2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)
March 29–31, 2018, Xiamen, China

Improving Estimation of Distribution Algorithms with Heavy-Tailed Student's t Distributions

Bin Liu
School of Computer Science
Nanjing University of Posts and Telecommunications
Jiangsu Key Lab of Big Data Security & Intelligent Processing
Nanjing, Jiangsu, China
bins@ieee.org

Shi Cheng
School of Computer Science
Shaanxi Normal University
Xi'an, China
cheng@snnu.edu.cn

Yuhui Shi
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
shiyh@sustc.edu.cn

***Abstract*—As a derivative-free optimization method, the estimation of distribution algorithm (EDA) usually leverages a Gaussian or a Gaussian mixture model to represent the distribution of promising solutions that have been found so far. This paper investigates the application of an alternative model, namely the heavier-tailed Student's *t* distribution, to implement EDA. Two corresponding algorithms, termed ESTDA and EMSTDA, are developed, respectively. The ESTDA employs a single Student's *t* model to represent the distribution of the promising solutions. The EMSTDA uses a mixture of Student's *t* models to take account of hard multimodal cases. These methods are evaluated through extensive and in-depth numerical experiments using over a dozen of benchmark objective functions. Empirical results show that they provide remarkably better performance than their Gaussian counterparts in most cases under consideration.**

***Keywords*—estimation of distribution algorithm; EDA; derivative-free optimization; Student's t distribution; mixture**

I. Introduction

In this paper, we are concerned with the global optimization problem as follows

$$\min_{x\in\chi} f(x) \tag{1}$$

where χ denotes the nonempty solution space defined in $\mathbb{R}^n$, and f: $\chi \to \mathbb{R}$ is a continuous real-valued function. The basic assumption here is that f is bounded on χ, which means $\exists f_l > -\infty, f_u < \infty$ such that $f_l \le f(x) \le f_u$, $\forall x \in \chi$. We denote the minimal function value as f^*, i.e., there exists an x^* such that $f(x) \ge f(x^*) = f^*$, $\forall x \in \chi$.

We focus on a type of derivative-free evolutionary computation (EC) method, namely the estimation of distribution algorithm (EDA) [1], to address the above problem. In contrast with the other EC methods equipped with meta-heuristics inspired operations, the EDA is characterized by its unique operations that leverage a probabilistic model $p(\cdot|\theta)$ to lead the search towards promising areas of the solution space [2], [3]. In EDAs, the model $p(\cdot|\theta)$ is tuned through adapting the value of its parameter θ with the hope to learn some useful structural information of $f(x)$ that is beneficial for conducting a more effective search. The basic operation of a generic EDA is presented in Algorithm 1.

This work was partly supported by National Natural Science Foundation (NSF) of China (Nos. 61571238,61773119, 61703256, and 61771297), and Nanjing University of Posts and Telecommunications Yancheng Big Data Research Institute.

Algorithm 1: The operations of a basic EDA

Input: population size $N \in \mathbb{N}$, the selection size $M \in \mathbb{N}, M < N$, and θ_1, an initialized value of θ.
Output: $\hat{x}$ and $f(\hat{x})$, which represent estimates of x^* and $f(x^*)$, respectively.

1 Initialization: Set $\theta = \theta_1$ and then draw a random sample $\hat{x}$ from $p(\cdot|\theta)$. Set the iteration index $k = 1$;
2 **while** *the stopping criterion is not met* **do**
3 Sample N individuals from $p(x|\theta)$, denote them by $x_1, x_2, \ldots, x_N$;
4 Calculate the objective function values of the individuals: $y_i = f(x_i)$, $i = 1, 2, \ldots, N$;
5 Find the minimum value y_{min} in $\{y_1, \ldots, y_N\}$; Get x_{min} from $\{x_i\}_{i=1}^N$ that satisfies $f(x_{min}) = y_{min}$;
6 **if** $y_{min} < f(\hat{x})$ **then**
7 Set $\hat{x} = x_{min}$, and $f(\hat{x}) = y_{min}$;
8 Select the best M individuals from $\{x_i\}_{i=1}^N$ based on $\{y_i\}_{i=1}^N$;
9 Update the value of θ with the goal to approximate the distribution of the selected individuals in the above step.

In traditional EDAs, the model $p(\cdot|\theta)$ is usually specified to be Gaussian, namely $p(x|\theta) = \mathcal{N}(x|\mu, \Sigma)$, where $\theta \triangleq \{\mu, \Sigma\}$, μ and Σ denote the mean and covariance, respectively. Then the parameter updating operation presented in Step 9 of Algorithm 1 is just

$$\mu = \frac{\sum_{j=1}^{M} x_j}{M} \tag{2}$$

$$\Sigma = \frac{\sum_{j=1}^{M} (x_j - \mu)(x_j - \mu)^T}{M - 1}, \tag{3}$$

where A^T denotes transposition of A. All vectors involved here and hereafter are assumed to be column vectors.

EDAs fall within the paradigm of model-based or learning guided optimization [3], [4]. In this regard, different EDAs can be distinguished by the class of probabilistic models used. An investigation of the boundaries of the effectiveness of EDAs shows that the limits of EDAs are mainly imposed by the probabilistic model they rely on. The more complex the model, the greater ability it offers to capture possible interactions among the variables of the problem. However, increasing the complexity of the model usually leads to more computational cost. The EDA paradigm could admit any type of probabilistic model, among which the most popular model class is Gaussian. The simplest EDA algorithm just employs univariate Gaussian models, which regard all design variables to be independent to each other [5], [6]. The simplicity of such models makes the corresponding algorithms easy to implement, while their effectiveness halts when the design variables have strong interdependencies. To get around this limitation, several multivariate Gaussian-based EDAs (Gaussian-EDAs) have been proposed [5], [7], [8]. To represent variable linkages elaborately, Bayesian networks (BNs) are usually adopted in the framework of multivariate Gaussian-EDA, while the learning of the BN structure and parameters can be very time-consuming [9], [10]. To handle hard multimodal and deceptive problems in a more appropriate way, Gaussian mixture model (GMM) based EDAs (GMM-EDAs) have also been proposed [11], [12]. It is worthwhile to mention that the more complex the model used, the more likely it encounters model overfitting, which may mislead the search procedure [13].

Our contribution in this paper lies in illustrating how to leverage multivariate Student's *t* models, instead of Gaussian, to facilitate searching of the global optimum with aid of the EDA-type mechanism. In particular, we derive two novel EDAs, entitled as Estimation of Student's *t* Distribution Algorithm (ESTDA) and Estimation of Mixture of Student's *t* Distribution Algorithm (EMSTDA), respectively. We show that they outperform their Gaussian counterparts in performance through evaluations with over a dozen of benchmark objective functions. The heavier-tailed property renders the Student's *t* distribution a desirable choice to design Bayesian simulation techniques such as the adaptive importance sampling (AIS) algorithms [14]–[16], while the most related work in literature includes the posterior exploration based Sequential Monte Carlo (PE-SMC) algorithm [14] and the annealed adaptive mixture importance sampling algorithm [17]. In addition, the faster Evolutionary programming (FEP) algorithm proposed in [18] is similar in the spirit of replacing the Gaussian distribution with a heavier-tailed alternative.

The remainder of this paper is organized as follows. In Section II, we present the proposed algorithms. In Section III, we present results from the numerical experiments, and finally, in Section IV we conclude the paper.

II. Student's t distribution based EDA

In this section, we present the proposed algorithms, ESTDA and EMSTDA, in detail. These algorithms attempt to employ the Student's *t* model and the Student's *t* mixture model instead of their Gaussian counterparts in the EDA framework. To begin with, we give a brief introduction to the Student's *t* distribution.

Algorithm 2: The EM procedure for estimating parameters of a mixture of Student's t model, which is employed in the EDA paradigm presented in Algorithm 1

Input: The individuals $\{x_j\}_{j=1}^M$, current parameter value $\theta = \{w_l, \mu_l, \Sigma_l\}_{l=1}^L$ and the admitted smallest weight W of a mixing component (we set $W = 0.02$ as a default value for use in our experiments).

Output: Updated value of θ, denoted as $\theta_{new} = \{w_{new,l}, \mu_{new,l}, \Sigma_{new,l}\}_{l=1}^L$

Initialization: Set the iteration index $i = 1$;

while *the stopping criterion is not met* **do**

The Expectation step:

$$w_{new,l} = \sum_{j=1}^{M} \epsilon(l|x_j)/M, \quad l = 1, \ldots, L, \tag{4}$$

where $\epsilon(l|x_j)$ denotes the probability of the event individual x_j belonging to the lth mixing component, which is calculated as follows

$$\epsilon(l|x_j) = \frac{w_l \mathcal{S}(x_j|\mu_l, \Sigma_l, v)}{\sum_{l=1}^{L} w_l \mathcal{S}(x_j|\mu_l, \Sigma_l, v)}. \tag{5}$$

Delete mixing components whose mixing weights are smaller than W. Update the value of L and increase weights of the remaining mixing components proportionally to guarantee that their summation is 1;

The Maximization step: See Eqn.(8) in the next page;

Set $\theta_{new} = \{w_{new,l}, \mu_{new,l}, \Sigma_{new,l}\}_{l=1}^L$;

Set $\theta = \theta_{new}$ and let $i = i + 1$;

A. Student's t distribution

Suppose that x is a d dimensional random variable that follows the multivariate Student's t distribution, denoted by $\mathcal{S}(\cdot|\mu, \Sigma, v)$, where μ denotes the mean, Σ a positive definite inner product matrix and $v \in (0, \infty]$ is the degrees of freedom (DoF). Then the density function of x is:

$$\mathcal{S}(x|\mu, \Sigma, v) = \frac{\Gamma(\frac{v+d}{2})|\Sigma|^{-0.5}}{(\pi v)^{0.5d}\Gamma(\frac{v}{2})\{1 + M_d(x, \mu, \Sigma)/v\}^{0.5(v+d)}}, \tag{6}$$

where

$$M_d(x, \mu, \Sigma) = (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{7}$$

denotes the Mahalanobis squared distance from x to μ with respect to Σ, A^{-1} the inverse of A and $\Gamma(\cdot)$ the gamma

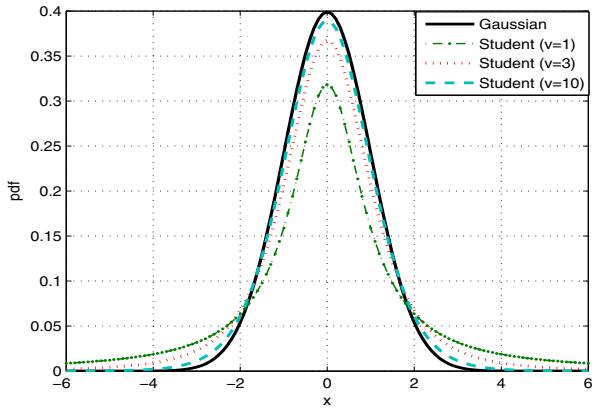


Fig. 1. Comparison between the univariate Gaussian and Student's t distributions (represented by "Student" in the figure). All distributions have zero mean with a fixed variance 1. v denotes the DoF.

$$\begin{cases} \mu_{new,l} = \frac{\sum_{j=1}^{M} \epsilon(l|x_j) x_j (v+d)/(v+M_d(x_j,\mu_{old,l},\Sigma_{old,l}))}{\sum_{j=1}^{M} \epsilon(l|x_j)(v+d)/(v+M_d(x_j,\mu_{old,l},\Sigma_{old,l}))}, \ l = 1,\ldots,L, \\ \Sigma_{new,l} = \frac{\sum_{j=1}^{M} \epsilon(l|x_j)(x_j-\mu_{new,l})(x_j-\mu_{new,l})^T (v+d)/(v+M_d(x_j,\mu_{old,l},\Sigma_{old,l}))}{\sum_{j=1}^{M} \epsilon(l|x_j)}, \ l = 1,\ldots,L. \end{cases} \tag{8}$$

function. For ease of understanding, we give a graphical description of univariate Student's t possibility density functions (pdfs) corresponding to different DoFs in comparison with a standard Gaussian pdf in Fig.1. It is shown that the Student's t distribution is symmetric and bell-shaped, like the Gaussian distribution, but has heavier and longer tails. The smaller the DoF v, the heavier the tails. Due to its long and flat tails, the Student's t pdf is more likely to generate an individual further away from its mean location than the Gaussian distribution.

B. ESTDA

In ESTDA, the probabilistic model $p(\cdot|\theta)$ in Algorithm 1 is specified to be a Student's t distribution, namely $p(\cdot|\theta) = \mathcal{S}(x|\mu,\Sigma,v)$, where $\theta \triangleq \{\mu,\Sigma\}$, and v is specified beforehand as a constant. First, let us figure out, given the parameter value θ, how to sample an individual x from $p(\cdot|\theta)$. It consists of two steps: simulate a random draw τ from the gamma distribution, whose shape and scale parameters are set identically to $v/2$; and then sample x from the Gaussian distribution $\mathcal{N}(\cdot|\mu,\Sigma/\tau)$.

Now let us focus on, given a set of selected individuals $\{x_j\}_{j=1}^M$, how to update the parameters of the Student's t distribution in an optimal manner in terms of maximizing the likelihood. As mentioned above, in the generation of $\{x_j\}_{j=1}^M$, a corresponding set of gamma-distributed variables $\{\tau_j\}_{j=1}^M$ is used. We can record these gamma variables and then use them to easily derive an ML estimate for the parameters of the Student's t distribution [19], [20]:

$$\mu = \frac{\sum_{j=1}^{M} \tau_j x_j}{\sum_{i=1}^{M} \tau_i}, \tag{9}$$

$$\Sigma = \frac{\sum_{j=1}^{M} \tau_j (x_j-\mu)(x_j-\mu)^T}{\sum_{i=1}^{M} \tau_i}. \tag{10}$$

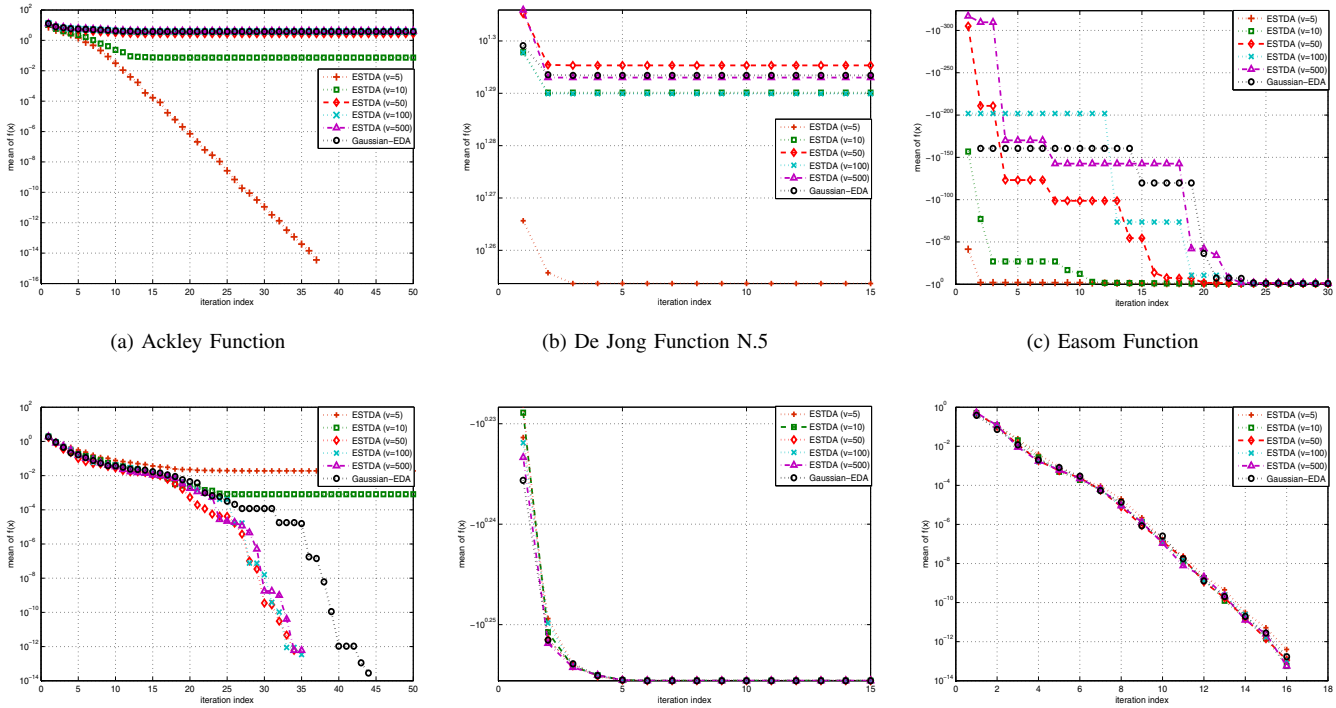
C. EMSTDA

The EMSTDA employs a mixture of Student's t distributions to play the role of the probabilistic model $p(\cdot|\theta)$. Now we have $p(x|\theta) = \sum_{l=1}^{L} w_l \mathcal{S}(x|\mu_l,\Sigma_l,v)$, where $\theta \triangleq \{w_l,\mu_l,\Sigma_l\}_{l=1}^L$, $\sum_{l=1}^{L} w_l = 1$, and v is specified beforehand as a constant. Given the selected individuals $\{x_j\}_{j=1}^M$, we resort to the Expectation-Maximization (EM) method to update parameter values of the mixture of Student's t distributions. The related operations are presented in Algorithm 2 in the above page. The EM-based parameter estimation for the mixture of Student's t model can also be found in [14]–[16], where it is performed on a weighted sample set but the samples are equally weighted here. We add a components deletion operation at the end of Expectation step. Such deletion operation is suggested in [14], [15], which show that deleting components with extremely small mixing weights can avoid consumption of unnecessary computing resources to negligible components and numerical issues such as singularity of the covariance matrix.

III. Performance Evaluation

In this section, we evaluate the presented ESTDA and EMSTDA using a number of benchmark objective functions, which are designed and commonly used in the literature for testing and comparing optimization algorithms. See the Appendix Section in [21] for definitions of the involved functions.

The baseline methods, Gaussian-EDA [5] and GMM-EDA [11], [12], are also involved in the comparison. As the main goal here is to evaluate the benefits resulted from the heavier



(a) Ackley Function (b) De Jong Function N.5 (c) Easom Function

(d) Rastrigin Function (e) Michalewicz Function (f) Lèvy N.13 Function

Fig. 2. Convergence behaviors of ESTDAs and Gaussian-EDA. The Y coordinate denotes the averaged objective function values of the best solutions obtained from 30 independent runs of each algorithm. All test functions involved here are 2D.

tails of the Student's t distribution in exploring the solution space and searching the global optimum, we only consider cases with $d \leq 10$ here for ease of performance comparison, while the obtained empirical conclusion may be extended to higher dimensional cases straightforward.

A. Experimental study for ESTDA

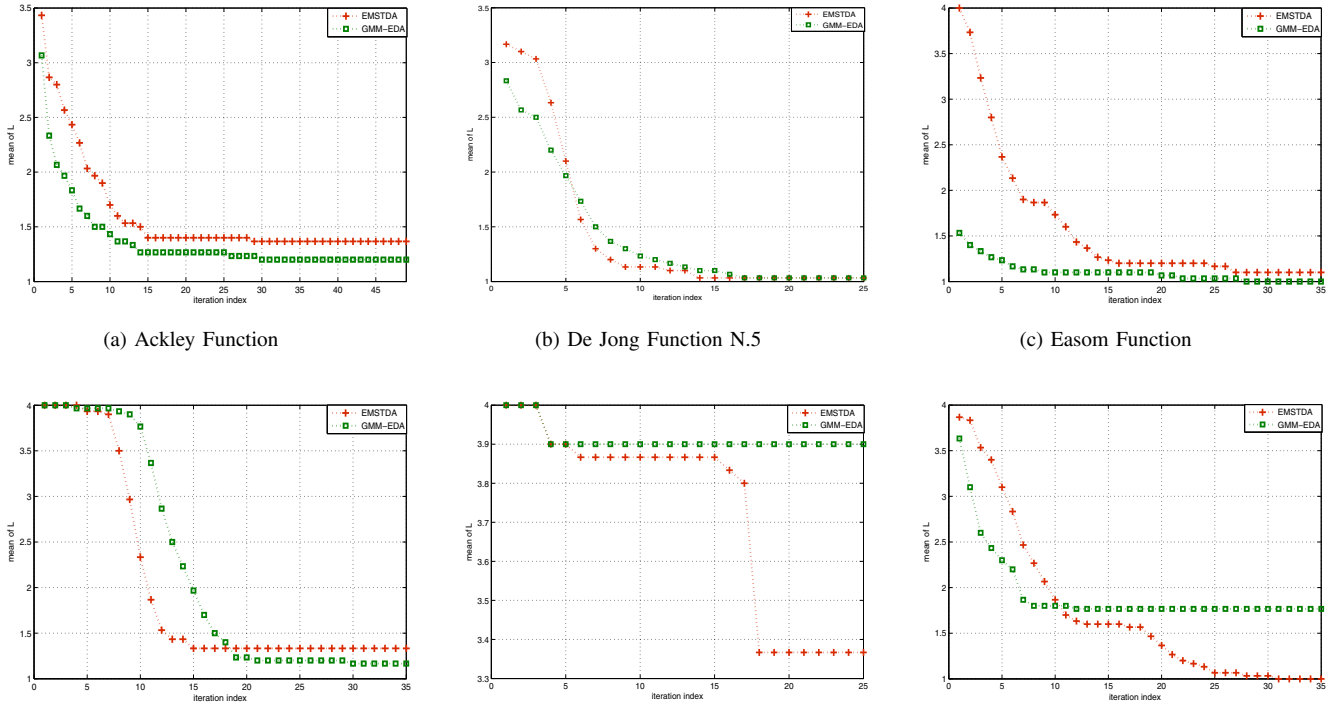
As described in Subsection II-A, the degree of difference between the shapes of a Student's t and a Gaussian distribution mainly depends on the value of DoF v. This Subsection is dedicated to investigating the influence of v on the performance of ESTDA through empirical studies. We consider 6 objective functions here, including the Ackley, De Jong Function N.5, Easom, Rastrigin, Michalewicz and the Lèvy N.13 functions. We only consider the 2D case (i.e., d=2). We consider 5 ESTDAs, corresponding to $v = 5, 10, 50, 100, 500$, respectively. The sample size and selection size are set at $N = 1000$ and $M = 200$, respectively. The Gaussian-EDA is involved as the benchmark for comparison. We run each algorithm 30 times independently, record the best solutions it obtained at the end of each iteration, and plot the means of the best fitness function values over these runs in Fig.2. It is shown that ESTDA with $v = 5$ performs strikingly better than the others for the Ackley, De Jong Function N.5, and Easom Functions. ESTDAs with $v = 50$ and $v = 500$ outperform the others significantly in handling the Rastrigin function, and all algorithms give the similar convergence rate for the remaining functions. To summarize, the ESTDA outperforms the Gaussian-EDA markedly for four problems and performs similarly as the Gaussian-EDA for the other two problems considered.

B. Performance evaluation for ESTDA and EMSTDA

In this subsection, we perform a thorough performance evaluation for ESTDA and EMSTDA. The Gaussian counterpart algorithms, Gaussian-EDA and GMM-EDA, are also included for comparison.

To make a representative evaluation, we consider in total 17 objective functions listed in Table II in the last page. For all but the Rastrigin and Michalewicz functions, we consider 2D cases. For the Rastrigin and Michalewicz functions, we also consider 5D and 10D cases. The population size per iteration N is set to be 10^3, 10^4, and 10^5 for 2D, 5D, 10D cases, respectively. The selection size M is fixed to be $0.2 \times N$ for all cases involved.

We fix the value of the DoF v to be 5 for all test functions, except that, for problems involving the Rastrigin function, v is set to be 50, since it gives better convergence behavior as reported in Section III-A. For every algorithm, the iterative EDA procedure terminates when the iteration number k in Algorithm 1 is bigger than 50. For EMSTDA and GMM-EDA, the EM procedure terminates when its iteration number, i.e., i in Algorithm 2, exceeds 2. Each algorithm is run 30 times independently. Then we calculate the average and standard



(a) Ackley Function (b) De Jong Function N.5 (c) Easom Function

(d) Rastrigin Function (e) Michalewicz Function (f) Lèvy N.13 Function

Fig. 3. The mean number of the survival mixing components L per iteration in the mixture model of GMM-EDA and EMSTDA, averaged over 30 independent runs of each algorithm.

error of the converged fitness values. The results are presented in Table II in the next page, wherein the best solution for each test function is marked with bold font. Then we count for each algorithm how many times it outputs a better solution as compared with all the other algorithms and record the result in Table I. We see that the EMSTDA gets the highest score 7, followed by score 5 obtained by ESTDA, while their Gaussian counterparts only obtain relatively lower scores 0 and 2, respectively. This result further coincides with our argument that the Student's t distribution is preferable than the Gaussian for use in designing EDAs. We also see that, between ESTDA and EMSTDA, the latter performs better than the former. This demonstrates that using mixture models within a Student's t based EDA is likely to be able to bring additional advantages. What slightly surprises us in Table I is that the GMM-EDA seems to perform worse than the Gaussian-EDA, while after a careful inspection of Table II, we see that GMM-EDA beats Gaussian-EDA strikingly in cases involving the Ackley, Dejong N.5, Easom, Michalewicz 10D, Eggholder, Griewank, Holder table, Schwefel and Rosenbrock functions, while for cases involving the Michalewicz 2D, Lèvy N.13, Cross-in-tray, Drop-wave, Lèvy, Schaffer, Shubert and Perm functions, it performs identically or similarly as the Gaussian-EDA. Hence, in summarize, GMM-EDA actually performs better than Gaussian-EDA in this experiment.

To investigate the effect of the component deletion operator used in Algorithm 2, we record the number of survival mixing components per iteration of GMM-EDA and EMSTDA and calculate its mean averaged over 30 independent runs of each algorithm. The result is depicted in Fig. 3. It is shown that, for both algorithms, the mean number of survival components decreases as the algorithm iterates on.

TABLE I. A simple statistics devised for abstracting results shown in Table II. Each number in the table represents for how many test problems the corresponding algorithm outperforms all the other competitors. Note that for some test problems, e.g, the Cross-in-tray function case, more than one algorithms give the identical best solution. For such cases, we regard that no algorithm outperforms all the other competitors.

ESTDA	EMSTDA	Gaussian-EDA	GMM-EDA
5	7	2	0

IV. Concluding Remarks

EDA is a widely used derivative-free optimization framework. Most of existing EDAs are developed based on the Gaussian density models, while it lacks investigations to check whether there is preferable choice than Gaussian. To this end, we investigated the heavier-tailed Student's t distribution with its applications in the EDA framework. We derived two novel EDAs, namely ESTDA and EMSTDA, which leverage

TABLE II. Convergence results yielded from 30 independent runs of each algorithm on benchmark test problems. $a \pm b$ in the table denotes that the average and standard error of the best fitness values obtained from 30 independent runs of the corresponding algorithm are a and b, respectively. The best solution for each problem is marked with bold font.

Test Problems		Goal:$f(x^\star)$	ESTDA	EMSTDA	Gaussian-EDA	GMM-EDA
Ackley	2D	0	**0**±0	0.01±0.0701	3.18±1.30	1.53±1.82
Dejong N.5	2D	1	18.12±1.81	**3.24**±2.64	19.65±1.17	6.47±6.35
Easom	2D	-1	-0.93±0.25	**-0.96**±0.19	0.23±0.42	-0.32±0.46
Rastrigin	2D	0	**0**±0	0.005±0.02	**0**±0	0.02±0.06
	5D	0	**0**$\pm 2.13 \times 10^{-12}$	0.66±0.80	0$\pm 1.70 \times 10^{-11}$	0.36±0.63
	10D	0	**0.04**±0.05	0.54±0.60	0.04±0.03	0.53±0.83
Michalewicz	2D	-1.80	**-1.80**±0	**-1.80**±0	**-1.80**±0	**-1.80**±0
	5D	-4.69	**-4.69**$\pm 9.36 \times 10^{-9}$	-4.64±0.06	-4.65±0.01	-4.65±0.02
	10D	-9.66	**-9.54**±0.048	-9.42±0.21	-9.10±0.14	-9.14±0.14
Lèvy N.13	2D	0	**0**±0	0.00±0.01	**0**±0	0.00±0.02
Cross-in-tray	2D	-2.06	**-2.06**±0	**-2.06**±0	**-2.06**±0	**-2.06**±0
Drop-wave	2D	-1	-0.99±0.013	-0.99±0.01	**-0.10**±0.00	-0.99±0.01
Eggholder	2D	-959.64	-588.92±75.24	**-731.80**±145.54	-560.73±4.21	-686.52±147.54
Griewank	2D	0	19.58±3.79	**1.52**±5.56	30.42±1.04	15.79±16.30
Holder table	2D	-19.21	-19.08±0.19	**-19.21**±0	-19.19±0.08	**-19.21**±0
Lèvy	2D	0	**0**±0	**0**±0	**0**±0	0$\pm 1.06 \times 10^{-4}$
Schaffer	2D	0	0$\pm 1.71 \times 10^{-6}$	0.00$\pm 4.53 \times 10^{-4}$	**0**±0	0$\pm 2.02 \times 10^{-4}$
Schwefel	2D	0	368.31±75.48	**184.28**±118.46	436.82±2.55	247.06±123.30
Shubert	2D	-186.73	-186.73$\pm 4.05 \times 10^{-13}$	**-186.73**$\pm 1.64 \times 10^{-13}$	-186.73$\pm 1.38 \times 10^{-4}$	-186.73$\pm 2.48 \times 10^{-5}$
Perm	2D	0	**0**±0	**0**±0	**0**±0	0$\pm 4.00 \times 10^{-6}$
Rosenbrock	2D	0	0.04±0.04	**0.00**±0.013	0.05±0.06	0.02±0.04

a single Student's t model and a Student's t mixture model, respectively, to lead search towards promising solutions. Both algorithms are easy to implement, the main operation consisting of sampling from and then fitting a Student's t model (or a Student's t mixture model) to a set of samples. Experimental results illustrate that the Student's t based EDAs perform remarkably better than their Gaussian counterparts in most cases under consideration. It also shows that mixture modeling is a useful strategy to improve performance of EDAs.

REFERENCES

[1] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2001.

[2] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," Computational optimization and applications, vol. 21, no. 1, pp. 5–20, 2002.

[3] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," Swarm and Evolutionary Computation, vol. 1, no. 3, pp. 111–128, 2011.

[4] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," Applied Soft Computing, vol. 55, pp. 154–167, 2017.

[5] P. Larranaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2002, vol. 2.

[6] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in *International Conf. on Parallel Problem Solving from Nature*. Springer, 1998, pp. 418–427.

[7] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *The Genetic and Evolutionary Computation Conference Workshop Program*. Citeseer, 2000, pp. 201–204.

[8] P. A. Bosman and D. Thierens, "Expanding from discrete to continuous estimation of distribution algorithms: The IDEA," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2000, pp. 767–776.

[9] W. Dong, T. Chen, P. Tiňo, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization," IEEE Trans. on Evolutionary Computation, vol. 17, no. 6, pp. 797–822, 2013.

[10] Q. Zhang, J. Sun, E. Tsang, and J. Ford, "Hybrid estimation of distribution algorithm for global optimization," Engineering computations, vol. 21, no. 1, pp. 91–107, 2004.

[11] Q. Lu and X. Yao, "Clustering and learning Gaussian distribution for continuous optimization," IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 35, no. 2, pp. 195–204, 2005.

[12] C. W. Ahn and R. S. Ramakrishna, "On the scalability of real-coded Bayesian optimization algorithm," IEEE Trans. on Evolutionary Computation, vol. 12, no. 3, pp. 307–322, 2008.

[13] H. Wu and J. L. Shapiro, "Does overfitting affect performance in estimation of distribution algorithms," in *Proc. of the 8th annual conf. on Genetic and evolutionary computation*. ACM, 2006, pp. 433–434.

[14] B. Liu, "Posterior exploration based Sequential Monte Carlo for global optimization," Journal of Global Optimization, vol. 69, no. 4, pp. 847–868, 2017.

[15] ——, "Adaptive annealed importance sampling for multimodal posterior exploration and model selection with application to extrasolar planet detection," The Astrophysical Journal Supplement Series, vol. 213, no. 14, pp. 1–16, 2014.

[16] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Adaptive importance sampling in general mixture classes," Statistics and Computing, vol. 18, no. 4, pp. 447–459, 2008.

[17] B. Liu and C. Ji, "A general algorithm scheme mixing computational intelligence with bayesian simulation," in *Proc. of the 6th Int'l Conf. on Advanced Computational Intelligence (ICACI)*. IEEE, 2013, pp. 1–6.

[18] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," IEEE Trans. on Evolutionary Computation, vol. 3, no. 2, pp. 82–102, 1999.

[19] C. Liu, "ML estimation of the multivariate t distribution and the EM algorithm," Journal of Multivariate Analysis, vol. 63, no. 2, pp. 296–312, 1997.

[20] C. Liu and D. B. Rubin, "ML estimation of the t distribution using EM and its extensions, ECM and ECME," Statistica Sinica, pp. 19–39, 1995.

[21] B. Liu, S. Cheng, and Y. Shi, "Student's t distribution based estimation of distribution algorithms for derivative-free global optimization," arXiv preprint arXiv:1509.08870, 2016.