doi:10.11959/j.issn.1000-436x.2015200

M-精英协同进化分子动理论优化算法

范朝冬，章兢，易灵芝
(湘潭大学 信息工程学院，湖南 湘潭 411105)

摘 要：提出了一种 *M*-精英协同进化分子动理论优化算法(MECKMTOA, *M*-elite coevolutionary KMTOA)。该算法基于 *M* 个精英以尽量避免发生错误引导，通过精英间的学习与协作提高算法的收敛精度，采用新型的波动算子防止算法陷入按维早熟。测试结果表明，MECKMTOA 在求解精度、算法稳定性、高维函数求解等方面均表现出良好性能。

关键词：优化算法；分子动理论优化算法；函数优化；精英策略；协同进化
中图分类号：TP18 **文献标识码**：A

M-elite coevolutionary kinetic-molecular theory optimization algorithm

FAN Chao-dong, ZHANG Jing, YI Ling-zhi
(College of Information Engineering, Xiangtan University, Xiangtan 411105, China)

Abstract: *M*-elite coevolutionary kinetic-molecular theory optimization algorithm (MECKMTOA) was proposed. MECKMTOA uses *M* elites to avoid misleading, improves the convergence precision by learning and collaboration among the elites, employs a new wave operator to prevent premature by dimension. The results show that MECKMTOA has good performance in precision and stability, and can solve the high-dimensional function optimization problems well.
Key words: optimization algorithm; kinetic-molecular theory optimization algorithm; function optimization; elitist strategy; coevolution

1 引言

优化问题是科学研究和工程应用中的一个热点问题，寻求快速、稳健、有效的优化技术是各行各业长期探讨的课题，其中智能优化算法因实现简单、运算速度快、优化效果好，已引起国内外学者的广泛关注。依据原理的不同，现有的智能优化算法大致可分为 3 类：①基于遗传进化机制的优化算法，如遗传算法[1]、进化策略[2]、进化规划[3]等；②基于生物群体智能的优化算法，如微粒群算法[4]、猴群算法[5]、人工蜂群算法[6]等；③基于物理现象或规律的优化算法，如类电磁机制优化算法[7]、中心力算法[8]、万有引力算法[9]等。由于物理学的高度发展，模拟物理现象或物理规律的优化算法因具有理论来源清晰、算法模型简单、规范且寻优效果较好等诸多优点，正逐渐成为优化算法研究的热点。

鉴于基于物理规律优化算法的诸多优点，通过模拟分子机制，笔者于 2013 年首次提出了分子动理论优化算法(KMTOA)并将其应用于函数优化问题[10]。KMTOA 能较好地兼顾收敛性和种群多样性，在快速收敛的同时能尽量避免陷入局部极值，表现出良好的优化性能。2014 年，笔者将其与斜分 Otsu 法相结合以对氧化铝回转窑火焰图像进行分割，取得了理想的分割效果[11]。KMTOA 虽然具备良好的优化性能，但仍然存在

收稿日期：2014-12-20；**修回日期**：2015-03-20
基金项目：国家自然科学基金资助项目(61440026)；湖南省教育厅科学研究重点基金资助项目(15C1327)
Foundation Items: The National Natural Science Foundation of China (61440026); Scientific Research Project of Hunan Provincial Education Department (15C1327)

一些不足，如缺乏局部搜索机制，求解精度有待提高；当前最优值为局部极值时，算法可能发生错误引导。

协同进化思想能充分地考虑种群与种群间、种群与环境间在进化过程中的相互协调关系，通过协同进化以改善各自性能，已成为智能优化领域的研究热点[12,13]。精英通常指种群中适应值较高的个体，这些个体对整个种群的进化起着重要的引导作用[14,15]。慕彩红等基于 M 个精英来建立团队，提出了 *M*-精英协同进化算法并将其应用于函数优化，取得了较好的效果[16,17]。因此，本文针对 KMTOA 的不足，提出了一种 *M*-精英协同进化分子动理论优化算法(MECKMTOA, *M*-elite coevolutionary KMTOA)。该算法采用 M 个精英代替单个精英来实现引导操作，以减小发生错误性引导的概率；基于学习与协作机制，精英能学习其他精英的先进知识而对自身进行优化搜索，从而提高算法的收敛精度；设计了一种自适应的波动算子，以防止算法陷入按维早熟。实验结果表明，MECKMTOA 不仅具备更好的求解精度和算法稳定性，而且能更好地求解高维函数问题。

2　分子动理论优化算法及其不足

2.1　算法简介

在基于群体的优化算法中，算法由可行域中的一组随机点出发，依据随机点的目标函数值，通过一定的搜索策略使其收敛到问题的最优解。依据不同的原理各种算法采用了不同的搜索策略，其中 KMTOA 是受分子动理论的启发而提出的一种全局优化算法。该算法将问题的每个解视为一个分子，各分子在当前最优分子的吸引或排斥作用下运动而完成搜索过程；为增强算法跳出局部极值的能力，对于不受力的分子，通过模拟分子热运动而进行随机扰动。基于分子相互作用和热运动机制，KMTOA 在搜索过程中能较好地兼顾收敛性和种群多样性。

设分子总数为 N，问题维数为 D，分子 i 的位置和质量分别为 x_i 和 m_i，当前最优分子的位置和质量分别为 x_{Best} 和 m_{Best}。$P_{\mathrm{attraction}}$、$P_{\mathrm{repulsion}}$ 和 P_{wave} $(P_{\mathrm{attraction}}+P_{\mathrm{repulsion}}+P_{\mathrm{wave}}=1)$ 分别表示分子受当前最优分子吸引、排斥和不受力的概率；当分子不受力时，对其进行随机扰动以增强算法的全局搜索能力。KMTOA 可简要描述如下[10]。

当 $rand < P_{\mathrm{attraction}}$ ($rand$ 为 0~1 的随机变量)时，分子 i 受当前最优分子的吸引，其引力加速度可计算如下

$$a_i=\frac{Gm_{\mathrm{Best}}m_i(x_{\mathrm{Best}}-x_i)}{m_i}=Gm_{\mathrm{Best}}(x_{\mathrm{Best}}-x_i) \quad (1)$$

其中，G 为引力常量。

当 $P_{\mathrm{attraction}} < rand \leqslant (P_{\mathrm{attraction}}+P_{\mathrm{repulsion}})$ 时，分子 i 受当前最优分子的排斥，其斥力加速度可计算如下

$$a_i=-\frac{Gm_{\mathrm{Best}}m_i(x_{\mathrm{Best}}-x_i)}{m_i}=-Gm_{\mathrm{Best}}(x_{\mathrm{Best}}-x_i) \quad (2)$$

当 $(P_{\mathrm{attraction}}+P_{\mathrm{repulsion}}) < rand \leqslant 1$ 时，分子 i 不受力。此时，该分子进行随机扰动，其加速度可计算如下

$$a_{ij}=\begin{cases}A(x_{\max_j}-x_{\min_j})N(0,1), rand1 \leqslant p_m\\ 0, rand1 > p_m\end{cases} \quad (3)$$

其中，a_{ij} 为分子 x_i 的 j 维扰动加速度，p_m 为变异率，($rand1$ 为 0~1 的随机变量)，$x_{\max_j}$、$x_{\min_j}$ 分别为解空间第 j 维的上、下界。$A(A=1-0.9t/T)$为振动幅度，其中，t 为当前迭代次数，T 为算法的总迭代次数；$N(0,1)$为服从标准正态分布的随机变量。

计算出分子 i 的加速度后，可定义其速度和位置更新公式如下

$$V_i(t+1)=\left(\frac{0.9-0.5t}{T}\right)V_i(t)+a_i \quad (4)$$

$$x_i(t+1)=x_i(t)+V_i(t+1) \quad (5)$$

2.2　算法存在的不足

KMTOA 仅依靠种群中的当前最优分子来实现寻优操作，故算法在优化过程中易出现图 1 所示的情形：当前最优分子 x_{Best} 为一局部极值，其他分子受其引导向该局部极值收敛，从而造成了算法的错误收敛。虽然波动算子使算法具有跳出局部极值的能力，但是波动率和变异率都很小，且变异是随机的，

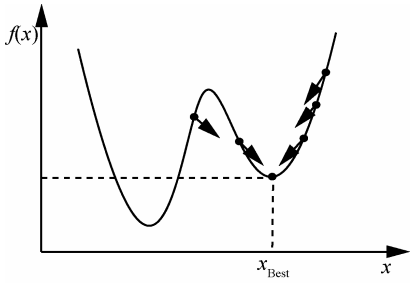

图 1　KMTOA 的错误性引导

故算法跳出局部极值的过程是艰难而漫长的。可见，KMTOA 仅依靠当前最优分子来引导搜索过程，存在一定的片面性。虽然当问题仅含一个极值时，算法效率较高；但是当问题包含多个局部极值时，该机制将严重影响算法的效率。

定义 1 在算法迭代过程中，如种群中的所有分子均满足

$$\left|\max(\boldsymbol{x}_{\bullet j})-\min(\boldsymbol{x}_{\bullet j})\right|\leqslant\varepsilon \tag{6}$$

则称算法陷入按维早熟。其中，$\boldsymbol{x}_{\bullet j}$ 表示种群中所有分子的第 j 维分量所构成的向量，$\max(\boldsymbol{x}_{\bullet j})$ 和 $\min(\boldsymbol{x}_{\bullet j})$ 分别表示种群中所有分子第 j 维分量的最大、最小值，ε 为一极小量。

由式(6)可知，当算法陷入按维早熟时，种群中的所有分子在某一维上的差异极小。此时，分子在其他维上仍然可能继续进化，故算法此时也许尚未发生真正的早熟，但如任其发展，则算法极有可能发生早熟。正是为了区别于传统概念上的早熟，本文将其称为按维早熟。由 KMTOA 的实现过程可知，当陷入按维早熟时，要使分子在该维上出现差异(即跳出按维早熟)是较为困难的。此外，KMTOA 还缺乏局部寻优机制，其寻优精度有待提高。

3 改进的分子动理论优化算法

3.1 基于能力的团队构建算子

在 *M*-精英协同进化算法中，每个精英团队的规模都相等[16,17]，这与自然界中优胜劣汰的生存法则严重不符。自然界中能力强的个体在竞争过程中将处于优势地位，往往占据更多的资源，能够繁殖更多后代[18]，故其团队规模也相对较大。在优化过程中，精英的能力越强，则意味着问题的解在其附近的可能性越大。因此，对于能力较强的精英，应组建规模较大的团队，以便引导更多分子进行优化搜索，这对提高算法效率具有重要意义。基于上述思想，本文设计了一种基于能力的团队构建算子，将每个精英的能力定义为

$$Ability_i=\frac{f(x_{\text{worst}})-f(x_i)}{\sum_{j=1}^{M}(f(x_{\text{worst}})-f(x_j))},\ j=1,2,\cdots,M \tag{7}$$

其中，M 表示精英种群的规模，x_{worst} 和 $f(x_{\text{worst}})$ 分别表示当前种群中的最差分子及其函数值。根据精英能力的定义，其团队规模可计算如下

$$Members_i=Popsize\cdot Ability_i \tag{8}$$

其中，$Members_i$ 表示精英 i 所能构建团队的规模，$Popsize$ 表示种群规模。可见，精英的目标函数值越小，其能力越强，能够组建团队的规模就越大，引导分子搜索到最小值的可能性也就越大。

3.2 基于学习机制的精英协作算子

在 KMTOA 中，由于缺乏相应的局部搜索机制，当求解复杂问题时，其求解精度往往难以令人满意。对此，当团队成员属于精英种群时，基于学习机制设计了一种协作算子。该算子通过在精英间交换信息以引导精英对其邻域进行搜索，从而实现局部搜索。基于学习机制的精英协作算子定义如下

$$u_{ik}=x_{ik}+(x_{ik}-x_{jk})N(0,1),k=1,2,\cdots,D \tag{9}$$

其中，$i\neq j$，x_{ik} 表示精英 i 的第 k 维分量，$N(0,1)$为服从标准正态分布的随机变量，D 为解向量的维数，u_i 为协作过程完成后所得到的临时个体。

得到精英 x_i 的临时个体 u_i 后，可分别按式(10)、式(11)对精英 x_i、x_j 进行更新。

$$x_i=\begin{cases}u_i, f(u_i)\leqslant f(x_i)\\ x_i, f(u_i)>f(x_i)\end{cases} \tag{10}$$

$$x_j=\begin{cases}u_i, f(x_j)\geqslant f(u_i)>f(x_i)\\ x_j, \text{其他}\end{cases} \tag{11}$$

由上式可知，如临时个体 u_i 对精英 x_i 进行了更新，则 u_i 将不会对 x_j 进行更新，这样做是为了防止种群快速收敛到同一个值而造成算法早熟。

3.3 防按维早熟的波动算子

当用 KMTOA 求解复杂性问题时，算法可能出现按维早熟。由 KMTOA 的实现过程可知，当出现按维早熟时，算法主要依靠波动算子来跳出局部极值；但是 KMTOA 中的波动率和变异率都比较小，这就决定了算法跳出局部极值的可能性较小。对此，本文设计了一种防按维早熟的波动算子，该算子依据种群在当前维上的差异来对变异率进行计算；如差异较小，则该维变异率较大，以使分子能以较大的概率发生变异而保持种群多样性。防按维早熟的变异率计算公式为

$$p_m=\left(1-\frac{\max(\boldsymbol{x}_{\bullet j})-\min(\boldsymbol{x}_{\bullet j})}{U_j^a-U_j^b}+\alpha\right)\beta \tag{12}$$

其中，$\max(\boldsymbol{x}_{\bullet j})$ 和 $\min(\boldsymbol{x}_{\bullet j})$ 的含义同上，U_j^a 和 U_j^b

分别表示问题取值空间在第 j 维上的上界和下界。α 为算法初期的调节因子，主要是为了使算法在初期具有一定的变异能力，故其值不应过大；β 为比例因子，主要是防止算法后期的变异率 p_m 大于 1 而影响算法的收敛性，故其值既不宜过大、也不宜过小。大量测试的结果表明，当 α=0.01、β=0.5 时，算法的性能较好。计算出分子的变异率后，可按式(3)计算其波动加速度，进而完成波动操作。

3.4　基于分子动理论的引导算子

当团队成员 x_j 来自普通种群时，精英 x_i 通过引导操作引导普通分子进行优化搜索。在引导过程中，精英分别依据吸引率 $P_{\text{attraction}}$、排斥率 $P_{\text{repulsion}}$ 和波动率 P_{wave} 来判断对分子进行何种引导，具体操作如下：①当精英对普通分子进行吸引操作时，按式(1)计算其相应的引力加速度；②当精英对普通分子进行排斥操作时，按式(2)计算其相应的斥力加速度；③当普通分子进行随机扰动时，分别按式(12)、式(3)计算其相应的变异率和波动加速度。

计算出普通分子的加速度 a_j 后，根据式(4)计算其速度 V_j，并计算其临时个体 *Tmp* 为

$$Tmp = x_j + V_j \tag{13}$$

根据临时个体 *Tmp*，分别按式(14)、式(15)对精英 x_i 和普通分子 x_j 进行更新

$$x_i = \begin{cases} Tmp, f(Tmp) \leqslant f(x_i) \\ x_i, \text{其他} \end{cases} \tag{14}$$

$$x_j = x_i \tag{15}$$

由上式可知，只有当临时个体 *Tmp* 比精英 x_i 好时，*Tmp* 才会取代 x_i 而成为新的精英；而无论如何，临时个体 *Tmp* 都会取代 x_j 而成为新的普通分子。这么做一方面能够避免精英退化，另一方面又能较好地维持普通种群的多样性。

3.5　算法流程

MECKMTOA 的算法流程如图 2 所示。

4　算法收敛性分析

定理 1　在算法运行过程中，MECKMTOA 的种群序列 $\{X_t, t \geqslant 0\}$ 为有限齐次 Markov 链。

证明　MECKMTOA 采用实数编码，如设定精度为 ε，则解空间 S 可视为 $|S| = \prod_{i=1}^{n} \left\lceil (\overline{x_i} - \underline{x_i}) / \varepsilon \right\rceil$ 的离散空间，故种群序列有限；由算法过程可知，第 $t+1$ 代的种群 X_{t+1} 仅与其上代种群 X_t 有关。因此，种群序列 $\{X_t, t \geqslant 0\}$ 为有限齐次的 Markov 链。

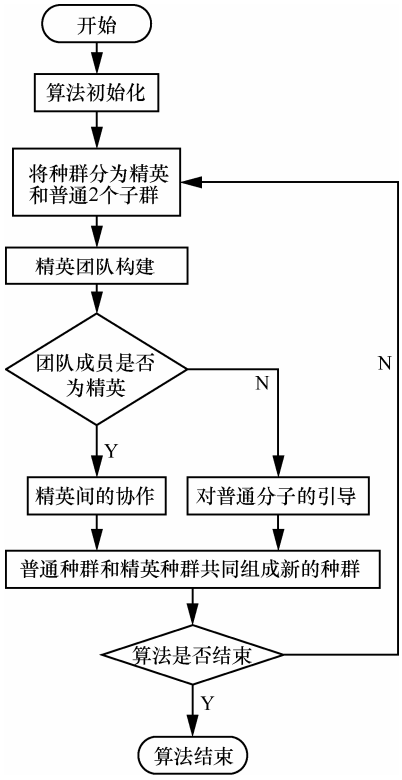

图 2　MECKMTOA 算法流程

定义 2　设 f 为 S 上的目标函数，f^* 为全局最优值，则最优解集 s^* 可表示为

$$s^* = \left\{ x \left| \min_{x \in X_t} f(x) = f^* \right. \right\} \tag{16}$$

定义 3　设 $f_t = \min_{x \in X_t} \{ f(x_i), i = 1, 2, \cdots, N \}, t \geqslant 1$ 为一随机变量序列，其中，f_j 表示 j 时刻状态中的最优值。如有

$$\lim_{t \to \infty} P\{ f_t = f^* \} = 1 \tag{17}$$

则称算法以概率 1 收敛到全局最优值 f^*。

定理 2　MECKMTOA 的种群最优值序列 $f_t = \min_{x \in X_t} \{ f(x_i), i = 1, 2, \cdots, n \}, t \geqslant 1$ 单调不增，即对于任意 $t \geqslant 0$，有 $f_{t+1} \leqslant f_t$。

证明　由式(10)、式(11)的精英更新策略可知，临时个体只有在优于当前精英时，才会取代当前精英，也即精英在算法运行过程中不会退化。故对任意 $t \geqslant 0$，有 $f_{t+1} \leqslant f_t$ 成立。

定理 3　MECKMTOA 以概率 1 收敛。

证明　设 $p_i(t)$ 为种群 X_t 在状态 s_i 的概率，

$s_i \in S$，令 $I=\{i|s_i \cap s^* \neq \varnothing\}$，则有 $p_t=\sum_{i\in I} p_i(t)$，由 Markov 链的性质可知

$$\begin{aligned} p_{t+1} &= \sum_{s_i\in S}\sum_{j\in I} p_i(t)p_{ij}(t) \\ &= \sum_{i\in I}\sum_{j\in I} p_i(t)p_{ij}(t)+\sum_{i\notin I}\sum_{j\in I} p_i(t)p_{ij}(t) \end{aligned} \tag{18}$$

由于

$$\begin{aligned} p_t &= \sum_{i\in I} p_i(t) \\ &= \sum_{i\in I}\sum_{j\in I} p_i(t)p_{ij}(t)+\sum_{i\in I}\sum_{j\notin I} p_i(t)p_{ij}(t) \end{aligned} \tag{19}$$

因 MECKMTOA 的种群最优值序列不会发生退化，故 $\sum_{i\in I}\sum_{j\notin I} p_i(t)p_{ij}(t)=0$，于是式(18)可转换为

$$p_{t+1}=p_t+\sum_{i\notin I}\sum_{j\in I} p_i(t)p_{ij}(t)>p_t \tag{20}$$

所以 $\lim_{t\to\infty} p_t=1$，即 MECKMTOA 以概率 1 收敛。

5 数值仿真与分析

为便于与 KMTOA 比较，选择文献[10]中的 f_6~f_{20} 15 个经典测试函数作为测试对象(记为 F_1~F_{15})，对 MECKMTOA 的性能进行测试。测试实验分为 4 个部分：第 1 部分为 M 取值及相关算子的合理性分析实验，主要讨论 M 的取值及各算子对算法性能的影响；第 2 部分为纵向比较实验，对改进前后算法的性能进行比较，以验证改进措施的有效性；第 3 部分为横向比较实验，将 MECKMTOA 与组织进化算法(OEA, organizational evolutionary algorithm)和 M-精英协同进化算法(MECA, M-elite coevolutionary algorithm)等协同进化算法进行比较；第 4 部分为高维及超高维函数测试实验，测试 MECKMTOA 解决高维及超高维问题的能力。所有实验运行的硬件平台均为：Intel(R) Celeron(R) 1.50 GHz 的 CPU，2 GB 内存，Windows 7 操作系统，编程语言为 Matlab R2011a。

5.1 M 及相关算子合理性分析

为测试精英数量 M 及各算子对 MECKMTOA 性能的影响，分别选用 F_5、F_6 和 F_9 这 3 个复杂函数对 MECKMTOA 进行测试。为便于后续与文献[16]中的 MECA 和 OEA 进行比较，MECKMTOA 的种群规模取 100，算法的最大迭代次数取 3 000，每种情况下算法均独立运行 50 次，实验结果取 50 次测试的平均值。

5.1.1 M 取值分析

为测试精英种群的规模 M 对 MECKMTOA 的影响，首先从种群规模范围内选取部分值作为 M 的取值，以确定 M 的大致取值范围，然后在该范围内对 M 的取值做进一步的讨论。参照文献[16]，分别取 M=3,5,10,15,20, 30,40,50,70,90，对 MECKMTOA 的不同结果进行比较以确定 M 的大致取值范围。由表 1 可知：对于函数 F_5、F_6 和 F_9，当 M 取值小于 10 时，算法的优化效果较好；当 M 大于 10 时，算法的求解结果随 M 的增大而越来越差，故 M 的理想取值应小于 10。

为讨论 M 的具体取值，对 M=1,2,3,4,5,6,7,8,9,10 时，MECKMTOA 的不同结果进行了比较，比较结果如表 2 所示。由表 2 可知：对于函数 F_5、F_6 和 F_9，M 取 4~6 的值时算法能取得较好的效果；M 过大或过小，都将对算法的性能造成影响；当 M 取 5 左右时，MECKMTOA 的实验结果相对较好。

表 1　M 对 MECKMTOA 的影响 1

M	平均值			标准差		
	F_5	F_6	F_9	F_5	F_6	F_9
3	$3.778\,7\times10^{-5}$	0.377 5	$-1.170\,7\times10^{4}$	$3.478\,2\times10^{-5}$	1.176 7	$6.434\,0\times10^{2}$
5	**$3.103\,0\times10^{-5}$**	**0.165 3**	**$-1.188\,1\times10^{5}$**	**$2.765\,7\times10^{-5}$**	**0.354 9**	$6.026\,0\times10^{2}$
10	$4.108\,8\times10^{-5}$	1.835 3	$-1.166\,8\times10^{4}$	$3.576\,6\times10^{-5}$	11.117 6	$6.469\,1\times10^{2}$
15	$5.084\,0\times10^{-5}$	1.760 0	$-1.165\,0\times10^{4}$	$5.105\,3\times10^{-5}$	10.147 2	$6.748\,1\times10^{2}$
20	$5.322\,4\times10^{-5}$	3.419 9	$-1.149\,0\times10^{4}$	$4.722\,5\times10^{-5}$	15.570 3	$7.025\,3\times10^{2}$
30	$6.832\,0\times10^{-5}$	3.545 2	$-1.087\,7\times10^{4}$	$6.793\,3\times10^{-5}$	16.891 7	$5.087\,5\times10^{2}$
40	$9.264\,3\times10^{-5}$	3.423 4	$-1.088\,3\times10^{4}$	$7.849\,0\times10^{-5}$	16.299 9	$6.268\,7\times10^{2}$
50	$1.867\,7\times10^{-5}$	5.017 7	$-1.049\,8\times10^{4}$	$1.874\,0\times10^{-4}$	17.830 8	$5.784\,5\times10^{2}$
70	$4.982\,3\times10^{-5}$	21.561 5	$-1.001\,4\times10^{4}$	$4.719\,2\times10^{-4}$	11.072 5	**$4.894\,7\times10^{2}$**
90	0.002 2	93.958 8	$-9.092\,9\times10^{3}$	0.001 7	42.249 1	$7.466\,7\times10^{2}$

表 2　　*M* 对 MECKMTOA 的影响 2

M	平均值			标准差		
	F_5	F_6	F_9	F_5	F_6	F_9
1	$4.556\,5\times10^{-5}$	0.411 5	$-1.162\,7\times10^{4}$	$4.490\,8\times10^{-5}$	1.122 1	$7.112\,7\times10^{2}$
2	$3.883\,8\times10^{-5}$	0.472 0	$-1.168\,6\times10^{4}$	**$2.661\,7\times10^{-5}$**	1.027 2	$6.789\,2\times10^{2}$
3	$3.778\,7\times10^{-5}$	0.377 5	$-1.170\,7\times10^{4}$	$3.478\,2\times10^{-5}$	1.176 7	$6.434\,0\times10^{2}$
4	$3.721\,0\times10^{-5}$	0.301 5	$-1.182\,1\times10^{4}$	$4.407\,2\times10^{-5}$	1.100 6	$6.674\,5\times10^{2}$
5	**$3.103\,0\times10^{-5}$**	**0.165 3**	**$-1.188\,1\times10^{4}$**	$2.765\,7\times10^{-5}$	**0.354 9**	**$5.868\,1\times10^{2}$**
6	$3.417\,2\times10^{-5}$	0.410 2	$-1.175\,1\times10^{4}$	$3.105\,1\times10^{-5}$	1.260 9	$6.769\,1\times10^{2}$
7	$3.717\,1\times10^{-5}$	0.400 8	$-1.173\,7\times10^{4}$	$3.539\,9\times10^{-5}$	1.003	$6.052\,4\times10^{2}$
8	$3.751\,4\times10^{-5}$	0.444 0	$-1.169\,8\times10^{4}$	$4.550\,1\times10^{-5}$	1.211 1	$6.058\,2\times10^{2}$
9	$3.868\,8\times10^{-5}$	1.812 9	$-1.167\,0\times10^{4}$	$2.915\,8\times10^{-5}$	11.183 9	$6.533\,2\times10^{2}$
10	$4.108\,8\times10^{-5}$	1.835 3	$-1.166\,8\times10^{4}$	$3.576\,6\times10^{-5}$	11.117 6	$6.469\,1\times10^{2}$

为分析 *M* 取值的合理性，借鉴文献[19]阈值 *Q* 的分析方法，对 F_1~F_{15} 在 *M* 分别取 1,2,⋯,10 时的不同情况进行测试，并运用最小二乘法对测试结果的均值及其平均标准差进行拟合。拟合结果如下，其拟合曲线分别如图 3 和图 4 所示。

$$f_1(x)=3.822\,8x^2-39.535\,2x-725.229\,3 \quad (21)$$

$$f_2(x)=1.191\,9x^2-11.981\,0x+53.442\,3 \quad (22)$$

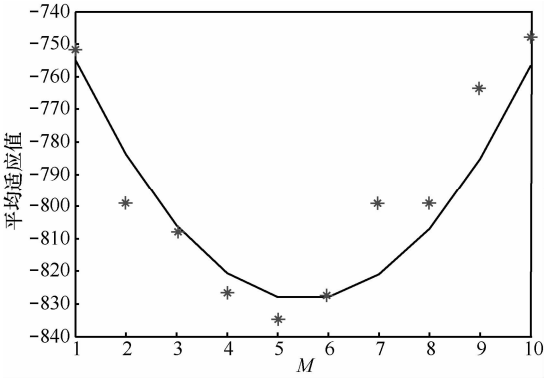

图 3　*M* 取不同值时，平均适应值拟合曲线

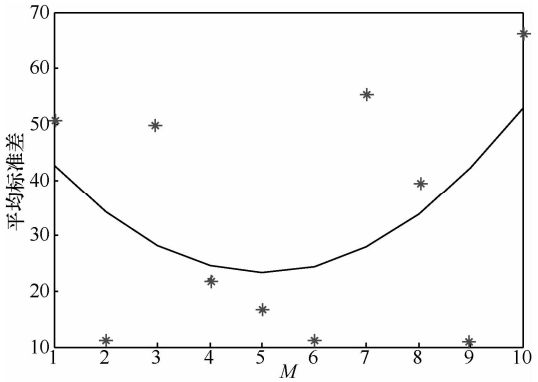

图 4　*M* 取不同值时，平均标准差拟合曲线

拟合结果表明：当 *M* 分别取 5.170 9 和 5.026 0 时，平均适应值和平均标准差能取得最小值。*M* 过大、过小都会影响算法效果，这是因为 *M* 过大，种群过于分散，不利于算法收敛；*M* 过小，则种群多样性差，易造成欺骗性引导。又因 *M* 取值为整数，故在后续的所有实验中均取 *M*=5。拟合结果有力地验证了前文的分析。

5.1.2　相关算子分析

为改善算法的性能，本文设计了基于学习机制的精英协作算子、基于能力的团队构建算子及防按维“早熟”的波动算子。为验证各算子对算法性能的影响，用 MECKMTOA1 表示 MECKMTOA 中剔除了精英协作算子的算法，用 MECKMTOA2 表示 MECKMTOA 中固定了团队规模的算法，用 MECKMTOA3 表示 MECKMTOA 中采用了传统波动算子的算法，4 种算法对函数 F_5、F_6 和 F_9 的测试结果如表 3 所示。由表 3 可知，3 个算子中基于能力的团队构建算子对 MECKMTOA 的影响最小，但因基于能力的团队构建算子能召集更多的成员对优秀分子附近进行搜索，故该算子对提高算法收敛的速度和精度具有一定影响；基于学习机制的精英协作算子通过在精英间进行交换信息，对精英附近进行搜索，故对提高算法的收敛精度具有重要影响；防按维早熟的波动算子能根据每维的情况自适应地调整变异率，能较好地防止算法陷入局部极值，如去掉该算子，则算法的性能将大幅下降。综合考虑，3 个算子能从不同的角度对算法的性能进行改进，故同时采用该 3 个算子的 MECKMTOA 最为理想。

表 3 **各算子对 MECKMTOA 的影响**

算法	平均值			标准差		
	F_5	F_6	F_9	F_5	F_6	F_9
MECKMTOA1	5.4133	0.225 2	$-1.151\ 4\times10^{-4}$	$6.178\ 8\times10^{-5}$	0.581 9	$6.794\ 9\times10^{-2}$
MECKMTOA2	$4.645\ 6\times10^{-5}$	0.178 3	$-1.185\ 6\times10^{-4}$	$5.854\ 2\times10^{-5}$	0.387 3	$6.533\ 2\times10^{-2}$
MECKMTOA3	$5.922\ 7\times10^{-5}$	0.478 0	$-1.135\ 8\times10^{-4}$	$4.292\ 9\times10^{-5}$	1.292 6	$\mathbf{5.114\ 2\times10^{-2}}$
MECKMTOA	$\mathbf{3.103\ 0\times10^{-5}}$	**0.165 3**	$\mathbf{-1.188\ 1\times10^{-4}}$	$\mathbf{2.765\ 7\times10^{-5}}$	**0.354 9**	$5.868\ 1\times10^{-2}$

表 4 **MECKMTOA 与 KMTOA 的性能对比**

函数	算法	最小值	最大值	平均值	标准差
F_1	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_2	KMTOA	$7.903\ 5\times10^{-6}$	$8.056\ 3\times10^{-5}$	$2.374\ 3\times10^{-5}$	$1.368\ 5\times10^{-5}$
	MECKMTOA	$\mathbf{2.153\ 3}\times10^{-13}$	$\mathbf{4.829\ 5}\times10^{-10}$	$\mathbf{6.494\ 6}\times10^{-11}$	$\mathbf{1.184\ 5}\times10^{-10}$
F_3	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_4	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_5	KMTOA	$3.680\ 2\times10^{-13}$	$3.240\ 1\times10^{-4}$	$8.346\ 8\times10^{-5}$	$8.244\ 8\times10^{-5}$
	MECKMTOA	$\mathbf{9.474\ 7}\times10^{-7}$	$\mathbf{1.770\ 7\times10^{-4}}$	$\mathbf{3.103\ 0\times10^{-5}}$	$\mathbf{2.765\ 7}\times10^{-5}$
F_6	KMTOA	$1.731\ 7\times10^{-6}$	4.323 6	0.173 6	0.669 9
	MECKMTOA	$\mathbf{5.380\ 5}\times10^{-28}$	**1.035 8**	**0.165 3**	**0.354 9**
F_7	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_8	KMTOA	0.397 9	0.397 9	0.397 9	$3.364\ 5\times10^{-16}$
	MECKMTOA	0.397 9	0.3979	0.397 9	$3.364\ 5\times10^{-16}$
F_9	KMTOA	$-1.256\ 9\times10^{4}$	$-1.067\ 3\times10^{4}$	$-1.156\ 2\times10^{4}$	$\mathbf{4.958\ 7}\times10^{2}$
	MECKMTOA	$\mathbf{-1.256\ 9}\times10^{4}$	$\mathbf{-1.043\ 8}\times10^{4}$	$\mathbf{-1.188\ 1}\times10^{4}$	$5.868\ 1\times10^{2}$
F_{10}	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_{11}	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_{12}	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_{13}	KMTOA	**0**	**0**	**0**	**0**
	MECKMTOA	**0**	**0**	**0**	**0**
F_{14}	KMTOA	$1.944\ 5\times10^{-8}$	$1.961\ 9\times10^{-6}$	$4.081\ 8\times10^{-7}$	$3.598\ 8\times10^{-7}$
	MECKMTOA	$\mathbf{1.499\ 8\times10^{-32}}$	$\mathbf{2.175\ 1\times10^{-12}}$	$\mathbf{8.616\ 7\times10^{-14}}$	$\mathbf{3.826\ 2\times10^{-13}}$
F_{15}	KMTOA	$\mathbf{-1.867\ 3\times10^{2}}$	$\mathbf{-1.867\ 3\times10^{2}}$	$\mathbf{-1.867\ 3\times10^{2}}$	$9.205\ 2\times10^{-14}$
	MECKMTOA	$\mathbf{-1.867\ 3\times10^{2}}$	$\mathbf{-1.867\ 3\times10^{2}}$	$\mathbf{-1.867\ 3\times10^{2}}$	$\mathbf{2.842\ 2\times10^{-14}}$

5.2 改进前后算法性能比较

为测试 MECKMTOA 的性能，将 MECKMTOA 与 KMTOA 进行比较。2 种算法的种群规模均为 100，最大迭代次数均为 3 000，吸引率、排斥率、波动率等与文献[10]中一致，每个算法独立运行 50 次，选用 50 次结果中的最小值、最大值、平均值和标准差 4 项指标对算法的性能进行比较，比较结果如表 4 所示。由表 4 可知：对于 F_1 等 KMTOA 能取得最优值的函数，MECKMTOA 也均能取得最优值；对于 F_2 等 KMTOA 未能取得最优值的函数，

虽然 MECKMTOA 也未能取得最优值，但是其求解精度具有明显改善，如对于 F_2、F_{14} 等函数，MECKMTOA 的求解精度甚至比 KMTOA 高几个数量级。此外，由比较结果还能看出 MECKMTOA 几乎在各项指标上均优于 KMTOA。

5.3　与其他协同进化算法的性能比较

为进一步验证 MECKMTOA 的性能，将其与文献[16]中的 MECA 和 OEA 这 2 种协同进化算法进行比较。实验选取本文与文献[16]所共有的 8 个测试函数作为测试对象，表 5 为 3 种算法的比较结果，其中 MECA 和 OEA 的数据选自文献[16]。由表 5 可知：除 F_5 和 F_9 外，MECKMTOA 能对所有的函数求得最优值；对于 F_5，虽然 MECKMTOA 未能求得最优值，但其求解精度仍比 MECA 和 OEA 高 1~2 个数量级；在 8 个测试函数中，MECKMTOA 只有对 f_{14} 的求解结果比 MECA 和 OEA 差。故总体考虑，MECKMTOA 在求解精度、算法稳定性等方面比 MECA 和 OEA 具有更好的效果。

5.4　高维及超高维函数测试

为检验 MECKMTOA 求解复杂问题的能力以及问题复杂度的增加对算法性能的影响，用 MECKMTOA 分别求解高维(维数为 10~1 000)和超高维(维数 1 000 以上)条件下的 F_{10} 和 F_{13}，并将其求解结果与文献[20]中的免疫记忆克隆规划算法(IMCPA, immune memory clonal programming algorithm)及无记忆功能的免疫记忆克隆规划算法(nIMCPA)进行比较。算法终止条件为 $\left|f_{\text{best}}\right|<\varepsilon$，$f_{\text{best}}$ 为算法当前的最优值，ε 为算法的收敛精度。参照文献[20]，对于高维函数，平均函数评价次数取算法 10 次运行的平均值；对于超高维函数，平均函数评价次数取算法 5 次运行的平均值；IMCPA 和 nIMCPA 的数据选自文献[20]，“—”表示没有进行该实验。

由表 6 可知：与 IMCPA 和 *n*IMCPA 相比，在相同的收敛精度下，MECKMTOA 所需的平均函数评价次数相对较少；随着维数的增加，MECKMTOA、IMCPA 和 nIMCPA 的平均函数评价次数均有所增长，但 MECKMTOA 的平均函数评价次数增长的相对缓慢。比较结果表明：MECKMTOA 即使在超高维条件下仍能表现出良好的性能，且其性能不随问题维数的增加而迅速下降。

6　结束语

针对 KMTOA 存在的不足，通过引入精英和协同

表 5　MECKMTOA 与 MECA 及 OEA 的比较

函数	平均值			标准差		
	MECKMTOA	MECA	OEA	MECKMTOA	MECA	OEA
F_1	**0**	4.228×10^{-183}	2.481×10^{-30}	**0**	**0**	1.128×10^{-29}
F_3	**0**	1.845×10^{-110}	2.068×10^{-13}	**0**	3.113×10^{-110}	1.440×10^{-12}
F_4	**0**	3.274×10^{-95}	1.883×10^{-9}	**0**	2.313×10^{-94}	3.726×10^{-9}
F_5	$\mathbf{3.103\times10^{-5}}$	4.083×10^{-4}	3.297×10^{-3}	$\mathbf{2.766\times10^{-5}}$	3.800×10^{-4}	1.096×10^{-3}
F_9	-1.188×10^{4}	$\mathbf{-1.257\times10^{4}}$	$\mathbf{-1.257\times10^{4}}$	5.868×10^{2}	7.350×10^{-12}	$\mathbf{5.555\times10^{-12}}$
F_{10}	**0**	**0**	5.430×10^{-17}	**0**	**0**	1.683×10^{-16}
F_{12}	**0**	**0**	5.336×10^{-14}	**0**	**0**	2.945×10^{-13}
F_{13}	**0**	3.844×10^{-3}	1.317×10^{-2}	**0**	7.130×10^{-3}	1.561×10^{-2}

表 6　高维及超高维函数测试结果

函数	算法	*n*=20	*n*=100	*n*=200	*n*=400	*n*=1 000	*n*=2 000	*n*=5 000	*n*=10 000
$F_{10}(\varepsilon=10^{-1})$	IMCPA	1 469	4 988	5 747	12 563	24 408	37 879	87 245	143 700
	nIMCPA	2 777	7 221	13 354	15 475	37 250	66 145	—	—
	MECKMTOA	**1 260**	**2 070**	**2 320**	**2 920**	**3 820**	**3 980**	**4 420**	**5 800**
$F_{13}(\varepsilon=10^{-4})$	IMCPA	2 421	6 713	8 460	15 365	30 906	43 003	125 847	147 037
	nIMCPA	4 657	12 166	21 577	36 515	75 217	81 447	—	—
	MECKMTOA	**1 170**	**1 860**	**2 200**	**2 380**	**2 860**	**3 780**	**3 940**	**4 340**

进化思想，提出了一种 M 精英协同分子动理论优化算法。该算法采用 M 个精英，减小了传统算法发生错误性引导的概率，改善了算法的收敛效率；精英通过协作算子，学习先进知识以改善自己，提高了算法的收敛精度；防按维早熟的波动算子因能自适应地调节波动率，能有效地防止算法发生早熟。为分析 M 的取值对算法性能的影响，讨论了 M 的不同取值情况，确定了其大致取值。为验证各算子的合理性，分析了各算子对算法所造成的影响。与 KMTOA 的纵向比较表明，MECKMTOA 的改进策略是有效的；与 MECA 和 OEA 这 2 种协同进化算法的横向比较结果表明，MECKMTOA 的协同引导机制具有明显优势；对高维及超高维函数的测试结果表明，MECKMTOA 具备求解复杂性问题的能力，且其性能受问题维数的影响相对较小。综合考虑，MECKMTOA 具备较强的优化性能，有望应用于复杂问题的求解，具有重要的理论研究意义和工程应用价值。

参考文献：

[1] LIN C H, CHOY K L, HO G T S, NG T W. A genetic algorithm-based optimization model for supporting green transportation operations[J]. Expert Systems with Applications, 2014, 41(7): 3284-3296.

[2] DONG W Y, ZHOU M C. Gaussian classifier-based evolutionary strategy for multimodal optimization[J]. IEEE Transactions on Neural Networks & Learning Systems, 2014, 25(6): 1200-1216.

[3] KHATOD D K, PANT V, SHARMA J. Evolutionary programming based optimal placement of renewable distributed generators[J]. IEEE Transactions on Power Systems, 2013, 28(2): 683-695.

[4] MIRJALILI S, LEWIS A, SADIQ A S. Autonomous particles groups for particle swarm optimization[J]. Arabian Journal for Science and Engineering, 2014, 39(6): 4683-4697.

[5] ZHENG L. An improved monkey algorithm with dynamic adaptation[J]. Applied Mathematics and Computation, 2013, 222(1): 645-657.

[6] 张冬丽，唐英干，关新平. 用改进的人工蜂群算法设计 AVR 系统最优分数阶 PID 控制器[J].自动化学报，2014, 40(5): 973-979.
ZHANG D L, TANG Y G, GUAN X P. Optimum design of fractional order pid controller for an avr system using an improved artificial bee colony algorithm[J]. Acta Automatica Sinica, 2014, 40(5): 973-979.

[7] ZHANG C J, LI X Y, GAO L WU Q. An improved electromagnetism-like mechanism algorithm for constrained optimization[J]. Expert Systems with Applications, 2013, 40(14): 5621-5634.

[8] FORMATO R A. Central force optimization with variable initial probes and adaptive decision space[J]. Applied Mathematics and Computation, 2011, 2(17): 8866-8872.

[9] MIRJALILI S, LEWIS A. Adaptive gbest-guided gravitational search algorithm[J]. Neural Computing and Applications, 2014, 25(7): 1569-1584.

[10] FAN C D, OUYANG H L, ZHANG Y J, *et al*. Optimization algorithm based on kinetic-molecular theory[J]. Journal of Central South University, 2013, 20(12): 3504-3512.

[11] 范朝冬, 张英杰, 欧阳红林等. 基于改进斜分 Otsu 法的回转窑火焰图像分割[J]. 自动化学报, 2014, 40(11): 2480-2489.
FAN C D, ZHANG Y J, OUYANG H L, *et al*. Improved Otsu method based on histogram oblique segmentation for segmentation of rotary kiln flame image[J]. Acta Automatica Sinica, 2014, 40(11): 2480-2489.

[12] JIAO L C, WANG H D, SHANG R H, *et al*. A co-evolutionary multi-objective optimization algorithm based on direction vectors[J]. Information Sciences, 2013, 228(10): 90-112.

[13] LI H C, FANG L. Co-evolutionary algorithm: an efficient approach for bilevel programming problems[J]. Engineering Optimization, 2014, 46(3): 361-376.

[14] CHEN P H. Two-level hierarchical approach to unit commitment using expert system and elite PSO[J]. IEEE Transactions on Power Systems, 2012, 27(2): 780-789.

[15] YOUSEFIKHOSHBAKHT M, DIDEHVAR F, RAHMATI F. An efficient solution for the vrp by using a hybrid elite ant system[J]. International Journal of Computers Communications & Control, 2014, 9(3): 340-347.

[16] 慕彩红, 焦李成, 刘逸. *M*-精英协同进化数值优化算法[J]. 软件学报, 2009, 20(11): 2925-2938.
MU C H, JIAO L C, LIU Y. *M*-elite coevolutionary algorithm for numerical optimization[J]. Journal of Software, 2009, 20(11): 2925-2938.

[17] 慕彩红, 焦李成, 刘逸. *M*-精英协同进化算法及其在 V-BLAST 系统中的应用[J]. 电子与信息学报, 2009, 31(10): 2443-2448.
MU C H, JIAO L C, LIU Y. *M*-elitist evolutionary algorithm and its application to v-blast system[J]. Journal of Electronics & Information Technology, 2009, 31(10): 2443-2448.

[18] 秦进, 李歆. 一种简单的资源受限的群体演化模型[J]. 复杂系统与复杂性科学, 2009, 6(2): 82-86.
QIN J, LI X. A simple model of population evolution with limited resource[J]. Complex Systems and Complexity Science, 2009, 6(2): 82-86.

[19] 范朝冬, 欧阳红林, 肖乐意. 基于空间截面投影的 Otsu 图像分割算法[J]. 通信学报, 2014, 35(5): 70-78.
FAN C D, OUYANG H L, XIAO L Y. Otsu thresholding method based on projection of cross section for image segmentation[J]. Journal on Communications, 2014, 35(5): 70-78.

[20] 杜海峰, 公茂果, 焦李成等. 用于高维函数优化的免疫记忆克隆规划算法[J]. 自然科学进展, 2004, 14(8): 925-933.
DU H F, GONG M G, LIAO L C, *et al*. Immune memory clonal programming algorithm for high dimension function optimization[J]. Progress in Natural Science, 2004, 14(8): 925-93.

作者简介：

范朝冬（1984-），男，江西宜春人，博士，湘潭大学讲师，主要研究方向为智能信息处理、智能电网、模式识别与人工智能等。

章兢（1957-），男，湖南韶山人，博士，湘潭大学教授、博士生导师，主要研究方向为智能计算、复杂系统控制等。

易灵芝（1966-），女，湖南宁乡人，湘潭大学教授、硕士生导师，主要研究方向为交流调速与电力电子装置、智能电网、新能源发电等。