

Cellular Estimation Gaussian Algorithm for Continuous Domain

Yoa Martínez-López^a, Julio Madera^a, Ansel Y. Rodríguez-González^{b,c,*} and Stephen Barigye^d

^a*Department of Computer Sciences, Faculty of Informatics, Camagüey University, Camagüey City, Camagüey, Cuba*

^b*Mexican National Research Council (CONACyT), Mexico*

^c*CICESE-UT3, Ciudad del Conocimiento, Tepic, Nayarit, México*

^d*Department of Chemistry, McGill University, Montreal, Canada*

Abstract. Optimization algorithms are important in problems of pattern recognition and artificial intelligence, i.e., the image recognition, face recognition, data analysis, optical recognition, etc. Estimation distribution algorithms (*EDAs*) is kind of optimization algorithms based on substituting the crossover and mutation operators of the Genetic Algorithms by the estimation and later sampling the probability distribution learned from the selected individuals. However, a weakness of these algorithms is the efficiency in terms of the number of evaluations of the fitness function. In this paper, a Cellular Gaussian Estimation Algorithm (*CEGA*) for solving continuous optimization problems is proposed. *CEGA* is derived from evidence-based learning of independence and decentralized schemes of local populations. The experimental results showed that the present proposal reduces the number of evaluations of the fitness function in the search for optimums, maintaining its effectiveness in comparison to other algorithms of state-of-art using the same benchmark of continuous functions.

Keywords: Cellular EDA, learning, probabilistic graph model, Gaussian networks

1. Introduction

Estimation of Distribution Algorithms (*EDAs*) [1, 2] have been widely used to find solutions in discrete [3] and continuous [4] optimization problems. These kinds of algorithms are based on substituting the crossover and mutation operators of the Genetic Algorithms (*GAs*) [5, 6] by the estimation and later sampling the probability distribution learned from the selected individuals. In every optimization problem, there are dependencies between the variables, which are not inferred by most of the current optimization methods (Genetic Algorithms, Particle Swarm Optimization, etc.). To detect the dependencies, *EDAs* use statistical techniques. The main advantage of

EDAs over *GAs* is that they estimate the values of each variable using a probability distribution, while Genetic Algorithms seek a solution to a problem by directly coding the variables.

For continuous optimization problems, several *EDAs* have been proposed: *UMDA_c* [4], *PBIL_c* [7], *MIMIC_c* [8], *EMNA* (and its variants) [8–10] and *PolyEDA* [11]. However, a weakness of these algorithms, as well as the *EDAs* for discrete optimization, is the efficiency in terms of the number of evaluations of the objective function. In order to deal with this weakness for discrete optimization, a new kind of *EDAs* named the *Cellular EDAs* was proposed [12, 13], which allow for the decentralization of the individuals in the population. But in the best of our knowledge, this idea has not been used for continuous optimization problems.

*Corresponding author. Ansel Y. Rodríguez-González. E-mail: ansel@cicese.mx.

In this paper, a new kind *Cellular EDA*, the Cellular Estimation Gaussian Algorithm (*CEGA*), which learns the structure and parameters of Gaussian networks from local populations using independence tests and decentralized schemes to reduce the number of evaluations, preserving their effectiveness in the solution of continuous optimization problems is presented.

This paper is organized as follows. Section 2 presents a brief review of related work. Section 3 provides basic concepts. Section 4 describes the proposed algorithm. Section 5 presents the experimental results. Section 6 shows statistical validations and finally, Section 7 contains conclusions of this study.

2. Related work

The simplest of the *EDAs* used to find solutions in problems of continuous optimization is *UMDA_c* (Univariate Marginal Distribution Algorithm for Continuous Domain) [4], which assumes in each generation that the variables are independent following a normal distribution (other distributions can be taken into account).

Other *EDAs* developed for problems of continuous optimization are *PBIL_c* [7], *MIMIC_c* [8], *EMNA* (and its variants) [8–10], *PolyEDA* [11], *RECEDA* [14], *EBCOA* [15] and *CMA – ES* [16, 17] and *sEDA* [18].

PBIL_c (Population Based Incremental Learning for Continuous Domain) [7] follows a scheme similar to the *UMDA_c*. While *MIMIC_c* algorithm (Mutual Information Maximization for Clustering for Continuous Domain) [8] is an extension of the *MIMIC* algorithm [19], in which the variables take real values, sets the model to the empirical data and analyzes only the relationships that exist between pairs of variables in the same way as *UMDA*.

EMNA (Estimation of Multivariate Normal Algorithm) [9] uses a multivariate normal density function to learn the factorization of the selected individuals. Alternatives to this algorithm (also known as *EMNA_{global}*) are *EMNA_a* and *EMNA_i* [10]. Both generate a single individual. The first is adaptive, incorporating the individual if it is better than the worst in the population. The second is incremental, adds the individual to the population. Likewise, the *EGNA* (Estimation of Gaussian Network Algorithm) [8] uses learning and simulation of Gaussian networks. One of the proposed variants is the *EGNA_{BGe}* [10]. In this variant the induction of

the model is done by a scoring method that uses a Bayesian measure. The *EGNA_{BIC}* [10] follows a similar scheme but uses the *BIC* metric for continuous domain and the *EGNA_{ee}* [10] (Algorithm with Gaussian Network Estimation with Arcs Exclusion) uses independence detection to build the Gaussian network.

PolyEDA [11] is a combination of estimation algorithms of distributions and constraints with linear inequalities.

RECEDA [14] (Real-Coded Estimation of Distribution Algorithm) uses only the means and covariance matrix of the variables estimated from the selected promising individuals of a population, to generate offspring.

EBCOA [15] (Evolutionary Bayesian Classifier-based Optimization Algorithm) generates the probabilistic graphical model that will be applied for sampling the next population, taking into account the fitness function as a new variable.

CMA – ES [16, 17] (Covariance Matrix Adaptation Evolution Strategy) uses information embedded in the evolution path to accelerate the convergence. It is a completely derandomized self-adaptation scheme, which the full covariance matrix of the probability density function is adapted for the mutation of the objective parameter vector.

sEDA [18] (Screening EDA) identifies important variables and uses these to control the degree of covariance modeling in the Gaussian EDA model. The algorithm provides improved numerical stability and can use a smaller selected population.

On the other hand, a cellular *EDA* for discrete optimization is a collection of collaborative and decentralized *EDAs*, also called member algorithms that develop overlapping populations [12, 13]. A distinctive feature of this kind of algorithm is that they are decentralized at the level of the algorithms and selection in another evolutionary algorithm usually occurs at the recombination level. The organization of the cellular *EDAs* is based on the traditional structure of grids, where each grid contains a set of neighboring individuals, which form a cell.

3. Basic concepts

In this section the relevant concepts for the proposed algorithm like probabilistic graphics models, Gaussian networks, regularization of probabilistic parameters, learning strategy, neighborhoods and benchmark of continuous functions, are provided.

3.1. Probabilistic graphics models

The main characteristic of most *EDAs* (cellular or not) is the use of probabilistic graphic models to detect the dependencies between the variables of the problem to be solved. It is the most complex step lies within the estimation of the probability distribution.

The Graphical Models (*MGs*) are tools that allow representing joint probability distributions. Probabilistic Graphical Models (*MGPs*) are graphs in which the nodes represent random variables and the arcs represent conditional dependence relations. These graphs provide a compact way to represent the probability distribution [20].

The *MGPs* used by the *EDAs* vary depending on the domain of the problem variables. If these variables are discrete, Bayesian networks are used. If, on the contrary, they are continuous variables, Gaussian networks are used. There is the possibility of generating hybrid probabilistic models, adapted for problems with discrete and continuous variables, but its out of the scope of this paper.

3.2. Gaussian networks

Gaussian networks are described in [21] as Interaction Graph Models for the multivariate normal distribution. These models in the Graphical Gaussian Model (GGM) is based on conditional independence. They only contain non-directed arcs and this makes them not only one of the conceptually simplest models but also one of the most applied.

In the X data is assumed to be mutually independent, with a normal p -varied distribution $N_p(\mu, \Sigma)$, a mean vector $\mu = (\mu_1, \dots, \mu_n)^T$ and the defined positive matrix of variance-covariance $\Sigma = (\sigma_{ij})$, where $1 \leq i, j \leq p$. Using the formula $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$, the covariance matrix can be decomposed into components of variance, with σ_i^2 , and the Pearson's correlation $P = (\rho_{ij})$. The multivariate normal density is given by:

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

In exponential terms, the alternative parameterization is given through canonical parameters that are defined as $\Omega = \Sigma^{-1}$ and $\beta = \Sigma^{-1}\mu$, so the multivariate normal density of the above equation can be redefined as:

$$f(x) = \exp \left\{ \alpha + \beta^T x - \frac{1}{2} x^T \Omega x \right\} = \exp \left\{ \alpha + \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p w_{ij} x_i x_j \right\} \quad (2)$$

Where, α is the normalization constant and $\Omega = (ij)$ is called the precision or concentration matrix.

The proposed algorithm *CEGA* is based on the estimation of the covariance matrix.

3.3. Regularization of probabilistic parameters

Regularization techniques are used frequently in statistical learning to obtain a more robust estimation of probabilistic models with a small prediction error. The regularized estimation model tries to reduce the general prediction error of the estimation model by reducing the large variance caused by the prediction of new samples, at the cost of introducing a small margin of error in the model.

The estimation model in the *EDAs* presents some characteristics that motivate the use of regularization techniques. The lack of adequate statistics can lead to the model becoming very partial in specific regions of the search space, this reduces its ability to generalize which is an important factor when sampling the model. The use of regularization can reduce the general error of the model estimated in the *EDAs*.

Another important aspect is the scalability of the *EDAs* with respect to the size of the problem since estimating the model of the probability distribution of large search spaces requires large populations, given that the estimation of the model and the subsequent sampling in the *EDAs* are large consumers of time, the performance of the algorithm can drop precipitously if the size of the population is very large. Estimating a comparable quality model using much smaller populations is an important requirement in these algorithms [22].

In [23] a discussion of the estimation technique with contraction or restriction, this gives the *EDA* the ability to build better models of the search distributions on small populations, was proposed. The merit of this contraction technique is that it improves the efficiency and precision of the estimation and provides a well-conditioned and positively defined covariance matrix, which is an important aspect to calculate its inverse. The idea of the estimation with contraction is simple, assuming that we have an unrestricted model of large dimensions and a reduced submodel with restricted dimensions, by adjusting each of the two

different models to the corresponding observed data we obtain estimates. The unrestricted estimate will exhibit a relatively high variation due to the greater number of parameters that need to be adjusted, while its small-sized counterpart will have lower variance, but potentially it will also be considerable as an estimator of the true unrestricted model [23].

3.4. Learning strategy

A critical issue in a cellular *EDA* is the use of a strategy that they learn from the probabilistic model because they are usually not efficient from the evaluative point of view, which can affect the performance of the algorithm, so the learning of the structure and parameters from local populations, can be one of the alternatives to solve this problem. As The proposed algorithm *CEGA* uses the Shrinkage estimator algorithm from the covariance matrix of a data set [24] as learning operator.

3.5. Neighborhoods

In a cellular *EDA* the reproductive cycle is executed within each number of individuals of the local population, which is usually called a cell, they have their own local populations defined by neighboring subpopulations and at the same time, a cell belongs to many local populations. The set of all the cells defines a partition of the global population.

A neighborhood is a set of individuals that are neighbors to a given one, that is, they are located close to it in the population according to a given spatial topology of the grid [25]. The neighborhood of 5 individuals, commonly called NEWS (North, East, West, South), considers the central individual and the immediately superior, inferior, left and right. There are other neighborhoods, as in the case of One, L9, C9, C13 or compact C25 and C41, using a neighborhood of smaller radius causes the solutions to spread more slowly through the population, inducing a lower global selective pressure and maintaining greater genetic diversity than when using larger neighborhoods, as shown in Fig 1.

The proposed algorithm *CEGA* uses the neighborhoods One, L5, L9, C9, C13, C25 and C41.

3.6. Benchmark of continuous functions

Let $x = (x_1, x_2, \hat{A} \cdot \hat{A} \cdot \hat{A} \cdot, x_N)$ be a N -dimensional vector, $x_i \in \mathbb{R}$, the fitness functions $F : \mathbb{R}^N \rightarrow \mathbb{R}$, with the optimal value located at 0 [26], defined below are used to test the proposed algorithm *CEGA*.

Griewangk function:

$$F_{griewangk}(x) = 1 + \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (3)$$

where $-600 \leq x_i \leq 600$

Rastrigin's function:

$$F_{rastrigin}(x) = \sum_{i=1}^N [x_i^2 - 10\cos(2\pi x_i) + 10] \quad (4)$$

where $-5.12 \leq x_i \leq 5.12$

Rosenbrock's function:

$$F_{rosenbrock}(x) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (5)$$

where $-30 \leq x_i \leq 30$

Sphere function:

$$F_{sphere}(x) = \sum_{i=1}^N x_i^2 \quad (6)$$

where $-600 \leq x_i \leq 600$

Ackley's function:

$$F_{ackley}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^N \cos(2\pi x_i)\right) + 20 + e \quad (7)$$

where $-6 \leq x_i \leq 6$

4. Proposed algorithm

Starting from a population randomly generated, Cellular Estimation Gaussian Algorithm (*CEGA*) consists of loop of iterations until a termination criteria be met. For each iteration of *CEGA* exactly one iteration of all the member algorithms is done. Each of these member algorithms is responsible for updating exactly one subpopulation, and this is made by applying a local *EDA* model to the population composed of its individuals and those of its neighbor subpopulations (steps 5 to 7). The new individuals generated by the local learning and sampling steps a

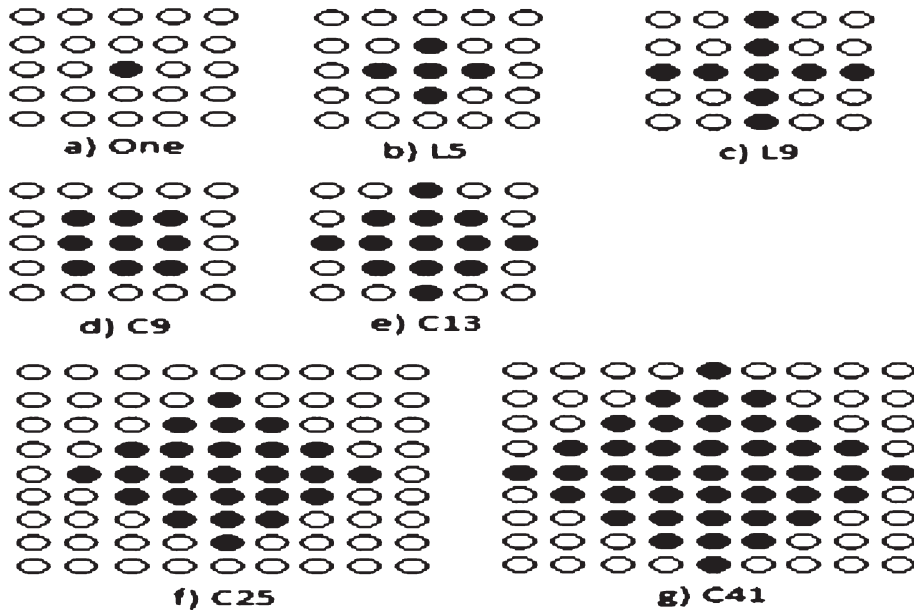


Fig. 1. Representation of different neighborhoods.

Algorithm 1 Cellular Estimation Gaussian Algorithm (*CEGA*)

```

1:  $t \leftarrow 1$ 
2: Generate  $N$  individuals randomly
3: while not terminationCriteria do
4:   for all cell do
5:     Select locally  $M \leq \text{SizeOf}(\text{Neighborhood}) \times \text{SizeOf}(\text{cell})$  individuals of the neighborhood
6:     Estimate the Gaussian distribution  $G(x, t)$  of the  $M$  selected individuals
7:     Generate  $\text{SizeOf}(\text{cell})$  new individuals according to the distribution  $G(x, t)$ 
8:     Insert the generated individuals in the same cell of an auxiliary population
9:   end for
10:  Replace the current population with the auxiliary one
11:  Compute and update the statistics
12:   $t \leftarrow t + 1$ 
13: end while

```

replaced in a temporal population (step 8). The successive populations replace each other at once (step 10). In this step, the old population can be taken into account (i.e., replacing a individuals if the new one is better) or not (always adding the new string to the next population). Finally, computing basic statistics in step 11.

The pseudo code of the proposed algorithm, *CEGA*, is presented in Algorithm 1.

5. Experimental results

In this section, the behavior of the proposed algorithm *CEGA* is evaluated. First, a compari-

son in terms of the approximation to the optimum and the numbers of iterations and evaluations needed, for different neighborhoods is presented over the continuous functions: Griewangk, Rastrigin's, Rosenbrock's, Sphere and Ackley's. After that *CEGA* is compared with the other continuous EDAs reported in the literature for the same continuous functions.

To evaluate the behavior of the proposed algorithm *CEGA*, it was executed 25 times, for each objective function and each neighborhood shown in the Fig 1, using $1E - 8$ as margin of error, 0.3 as truncation threshold and selection elitism of an individual. The dimension N of the vector x was set at 10. Table 1

Table 1
Results of CEGA for the functions: Griewangk, Rastrigin, Rosenbrock, Sphere and Ackley

Function	Neighborhood	Optimum	stDev	Eval	stDev
Griewangk	One	-4.83E+05	2.56E+05	12290.20	4408.6
	L5	-2.27E-09	2.05E-09	37510.60	325.78
	L9	1.08E-13	1.61E-13	11908.00	0.00
	C9	-6.33E-14	1.23E-13	13251.68	250.24
	C13	-1.23E-13	1.94E-13	12113.96	79.80
	C25	-4.95E-14	8.75E-14	11980.00	0.00
	C41	-8.56E+00	13.72	12130.96	13.71
Rastrigin	One	-3.75E+05	2.53E+05	12513.64	1496.83
	L5	-1.79E+05	1.80E+05	13719.68	340.13
	L9	-9.09E-02	2.28E-01	11229.92	110.48
	C9	-2.84E-02	1.16E-01	11357.60	199.50
	C13	-5.23E-01	2.56E+00	11213.96	79.80
	C25	-3.41E-02	1.25E-01	11198.00	0.00
	C41	-2.84E-02	0.08	12451.88	0.08
Rosenbrock	One	-5.12E+07	2.98E+07	17949.08	459.57
	L5	-4.42E+05	2.71E+05	18204.44	781.88
	L9	-1.02E+05	2.24E+05	15698.72	423.52
	L9	-6.89E-09	1.69E-09	17221.84	1513.57
	C13	-2.22E-09	2.70E-09	13544.12	175.44
	C25	-1.51E-09	2.10E-09	13208.96	335.41
	C41	-5.59E-11	0	11389.52	2.79E-10
Sphere	One	-5.08E+05	2.78E+05	10774.00	162.89
	L5	-5.32E+05	2.63E+05	16863.8	162.89
	L9	-2.96E+05	2.19E+05	15379.52	203.45
	C9	-3.66E+05	2.33E+05	15475.28	182.84
	C13	-1.45E+05	1.72E+05	13703.72	182.84
	C25	-2.74E+05	2.79E+05	13240.88	289.56
	C41	-3.82E-02	0.14	10373.56	0.13
Ackley	One	-4.66E-11	1.28E-10	16439.80	0.00
	L5	-9.34E-10	1.74E-09	10486.72	182.84
	L9	-1.56E-09	2.47E-09	10254.24	208.6
	C9	-2.39E-09	2.89E-09	10333.12	209.87
	C13	-1.84E-09	2.91E-09	10363.56	202.14
	C25	-1.81E-09	2.82E-09	10789.00	282.14
	C41	-4.44E-16	0	10069.32	3.01E-31

shows the results (gray cells contain the best values). In addition to the optimal value obtained for each case, Table 1 shows, the amount of the evaluations of the fitness functions, and the standard deviation of both for the 25 executions.

The obtained results show that *CEGA* for the Griewangk function, the *L9* neighborhood yielded the best result, with the value of the found optimum being $-1.08E - 13$. On the other hand, the best result for the Rastrigin function was using the *C25* neighborhood, with an optimum value of $-3.41E - 02$ obtained. Moreover, for the Rosenbrock function, the neighborhood *C41* yielded the best result, with the value of the found optimum being $-5.59E - 11$. In addition, for the Sphere function, the best neighborhood was *C41*, with an optimum value of $-3.82E - 02$. Likewise, the best result for the Ackley function was the neighborhood *C41*, with an optimum value of $-4.44E - 16$.

To compare the behavior of the proposed algorithm *CEGA* with the other continuous *EDAs* reported in the literature (*RECEDA* [14], *EGNA_{BGe}* [10], *EGNA_{ee}* [10], *EMNA_{global}* [9], *sEDA* [18], *EBCOA_{NB}* [15], *UMDA_c* [4], *MIMIC_c* [8], *CMA - ES* [16, 17]), all algorithms was executed 25 times, for each objective function with similar conditions. Table 2 shows the results (gray cells contain the best values)¹.

As observed in Table 2, the proposed algorithm *CEGA* yields the best performance for all fitness function. *CEGA* with neighborhood *C41* yields superior performance than other algorithms reported in the literature, for the functions: Ackley, Sphere, Rosenbrock, while *CEGA* with neighborhood *L9* was the best for the Griewangk's function and *CEGA*

¹The symbol * represents that there are not results reported in the literature for this algorithm and fitness function.

Table 2

Results of CEGA and other continuous EDAs reported in the literature for the functions: Griewangk, Rastrigin, Rosenbrock, Sphere and Ackley

Algorithm	Parameter	Ackley	Sphere	Griewangk	Rosenbrock	Rastrigin
<i>RECEDA</i>	Best Fitness	8.38E-07	7.40E-07	*	*	*
	Fitness Evaluations	20900	15600	*	*	*
<i>EGNA_{BGe}</i>	Best Fitness	8.50E-06	7.00E-06	9.20E-02	8.60E+00	*
	Fitness Evaluations	22904	14884	54784	26375	*
<i>EGNA_{ee}</i>	Best Fitness	7.90E-06	6.70E-06	5.40E-02	8.60E+00	*
	Fitness Evaluations	22983	14884	58654	28889	*
<i>EMNA_{global}</i>	Best Fitness	8.40E-09	7.67E-09	7.32E-09	7.80E+00	6.91E-09
	Fitness Evaluations	210000	140000	140000	300000	220000
<i>sEDA</i>	Best Fitness	8.27E-09	6.98E-09	6.82E-09	7.51E+00	1.99E-02
	Fitness Evaluations	78000	55000	60000	300000	78000
<i>EBCOA_{NB}</i>	Best Fitness	7.70E-06	4.40E-06	3.00E-02	4.40E+03	*
	Fitness Evaluations	18116	19632	33597	18235	*
<i>UMDA_c</i>	Best Fitness	8.50E-09	6.91E-09	5.40E-02	8.70E+00	7.32E-09
	Fitness Evaluations	200000	150000	58814	52589	220000
<i>MIMIC_c</i>	Best Fitness	7.80E-06	6.70E-06	8.20E-02	8.70E+00	*
	Fitness Evaluations	23382	15163	59093	44968	*
<i>CMA – ES</i>	Best Fitness	1.90E-07	3.70E-08	2.70E-08	3.90E-08	*
	Fitness Evaluations	23962	13802	14562	44082	*
<i>CEGA – one</i>	Best Fitness	-4.66E-11	-5.08E+05	-4.83E+05	-5.12E+07	-3.75E+05
	Fitness Evaluations	16440	10774	12290	17949	12514
<i>CEGA – L5</i>	Best Fitness	-9.34E-10	-5.32E+05	-2.27E-09	-4.42E+05	-1.79E+05
	Fitness Evaluations	10487	16864	37511	18204	13720
<i>CEGA – L9</i>	Best Fitness	-1.56E-09	-2.96E+05	-1.08E-13	-1.02E+05	-9.09E-02
	Fitness Evaluations	10254	15380	11908	15699	11230
<i>CEGA – C9</i>	Best Fitness	-2.39E-09	-3.66E+05	-6.33E-14	-6.89E-09	-2.84E-02
	Fitness Evaluations	10333	15475	13252	17222	11358
<i>CEGA – C13</i>	Best Fitness	-1.84E-09	-1.45E+05	-1.23E-13	-2.22E-09	-5.23E-01
	Fitness Evaluations	10364	13704	12114	13544	11214
<i>CEGA – C25</i>	Best Fitness	-1.81E-09	-2.74E+05	-4.95E-14	-1.51E-09	-3.41E-02
	Fitness Evaluations	10789	13241	11980	13209	11198
<i>CEGA – C41</i>	Best Fitness	-4.44E-16	-3.82E-02	-8.56E+00	-5.59E-11	-2.84E-02
	Fitness Evaluations	10069	10374	12131	11390	12452

with neighborhood C25 was the best for the Rastrigin's function, evidenced by the lowest optimum value, achieved with, the least number of evaluations.

6. Statistical validations

With the experimental results of *CEGA*, two alternative methods were carried out based on non-parametric tests [27–29]:

- Application of Iman and Davenport's test [30] and Holm's method [31] as post hoc procedures. The first test can be employed to see whether there are significant statistical differences among the algorithms in certain groups (three or more algorithms). If differences are detected, then Holm's test is used to compare the best ranking algorithm (control algorithm) with the remaining ones.

- Application of the Wilcoxon [32] matched-pairs signed-ranks test. With this test, the results of two algorithms can be directly compared.

6.1. Analysis of neighborhoods

For this comparison, the amount of the evaluations of the fitness functions is taking into account for the continuous functions: Ackley, Sphere, Griewangk, Rosenbrock and Rastrigin. Wilcoxon's test is applied to compare if exist the difference statistically significant or not between different Neighborhoods. In this test, the values of $R-$ and $R+$ (associated with the control algorithm in comparison) are specified (the lowest ones, which correspond to the worst results), together with the p -values computed for this test and whether the hypothesis is rejected (the p -value is lower than the significance value) or not. The results of this test are showed in the Table 3.

Table 3
Results obtained by the Wilcoxon's test for all algorithms

Algorithm 1	Algorithm 2	R+	R-	p-value	Hypothesis
<i>CEGA</i> – <i>one</i> vs	<i>CEGA</i> – <i>L5</i>	12	3	0.17753	Accept
<i>CEGA</i> – <i>C9</i> vs	<i>CEGA</i> – <i>one</i>	9	6	0.589639	Accept
	<i>CEGA</i> – <i>L5</i>	15	0	0.030971	Reject
<i>CEGA</i> – <i>L9</i> vs	<i>CEGA</i> – <i>one</i>	11	4	0.280713	Accept
	<i>CEGA</i> – <i>L5</i>	15	0	0.030971	Reject
	<i>CEGA</i> – <i>C9</i>	15	0	0.030971	Reject
<i>CEGA</i> – <i>C13</i> vs	<i>CEGA</i> – <i>one</i>	12	3	0.17753	Accept
	<i>CEGA</i> – <i>L5</i>	15	0	0.030971	Reject
	<i>CEGA</i> – <i>L9</i>	10	5	0.418492	Accept
	<i>CEGA</i> – <i>C9</i>	14	1	0.059058	Accept
<i>CEGA</i> – <i>C25</i> vs	<i>CEGA</i> – <i>one</i>	12	3	0.17753	Accept
	<i>CEGA</i> – <i>L5</i>	14	1	0.059058	Accept
	<i>CEGA</i> – <i>L9</i>	10	5	0.418492	Accept
	<i>CEGA</i> – <i>C9</i>	13	2	0.105645	Accept
	<i>CEGA</i> – <i>C13</i>	11	4	0.280713	Accept
<i>CEGA</i> – <i>C41</i> vs	<i>CEGA</i> – <i>one</i>	15	0	0.030971	Reject
	<i>CEGA</i> – <i>L5</i>	15	0	0.030971	Reject
	<i>CEGA</i> – <i>L9</i>	10	5	0.418492	Accept
	<i>CEGA</i> – <i>C9</i>	13	2	0.105645	Accept
	<i>CEGA</i> – <i>C13</i>	11	4	0.280713	Accept
	<i>CEGA</i> – <i>C25</i>	11	4	0.280713	Accept

It can be seen from Table 3 that *CEGA* – *one* obtained better results than the *CEGA* – *L5* (the *R*+ values are higher than the *R*– ones), but this difference is not statistically significant, which *p* – *value* > 0.05. Moreover, the *CEGA* – *L9* obtained better result than *CEGA* – *one*, *CEGA* – *L5* and *CEGA* – *C9*, but this difference is not statistically significant with regard to *CEGA* – *one*, while it was statistically better than *CEGA* – *L5* and *CEGA* – *C9*. Additionally, *CEGA* – *C9* is not statistically better than *CEGA* – *one*, but is statistically significant than *CEGA* – *L5*. The *CEGA* – *C13* was statistically better than *CEGA* – *L5*, but this difference is not statistically significant with regard to *CEGA* – *one*, *CEGA* – *L9* and *CEGA* – *C9*. In addition, *CEGA* – *C25* is not statistically better than *CEGA* – *one*, *CEGA* – *L5*, *CEGA* – *L9*, *CEGA* – *C9* and *CEGA* – *C13*. Finally, *CEGA* – *C41* is statistically significant than *CEGA* – *one*, *CEGA* – *L5*, but this difference is not statistically significant with regard to *CEGA* – *L9*, *CEGA* – *C9*, *CEGA* – *C13* and *CEGA* – *C25*.

6.2. Comparison with other continuous EDAs reported in the literature

In this study, the non-parametric tests are applied to analyze the behavior of *CEGA*, in order to obtain the best algorithm, taking into account the fitness evaluations for the continuous functions: Ackley, Sphere, Griewangk, Rosenbrock and Rastrigin. In Table 4 and

Table 4
Average ranking of the algorithms based on the Friedman test

Algorithm	Ranking
<i>CEGA</i> – <i>one</i>	5.000
<i>CEGA</i> – <i>L5</i>	8.000
<i>CEGA</i> – <i>L9</i>	4.000
<i>CEGA</i> – <i>C9</i>	6.000
<i>CEGA</i> – <i>C13</i>	3.500
<i>CEGA</i> – <i>C25</i>	3.250
<i>CEGA</i> – <i>C41</i>	1.750
<i>EGNA</i> _{BGe}	8.625
<i>EGNA</i> _{ee}	9.375
<i>EMNA</i> _{global}	14.625
<i>sEDA</i>	13.625
<i>EBCOA</i> _{NB}	9.000
<i>UMDA</i> _c	13.500
<i>MIMIC</i> _c	11.000
<i>CMA</i> – <i>ES</i>	8.750

Fig. 2 the average ranking based on the Friedman [33] test of the all algorithms is showed (gray cell in table contains the best rank).

As can be observed, the best ranking is obtained by *CEGA* – *C41*, the second place, *CEGA* – *C25* and third, *CEGA* – *L9*, which had good development.

Iman-Davenport's test is carried out (employing *F*-distribution 14 degrees of freedom for *Nds* = 45.9), in order to find statistical differences among the algorithms, obtaining a *p* – *value* of 0.000029. The results of Iman-Davenport's test show that *CEGA* – 41 presents indeed significant performance differences in the group (*p* – *value* < 0.05).

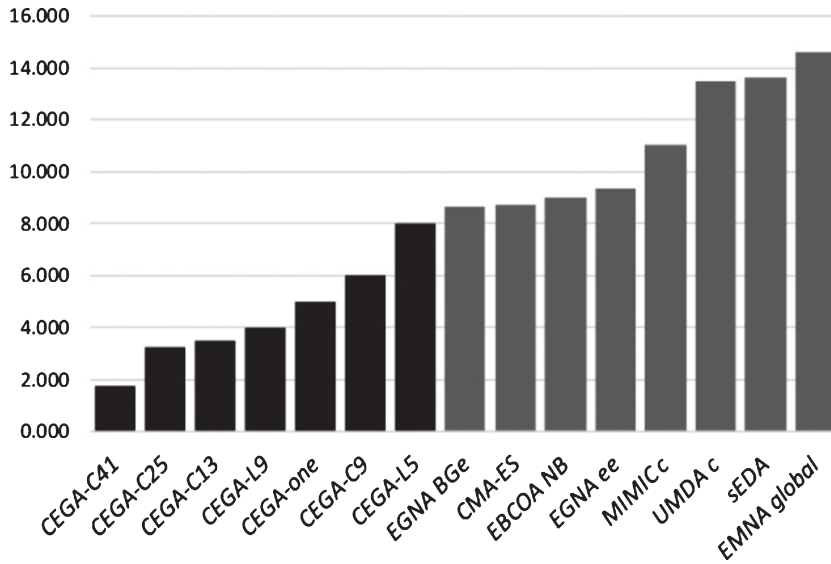


Fig. 2. Average ranking of the algorithms based on the Friedman test.

Table 5
Holm's test applied to compare *CEGA* – *C41* with the rest of algorithms

i	Algorithm	$z = \frac{R_0 - R_i}{SE}$	$p - value$	$\frac{\alpha}{i}$	Hypothesis
14	<i>EMNA_{global}</i>	4.071432	0.000047	0.003571	Reject
13	<i>sEDA</i>	3.755205	0.000173	0.003846	Reject
12	<i>UMDA_c</i>	3.715676	0.000203	0.004167	Reject
11	<i>MIMIC_c</i>	2.925107	0.003443	0.004545	Reject
10	<i>EGNA_{ee}</i>	2.411237	0.015899	0.005000	Reject
9	<i>EBCOA_{NB}</i>	2.292651	0.021868	0.005556	Reject
8	<i>CMA – ES</i>	2.213594	0.026857	0.006250	Reject
7	<i>EGNA_{BGe}</i>	2.174066	0.029700	0.007143	Reject
6	<i>CEGA – L5</i>	1.976424	0.048107	0.008333	Reject
5	<i>CEGA – C9</i>	1.343968	0.178959	0.010000	Accept
4	<i>CEGA – one</i>	1.027740	0.304072	0.012500	Accept
3	<i>CEGA – L9</i>	0.711512	0.476767	0.016667	Accept
2	<i>CEGA – C13</i>	0.553399	0.579991	0.025000	Accept
1	<i>CEGA – C25</i>	0.474342	0.635256	0.000000	Accept

Holm's test is applied to compare *CEGA* – *C41* with the rest of algorithms; for this test, the algorithms are ordered in descending order, according to rank. Table 5 contains all the computations associated with Holm's procedure ($z = \frac{R_0 - R_i}{SE}$, $p - value$, $\frac{\alpha}{i}$, and Hypothesis).

For this comparison, the results obtained with *CEGA* – *C41* are significantly superior to *EMNA_{global}*, *sEDA*, *UMDA_c*, *MIMIC_c*, *EGNA_{ee}*, *EBCOA_{NB}*, *CMA – ES*, *EGNA_{BGe}* and *CEGA – L5*, which the test is rejected, but the differences are insignificant with *CEGA – C9*, *CEGA – one*, *CEGA – L9*, *CEGA – C13*, *CEGA – C25*, which the test is accepted.

7. Conclusions

In this paper, the Cellular Gaussian Estimation Algorithm (*CEGA*) for solving continuous optimization problems was proposed. The experimental results showed that *CEGA* can reduce the number of evaluations of the fitness functions in solving continuous optimization problems. This was evidenced through the results obtained when studying the different neighborhoods of local populations. In addition, it was shown that *CEGA* decreases number of the evaluations of the fitness functions, compared to other *EDAs* in the literature for solving continuous optimization problems.

From statistical analysis performed, it can be concluded that the use of different neighborhoods by *CEGA* does not produce significantly better result, according to Wilcoxon's test. However, *CEGA* using any neighborhoods, has better performance than the other continuous *EDAs* in the literature according to the ranking determined by the Friedman's test, taking into account the fitness evaluations.

The best ranking position was achieved by *CEGA* using the neighborhood *C41* and the Iman-Davenport's test shows that this choice is significantly superior to other continuous *EDAs* in the literature.

As future work, applying and adapting *CEGA* for practical problems where there are dependencies amongst the variables and the number of evaluations of the fitness functions is restricted, will be an interesting and imperative research direction. Another interesting work will be compare *CEGA* with other algorithms like Differential Evolution Algorithm, Particle Swarm Optimization and Firefly Algorithm in this kind of problems.

References

- [1] H. Mühlenbein and G. Paass, From recombination of genes to the estimation of distributions I. Binary parameters, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 1996, pp. 178–187.
- [2] K.-L. Du and M. Swamy, Estimation of Distribution Algorithms, in: *Search and Optimization by Metaheuristics*, Springer, 2016, pp. 105–119.
- [3] P. Larrañaga, J.A. Lozano and H. Mühlenbein, Estimation of distribution algorithms applied to combinatorial optimization problems, *Revista Iberoamericana de Inteligencia Artificial* **19** (2003), 149–168.
- [4] P. Larrañaga, R. Etxeberria, J. Lozano, J. Peña, J. Pe, et al., *Optimization by learning and simulation of Bayesian and Gaussian networks*, 1999.
- [5] C. Li, X. Chu, Y. Chen and L. Xing, A knowledge-based technique for initializing a genetic algorithm, *Journal of Intelligent & Fuzzy Systems* **31**(2) (2016), 1145–1152.
- [6] Y. Zhou and H. Fan, Research on multi objective optimization model of sustainable agriculture industrial structure based on genetic algorithm, *Journal of Intelligent & Fuzzy Systems* **35**(3) (2018), 2901–2907.
- [7] M. Sebag and A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 1998, pp. 418–427.
- [8] P. Larrañaga, R. Etxeberria, J.A. Lozano and J.M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, in: *Genetic and Evolutionary Computation Conference*, 2000, pp. 201–204.
- [9] P. Larrañaga, J.A. Lozano and E. Bengoetxea, Estimation of distribution algorithms based on multivariate normal and Gaussian networks, Technical Report, Technical Report EHUKZAA-IK-1, 2001.
- [10] P. Larrañaga and J.A. Lozano, Estimation of distribution algorithms: A new tool for evolutionary computation, Vol. 2, Springer Science & Business Media, 2001.
- [11] J. Grahl and F. Rothlauf, PolyEDA: Combining Estimation of Distribution Algorithms and Linear Inequality Constraints, in: *Genetic and Evolutionary Computation – GECCO 2004*, K. Deb, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 1174–1185. ISBN ISBN 978-3-540-24854-5.
- [12] E. Alba, J. Madera, B. Dorronsoro, A. Ochoa and M. Soto, Theory and practice of cellular UMDA for discrete optimization, in: *Parallel Problem Solving from Nature-PPSN IX*, Springer, 2006, pp. 242–251.
- [13] Y. Martínez-López, J. Madera Quintana and I. Leguen de Varona, Algoritmos evolutivos con estimación de distribución celulares, *Revista Cubana de Ciencias Informáticas* **10** (2016), 159–170.
- [14] T.K. Paul and H. Iba, Real-coded estimation of distribution algorithm, in: *Proceedings of The Fifth Metaheuristics International Conference*, Citeseer, 2003.
- [15] T. Miquélez, E. Bengoetxea and P. Larrañaga, Evolutionary Bayesian classifier-based optimization in continuous domains, in: *Asia-Pacific Conference on Simulated Evolution and Learning*, Springer, 2006, pp. 529–536.
- [16] N. Hansen and S. Kern, Evaluating the CMA evolution strategy on multimodal test functions, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 282–291.
- [17] N. Hansen, The CMA evolution strategy: A comparing review, in: *Towards a New Evolutionary Computation*, Springer, 2006, pp. 75–102.
- [18] K.M. Mishra, *Data-driven analysis of variables and dependencies in continuous optimization problems and estimation of distribution algorithms*, 2015.
- [19] J.S. De Bonet, C.L. Isbell Jr and P.A. Viola, MIMIC: Finding optima by estimating probability densities, in: *Advances in Neural Information Processing Systems*, 1997, pp. 424–430.
- [20] J. Madera and A. Ochoa, Una versión paralela del algoritmo MMHCEDA, ICIMAF, Department de Matemàtica Interdisciplinària, 2006.
- [21] J. Schaefer, Small-Sample Analysis and Inference of Networked Dependency Structures from Complex Genomic Data, PhD thesis, Imu, 2006.
- [22] H. Karshenas, R. Santana, C. Bielza and P. Larrañaga, Regularized continuous estimation of distribution algorithms, *Applied Soft Computing* **13**(5) (2013), 2412–2432.
- [23] A. Ochoa, Opportunities for expensive optimization with estimation of distribution algorithms, in: *Computational Intelligence in Expensive Optimization Problems*, Springer, 2010, pp. 193–218.
- [24] Y. Chen, A. Wiesel, Y.C. Eldar and A.O. Hero, Shrinkage algorithms for MMSE covariance estimation, *IEEE Transactions on Signal Processing* **58**(10) (2010), 5016–5029, ISSN1053-587X. doi: 10.1109/TSP.2010.2053029
- [25] B. Dorronsoro, E. Alba, G. Luque and P. Bouvry, A self-adaptive cellular memetic algorithm for the DNA fragment assembly problem, in: *Evolutionary Computation, 2008 CEC 2008(IEEE World Congress on Computational Intelligence) IEEE Congress on*, IEEE, 2008, pp. 2651–2658.
- [26] X. Li, K. Tang, M.N. Omidvar, Z. Yang, K. Qin and H. China, Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization, *Gene* **7**(33) (2013), 8.
- [27] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7**(Jan) (2006), 1–30.

- [28] S. García and F. Herrera, An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons, *Journal of Machine Learning Research* **9**(Dec) (2008), 2677–2694.
- [29] S. García, A. Fernández, J. Luengo and F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Information Sciences* **180**(10) (2010), 2044–2064.
- [30] R.L. Iman and J.M. Davenport, Approximations of the critical region of the fbietkan statistic, *Communications in Statistics-Theory and Methods* **9**(6) (1980), 571–595.
- [31] S. Holm, A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics* (1979), 65–70.
- [32] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics Bulletin* **1**(6) (1945), 80–83.
- [33] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, *The Annals of Mathematical Statistics* **11**(1) (1940), 86–92.