Variable Screening for Reduced Dependency Modelling in Gaussian-based Continuous Estimation of Distribution Algorithms

Krishna Manjari Mishra, Marcus Gallagher
School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia, 4072
Email: krishna@itee.uq.edu.au, Email: marcusg@itee.uq.edu.au

Abstract

Estimation of Distribution Algorithms (EDAs) focus on explicitly modelling dependencies between solution variables. A Gaussian distribution over continuous variables is commonly used, with several different covariance matrix structures ranging from diagonal i.e. Univariate Marginal Distribution Algorithm ($\mathrm{UMDA_c}$*) to full i.e. Estimation of Multivariate Normal density Algorithm (*EMNA*). A diagonal covariance model is simple but is unable to directly represent covariances between problem variables. On the other hand, a full covariance model requires estimation of (more) parameters from the selected population. In practice, numerical issues can arise with this estimation problem. In addition, the performance of the model has been shown to be sometimes undesirable. In this paper, a modified Gaussian-based continuous EDA is proposed, called sEDA, that provides a mechanism to control the amount of covariance parameters estimated within the Gaussian model. To achieve this, a simple variable screening technique from experimental design is adapted and combined with an idea inspired by the Pareto-front in multi-objective optimization. Compared to* $\mathrm{EMNA_{global}}$*, the algorithm provides improved numerical stability and can use a smaller selected population. Experimental results are presented to evaluate and compare the performance of the algorithm to* $\mathrm{UMDA_c}$ *and* $\mathrm{EMNA_{global}}$.

Index Terms

Estimation of Distribution algorithms, Optimization problems, Screening Technique.

1. Introduction

EDAs are a recent development in the field of evolutionary algorithms [1], [2]. In EDAs a population is approximated with a probability distribution and new candidate solutions are obtained by sampling from this distribution. The aim is to avoid the use of arbitrary operators (such as mutation and crossover) in favor of explicitly modelling and exploiting the distribution of promising individuals. The model fitting task within each iteration of an EDA is typically carried out by a maximum likelihood estimation procedure. EDAs have been developed for both discrete and continuous problems. Reviews of EDAs can be found in [1], [3], [4]. This paper focuses on continuous EDAs. Most optimization algorithms make some explicit or implicit assumption about the nature of the objective function. In EDAs the interrelations are explicitly expressed through the joint probability distribution associated with the individuals selected at each iteration.

EDAs focus on explicitly modelling dependencies between solution variables. A Gaussian distribution over the variables is commonly used, with several different covariance matrix structures ranging from diagonal (UMDA_c) to full (EMNA). A diagonal covariance model is simple but is unable to directly represent covariances between problem variables. On the other hand, a full covariance model requires estimation of (more) parameters from the selected population.

There are specific examples where EDAs do not perform well [5] [6] and some numerical issues associated with their implementation can occur [7]. In this paper, a new continuous Gaussian-based EDA, called sEDA, is proposed that introduces a notion of variable importance by adapting a screening technique from experimental design. The algorithm also improves on numerical stability in EDAs by allowing the level of dependency modelling to be controlled.
The remainder of the paper is organized as follows. Section 2 provides an overview of existing continuous Gaussian-based EDAs and reviews issues around the covariance dependency modelling in these algorithms. In Section 3, a modification to the algorithm is proposed, the Screening EDA (sEDA). The implementation of the sEDA and parameter values is described in detail. In Section 4 experimental results are presented evaluating the sEDA and comparing it with $\mathrm{UMDA_c}$ and $\mathrm{EMNA_{global}}$. Section 5 provides a conclusion and summary of the paper.

U.S. Government work not protected by U.S. copyright
Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

2. Analyzing the existing continuous Gaussian based EDAs based on the interaction of variables.

The continuous (global) optimization problem is to find $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \epsilon S,$$

where $S \subseteq R^n$ is the set of feasible solutions, $f(\mathbf{x})$ is the fitness or objective function and $\mathbf{x} = (x_1, \ldots, x_n)$ is an individual or candidate solution vector.

Table 1. General EDA Algorithm.

```
BEGIN (set t = 0)
Generate initial population (uniformly in S)
REPEAT until stopping criterion is met
    Evaluate f(x^i) for each individual x^i in population
    Selection
    Build probabilistic model p_t(x) based on selected individuals
    Generate new population by sampling from p_t(x)
  t = t + 1
ENDREPEAT
END
```

EDAs are one class of metaheuristics that have been developed to solve optimization problems. The general pseudo-code framework for an EDA is shown in Table 1. In EDAs, the initial population is generated randomly in the search space. Selection is typically done via the truncation method. A probability density estimator is then used to model the promising regions of the search space through maximum likelihood estimates which is subsequently used for generating the candidate solutions for the next generation. The family and structure of the model used in an EDA is typically fixed (e.g. a set of Bernoulli distributions to generate the bit strings in the PBIL algorithm [8], or a factorized Gaussian distribution over a continuous search space in the $\mathrm{UMDA_c}$ algorithm [1]).

A number of different types of density estimation models have been used in EDAs for both discrete and continuous search spaces. In the continuous domain, the Gaussian distribution is the most widely used model [1], [9], [10]. Other models considered include semi-parametric, e.g. mixture models [11] and non-parametric, e.g. histograms and kernel density estimators [12]–[15]. In this paper, we focus on Gaussian EDA models although the contributions are likely to be extendable to other models. A Gaussian distribution over the variables is commonly used, with several different covariance matrix structures ranging from diagonal (e.g. in the $\mathrm{UMDA_c}$) to full (e.g. EMNA).

The continuous Univariate Marginal Distribution Algorithm ($\mathrm{UMDA_c}$) was introduced by Larrañaga et al. [16]. A factorized Gaussian model (i.e. a diagonal covariance matrix) is used which assumes all the variables are independent to each other. The model parameters are estimated by using maximum likelihood. This algorithm is easy to apply and computationally efficient. But the model may have difficulty in solving problems where the variables have strong dependencies.

To address this limitation, multivariate Gaussian models have been proposed. Estimation of Multivariate Normal Algorithm ($\mathrm{EMNA_{global}}$) is an example, where dependencies between all pairs of variables are modelled. In $\mathrm{EMNA_{global}}$, the mean and covariance matrix are also computed via maximum likelihood [1]. The computational cost is higher than $\mathrm{UMDA_c}$.

In practice, numerical issues can arise with the EDA model estimation problem [7]. The covariance matrix, which is estimated by using maximum likelihood in $\mathrm{EMNA_{global}}$ is positive semi-definite by its definition. But in practice this is not guaranteed because of finite precision computation. There will be computation error or numerical issues arising when the sample used to estimate the model does not adequately span all dimensions of the search space, which is especially likely when the sample size is relatively small compared to the problem dimensionality. Various methods and techniques (e.g. eigen-value tuning strategies) have been proposed in [7], [9] in order to avoid the ill-posed condition of the covariance matrix. Eigen-decomposition EDA (ED-EDA) described in [9] is one of the techniques to repair the ill-posed covariance matrix. The experimental results in [7], [9] show that the values from different covariance repairing methods avoid the numerical difficulties and give good results with respect to best solution found as well as using less number of sample size as compared to classical EDAs.

The performance of EDA model has also been shown to be sometimes undesirable [5], [6]. In some landscapes, EDAs do not perform very well due to the premature shrinking of the variance at an exponential rate. Eg., in slope like regions of the search space, described in [6] and in elliptical surface which is described in [5], premature convergence is likely to occur. In these cases, the direction of descent is much more important than selected solutions. Different techniques have been introduced in [5], [6] e.g. Adaptive Variance Scaling (AVS), Anticipated Mean Shift (AMS), Covariance Matrix Adaptation(CMA) etc. to overcome this issue.

In this paper, a modified Gaussian-based continuous EDA is proposed, called sEDA, that provides a mechanism to control the amount of covariance parameters estimated within the Gaussian model. To achieve this, a simple variable screening technique from experimental design is adapted and combined with an idea inspired by the Pareto-front in multi-objective optimization. Compared to $\mathrm{EMNA_{global}}$, the algorithm provides improved numerical stability and can use a smaller selected population. Experimental results are presented to evaluate and compare the performance of the algorithm to $\mathrm{UMDA_c}$ and $\mathrm{EMNA_{global}}$.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

3. Variable Screening in an EDA

In this Section we introduce a Screening EDA (sEDA). The main idea of the algorithm is to allow a degree of dependency modelling as a trade-off between the $\mathrm{UMDA_c}$ model (no covariance) and the $\mathrm{EMNA_{global}}$ model (full covariance matrix), to try and capture the advantages and limit the potential problems of both approaches. Hence, the sEDA uses a multivariate Gaussian model where the covariance matrix contains some degree of "sparseness". On each generation, some variables are selected to be modelled using covariance terms between them while others are fixed at $\sigma_{ij} = 0$. The immediate question is to determine a way of selecting which variables to model covariance between.

3.1. Measuring the Strength or Importance of Dependencies Between the Variables Using Elementary Effects

A number of techniques for identifying variable interactions and importance have been developed and applied in the field of experimental design and particularly in the model-based engineering design. A simple technique proposed by Morris in 1991 [17] is based on measuring the mean and standard deviation of perturbations of individual variables for a given problem, calculated via so-called elementary effects terms.

The elementary effect for the ith variable, $E_i(\mathbf{x})$, is defined as follows. Let Δ be a pre-determined amount to perturb each variable. For a given $\mathbf{x}$

$$E_i(\mathbf{x}) = \frac{f(x_1, x_2, ..., x_{i-1}, x_i + \Delta, x_{i+1}, ..., x_n) - f(\mathbf{x})}{\Delta}$$

where $\mathbf{x}$=$(x_1, x_2, ..., x_n)$ is a given starting or "baseline" vector in the solution space. The perturbations, Δ are by default determined according to a full factorial sampling grid of some fixed resolution or increment size. In other words, for each variable x_i, over some fixed range and increment size, the value of x_i is changed and f is recalculated, producing a sample or set of values of $E_i(\mathbf{x})$.

Given a set of elementary effect values, the mean, $\overline{E}_i(\mathbf{x})$, and standard deviation, $std(E_i(\mathbf{x}))$ over the sample can be calculated. A high value of $\overline{E}_i(\mathbf{x})$ then indicates a variable that has a large average influence on the value of f. A high value of $std(E_i(\mathbf{x}))$ indicates that variable x_i has a fluctuating influence on the value of f, which may indicate that it is involved in interactions with other variables [18]. Either or both may be important. In engineering design, these values are typically examined manually to inform decisions, e.g. about which variables to include in a model [19].

3.2. Incorporating Variable Screening in an EDA

To incorporate elementary effect values into an EDA, two main decisions must be made. The first is how to calculate the $E_i(\mathbf{x})$ values themselves, which will happen on each generation of the sEDA. Using a sampling plan such as a full factorial design requires the specification of a grid of some predetermined resolution, perturbing each $\mathbf{x}^i$ along vertices in the grid and evaluating f at these points. We propose a simpler alternative here using only the selected population. Specifically, the mean of the selected population is calculated for each dimension x_i. The population, P_{tot} is then formed by creating new solution vectors where the mean value is substituted in turn for each problem variable (e.g. $\mathbf{x^i} = x_i, \ldots, x_{i-1}, m_i, x_{i+1}, \ldots, x_n$). This produces nM_{sel} new solutions, which are evaluated using f.

Given the elementary effect values and their sample mean and standard deviation, the second decision to be made is how to use them to determine covariance matrix structure to be used in the sEDA model. To do this, we utilize the concepts of dominance and Pareto optimality from multi-objective optimization (see, e.g. chapter 9 of [20]). We consider the mean and standard deviation of elementary effects as two different (aka decision-making) criteria. One solution is said to dominate the other if its score is at least as high for all objectives, and is strictly better for at least one. Let $\mathbf{x}^a$ and $\mathbf{x}^b$ have n objectives as a n-dimensional vector $\mathbf{a}$. Using the $\succeq$ symbol to indicate domination, $A \succeq B$ is defined as:

$$\mathbf{x}^a \succeq \mathbf{x}^b \Leftrightarrow \forall i \ \epsilon \{1, ..., n\} \, a_i \geq b_i,$$
$$and \ \exists i \ \epsilon \{1, ..., n\}, a_i > b_i.$$

All non-dominated solutions possess the attribute that their quality cannot be increased with respect to any of the objective functions without detrimentally affecting one of the others. In the presence of constraints, such solutions usually lie on the edge of the feasible regions of the search space. The set of all non-dominated solutions is called the Pareto set or the Pareto front.

In the sEDA, a fixed fraction η of the variables need to be selected for covariance modelling. Variables that belong to the Pareto set are selected first. If more variables are required, then those which have the minimum (Euclidean) distance to the Pareto front are selected. On the other hand, if the number of variables on the Pareto front is greater than required, then a random subset of these variables is selected.

The complete sEDA algorithm is summarized in Table 2. The critical steps of the algorithm are as described above. Three algorithm parameters must be specified for implementation: the population size M, the selection parameter τ and the variable screening/selection parameter η. sEDA uses truncation selection: a fraction τ of the population with the best objective function values are retained for building/adapting the search model.[1]. The mean m of the selected population is then calculated for

1. Rounding if $M \cdot \tau$ is not an integer.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

expanding the population. This expanded population is then used to calculate elementary effects values and their mean ($\overline{E}(\mathbf{x})$) and standard deviation $std(E(\mathbf{x}))$ values. After selecting variables with respect to the Pareto set of the mean and standard deviation of E, the covariance matrix for the EDA model is formed as a sparse matrix, with non-zero covariance terms for selected variables. This is used in combination with the EDA mean vector (estimated from the selected expanded population) and the model is then used to generate the new population as in a standard EDA. The process is repeated until some stopping criterion is met.

Table 2. sEDA Algorithm.

Given: Population size M, dimension size n, selection parameter τ, model selection parameter η, and $B = \mathrm{Round}(n \cdot \eta)$, where B is the number of best variables.
BEGIN (set $t = 0$)
Initialize population P by generating M individuals uniformly in S
REPEAT FOR $t = 1, 2, \ldots$ UNTIL stopping criterion is met
  Evaluate f for population P.
  Selection: $M_{sel} = \mathrm{Round}(M \cdot \tau)$ individuals from P.
  Let selected population = P_{sel}.
  Calculate sample mean $\tilde{m}$ of P_{sel}.
  Calculate P_{tot} by expanding P_{sel}, successively replacing variable $1, \ldots, n$ with m_i
  Evaluate f for new individuals in P_{tot} population.
  Selection: $M_{tot}^{sel} = \mathrm{Round}(n \cdot M_{sel} \cdot \tau)$ individuals from P_{tot} .
  Let new selected population be P_{tot}^{sel}.
  Calculate Elementary Effect E of the fitness function of P_{tot}^{sel}.
  Calculate the mean μ of the P_{tot}^{sel}.
  Calculate standard deviation $std(E)$ and mean $\overline{E}$
  Determine the Pareto optimal solutions p_o using $std(E)$ and $\overline{E}$ as two objective function.
  If $p_o > B$, randomly choose B solutions from p_o.
  If $p_o < B$, select/add $B - p_o$ solutions nearest to the Pareto front
  When $p_o == B$, then
  Build Σ_t using covariance terms for the B selected variables
  Build Σ_t using variance terms for the $n - B$ variables
  $\theta = cov(\eta) + var(100 - \eta)$.
  $p(\mathbf{x}) \leftarrow (\theta, \mu)$
  Generate P new population by sampling from $p(\mathbf{x})$.
ENDREPEAT
END

4. Experiments

4.1. Selection Parameter Settings for sEDA

sEDA contains two selection parameters, τ and η, which is described above. Setting these values is important for experimental results. In this set of experiment, different combinations tried for τ and η are ((0.3,0.3), (0.3,0.5), (0.5,0.3), (0.5,0.5)). With these values, sEDA is tested using 6 different 10-D benchmarking functions, which are the first 6 functions from the Real-Parameter Black-Box Optimization Benchmarking (BBOB) experiment set [21]. sEDA is implemented with population size of 500 and 300000 function evaluations. The algorithm is terminated when difference between the best fitness value found and the global optimum is less than or equal to 1e-08 or the algorithm attains the maximum allowed number of function evaluations.

Table 3 lists mean and standard deviation of the best fitness values found over 30 runs for different combination of τ and η. Best results with minimal mean value are represented in bold font. It is discovered from Table 3 that, out of 6 functions, 5 functions are better when τ=0.3 and η=0.3.

In order to check the behavior between different combination, a two-tailed t test of the null hypothesis has been used (assumed unequal variances). Test has been done between the best fitness values found by (0.3,0.3) combination with 3 other different combination for each function.

From the set of experiments on different combinations and doing t test, we come to the conclusion in the following way. The best result obtained is not statistically significant for (0.3,0.3) and (0.3,0.5) combination, where as in most of the cases, the result of (0.3,0.3) combination has statistically significant difference for (0.5,0.5) combination. Except function F6, values of other functions show no statistical difference for (0.3,0.3) and (0.5,0.3) combination. Although, the t test does not clearly give any idea about the better combination, given the results presented, based on best results with minimum mean value (even the results have very little difference), we set the value of τ and η as 0.3 and 0.3 respectively for sEDA in the remainder of the paper.

4.2. Evaluation of sEDA and Comparisons

In the following experiments, we are considering 2 sets of problems. One is a set of commonly used artificial test functions (Sphere, Griewangk, Ackleys, Rosenbrock, Schwefel) which is described in Table 4. Here all are minimization problems. The second set of problem is Circle in a square (CiaS) packing problems, which is an example of real world problem.

The objective of the CiaS problem is to maximize the common radius of n_c non-overlapping circles contained in the unit square. Let $C(z^i, r)$ be the circle with center z^i and radius r, where $z^i = (x_i, y_i)\ \epsilon\ I\!R^2$, then the problem can be formalized as follows:

$$r_n = max \quad r$$
$$C(z^i, r) \subseteq D \qquad i = 1, ..., n_c$$
$$C^{int}(z^i, r) \cap C^{int}(z^j, r) = \Phi; \forall i \neq j$$

where C^{int} denotes the interior of a circle [22] [23].

Experiments are conducted on 10-D and 50-D versions of the artificial test functions. For the (CiaS) problems

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

Table 3. Solution quality comparison having different values of τ and η with 6 different function from BBOB. Bold font represents the best result.

Function	(τ, η)	Best fitness values
F1	**(0.5,0.3)**	**3.07E-09±1.8E-09(-)**
	(0.5,0.5)	1.21E-02±6.6E-02(+)
	(0.3,0.5)	4.43E-09±2.2E-09(-)
	(0.3,0.3)	4.35E-09±2.6E-09
F2	(0.5,0.3)	0.10±0.2(-)
	(0.5,0.5)	1.52±4.8(-)
	(0.3,0.5)	0.16±0.5(-)
	(0.3,0.3)	**0.09±0.4**
F3	(0.5,0.3)	2.14±1.1(-)
	(0.5,0.5)	2.48±1.7(-)
	(0.3,0.5)	2.15±1.3(-)
	(0.3,0.3)	**1.92±1.3**
F4	(0.5,0.3)	10.83±2.1(-)
	(0.5,0.5)	11.33±2.2(+)
	(0.3,0.5)	9.51±2.1(-)
	(0.3,0.3)	**9.65±2.1**
F5	**(0.5,0.3)**	**-1.00E-08±5.0E-024**
	(0.5,0.5)	0.02±0.1(-)
	(0.3,0.5)	**-1.00E-08±5.0E-02**(n)
	(0.3,0.3)	**-1.00E-08±5.0E-02**(n)
F6	(0.5,0.3)	16.70±8.9(+)
	(0.5,0.5)	17.08±7.5(+)
	(0.3,0.5)	10.24±6.8(-)
	(0.3,0.3)	**9.66±6.5**

+ sign, the value of t test (2 tailed)<0.05, indicates statistically significant difference when compared with results of (0.3,0.3).
- sign, the value of t test (2 tailed)>0.05, indicates no statistically significant difference when compared with results of (0.3,0.3).
(n), when the results are same.

Table 4. Test functions used in the experiments

Name	Function
Sphere	$f_{Sph}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ $-5.12 \leq x_i \leq 5.12, f_{Sph}(\mathbf{x}^*) = 0$
Griewangk	$f_{Gri}(\mathbf{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $-600 \leq x_i \leq 600, f_{Gri}(\mathbf{x}^*) = 0$
Ackleys	$f_{Ack}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{30}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$ $-15 \leq x_i \leq 30, f_{Ack}(\mathbf{x}^*) = 0$
Rosenbrock	$f_{Ros}(\mathbf{x}) = \sum_{i=1}^{n-1}(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$ $-2 \leq x_i \leq 2, f_{Ros}(\mathbf{x}^*) = 0$
Sumcan	$f_{sum}(\mathbf{x}) = 1/\left(10^{-5} + \sum_{i=1}^{n} \lvert y_i \rvert\right)$ $-0.16 \leq x_i \leq 0.16, f_{sum}(\mathbf{x}^*) = 100000$
Rastrigin	$f_{Ras}(\mathbf{x}) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$ $-5 \leq x_i \leq 5, f_{Ras}(\mathbf{x}^*) = 0$
Schwefel	$f_{sch}(\mathbf{x}) = \sum_{i=1}^{k} \lvert x_i \rvert + \prod_{i=1}^{k} \lvert x_i \rvert$ $-10 \leq x_i \leq 10, f_{sch}(\mathbf{x}^*) = 0$
$f9$	$f_8(\mathbf{x}) = -\sum_{i=1}^{k} x_i sin\left(\sqrt{\lvert x_i \rvert}\right)$ $-500 \leq x_i \leq 500, f_8(\mathbf{x}^*) = -418.9829k$

experiments are conducted on the number of circles, which is ranging from 2,. . . ,20.

The initial population for all the algorithms are generated using same random generation procedure based on a uniform distribution. For 10-D and 50-D functions, the parameters for UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$ are taken from chapter 8 of [1] i.e, the selection parameter is set to 0.5 and population size is 2000. For 10-D and 50-D functions, 500 and 1000 population are used respectively by sEDA. Comparing 10-D functions, we set 301850 function evaluations. But when it comes to 50-D functions, 10 times more function evaluations are used. Algorithms are terminated when the difference between the absolute value of the best fitness found and the known global optimum is less than or equal to 1e-08 or it exceeds the given number of function evaluations. For each single test, the result is averaged over 100 independent runs.

For circles in a square problems the results for UMDA_c are taken from [22]. For $\mathrm{EMNA}_{\mathrm{global}}$, the population size=1000 and the selection parameter as 0.5. are taken. For sEDA, the value of τ and η are 0.3 and 0.3 respectively, with the population size equals to 50 times the dimension of the problem, where dimension is defined as 2*number of circles in a problem. The number of function evaluations considered is 2000000. Algorithms are terminated when the difference between the absolute value of best fitness value and the known global optimum is less than equal to 1e-06 or it exceeds the given number of function evaluation. For each test, the result is averaged over 30 independent runs.

The number of function evaluations in each generation are different for different algorithms. In UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$ the number of function evaluations at each generation is equals to the total number of population considered, where as in sEDA the number of function evaluations in each generation is a function of population size, selection parameter and dimension of the problem. The number of function evaluations at each generation for sEDA is $M + (M * \tau * n)$, where M is the population size, n is the dimensionality of the problem and τ is the standard truncation ratio.

4.3. Results

Table 5 and Table 6 summarizes the results on 10-D and 50-D functions using UMDA_c, $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA algorithms. Best fitness value found and the number of evaluations required to reach the final solution is recorded for each of the experiments. The best result with the minimum mean value with respect to fitness as well as number of function evaluations is highlighted in bold font. A t-test has been done between best values found by sEDA and 2 other EDAs, (with null hypothesis being that each set of experimental results is drawn from distributions with equal mean (and assumed unequal variances)).

4.3.1. Discussion for 10-D and 50-D functions. Tables 5 and 6 show the comparison of the 3 algorithms using 6 different functions.

$Sphere\ function$: In Sphere function, all the variables are independent to each other and equally important to the problem. Although it always facilitates univariate model based EDAs in solving the problems but Tables 5 and 6 show that, the performance of sEDA and UMDA_c is similar in both 10-D and 50-D dimension. But the

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

performance of $\mathrm{EMNA}_{\mathrm{global}}$ is significantly degraded. For 10-D Sphere, sEDA requires less number of function evaluations where as requirement of function evaluations are more when dimension increases as compared to 2 other EDAs.

$Griewangk\ function$: This is a multimodal and non-separable function. For 10-D version, sEDA is significantly better than UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$ in terms of best fitness values found as well as number of function evaluations. For 50-D, although the performance of sEDA is better, in terms of minimum mean value of the best fitness but found no significant difference as compared to 2 other algorithms. sEDA requires significantly more number of function evaluations than other 2 EDAs. If sEDA capture the right percent of variables while modelling, then it may outperform other two algorithms significantly in high dimension also.

$Ackely\ function$: Ackely has several local minima, hence can be difficult to optimize. The performance of sEDA compared to UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$ is better in terms of minimum average best fitness values but shows no statistically significant difference for both 10-D and 50-D version of the function. But we can conclude that sEDA is better in 10-D because it requires less number of function evaluations as compared to other 2 algorithms. For 50-D Ackely, sEDA requires significantly more number of function evaluations.

$Rosenbrock\ function$: Rosenbrock function, is a unimodal but very hard problem. Although sEDA does not attain global optimum, but it outperforms UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$ with significantly better solutions. These 3 algorithms use same number of function evaluations to achieve best fitness values.

$Rastrigin\ function$: Rastrigin function which is a multimodal and separable problem shows that the performance of sEDA is degraded significantly as compared to UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$. For 10-D version of the function, $\mathrm{EMNA}_{\mathrm{global}}$ performs better (since adequate population is supplied), while for 50-D version UMDA_c performs better (due to separable nature of the problem), but in both the cases sEDA doesn't show its performance very efficiently.

$Schwefel\ function$: For 10-D and 50-D Schwefel function, the performance of $\mathrm{EMNA}_{\mathrm{global}}$ surpasses sEDA and UMDA_c. We can increase the performance of sEDA by modifying the value of η, since we know clearly from the definition of Schwefel function described in Table 4, that all the variables have interaction with each other. Further discussion is in sub-section 4.4.

4.3.2. Results for Circle in a Square (CiaS) Problems. [22] discussed about the performance of UMDA_c and Nelder-Mead simplex algorithm in CiaS problems. Here we compare UMDA_c, $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA with number of circles ranging from 2 ... 20, over a fixed number of function evaluations.

Figure 1 shows the performance of UMDA_c, $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA on the CiaS problems. The x-axis denotes the problem size (n_c) while the y-axis is a performance ratio given by $d_n/f(x_n)$, where d_n is the known global optimum and $f(x_n)$ is the solution found by the algorithm. Figure 1 shows that the performance of UMDA_c is worst than $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA. But when sEDA is compared with $\mathrm{EMNA}_{\mathrm{global}}$, we found out that up to n_c = 4, the performance is some how

Table 5. Solution quality comparison for 10-D problem. Bold font represents the best result.

Function	Alg.	Best Fitness	No. of Evaluations
Sphere	U	**6.91E-09±1.8E-09(-)**	1.5E+05±1.4E+03(+)
	E	7.67E-09±1.6E-09(+)	1.4E+05±1.8E+03(+)
	s	6.98E-09±1.9E-09	**5.5E+04±9.7E+02**
Griewangk	U	7.78E-09±1.6E-09(+)	1.4E+05±1.7E+03(+)
	E	7.32E-09±1.5E-09(+)	1.4E+05±1.5E+03(+)
	s	**6.82E-09±1.8E-09**	**6.0E+04±3.2E+03**
Ackely	U	8.50E-09±1.1E-09(-)	2.0E+05±1.4E+03(+)
	E	8.40E-09±1.1E-09(-)	2.1E+05±1.4E+03(+)
	s	**8.27E-09±1.2E-09**	**7.8E+04±9.2E+02**
Rosenbrock	U	8.21E+00±2.3E-02(+)	3.0E+05±0.0E+00(n)
	E	7.80E+00±1.5E-01(+)	3.0E+05±0.0E+00(n)
	s	**7.51E+00±2.6E-01**	3.0E+05±0.0E+00
Rastrigin	U	7.32E-09±1.8E-09(+)	2.2E+05±5.4E+03(+)
	E	**6.91E-09±1.8E-09(+)**	2.2E+05±4.6E+03(+)
	s	1.99E-02±1.4E-01	**7.8E+04±3.2E+03**
Schwefel	U	3.72E+00±3.5E+00(+)	3.0E+05±0.0E+00(n)
	E	**7.51E-09±2.1E-09(+)**	**1.3E+04±1.5E+03(+)**
	s	6.87E-01±1.0E+00	3.0E+05±0.0E+00

U stands for UMDA_c, E stands for $\mathrm{EMNA}_{\mathrm{global}}$, s stands for sEDA.
+ sign, the value of t test (2 tailed)<0.05, indicates statistically significant difference when compared with results of sEDA.
- sign, the value of t test (2 tailed)>0.05, indicates no statistically significant difference when compared with results of sEDA.
(n), when the results are same.

Table 6. Solution quality comparison for 50-D problem. Bold font represents the best result.

Function	Alg.	Best Fitness	No. of Evaluations
Sphere	U	8.88E-09±8.5E-10(-)	3.9E+05±1.7E+03(+)
	E	2.65E-08±1.8E-07(+)	**3.9E+05±2.6E+05(+)**
	s	**8.81E-09±9.1E-10**	1.3E+06±7.8E+03
Griewangk	U	8.87E-09±8.7E-10(-)	**3.5E+05±1.6E+03(+)**
	E	1.42E-08±3.6E-08(-)	4.6E+05±5.9E+05(+)
	s	**8.58E-09±7.2E+10**	1.1E+06±6.9E+03
Ackely	U	4.38E-08±1.8E-07(-)	7.2E+05±7.3E+05(+)
	E	9.41E-09±4.4E-10(-)	**5.3E+05±1.8E+03(+)**
	s	**9.35E-09±4.5E+10**	1.7E+06±8.2E+03
Rosenbrock	U	5.25E+01±1.8E+00(+)	3.0E+06±0.0E+00(n)
	E	4.78E+01±2.8E-02(+)	3.0E+06±0.0E+00(n)
	s	**4.73E+01±3.6E-01**	3.0E+06±0.0E+00
Rastrigin	U	**8.84E-09±8.1E-10(+)**	**5.7E+05±9.2E+03(+)**
	E	7.05E+00±5.0E+00(+)	3.0E+06±0.0E+00(n)
	s	3.18E-01±5.0E-01	3.0E+06±0.0E+00
Schwefel	U	7.22E+02±2.2E+01(+)	3.0E+06±0.0E+00(n)
	E	**9.52E-09±6.6E-09(+)**	**3.5E+05±2.6E+05(+)**
	s	2.09E+02±7.5E+01	3.0E+06±0.0E+00

U stands for UMDA_c, E stands for $\mathrm{EMNA}_{\mathrm{global}}$, s stands for sEDA.
+ sign, the value of t test (2 tailed)<0.05, indicates statistically significant difference when compared with results of sEDA.
- sign, the value of t test (2 tailed)>0.05, indicates no statistically significant difference when compared with results of sEDA.
(n), when the results are same.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

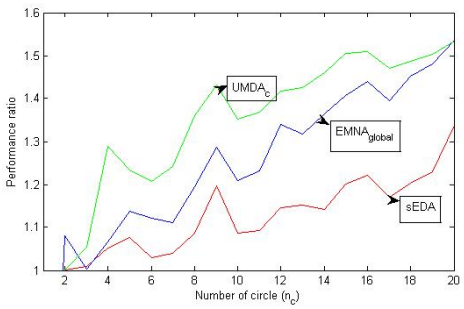

Figure 1. Median Performance of UMDA_c, $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA on CiaS problem.

similar but when n_c increases, the performance of sEDA also increases.

The nature of variables in the CiaS problems is that some variables are equally important and some are dependent on each other. Here, out of the total number of variables only 30 percent of variables are selected for modelling covariance matrix by sEDA. By analysing the results, we can conclude that sEDA captures the importance and dependency of variables in this problem very efficiently.

4.4. Examination of sEDA Behaviour

From a stability point of view, sEDA is much more stable than $\mathrm{EMNA}_{\mathrm{global}}$. In most of the cases where population size is less, $\mathrm{EMNA}_{\mathrm{global}}$ produces an ill-conditioned matrix [7] [9] which is a hindrance for obtaining any global solution. Although sEDA uses some characteristics of $\mathrm{EMNA}_{\mathrm{global}}$, it doest not produce any ill conditioned matrix.

To repair the ill posed covariance matrix a number of techniques has been found out and implemented which is clearly described in [7], [9]. A comparison table is discussed in [9] which shows different covariance matrix repairing methods. The methods include CMR, ECMR and ECMR0 which is implemented on different artificial test functions. The explanations of algorithms and parameter details are mentioned in [9].

Here we compare sEDA algorithm with the 3 algorithms discussed in [9] using Rosenbrock, Sumcan and $f9$ functions (described in Table 4). Except Sumcan, other 2 functions are minimization function. sEDA uses same population size and number of function evaluations as described in [9] with τ=0.3 and η=0.3 as selection ratios. The compare values shown in Table 7 summarizes the average performance of 100 independent runs of each algorithm. Table 7 as well as discussion from [9] revealed that CMR, ECMR and ECMR0 have approximately identical performance. From Table 7, we can conclude that, except Sumcan function, the performance of sEDA is better than other 3 algorithms in terms of the best fitness values, but sEDA requires more number of function evaluation in each case.

Table 7. Comparision of sEDA, CMR, ECMR and ECMR0. Bold font represents the best result.

Function	Alg.	Best fitness	No. of Evaluations
Rosenbrock	sEDA	**7.52E+00±3.3E-01**	3.0E+05±0.0E+00
	CMR	2.35E+02±1.2E+02	**3.4E+04±2.7E+04**
	ECMR	2.26E+02±9.9E+01	3.9E+04±4.6E+04
	ECMR0	2.16E+02±8.5E+01	4.5E+04±5.9E+04
Sumcan	sEDA	8.64E+02±1.1E+03	3.0E+05±0.0E+00
	CMR	**9.88E+04±7.9E+03**	4.9E+04±4.1E+04
	ECMR	9.73E+04±1.4E+04	5.2E+04±4.7E+04
	ECMR0	9.85E+04±9.7E+03	**4.9E+04±4.1E+04**
$f9$	sEDA	**-1.81E+04±2.1E+04**	3.0E+05±0.0E+00
	CMR	-4.93E+03±6.8E+02	5.9E+04±6.2E+04
	ECMR	-4.91E+03±7.0E+02	5.9E+04±6.2E+04
	ECMR0	-5.02E+03±6.8E+01	**5.6E+04±5.7E+04**

As discussed in Section 3, the value of η in sEDA determines the number of variables to be selected for covariance modelling.

Hence the value of η depends on the problem to be solved and some experimentation is expected to be required to find the optimal value in any given application. he definition in Table 4 for Schwefel function shows that, all the variables are dependent on each other. We can see from Table 5 and 6 that $\mathrm{EMNA}_{\mathrm{global}}$ performs very well in Schwefel function. Table 8 shows the comparison

Table 8. Role of η, in 10-D Schwefel function. Bold font represents the best result.

Alg.	Best fitness	No. of Evaluations
sEDA(η=1)	**7.42E-09±1.8E-09**	**5.9E+04±1.0E+03**
$\mathrm{EMNA}_{\mathrm{global}}$	7.51E-09±1.7E-09	1.3E+04±1.5E+03

between $\mathrm{EMNA}_{\mathrm{global}}$ and sEDA when η=1 in a 10-D Schwefel function. The performance of sEDA is better when compared to $\mathrm{EMNA}_{\mathrm{global}}$ with respect to the best fitness values as well as function evaluations required to achieve the solution. In other sense we can conclude that the role of η is significant if we can know the nature of variables in a problem.

5. Conclusion

This paper analysed classical EDAs over continuous variables in terms of dependency of variables and modelling and presented numerical issues as well as common difficulties w.r.t. performance of the model. Using different techniques and theoretical analysis, we have proposed a Gaussian based continuous EDA, called sEDA, a mechanism to control the amount of covariance parameters estimated within the Gaussian model, which clearly gives an idea about the number of dependent variables and important of these variables in a problem. A simple

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.

implementation of this framework, sEDA, has been experimentally compared with UMDA_c and $\mathrm{EMNA}_{\mathrm{global}}$. While considering artificial test functions and real world problems (CiaS problem), sEDA showed significantly better performance in most of the cases. Most of the results are better primarily due to better selection of variables for covariance modelling. The role of η is significant if we can know the nature of variables in a problem. The main advantage of sEDA is not only finding a good solution with less number of population, but also gives an idea of variable dependency and importance of the problem. Moreover, it doesn't show ill posed covariance matrix while modelling.

References

[1] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation*. Kluwer, 2001.

[2] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," in *Parallel Problem Solving from Nature - PPSN IV*, ser. Lecture Notes in Computer Science, vol. 1411. Springer, 1996, pp. 178–187.

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

[4] S. Kern, S. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos, "Learning probability distributions in continuous evolutionary algorithms–a comparative review," *Natural Computing*, vol. 3, no. 1, pp. 77–112, 2004.

[5] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.

[6] P. Bosman, J. Grahl, and D. Thierens, "Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift," *Parallel Problem Solving from Nature–PPSN X*, pp. 133–143, 2008.

[7] W. Dong and X. Yao, "Covariance matrix repairing in Gaussian based EDAs," in *Proc. Congress on Evolutionary Computation (CEC)*. IEEE, 2007, pp. 415–422.

[8] S. Baluja, "Population-Based Incremental Learning: A method for integrating genetic search based function optimization and competitive learning," School of Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-94-163, 1994.

[9] W. Dong and X. Yao, "Unified eigen analysis on multivariate Gaussian based Estimation of Distribution Algorithms," *Information Sciences*, vol. 178, no. 15, pp. 3000–3023, 2008.

[10] P. Bosman and D. Thierens, "Continuous iterated density estimation evolutionary algorithms within the IDEA framework," in *Proc. of the Optimization by Building and using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference (GECCO-2000)*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 197–200.

[11] M. Gallagher, M. Frean, and T. Downs, "Real-valued evolutionary optimization using a flexible probability density estimator," in *Proc. Genetic and Evolutionary Computation Conference (GECCO'99)*, W. Banzhaf and et al., Eds. San Francisco, CA: Morgan Kaufmann, 1999, pp. 840–846.

[12] M. Gallagher, "Multi-layer perceptron error surfaces: visualization, structure and modelling," Ph.D. dissertation, Dept. Computer Science and Electrical Engineering, University of Queensland, 2000.

[13] B. Yuan and M. Gallagher, "Playing in continuous spaces: Some analysis and extension of population-based incremental learning," in *Proc. Congress on Evolutionary Computation (CEC)*, R. S. et. al., Ed. IEEE, 2003, pp. 443–450.

[14] N. Ding, S. Zhou, and Z. Sun, "Optimizing continuous problems using estimation of distribution algorithm based on histogram model," *Simulated Evolution and Learning*, pp. 545–552, 2006.

[15] P. A. N. Bosman and D. Thierens, "IDEAs based on the normal kernels probability density function," Department of Computer Science, Utrecht University, (ftp://ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-2000/2000-11.ps.gz), Tech. Rep. 2000-11, Mar. 2000.

[16] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization by learning and simulation of Bayesian and Gaussian networks," University of the Basque Country, Spain, Tech. Rep. KZZA-IK-4-99, 1999.

[17] M. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[18] F. Campolongo, J. Cariboni, A. Saltelli, and W. Schoutens, "Enhancing the morris method," in *Sensitivity Analysis of Model Output. Proceedings of the 4th International Conference on Sensitivity Analysis of Model Output (SAMO 2004)*, 2005, pp. 369–379.

[19] A. Forrester, A. Sóbester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008, vol. 226.

[20] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer, 2003.

[21] N. Hansen, S. Finck, R. Ros, A. Auger *et al.*, "Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions," INRIA, Tech. Rep. RR-6829, 2009.

[22] M. Gallagher, "Investigating circles in a square packing problems as a realistic benchmark for continuous metaheuristic optimization algorithms," in *The VIII Metaheuristic International Conference MIC 2009*, 2009.

[23] I. Castillo, F. Kampas, and J. Pintér, "Solving circle packing problems by global optimization: numerical results and industrial applications," *European Journal of Operational Research*, vol. 191, no. 3, pp. 786–802, 2008.

Authorized licensed use limited to: Univ Politecnica de Madrid. Downloaded on March 09,2025 at 18:41:36 UTC from IEEE Xplore. Restrictions apply.