Real-Coded Estimation of Distribution Algorithm

Topon Kumar Paul∗ Hitoshi Iba∗

∗Graduate School of Frontier Sciences, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan
{topon,iba}@miv.t.u-tokyo.ac.jp

Abstract

In this paper, we show how Estimation of Distribution Algorithms (EDAs) can be applied to the optimization of multivariate functions in continuous domain. First, we describe the relationship between Gaussian network model and multivariate normal densities along with the methods used for the learning and simulation of Gaussian networks. Next, we propose our algorithm that uses only the means and covariance matrix of the variables, estimated from the selected promising individuals of a population, to generate offspring. Finally, we apply that algorithm to three bench-mark functions: Summation cancellation, Ackley and Sphere model, and provide the experimental results. The experimental results show that our proposed algorithm may produce results more accurately and more efficiently than some other EDAs.

1 Introduction

Recently, there have been proposed evolutionary algorithms based on probabilistic models, where recombination operators of building blocks of Genetic Algorithms (GAs) are replaced with generating new individuals by sampling the probability distribution, which is calculated from the selected promising individuals of a population. These Algorithms are called Estimation of Distribution Algorithms (EDAs) [7] or Probabilistic Model Building Genetic Algorithms (PMBGAs)[9]. EDAs explicitly take into account the problem specific correlations among the variables. Besides finding a solution of a problem, EDAs try to capture the structure of the variables of a problem. That is why, EDAs are thought to be more efficient than GAs for some problems. See [8] for a review of different EDAs in discrete domain.

Many EDAs such as Estimation of Gaussian Networks Algorithm (EGNA), Estimation of Multivariate Normal Algorithm (EMNA)[6] have been proposed for the multivariate function optimizations in continuous domain. EGNA optimizes functions based on the learning and simulation of Gaussian networks; EMNA does so by the estimation of a multivariate density function at each generation. It is not easy to learn Gaussian networks from data, probably an NP-hard problem, and the accuracy of the results found by those algorithms is not higher.

The purpose of this paper is to introduce a new Estimation of Distribution Algorithm that calculates the means and covariance matrix of variables from the selected individuals of a
population, and produces new offspring by sampling those means and covariances. We consider all variables as normally distributed, and decompose their covariance matrix using Cholesky decomposition. We then sample those variables in a tree like fashion. We perform experiments with three multivariate functions: the Summation cancellation, Ackley and Sphere model, and provide the experimental results. The experimental results show that our proposed algorithm is able to solve the tested problems more efficiently and more accurately.

2 Gaussian Networks for Multivariate Normal Distributions

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of $n$ continuous variables, and $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be the values of $X$. Assume that the Joint Probability Density Function (JPDF) of $X$ is a multivariate normal distribution $N(\mu, \Sigma)$, that is,

$$f(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right\} \quad (1)$$

where $\mu$ is the $n$-dimensional mean vector, $\Sigma$ is the $n \times n$ covariance matrix, $|\Sigma|$ is the determinant of $\Sigma$, and $\mu^T$ denotes the transpose of $\mu$. Sometimes, the inverse of the covariance matrix $(\Sigma^{-1})$ is referred as precision matrix, and denoted by $W$. This multivariate JPDF $f(\mathbf{x})$ can be decomposed as follows:

$$f(x_1, x_2, \ldots, x_n) = n \prod_{i=1}^{n} f_i(x_i|x_1, x_2, \ldots, x_{i-1}) \quad (2)$$

where

$$f_i(x_i|x_1, x_2, \ldots, x_{i-1}) \sim N(\mu_i + \sum_{j=1}^{i-1} \beta_{ij}(x_j - \mu_j), v_i) \quad (3)$$

where $\mu_i$ is the unconditional mean of $X_i$, $v_i$ is the conditional variance of $X_i$ given values of $X_1, X_2, \ldots, X_{i-1}$ and $\beta_{ij}$ is the linear regression coefficient of $X_j$ when $X_i$ is regressed on $x_1, x_2, \ldots, x_{i-1}$.

Shatcher and Kenley [11] introduced Gaussian Networks as special cases of multivariate normal distributions in which conditional probability distributions are defined in (3), JPDF is defined by (2), and $\beta_{ij} = 0$ indicates no link between $X_i$ and $X_j$. They described the general transformation from $v$ and $\{\beta_{ij}|j < i\}$ to the precision matrix $W$ of the normal distribution. The recursive formula for $W$ is

$$W(i + 1) = \begin{pmatrix} W(i) + \beta_{i+1}v_{i+1}^T & -\beta_{i+1}v_{i+1} \\ -\beta_{i+1}^Tv_{i+1} & \frac{1}{v_{i+1}} \end{pmatrix} \quad (4)$$

where $W(i)$ denotes the $i \times i$ upper left sub-matrix of $W$, $\beta_i$ denotes the column vector $\{\beta_{ij}|j < i\}$, and $W(1) = \frac{1}{v_1}$. The method of calculation of the conditional variances ($v$) is described in the next section.

The question is how to build the best fit Gaussian Networks from data. It is done by an intelligent search using score metrics through the space of different structures. Two score metrics: penalized maximum likelihood [6] and Bayesian Gaussian equivalence [5], are used.

Kyoto, Japan, August 25–28, 2003
for searching of the best fit Gaussian Network. Estimation of Gaussian Networks Algorithm (EGNA)\[6\] uses these two scores for optimization in continuous domain based on the learning and simulation of Gaussian networks.

3 Sampling from Multivariate Normal Distribution

Suppose \(X\) has a multivariate normal distribution \(N(\mu, \Sigma)\) with density function defined in (1). The general approach is to generate \(X \sim N(\mu, \Sigma)\) by generating \(X_1\), then \(X_2\) conditional on \(X_1\), and so on. Let \(X_i\) is conditionally dependent on \(Y = \{X_1, X_2, \ldots, X_{i-1}\}\), and \(\Sigma_{x,y}\) denote the covariance matrix of \((X_i, Y)\), and \(\sigma^2_{x_i}\) the unconditional variance of \(X_i\). Then, the conditional mean and variance of \(X_i\) can be calculated using the following two equations[2]:

\[
E(X_i|Y = y) = \mu_i + \Sigma_{x_i,y} \Sigma^{-1}_y (y - \mu_y)
\]

and

\[
Var(X_i|Y = y) = \sigma^2_{x_i} - \Sigma_{x_i,y} \Sigma^{-1}_y \Sigma_{x_i,y}^T.
\]

Here, \(\Sigma_{x_i,y} \Sigma^{-1}_y\) is the matrix of regression coefficients of \(X_i\) on \(Y\), and in Gaussian network it is denoted by \(\beta_i^T\). \(\Sigma^{-1}_y\) can be calculated efficiently using (4).

The other approach to generate the values from the multivariate normal distribution is based on the Cholesky decomposition. Since the covariance matrix \(\Sigma\) is symmetric and positive definite, it can be decomposed into the unique lower-triangular matrix \(L\) with \(LL^T = \Sigma\). Then if \(Z_1, Z_2, \ldots, Z_n\) are independent normal deviates, we can generate \(X \sim N(\mu, \Sigma)\) by using the following equation[10]:

\[
X = \mu + LZ.
\]

If \(\Sigma^{-1}\) is specified, and \(LL^T = \Sigma^{-1}\), we can form two equations: \(X = \mu + Y, L^T Y = Z\). This triangular system of equations can be easily solved, and thus the values of \(X\) can be generated.

4 The Proposed Algorithm

We call our proposed algorithm Real-Coded Estimation of Distribution Algorithm (RECEDA). We call so because in RECEDA, the individuals of a population are encoded as vectors of real numbers instead of binary encoding of real numbers, and it is an extension of the Estimation of Distribution Algorithm. In our algorithm, the means and covariance matrix are calculated from the selected individuals of a population. Since the covariance matrix is symmetric and positive definite, it can be decomposed into lower-triangular matrix using Cholesky decomposition. The new offspring are generated using (7). Here is the algorithm:

1. Generate initial population with uniform distribution of variables, and evaluate them.
2. Select \(N\) promising individuals from the population.
3. Calculate means(\(\mu\)) and covariance matrix(\(\Sigma\)) of the variables using the selected individuals.
4. Decompose the covariance matrix using Cholesky decomposition: $LL^T = \Sigma$.

5. Generate a vector of independent normal deviates ($Z$), and produce offspring using the equation: $X = \mu + LZ$.

6. Evaluate offspring, and replace old population with some offspring.

7. If termination-criteria is not met, go to step 2.

In this algorithm, the generated $X_i$ is a function of the mean calculated from the selected individuals of a population, independent normal variables, and the covariance matrix i.e.

$$X_i = \mu_i + \beta_{i1}Z_1 + \beta_{i2}Z_2 + \ldots + \beta_{i,i-1}Z_{i-1} + \beta_{ii}Z_i$$  \hspace{1cm} (8)

where $\beta_{ki}$ is the element at $i$th row and $k$th column in the decomposed matrix $L$. The initial population in the algorithm is generated with uniform distribution of variables, i.e. the value of a variable is chosen from the given range randomly. Therefore, if the range of a variable is larger, it needs more fitness evaluations and much time to converge to a solution.

5 Experiments

The experiments have been done for functions taken from [3] and [1]. The following subsections describe these functions, and present experimental results.

5.1 Test Functions

In this subsection we describe three test functions: Summation cancellation, Ackley and Sphere model functions. In all functions, $x = \{x_1, x_2, \ldots, x_n\}$.

Summation cancellation is a maximization problem defined as follows [3]:

$$f(x) = \frac{1}{10^{-5} + \sum_{j=1}^{n} |y_j|}$$  \hspace{1cm} (9)

where $-0.16 \leq x_i \leq 0.16$, $y_1 = x_1$, and $y_j = x_j + y_{j-1}$, $j = 2, \ldots, n$. It has the maximum value of 100000 when $x_i = 0$, $i = 1, \ldots, n$.

Ackley is a minimization problem proposed by Ackley [1], and has the following form:

$$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + \exp(1).$$  \hspace{1cm} (10)

The range of each $x_i$ is not defined in [1]. For our experiment, we set the range as $-32.768 \leq x_i \leq 32.768$. This function has the minimum value of 0 when $x_i = 0$, $i = 1, 2, \ldots, n$.

Another minimization problem is the Sphere model which is defined as

$$f(x) = \sum_{i=1}^{n} x_i^2$$  \hspace{1cm} (11)

where $-600 \leq x_i \leq 600$, $i = 1, 2, \ldots, n$. It has the optimum value of 0 when all $x_i$’s are 0.
5.2 Experimental Results

Here we present the experimental results. For all the problems, the experimental results of our algorithm of 10 independent runs are provided.

For all the problems, the algorithm runs until either the result obtained is closer than $10^{-6}$ from the optimum solution to be found or the maximum no. of generations has passed. In each generation we apply truncation selection: the best half of the population is selected for the calculation of the means and covariance matrix of the parameters. For each problem, we set population size $= 40 \times n$ where $n$ is the size of the problem, Elite $= 50\%$, and maximum no. of generations $= 300$. Our replacement strategy is elitism so that the best individual of a generation is not lost. The initial population is generated with uniform distribution. To make it clear that our algorithm is better than other EDAs, we provide experimental results of the above three problems of $EGNA_{BGe}$ and $EMNA_{global}$ taken from [4]. The results are shown in tables 1, 2 and 3. In the tables, each value of the form $x \pm y$ indicates the average value $x$ with the standard deviation $y$.

Table 1: The best fitness value and number of fitness evaluations required for the Summation cancellation function (optimum fitness value = 100000).

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Best fitness value</th>
<th>$EGNA_{BGe}$</th>
<th>$EMNA_{global}$</th>
<th>No. of fitness evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9.99999E+04 ± 0.0</td>
<td>9.99991E+04</td>
<td>9.99991E+04</td>
<td>4.42E+04 ± 286.7 ± 1836.9</td>
</tr>
<tr>
<td>50</td>
<td>9.9615E+04 ± 3.0E+00</td>
<td>9.17252E+03</td>
<td>8.61907E+04</td>
<td>3.01E+05 ± 17761.2 ± 1032.3</td>
</tr>
</tbody>
</table>

† The value 9.17252E+03 in [4] seems to be an editorial mistake.

Table 2: The best fitness value and number of fitness evaluations required for the Ackley function (optimum fitness value = 0).

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Best fitness value</th>
<th>$EGNA_{BGe}$</th>
<th>$EMNA_{global}$</th>
<th>No. of fitness evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>8.3763E-07 ± 1.29E-07</td>
<td>7.9046E-06</td>
<td>8.9265E-06 ± 6.89E-07</td>
<td>2.09E+04 ± 169.9 ± 1632.2</td>
</tr>
<tr>
<td>50</td>
<td>9.589E-07 ± 3.86E-08</td>
<td>8.6503E-06</td>
<td>9.5926E-06 ± 2.39E-07</td>
<td>2.50E+05 ± 1449.1 ± 632.1</td>
</tr>
</tbody>
</table>

Table 3: The best fitness value and number of fitness evaluations required for the Sphere model function (optimum fitness value = 0).

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Best fitness value</th>
<th>$EGNA_{BGe}$</th>
<th>$EMNA_{global}$</th>
<th>No. of fitness evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7.3969E-07 ± 2.02E-07</td>
<td>7.1938E-06</td>
<td>7.3350E-06 ± 2.24E-06</td>
<td>1.56E+04 ± 282.8 ± 1032.2</td>
</tr>
<tr>
<td>50</td>
<td>8.4637E-07 ± 6.29E-08</td>
<td>8.7097E-06</td>
<td>8.5225E-06 ± 1.35E-06</td>
<td>1.99E+05 ± 1013.8 ± 1264.2</td>
</tr>
</tbody>
</table>

Kyoto, Japan, August 25–28, 2003
6 Summary and Conclusion

In this paper, we have proposed an algorithm that is an extension of Estimation of Distribution Algorithm (EDA) in continuous domain. We calculate the means and covariance matrix of the variables in a problem from the selected individuals in a generation. During sampling of values for the variables, we assume that the joint probability distribution of the variables follows a multivariate normal distribution, and generate values according. We sample values based on the Cholesky decomposition of the covariance matrix. We then, apply our algorithm to three well known multivariate functions in continuous domain. Our proposed algorithm can find solutions of the stated problems more accurately in fewer no. of fitness evaluations than other EDAs.

During calculation of the covariance matrix, we have assumed the completely connected structure of the variables of a problem. The number of parameters that our algorithm estimates at each generation is greater than those done in the algorithms which optimize functions by learning and simulation of Gaussian networks. But, the mathematics needed to generate offspring by our method are simple.

References


Kyoto, Japan, August 25–28, 2003