CELLULAR ESTIMATION BAYESIAN ALGORITHM FOR DISCRETE OPTIMIZATION PROBLEMS

Yoan Martínez-López [*1], Julio Madera [*], Gaafar Sadeq S. Mahdi[**], Ansel Y. Rodríguez-González [***]
[*]Facultad de Informática y Ciencias Exactas, Universidad de Camagüey, Camagüey, Cuba
[**] Universidad de Ciencias Informáticas, La Habana, Cuba
[***] CICESE-UT3, Nayarit, México

ABSTRACT
In this paper, a new Cellular Estimation Bayesian Algorithm for discrete optimization problems is presented. This class of stochastic optimization algorithm with learning from the structure and parameters of local populations are based on independence test and decentralized populations scheme, which can reduce the number of function evaluations solving for discrete optimization problems. The experimental results showed that this proposal reduces the number of evaluations in the search of the optimal for a benchmark discrete function with respect to other approaches of the literature. Also, it achieved better performance than them.

KEYWORDS: Cellular EDAs, Bayesian networks, learning, evolutionary algorithm.

MSC: 60-08

RESUMEN
En este documento, se presenta un nuevo algoritmo bayesiano de estimación celular para problemas de optimización discretos. Esta clase de algoritmo de optimización estocástica con aprendizaje de la estructura y los parámetros de las poblaciones locales se basa en la prueba de independencia y el esquema de poblaciones descentralizadas, lo que puede reducir el número de evaluaciones de funciones que resuelven problemas de optimización discretos. Los resultados experimentales mostraron que esta propuesta reduce el número de evaluaciones en la búsqueda del óptimo para funciones discretas de referencia con respecto a otros enfoques de la literatura. Además, tuvo mejores resultados con respecto a los algoritmos del estado del arte.

PALABRAS CLAVES: EDA celulares; Redes bayesianas; aprendizaje; algoritmo evolutivo.

1. INTRODUCTION

Estimation of Distribution Algorithms (EDAs) are a group of evolutionary algorithms (EAs) for solving optimization problems, which allow adjusting the model to the structure of a particular issue, by an estimation of probability distributions of the selected solutions [4]. The model bias is reflected by a probability distribution. These algorithms are based on meta-heuristics replace operator's crossover and mutation of individuals of genetic algorithms for the estimation and subsequent sampling of a probability distribution learned from the individuals selected from a population [15].For discrete optimization problems, some EDAs have been proposed: UMDA [2], PADA [12, 18]and SPADA [14]. Univariate Marginal Distribution Algorithm for Continuous Domain (UMDA), assumes in each generation that the variables are independent. Algorithm with Factorized Distribution based on Polytrees is published with the name of PADA [12, 18], which is named as Polytree Approximation Distribution Algorithm. PADA is an EDA that uses single connected models as a model of its distributions. On the other hand, SPADA [14], unlike those based on independence tests, requires computing higher-order distributions, as occurs with interconnected networks. However, as in the polytrees it is only possible to insert edges and their cost is lower than the interconnected networks. The local search algorithm using SPADA is a glutton procedure that checks for the existence of non-directed, rather than directed, cycles. On the other hands, cellular evolutionary algorithms [2, 1] are a type of evolutionary algorithms discrete groups based on spatial structures, where each individual interacts

[1]yoan.martinez@reduc.edu.cu

with its adjacent neighbor. An overlapped neighborhood helps to the exploration of the search space, while the exploitation takes place within a neighborhood by stochastic operators. In addition, a cellular EDA is a collection of collaborative and decentralized EDAs, also called members algorithms that develop overlapping populations [3]. A distinctive feature of this kind of algorithms is that they are decentralized level algorithms and other evolutionary algorithm selection usually occurs at the level of recombination [10, 9, 11].

2. CELLULAR ESTIMATION BAYESIAN ALGORITHM

We propose a new method that employs Bayesian Network in estimation and sampling phases in the cellular EDAs. Relevant concepts for the proposed algorithm like probabilistic graphics models, Bayesian networks, learning strategy, neighborhoods and benchmark discrete functions, are provided below.

2.1. Probabilistic Graphical Models.

Graphics Models (GM) are tools to represent joint probability distributions. Probabilistic Graphical Models (PGMs) [10] are graphs in which nodes represent random variables and arcs represent conditional dependence relations. These graphs provide a compact way to represent the probability distribution [5, 8]. PGMs used by EDA algorithms vary depending on the domain of the problem variables. If these variables are discrete Bayesian networks are used.

Bayesian network [6] is a type of GM that uses directed acyclic graphs (DAG), which takes into account the direction of the arcs. A Bayesian network is defined by the pair where G is a graph representing the dependency relationships between variables and P is the factorization of the probability distribution represented by G. Formally a Bayesian network on a set, V = {V_1, ...,V_n} of random variables defined. The factorization of the joint probability can be expressed as:

$$P\,(V) = P\,(V_1, V_2, \ldots\ldots, V_n) = \prod_{i=1}^{n} P\,(V_i - P_{ai}) \qquad (2.1)$$

The expression used to define a Bayesian network status Markov [16], each variable (Vi) is independent of any subset of the variables not a descendant of it, conditioned as a whole parent (P_{ai}).

Learning Bayesian Networks, [17] there are two basic techniques: learning based on constraints (constraint based learning) or algorithms that detect independence, and learning based on optimization metrics (search-and-score based learning) known as scoring methods. These models perform two basic tasks: first perform a structural learning to identify the topology of the network and from this estimate, the parameters (parametric learning) represented by conditional probabilities.

2.2. Learning Strategy.

A critical issue in a cellular EDA is using a strategy learning the probabilistic model because they usually are not efficient from the point of evaluative view, which can affect the performance of the algorithm, so learning the structure and parameters from local populations, it may be one of the alternatives to solve this problem. This learning is performed by generating several points, which the dependency graph are constructed in a Bayesian Network (learning dependency graph of Bayesian Networks ie. the structure) and then their probabilities are estimated from this graph dependence (learning parameters)[2].

Neighborhoods is a set of neighbor's individuals to a given one, ie., that are located next to it in a given population according to the grid [7, 13] spatial topology. The neighborhood of 5 individuals, commonly called NEWS (North, East, West, South), considered the central individual and immediately top, bottom, left and right. There are other neighborhoods, such as One, L9, C9, C13 or compact C25 and C41, using a neighborhood of smaller radius makes solutions extend more slowly by the population, leading to a lower overall selective pressure and maintaining greater genetic diversity when using larger neighborhoods, as shown in figure 1.

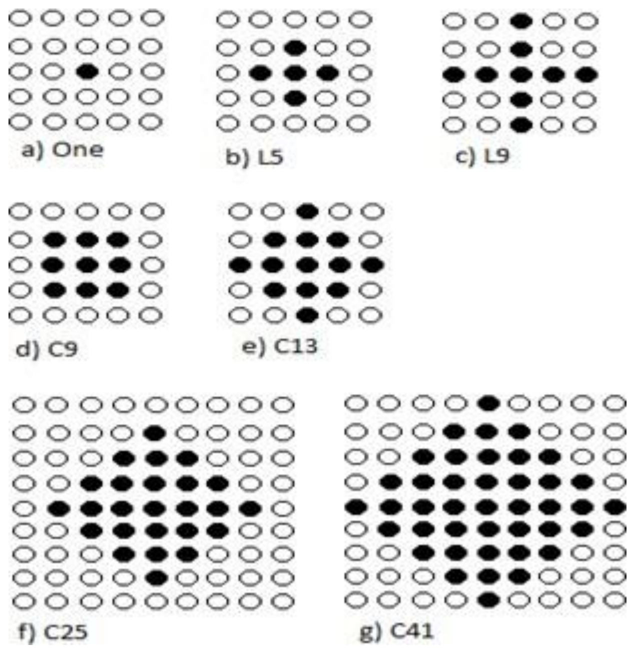

Figure 1. Representation of different neighborhoods

2.3. Benchmark discrete function.

In this section, the preliminary experiments on discrete functions can to verify the effectiveness of the proposed method. The aim of these benchmark function is tested the performance of the discrete optimization algorithms.

The authors [4] suggest that an optimization problem can be defined as a set of potential solutions to the problem and a method for assessing the quality of these solutions. The task is to find a solution from the set of potential solutions that maximizes quality defined by the evaluation procedure. An optimization problem can be defined as follows: In the above function xi = (x_1, x_2, ..., x_n) denotes a vector of discrete random variables domain. For the discrete case, each x_i takes values from 1 to r_i, in other words, the variable xi can take $r_i + 1$ values. For the continuous, case each xi takes values in a range, the variable xi takes, continuously, all possible values in the range. The solution to this problem is to find the maximum (optimal) point of the function f (x) => R. From this time, the candidate solutions will be treated as individuals in the population with which they work.

IsoPeak is a non-separable function and it is composed of the bivariate functions Iso1 and Iso2. The solution for this function is an n-dimensional vector such that n = 2 β m (the variables are divided into groups of two). First, Iso1 and Iso2 auxiliary functions are defined as follows in figure 2.

$\vec{x}$	00	01	10	11
$Iso1$	m	0	0	$m-1$
$Iso2$	0	0	0	m

Figure 2. Iso1 and Iso2 auxiliary functions

Then IsoPeak function is defined:

$$F(\vec{x}\,) = Iso1(x_1, x_2) + \sum_{i=1}^{m} Iso1(x_{2i-1}, x_{2i}) \quad (2.2)$$

The aim is to maximize the IsoPeak function and the global optimum is located at the point (1, 1, 0, 0... 0, 0).

FirstPolytree3 function is a separable function that decomposes meditatively with blocks of length three. In each block is valued function f_3^{Poly} property that Boltzmann distribution with parameter β=2 has a structure Polytree the following arcs: x1β x2 and x3β x2. After the function f_3^{Poly} called as F3Poly can define as:

$$\mathrm{FP3}(\vec{x}) = \sum_{i=1}^{l} f_3^{Poly}(\mathrm{x}_{3i-2}, \mathrm{x}_{3i-1}, \mathrm{x}_{3i})$$

(2.3)

The aim is to maximize F3Poly, taking the global optimum located at the point 0; 0; 1;. . . ; 0; 0; 1) and the function takes the value l*1.047; where: n=3*l.

FirstPolytree5 function is a separable function that decomposes meditatively with five blocks long. In each block is valued function f_5^{Poly} . This function has the property that Boltzmann distribution with parameter β= 2 has a structure Polytree the following arcs:

x1β x3, x2β x3, x3β x5 and x4β x5.

After the function, defined F5Poly can define the function as:

$$\mathrm{FP5}(\vec{x}) = \sum_{i=1}^{l} f_5^{Poly}(\mathrm{x}_{5i-4}, \mathrm{x}_{5i-3}, \mathrm{x}_{5i-2}, \mathrm{x}_{5i-1}, \mathrm{x}_{5i})$$

(2.4)

The aim is to maximize F5Poly, taking the global optimum located at the point (0; 1; 0; 0; 1. . . ; 0; 1; 0; 0; 1) and the function takes the value l*1.723; where n=5*l.

OneMax discrete target, is a function has (n + 1) different values of fitness, which are distributed polynomial where the goal is to maximize the OneMax function.

$$\mathrm{Onemax}(x) = \sum_{i=1}^{n} x_i$$

(2.5)

Where n is the number of variables and x_i is the ith variable in the problem. The global optimum is reached at the point (1,.....,1) and its value is n. Furthermore, a function additively decomposed where the value of the global optimum (1, 1) is equal to the number of variables n.

Deceptive3 functions of order k are defined as the sum of the most basic functions of variables k Deceptive:

$$f_{dec}^{3} = \begin{cases} 0.9 & for\, u = 0 \\ 0.8 & for\, u = 1 \\ 0.0 & for\, u = 2 \\ 1.0 & for\, u = 3 \end{cases} \tag{2.6}$$

Where are strings containing k elements in the Deceptive function of order 3 (Deceptive3) which is defined as follows: The global optimum is reached at point (1, 1..., 1) where n=3*l.

3. EXPERIMENTAL STUDY

To check the correct operation of the algorithm a series of experiments consistent in carrying out executions to different neighborhoods, so that a set of common parameters were chosen, the target population will always be 30% of the global population, the selection method used were made was the path where the best individuals in the population are selected, in all experiments elitism like we used to 1. For all algorithms 100 executions were carried out, the stop criterion is to find the optimal or perform a fixed number of iterations in the case of discrete functions it is 30. For this study were selected the set of test functions before mentioned. It is composed of six problems having many different features.

3.1. Cellular EDA using different neighborhoods

Table 1. shows the experimental results of executing the algorithm for different neighborhoods, taking into account the objective functions: F3Poly, F5Poly and IsoPeak.

Table 1. Experimental results of F3Poly, F5Poly and IsoPeak functions

neighborhood	***Obj. Func.***	***n***	**τ**	***Generations***	***%Success***
One	F3Poly	30	0.3	7.39	29

L5				4	100
C13				5.29	93
C25				6.20	93
C41				9.94	76
C9				4.05	100
L9				4.86	99
One	F5Poly	30	0.3	28.41	13
L5				5.01	98
C13				6.46	90
C25				7.34	94
C41				9.42	79
C9				4.88	100
L9				5.00	100
One	IsoPeak	30	0.3	23.76	43
L5				3.85	100
C13				4.85	96
C25				6.30	93
C41				8.32	82
C9				3.96	100
L9				4.51	100

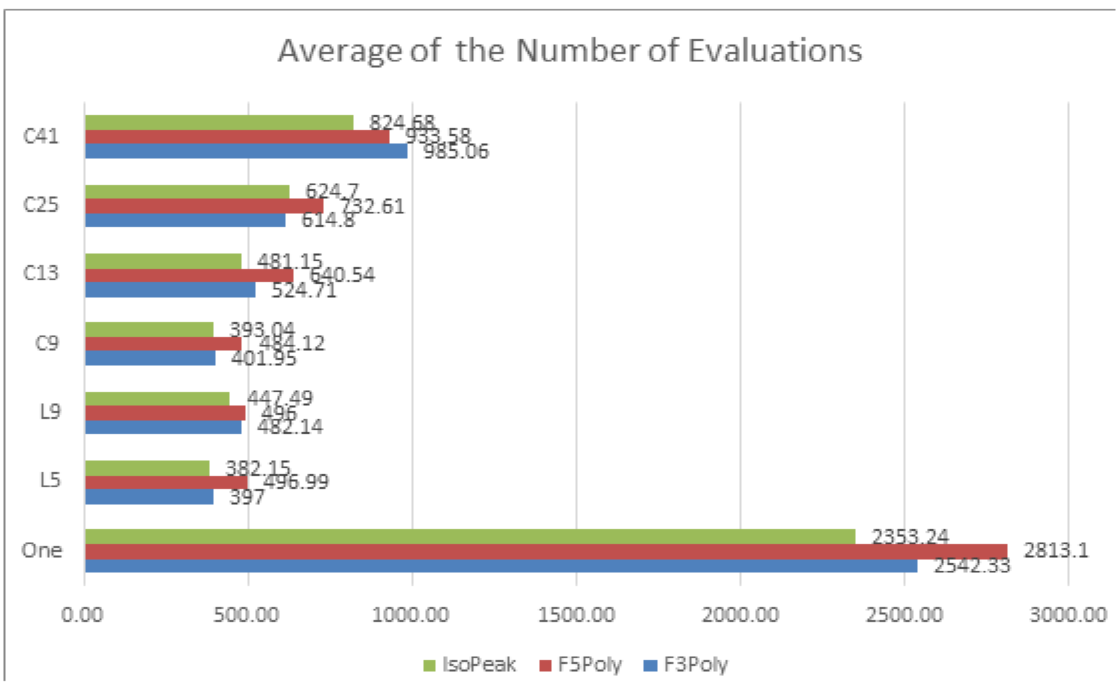

Figure 3. Results of number of evaluations for F3Poly, F5Poly and IsoPeak functions

The cellular EDA algorithm with L5, C9 and L9 neighborhoods requires fewer generations to find optimal of these function, which had the best percentage of success.

As shown in Figure 3, the neighborhoods using the cellular EDA with local learning structures and parameters, learning the Bayesian Networks, the one with the best result is the L5, C9 and L9 that performs less than evaluations for each number of variables. The neighborhood One is the worse efficiency, since for these functions it had a high number of evaluations, to find the optimum. In addition, the neighborhoods C13 and C25 had good evaluative efficiency.

3.2. Cellular EDA using different Grids

Table 2. shows the experimental results of executing the algorithm for different grids, taking into account the OneMax objective function.

Table 2. Experimental results OneMax functions for different grids

neighborhood	***Obj. Func.***	***n***	***Generations***	***%Success***
L5(5x5x2x2)	OneMax	100	7.71	95
L5(10x10x2x2)			2.08	100
L5(20x20x2x2)			2	100
L5(5x5x4x4)			9.04	77
L5(10x10x4x4)			3.12	96
L5(2x2x5x5)			17.24	67
L5(4x4x5x5)			9.27	77
L5(2x2x10x10)			10.75	84
L5(4x4x10x10)			5.83	89
L5(2x2x20x20)			10.61	88

As can be observed, the cellular EDA algorithm with L5 (5x5x2x2) and (10x10x2x2) neighborhoods requires fewer generations to find optimal OneMax function.

As it can be seen in Figure 4, of the configurations of proposed grids the one of smaller number of evaluations for the neighborhood L5, is the configuration 10x10x2x2 and 5x5x2x2, and 2x2x20x20 was worst configurations.

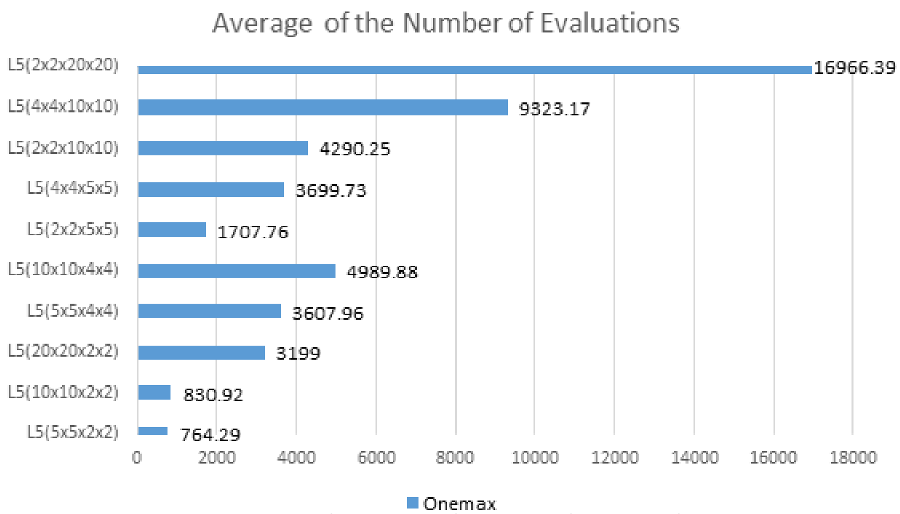

Figure 4. Results of the number of evaluations for Onemax function

3.3. Comparison between cellular EDA and other EDA approach

To study the efficacy of this cellular EDAs, several approaches were selected, including PADA, SPADA and CUMDA. Algorithm with Factorized Distribution based on Polytrees is published with the name of PADA [12, 18], which is named as Polytree Approximation Distribution Algorithm. PADA is an EDA that uses simply connected models as a model of its distributions. On the other hand, SPADA [14], unlike those based on independence tests, requires computing higher-order distributions, as occurs with interconnected networks. However, as in the polytrees it is only possible to insert edges and their cost is lower than the interconnected networks. The local search algorithm using SPADA is a glutton procedure that checks for the existence of non-directed, rather than directed, cycles. Table 3. show the results of the experiments with these approaches.
These results suggest that cellular EDA with neighborhoods L9, C9 and C41; and grid 20x20x2x2 had better performance that EDA approaches (Simple EDA , PADAp3, PADAp2, PADAt3, PADAt2, SPADA π=1, SPADA π=6), taking into account evaluation number and generations.

Table 3. Experimental results of deceptive function (n=30)

Algorithm	*%Success*	*NumEva*	*Generations*
cEDA(L9-20x20x2x2)	100	6397	4
cEDA(C9-20x20x2x2)	100	6397	4
cEDA(C41-20x20x2x2)	100	6412.99	4.01
Simple EDA(2000)[2]	100	10795.60	5.40
PADAp3(1000)[12, 18]	77	18142.84	18.16
PADAp2(1000) [12, 18]	72	17613.37	17.63
PADAt3(1000) [12, 18]	98	10400.59	10.41
PADAt2(1000) [12, 18]	96	10800.19	10.81
SPADA π=1(1000) [14]	92	9071.92	9.08
SPADA π=6(1000) [14]	96	10150.84	10.16

Another study carried out with cellular EDA and the cellular Univariate Marginal Distribution Algorithm (CUMDA)[2], where table 4. shows the result of the cellular EDA using different neighborhood versus CUMDA, taking into account Isopeak function for thirty variables.

Table 4. Experimental results of Isopeak function (n=30)

Algorithm	*%Success*	*Generations*	*NumEva*
cEDA(L9-5x5x2x2)	100	4.51 ± 2.33	447.49 ± 231.00
cEDA(C9-5x5x2x2)	100	3.96 ± 1.92	393.04 ± 190.31
cEDA(C13-5x5x2x2)	96	4.85 ± 5.31	481.15 ± 525.63
cEDA(C25-5x5x2x2)	93	6.30 ± 6.93	624.70 ± 685.96
cEDA(L5-5x5x2x2)	100	3.86 ± 1.20	3.8 ± 1.20
CUMDA(C9-10x10-2x2)[2]	86.66	9.69 ± 2.26	4085 ± 908

These results suggest that cellular EDA with neighborhoods L9, C9, C13, C25 and L5; and grid 5x5x2x2 had better performance that CUMDA with neighborhood C9 and grid 10x10x2x2, despite its grid using minor population. Moreover, the better cellular EDA are cEDA with learning and structure of local population (L5 and C9) with generation (3.86 ± 1.20) and (3.96 ± 1.92); and number of evaluation (382 ± 118.88) and (393.04 ± 190.31) respectively.

4. CONCLUSIONS

In this work, we made the study of the EDA decentralized with local learning structures and parameters, using Bayesian networks for learning based independence tests, which can reduce the number of evaluations in solving discrete optimization problems. Moreover, it was shown that cEDA decreases number of the evaluations of the fitness functions, compared to other EDAs in the literature for solving discrete optimization problems. As future work, adapting cEDA 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 research direction. Another interesting work will be compare cEDA with other algorithms like Differential Evolution Algorithm, Particle Swarm Optimization and Firefly Algorithm in this kind of problems.

RECEIVED: MAY, 2019
REVISED: JULY, 2020

REFERENCES

[1] ALBA, E. and DORRONSORO, B. (2009): **Cellular Genetic Algorithms**, 42 Springer Science & Business Media, Switzerland.

[2] ALBA, E., MADERA, J., DORRONSORO, B., OCHOA, A., and SOTO, M. (2006): Theory and Practice of Cellular UMDA For Discrete Optimization. In **Parallel Problem Solving from Nature-PPSN IX**, 242–251. Springer, Berlin, Heidelberg.

[3] ALBA, E. and TROYA, J. (2000): Cellular Evolutionary Algorithms: Evaluating the Influence of Ratio. In **Parallel Problem Solving from Nature PPSN VI**, 29–38, Berlin, Heidelberg. Springer.

[4] BACK, T., FOGEL, D. B., and MICHALEWICZ, Z. (1997): **Handbook of Evolutionary Computation**. CRC Press, N. York.

[5] DING, Q. and ZHENG, G. (2015): The Cellular Differential Evolution Based On Chaotic Local Search. **Mathematical Problems in Engineering**, 2015, 15.

[6] ETXEBERRIA, R. and LARRANAGA, P. (1999): Global Optimization Using Bayesian Networks. In **Second Symposium on Artificial Intelligence** (CIMAF-99), 332–339. Habana.

[7] HARIK, G. R., LOBO, F. G., and GOLDBERG, D. E. (1999): The Compact Genetic Algorithm. **IEEE Transactions on Evolutionary Computation**, 3, 287–297.

[8] HUANG, A., LI, D., HOU, J., and BI, T. (2015): An Adaptive Cellular Genetic Algorithm Based on Selection Strategy for Test Sheet Generation. **International Journal of Hybrid Information Technology**, 8, 33–42.

[9] MARTÍNEZ-LÓPEZ, Y., MADERA, J., RODRÍGUEZ-GONZÁLEZ, A. Y., and BARIGYE, S. (2019): Cellular Estimation Gaussian Algorithm for Continuous Domain. **Journal of Intelligent & Fuzzy Systems**, 36, 4957–4967.

[10] MARTÍNEZ-LÓPEZ, Y., QUINTANA, J. M., and DE VARONA, I. L. (2016): Algoritmos Evolutivos Con Estimación De Distribución Celulares. **Revista Cubana de Ciencias Informáticas**, 10, 159–170.

[11] MARTÍNEZ-LÓPEZ, Y., RODRÌGUEZ-GONZÀLEZ, A. Y., QUINTANA, J. M., MOYA, A., MORGADO, B., and MAYEDO, M. B. (2019): Cumdancauchy-C1: A Cellular Eda Designed to Solve the Energy Resource Management Problem Under Uncertainty. In **Proceedings of the Genetic and Evolutionary Computation Conference Companion**, 13–14. ACM.

[12] MUHLENBEIN, H., MAHNIG, T., and RODRIGUEZ, A. O. (1999): Schemata, Distributions and Graphical Models in Evolutionary Optimization. **Journal of Heuristics**, 5, 215–247.

[13] MUHLENBEIN, H. and PAASS, G. (1996): From Recombination of Genes to The Estimation of Distributions I. Binary Parameters. **Parallel problem solving from nature—PPSN IV**, 178–187.

[14] OCHOA, A., MU¨HLENBEIN, H., and SOTO, M. (2000): A Factorized Distribution Algorithm Using Single Connected Bayesian Networks. In **International Conference on Parallel Problem Solving from Nature**, 787–796, Berlin, Heidelberg. Springer.

[15] PELIKAN, M., HAUSCHILD, M., and LOBO, F. G. (2012): Introduction to Estimation of Distribution Algorithms. **MEDAL Report**, (2012003).

[16] SANTANA, R. (2003): A Markov Network Based Factorized Distribution Algorithm for Optimization. In **European Conference on Machine Learning**, 337–348. Springer, Berlin.

[17] SHAKYA, S. and MCCALL, J. (2007): Optimization by Estimation of Distribution with Framework Based on Markov Random Fields. **International Journal of Automation and Computing**, 4, 262–272.

[18] SOTO, M., OCHOA, A., ACID, S., and CAMPOS, L. (1999): Bayesian Evolutionary Algorithms Based On Simplified Models. In **Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99),** 360–367. Habana, Cuba.