Global Optimization with the Gaussian Polytree EDA

Ignacio Segovia Domínguez, Arturo Hernández Aguirre,
and Enrique Villa Diharce

Center for Research in Mathematics
Guanajuato, México
{ijsegoviad,artha,villadi}@cimat.mx

Abstract. This paper introduces the Gaussian polytree estimation of distribution algorithm, a new construction method, and its application to estimation of distribution algorithms in continuous variables. The variables are assumed to be Gaussian. The construction of the tree and the edges orientation algorithm are based on information theoretic concepts such as mutual information and conditional mutual information. The proposed Gaussian polytree estimation of distribution algorithm is applied to a set of benchmark functions. The experimental results show that the approach is robust, comparisons are provided.

Keywords: Polytrees, Estimation of Distribution Algorithm, Optimization.

1 Introduction

The polytree ia a graphical model with wide applications in artificial intelligence. For instance, in belief networks the polytrees are the de-facto graph because they support probabilistic inference in linear time [13]. Other applications make use of polytrees in a rather similar way, that is, polytrees are frequently used to model the joint probability distribution (JPD) of some data. Such JPD is also called a factorized distribution because the tree encodes a joint probability as a product of conditional distributions.

In this paper we are concerned with the use of polytrees and their construction and simulation algorithms. Further more, we asses the improvement that polytrees bring to the performance of Estimation of Distribution Algorithms (EDAs). As mentioned the polytree graphs have been applied by J. Pearl to belief networks [13], but also Acid and de Campos researched them in causal networks [1], [14]. More recently, M. Soto applied polytrees to model distributions in EDAs and came up with the polytree approximation distribution algorithm, known as PADA [11]. However, note that in all the mentioned approaches the variables are binary. The goal of this paper is to introduce the polytree for continuous variables, that is, a polytree in continuous domain with Gaussian variables and its application to EDAs for optimization. The proposed approach is called the Gaussian Polytree EDA. Polytrees with continuous variables have been studied

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 165–176, 2011.

by Ouerd [12], [9]. In this paper we extend a poster presented [16] and we further develop the work of Ouerd [12]. We introduce two new algorithmic features to the gaussian polytree: 1) a new orientation principle based on conditional mutual information. We also prove that our approach is correct, 2) overfitting control of the model through a comparison of conditional and marginal mutual information strengths. The determination of the threshold value is also explained.

This paper is organized as follows. Section 2 describes two polytree algorithms in discrete variables; Section 3 explains how to build a Gaussian polytree while Section 4 provides the implementation details. Section 5 describes two sets of experiments and provides a comparison with other related approaches. Section 6 provides the conclusions and lines of future research.

2 Related Work

A polytree is a directed acyclic graph (DAG) with no loops when the edges are undirected (only one path between any two nodes) [6],[8]. For binary variables the polytree approximation distribution algorithm (PADA) is the first work to propose the use of polytrees in estimation distribution algorithm [11]. The construction algorithm of PADA uses (marginal) mutual information and conditional mutual information as a measure of the dependency. Thus, a node X_k is made head to head whenever the conditional mutual information $CMI(X_i, X_j|X_k)$ is greater than the marginal mutual information $MI(X_i, X_j))$. Thus, the head to head node means that the information shared by two nodes X_i, X_j increases when the third node X_k is included. For overfitting control two parameters ϵ_1, ϵ_2 aim to filter out the (weak) dependencies. However no recommendations about how to set these parameters is given in the PADA literature.

A Gaussian polytree is a factorized representation of a multivariate normal distribution [10],[4]. Its JPDF is a product of Gaussian conditional probabilites times the product of the probabilities of the root nodes (R), as follows: $JPDF(X_1, X_2, \ldots, X_n) = \prod_{\forall i \in R} P(X_i) \prod_{\forall j \notin R} P(X_j|pa(X_j))$. A recent approach uses a depth first search algorithm for edge orientation [9]. Based on the previous work of Rebane and Pearl [15],[13], Ouerd at al. assume that a Chow & Liu algorithm is ran to deliver a dependence tree from the data [9]. Then they propose to orient the edges by traversing the dependence tree in a depth first search order. Articulation points and causal basins must be detected first. With their approach they try to solve four issues (not completely solved by Rebane and Pearl) such as how to traverse the tree, and what to do with the edges already traversed. For edge orientation their algorithm performs a marginal independence test on the parents X and Y of a node Z to decide if Z has X and Y as parents. If they are independent the node Z is a head to head node.

3 Building the Gaussian Polytree

In the following we describe the main steps needed to construct a Gaussian polytree.

1. The Gaussian Chow & Liu tree. The first step to construct a Gaussian polytree is to construct a *Gaussian Chow & Liu dependence tree* (we use the same approach of the binary dependence tree of Chow & Liu [3]). Recall *mutual information* is the measure to estimate dependencies in Chow & Liu algorithm. The algorithm randomly chooses a node and declares it the root. Then the Kruskal algorithm is used to create a maximum weight spanning tree. The tree thus created maximizes the total mutual information, and it is the best approximation to the true distribution of the data whenever that distribution comes from a tree like factorization. A Gaussian Chow & Liu tree is created in a way similar to the discrete variables case. Mutual information is also the maximum likelihood estimator, and whenever a multivariate normal distribution is factorized as the product of second order distributions the Gaussian Chow & Liu tree is the best approximation. For normal variables, mutual information is defined as:

$$MI(X,Y) = -\frac{1}{2}\log\left(1 - r_{x,y}^2\right). \tag{1}$$

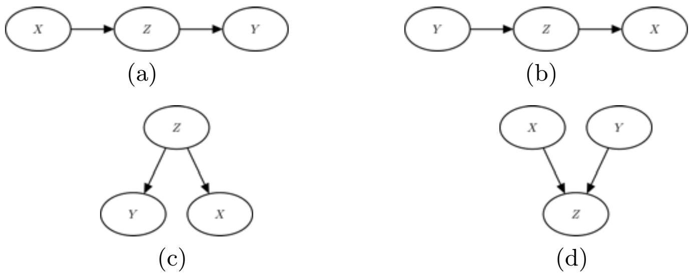
The term $r_{x,y}$ is the Pearson's correlation coefficient which for Gaussian variables is defined as:

$$r_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{2}$$

2. Edge orientation. The procedure to orient the edges of the tree is based on the orienting principle [15]: if in a triplet $X - Z - Y$ the variables X and Y are independent then Z is a head to head node with X and Y as parents, as follows: $X \rightarrow Z \leftarrow Y$. Similarly, if in a triplet $X \rightarrow Z - Y$ the variables X and Y are independent then Z is a head to head node with X and Y as parents:$X \rightarrow Z \leftarrow Y$; otherwise Z is the parent of Y: $X \rightarrow Z \rightarrow Y$.
In this paper we propose information theoretic measures such a conditional mutual information (CMI) and (marginal) mutual information (MI) to estimate the dependency between variables.
Proposed orientation based on information measures: for any triplet $X - Z - Y$, if $CMI(X,Y|Z) > MI(X,Y)$ then Z is a head to head node with X and Y as parents, as follows: $X \rightarrow Z \leftarrow Y$.
Proof. We shall prove that the proposed measure based on mutual information finds the correct orientation. That is, (in Figure 1 the four possible models made with three variables are shown), model M_4, head to head, is the correct one for $CMI(X,Y|Z) > MI(X,Y)$.
The quality of the causal models shown in the Figure 1 can be expressed by its log-likelihood. If the parents of any node X_i is the set of nodes $pa(X_i)$, the negative of the log-likelihood of a model M is [5]:

$$-\,ll(M) = \sum_{i=1}^{n} H(X_i|pa(X_i)) \tag{3}$$

where $H(X_i|pa(X_i))$ is the conditional entropy of X_i given its parents $pa(X_i)$. It is well known that the causal models M_1, M_2 and M_3 are equivalent,



(a) (b) (c) (d)

Fig. 1. The causal models that can be obtained with three variables X, Y y Z. (a) Model M_1. (b) Model M_2. (c) Model M_3. (d) Model M_4.

or indistinguishable in probability [15]. The negative log-likelihood are the Equations 4, 5 and 6, respectively.

$$\begin{aligned} -ll(M_1) &= H(X) + H(Z|X) + H(Y|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad -H(X,Y,Z) + H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{4}$$

$$\begin{aligned} -ll(M_2) &= H(Z) + H(X|Z) + H(Y|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad +H(X,Y,Z) - H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{5}$$

$$\begin{aligned} -ll(M_3) &= H(Y) + H(Z|Y) + H(X|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad -H(X,Y,Z) + H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{6}$$

For the head to head model (M_4), the negative of the log-likelihood is Equation 7.

$$\begin{aligned} -ll(M_4) &= H(X) + H(Y) + H(Z|X,Y) \\ &= H(X) + H(Y) + H(X,Y,Z) - H(X,Y) \\ &= H(X,Y,Z) + MI(X,Y) \end{aligned} \tag{7}$$

The best model is that one with the smallest negative log-likelihood or smallest summation of conditional entropy. When is the negative log-likelihood of Model M_4 smaller than the log-likelihood of model M_1 or M_2 or M_3 ?

$$H(X,Y,Z) + MI(X,Y) < H(X,Y,Z) + CMI(X,Y|Z) \tag{8}$$

The answer is in Equation 8. When the conditional mutual information $CMI(X,Y|Z)$ is larger than $MI(X,Y)$ the model M_4 has smaller negative log-likelihood value, therefore, $M4$ is the "best". □

In this work, the edge orientation principle runs on the depth first search algorithm [9]. The principle is applied to every pair of parent nodes in the

following way. Assume node A has nodes B, C, and D as candidate parents. There are 3 triplets to test: $B-A-C$, $B-A-D$ and $C-A-D$. As soon as a pair agrees with the proposed orientation principle, the edges are oriented as a head to head node. When the next triplet is tested but one of the edges is already directed the new test do not modify its direction.
The equation to compute the conditional mutual information of Gaussian variables is:

$$CMI(X,Y|Z) = \frac{1}{2}\log\left[\frac{\sigma_x^2\sigma_y^2\sigma_z^2\left(1-r_{xz}^2\right)\left(1-r_{yz}^2\right)}{|\Sigma_{xyz}|}\right] \tag{9}$$

3. Over-fitting control. The inequality
$MI(X,Y) < CMI(X,Y|Z)$ could be made true due to the small biases of the data and creating false positive parents. As a rule, the larger the allowed number of parents the larger the over-fitting. Multi parent nodes are great for polytrees but these nodes and their parents must be carefully chosen. A hypothesis test based on a non parametric bootstrap test over the data vectors X, Y and Z can be performed to solve the over-fitting problem. In this approach we used the statistic $\hat{\theta} = \overline{CMI(X^*,Y^*|Z^*)} - \overline{MI(X^*,Y^*)}$, the significance level 5%, null hypothesis $H_0 = \overline{CMI(X^*,Y^*|Z^*)} \leq \overline{MI(X^*,Y^*)}$ and alternative hypothesis $H_1 = \overline{CMI(X^*,Y^*|Z^*)} > \overline{MI(X^*,Y^*)}$. However this approach is computationally expensive. A better approach would be based on a threshold value but which value? Hence the question is: how many times the CMI must be larger than the MI as to represent true parents? Which is a good threshold value?. Empirically we solve this question by randomly creating a huge database of triplet-vectors X, Y and Z (from random gaussian distributions) that made true the inequality $MI(X,Y) < CMI(X,Y|Z)$. Within this large set there are two subsets: triplets that satisfy the null hypothesis and those that not. We found out that false parents are created in 95% of the cases when $\frac{CMI(X,Y|Z)}{MI(X,Y)} < 3$. Therefore the sought threshold value is 3. Thus a head to head node is created whenever $\frac{CMI(X,Y|Z)}{MI(X,Y)} \geq 3$.

4 Aspects of the Gaussian Polytree EDA

In the previous section we explained the algorithm to build a gaussian polytree. An Estimation Distribution Algorithm was created using our model. Two aspects of the Gaussian polytree EDA are important to mention.

1. Data simulation. The procedure to obtain a new population (or new samples) from a polytree follows the common strategy of sampling from conditional Gaussian variables. If variable X_i is conditioned on $Y = \{X_j, X_k, \ldots, X_z\}$, with $X_i \notin Y$, their conditional Gaussian distribution:

$$\mathcal{N}_{X_i|Y=\mathbf{y}}\left(\mu_{X_i|Y=\mathbf{y}}, \Sigma_{X_i|Y=\mathbf{y}}\right)$$

can be simulated using the conditional mean

$$\mu_{X_i|Y=\mathbf{y}} = \mu_{X_i} + \Sigma_{X_iY}\Sigma_{YY}^{-1}\left(\mathbf{y}-\mu_Y\right) \tag{10}$$

and the conditional covariance:

$$\Sigma_{X_i|Y=\mathbf{y}} = \Sigma_{X_iX_i} - \Sigma_{X_iY}\Sigma_{YY}^{-1}\Sigma_{YX_i} \tag{11}$$

The simulation of samples at time t follows the gaussian polytree structure. If X_i^t has no parents then $X_i^t \sim \mathcal{N}(\mu_{X_i^{t-1}}, \Sigma_{X_i^{t-1}})$; otherwise X_i^t follow the gaussian distribution conditioned to $Y = \mathbf{y}^{t-1}$. This method adds exploration to the gaussian polytree EDA. Notice it is different of common ancestral sampling.

2. Selection. In EDAs truncation selection is commonly used. Our approach differs. We select the K best individuals whose fitness is better than the average fitness of the entire population. By including all members of the population the average gets a poor value. Then the selection pressure is low and many different individuals (high diversity) are selected and used as information to create the next polytree.

5 Experiments

The Gaussian polytree EDA is tested in two sets of benchmark functions.

5.1 Experiment 1: Convex Functions

This set of 9 convex functions was solved using the IDEA algorithm adapted with mechanisms to avoid premature convergence and to improve the convergence speed [7],[2]. The functions are listed in Table 3. In [7] the mechanism increases or decreases the variance accordingly to the rate the fitness function improves. In [2] the mechanism computes the shift of the mean in the direction of the best individual in the population. These mechanism are necessary due to premature convergence of the IDEA algorithm. Notice that the Gaussian polytree EDA does not need any additional mechanism to converge to the optimum. 30 runs were made for each problem.

Initialization. Asymmetric initialization is used for all the variables: $X_i \in [-10, 5]$.
Population size. For a problem in l dimensions, the population is $2 \times (10(l^{0.7}) + 10)$ [2]
Stopping conditions. Maximum number of fitness function evaluations is reached: 1.5×10^5; or target error smaller than 1×10^{-10}; or no improving larger than 1×10^{-13} is detected after 30 generations and the mean of l standard deviations, one for each dimension, is less than 1×10^{-13}.

The Figure 2 shows the best number of evaluations needed to reach the target error for dimensions 2, 4, 8, 10, 20, 40, and 80. The success rate VS the problem dimensionality is listed in Table 1 and Table 2 details the number of evaluations found in our experiments.

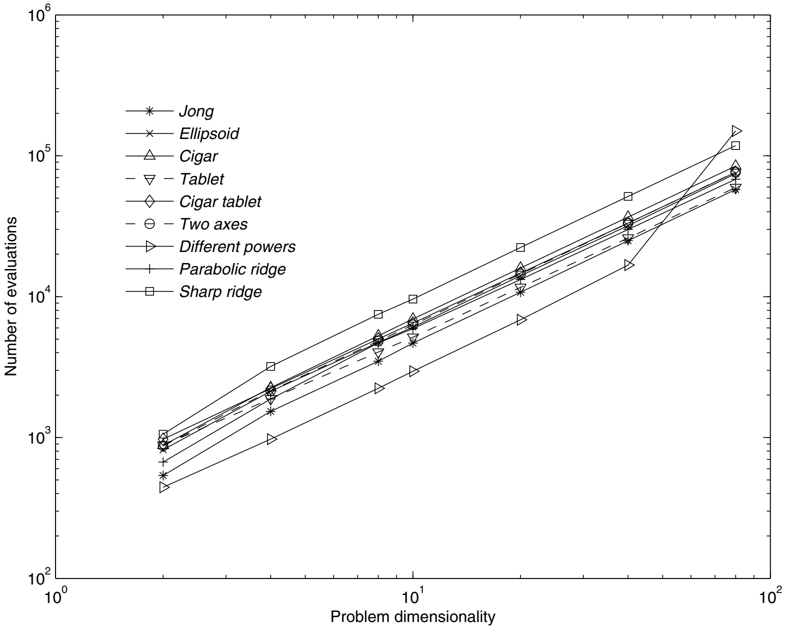


Fig. 2. Best number of evaluations VS problem dimensionality

Comments to Experiment 1. Note that the increment in the number of evaluations increases proportional to the increment in the dimensionality of the problem. The gaussian polytree EDA maintains a high success rate of global convergence, even in dimension 80. Out of these functions, just the different powers function (and slightly the two axes) were difficult to solve.

5.2 Experiment 2: Non-convex Functions

In this experiment we use four functions that Larrañaga and Lozano tested with different algorithms, including the estimation of Gaussian network algorithm

Table 1. Success rate of functions (%) VS problem dimensionality

Function	2-D	4-D	8-D	10-D	20-D	40-D	80-D
$\mathcal{F}_1$	100	100	100	100	100	100	100
$\mathcal{F}_2$	96.6	96.6	93.3	90.0	96.6	90.0	86.6
$\mathcal{F}_3$	96.6	93.3	86.6	86.6	93.3	96.6	93.3
$\mathcal{F}_4$	100	90.0	96.6	100	100	100	100
$\mathcal{F}_5$	90.0	93.3	93.3	100	96.6	100	100
$\mathcal{F}_6$	96.6	90.0	83.3	80.0	63.3	70.0	60.0
$\mathcal{F}_7$	100	100	96.6	93.3	73.3	26.6	0.0
$\mathcal{F}_8$	80.0	73.3	83.3	86.6	83.3	90.0	100
$\mathcal{F}_9$	73.3	83.3	96.6	100	100	100	100

Table 2. Number of evaluations performed by the Gaussian polytree EDA needed to reach the target error in 30 repetitions (see stopping conditions)

$\mathcal{F}_i$	Dim	Best	Worst	Mean	Median	SD
$\mathcal{F}_1$	2	5.3700 e2	8.2500 e2	7.3300 e2	7.5200 e2	6.2433 e1
	4	1.5340 e3	1.8090 e3	1.6739 e3	1.6770 e3	5.9753 e1
	8	3.4780 e3	3.9450 e3	3.7791 e3	3.7980 e3	9.5507 e1
	10	4.6760 e3	5.1220 e3	4.8663 e3	4.8690 e3	9.2939 e1
	20	1.0744 e4	1.1258 e4	1.1048 e4	1.1069 e4	1.3572 e2
	40	2.4931 e4	2.5633 e4	2.5339 e4	2.5308 e4	1.8670 e2
	80	5.7648 e4	5.8966 e4	5.8510 e4	5.8574 e4	3.1304 e2
$\mathcal{F}_2$	2	8.1800 e2	3.2950 e3	1.0650 e3	1.0115 e3	4.2690 e2
	4	2.1280 e3	5.8800 e3	2.3583 e3	2.2495 e3	6.6716 e2
	8	4.7180 e3	1.0001 e5	8.2475 e3	4.8910 e3	1.7363 e4
	10	6.0830 e3	2.0292 e4	7.2357 e3	6.3480 e3	2.9821 e3
	20	1.4060 e4	2.4260 e4	1.4686 e4	1.4303 e4	1.8168 e3
	40	3.1937 e4	5.1330 e4	3.4468 e4	3.2749 e4	5.4221 e3
	80	7.4495 e4	1.2342 e5	8.0549 e4	7.5737 e4	1.2893 e4
$\mathcal{F}_3$	2	8.8000 e2	3.5210 e3	1.0819 e3	1.0145 e3	4.6461 e2
	4	2.2600 e3	7.2280 e3	2.6692 e3	2.4375 e3	9.8107 e2
	8	5.2700 e3	1.5503 e4	6.3378 e3	5.5220 e3	2.3176 e3
	10	6.9430 e3	1.3732 e4	7.9081 e3	7.1060 e3	2.0858 e3
	20	1.5956 e4	2.6813 e4	1.6900 e4	1.6237 e4	2.6287 e3
	40	3.6713 e4	5.4062 e4	3.7592 e4	3.7017 e4	3.1153 e3
	80	8.4462 e4	1.1823 e5	8.7323 e4	8.5144 e4	8.2764 e3
$\mathcal{F}_4$	2	8.8300 e2	1.1120 e3	9.9520 e2	9.8900 e2	5.8534 e1
	4	1.8830 e3	5.8250 e3	2.3616 e3	1.9990 e3	1.1030 e3
	8	4.0430 e3	9.3870 e3	4.4143 e3	4.2545 e3	9.4333 e2
	10	5.1480 e3	5.6070 e3	5.4052 e3	5.4285 e3	1.1774 e2
	20	1.1633 e4	1.2127 e4	1.1863 e4	1.1861 e4	1.0308 e2
	40	2.6059 e4	2.6875 e4	2.6511 e4	2.6487 e4	2.2269 e2
	80	5.9547 e4	6.1064 e4	6.0308 e4	6.0302 e4	3.6957 e2
$\mathcal{F}_5$	2	9.7300 e2	3.6130 e3	1.3396 e3	1.1155 e3	7.4687 e2
	4	2.2230 e3	6.0680 e3	2.6141 e3	2.3760 e3	9.0729 e2
	8	5.0060 e3	1.0809 e4	5.5754 e3	5.2045 e3	1.4230 e3
	10	6.4820 e3	6.9730 e3	6.7031 e3	6.7075 e3	1.1929 e2
	20	1.4687 e4	2.7779 e4	1.5381 e4	1.4983 e4	2.3449 e3
	40	3.3287 e4	3.4203 e4	3.3852 e4	3.3865 e4	2.0564 e2
	80	7.6250 e4	7.8009 e4	7.7247 e4	7.7359 e4	3.8967 e2
$\mathcal{F}_6$	2	8.7100 e2	2.9510 e3	1.0655 e3	9.9550 e2	3.5942 e2
	4	2.1480 e3	5.5960 e3	2.5739 e3	2.2475 e3	1.0015 e3
	8	4.8380 e3	1.6298 e4	6.0937 e3	5.0160 e3	2.6565 e3
	10	6.3130 e3	2.3031 e4	8.1936 e3	6.5415 e3	3.8264 e3
	20	1.4455 e4	6.0814 e4	2.0558 e4	1.4919 e4	1.0252 e4
	40	3.3222 e4	6.2568 e4	3.9546 e4	3.3955 e4	9.2253 e3
	80	7.6668 e4	1.0019 e5	8.6593 e4	7.8060 e4	1.1221 e4
$\mathcal{F}_7$	2	4.4400 e2	6.2100 e2	5.2970 e2	5.3450 e2	5.1867 e1
	4	9.7500 e2	1.2580 e3	1.1103 e3	1.1100 e3	6.8305 e1
	8	2.2360 e3	7.3335 e4	4.7502 e3	2.4010 e3	1.2953 e4
	10	2.9530 e3	9.9095 e4	7.7189 e3	3.1475 e3	1.8871 e4
	20	6.8480 e3	1.0011 e5	3.1933 e4	7.2465 e3	4.1782 e4
	40	1.6741 e4	1.0017 e5	7.7923 e4	1.0003 e5	3.7343 e4
	80	1.5001 e5	1.5024 e5	1.5010 e5	1.5008 e5	7.1759 e1
$\mathcal{F}_8$	2	6.7000 e2	3.8730 e3	1.3424 e3	8.5950 e2	1.0699 e3
	4	1.8780 e3	8.8220 e3	3.2186 e3	2.2065 e3	1.8858 e3
	8	4.6880 e3	1.0773 e4	5.7467 e3	4.8275 e3	2.1246 e3
	10	5.9350 e3	1.2863 e4	7.0149 e3	6.1555 e3	2.2485 e3
	20	1.3228 e4	2.6804 e4	1.5504 e4	1.3667 e4	4.3446 e3
	40	2.9959 e4	8.3911 e4	3.3521 e4	3.0451 e4	1.0781 e4
	80	6.8077 e4	7.0542 e4	6.9092 e4	6.9069 e4	4.7975 e2
$\mathcal{F}_9$	2	1.0560 e3	4.2000 e3	2.0126 e3	1.3910 e3	1.1536 e3
	4	3.1980 e3	7.5810 e3	4.0188 e3	3.4055 e3	1.4445 e3
	8	7.4930 e3	1.4390 e4	7.9337 e3	7.7140 e3	1.2243 e3
	10	9.6110 e3	1.0325 e4	1.0013 e4	9.9930 e3	1.5436 e2
	20	2.2342 e4	2.3122 e4	2.2776 e4	2.2780 e4	1.9712 e2
	40	5.1413 e4	5.2488 e4	5.1852 e4	5.1827 e4	2.4254 e2
	80	1.1796 e5	1.2033 e5	1.1896 e5	1.1904 e5	5.3493 e2

Table 3. Set of convex functions of Experiment 1

Name	Alias	Definition
Sphere	$\mathcal{F}_1$	$\sum_{i=1}^{N} X_i^2$
Ellipsoid	$\mathcal{F}_2$	$\sum_{i=1}^{N} 10^{6\frac{i-1}{N-1}} X_i^2$
Cigar	$\mathcal{F}_3$	$X_1^2 + \sum_{i=2}^{N} 10^6 X_i^2$
Tablet	$\mathcal{F}_4$	$10^6 X_1^2 + \sum_{i=2}^{N} X_i^2$
Cigar Tablet	$\mathcal{F}_4$	$X_1^2 + \sum_{i=2}^{N-1} 10^4 X_i^2 + 10^8 X_N^2$
Two Axes	$\mathcal{F}_6$	$\sum_{i=1}^{[N/2]} 10^6 X_i^2 + \sum_{i=[N/2]}^{N} X_i^2$
Different Powers	$\mathcal{F}_7$	$\sum_{i=1}^{N} \lvert X_i \rvert^{2+10\frac{i-1}{N-i}}$
Parabolic Ridge	$\mathcal{F}_8$	$-X_1 + 100 \sum_{i=2}^{N} X_i^2$
Sharp Ridge	$\mathcal{F}_9$	$-X_1 + 100\sqrt{\sum_{i=2}^{N} X_i^2}$

($EGNA$). $EGNA$ is interesting for this comparison because it is a graph with continuous variables built with scoring metrics such as the Bayesian information criteria (BIC). The precision matrix is created from the graph structure which allows none or more parents to any node. Therefore, the Gaussian polytree and the $EGNA$ allow several parents.
The experimental settings are the following:

Population size. For a problem in l dimensions, the population is $2 \times (10(l^{0.7}) + 10)$ [2]
Stopping conditions. Maximum number of fitness function evaluations is: 3×10^5; or target error smaller than 1×10^{-6}, 30 repetitions. Also stop when no improving larger than 1×10^{-13} is detected after 30 generations and the mean of l standard deviations, one for each dimension, is less than 1×10^{-13}.

The set of test functions is shown in Table 4. Experiments were performed for dimensions 10 and 50. The comparison for the Sphere function is shown in Figure 5, for the Rosenbrock function in Table 6, for the Griewangk in Table 7, and for the Ackley function in Table 8.

Table 4. Set of test functions of Experiment 2

Name	Alias	Definition	Domain
Sphere	$\mathcal{F}_1$	$\sum_{i=1}^{N} X_i^2$	$-600 \leq X_i \leq 600$
Rosenbrock	$\mathcal{F}_2$	$\sum_{i=1}^{N-1} \left[(1 - X_i)^2 + 100 \left(X_{i+1} - X_i^2 \right)^2 \right]$	$-10 \leq X_i \leq 10$
Griewangk	$\mathcal{F}_4$	$\sum_{i=1}^{N} \frac{X_i^2}{4000} - \prod_{i=1}^{N} \cos\left(\frac{X_i}{\sqrt{i}} \right) + 1$	$-600 \leq X_i \leq 600$
Ackley	$\mathcal{F}_5$	$-20 \exp\left(-0.2\sqrt{\frac{1}{N} \sum_{i=1}^{N} X_i^2} \right)$	$-10 \leq X_i \leq 10$
		$- \exp\left(\frac{1}{N} \sum_{i=1}^{N} \cos(2\pi X_i) \right) + 20 + e$	

Table 5. Comparative for the Sphere function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	2.5913e-5 ± 3.71e-5	77162.4 ± 6335.4
	$EGNA_{BGe}$	7.1938e-6 ± 1.78e-6	74763.6 ± 1032.2
	$EGNA_{ee}$	7.3713e-6 ± 1.98e-6	73964 ± 1632.1
	PolyG	7.6198e-7 ± 1.75e-7	4723.9 ± 78.7
50	$EGNA_{BIC}$	1.2126e-3 ± 7.69e-4	263869 ± 29977.5
	$EGNA_{BGe}$	8.7097e-6 ± 1.30e-6	204298.8 ± 1264.2
	$EGNA_{ee}$	8.3450e-6 ± 1.04e-6	209496.2 ± 1576.8
	PolyG	8.9297e-7 ± 8.05e-8	32258.4 ± 274.1

Table 6. Comparative for the Rosenbrock function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	8.8217 ± 0.16	268066.9 ± 69557.3
	$EGNA_{BGe}$	8.6807 ± 5.87e-2	164518.7 ± 24374.5
	$EGNA_{ee}$	8.7366 ± 2.23e-2	301850 ± 0.0
	PolyG	7.9859 ± 2.48e-1	18931.8 ± 3047.6
50	$EGNA_{BIC}$	50.4995 ± 2.30	301850 ± 0.0
	$EGNA_{BGe}$	48.8234 ± 0.118	301850 ± 0.0
	$EGNA_{ee}$	48.8893 ± 1.11e-2	301850 ± 0.0
	PolyG	47.6 ± 1.52e-1	81692.2 ± 6704.7

Table 7. Comparative for the Griewangk function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	3.9271e-2 ± 2.43e-2	301850 ± 0.0
	$EGNA_{BGe}$	7.6389e-2 ± 2.93e-2	301850 ± 0.0
	$EGNA_{ee}$	5.6840e-2 ± 3.82e-2	301850 ± 0.0
	PolyG	3.6697e-3 ± 6.52e-3	60574.3 ± 75918.5
50	$EGNA_{BIC}$	1.7075e-4 ± 6.78e-5	250475 ± 18658.5
	$EGNA_{BGe}$	8.6503e-6 ± 7.71e-7	173514.2 ± 1264.3
	$EGNA_{ee}$	9.1834e-6 ± 5.91e-7	175313.3 ± 965.6
	PolyG	8.9551e-7 ± 6.24e-8	28249.8 ± 227.4

Comments to Experiment 2. The proposed Gaussian polytree EDA reaches better values than the $EGNA$ requiring lesser number of function evaluations in all function (except for the Rosenbrock were both show a similar performance).

Table 8. Comparative for the Ackley function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
	$EGNA_{BIC}$	5.2294 ± 4.49	229086.4 ± 81778.4
10	$EGNA_{BGe}$	7.9046e-6 ± 1.39e-6	113944 ± 1632.2
	$EGNA_{ee}$	74998e-6 ± 1.72e-6	118541.7 ± 2317.8
	PolyG	8.3643e-7 ± 1.24e-7	5551.5 ± 104.0
	$EGNA_{BIC}$	19702e-2 ± 7.50e-3	288256.8 ± 29209.4
50	$EGNA_{BGe}$	8.6503e-6 ± 3.79e-7	282059.9 ± 632.1
	$EGNA_{ee}$	6.8198 ± 0.27	301850 ± 0.0
	PolyG	9.4425e-7 ± 4.27e-8	36672.9 ± 241.0

6 Conclusions

In this paper we described a new EDA based on Gaussian polytrees. A polytree is a rich modeling structure that can be built with moderate computing costs. At the same time the Gaussian polytree is found to have a good performance on the tested functions. Other algorithms have shown convergence problems on convex functions and need special adaptations that the Gaussian polytree did not need. The new sampling method favors diversity of the population since it is based on the covariance matrix of the parent nodes and the children nodes. Also the proposed selection strategy applies low selection pressure to the individuals therefore improving diversity and delaying convergence.

References

1. Acid, S., de Campos, L.M.: Approximations of Causal Networks by Polytrees: An Empirical Study. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) IPMU 1994. LNCS, vol. 945, pp. 149–158. Springer, Heidelberg (1995)
2. Bosman, P.A.N., Grahl, J., Thierens, D.: Enhancing the performance of maximum-likelihood gaussian edas using anticipated mean shift. In: Proceedings of BNAIC 2008, the Twentieth Belgian-Dutch Artificial Intelligence Conference, pp. 285–286. BNVKI (2008)
3. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory IT-14(3), 462–467 (1968)
4. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
5. Dasgupta, S.: Learning polytrees. In: Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI 1999), pp. 134–141. Morgan Kaufmann, San Francisco (1999)
6. Edwards, D.: Introduction to Graphical Modelling. Springer, Berlin (1995)
7. Grahl, P.A.B.J., Rothlauf, F.: The correlation-triggered adaptive variance scaling idea. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 397–404. ACM (2006)
8. Lauritzen, S.L.: Graphical models. Clarendon Press (1996)

9. Ouerd, B.J.O.M., Matwin, S.: A formal approach to using data distributions for building causal polytree structures. Information Sciences, an International Journal 168, 111–132 (2004)
10. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall series in Artificial Intelligence (2004)
11. Ortiz, M.S.: Un estudio sobre los Algoritmos Evolutivos con Estimacion de Distribuciones basados en poliarboles y su costo de evaluacion. PhD thesis, Instituto de Cibernetica, Matematica y Fisica, La Habana, Cuba (2003)
12. Ouerd, M.: Learning in Belief Networks and its Application to Distributed Databases. PhD thesis, University of Ottawa, Ottawa, Ontario, Canada (2000)
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
14. de Campos, L.M., Moteos, J., Molina, R.: Using bayesian algorithms for learning causal networks in classification problems. In: Proceedings of the Fourth International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), pp. 395–398 (1993)
15. Rebane, G., Pearl, J.: The recovery of causal poly-trees from statistical data. In: Proceedings, 3rd Workshop on Uncertainty in AI, Seattle, WA, pp. 222–228 (1987)
16. Segovia-Dominguez Ignacio, H.-A.A., Enrique, V.-D.: The gaussian polytree eda for global optimization. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2011, pp. 69–70. ACM, New York (2011)