

## 基于概率分布的动态矩阵演化算法

谭 阳<sup>1,2</sup>, 方 颂<sup>2</sup>, 陈 琳<sup>2</sup>

(1. 湖南师范大学 数学与计算机科学学院, 长沙 410081; 2. 湖南广播电视大学 网络技术系, 长沙 410004)

**摘 要:** 传统演化算法通常以宏观层面的种群之间或个体之间的相互作用来进行协同演化, 较少考虑个体基因编码在微观层面进行局部优化时的相互作用。针对该情况, 提出基于种群基因分布结构的动态矩阵演化算法。利用二进制基因矩阵的方式构建种群个体, 结合基因编码差异及适应度评价种群个体, 通过对比种群基因列决定个体基因结构调整的位置, 并根据优势种群的基因结构产生下一代个体, 通过微观层面上基因位之间的协同作用引导种群的演化。实验结果表明, 该算法对于中、高维函数均表现出良好的优化性能, 同时能较好地平衡宏观全局优化和微观局部优化之间的关系。

**关键词:** 演化算法; 基因编码; 矩阵; 概率分布; 全局优化

**中文引用格式:** 谭 阳, 方 颂, 陈 琳. 基于概率分布的动态矩阵演化算法[J]. 计算机工程, 2017, 43(12): 147-154.

**英文引用格式:** TAN Yang, FANG Song, CHEN Lin. Dynamic Matrix Evolutionary Algorithm Based on Probability Distribution[J]. Computer Engineering, 2017, 43(12): 147-154.

## Dynamic Matrix Evolutionary Algorithm Based on Probability Distribution

TAN Yang<sup>1,2</sup>, FANG Song<sup>2</sup>, CHEN Lin<sup>2</sup>

(1. College of Mathematics and Computer Science, Hunan Normal University, Changsha 410081, China;

2. Department of Network Technology, Hunan Radio and TV University, Changsha 410004, China)

**[Abstract]** Traditional evolutionary algorithms usually use the macro-level interaction between individuals or populations to co-evolution, with less consideration in the individual gene encoding in micro level when interacting local optimization. Aiming at this situation, this paper proposes Dynamic Matrix Evolutionary Algorithm (DMEA) based on population gene distribution structure. It uses binary gene matrix to construct individual genes. It evaluates population individuals according to the difference of genes encoding and fitness and decide the adjustment position of individual gene structure by comparing the population gene series. According to structure of the dominant population, it generates the next generation of individuals and guides the evolution of the population by the synergistic effect of micro-level gene location. Experimental results show that the proposed algorithm has good performance in the optimization of mid-dimensional or high-dimensional functions, and it also can better balance the relationship between the microscopic optimization and the macroscopic optimization.

**[Key words]** evolutionary algorithm; gene encoding; matrix; probability distribution; global optimization

**DOI:** 10.3969/j.issn.1000-3428.2017.12.028

### 0 概述

演化算法是一种模拟生物进化中遗传选择和自然淘汰过程的计算模型<sup>[1-2]</sup>, 其设计思想源于生物遗传学和适者生存的自然规律, 通过采用“生存 + 检测”的迭代过程进行寻优搜索。演化算法具有全局搜索性和与问题无关性等特点, 因此, 该算法自提出以来在工程设计、机器学习、模式识别等方面得到了广泛的应用<sup>[3-4]</sup>, 其中遗传算法<sup>[5]</sup> (Genetic Algorithm, GA)、粒子群优化 (Particle Swarm Optimization, PSO)

算法<sup>[6]</sup>和分布估计算法<sup>[7]</sup> (Estimation of Distribution Algorithm, EDA) 是 3 种比较具有代表性的演化算法。遗传算法是产生最早、影响最大的一种演化算法, 它采用适者生存理念对种群中的个体进行优胜劣汰, 强调个体之间的比较和竞争, 特点是全局优化性能较好; 粒子群优化算法是模仿昆虫、鸟群和鱼群等的群集行为的仿生演化算法, 其中种群个体按照一种合作的方式进行觅食 (寻优), 强调个体间的协作及经验传递, 特点是收敛速度较快; 分布估计算法是一种通过采样和统计学习来预测搜索空间中解分布情

**基金项目:** 国家自然科学基金 (10971060); 湖南省教育厅科研项目 (14C0781, 15C0928)。

**作者简介:** 谭 阳 (1979—), 男, 副教授, 主研方向为智能计算; 方 颂, 高级工程师; 陈 琳, 讲师。

**收稿日期:** 2016-10-24 **修回日期:** 2016-12-02 **E-mail:** TanYang334@126.com

况的演化算法,由于算法以拟合的方式在搜索空间中进行演化,因此其特点是兼具较好的全局搜索能力和收敛速度。

演化算法通常会考虑种群与种群之间或个体与个体间的相互协同作用,较少涉及演化个体基因位之间的相互协同作用,以及在当前种群环境下由全部基因位所构成的基因编码环境,更少涉及整体基因编码环境与个体基因位之间的相互影响以及对种群演化的作用。文献[8]对种群个体的二进制编码差异进行评价,并将评价结果作为计算种群个体适应度的权系数,有效地维护了算法种群的多样性。文献[9]将算法中的个体分成“家族”化的小群体环境,通过比较个体在小环境中的差异来稳定个体的搜索区域,使得“家族”成员之间不产生冲突,从而使算法具备更高的寻优精度。文献[10]将非渐近随机矩阵理论应用于EDA中,使得算法可以对解空间结构进行随机投影,从而避免陷入高维解空间大量且复杂的计算,能有效地对大规模高维空间进行连续的全局优化。

在上述研究的基础上,本文提出基于种群基因分布结构的动态矩阵演化算法(Dynamic Matrix Evolutionary Algorithm, DMEA)。在种群个体二进制基因编码的基础上,以基因位协同作用的方式引导种群的演化,对种群中的个体采用横向对比相同基因位的方法分析种群基因结构的分布情况,以此找到趋近编码一致的特定基因位置,并依据该基因位置的编码分布进行相应的编码调整。

## 1 基于种群基因分布结构的动态矩阵演化算法

不失一般性,若演化算法的种群为 $N$ ,种群规模为 $n$ ,采用固定长度为 $L$ 的二进制对个体进行编码,则个体空间 $B = \{0, 1\}^L$ 。种群 $N = \{a_1, a_2, \dots, a_n\}$  ( $N \subseteq B$ )为一有限点的集合,分别记 $N$ 中的个体为 $a_1, a_2, \dots, a_n$ ,其中任意个体 $a_i = \{a_{i,1}, a_{i,2}, \dots, a_{i,L}\}$  ( $i = 1, 2, \dots, n$ )为空间 $B$ 中的一个点, $a_{i,l} \in \{0, 1\}$  ( $l = 1, 2, \dots, L$ )为第 $i$ 个个体的第 $l$ 个基因编码位。

设演化算法的最大迭代数为 $Gen$ ,则第 $t$  ( $t = 1, 2, \dots, Gen$ )代的种群为 $N_t$ 。若依据目标函数计算 $N_t$ 中所有个体的适应值,并以适应值对个体进行排序(降序),则得到大小为 $n \times L$ 的二进制矩阵 $A_t = (a_{i,j})_{n \times L}$  ( $a_{i,j} \in \{0, 1\}$ )称为第 $t$ 代基因矩阵。

$$A_t = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,L} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,L} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,L} \end{bmatrix} \quad (1)$$

### 1.1 算法流程

综上所述,基于概率分布的动态矩阵演化算法的流程如图1所示。

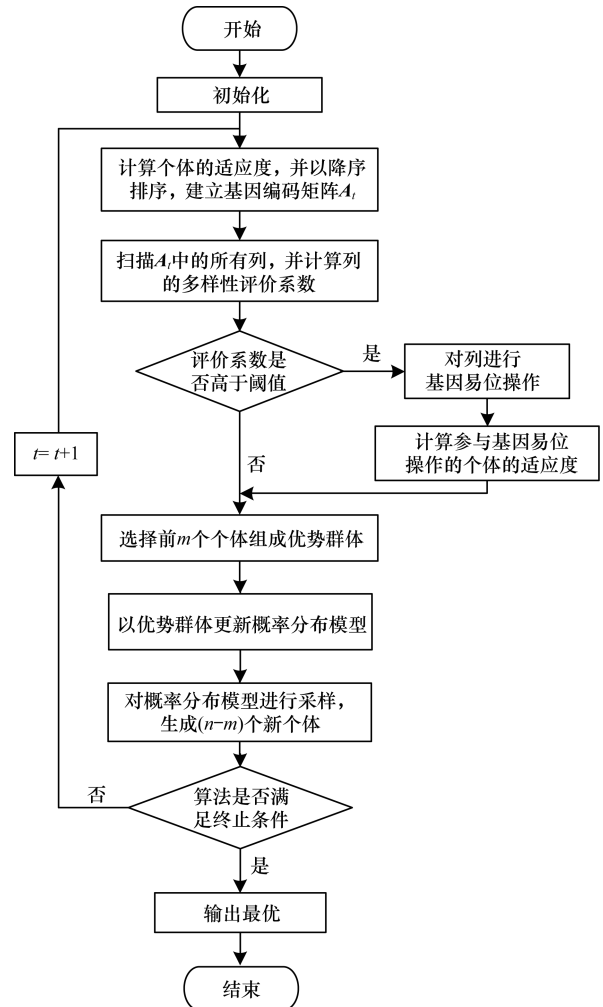


图1 DMEA的算法流程

### 1.2 个体基因的同质化评价

在采用二进制编码的演化算法中,若个体编码趋近于一致,则演化算法会表现出早熟收敛的现象。这里将基因矩阵 $A_t$ 中个体编码趋近于一致的现象称为种群个体基因结构同质化,具体表现为基因矩阵 $A_t$ 中部分列值完全相同。为了能够对 $A_t$ 中任意列 $j$  ( $j = 1, 2, \dots, L$ )的同质化程度进行度量,本文通过式(2)对列 $j$ 进行计算,获取第 $j$ 列的0-1均衡差异值 $\alpha_j$ ,并将其记作列 $j$ 的同质化评价系数。

$$\alpha_j = \frac{\left| \sum_{i=1}^n a_{i,j} - \sum_{i=1}^n (1 - a_{i,j}) \right|}{n}, \alpha_j \in [0, 1] \quad (2)$$

$\alpha_j$ 的值越小则表示该列中的0-1分布数量相对均衡,种群在第 $j$ 个基因编码位的0-1分布较为均衡;反之, $\alpha_j$ 越大则表明该基因编码位的0-1分布偏差越大,同质化程度越高。

### 1.3 基因易位操作

设 $m$  ( $m < n$ )为优势种群的大小,则 $t$ 代基因矩阵 $A_t$ 易位操作示意图如图2所示。

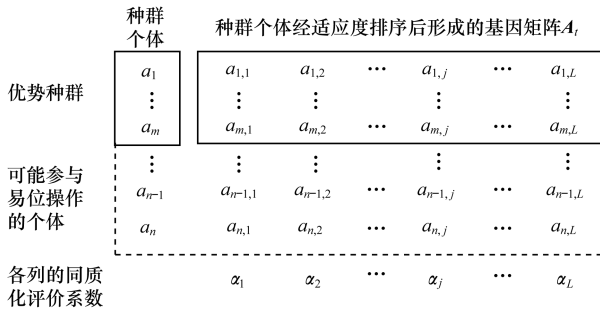


图 2 基因矩阵易位示意图

给定易位阈值  $k$ ,若  $A_t$  中的列  $j$  存在  $\alpha_j > k$ ,则调整列  $j$  中的基因编码以均衡该列的 0-1 分布偏差,并以列  $j$  的评价系数  $\alpha_j$  计算基因易位比例  $\alpha_{jk}$ 。

$$\alpha_{jk} = 2\alpha_j - 1.05 \quad (3)$$

由式(3)可知列  $j$  同质化程度越高其基因位参与易位的比例越大,且该式使得至少 5% 的精英个体被隔离,不参与易位操作,从而确保演化算法不发生退化。确定  $\alpha_{jk}$  后,以基因位行值的降序  $(a_{n,j}, a_{n-1,j}, \dots, a_{\lfloor n \times \alpha_{jk} \rfloor, j})$  对列  $j$  通过式(4)进行基因位易位操作,调整列  $j$  中的基因编码位。

$$a_{x,j} = \begin{cases} 1, & \sum_{i=1}^n a_{i,j} < \sum_{i=1}^n (1 - a_{i,j}) \\ 0, & \sum_{i=1}^n a_{i,j} > \sum_{i=1}^n (1 - a_{i,j}) \end{cases}$$

$$x = n, n-1, \dots, \lfloor n \times \alpha_{jk} \rfloor \quad (4)$$

### 1.4 列的基因易位算子

在矩阵  $A_t$  中,列  $j$  由基因位  $a_{1,j}, a_{2,j}, \dots, a_{n,j}$  组成,并且可知行值越大的基因位属于适应度越低的个体,因此,当存在  $\alpha_j > k$  时,对列  $j$  基因位进行易位操作不但需要以行值的降序进行,而且还需要确定列  $j$  的易位操作范围。设范围系数为  $\beta$ ,列  $j$  的范围系数为  $\beta_j$ ,且  $\beta_j = \lfloor n \times \alpha_{jk} \rfloor$ ,则列  $j$  的基因易位顺序为:  $a_{n,j}, a_{n-1,j}, \dots, a_{\beta_j,j}$ 。

基因列易位算子的具体流程如下:

**步骤 1** 初始化列  $j$  的基因易位概率  $\rho_j, \rho_j = \alpha_{jk}$ , 基因易位位置  $q, q = n$ 。

**步骤 2** 判断条件  $(\rho_j \geq 0) \cap (q \geq n - \beta_j)$  是否成立,成立则进行下一步,否则退出基因易位操作算子。

**步骤 3** 以概率  $\rho_j$  对基因  $a_{q,j}$  通过式(4)进行易位,并判断基因  $a_{q,j}$  是否参与易位,是则  $\rho_j = \rho_j - \frac{1}{n}$ , 否则  $\rho_j$  维持不变。

**步骤 4**  $q = q - 1$ ,转到步骤 2。

### 1.5 算法种群的更新

选取种群  $N_t$  中适应度最优的前  $m$  个个体组成

优势种群  $N_t^m$  来构建概率模型。向量  $P(x)$  为概率模型,  $P(x) = [p(x_1), p(x_2), \dots, p(x_L)]$ , 其中  $p(x_i)$  表示新个体在第  $i$  个基因位置上取“1”的概率。设第  $t$  代概率向量为  $P_t(x), P_t(x) = [p_t(x_1), p_t(x_2), \dots, p_t(x_L)]$ , 则第  $t+1$  代向量可以通过下式进行更新:

$$p_{t+1}(x_i) = (1 - \delta)p_t(x_i) + \delta \prod_{j=1}^w \left( \frac{1}{m} \sum_{i=1}^m x_i^t \right) \quad (5)$$

其中,  $\delta (0 < \delta < 1)$  为机器学习速率,  $w$  为待优化问题的维数。在初始概率向量  $P_0(x)$  中,  $p(x_i) = 0.5 (i = 1, 2, \dots, L)$ 。随着算法的迭代,第  $(t+1)$  代中的新个体由概率向量  $P_t(x)$  随机产生。

## 2 DMEA 算法分析

### 2.1 计算复杂度分析

通常演化算法的寻优过程都比较复杂,难以采用精确的模型进行计算复杂度的计算。本文通过计算 DMEA 中关键步骤的计算复杂度,以分别确定各个步骤的计算复杂度;再以累加方式来推算 DMEA 整体的计算复杂度。在 DMEA 算法中,对基因矩阵进行排序计算复杂度为  $O(n \lg n)$ ,用于计算列评价系数的复杂度为  $O(L \sum_{j=1}^n L) \approx O(L^2)$ 。基因易位操作以阈值  $k$  来确定范围,  $O(L \times n \times k) \approx O(Ln)$ 。选择优势个体  $m$  更新概率分布模型的计算复杂度为  $O(L \times n \times \lg m) \approx O(Ln \lg n)$ ;种群更新  $(n - m)$  个新个体的计算复杂度为  $O(L \times (n - m)) \approx O(Ln)$ 。通过对 DEMA 的分析,可知其在标准 EDA 的基础上增加了列评价系数及基因易位操作 2 个算子,其计算复杂度都不大。因此, DMEA 的总体计算复杂度是在标准 EDA 的基础上线性增加的。

### 2.2 收敛性分析

由上文可知 DMEA 的运行过程是有限齐次马尔科夫链,且任意时刻都符合  $t \geq 0, f(N_{t+1}) \geq f(N_t)$ 。

**定义 1** 设  $M$  为规模为  $n$  的种群组合的集合, 则  $|M| = \begin{bmatrix} 2^t + n - 1 \\ n \end{bmatrix}$ 。

**定义 2** 设在第  $t$  代的种群序列为  $X_t, X_t = \{x_1, x_2, \dots, x_n\}$ , 令下式存在:

$$M^* = \{x \mid \max_{x \in X_t} f(x) = f^*\} \quad (6)$$

其中,  $f^*$  为全局最优值,  $M^*$  为最优解集。

**定义 3** 设  $f_t = \max_{x_i \in X_t} \{f(x_i) : i = 1, 2, \dots, n\}$  为随机变量序列,其中变量代表在第  $t$  代中的最佳适应度值,若对于任意给定的  $\varepsilon > 0$  均满足:

$$\lim_{t \rightarrow \infty} P \{ |f_t - f^*| \leq \varepsilon \} = 1 \quad (7)$$

则表明该算法具备全局收敛性。

**定理** DMEA 具备全局收敛性

证明: 设  $M_i \in M$ , 表示  $M_i$  是  $M$  中的第  $i$  个状态,  $I$  为全集, 设  $p_i(t)$  为种群  $X_i$  处于  $M_i$  状态的概率,  $p_i = \sum_{i \notin I} p_i(t)$ , 则由马尔科夫链的性质可以得到:

$$\begin{aligned} p_{i+1} &= \sum_{M_i \in M_j \notin I} p_i(t) p_{ij}(t) \\ &= \sum_{i \in I_j \notin I} p_i(t) p_{ij}(t) + \sum_{i \notin I_j \in I} p_i(t) p_{ij}(t) \\ &\quad - \sum_{i \notin I_j \in I} p_i(t) p_{ij}(t) + \sum_{i \notin I_j \in I} p_i(t) p_{ij}(t) \end{aligned} \quad (8)$$

又知:

$$p_i = \sum_{i \notin I_j \notin I} p_i(t) p_{ij}(t) + \sum_{i \notin I_j \in I} p_i(t) p_{ij}(t) \quad (9)$$

将式(9)代入式(8)后可得式(10), 由此可得式(11)。

$$\begin{aligned} p_{i+1} &= \sum_{i \in I_j \notin I} p_i(t) p_{ij}(t) \\ &\quad - \sum_{i \notin I_j \in I} p_i(t) p_{ij}(t) + p_i = p_i \end{aligned} \quad (10)$$

$$\lim_{t \rightarrow \infty} p_i = 0 \quad (11)$$

由式(12)和式(13), 可得到式(14)。

$$\lim_{t \rightarrow \infty} P \{ |f_t - f^*| \leq \varepsilon \} = 1 - \lim_{t \rightarrow \infty} \sum_{i \notin I} p_i(t) \quad (12)$$

$$\lim_{t \rightarrow \infty} \sum_{i \notin I} p_i(t) = \lim_{t \rightarrow \infty} p_i \quad (13)$$

$$\lim_{t \rightarrow \infty} P \{ |f_t - f^*| \leq \varepsilon \} = 1 \quad (14)$$

因此, DMEA 具备全局收敛性。

证毕。

### 3 仿真测试及分析

为测试 DMEA 算法的性能, 仿真实验选取遗传算法(GA)、粒子群算法(PSO)和分布估计算法(EDA)3种传统的演化算法, 以及预测轨迹的遗传算法(TPGA)<sup>[11]</sup>、强化引导的粒子群算法(ELPSO)<sup>[12]</sup>和玻尔兹曼单变量边缘分布算法(BEDA)<sup>[13]</sup>, 后3种寻优算法属于近期其他学者对遗传算法、粒子群算法和分布估计算法的改进。为尽量获取一致的数据, 笔者采用原文献中所定义参数用C语言重写了所有算法, 因此, 不可避免地与原算法存在一定的计算偏差。

#### 3.1 函数优化实验

函数优化实验是测试寻优算法搜索性能的必备实验, 通过搜寻测试函数的最小值可以评价寻优算法的性能。本文选取的这6个函数具有较强的代表性<sup>[14]</sup>, 且最优值均为0。测试所用的函数包括单峰函数  $f_1$ 、多峰函数  $f_2 \sim f_4$ 、变换旋转函数  $f_5, f_6$ , 如表1所示。

表1 用于优化仿真测试的函数 (Optimal = 0)

函数	寻优空间
$f_1 = \sum_{i=1}^w [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$x_i \in [-100, 100]$
$f_2 = \sum_{i=1}^w [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$x_i \in [-5.12, 5.12]$
$f_3 = \frac{1}{4000} \sum_{i=1}^w x_i^2 - \prod_{i=1}^w \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$
$f_4 = 418.9829w + \sum_{i=1}^w [x_i \sin(\sqrt{ x_i })]$	$x_i \in [-500, 500]$
$f_5 = \sum_{i=1}^w z_i^2 + f\_bias_1$	$x_i \in [-100, 100]$
$z = x - o, o = [o_1, o_2, \dots, o_{30}]$	
$f_6 = \sum_{i=1}^w [z_i^2 - 10 \cos(2\pi z_i) + 10] + f\_bias_9$	$x_i \in [-5, 5]$
$z = x - o, o = [o_1, o_2, \dots, o_n]$	

对 DMEA 算法的参数设定为: 种群规模  $N = 200$ , 优势种群  $m = \lceil 0.05N \rceil$ , 机器学习速率  $\delta = 0.05$ , 最大迭代数为  $Gen = 500$ 。设定同质化评价阈值  $k$ , 以仿真测试函数  $f_1$  为例进行分析, 表2显示了  $k$  取不同值时 DMEA 分别运行30次对  $f_1$  ( $w = 10$ ) 优化的结果。可以看出, 当阈值  $k$  的取值为0.75时, 既能保证算法具备较好的寻优精度, 又能满足适中的运行速度。因此, 综合考虑确定阈值  $k = 0.75$ 。测试环境: CPU 为 Core i7 2.2 GHz, RAM 为 8 GB, 系统环境为 Ubuntu 15.10。

表2 不同  $k$  值下 DMEA 的优化情况

$k$ 值	优化均值	方差	耗时/s
0.60	1.68 e-12	4.50 e-13	0.856
0.70	3.77 e-11	2.41 e-12	0.601
0.75	3.31 e-11	3.79 e-12	0.448
0.80	7.83 e-9	3.01 e-10	0.409
0.85	6.43 e-9	1.47 e-9	0.334
0.90	1.93 e-8	6.76 e-9	0.225

此处约定所有寻优算法单次最大迭代次数为500, 且算法终止的条件为: 达到最大迭代次数或  $Best\_N > 20$ , 其中  $Best\_N$  表示连续的第  $t_{i+1}$  代与第  $t_i$  代种群最优值之差为0的次数。每种对比算法分别用0~49作为随机数种子独立优化50次, 表3给出了对比算法在函数优化时的情况。可以看出, DMEA 除在函数  $f_5$  外均取得了更好的寻优结果, 这表明 DMEA 中的列评价系数与染色体易位算子可以更均匀地在解空间中散布种群个体, 扩大搜索空间, 同时精英个体的保留机制可以强化优势个体在解空间中的引导作用, 提高收敛速度。在与其他类型算法的比较中, DMEA 在对函数  $f_3, f_4, f_6$  的优化上取得了比其他对比算法更好的优化结果, 在函数  $f_1, f_2, f_5$  上 DMEA 与其他对比算法的优化结果的偏差控制在  $e-8$  以内, 总体上表现出较强的寻优性能和稳定性。

表 3 各算法的函数优化性能比较 ( $w = 10$ )

函数	比较项目	GA	PSO	EDA	TPGA	ELPSO	BEDA	DMEA
$f_1$	均值	8.501 e-5	2.419 e-8	4.796 e-6	3.008 e-10	5.474 e-13	6.761 e-11	3.441 e-11
	方差	5.683 e-7	4.779 e-10	2.311 e-8	7.830 e-13	6.436 e-15	1.934 e-13	9.732 e-12
$f_2$	均值	3.461 e-4	7.880 e-6	2.729 e-5	4.037 e-8	5.318 e-12	9.108 e-8	6.418 e-10
	方差	4.210 e-5	3.348 e-8	2.040 e-8	7.571 e-11	7.673 e-14	5.001 e-9	1.691 e-12
$f_3$	均值	1.047 e-4	1.019 e-10	1.029 e-8	1.829 e-5	4.797 e-10	9.736 e-9	1.019 e-11
	方差	3.478 e-6	7.471 e-11	3.376 e-10	6.347 e-7	3.977 e-13	8.321 e-12	3.373 e-12
$f_4$	均值	5.475 e-4	7.721 e-10	3.657 e-5	3.659 e-13	2.478 e-13	6.609 e-10	1.843 e-13
	方差	2.626 e-6	3.681 e-11	4.776 e-6	1.736 e-15	3.450 e-15	2.760 e-11	2.375 e-14
$f_5$	均值	6.375 e-4	4.990 e-6	9.923 e-7	3.705 e-9	9.879 e-10	4.381 e-10	4.105 e-9
	方差	3.461 e-6	7.880 e-8	2.729 e-9	4.037 e-10	4.773 e-13	5.318 e-11	9.108 e-11
$f_6$	均值	9.453 e-2	1.469 e-6	6.826 e-7	5.694 e-7	3.684 e-11	2.437 e-12	6.453 e-13
	方差	1.293 e-5	4.271 e-9	2.340 e-10	7.709 e-8	3.991 e-14	4.344 e-14	4.357 e-15

通常优化算法在高维空间中会因为维数的增加,导致解空间的急剧膨胀,使得求解难度大幅增加。为进一步比较寻优算法在高维空间中的性能,分别将待优化函数的维数  $w$  增加到 50 和 100,以分析算法在高维情况下的寻优性能,实验数据如表 4 所示。可以看

出,随着问题维数的增加,所有的优化算法的优化精度均出现不同程度的下降。当  $w = 50$  时,DMEA 在函数  $f_1 \sim f_4$  上取得了优于其他对比算法的结果;当  $w = 100$  时,DMEA 在所有函数的优化上均优于其他对比算法,表现出较强的在高维空间中的寻优能力。

表 4 各算法在不同维度上的优化性能

函数	维数	比较项目	TPGA	ELPSO	BEDA	DMEA
$f_1$	$w = 50$	均值	5.624	3.025	3.642	2.024
		方差	1.731	0.913	1.015	0.701
	$w = 100$	均值	22.633	21.577	23.866	17.648
		方差	11.214	9.944	10.176	5.439
$f_2$	$w = 50$	均值	12.034	10.934	11.682	10.108
		方差	2.257	2.189	1.297	1.409
	$w = 100$	均值	25.255	21.871	19.252	12.531
		方差	7.307	8.177	5.304	2.978
$f_3$	$w = 50$	均值	0.017	0.002	0.009	0.007
		方差	0.008	0.004	0.003	0.002
	$w = 100$	均值	7.686	0.451	1.145	0.146
		方差	0.794	0.371	0.931	0.017
$f_4$	$w = 50$	均值	21.436	13.133	24.251	15.145
		方差	6.206	5.081	7.147	6.021
	$w = 100$	均值	76.330	33.228	29.703	28.086
		方差	9.478	9.036	10.793	8.328
$f_5$	$w = 50$	均值	5.837	2.009	4.810	1.137
		方差	1.636	0.789	1.015	0.063
	$w = 100$	均值	51.812	77.264	96.754	47.619
		方差	6.303	5.248	8.347	5.568
$f_6$	$w = 50$	均值	7.956	5.893	3.699	2.547
		方差	0.947	1.004	1.120	0.833
	$w = 100$	均值	40.036	44.175	51.235	39.062
		方差	5.004	6.143	7.791	5.416

通过图3和图4也可以看出,在高维情况下DMEA相较于其他对比算法具有更好的优化精度,同时也具备较好的收敛速度。

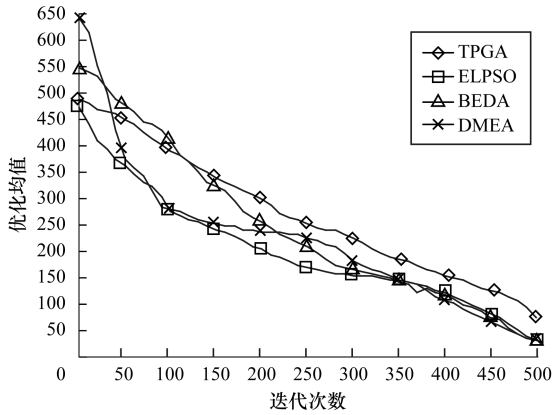


图3 各算法在函数  $f_4 (w=100)$  上的仿真结果比较

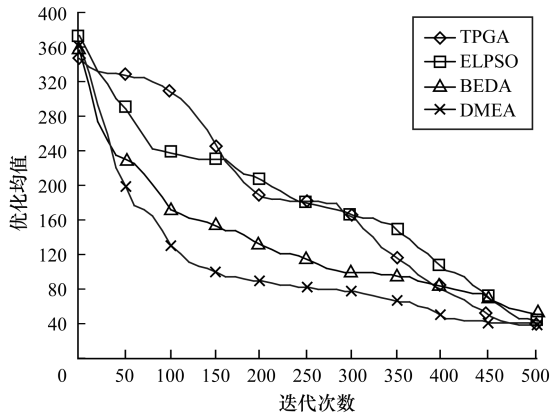


图4 各算法在函数  $f_6 (w=100)$  上的仿真结果比较

### 3.2 算法种群的同质化测试

采用简单函数  $f_7 (f_7 = \sum_{i=1}^w x_i^2, x_i \in [-100, 100])$  进行算法种群的同质化测试。函数  $f_7$  极值明显且单调区间巨大,优化算法极容易受到极值的吸引产生早熟收敛,使种群个体出现基因结构同质化,从而丧失多样性。

表5显示了对比算法在不同维度下优化函数  $f_7$  的同质化评价均值。可以看出,DMEA的同质化评价均值在3种不同维度下均低于设定的阈值0.75,这也表明DMEA的翻转易位算子是有效的,能够有效避免演化算法同质化现象的出现。

表5 对比算法在不同维度下优化函数  $f_7$  的同质化评价

算法	w = 10	w = 50	w = 100
TPGA	0.873	0.598	0.590
ELPSO	0.850	0.558	0.551
BEDA	0.825	0.572	0.575
DMEA	0.648	0.496	0.495

分别对不同维度下的函数进行测试,结果如图5~图7所示。可以看出,DMEA在不同维度下的寻优过程中,其同质化评价低于其他对比算法。

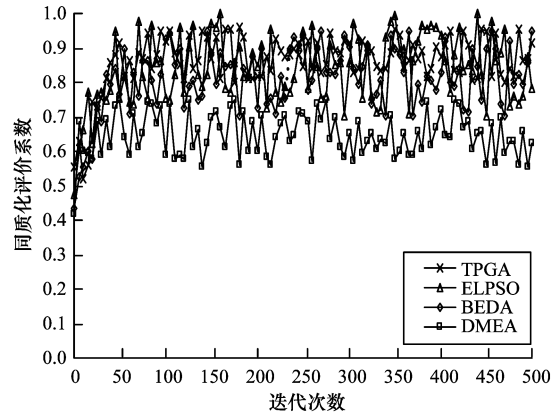


图5 各算法在函数  $f_7 (w=10)$  上的同质化评价

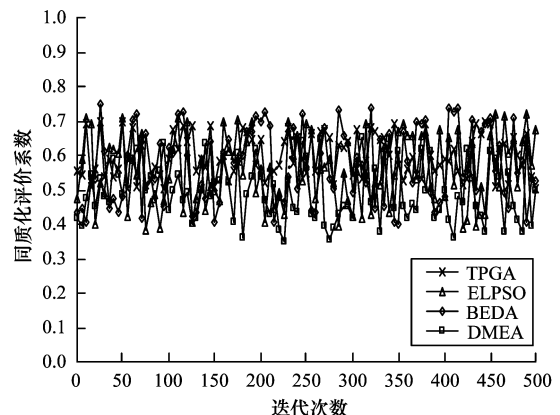


图6 各算法在函数  $f_7 (w=50)$  上的同质化评价

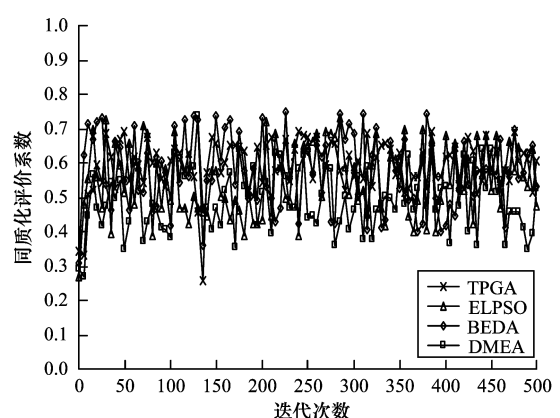


图7 各算法在函数  $f_7 (w=100)$  上的同质化评价

### 3.3 针对多维多选择背包问题的仿真实验

为能更好地测试寻优算法的局部优化性能,选择多维多选择背包问题(Multidimensional Multi-choice Knapsack Problem, MMKP)对寻优算法进行仿真测试。MMKP是经典的NP难题问题,其优化的主要困难之处在于众多约束条件的限制,通常要求优化算法兼具很强

的全局和局部搜索的能力,可以较为全面地测试优化算法的寻优能力。测试实例来源于国际标准测试集 MMKPLIB 实例库<sup>[15]</sup>中的经典算例 I01 ~ I06。实验结果如表 6 所示。其中:算例名称后括号内数据为已知最优值; $t_{Best}$ 为算法首次获得最佳值所需的迭代次数。可

以看出,DMEA 在对 6 个 MMKP 问题的优化上均能获得最优解。在寻优速度上,除在实例 I01 和 I02 略逊于 BEDA 之外,优于其他 5 种算法。

从图 8 和图 9 也可以看出,DMEA 具有更好的寻优速度,能够较早地收敛并稳定在最优解上。

表 6 各算法在多维多选择背包问题上的优化性能

算法	比较项目	I01 (173)	I02 (364)	I03 (602)	I04 (597)	I05 (3 905)	I06 (4 799)
GA	优化均值	131	297	1 410	3 074	3 401	4 137
	最优值	156	337	1 530	3 362	3 610	4 459
	$t_{Best}$	247	231	318	324	343	391
PSO	优化均值	127	273	1 273	3 181	3 327	4 089
	最优值	148	318	1 488	3 370	3 585	4 448
	$t_{Best}$	107	113	234	218	308	273
EDA	优化均值	151	327	1 473	3 173	3 340	4 387
	最优值	154	344	1 524	3 384	3 601	4 467
	$t_{Best}$	11	47	94	157	64	143
TPGA	优化均值	170	352	1 560	3 375	3 535	4 126
	最优值	173	364	1 602	3 597	3 905	4 783
	$t_{Best}$	143	146	173	167	157	158
ELPSO	优化均值	157	330	1 468	3 184	3 352	4 114
	最优值	171	348	1 521	3 407	3 597	4 473
	$t_{Best}$	14	57	77	87	107	121
BEDA	优化均值	168	341	1 436	3 346	3 527	4 247
	最优值	173	364	1 597	3 514	3 905	4 623
	$t_{Best}$	15	43	104	117	121	147
DMEA	优化均值	171	359	1 563	3 474	3 635	4 326
	最优值	173	364	1 602	3 597	3 905	4 799
	$t_{Best}$	38	98	127	158	105	234

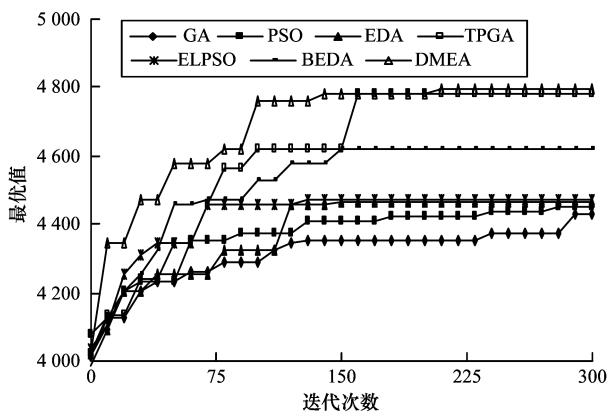


图 8 各算法在实例 I05 上的仿真结果

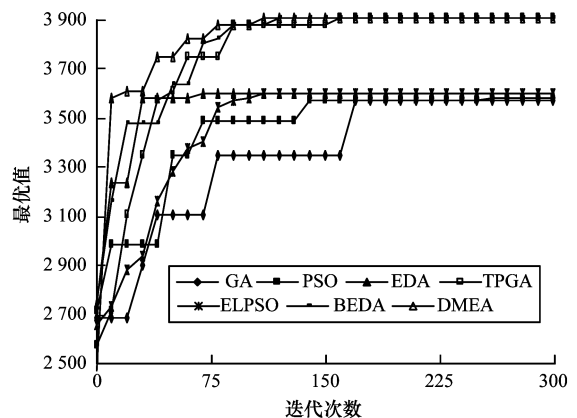


图 9 各算法在实例 I06 上的仿真结果

#### 4 结束语

通过种群基因矩阵可以在各演化个体的基因位之间以分布差异的方式进行比较,以此可构建个体基因位之间的相互协同进化机制。这种机制可以方便地调控种群的基因组成结构,依据演化算法当前

基因矩阵所构成的基因位环境,协调算法全局搜索与局部搜索之间的关系,从而提高演化算法的搜索效率并降低种群的同质化现象。本文提出一种以种群个体二进制编码构建动态基因矩阵的演化算法。通过对基因矩阵中基因列的同质化评价来评价种群构成的多样性,并针对种群个体的特定基因位置做

翻转易位操作,使得算法能够在解空间中进行随机投影。通过对比实验结果表明,DMEA在函数优化和MMKP中均表现出较好的性能。算法种群同质化测试中的数据也表明DMEA具有较强的多维函数优化和解决复杂问题的能力。

#### 参考文献

- [1] 陈皓,崔杜武,崔颖安,等. 族群进化算法[J]. 软件学报,2010,21(5):978-990.
- [2] 刘振,胡云安,彭军. 协同进化扩展紧致量子进化算法[J]. 控制与决策,2014(2):320-326.
- [3] ZHAO Fuqing, SHAO Zhongshi, WANG Junbiao, et al. A Hybrid Optimization Algorithm Based on Chaotic Differential Evolution and Estimation of Distribution[J]. Computational and Applied Mathematics, 2015, 36(1): 1-26.
- [4] 周永权,黄正新,刘洪霞. 求解TSP问题的离散型萤火虫群优化算法[J]. 电子学报,2012,40(6):1164-1170.
- [5] HALDAR A. A Review on Genetic Algorithm[J]. International Journal of Engineering Sciences & Research Technology, 2014, 3(1): 369-372.
- [6] MUSA H, IBRHIM S B. A Review of Particle Swarm Optimization(PSO) Algorithms for Optimal Distributed Generation Placement[J]. International Journal of Energy and Power Engineering, 2015, 4(4): 232-239.
- [7] SANTANA R, MENDIBURU A, LOZANO J A. A Review of Message Passing Algorithms in Estimation of Distribution Algorithms[J]. Natural Computing, 2014, 15(1): 1-16.
- [8] 刘全,王晓燕,傅启明,等. 双精英协同进化遗传算法[J]. 软件学报,2012,23(4):765-775.
- [9] MARTÍNEZ J L F, GONZALO E G. The PSO Family: Deduction, Stochastic Analysis and Comparison[J]. Swarm Intelligence, 2015, 3(4): 245-273.
- [10] KABÁN A, BOOTKRAJANG J, DURRANT R J. Towards Large Scale Continuous EDA: A Random Matrix Theory Perspective[C]//Proceedings of Conference on Genetic and Evolutionary Computation. New York, USA: ACM Press, 2015:383-390.
- [11] GROBA C, SARTAL A, VÁZQUEZ X H. Solving the Dynamic Traveling Salesman Problem Using a Genetic Algorithm with Trajectory Prediction: An Application to Fish Aggregating Devices[J]. Computers & Operations Research, 2015, 56(C): 22-32.
- [12] JORDEHI A R. Enhanced Leader PSO (ELPSO): A New PSO Variant for Solving Global Optimization Problems[J]. Applied Soft Computing, 2015, 26: 401-417.
- [13] VALDEZ S I, HERNÁNDEZ A, BOTELLO S. A Boltzmann Based Estimation of Distribution Algorithm[J]. Information Sciences, 2013, 236: 126-137.
- [14] LIANG J J, RUNARSSON T, MEZURA-MONTES E, et al. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization[EB/OL]. [2016-09-02]. <http://www.lania.mx/~emezura>.
- [15] HIFI M, MICHRAFY M, SBIHI A. Heuristic Algorithms for the Multiple-choice Multidimensional Knapsack Problem[J]. Journal of the Operational Research Society, 2004, 55(12): 1323-1332.
- [3] 王元卓,靳小龙,程学旗. 网络大数据:现状与展望[J]. 计算机学报,2013,36(6):1125-1138.
- [4] 孟小峰,慈祥. 大数据管理:概念、技术与挑战[J]. 计算机研究与发展,2013,50(1):146-169.
- [5] 任丰原,林闯,黄小猛,等. 主动队列管理算法的分类器实现[J]. 电子学报,2004,32(11):1796-1800.
- [6] HOLLOTT V, MISRA V, TOWSLEY D, et al. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows[J]. IEEE Transactions on Automatic Control, 2002, 47(6): 945-959.
- [7] 任丰原,林闯,王福豹. RED算法的稳定性:基于非线性控制理论的分析[J]. 计算机学报,2002,25(12): 1302-1307.
- [8] 何阳,周井泉. 一种改进的RED主动队列管理算法[J]. 计算机技术与发展,2013,23(8):111-115.
- [9] CHEN Weifeng, LIAN Fenhuang, CHENG Xu, et al. Congestion Control Scheme Performance Analysis Based on Non-linear RED[J]. IEEE Systems Journal, 2015(99): 1-8.
- [10] PATEL M. URED: Upper Threshold RED an Efficient Congestion Control Algorithm[C]//Proceedings of the 4th International Conference on Computing, Communications and Networking Technologies. Washington D. C., USA: IEEE Press, 2014: 1-5.
- [11] OTTT J, LAKSHMAN T V, WONG L H. SRED: Stabilized RED[C]//Proceedings of INFOCOM'99. Washington D. C., USA: IEEE Press, 1999: 1346-1355.
- [12] HOLLOTT V, MISRA V, TOWSLEY D, et al. On Designing Improved Controllers for AQM Routers Supporting TCP Flows[C]//Proceedings of INFOCOM'01. Washington D. C., USA: IEEE Press, 2001: 1726-1734.
- [13] RICHARD J L, PRIYA R, EYAD H A. Analysis of Adaptive Random Early Detection (ARED)[J]. Teletraffic Science & Engineering, 1993, 51(8): 1271-1280.
- [14] 刘明. 面向网络拥塞控制的主动队列管理机制研究[D]. 长沙:国防科学技术大学,2007.
- [15] GONG Y, ROSSI D, TESTA C, et al. Fighting the Bufferbloat: On the Coexistence of AQM and Low Priority Congestion Control[C]//Proceedings of INFOCOM'13. Washington D. C., USA: IEEE Press, 2013: 3291-3296.

编辑 金胡考

编辑 刘冰

(上接第140页)