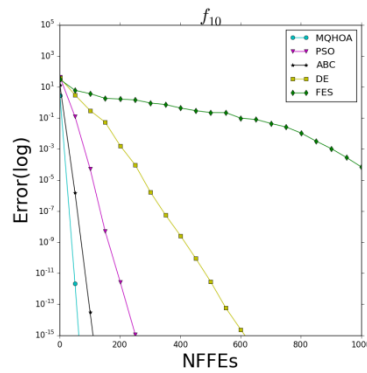
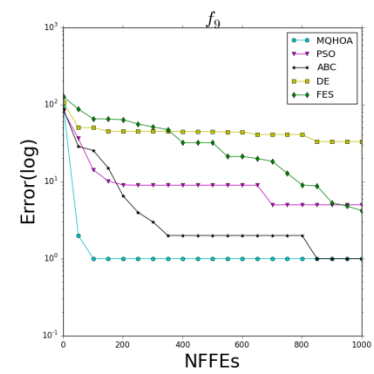
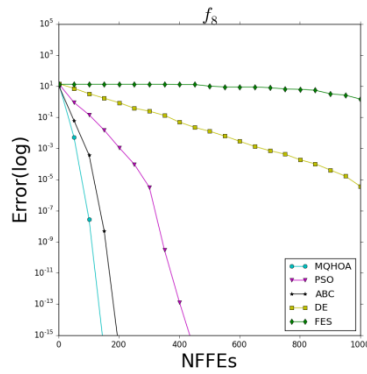
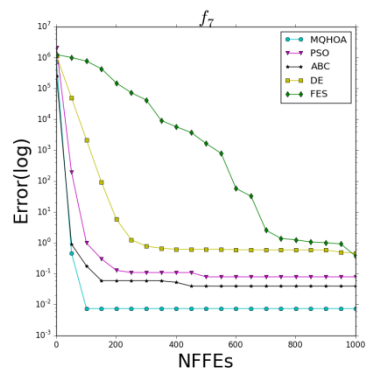
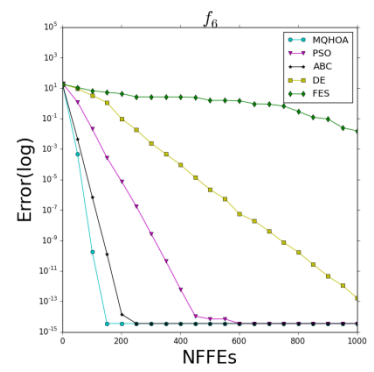
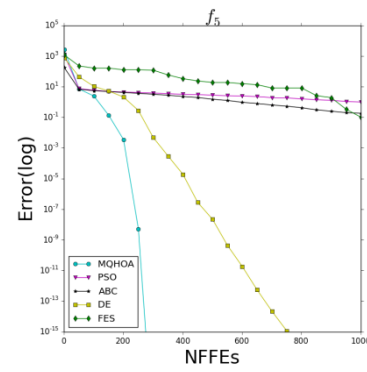
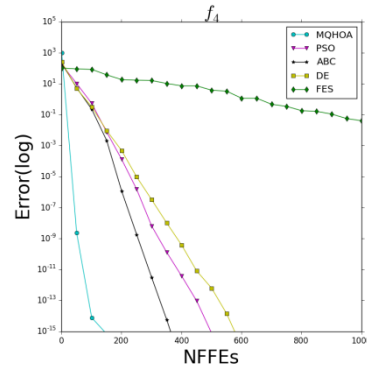
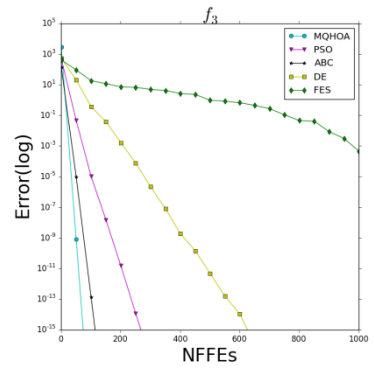
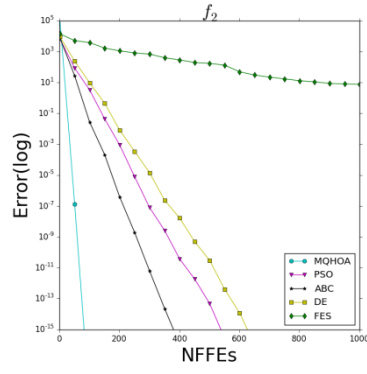
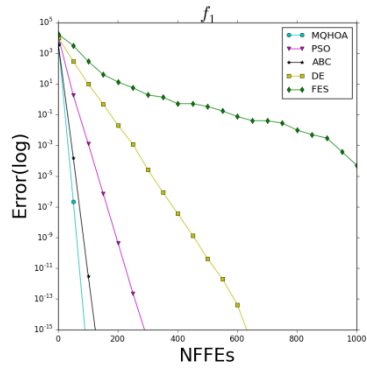
整体来说初始种群数目越大算法收敛越快，实验设置种群数目为 120，对于 10 维问题设置 NFFEs 为 1000，对于 30 维问题设置 NFFEs 为 5000。为使实验数据稳定可靠，经过 30 次独立重复实验后统计最优解对应的适应度均值和标准差记录在表 2 与表 3，最好的值用粗体标出。此外使用自由度为 29 的 t 双边检验来说明结果的有效性，显著性水平设为 0.05，用[†] 标记统计有效性。

表 2 10 维函数比较结果
Table 2 Compare results for 10-D problems

F	MQHOA	PSO	FES	ABC	DE
f_1	2.51E-154±1.19E-155[†]	5.12E-28±3.48E-29 [†]	7.45E-05±2.61E-05	4.62E-149±4.29E-150 [†]	3.19E-25±2.39E-25
f_2	1.42E-152±4.84E-154[†]	1.32E-28±1.18E-29 [†]	1.04E+01±4.21E+00 [†]	9.32E-47±1.59E-48 [†]	2.06E-25±9.23E-26 [†]
f_3	8.47E-156±4.74E-157[†]	5.08E-29±4.08E-29	3.35E-04±1.16E-04	1.26E-149±2.48E-149	5.53E-27±3.92E-27
f_4	2.72E-32±5.03E-32	7.45E-29±6.03E-30 [†]	5.88E-02±2.20E-02	1.85E-51±1.93E-51[†]	2.15E-27±1.09E-28 [†]
f_5	3.00E-32±4.15E-33[†]	8.62E-01±1.60E-02 [†]	8.27E-01±1.13E+00 [†]	2.09E-01±6.05E-02 [†]	1.92E-22±2.57E-23 [†]
f_6	3.55E-15±2.25E-16[†]	3.55E-15±1.21E-15 [†]	1.44E-02±1.32E-03 [†]	8.53E-14±1.04E-15 [†]	4.38E-13±3.50E-13 [†]
f_7	5.90E-03±1.18E-03	5.07E-02±2.45E-02	6.61E+03±7.69E+03	9.28E-02±7.63E-02	3.17E-01±1.32E-01
f_8	0.0±0.0[†]	0.0±0.0[†]	1.53E+00±1.86E-01 [†]	0.0±0.0[†]	9.65E-06±8.38E-06
f_9	1.01E+00±1.55E+00	1.79E+00±1.16E+00	5.58E+00±8.75E-01 [†]	9.95E-01±8.90E-01[†]	2.96E+01±4.78E+00 [†]
f_{10}	1.79E-32±3.58E-33[†]	6.45E-29±1.09E-28	5.45E-05±2.01E-05 [†]	2.41E+01±4.81E+01	1.31E-26±8.71E-27 [†]
f_{11}	4.56E-32±1.12E-32[†]	1.81E-28±1.78E-28 [†]	1.40E+04±7.81E+03 [†]	2.03E-06±4.06E-06	1.82E-24±5.81E-25 [†]
f_{12}	4.64E-27±2.94E-28[†]	1.90E+01±2.88E+01	9.72E+00±1.34E+00 [†]	5.49E+00±2.08E+00 [†]	4.28E-23±4.16E-23 [†]
f_{13}	3.55E-15±1.02E-15[†]	1.61E+01±8.06E+00 [†]	7.92E+00±5.88E+00	1.72E+00±7.08E-01 [†]	4.70E+00±7.05E+00
f_{14}	1.23E-02±8.67E-03[†]	2.05E-01±9.55E-02 [†]	1.79E+01±1.29E+00 [†]	2.30E-01±2.47E-01 [†]	4.23E-01±9.77E-02 [†]
f_{15}	2.36E-01±1.79E-01[†]	6.00E+00±1.65E+00 [†]	4.40E+00±4.08E-01 [†]	3.13E+00±4.04E+00	2.04E-01±1.05E-01 [†]
f_{16}	1.21E-01±1.35E-01[†]	3.03E+01±2.33E+01 [†]	2.05E+01±2.07E+00 [†]	3.35E+01±3.13E+01 [†]	2.82E+00±2.68E+00
f_{17}	1.09E-01±1.76E-01 [†]	8.63E+00±1.09E+01 [†]	3.66E+00±1.21E+00	2.53E-01±1.67E-01 [†]	3.01E-02±5.26E-03[†]
f_{18}	3.31E-32±2.91E-32[†]	7.82E+02±1.04E+02 [†]	5.17E+04±8.26E+03 [†]	3.82E-05±5.38E-05	1.22E-23±6.50E-24 [†]

从上表可见 A 组函数中除 f_4 外基于协方差矩阵的 MQHOA 性能明显好于其他 4 个算法，而 f_4 的结果与 ABC 相差也不多，B 组和 C 组函数中也显示 MQHOA 算法性能整体好于其他算法，除了 MQHOA 外 A、B

组中表现最好的是 ABC 算法，而 C 组中表现最好的算法是 DE。为更清楚描述算法收敛过程，绘制 10 维问题下算法收敛过程比较图，见图 5。



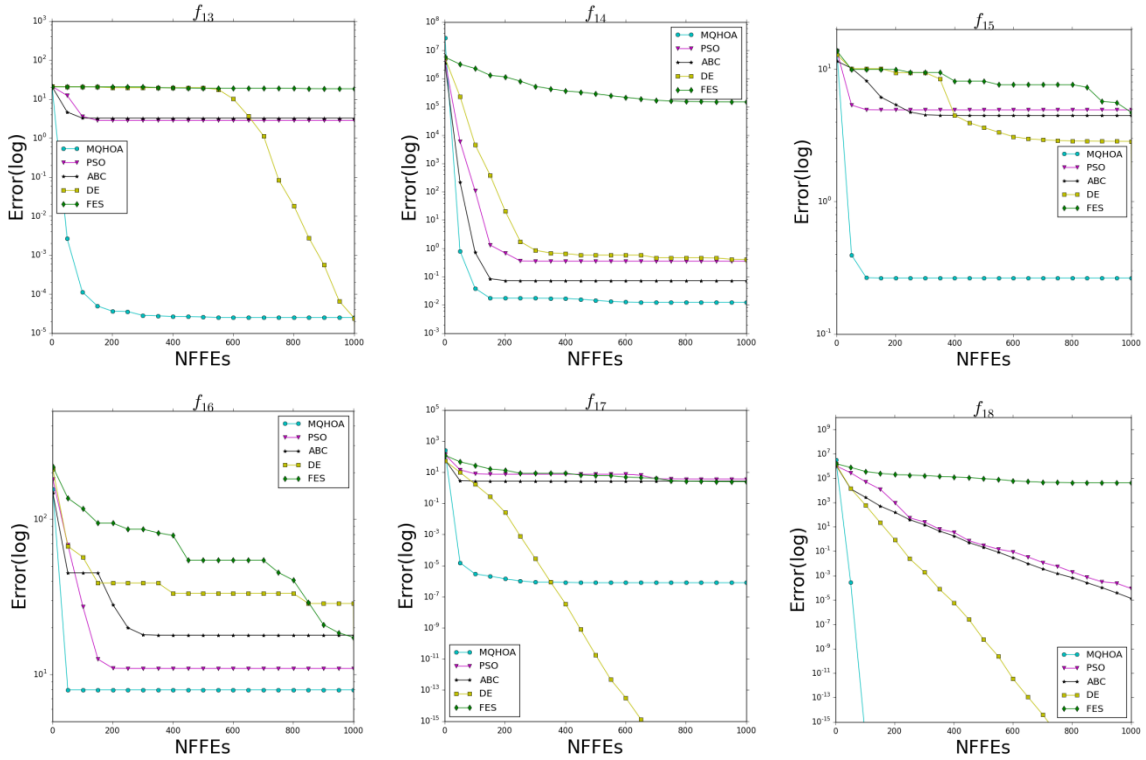


图 5. 5 种算法对 18 个函数测试的收敛情况
Fig.5. Convergence performance of the 5 algorithms on the 18 test functions.

图 5 中纵坐标 $\text{Error}=f(x)-f(x^*)$ ，采用对数尺度，由于计算机使用 IEEE 754 标准，故实验浮点数精确至 $1\text{E-}15$ 。从图中可见基于协方差矩阵的 MQHOA 相比与其他算法有更快的收敛速度和收敛精度，尤其在 C 组函数的比较中，由于基于协方差矩阵的

MQHOA 的各维尺度是通过旋转矩阵变换的，故相比于其他算法有更多优势，从图中看最明显的就是收敛速度，而如果换作原 MQHOA 则基本无法收敛。同时 B 组与 C 组中的同类函数算法除了收敛精度外，收敛趋势也基本一致。

表 3 30 维函数比较结果

Table3 Compare results for 30-D problems

F	MQHOA	PSO	FES	ABC	DE
f_1	$2.17\text{E-}32 \pm 1.19\text{E-}34^\dagger$	$1.65\text{E-}28 \pm 4.00\text{E-}29^\dagger$	$5.04\text{E-}03 \pm 1.45\text{E-}04^\dagger$	$0.0 \pm 0.0^\dagger$	$1.19\text{E-}02 \pm 1.70\text{E-}02^\dagger$
f_2	$1.40\text{E-}28 \pm 1.20\text{E-}29^\dagger$	$5.00\text{E}+03 \pm 1.50\text{E}+02^\dagger$	$1.41\text{E}+03 \pm 6.70\text{E}+01^\dagger$	$1.34\text{E-}12 \pm 9.55\text{E-}13^\dagger$	$3.56\text{E-}02 \pm 7.48\text{E-}03^\dagger$
f_3	$5.53\text{E-}42 \pm 1.20\text{E-}43^\dagger$	$2.13\text{E}+00 \pm 1.03\text{E}+00^\dagger$	$1.88\text{E-}01 \pm 1.34\text{E-}02^\dagger$	$1.73\text{E-}32 \pm 2.67\text{E-}34^\dagger$	$2.36\text{E-}02 \pm 9.02\text{E-}03^\dagger$
f_4	$8.26\text{E-}24 \pm 2.10\text{E-}24^\dagger$	$1.98\text{E-}19 \pm 2.02\text{E-}20^\dagger$	$1.01\text{E-}01 \pm 1.17\text{E-}01^\dagger$	$7.82\text{E-}11 \pm 2.10\text{E-}11^\dagger$	$2.53\text{E-}02 \pm 1.63\text{E-}02^\dagger$
f_5	$2.33\text{E-}02 \pm 1.70\text{E-}02^\dagger$	$1.56\text{E}+01 \pm 2.54\text{E}+00^\dagger$	$2.50\text{E}+01 \pm 3.80\text{E-}01^\dagger$	$8.48\text{E}+00 \pm 6.71\text{E-}01^\dagger$	$7.48\text{E}+01 \pm 6.52\text{E}+00^\dagger$
f_6	$3.74\text{E-}15 \pm 2.42\text{E-}15^\dagger$	$7.11\text{E-}15 \pm 3.04\text{E-}15^\dagger$	$1.69\text{E-}01 \pm 2.54\text{E-}02^\dagger$	$1.07\text{E-}9 \pm 1.87\text{E-}9^\dagger$	$2.17\text{E-}01 \pm 2.85\text{E-}01^\dagger$
f_7	$6.08\text{E-}02 \pm 1.15\text{E-}02^\dagger$	$6.48\text{E-}02 \pm 2.18\text{E-}02^\dagger$	$1.01\text{E}+02 \pm 1.03\text{E}+02^\dagger$	$6.93\text{E-}02 \pm 2.20\text{E-}02^\dagger$	$6.79\text{E}+01 \pm 2.49\text{E}+01^\dagger$
f_8	$0.0 \pm 0.0^\dagger$	$0.0 \pm 0.0^\dagger$	$1.56\text{E}+01 \pm 6.34\text{E-}01^\dagger$	$0.0 \pm 0.0^\dagger$	$4.70\text{E}+00 \pm 1.30\text{E}+00^\dagger$
f_9	$5.04\text{E}+01 \pm 1.03\text{E}+00^\dagger$	$1.22\text{E}+02 \pm 6.01\text{E}+00^\dagger$	$7.04\text{E}+01 \pm 4.97\text{E}+00^\dagger$	$1.79\text{E}+01 \pm 7.96\text{E}+00^\dagger$	$1.68\text{E}+02 \pm 1.07\text{E}+01^\dagger$
f_{10}	$1.92\text{E-}32 \pm 1.41\text{E-}32^\dagger$	$1.64\text{E}+00 \pm 1.18\text{E}+00^\dagger$	$5.86\text{E-}01 \pm 2.15\text{E-}01^\dagger$	$1.50\text{E-}32 \pm 1.41\text{E-}32^\dagger$	$6.41\text{E-}01 \pm 1.06\text{E-}01^\dagger$
f_{11}	$1.99\text{E-}16 \pm 5.14\text{E-}17^\dagger$	$1.08\text{E}+03 \pm 1.25\text{E}+03^\dagger$	$1.01\text{E}+03 \pm 2.13\text{E}+02^\dagger$	$5.31\text{E-}11 \pm 5.20\text{E-}11^\dagger$	$4.70\text{E}+01 \pm 3.14\text{E}+01^\dagger$
f_{12}	$1.73\text{E-}01 \pm 1.21\text{E-}01^\dagger$	$3.92\text{E}+03 \pm 3.32\text{E}+03^\dagger$	$2.15\text{E}+02 \pm 1.66\text{E}+01^\dagger$	$2.26\text{E}+01 \pm 1.47\text{E-}01^\dagger$	$4.97\text{E}+01 \pm 5.17\text{E}+00^\dagger$
f_{13}	$2.01\text{E}+00 \pm 1.12\text{E-}03^\dagger$	$2.07\text{E}+01 \pm 6.51\text{E-}03^\dagger$	$2.01\text{E}+01 \pm 9.77\text{E-}03^\dagger$	$2.09\text{E}+01 \pm 4.33\text{E-}01^\dagger$	$2.09\text{E}+01 \pm 4.43\text{E-}02^\dagger$
f_{14}	$1.02\text{E}+00 \pm 2.08\text{E}+00^\dagger$	$7.84\text{E}+02 \pm 3.61\text{E}+03^\dagger$	$4.47\text{E}+02 \pm 1.09\text{E}+02^\dagger$	$4.25\text{E}+00 \pm 4.25\text{E}+00^\dagger$	$1.61\text{E}+02 \pm 1.42\text{E}+03^\dagger$
f_{15}	$1.73\text{E}+00 \pm 1.25\text{E-}01^\dagger$	$2.71\text{E}+00 \pm 1.34\text{E}+00^\dagger$	$2.49\text{E}+00 \pm 4.84\text{E-}01^\dagger$	$4.18\text{E}+00 \pm 2.18\text{E}+00^\dagger$	$2.04\text{E}+00 \pm 4.00\text{E}+00^\dagger$
f_{16}	$1.28\text{E}+01 \pm 1.11\text{E}+00^\dagger$	$2.61\text{E}+02 \pm 5.09\text{E}+01^\dagger$	$1.59\text{E}+02 \pm 3.15\text{E}+00^\dagger$	$2.09\text{E}+02 \pm 7.09\text{E}+00^\dagger$	$1.33\text{E}+02 \pm 3.31\text{E}+01^\dagger$
f_{17}	$8.35\text{E}+01 \pm 3.41\text{E}+01^\dagger$	$1.15\text{E}+02 \pm 3.25\text{E}+01^\dagger$	$1.35\text{E}+02 \pm 2.33\text{E}+01^\dagger$	$4.05\text{E}+01 \pm 1.92\text{E}+00^\dagger$	$1.10\text{E}+01 \pm 1.14\text{E}+01^\dagger$
f_{18}	$2.27\text{E-}01 \pm 1.13\text{E-}01^\dagger$	$6.54\text{E}+05 \pm 2.03\text{E}+05^\dagger$	$3.32\text{E}+04 \pm 2.39\text{E}+04^\dagger$	$3.16\text{E}+04 \pm 8.16\text{E}+04^\dagger$	$2.53\text{E}+02 \pm 1.25\text{E}+01^\dagger$

30 维的函数比较表与表 2 类似，整体上来说基于协方差矩阵的 MQHOA 经过 5000 次迭代后收敛精度更高，不管是对单峰、多峰还是旋转函数，

基于协方差矩阵表现均衡，尤其是对 C 组中的旋转函数，其具有更高的性能。

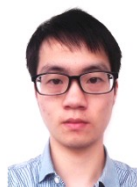
5 结论

本文改进了原 MQHOA 算法, 利用新的协方差矩阵和旋转矩阵等技术解决了原算法在 QHO 收敛和 M 收敛过程中存在的问题, 使算法在各维度上能动态调节, 通过与原 MQHOA 算法的对比实验可知基于协方差矩阵的 MQHOA 算法收敛速度更快, 并且更适合处理高维问题. 随后本文将基于协方差矩阵的 MQHOA 算法与群体优化领域内经典算法作对比, 包括 PSO 算法、FES 算法、ABC 算法和 DE 算法, 组织了 18 个评测函数分三个不同

的组和两种维度进行对比, 实验结果表明改进的 MQHOA 算法在收敛精度、收敛速度、鲁棒性上的表现总体优于其他算法, 尤其处理旋转过后的函数性能会更强. 尽管在本文中该算法表现较好, 但还有很多潜在的地方值得去挖掘和完善, 比如迭代步长的构造, α 值的自动选取等, 这些将成为今后论文的研究点, 以进一步提高 MQHOA 的优化能力.

参考文献 (References) :

- [1] Sastry K, Goldberg D E, Kendall G. Genetic algorithms [M]. Search methodologies, 2014: 93-117
- [2] Muc A, Muc-Wierzoń M. An evolution strategy in structural optimization problems for plates and shells [J]. Composite Structures, 2012, 94(4): 1461-1470
- [3] Regis R G. Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions[J]. Evolutionary Computation, IEEE Transactions on, 2014, 18(3): 326-347
- [4] Brest J, Greiner S, Bošković B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems[J]. Evolutionary Computation, IEEE Transactions on, 2006, 10(6): 646-657
- [5] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. Evolutionary Computation, IEEE Transactions on, 2006, 10(3): 281-295
- [6] Karaboga D, Gorkemli B, Ozturk C, et al. A comprehensive survey: artificial bee colony (ABC) algorithm and applications[J]. Artificial Intelligence Review, 2014, 42(1): 21-57
- [7] Dorigo M, Birattari M, Stützle T. Ant colony optimization[J]. Computational Intelligence Magazine, IEEE, 2006, 1(4): 28-39
- [8] Yang X S, Hossein GANDOMI A. Bat algorithm: a novel approach for global engineering optimization[J]. Engineering Computations, 2012, 29(5): 464-483
- [9] Wang Peng, Huang Yan, Ren Chao, et al. Multi-Scale quantum harmonic oscillator for High-Dimensional function global optimization algorithm. Acta Electronica Sinica, 2013, 41(12): 2468-2473(in chinese)
(王鹏, 黄焱, 任超, 郭又铭. 多尺度量子谐振子高维函数全局优化算法. 电子学报, 2013, 41(12): 2468-2473)
- [10] Lu Zhi-Jun, An Jun-Xiu, Wang Peng. Partition-based MQHOA for Multimodal Optimization. Acta Automatica Sinica. 2016, 42(2): 235-245. (in chinese)
(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)
- [11] Wang Peng, Huang Yan. Physical model of multi-scale quantum harmonic oscillator optimization algorithm. Journal of Frontiers of Computer Science and Technology, 2015, 9(10):1271-1280.(in chinese)
(王鹏, 黄焱. 多尺度量子谐振子优化算法物理模型. 计算机科学与探索, 2015, 9(10): 1271-1280)
- [12] Larranaga P. A review on estimation of distribution algorithms [M]. Estimation of distribution algorithms, 2002: 57-100
- [13] Lagragnaga P, Lozano J A, Bengoetxea E. Estimation of distribution algorithms based on multivariate normal and Gaussian network [R]. Spain: University of the Basque Country, 2001
- [14] Larranaga P, Etxeberria R, Lozano J A, et al. Optimization by learning and simulation of Bayesian and Gaussian networks[J]. 1999
- [15] Rubinstein R Y, Kroese D P. The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning [M]. Springer Science & Business Media, 2013
- [16] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. Evolutionary Computation, IEEE Transactions on, 1999, 3(2): 82-102
- [17] Sabat S L, Ali L, Udgata S K. Integrated learning particle swarm optimizer for global optimization[J]. Applied Soft Computing, 2011, 11(1): 574-584
- [18] Salomon R. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms[J]. BioSystems, 1996, 39(3): 263-278
- [19] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005
- [20] Beyer H G. The theory of evolution strategies [M]. Springer Science & Business Media, 2013
- [21] Civicioglu P, Besdok E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms[J]. Artificial Intelligence Review, 2013, 39(4): 315-346



Lu Zhijun, born in 1991. Master. His main research interests include distributed computing and intelligent algorithm.



Wang Peng, born in 1975. Professor at the College of computer science and technology, Southwest University for Nationalities. Received his PhD degree from Chengdu Institute of Computer Application, Chinese Academy of Sciences in 2004. His main research interests include distributed computing and intelligent algorithm. (wp002005@163.com)



An Junxiu, born in 1970. Professor at the College of Software, Chengdu University of Information Technology. She received her master degree from Xi'an Jiaotong University in 2004. Her research interests include social computing, intelligent searching, and distributed computing. (anjunxiu@cuit.edu.cn)