计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20170155
54(8):1644-1654,2017

求解大尺度优化问题的学生 t-分布估计算法

王豫峰[1,2] 董文永[1] 董学士[1] 王 浩[3]
[1](武汉大学计算机学院 武汉 430072)
[2](南阳理工学院软件学院 河南南阳 473000)
[3](岩土力学与工程国家重点实验室(中国科学院武汉岩土力学研究所) 武汉 430071)
(wangyufeng@whu.edu.cn)

Adaptive Estimation of Student's t-Distribution Algorithm for Large-Scale Global Optimization

Wang Yufeng[1,2], Dong Wenyong[1], Dong Xueshi[1], and Wang Hao[3]
[1](*Computer School, Wuhan University, Wuhan* 430072)
[2](*Software School, Nanyang Institute of Technology, Nanyang, Henan* 473000)
[3](*State Key Laboratory of Geomechanics and Geotechnical Engineering (Institute of Rock and Soil Mechanics, Chinese Academy of Sciences), Wuhan* 430071)

Abstract In this paper, an adaptive estimation of student's t-distribution algorithm (EDA-t) is proposed to deal with the large-scale global optimization problems. The proposed algorithm can not only obtain optimal solution with high precision, but also run faster than EDA and their variants. In order to reduce the number of the parameters in student's t-distribution, we adapt its closed-form in latent space to replace it, and use the expectation maximization algorithm to estimate its parameters. To escape from local optimum, a new strategy adaptively tune the degree of freedom in the t-distribution is also proposed. As we introduce the technology of latent variable, the computational cost in EDA-t significantly decreases while the quality of solution can be guaranteed. The experimental results show that the performance of EDA-t is super than or equal to the state-of-the-art evolutionary algorithms for solving the large scale optimization problems.

Key words probabilistic PCA; student's t-distribution; estimation of distribution algorithm (EDA); large scale global optimization (LSGO); expectation maximization (EM) algorithm

摘 要 针对处理大尺度全局优化问题,提出一种基于自适应 t-分布的分布估计算法(EDA-t).该算法不仅求解效果良好,而且求解速度也比同类型算法快.其基本思想是:在迭代搜索过程,首先利用期望最大化算法对演化种群进行概率主成分分析,然后根据得到的概率隐变量建立算法的概率模型,并通过 t-分布自由度自适应方法,在算法收敛停滞时跳出局部最优.由于在构建模型时进行了数据降维,在不影响算法求解精度的前提下,其计算开销得到了明显降低.通过和目前主流的演化算法在大尺度优化测试函数上的仿真实验和分析,验证了所提算法的有效性和适用性.

收稿日期:2017-03-17;**修回日期**:2017-05-15
基金项目:国家自然科学基金项目(61170305,61672024,41472288);河南省高等学校重点科研项目计划(17A520046)
This work was supported by the National Natural Science Foundation of China (61170305, 61672024, 41472288) and the Key Research Program in Higher Education of Henan Province (17A520046).
通信作者:董文永(dwy@whu.edu.cn)

关键词 概率主成分分析;学生 t-分布;分布估计算法;大尺度全局优化;最大期望算法

中图法分类号 TP301

分布估计算法(estimation of distribution algorithm, EDA)[1],是元启发算法的重要分支之一.它最开始被用于求解组合优化问题,后来扩展到求解连续优化问题. EDA 是一种随机优化算法,通过对有希望的候选解采样来建立概率模型,从而探索潜在的解空间[2]. 通过学习的模型(单变量模型、双变量模型、多变量模型、混合变量模型),EDA 可以提取变量中的一些隐藏的关系[3]. 例如:PBIL[4], UMDAG c[5], MIMIC[6], BMDA[7], BOA[8], IDEA[9], GMM-EDA[10]等. 它们都被成功应用到很多低维优化问题中.

然而,当优化问题的维度增加时,由于问题变量之间的强相关性和高复杂度,原始的元启发算法面临着巨大的挑战. 大尺度优化问题(large scale global optimization, LSGO)极难被处理,主要有 2 方面原因:1)维度灾难. 随着问题维度的增长,搜索空间指数级地增长. 2)问题维度的增长导致问题的特性(多峰、奇异点等)变得更加复杂. 比如:Rosenbrock 函数在两维空间中是一个单峰问题,但是,随着问题维度的增加,该函数变成多峰问题[11]. 当问题的维度增长时,EDA 的性能下降非常快. 为了克服这一个问题,一些 EDA 的变体算法被提出. Posik[12] 提出利用独立成分分析方法对原始坐标系进行变换的遗传算法. Dong 等人[13] 提出一种基于模型复杂度控制的分布估计算法框架,它通过利用识别变量的弱相关行进行子空间建模. Kaban 等人[14] 提出一种基于随机投影的分布估计算法,它通过随机投影方式对原始协方差矩阵进行压缩.

本文提出一种基于自适应 t-分布的分布估计算法(adaptive estimation of student's t-distribution algorithm, EDA-t). EDA-t 利用 t-分布的重尾特性在全局搜索中探索更多的解空间. 在整个演化过程中,利用概率主成分分析方法学习当前最优种群的主特征隐变量空间. 在学习过程中,通过期望最大值估计(expectation maximization, EM)算法代替原来的特征值分解方法,来学习当前种群主特征值的概率隐空间. 根据得到的概率隐空间构建算法的概率模型,从而减少生成模型的计算开销,并加快算法收敛速度. 通过对 t-分布自由度的自适应,帮助算法在收敛停滞时跳出局部最优. 本文提出的算法主要有 3 点创新:1)由传统的基于高斯分布的 EDA 算法扩展到学生 t-分布;2)在概率模型参数的估计方面,采用 EM 算法来降低原始 EDA 估计参数的复杂度;3)由于学生 t-分布的自由度参数可以控制分布的形状,自适应调整该参数能够很好地解决基于种群算法的"探索"与"开发"之间的矛盾.

1 相关工作

1.1 EDA 框架

EDA 从提出到现在,它的变体算法使用了很多种不同的概率模型,也有很多种实现方法. 但是,EDA 的基本框架可以总结如算法 1 所示.

算法 1. EDA 的基本框架.

输入:种群大小 N、种群选择数 K;

输出:最优种群 Pop_g.

① $Pop_0 \leftarrow InitialPop(N)$; /* 初试化种群,随机生成 N 个个体 */

② $F_0 \leftarrow f(Pop_0)$; /* 求出 Pop_0 每个个体的函数值 */

③ repeat

④ $Pop_{g-1}^{Se} \leftarrow getSelectionPop(Pop_{g-1}, Pop_g)$; /* 从上一代($g-1$)的种群里选择前 K 个个体 */

⑤ $p_g(x)=p(x|Pop_{g-1}^{Se})$; /* 根据选择的个体生成概率分布 */

⑥ $Pop_g \leftarrow getSamplePop(p_g(x))$; /* 根据选择的个体生成概率分布 */

⑦ $F_g \leftarrow f(Pop_g)$; /* 求出 Pop_g 每个个体的函数值 */

⑧ $Pop_g \leftarrow Combine(Pop_{g-1}, Pop_g)$; /* 根据函数值,将种群 Pop_{g-1}, Pop_g 中的最优个体进行合并 */

⑨ until a stopping criterion is met.

EDA 是一种随机优化算法,它通过对已找到的可能解进行抽样,构建算法的概率模型,探测候选解空间[2]. 在算法 1 中,EDA 首先通过评估函数值来选择最优个体组成种群 Pop_{g-1}^{Se},根据种群 Pop_{g-1}^{Se} 学习概率模型. 然后,基于学习到的概率模型生成 Pop_g. 最后,将新生成的种群 Pop_g 和上一代种群 Pop_{g-1} 的最优个体组成新的种群 Pop_g. 概率模型的选择对于 EDA 的性能影响非常大.

1.2 学生 t-分布

学生 t-分布是一种统计分布. 它的分布曲线呈

对称钟形形状,并且曲线的图形与它的自由度 υ 大小有关. 当 υ 比较小的时候,t-分布曲线中间较低,两侧尾部翘得较高;当 υ 增大时,分布曲线接近正态分布曲线;当 $\upsilon\to+\infty$ 时,t-分布曲线即为标准正态分布曲线. 合理利用 t-分布的这个重尾特性,能够设计出一个非常好的概率模型.

假设 $\boldsymbol{x}$ 是通过多元学生 t-分布 $S(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\upsilon)$ 生成的 D 维随机变量:

$$S(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\upsilon)=\frac{\Gamma\left(\frac{\upsilon+D}{2}\right)|\boldsymbol{\Sigma}|^{-\frac{1}{2}}}{(\pi\upsilon)^{\frac{D}{2}}\Gamma\left(\frac{\upsilon}{2}\right)}\times$$

$$\left[1+\frac{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{\upsilon}\right]^{-\frac{(\upsilon+D)}{2}},\tag{1}$$

其中,$\boldsymbol{\mu}$ 代表均值;$\boldsymbol{\Sigma}$ 代表正定内积矩阵;υ 代表自由度,$\upsilon>0$;$(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ 代表 $\boldsymbol{x}$ 到 $\boldsymbol{\mu}$ 关于 $\boldsymbol{\Sigma}$ 的马氏距离的平方;$\boldsymbol{\Sigma}^{-1}$ 是矩阵 $\boldsymbol{\Sigma}$ 的逆;$\Gamma(\boldsymbol{x})$ 代表伽马函数. 当 $\upsilon<2$ 时,方差不存在. 当 $\upsilon>2$ 时,$\boldsymbol{\Sigma}\upsilon/\upsilon-2$ 就是协方差矩阵. 正态分布 $N(\boldsymbol{\mu},\boldsymbol{\Sigma})$ 实际也就等于 $S(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\infty)$.

为了更加清楚地理解 t-分布,我们给出不同自由度下的单变量 t-分布和标准高斯分布的概率密度函数图对比,如图 1 所示. 其中,所有分布均值为 0,固定方差为 1. 从图 1 中可以看到,当自由度 υ 比较小的时候,t-分布比高斯分布的尾部更翘. 从而导致生成的个体比高斯分布有更高的概率远离均值,具有较强的全局搜索能力. 在求解多峰问题的时候,更加容易逃离局部最优点,跳出停滞区. 但是,它在均值点的概率密度更低,意味着减少局部搜索时间,局部微调能力降低. 因此,t-分布的重尾特性有利也有弊.

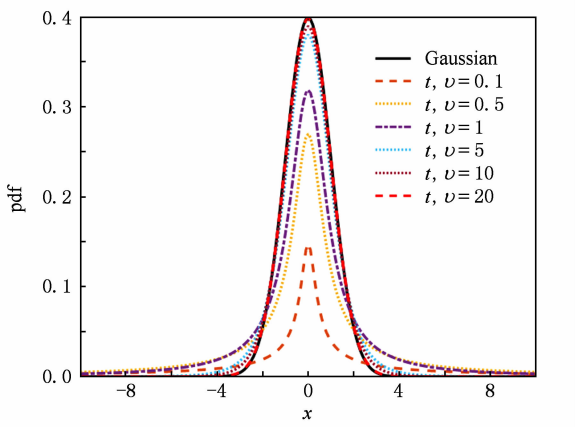


Fig. 1 Comparison between the univariate Gaussian and Students t-distributions

图 1 单变量 t-分布和标准高斯分布对比图

2 EDA-t 算法

本文提出一种新的求解大尺度优化问题的分布估计算法. 算法使用基于概率隐空间建立的多元 t-分布模型代替传统的高斯分布模型. 通过对 t-分布的自由度调整来控制算法搜索的特性. 同时,引入 EM 算法对模型参数进行估计. 和传统的 PCA 相比,大大减少了计算过程中的时间复杂度.

2.1 概率主成分分析

传统的主成分分析需要计算数据集的均值 $\boldsymbol{\mu}$ 和协方差矩阵 $\boldsymbol{\Sigma}$,然后进行特征值分解,寻找矩阵 $\boldsymbol{\Sigma}$ 的前 M 个最大特征值所对应的特征向量[15]. 而在计算特征值分解的时候,对 $D\times D$ 矩阵进行特征值分解的代价为 $O(D^3)$. 当 D 增大时,算法复杂度增长特别快[16].

概率主成分分析可以视为概率隐变量模型的最大似然解. 其主要思想是:先根据隐变量的先验分布 $p(\boldsymbol{z})$ 中生成隐变量 $\boldsymbol{z}$. 然后,通过对隐变量 $\boldsymbol{z}$ 进行线性变换和附加噪声的方式生成观测数据点 $\boldsymbol{x}$[17]. 噪声服从给定隐变量下的数据变量的某个条件概率分布 $p(\boldsymbol{x}|\boldsymbol{z})$.

$$p(\boldsymbol{z})=S(\boldsymbol{z}|0,\boldsymbol{I},\theta),\tag{2}$$

其中,$\boldsymbol{z}$ 是一个 M 维隐变量,S 为学生 t-分布,θ 为 t-分布自由度,设置为固定值 $\theta=20$.

$$p(\boldsymbol{x}|\boldsymbol{z})=S(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}+\mu,\sigma^2\boldsymbol{I},\upsilon),\tag{3}$$

$\boldsymbol{W}$ 是 $D\times M$ 的因子载荷,$\boldsymbol{\mu}$ 为 D 维样本均值,υ 为 t-分布的自由度.

$$\boldsymbol{x}=\boldsymbol{W}\boldsymbol{z}+\boldsymbol{\mu}+\boldsymbol{\varepsilon},\tag{4}$$

$\boldsymbol{\varepsilon}$ 为 D 维噪声变量.

$$p(\boldsymbol{\varepsilon})=S(\boldsymbol{\varepsilon}|0,\boldsymbol{I},\theta).\tag{5}$$

式(4)中有 4 个参数 $\boldsymbol{W},\boldsymbol{\mu},\sigma^2$ 和 υ,其中,υ 为固定值,$\boldsymbol{\mu}$ 可以直接根据种群算出. $\boldsymbol{W}$ 和 σ^2 需要通过 EM 算法进行估计. 根据概率的加和规则和乘积规则,边缘概率分布的形式为

$$p(\boldsymbol{x})=\int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})\mathrm{d}\boldsymbol{z}.\tag{6}$$

由于这对应于一个线性 t-分布模型,所以,边缘概率分布还是 t-分布,即:

$$p(\boldsymbol{x})=S(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{C},\upsilon),\tag{7}$$

其中,$D\times D$ 的协方差矩阵 $\boldsymbol{C}$ 被定义为

$$\boldsymbol{C}=\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}+\sigma^2\boldsymbol{I}.\tag{8}$$

在进行特征值分解计算的过程中,需要对 $\boldsymbol{C}$ 求逆,即需要对 $D\times D$ 的矩阵求逆:

$$\boldsymbol{C}^{-1}=\sigma^{-2}\boldsymbol{I}-\sigma^{-2}\boldsymbol{W}\boldsymbol{B}^{-1}\boldsymbol{W}^{\mathrm{T}}, \tag{9}$$

其中,$M\times M$ 的矩阵 $\boldsymbol{B}$ 为

$$\boldsymbol{B}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}+\sigma^{2}\boldsymbol{I}, \tag{10}$$

由于我们对 $\boldsymbol{B}$ 求逆,而不是直接对 $\boldsymbol{C}$ 求逆.因此,计算 $\boldsymbol{C}^{-1}$ 的算法复杂度从 $O(D^3)$ 减小到 $O(M^3)$.

2.2 模型参数估计方法

种群 Pop_g^{Se} 是将上一代种群和这代种群的最优个体组合而成,这样可以促进种群的多样性,提高概率模型的精度.μ 是种群的加权平均值.

$$\mu_g=\sum_{i=1}^{N}\omega_i\boldsymbol{x}_i, \tag{11}$$

$$\sum_{i=1}^{N}\omega_i=1,\ \omega_1\geqslant\omega_2\geqslant\cdots\geqslant\omega_N\geqslant 0, \tag{12}$$

$\omega_i\in\mathbb{R}^+$,$1\leqslant i\leqslant N$,是种群组合权重系数.当 $\omega_i=1/N$ 时,式(11)就是计算所有种群的平均值.

EM 是一种迭代算法.首先,通过使用旧的参数值计算隐变量的后验概率分布求期望.然后,最大化完整数据对数似然函数的期望就会产生新的参数值,直到迭代终止.完整数据的对数似然函数的形式为

$$\ln p(X,\boldsymbol{Z}\mid\boldsymbol{\mu},\boldsymbol{W},\sigma^2,\upsilon)=\sum_{n=1}^{N}\{\ln p(\boldsymbol{x}_n\mid\boldsymbol{z}_n)+\ln p(\boldsymbol{z}_n)\}, \tag{13}$$

其中,矩阵 $\boldsymbol{Z}$ 的第 n 行由 $\boldsymbol{z}_n$ 给出.均值 $\boldsymbol{\mu}$ 由式(11)计算.分别使用式(2)和式(3)给出的隐变量概率分布和条件概率分布,对隐变量上的后验概率分布求期望,于是有:

$$E[\ln p(X,\boldsymbol{Z}\mid\boldsymbol{\mu},\boldsymbol{W},\sigma^2,\upsilon)]=-\sum_{n=1}^{N}\left\{\frac{D}{2}\ln(2\pi\sigma^2)+\frac{1}{2}Tr(E[\boldsymbol{z}_n\boldsymbol{z}_n^{\mathrm{T}}])+\frac{1}{2\sigma^2}\|\boldsymbol{x}_n-\boldsymbol{\mu}\|^2-\frac{1}{\sigma^2}E[\boldsymbol{z}_n]^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{x}_n-\boldsymbol{\mu})+\frac{1}{2\sigma^2}Tr(E[\boldsymbol{z}_n\boldsymbol{z}_n^{\mathrm{T}}]\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})+\frac{M}{2}\ln(2\pi)\right\}, \tag{14}$$

在 E 步,利用旧的参数去计算期望值:

$$E[\boldsymbol{z}_n]=\boldsymbol{B}^{-1}\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{x}_n-\bar{\boldsymbol{x}}), \tag{15}$$

$$E[\boldsymbol{z}_n\boldsymbol{z}_n^{\mathrm{T}}]=\sigma^2\boldsymbol{B}^{-1}+E[\boldsymbol{z}_n]E[\boldsymbol{z}_n]^{\mathrm{T}}, \tag{16}$$

在 M 步,估计新的参数值:

$$\boldsymbol{W}_{\mathrm{new}}=\Big[\sum_{n=1}^{N}(\boldsymbol{x}_n-\bar{\boldsymbol{x}})E[\boldsymbol{z}_n]^{\mathrm{T}}\Big]\Big[\sum_{n=1}^{N}E[\boldsymbol{z}_n\boldsymbol{z}_n^{\mathrm{T}}]\Big]^{-1}, \tag{17}$$

$$\sigma^2_{\mathrm{new}}=\frac{1}{N\times D}\sum_{n=1}^{N}\{\|\boldsymbol{x}_n-\bar{\boldsymbol{x}}\|^2-2E[\boldsymbol{z}_n]^{\mathrm{T}}\boldsymbol{W}_{\mathrm{new}}^{\mathrm{T}}(\boldsymbol{x}_n-\bar{\boldsymbol{x}})+Tr(E[\boldsymbol{z}_n\boldsymbol{z}_n^{\mathrm{T}}]\boldsymbol{W}_{\mathrm{new}}^{\mathrm{T}}\boldsymbol{W}_{\mathrm{new}})\}, \tag{18}$$

算法轮流执行 E 步和 M 步,一直到收敛.

2.3 自由度自适应策略

对于 t-分布,当 $\upsilon=1$ 时,t-分布就是标准柯西分布 $C(0,1)$;当 $\upsilon\to+\infty$ 时,t-分布趋近标准高斯分布 $N(0,1)$.柯西分布的全局探索能力较强,能够保持种群的多样性;而高斯分布的局部开发能力较强,可以保证种群的收敛精度.为了综合两者优点,通过采取自由度自适应的方法来提高算法的灵活性和求解精度.

算法在每一次迭代计算完成后,设定一个进化状态监视器 U,记录当前种群中最优个体.当连续 k 代最优个体函数值变化为 0 时,出现收敛停滞状态,算法按比例 α 减小自由度,扩大算法全局探索能力,跳出局部最优值区域.

$$\upsilon_g=\begin{cases}(1-\alpha)\times\upsilon_{g-1}, & U_g=U_{g-1}=\cdots=U_{g-k},\ g>k;\\ \upsilon_{g-1}+1, & \text{otherwise.}\end{cases} \tag{19}$$

2.4 算法框架

算法 2 展示了 EDA-t 的伪代码,算法由 4 个部分组成:初始化、抽样、选择和参数估计.其中,行①为初始化部分.参数 $\boldsymbol{W}$ 是 $D\times M$ 的旋转矩阵,从隐空间变化到种群原始空间,初始设置为 $D\times D$ 的单位矩阵前 M 列,参数 σ 初始设置为 1;代码行③④为样本抽样部分;代码行⑤~⑨为样本选择部分;代码行⑩~⑫为参数估计部分,利用新获取的种群对概率模型的参数进行估计;

算法 2. EDA-t 算法.

输入:种群大小 N、隐空间大小 M、混合率 ω、自由度 υ;

输出:最优种群 Pop_g.

① 以随机方式初始化 $\mu,\boldsymbol{W},\sigma,g=0,\alpha=0.2,k=15$;
② while 未达到终止条件时 do
③ 依据式(2)生成 N 个 M 维隐变量 $\boldsymbol{Z}$;
④ 依据式(3)生成 N 个 D 维个体 $\boldsymbol{X}$;
⑤ if $g>0$ then
⑥ 从种群 Pop_{g-1} 中选择前 $\omega\times N$ 个最好个体和从种群 Pop_g 中选择前 $(1-\omega)\times N$ 个最好个体组成 Pop_g^{Se};
⑦ else
⑧ 将种群 Pop_l 赋值给 Pop_g^{Se};
⑨ end if
⑩ 对种群 Pop_g^{Se},依据式(11)和式(12)计算均值 μ;
⑪ $U=U\cup Pop_{\mathrm{best}}$;

⑫ 依据式(14)～(18),利用 EM 算法,计算概率分布的参数 $\boldsymbol{W}$ 和参数 σ;

⑬ 依据式(19)更新参数 υ;

⑭ $g=g+1$;

⑮ end while

2.5 算法复杂度分析

设问题维度为 D,种群大小为 N,隐变量大小为 M.算法 2 中行③初始化隐变量 $\boldsymbol{Z}$ 的算法复杂度为 $O(NM)$;行④生成观测变量 $\boldsymbol{X}$ 的算法复杂度为 $O(ND)$;行⑤～⑨对种群进行重新选择组合,算法复杂度为 $O(N)$;行⑩计算种群均值算法复杂度为 $O(N)$;行⑪通过 EM 算法估计参数 W 和参数 σ.其中,EM 算法的迭代次数为 k,算法的复杂度为 $O(kNDM)$;在本算法中,去除所有低阶项,算法 2 的时间复杂度为 $O(kNDM)$.由于 k、N 和 M 都为固定值并且远远小于 D,所以,整个算法的复杂度是远远小于基于特征值分解的算法 $O(D^3)$.

3 仿真实验与分析

为了充分验证 EDA-t 算法在不同维度下的性能,实验选取了 13 个测试函数进行了实验分析.其中,F_1-F_2 是可分离单峰问题;F_3-F_{10} 是不可分离并且有极少(≤2)局部最优问题;$F_{11}-F_{13}$ 是多峰并且具有多局部最优问题.函数具体情况如表 1 所示,详细函数的设置见文献[13].

3.1 实验设置

在对比实验中,测试问题维度为 1000 维,函数的最大评估次数 $FEs=2.5\times10^6$,实验中每一个算法对于每一个测试函数均独立运行 25 次.实验系统环境为 64 位的 Windows 7 系统,CPU 为 Intel® Core™ i3 3.60 GHz、内存为 4 GB、代码运行环境为 Matlab R2015a.

Table 1 Benchmark Functions

表 1 基准测试函数

Fun	Description	Range
F_1	Sphere	[−100,100]
F_2	Shifted sphere	[−100,100]
F_3	Schwefel's problem 2.21	[−100,100]
F_4	Shifted F_3	[−100,100]
F_5	Schwefel	[−10,10]
F_6	Shifted F_5	[−10,10]
F_7	Rosenbrock	[−100,100]
F_8	Shifted Rosenbrock	[−100,100]
F_9	Shifted rotated high conditioned elliptic	[−100,100]
F_{10}	Schwefel 2.6 with global optimum on bounds	[−100,100]
F_{11}	Rastrigin	[−5,5]
F_{12}	Shifted rotated Rastrigin	[−5,5]
F_{13}	Shifted expanded Griewank plus Ronsenbrock	[−3,1]

在 1000 维实验中,选取了 4 个知名算法进行比较,分别是 SaNSDE[18],DECC-G[19],EDA-MCC[13] 和 RP-EDA[14],算法参数保持与原文献一致的设置.EDA-t 算法中,种群大小 $N=200$、隐空间大小 $M=3$、混合率 $\omega=0.8$,详细参数设置见 3.4 节.

为了客观公正的评价 EDA-t 算法的性能,采用 Wilcoxon 秩和检验和 Friedman 检验方法对实验结果进行统计分析.Wilcoxon 秩和检验的显著性水平为 0.05,在表 2 的底部的符号"−","+"和"≈"分别表示劣于、优于和相当于 EDA-t 的结果.

Table 2 Mean Error, Standard Deviation and Comparison Results Based on Wilcoxon's Rank Sum Test

表 2 各算法的平均误差值、标准差和 Wilcoxon 秩和检验结果

Fun	Mean Error±Std Dev				
	SaNSDE	DECC-G	EDA-MCC	RP-EDA	EDA-t
F_1	6.97E+00±3.12E−01−	2.17E−15±9.24E−16−	2.17E−25±8.24E−26−	2.59E−42±3.89E−43−	**4.60E−84±1.12E−84**
F_2	1.60E+02±7.21E+01−	1.55E−08±8.35E−09−	1.55E−08±3.19E−09−	1.23E−05±7.38E−07−	**8.21E−18±1.15E−18**
F_3	4.99E+01±1.04E+00−	**1.01E−01±7.84E−02+**	3.68E+00±7.23E−01≈	2.72E+01±5.83E+00−	3.26E+00±8.47E−01
F_4	1.17E+02±7.23E+01−	8.65E+01±5.61E+01≈	1.24E+02±5.15E+01−	**3.82E+01±1.19E+00≈**	6.45E+01±1.28E+01
F_5	1.09E+01±8.12E+00−	1.19E−03±9.32E−04−	1.91E−12±7.35E−13−	5.42E−01±3.15E−02−	**5.12E−18±9.23E−19**
F_6	3.16E+01±9.78E+00−	8.48E−04±5.12E−04−	6.14E−09±1.01E−09−	1.28E+10±2.11E+07−	**2.89E−22±7.16E−23**
F_7	3.31E+04±8.56E+03−	**9.87E+02±4.25E+02≈**	4.36E+03±9.81E+02≈	1.05E+03±8.33E+02≈	1.00E+03±5.47E+02
F_8	3.96E+06±1.03E+06−	6.78E+03±1.34E+02−	2.47E+03±6.21E+02≈	1.08E+03±7.67E+02≈	**1.03E+03±7.64E+02**
F_9	7.79E+08±1.11E+07−	2.37E+09±8.79E+08−	7.19E+07±9.19E+06−	8.83E+08±5.48E+07−	**2.38E+07±9.42E+08**

Continued (Table 2)

Fun	Mean Error±Std Dev				
	SaNSDE	DECC-G	EDA-MCC	RP-EDA	EDA-t
F_{10}	1.99E+05±8.42E+04−	2.38E+05±9.87E+04−	1.09E+05±5.32E+04≈	**9.15E+04±3.98E+03**≈	9.84E+04±1.12E+04
F_{11}	8.69E+02±4.23E+02−	3.55E+03±8.08E+02−	7.13E+03±2.01E+03−	7.17E+02±6.05E+01−	**2.18E+02±8.01E+01**
F_{12}	8.84E+03±4.10E+03−	1.47E+04±8.71E+03−	9.68E+03±1.18E+03−	**8.78E+02±6.82E+01**+	2.81E+02±8.36E+01
F_{13}	6.27E+02±3.57E+02≈	4.65E+02±1.67E+02≈	8.71E+02±1.21E+02−	**1.16E+02±3.01E+00**≈	3.71E+02±4.13E+01
−/+/≈	12/0/1	9/1/3	9/0/4	7/1/5	

3.2 实验结果分析

通过表 2、表 3、图 2 和图 3 分析可发现,EDA-t 算法在收敛速度、收敛精度和运行时间方面,与其他算法相比占有较大优势. 特别是在 F_1,F_2,F_5,F_6,F_8 和 F_9 测试函数上,EDA-t 算法可以快速收敛,并取得最优解.

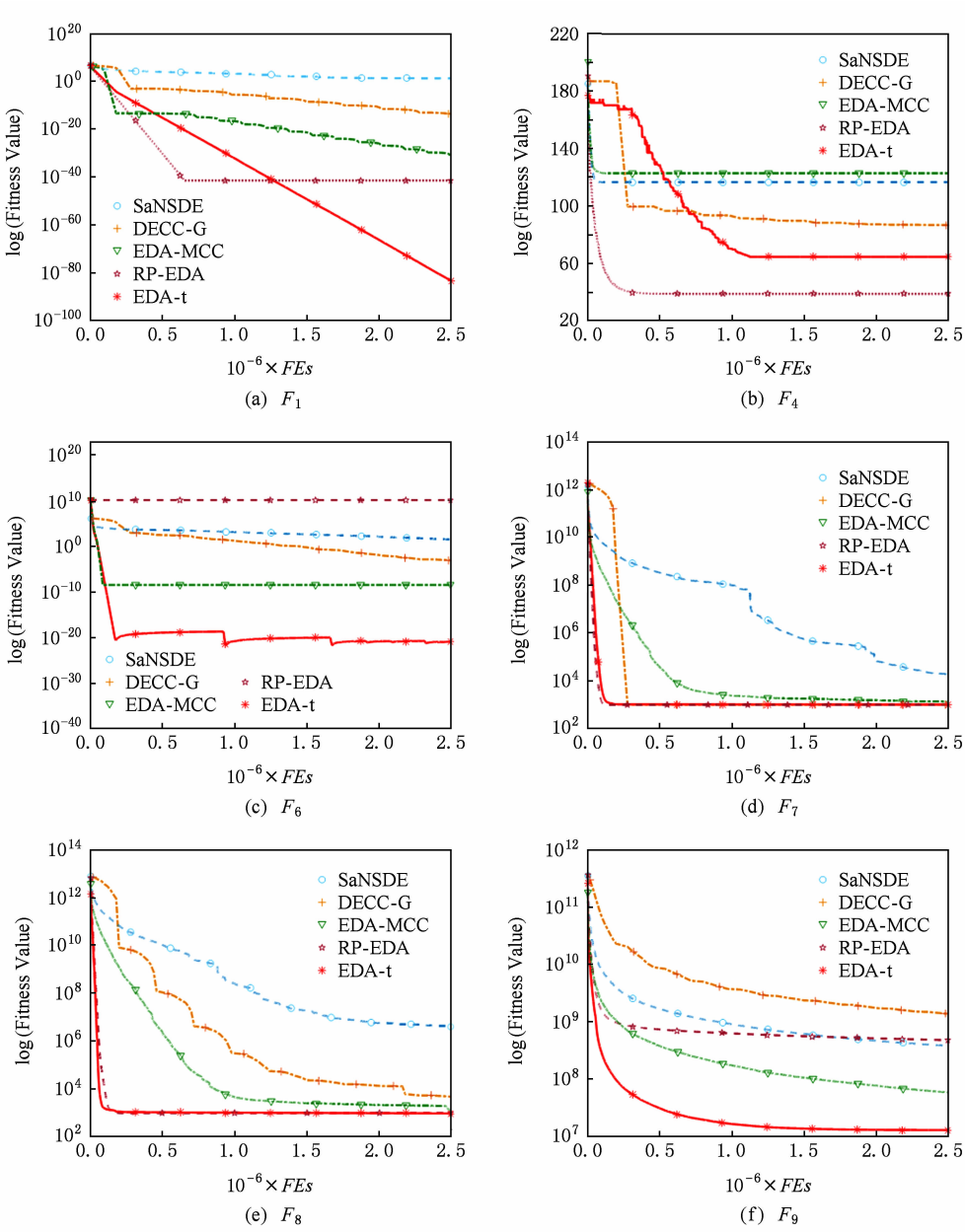


Fig. 2 Convergence curve of five algorithms on functions F_1, F_4, F_6, F_7, F_8 and F_9

图 2 各算法在函数 F_1, F_4, F_6, F_7, F_8 和 F_9 上的收敛曲线

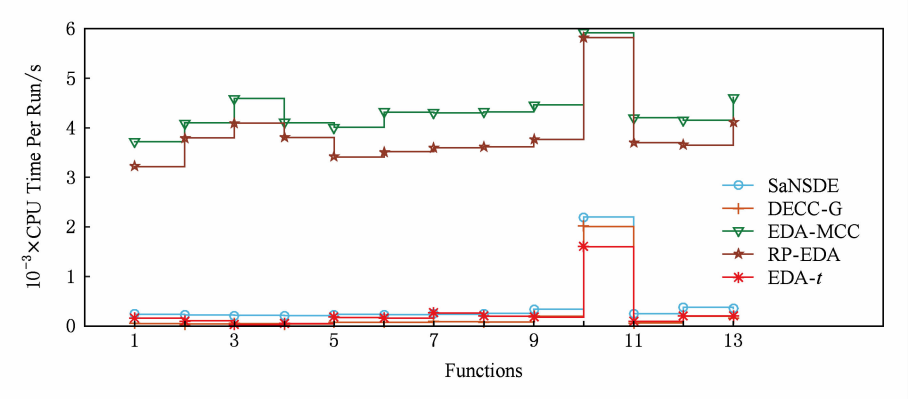


Fig. 3 CPU time per run of five algorithms

图 3 各算法的运行时间

Table 3 Average Ranking Achieved by Friedman Test

表 3 各算法的 Friedman 检验排名

Algorithm	Rank
SaNSDE	4.69
DECC-G	3.50
EDA-MCC	3.58
RP-EDA	3.08
EDA-*t*	1.38

在表 2 中，EDA-*t* 除 F_3，F_7 和 F_{13} 三个函数外均全面优于 SaNSDE 和 DECC-G. SaNSDE 采用邻域搜索的差分演化算法，该算法对低维的优化问题能取得较好的解. 但是，对高维优化问题的搜索代价过大，往往不能得到很好的解. DECC-G 是采用随机分组的协同演化算法，对于大尺度可分离优化问题能够得到较好的解. 但是，对于非可分的优化问题，求解效果不太好. 因此，EDA-*t* 在求解大尺度全局优化问题全面优于这 2 种算法.

EDA-*t* 在 F_1，F_2，F_5，F_6，F_9，F_{11} 和 F_{12} 七个函数上要优于 EDA-MCC 和 RP-EDA，在 F_3，F_4，F_7，F_8，F_{10} 和 F_{13} 六个函数上表现相似. EDA-MCC 通过识别变量之间的弱相关性来控制模型的复杂度，而变量之间的相关性识别准确率直接影响算法的求解精度. RP-EDA 通过将个体随机投影到低维空间，然后在低维空间抽样再还原到原始空间的方式进行演化，在演化的过程中具有很大的随意性. EDA-*t* 是通过隐空间建立概率模型，在全局搜索的过程按当前种群主成分方向进行引导搜索，进而搜索更有效率，带来更优解.

从表 2 底部的 Wilcoxon 秩和检验的统计结果可以看出，EDA-*t* 算法相对于其他近几年出现的国际知名算法更具有明显的优势. 表 3 给出了 EDA-*t*

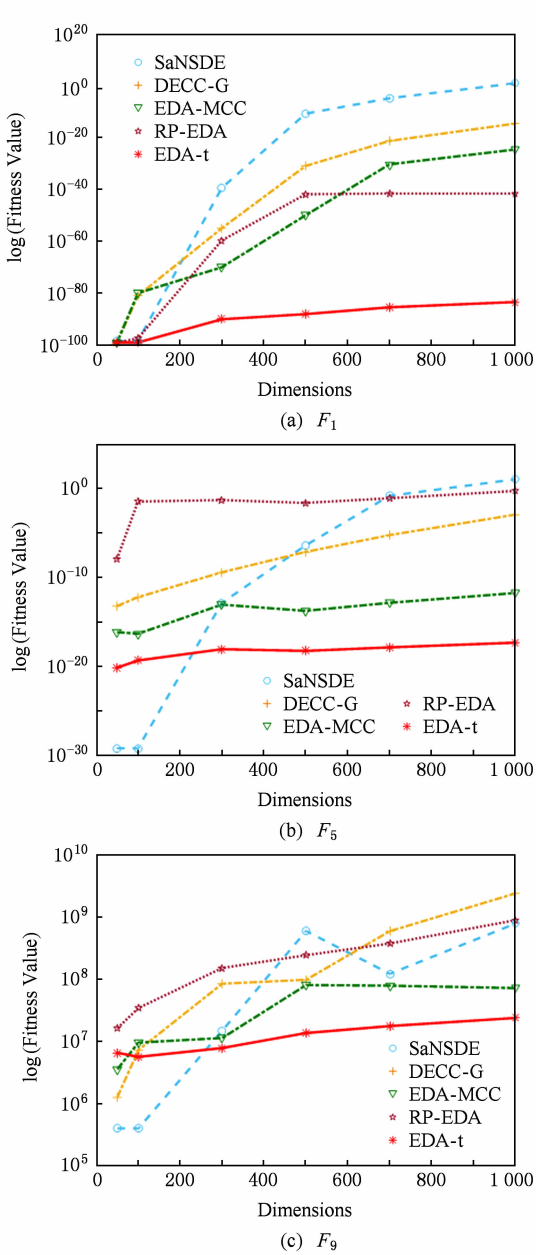


Fig. 4 Fitness value of five algorithms on different dimensions

图 4 各个算法在不同维度下收敛精度

和其他算法的 Friedman 检验排名,EDA-t 排在第一,从统计学上说明了 DECC-CLV 算法的优势.从图 3 可以看出,EDA-t 算法和其他 EDA 优化算法(EDA-MCC 和 RP-EDA)相比,算法单次运行时间大大降低.并且,基于非分解策略的 EDA-t 算法运行时间已经比较接近基于分解策略的协同演化算法 DECC-G.

综合以上实验分析可以说明 EDA-t 算法具有较快的收敛速度、较强的收敛精度和较好的函数适用性,有效提高了对大尺度全局优化问题的收敛效率.

3.3 算法的稳定性分析

为了测试算法在不同维度下的表现稳定性情况,我们将所有对比算法分别在不同问题维度 $D=[50,100,300,500,700,1\,000]$ 情况下,进行对比实验.由于篇幅有限,我们从大尺度的测试函数中选 F_1,F_5 和 F_9 三个函数进行测试.

从图 4 可以看出,对于大尺度全局优化问题,当问题维度小于 300 时,EDA-t 劣于 SaNSDE.因为 SaNSDE 是基于领域搜索的差分演化算法,在对待低维问题,可以充分发挥 SaNSDE 全局的搜索能力,求到最优解.但是,当问题维度增加时,基于领域搜索的策略导致 SaNSDE 算法搜索开销增大,性能降低.当问题维度大于 300 的时候,EDA-t 算法在不同维度下都要比其他各种算法表现要好,表现出了算法的稳定性.

3.4 参数敏感性实验

为了研究算法中参数的不同设置对于算法性能的影响.在本节中,首先对种群大小 N 的不同取值进行对比实验.然后,在种群大小固定的前提下,通过使用不同的参数设置进行对比实验.在本节对比实验中,函数的最大评估次数 $FEs=1\,000\times D$.

从表 4 可以看出,在不同的问题维度下,EDA-t 取得最优值时,所需的种群大小也不一样.因为,本文算法是在 1000 维下和其他知名算法进行对比实验.所以,在实际计算中,种群大小设置为 $N=200$.

Table 4 Experimental Results of Different N on F_2

表 4 F_2 函数上不同种群大小 N 的对比实验

F_2	N										
	30	50	80	100	150	200	300	500	800	1 000	2 000
30-D	**2.15E−26**	**3.44E−26**	1.65E−22	1.41E−17	1.01E−10	3.74E−07	1.21E−03	7.79E−01	3.08E+01	1.03E+02	1.20E+03
50-D	**8.62E−26**	**9.74E−26**	**1.94E−25**	3.35E−22	5.31E−14	1.03E−09	2.98E−05	9.94E−02	1.14E+01	5.00E+01	1.18E+03
80-D	**3.59E−25**	**2.26E−25**	**3.61E−25**	**6.83E−25**	1.02E−17	1.31E−12	3.54E−07	7.76E−03	2.56E+00	1.83E+01	9.26E+02
100-D	**6.31E−25**	**3.61E−25**	**5.52E−25**	**8.95E−25**	1.13E−19	3.70E−14	2.53E−08	1.80E−03	1.11E+00	9.82E+00	7.64E+02
300-D	6.56E+03	**6.08E−24**	**3.82E−24**	**5.47E−24**	**1.24E−23**	**2.26E−23**	1.08E−15	3.65E−08	1.41E−03	5.23E−02	8.30E+01
500-D	1.12E+05	**3.00E−23**	**9.90E−24**	**1.43E−23**	**2.78E−23**	**3.64E−23**	3.32E−20	3.39E−11	1.47E−05	1.42E−03	1.59E+01
800-D	5.16E+05	2.66E+03	**2.35E−23**	**3.58E−23**	**6.37E−23**	**7.36E−23**	**4.99E−22**	1.38E−14	7.41E−08	2.03E−05	1.98E+00
1000-D	9.20E+05	3.54E+04	3.68E−22	5.99E−22	9.51E−22	**1.05E−23**	**7.62E−22**	2.17E−16	4.15E−09	1.71E−06	6.50E−01

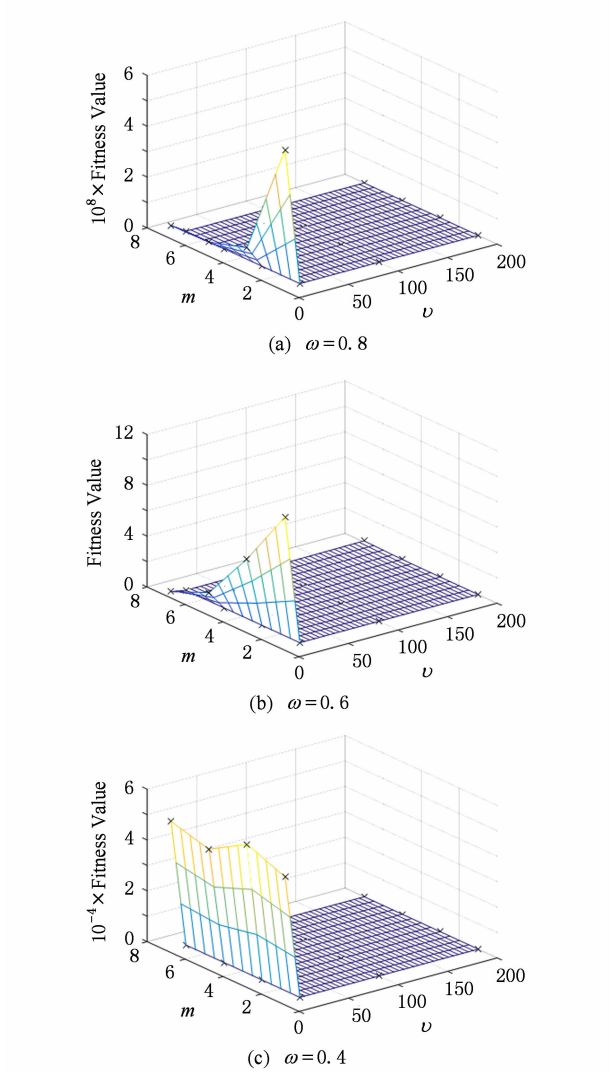
我们对参数 M,ω,υ 进行正交对比实验,隐空间大小 $M=[1,3,5,7]$;混合率 $\omega=[0.8,0.6,0.4]$;自由度 $\upsilon=[5,20,100,200]$.由于篇幅限制,文中只列出 F_5、F_7 和 F_{11} 函数的实验结果图.

从图 5~7 可以看出,当 $M\geqslant 3$ 时,算法的求解精度逐渐变好.但是,随着 M 的变大,算法的复杂度也加倍变大.所以,当 $M=3$ 时,可以较好地平衡求解精度和算法复杂度.当 $\omega=0.8$ 时,EDA-t 算法采用不同参数设置所取得的最差求解精度,也要比 $\omega=0.4$ 和 $\omega=0.6$ 所得到的求解精度好,表现了较强的鲁棒性.算法自由度 υ 从 5 变为 20 时,算法的求解精度明显变好.但是,当 υ 继续增大时,算法求解精度的变化不是很大.综上分析,在综合考虑算法计算代价与计算精度的平衡,算法参数可以取 $M\geqslant 3$,$\omega=0.8$ 和 $\upsilon=20$.

3.5 自由度自适应策略有效性分析

为了研究 t-分布的自由度自适应策略对算法性能的影响,在本节中,使用自由度固定的 EDA-t 算法和自由度自适应的 EDA-t 算法进行对比实验.EDA-t 中的固定自由度按 3.4 节中最优值进行设置,$\upsilon=20$.对比实验结果如表 5 所示.

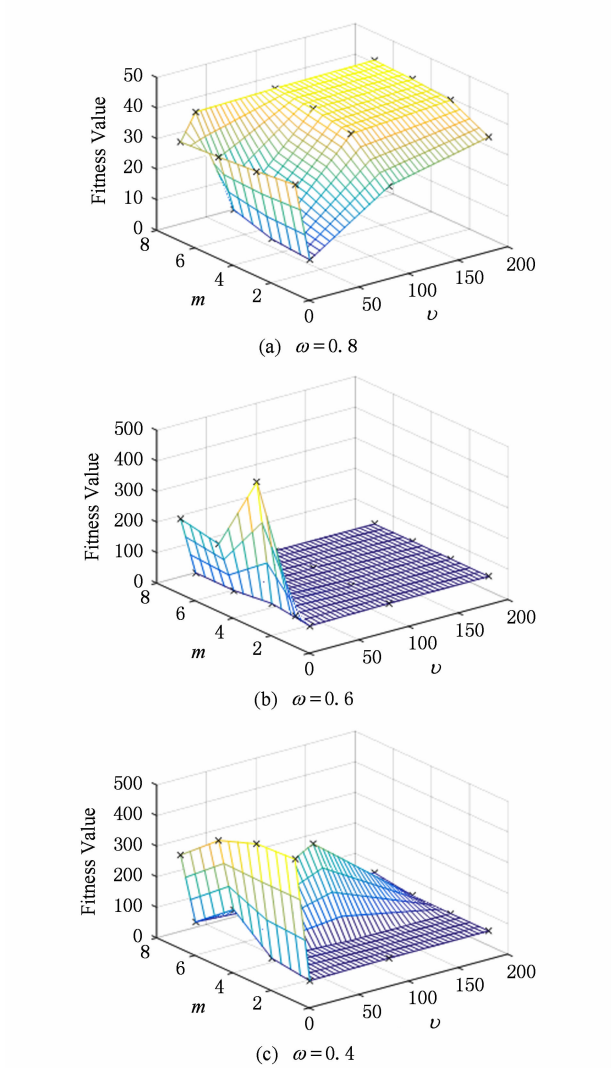
从表 5 可以看出,EDA-t 算法全面优于 EDA-t



(a) $\omega=0.8$ (b) $\omega=0.6$ (c) $\omega=0.4$

Fig. 5 Mesh grid of different parameters on functions F_5

图 5 不同参数在函数 F_5 上的曲面网状图



(a) $\omega=0.8$ (b) $\omega=0.6$ (c) $\omega=0.4$

Fig. 6 Mesh grid of different parameters on functions F_7

图 6 不同参数在函数 F_7 上的曲面网状图

算法,表明采取自由度自适应策略在整个演化过程中可以有效避免陷入局部最优.从而快速收敛到局部最优,并表现出较好的稳定性.

Table 5 Experimental Results of Fixed EDA-*t* and Adaptive EDA-*t*

表 5 固定自由度 EDA-*t* 和自适应自由度 EDA-*t* 算法对比

Fun	Mean Error±Std Dev	
	Fixed EDA-*t* ($\upsilon=20$)	Adaptive EDA-*t*
F_1	6.24E−30±1.12E−30−	**4.60E−84±1.12E−84**
F_2	2.01E−12±8.15E−13−	**8.21E−18±1.15E−18**
F_3	9.11E+00±1.73E+00−	**3.26E+00±8.47E−01**
F_4	9.31E+01±2.27E+01−	**6.45E+01±1.28E+01**
F_5	6.24E−16±1.01E−16−	**5.12E−18±9.23E−19**
F_6	6.14E−18±1.24E−18−	**2.89E−22±7.16E−23**
F_7	3.54E+03±7.26E+02≈	**1.00E+03±5.47E+02**
F_8	2.93E+03±6.36E+02≈	**1.03E+03±7.64E+02**
F_9	9.31E+07±2.75E+06−	**2.38E+07±9.42E+08**
F_{10}	1.13E+05±6.74E+04≈	**9.84E+04±1.12E+04**
F_{11}	5.29E+03±1.10E+03−	**2.18E+02±8.01E+01**
F_{12}	8.43E+03±1.02E+03−	**2.81E+02±8.36E+01**
F_{13}	8.51E+02±9.32E+01−	**3.71E+02±4.13E+01**
−/+/≈	10/0/3	

3.6 概率 PCA 的有效性分析

为了研究 ED-*t* 中使用概率 PCA 对算法性能的影响,在本节中,使用基于 PCA 的 EDA-*t* 算法和基于概率 PCA 的 EDA-*t* 算法进行对比实验.对比实验结果如表 6、表 7 所示.

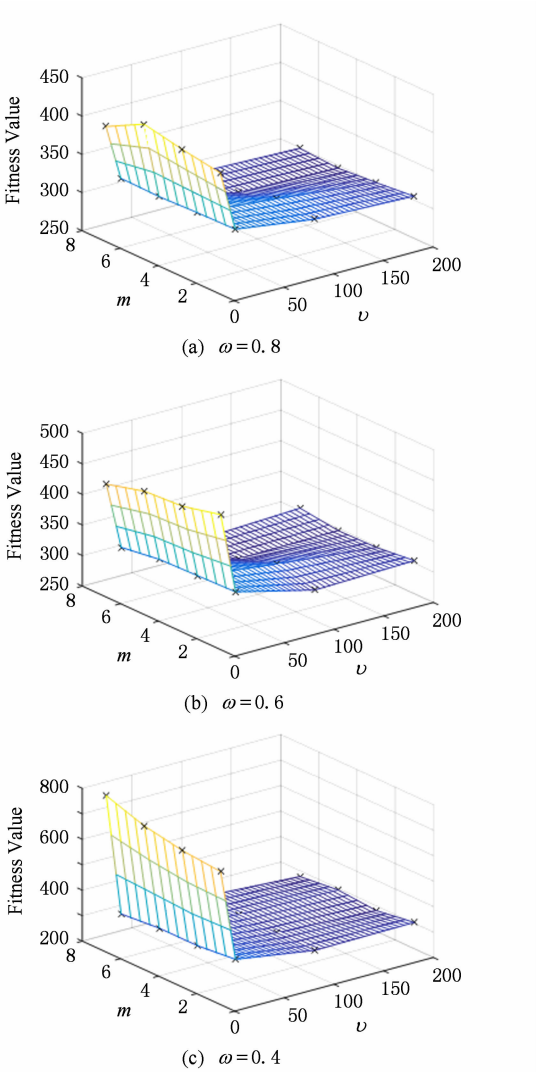


Fig. 7 Mesh grid of different parameters on functions F_{11}

图 7 不同参数在函数 F_{11} 上的曲面网状图

Table 6 Experimental Results of EDA-t Based on PCA and PPCA

表 6 算法运行结果对比

F_2	PCA	PPCA
30-D	2.54E−07	3.74E−07
50-D	1.12E−09	1.03E−09
80-D	1.16E−12	1.31E−12
100-D	3.81E−14	3.70E−14
300-D	1.72E−23	2.26E−23
500-D	3.59E−23	3.64E−23
800-D		7.36E−23
1000-D		1.05E−23

从表 6、表 7 中可以看出,当 EDA-t 使用 PCA 而不使用概率 PCA 的时候,算法求解精度变化不太大.但是,算法求解的时间随着问题维度的增加而急剧增加,直接导致算法在问题维度大于 500 维的时候,因为运算超时而无法求解.

Table 7 CPU Time of EDA-t Based on PCA and PPCA

表 7 算法运行时间对比 s

F_2	PCA	PPCA
30-D	3.01	1.94
50-D	6.67	4.04
80-D	12.13	6.87
100-D	13.84	12.08
300-D	896.03	75.58
500-D	4 533.47	89.35
800-D		95.37
1000-D		106.58

4 总 结

本文提出了一种新的基于自适应 t-分布的分布估计算法 EDA-t.它通过对 t-分布自由度自适应的调节,提高了算法的全局探索能力和跳出局部最优停滞区的能力.通过使用隐空间的转换,加快了算法的收敛速度.通过对 13 个大尺度全局优化进行测试,实验结果表明与当前主流算法相比,本文算法在求解的精度与收敛速度上具有更优的性能.另外,如何将算法与实际问题相结合是未来研究工作的重点.

参 考 文 献

[1] Muhlenbein H, Bendisch J, Voigt H M. From recombination of genes to the estimation of distributions Ⅱ. Continuous parameters [C] //Proc of the Int Conf on Parallel Problem Solving From Nature Ⅳ. Berlin: Springer, 1996: 188-197

[2] Hauschild M, Pelkan M. An introduction and survey of estimation of distribution algorithms [J]. Swarm & Evolutionary Computation, 2011, 1(3): 111-128

[3] Zhang Bo, Hao Jie, Ma Gang, et al. Mixture of probabilistic canonical correlation analysis [J]. Journal of Computer Research and Development, 2015, 52(7): 1463-1476 (in Chinese)
(张博, 郝杰, 马刚, 等. 混合概率典型相关性分析[J]. 计算机研究与发展, 2015, 52(7): 1463-1476)

[4] Sebag M, Ducoulombier A. Extending population-based incremental learning to continuous search spaces [C] //Proc of the Parallel Problem Solving From Nature Ⅴ. Berlin: Springer, 1998: 418-427

[5] Larraaga P, Lozano J A. Estimation of distribution algorithms. A new tool for evolutionary computation [J]. IEEE Trans on Instrumentation & Measurement, 2002, 64(5): 1140-1148

[6] Bonet J S D, JR C L I, Viola P A. MIMIC: Finding optima by estimating probability densities [C] //Advances in Neural Information Processing Systems. Cambridge: MIT Press, 1997: 424-430

[7] Pelikan M, Muehlenbein H. The bivariate marginal distribution algorithm [M]. Berlin: Springer, 1998: 521-535

[8] Pelikan M, Goldberg D E, Cant E. BOA: The Bayesian optimization algorithm [C] //Proc of the 1st Conf on Genetic and Evolutionary Computation. San Francisco: Morgan Kaufmann, 1999: 525-532

[9] Bosman P A N, Thierens D. Expanding from discrete to continuous estimation of distribution algorithms: The IDEA [C] //Proc of the Int Conf on Parallel Problem Solving From Nature Ⅵ. Berlin: Springer, 2000: 767-776

[10] Lu Qiang, Yao Xin. Clustering and learning gaussian distribution for continuous optimization [J]. IEEE Trans on Systems Man & Cybernetics Part C Applications & Reviews, 2005, 35(2): 195-204

[11] Shang Yunwei, Qiu Yuhuang. A note on the extended Rosenbrock function [J]. Evolutionary Computation, 2006, 14(1): 119-126

[12] Posik P. On the utility of linear transformations for population-based optimization algorithms [C] //Proc of the 16th Congress of the Int Federation of Automatic Control. Prague: IFAC, 2005: 281-286

[13] Dong Weishan, Chen Tianshi, TINO P, et al. Scaling up estimation of distribution algorithms for continuous optimization [J]. IEEE Trans on Evolutionary Computation, 2013, 17(6): 797-822

[14] Kab N A, Bootkrajang J, Durrant R J. Toward large-scale continuous EDA: A random matrix theory perspective [J]. Evolutionary Computation, 2016, 24(2): 255-291

[15] Fang Minquan, Zhang Weimin, Zhou Haifang. Parallel algorithm of fast independent component analysis for dimensionality reduction on many integrated core [J]. Journal of Computer Research and Development, 2016, 53(5): 1136-1146 (in Chinese)
(方民权, 张卫民, 周海芳. 集成众核上快速独立成分分析降维并行算法[J]. 计算机研究与发展, 2016, 53(5): 1136-1146)

[16] Zhang Cheng, Wang Dong, Shen Chuan, et al. Separable compressive imaging method based on singular value decomposition [J]. Journal of Computer Research and Development, 2016, 53(12): 2816-2823 (in Chinese)
(张成, 汪东, 沈川, 等. 基于奇异值分解的可分离压缩成像方法[J]. 计算机研究与发展, 2016, 53(12): 2816-2823)

[17] Bishop C M. Pattern recognition and machine learning [M]. Berlin: Springer, 2006, 571-586

[18] Yang Zhenyu, Tang Ke, Yao Xin. Self-adaptive differential evolution with neighborhood search [C] //Proc of IEEE World Congress on Computational Intelligence, NJ: IEEE, 2008: 1110-1116

[19] Yang Zhenyu, Tang Ke, Yao Xin. Large scale evolutionary optimization using cooperative coevolution [J]. Information Sciences, 2008, 178(15): 2985-2999

Wang Yufeng, born in 1982. PhD candidate of Wuhan University. Student member of CCF. His main research interests include machine learning, evolutionary computation, data dimension reduction.

Dong Wenyong, born in 1973. Professor and PhD supervisor in Wuhan University. His main research interests include evolutionary computation, system control, machine learning, parallel computing.

Dong Xueshi, born in 1985. PhD candidate of Wuhan University. His main research interests include evolutionary computation, machine learning.

Wang Hao, born in 1972. PhD. Research Professor in the Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan. His main research interests include evolutionary computation, machine learning.