

文章编号: 1001-0920(2011)09-1333-05

利用 Copula 函数估计概率模型并采样的分布估计算法

王丽芳^{1,2}, 曾建潮², 洪毅¹

(1. 兰州理工大学 电气工程与信息工程学院, 兰州 730050; 2. 太原科技大学 复杂系统计算智能实验室, 太原 030024)

摘要: 将 Copula 理论引入分布估计算法的研究中, 并在估计概率模型时分两个步骤进行: 1) 估计各变量的边缘分布函数; 2) 构造经验 Copula 函数或正态 Copula 函数. 根据 Copula 函数和各边缘分布进行采样, 在简化估计模型运算复杂度的同时, 充分反映了变量之间的关系. 仿真实验验证了该算法的可行性和有效性.

关键词: 进化计算; 分布估计算法; Copula 函数

中图分类号: TP18

文献标识码: A

Estimation of distribution algorithm modeling and sampling by means of Copula

WANG Li-fang^{1,2}, ZENG Jian-chao², HONG Yi¹

(1. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China; 2. Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan 030024, China. Correspondent: WANG Li-fang, E-mail: wlf1001@163.com)

Abstract: Copula theory is introduced into estimation of distribution algorithms(EDA). This algorithm estimates the probability model of selected population in two steps: the first is to estimate the margins of each variable, the second is to construct a empirical copula or a Gaussian Copula. The new population is sampled from the copula and the margins. Thus the computational cost is simplified, which shows the full dependencies of the variables. The experimental results show the feasibility and effectiveness of the proposed algorithm.

Key words: evolutionary computation; estimation of distribution algorithms; Copulas

1 引言

分布估计算法已经成为目前智能计算领域的研究热点^[1-3]. 理论上已经证明, 对于种群规模无穷大的分布估计算法, 如果概率模型能够精确反映选择后群体的分布, 则在采用比例选择、截断选择、两个体锦标赛选择这 3 种选择算子的情况下, 分布估计算法是全局收敛的^[4]. 事实上, 种群规模无穷大以及精确反映群体分布都难以实现. 目前的分布估计算法不同于最初提出的一些分布估计算法^[5-6], 多数考虑了优化问题中变量之间的非线性关系^[7-8]. 如 EMNA^[1]用样本的协方差阵估计变量之间的关系, 对下一代进行采样. 众所周知, 相关系数反映的只是变量之间的线性相关性, 而对于一些相互之间有明显非线性关系的变量, 其相关系数却为 0. 如随机变量 $X \sim N(0, 1)$, $Y = X^2$, 显然 X 与 Y 之间存在着很

强的相关关系, 但其相关系数 $\rho(X, Y) = 0$. 因此, 采用联合正态分布也不能准确反映群体的分布.

在统计学及经济学领域, 存在着对大量数据的分布模型进行精确估计以及由此采样的问题. Sklar 定理的提出, 将多变量联合分布的估计简化为各变量边缘分布的估计与 Copula 函数的确定两部分. 显然单变量分布模型的估计比多变量联合分布的估计要简单的多. 正因如此, Copula 理论成为目前的一个研究热点. 在分布估计算法中同样存在着对优势群体分布情况的估计和根据估计的模型进行采样的问题, 这两个问题也是目前分布估计算法研究的重点. 将 Copula 理论用于分布估计算法将会使算法大大简化, 而且能够更精确地估计优势群体的分布模型, 从而使算法能够更快地收敛于最优解. 目前已有一些研究将二元 Copula 函数用于分布估计算法^[9].

收稿日期: 2010-06-13; 修回日期: 2010-08-09.

基金项目: 山西省青年科学基金项目(2010021017-2); 山西省高校产业化项目(2010015).

作者简介: 王丽芳(1975-), 女, 副教授, 博士生, 从事智能计算的研究; 洪毅(1954-), 男, 教授, 博士生导师, 从事控制理论与控制工程等研究.

2 Copula 理论简介

下面对 Copula 函数的定义和 Sklar 定理进行简单的介绍。

定义 1 n 维 Copula 函数 (或称为 n -Copula)^[10] 是一个函数 C , 具有如下性质:

- 1) 定义域为 $[0,1]^n$, 值域为 $[0,1]$;
- 2) 对于任意 $t = (t_1, \dots, t_n) \in [0,1]^n$, 若至少有一个 $t_k = 0$, 则 $C(t) = 0$;
- 3) 对于定义域中的任意真子集 $B = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$, 有

$$\sum_{\varepsilon_1, \dots, \varepsilon_n \in \{0,1\}} (-1)^{\varepsilon_1 + \dots + \varepsilon_n} C(\varepsilon_1 a_1 + (1 - \varepsilon_1) b_1, \dots, \varepsilon_n a_n + (1 - \varepsilon_n) b_n) \geq 0; \quad (1)$$

- 4) 对于任意 $u \in [0,1]$, C 的边缘函数 C_k 满足

$$C_k(u) = C(1, \dots, 1, u, 1, \dots, 1) = u. \quad (2)$$

定理 1 (Sklar 定理) 令 H 是 D 维联合分布函数, 其边缘分布为 F_1, \dots, F_D , 一定存在一个 D -Copula 函数 C , 对于任意向量 $\mathbf{x} = (x_1, \dots, x_D) \in R^D$, 有

$$H(x_1, \dots, x_D) = C(F_1(x_1), \dots, F_D(x_D)). \quad (3)$$

如果 F_1, \dots, F_D 都是连续的, 则 C 是唯一的; 否则 C 在 $\text{Ran } F_1 \times \dots \times \text{Ran } F_D$ 上是唯一的. 相反, 如果 C 是一个 D -Copula 且 F_1, \dots, F_D 是边缘分布函数, 则由式 (3) 确定的函数 H 是边缘分布为 F_1, \dots, F_D 的 D 维联合分布函数.

根据 Sklar 定理, 随机向量的联合分布可分解成各随机变量的一维边缘分布, 并通过一个 Copula 函数将它们连接起来. 对于难于估计的联合分布模型, 可简化为一维边缘分布模型的估计以及 Copula 函数的产生.

概率模型估计完成之后, 需要做的工作是根据估计的联合分布进行采样. 下面以二元 Copula 函数为例, 简单介绍根据 Copula 函数产生服从估计的联合分布 $C(F(x), G(y))$ 的点 (x, y) 的方法. 设 $U = F(x)$, $V = G(y)$, 则 $U \in [0, 1]$, $V \in [0, 1]$. 由 Sklar 定理, 只需产生服从联合分布 C 的随机向量 $(U, V) \in [0, 1]^2$ 的一组观测值 (u, v) , 然后由 X 和 Y 的边缘分布函数的反函数 $F^{(-1)}(u)$ 和 $G^{(-1)}(v)$ 产生点 (x, y) .

3 多元 Copula 函数的构造

3.1 多元经验 Copula 函数的构造方式

设 D 维随机向量 \mathbf{X} , 其联合分布为 $F(\mathbf{x})$, Copula 函数为 $C(\mathbf{u})$, 各变量的一维边缘分布为 $F_i(x)$, $i = 1, 2, \dots, D$, 则根据 Sklar 定理有 $C(u_1, \dots, u_D) = F(F_1^{-1}(u_1), \dots, F_D^{-1}(u_D))$. 产生一组服从分布 $F(\mathbf{x})$ 的随机数 (x_1, \dots, x_D) 的方法是: 首先产生服从分布

$C(\mathbf{u})$ 的一组随机数 u_1, \dots, u_D ; 然后由各边缘分布函数的反函数产生 (x_1, \dots, x_D) , 其中 $x_i = F_i^{-1}(u_i)$, $i = 1, 2, \dots, D$.

设 Copula 函数 $C(\mathbf{u})$ 相应的密度函数为 $c(\mathbf{u})$, 则边缘密度函数

$$c_d(u) = c_d(u_1, \dots, u_d) = \int_{u_{d+1}=0}^1 \dots \int_{u_D=0}^1 c(\mathbf{u}) du_{d+1} \dots du_D, \quad d = 1, 2, \dots, D-1. \quad (4)$$

显然 $c(u) = c_D(u)$. 为了产生非独立的随机数 u_1, \dots, u_D , 需要知道条件分布函数

$$C_d(u_d | u_1, \dots, u_{d-1}) = P\{U_d \leq u_d | U_1 = u_1, \dots, U_{d-1} = u_{d-1}\} = \frac{\int_{u=0}^{u_d} c_d(u_1, \dots, u_{d-1}, u) du}{c_{d-1}(u_1, \dots, u_{d-1})}, \quad d = 1, 2, \dots, D. \quad (5)$$

文献 [11] 提出了构造经验 Copula 函数的方式, 以及根据该经验 Copula 函数和经验分布函数产生服从原分布的新向量的方式. 现将该文中提出的方法作一简单介绍.

设 D 维随机向量 \mathbf{Z} , 已知其规模为 n 的样本, 用集合表示为 $\mathbf{z} = \{\mathbf{z}_i\} = \{(z_{i1}, \dots, z_{iD}), i = 1, 2, \dots, n\}$. 首先对随机向量的每一维构造其经验分布函数; 然后根据所得的经验分布函数将样本 \mathbf{z}_i 对应到其经验分布函数值, 即可得出向量 \mathbf{u}_i , \mathbf{u}_i 的每维取值在区间 $[0,1]$ 内. 从而由随机向量 \mathbf{Z} 的样本便可得到随机向量 \mathbf{U} 的样本.

由 Copula 理论可知, Copula 函数 $C(\mathbf{u})$ 实际上是随机向量 \mathbf{U} 的联合分布函数. 根据概率论的基本理论, 产生向量 \mathbf{u} 的方式即是根据 \mathbf{U} 的边缘分布计算出相应的条件分布, 然后由条件分布产生向量 \mathbf{u} . 因此, 首先需要估计 \mathbf{U} 的边缘分布, 而 \mathbf{U} 的样本已知, 从而边缘分布易于估计.

具体做法如下:

对于随机向量 \mathbf{Z} 的 n 个样本 \mathbf{z} , 分别按维由小到大进行排序, 设第 $d \in \{1, 2, \dots, D\}$ 维排序后的结果是 $z_{(1)d}, \dots, z_{(n)d}$, 则第 d 维的经验分布函数

$$F_d(z) = \begin{cases} 0, & z < z_{(1)d}; \\ i/n, & z_{(i)d} \leq z < z_{(i+1)d}, \quad i = 1, 2, \dots, n-1; \\ 1, & z_{(n)d} \leq z. \end{cases} \quad (6)$$

由式 (6) 可将 n 个样本 \mathbf{z}_i ($i = 1, 2, \dots, n$) 映射为 n 个向量 $\mathbf{u}_i = (u_{i1}, \dots, u_{iD}) \in \{1/n, 2/n, \dots, 1\}^D$, $i = 1, 2, \dots, n$, 作为空间 $[0,1]^D$ 中的 n 个点. 将区间 $[0,1]$ 分成 K 等份, 记为 S_1, \dots, S_K , 即 $S_i = ((i-1)\delta, i\delta)$,

$\delta = 1/K$. 这样, D 维单位空间 $[0, 1]^D$ 可分解成 K^D 个小空间 $\Delta_i = S_{i_1} \times \dots \times S_{i_D}$, 其中, $\mathbf{i} = (i_1, \dots, i_D) \in \{1, \dots, K\}^D$. 统计每个小空间 Δ_i 中点的个数, 记为 N_i , 则密度函数

$$c(\mathbf{u}) \triangleq f_i = N_i/n/\delta^D. \quad (7)$$

令 $\uparrow u = \max\{1, \lceil uK \rceil\}$, 则 $\mathbf{i} = (\uparrow u_1, \dots, \uparrow u_D) \in \{1, 2, \dots, K\}^D$. 由式(1), 有

$$c_d(u_1, \dots, u_d) = \delta^{D-d} f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}, \quad d = 1, 2, \dots, D-1. \quad (8)$$

其中

$$f_{i_1, \dots, i_d}^{(d)} = \sum_{(i_{d+1}, \dots, i_D)=(1, \dots, 1)}^{(K, \dots, K)} f_{\mathbf{i}}, \quad (i_1, \dots, i_d) = \{1, 2, \dots, K\}^d, \quad (9)$$

则条件分布函数(5)即为

$$C_2(u_2|u_1) = \delta^{D-1} \sum_{i_2=1}^{\downarrow u_2} f_{\uparrow u_1, i_2}^{(2)} + (u_2 - \downarrow u_2 \delta) \delta^{D-2} f_{\uparrow u_1, \uparrow u_2}^{(2)}, \quad (10)$$

$$C_d(u_d|u_1, \dots, u_{d-1}) = \frac{\delta \sum_{i_d=1}^{\downarrow u_d} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)} + (u_d - \downarrow u_d \delta) f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}}{\delta f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)}}, \quad (11)$$

其中 $\downarrow u = \uparrow u - 1$.

文献[11]中给出了计算式(9)中定义的 $\mathbf{f} = \{f_{i_1, \dots, i_d}^{(d)}\}$ 的方法, 具体如下:

算法1 $\mathbf{f} = \text{calculate}(\mathbf{z})$.

Step 1: 根据式(6)将样本点 $\mathbf{z}_i (i = 1, 2, \dots, n)$ 映射为 $[0, 1]^D$ 空间上的点 $\mathbf{u}_i, i = 1, 2, \dots, n$;

Step 2: 令数组 $f^{(d)}$ 中的元素为 0;

Step 3: for ($i = 1 : n$)

for ($d = 1 : D$)

$j_d := \uparrow u_{d,i}$

end

for ($d = 2 : D$)

$f_{j_1, \dots, j_d}^{(d)} := f_{j_1, \dots, j_d}^{(d)} + 1/n/\delta^D$

end

end

根据上述联合分布 C 产生一组随机数的过程如算法2所示.

算法2 $(u_1, \dots, u_D) = \text{generation}(\mathbf{f})$.

Step 1: 在 $[0, 1]$ 区间内按均匀分布产生随机数 u_1 和 u .

Step 2: 根据式(10), 有

$$u - \delta^{D-1} \sum_{i_2=1}^{\downarrow u_2} f_{\uparrow u_1, i_2}^{(2)}, \quad u_2 = \downarrow u_2 \delta + \frac{\delta^{D-2} f_{\uparrow u_1, \uparrow u_2}^{(2)}}{\delta^{D-2} f_{\uparrow u_1, \uparrow u_2}^{(2)}},$$

其中 $\downarrow u_2$ 是满足条件 $uK^{D-1} \leq \sum_{i_2=1}^{\downarrow u_2+1} f_{\uparrow u_1, i_2}^{(2)}$ 的 $\{0, \dots, K-1\}$ 内的最小整数.

Step 3: for ($d = 3 : D$).

Step 3.1: 在 $[0, 1]$ 区间内按均匀分布产生随机数 u ;

Step 3.2: 根据式(11), 有

$$u_d = \downarrow u_d \delta + \delta \frac{u f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)} - \sum_{i_d=1}^{\downarrow u_d} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)}}{f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}},$$

其中 $\downarrow u_d$ 是满足条件

$$u f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)} \leq \sum_{i_d=1}^{\downarrow u_d+1} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)}$$

的 $\{0, \dots, K-1\}$ 内的最小整数.

end

3.2 多元正态 Copula 函数的构造方式

文献[11]中 Strelen 采用的是由边缘经验分布构造经验 Copula 函数的方法, 并且根据经验 Copula 函数和边缘经验分布进行采样模拟原有的分布. 在连续优化问题中, 正态分布广泛使用. 根据 Strelen 的算法可以构造正态 Copula 函数, 唯一不同的是边缘分布采用正态分布, 即在计算 \mathbf{f} 的算法1中变式(6)为式(12), 采用式(12)将样本中各变量映射到 $[0, 1]$ 空间内.

对于随机向量 \mathbf{Z} 的 n 个样本 $\mathbf{z}_i = (z_{i1}, \dots, z_{iD}), i = 1, 2, \dots, n$, 第 $d \in \{1, 2, \dots, D\}$ 维的边缘正态分布函数为

$$F_d(z) = N(\mu_d, \sigma_d^2), \quad (12)$$

其中

$$\mu_d = \frac{1}{n} \sum_{i=1}^n z_{id}, \quad \sigma_d^2 = \frac{1}{n-1} \sum_{i=1}^n (z_{id} - \mu_d)^2. \quad (13)$$

在采样时边缘分布可以是经验分布(6)也可以是正态分布(12).

4 Copula 分布估计算法框架及复杂性分析

设优化问题 $\min f(x_1, \dots, x_D), x_i \in [a_i, b_i], i = 1, 2, \dots, D$. 基于上述 Copula 函数产生方式的分布估计算法 cEDA 如下:

算法3 cEDA.

Step 1: 在搜索空间中按均匀分布随机产生 N 个主体, 确定选择率 s 和变异率 t .

Step 2: 根据适应值选择其中的 $n = s \times N$ 个主体作为优势群体 $z = \{z_1, \dots, z_n\}$ 保留在群体中.

Step 3: $f = \text{calculatef}(z)$, 即执行算法 1, 其中的 Step 1 可以使用式 (6) 或 (12).

Step 4: 重复下列步骤 $m = (1 - s - t) \times N$ 次, 产生 m 个主体加入到群体中.

Step 4.1: $(u_1, \dots, u_D) = \text{generation}(f)$;

Step 4.2: for $(i = 1 : D)$

$$x_i = F_i^{-1}(u_i),$$

end

(x_1, \dots, x_D) 为产生的新主体.

Step 5: 在搜索空间中按均匀分布随机产生 $t \times N$ 个主体加入到群体中.

Step 6: 若满足算法终止条件, 则结束, 群体中具有最优适应值的主体, 即优化结果; 否则, 转 Step 2 继续执行.

该算法中 Step 4.2 的 F_i 可以是经验分布函数 (6) 或正态分布函数 (12).

文献 [11] 中已证明当 n 能被 K 整除时, 上述方法构造的函数 $C(u)$ 是一个 Copula 函数. 构造 Copula 函数的时间复杂度为 $O(D^2 n \log n)$. 根据 $C(u)$ 产生服从该分布的随机向量 (u_1, \dots, u_D) 的时间复杂度为 $O(D^2 \log K)$. 对于分布估计算法 cEDA, 设优化问题的维数为 D , 群体规模为 N , 选择率为 s , 则根据选择的群体估计概率模型并采样生成下一代群体的时间复杂度为 $O(D^2 s N \log(sN) + D^2 N \log K)$.

5 仿真实验

为测试算法的性能, 并与其他算法进行比较, 对下面 3 个函数进行测试:

测试函数 1: $f_1(x) = -\left\{10^{-5} + \sum_{i=1}^D |y_i|\right\}^{-1}$, 其中 $y_1 = x_1, y_i = y_{i-1} + x_i, i = 2, \dots, D, -0.16 \leq x_i \leq 0.16$.

测试函数 2: $f_2(x) = \sum_{i=1}^D [(x_1 - x_i^2)^2 + (x_i - 1)^2]$, 其中 $-10 \leq x_i \leq 10$.

测试函数 3: $f_3(x) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$, 其中 $-600 \leq x_i \leq 600$.

3 个测试函数均为最小化问题. cEDA 采用经验 Copula 函数和边缘正态分布函数对所选群体进行概率模型估计并采样, 用符号 cEDAPG 表示. 参数设置采用文献 [2] 中的参数, 即问题维数 $D = 10$; 3 个问题对应的群体规模分别是 2000, 2000 和 750. 算法最大

的适应值评价次数是 300 000 次, 并使用截断选择算子进行优势群体的选择. 在算法中还采用了变异算子, 变异率为 0.05. 在构造 Copula 函数时, 对 $[0, 1]^D$ 空间不进行划分, 即取 $K = 1$. 采样时使用的方差有下面 3 种策略^[3]:

1) 方差为固定值.

2) 将优势群体的样本方差作为采样时的方差, 即

$$\sigma_i = \sqrt{\left(\sum_{j=1}^S (x_{ij} - \bar{x}_i)^2\right) / (S - 1)}, \text{ 其中 } \bar{x}_i = \frac{1}{S} \sum_{j=1}^S x_{ij}.$$

3) 方差采用线性变化的方式, 即 $\sigma_i^{t+1} = (1 -$

$$\alpha)\sigma_i^t + \alpha \sqrt{\left(\sum_{j=1}^S (x_{ij} - \bar{x}_i)^2\right) / (S - 1)}, \alpha = 0.2.$$

表 1~表 3 中的实验结果表明, cEDAPG 受算法参数的影响较大. 与其他算法相比, 测试的 3 个函数在适当的参数设置下能够以较高的精度找到全局最优解. 特别是测试函数 3 在选择率为 0.2, 方差更新方式采用策略 B 时, 平均进化 68.7 代即可找到全局最优解.

表 1 cEDA 在测试函数 f_1 的运行结果及与其他算法的比较

算法	选择率	σ	mean	StdVar
UMDA _c ^G			-53 460	
MIMIC _c ^G			-58 775	
EGNA _{ee}			-100 000	
EGNA _{BGe}			-100 000	
ES			-5910	
cEDAPG	0.5	A:0.02	-30.277 5	3.434 5
	0.5	A:0.05	-12.517 4	1.483 7
	0.2B	B	-60 770	27 564
	0.2	B	-1 074.6	482.230 1
	0.5	C	-184.097 6	52.964 0
	1/3	C	-561.731 2	178.825 4
	0.2B	C	-9910.3	2 692.9
	0.2	C	-2 089.4	587.480 1

表 2 cEDA 在测试函数 f_2 的运行结果及与其他算法的比较

算法	选择率	σ	mean	StdVar
UMDA _c ^G			0.137 54	
MIMIC _c ^G			0.133 97	
EGNA _{ee}			0.099 14	
EGNA _{BGe}			0.025 0	
ES			0	
cEDAPG	0.5	A:0.02	4.074 3	0.857 2
	0.5	A:0.05	0.009 7	0.003 2
	0.2B	B	$2.383 3 \times 10^{-4}$	$5.242 9 \times 10^{-5}$
	0.2	B	0.147 8	0.025 9
	0.5	C	$8.244 3 \times 10^{-4}$	0.001 9
	1/3	C	$1.870 7 \times 10^{-5}$	$4.681 7 \times 10^{-5}$
	0.2	C	$3.003 4 \times 10^{-7}$	$7.317 9 \times 10^{-7}$

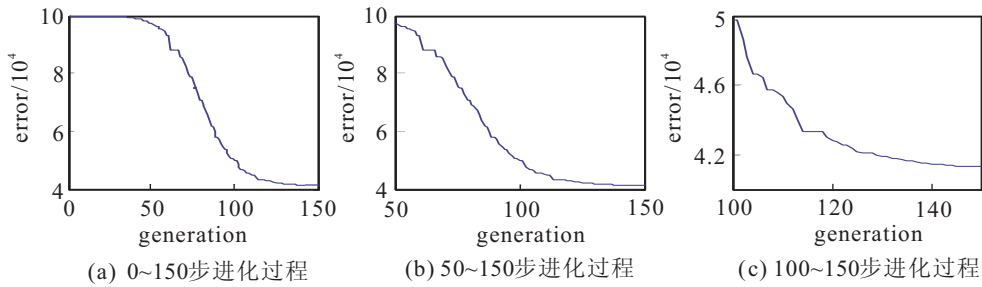


图 1 cEDA 在优化测试函数 f_1 时的一代进化过程

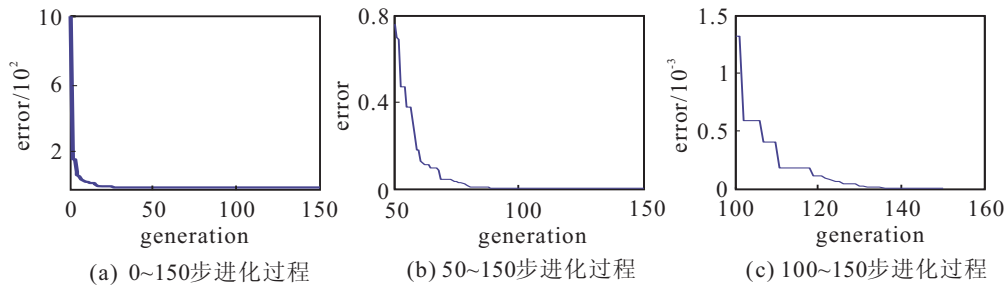


图 2 cEDA 在优化测试函数 f_2 时的一代进化过程

表 3 cEDA 在测试函数 f_3 的运行结果及与其他算法的比较

算法	选择率	σ	mean	StdVar
UMDA _c ^G			0.011 076	
MIMIC _c ^G			0.007 794	
EGNA _{ee}			0.008 175	
EGNA _{BGe}			0.012 605	
ES			0.034 477	
cEDAPG	0.5	A:.02	1.079 6	0.197 7
	0.5	A:.05	0.548 9	0.369 4
	0.2B	B	0.004 6	0.015 1
	0.2	B	0	0
	0.5	C	1.3192×10^{-9}	3.1529×10^{-9}
	1/3	C	4.2966×10^{-14}	1.0565×10^{-13}
	0.2	C	0	0

从图 1 和图 2 中可以看出, 算法 cEDAPG 的全局探测能力较强, 能够快速地找到全局最优解附近. 但 cEDAPG 的局部开采能力较弱, 虽然在进化后期算法找到的解仍然有所改进, 但速度非常慢.

6 结 论

在分布估计算法中, 重要的两个步骤是对所选择群体的概率模型进行估计, 以及根据估计的模型进行采样. Copula 理论将多元联合分布函数分解为一元的分布函数和反映它们之间联系的一个 Copula 函数. 算法 cEDA 将 Copula 理论引入分布估计算法中, 实验表明, cEDA 能够收敛到全局最优解, 但算法的局部开采能力较弱, 在进化后期收敛的速度较慢, 并且, 算法的参数设置对于 cEDA 的优化效果影响较大. 因此, 以下问题值得进一步研究.

1) 参数的选择. 研究各个参数对 cEDA 性能的影响程度, 哪些参数是起关键作用的, 从而对如何选择适当的参数进行指导.

2) 算法收敛性分析. 根据 Copula 理论以及现有的分布估计算法收敛性的理论研究, 对 cEDA 的收敛性进行理论上的分析.

3) 在算法后期采用局部搜索方法进行改进. 实验表明 cEDA 的全局探测能力较强, 但局部开采能力较弱. 因此, 可在 cEDA 中引入其他的进化算法以增强其局部开采能力, 或在算法后期采用爬山法等局部搜索法进行改进.

参考文献(References)

- [1] Larranaga P, Lozano J. Estimation of distribution algorithms, a new tool for evolutionary computation[M]. Holland: Kluwer Academic Publishers, 2002.
- [2] Larranaga P, Etxeberria R, Lozano J, et al. Optimization in continuous domains by learning and simulation of gaussian networks[C]. Proc of the GECCO-2000 Workshop in Optimization by Building and Using Probabilistic Models. Morgan Kaufmann, 2000: 201-204.
- [3] Sebag M, Ducoulombier A. Extending population-based incremental learning to continuous search spaces[C]. Parallel Problem Solving from Nature - PPSN V. Amsterdam, 1998: 418-427.
- [4] Zhang Q, Muhlenbein H. On the convergence of a class of estimation of distribution algorithms[J]. IEEE Trans on Evolutionary Computation, 2004, 8(2): 127-136.
- [5] Harik G, Lobo F, Goldberg D. The compact genetic algorithm[C]. IEEE CEC'98. Anchorage, 1998: 523-528.
- [6] Mühlenbein H. The equation for response to selection and its use for prediction[J]. Evolutionary Computation, 1997, 5(3): 303-346.