Journal of Engineering and Technology for Industrial Applications, 2017. Edition. 12.Vol: 03
https://www.itegam-jetia.org
ISSN ONLINE: 2447-0228
DOI: https://dx.doi.org/10.5935/2447-0228.20170072

Cultural algorithm with local search evaluated through non-parametric statistical tests

Carlos A. O. de Freitas[1], Roberto C.L. Oliveira[2], Deam J.A. Silva[3], Jandecy C. Leite[4], Jorge L.M. Rodriguez[5]

[1]Doctorade, PPGEE – Federal University of Pará (UFPA) - Belém - PA - Brazil.
[2]Laboratory of Bio-Inspired Computing - Federal University of Pará (UFPA) - Belém - PA – Brazil.
[3]IEG - Federal University of the West of Pará (UFOPA) - Santarém - PA - Brazil.
[4,5]Institute of Technology and Education Galileo of the Amazon (ITEGAM) - Manaus - AM - Brazil.

Email: carlos.freitas.br@ieee.org

Received: October 11th, 2017

Accepted: November 01th, 2017

Published: December 13th, 2017

ABSTRACT

This work aims to analyze the performance of the classical Cultural Algorithm (CA) with a new hybrid CA proposal with to two local search techniques (Simulated Annealing - SA and Tabu Search - TS). In order to diversify the tests, in the CA with SA there was variation of the parameter energy, and in the CA with TS, there was variation in the size of the tabu list. The algorithms were submitted to two scenarios (scenario 1 - Basic functions, scenario 2 - Hybrid functions). The proposed algorithm differs from others found in the literature, by the process of feeding the topographic knowledge that guides the research. The analysis was performed using the Friedman, Friedman Aligned and Quades tests, which serve to compare the behavior of a set of algorithms at one time.

Keywords: Tabu Search, Cultural Algorithm, Performance Measure, Simulated Annealing

Algoritmo cultural com busca local avaliado através de testes estatísticos não paramétricos

RESUMO

Este trabalho tem como objetivo analisar o desempenho do clássico algoritmo cultural (CA) com uma nova proposta CA hibrido com duas técnicas de pesquisa locais (Simulated Annealing - SA e Busca Tabu - BT). Para diversificar os testes, no AC com SA houve variação da energia do parâmetro, e no AC com BT, houve variação no tamanho da lista de tabu. Os algoritmos foram submetidos a dois cenários (cenário 1 - Funções básicas, cenário 2 - Funções híbridas). O algoritmo proposto difere de outros encontrados na literatura, pelo processo de alimentação do conhecimento topográfico que orienta a pesquisa. A análise foi realizada utilizando os testes Friedman, Friedman Aligned e Quades, que servem para comparar o comportamento de um conjunto de algoritmos de uma só vez.

Palavras Chaves: Busca Tabu, Algoritmo Cultural, Medida de Desempenho, Simulated Annealing

I INTRODUTION

For some time now, science has sought to model the natural evolution of living beings in computational systems [1, 2]. From the engineering point of view, these models will be seen as the basis for the development of meta-heuristics to solve problems, basically in systems optimization [3-5]. Research progress has shown that meta-heuristics with different operating mechanisms may be more suitable for problems with certain structures, and other meta-heuristics may work better in other classes of problems [6]. This led research to the development of new metaheuristics that were based on other processes of the nature other than the evolution of the species. Due to the new approaches, in which we notice the occurrence of the increase in the knowledge of the mechanisms that support evolutionary computation algorithms, it is noticed that the new Evolutionary Algorithms (EAs) are moving away from the strict biological inspiration. The new EAs tend to deepen the tendency to incorporate operations and mechanisms that are not bio-inspired, but rather inspired by mathematical or computational arguments [6]. We also have algorithms inspired by the adaptation and cultural evolution of individuals in a community, called Cultural Algorithms (CAs). These generate or alter their knowledge due to the relationship between individuals in the community. The Cultural Algorithms (CAs or CA) were proposed by [7]. Due to its characteristics of implicit parallelism and random search the CAs are used in the solution of traditional problems from complex optimization [8]. Techniques employing meta-heuristics (GAs, CAs, etc.) as global search heuristics and local search (hill climbing, TS, SA, etc.), are commonly referred to as memetic algorithms (MAs) [9], or hybrid algorithms. Normally, AMs may not only have a good exploratory capacity, similar to what a population-based global search algorithm does, but it also provides a good intensification performance during the search, similar to what a local search algorithm does. The hybridization of the CAs with the local search engine for extensive exploration in the solutions generated by the CAs can greatly improve the performance of this hybrid over the algorithms in their classical form. Comparisons of AEs generally tend to the analysis of their results after several executions of these in the attempt to solve several functions of benchmarks. These evaluations are performed using statistical hypotheses [10, 11].

It can be said that these are results-based analyzes [12], that is, it is an evaluation of the performance of the algorithm for certain benchmark functions. However, the difficulty is in comparing several algorithms, since this comparison is usually performed in pairs of algorithms [13] and increases with the number of algorithms to be evaluated, in addition to increasing the probability of making an error [14]. The interest in nonparametric statistical analysis has recently grown in the field of computational intelligence [10], since it can be a way of comparing evolutionary algorithms, tested for several different problems, with some statistical significance. In this proposal the hybridization of the CAs with two forms of local search (TS and SA) are compared with each other and with the classical CAs. These three algorithms (pure CAs, CAs with TS and CAs with SA) are used to find the minimum of eight real variable benchmark functions. The results are evaluated based on non-parametric tests: Friedman, Friedman Aligned and Quade.

The proposed article provides in section 2 a basic content of items taken as a basis for its development. In section 3 we present the materials and methods with the test scenarios used in this work. In section 4, the results of the simulations are presented. Finally, section 5 concludes the paper with observations and comments on the simulations performed.

II META-HEURISTICS FOR OPTIMIZATION

In this section the global search meta-heuristics 'Cultural Algorithm' and the two meta-heuristics used for local search 'Simulated Annealing' and 'Tabu Search' will be presented. The combination of a global search algorithm with a local search algorithm compose the basis of Memetic Computing [15].

II.1 CULTURAL ALGORITHMS

CAs are used to model the evolution of the cultural component in a computational evolutionary system over time, since it accumulates experience in solving a set of data in problem solving [8]. Cultural evolution allows societies to involve or adapt their environment at rates that exceed biological evolution, which is based only on genetic inheritance [7].

The CAs are formed basically of a population space, a space of beliefs, communication protocols (Acceptance and Influence Functions) between the two spaces and some auxiliary functions: Initialization, Selection, Update and Evaluation. The structure of the CAs is shown in Figure 1, their pseudocode is shown in Figure 2.

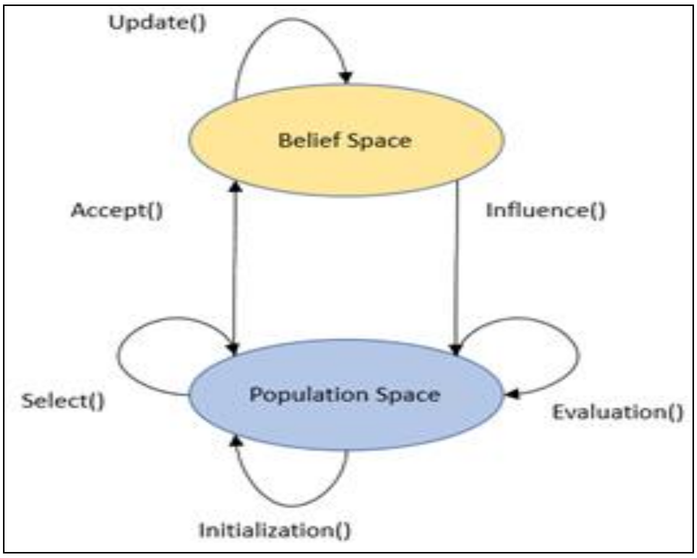
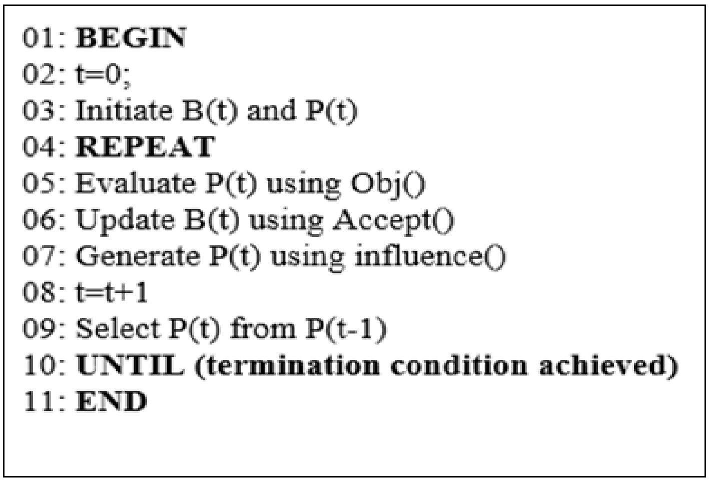


Figure 1: Structure of the Cultural Algorithm.
Source: Adapted from [16].



```
BEGIN
t=0;
Initiate B(t) and P(t)
REPEAT
Evaluate P(t) using Obj()
Update B(t) using Accept()
Generate P(t) using influence()
t=t+1
Select P(t) from P(t-1)
UNTIL (termination condition achieved)
END
```

Figure 2: Pseudocode of Cultural Algorithm.
Source: Adapted from [16].

The spaces mentioned are described below:

Population Space: set of solutions that can be modeled using any technique that makes use of a population of individuals;

The Space of Beliefs (Group Map): is the place where occurs the storage and representation of knowledge (experience or individual maps) acquired throughout the evolutionary process takes place. The sources of knowledge are five according [16], these are useful in decision making [16, 17]. For example: Situational knowledge has successful and unsuccessful solutions, etc.; Normative knowledge contains ranges of acceptable behaviors. Topographical knowledge has spatial patterns of behavior.

Population space and belief space are linked by a communication mechanism (protocol) composed of an acceptance function that is used to collect the experience of individuals from the selected population. The other function of the communication protocol is the influence function that can make use of the knowledge of solutions of problems in the space of belief to guide the evolution of individuals in the population space. CAs can explore both microevolution and macro evolution. Microevolution refers to the evolution that happens at the population level and macroevolution is that which occurs on the culture itself, that is, the evolution of the belief space [18].

II.2 SIMULATED ANNEALING

Simulated Annealing (SA) is a metaheuristic inspired by the physical process of annealing a solid to obtain low energy states in the area of condensed matter physics [6]. The SA establishes a connection between this type of thermodynamic behavior and the search for global minimums for a discrete optimization problem.

In the same way that the solid is slowly cooled to ensure a crystal structure, the algorithm cools the solution slowly to ensure that it has the best objective function, while allowing configurations to match the best value of the objective function found (situation correspondence the small heating) [18].

The acceptance of configurations that have higher temperature, for [18] this is an important feature of SA, which may seem worse, that is, it allows the acceptance of a configuration that provides a "worse" value for the objective function, thus avoiding convergence to a minimum. This acceptance is determined by a random number being controlled by expression (1):

$$P = e^{-\Delta/T} \qquad (1)$$

II.3 TABU SEARCH

The tabu search (TS) guides the heuristic procedure of local search by using characteristics of the current solution and the search history to explore the solution space. [19], in several cases, the methods described provide solutions that are very close to the optimal solution and are among the most effective, if not the best, solutions to the difficult problems in question. As a local search technique, TS starts from an initial solution and moves in the solutions space from one solution to another that is in its neighborhood [6].

The systematic use of adaptive memory is the property that distinguishes TS from other metaheuristics. The word "adaptive" means that the memory actualizes the storage of elements of solutions or complete solutions found during the exploration of solutions spaces [19].

The process of intensification is improved by the use of memory structures, called tabu lists. Each iteration is checked if the current solution has been visited previously or if some rule has been violated, if this solution is stored in the tabu list and marked "tabu". This procedure avoids the so-called cycling, that is, that a solution is visited again. With this memory strategy, the TS algorithm can go beyond the optimal location and access other regions of the solution space [6]. This strategy is based on the fact that in the exploration of the solution space, the oldest solutions are possibly "distant" from the region of the space under analysis and, as such, have no influence in the choice of the next solution in that region [6]. The size of the tabu list is considered a critical parameter. For according to [6], the size of the list cannot be so small, under penalty of cycling; nor so large, to unnecessarily store solutions that are not tied to the recent history of the search.

II.4 BEHAVIOR TESTS OF ALGORITHMS: FRIEDMAN, FRIEDMAN ALIGNED AND QUADE

The need to define the behavior of algorithms when submitted to problems of different natures, has opened a field of research in procedures of tests [12, 20]. The Friedman test is a multiple comparison test that aims to detect significant differences between the behavior of two or more algorithms [10]. The procedure for carrying out the Friedman test follows the following steps, according to [10]:

1. Gather all results from each algorithm / problem pair;
2. Classify the values of each problem i from 1 (best result) to k (worst result). Note this classification as r_j^i (1≤ j ≥k);
3. For each algorithm j, calculate the average of the classifications obtained in all problems to obtain the final classification $R_j = \frac{1}{n}\sum_i r_i^j$.

In this way the algorithms are classified for each problem separately. As indicated in item 2, the algorithm with the best performance is classified with 1, the second best with 2, etc.

The Friedman statistic is calculated according to equation 2.

$$F_f = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \qquad (2)$$

For the aligned Friedman test, a location value is calculated as the average performance achieved by all algorithms in each problem. The step of obtaining the difference between the performance of an algorithm and the location value is repeated for each combination of algorithms and problems. Equation 3, shows the definition for the statistical calculation of the aligned classification of Friedman.

$$F_{AR} = \frac{(k-1)\left[\sum_{j=1}^{k}\hat{R}_j^2 - (kn^2/4)(kn+1)^2\right]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k)\sum_{i=1}^{n}\hat{R}_i^2} \qquad (3)$$

The Quade test is the third test used in this work. This test differs from that of Friedman who considers equality in terms of importance among algorithms, takes into account the fact that some problems are more difficult or that the differences recorded in the sequence of various algorithms on them are larger. Therefore, the calculated rankings in each problem can be sized depending on the observed differences in the performance of the algorithms, obtaining, as a result, a weighted classification analysis of the sample of [10].

The Quade test can be calculated by equation 4, taking into account some definitions presented in [10]. Considerando também os termos A e B, dados pelas equações 5 e 6, respectivamente.

$$F_Q = \frac{(n-1)B}{A-B}, \qquad (4)$$

Considering also the terms A and B, given by equations 5 and 6, respectively:

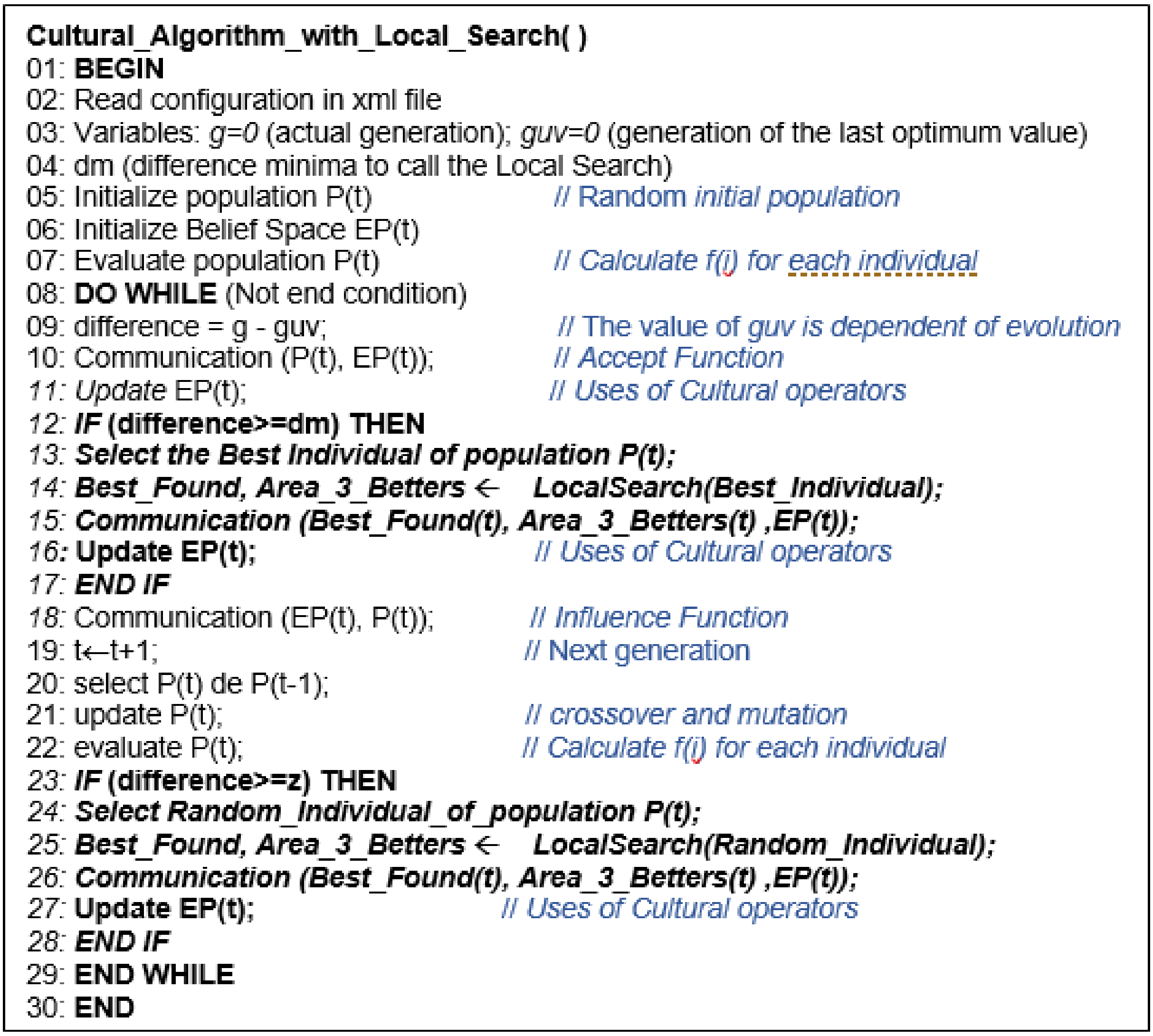
$$A = n(n+1)(2n+1)k(k+1)(k-1)/72 \quad , \qquad (5)$$

$$B = \frac{1}{n}\sum_{j=1} kS_j^2 \quad , \qquad (6)$$

II.5 ALGORITHMS USED

The CAs have in their population space the population evolved through the GAs. Within the mutation function, the local search techniques are applied (SA by varying local search energy at 5, 10 and 15. TS with tabu list size variations at 2, 4 and 6).

This differs from [17] which uses the concept of "ball" [18] as a technique to define neighborhood, where an area of radius 'r' is defined, where it should contain the possible solutions. The use of information from the 3 best individuals found in the local search is used to define an area of good behavior that feeds the topographic knowledge. This approach until then had not been used, according to the bibliographical research carried out, this shows the relevance of the research. Figure 3 shows the pseudocode of the algorithm used. Since, the function Cultural_Algorithm_with_Local_Search() is used to represent the use of SA or TS.



```
Cultural_Algorithm_with_Local_Search( )
BEGIN
  Read configuration in xml file
  Variables: g=0 (actual generation); guv=0 (generation of the last optimum value)
  dm (difference minima to call the Local Search)
  Initialize population P(t)                    // Random initial population
  Initialize Belief Space EP(t)
  Evaluate population P(t)                      // Calculate f(i) for each individual
  DO WHILE (Not end condition)
    difference = g - guv;                       // The value of guv is dependent of evolution
    Communication (P(t), EP(t));                // Accept Function
    Update EP(t);                               // Uses of Cultural operators
    IF (difference>=dm) THEN
      Select the Best Individual of population P(t);
      Best_Found, Area_3_Betters ←   LocalSearch(Best_Individual);
      Communication (Best_Found(t), Area_3_Betters(t) ,EP(t));
      Update EP(t);                             // Uses of Cultural operators
    END IF
    Communication (EP(t), P(t));                // Influence Function
    t←t+1;                                      // Next generation
    select P(t) de P(t-1);
    update P(t);                                // crossover and mutation
    evaluate P(t);                              // Calculate f(i) for each individual
    IF (difference>=z) THEN
      Select Random_Individual_of_population P(t);
      Best_Found, Area_3_Betters ←   LocalSearch(Random_Individual);
      Communication (Best_Found(t), Area_3_Betters(t) ,EP(t));
      Update EP(t);                             // Uses of Cultural operators
    END IF
  END WHILE
END
```

Figure 3: Pseudocode of the proposed Algorithm..
Source: Author,(2017).

III. MATERIALS AND METHODS

In this work, some simulation scenarios were created with the objective of measuring the performance of the CAs and their adaptations with local search. The modified parameters for each simulation were: Population size (Size Pop), Number of generations (Num Ger) and Number of repetitions (Num Rep). The maximum number of evaluations that is the product of Size Pop by Num Ger, was maintained in the value of 10,000 and 30,000 evaluations per repetition for functions depending on the scenario being used.

III.1 BENCHMARK FUNCTIONS

The benchmarks functions are widely used when you want to measure algorithm performance. These functions are used to compare the Cultural Algorithms (CAs) with two hybridization proposals, CAs with Simulated Annealing (SA) and CAs with Tabu Search (TS).

The three algorithms were tested in eight CEC2017 benchmark functions [21]. The search space definitions $[-100.0, 100.0]^D$ and the dimensions D = 10 and D = 30 are applied in four basic functions as shown in table 1 and in four hybrid functions as shown in table 2:

Table 1: Basic Benchmark Functions.

Function	Description and Expression	
F1	Bent Cigar Function	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$,
F3	Zakharov Function	$f_3 = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$,
F5	Rastrigin's Function	$f_5(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$
F15	Griewank's Function	$f_{15}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$,

Source: Adapted from [21].

Table 2:Hybrid Benchmark Functions

Function	Dom. Basic Functions	Description	Basic Functions
FH1	P = [0.2, 0.4, 0.4]	Hybrid Function 1	g1: Zakharov Function g2: Rosenbrock Function g3: Rastringin's Function
FH4	P = [0.2, 0.2, 0.2, 0.4]	Hybrid Function 4	g_1: High Conditioned Elliptic Function g_2: Ackley's Function g_3: Schaffer's Function g_4: Rastrigin's Function
FH5	P = [0.2, 0.2, 0.3, 0.3]	Hybrid Function 5	g_1: Bent Cigar Function g_2: HGBat Function g_3: Rastrigin's Function g_4: Rosenbrock Function
FH8	P = [0.2, 0.2, 0.2, 0.2, 0.2]	Hybrid Function 8	g_1: High Conditioned Elliptic Function g_2: Ackley's Function g_3: Rastringin's Function g_4: HGBat Function g_3: Discus Function

Source: Adapted from [21].

III.2 SCENARIO 1 AND 2

The two scenarios have Number of Repetitions equal to 50. Each new repetition generates a new random population with uniform probability density function defined within the limits of each variable. Tables 3 presents the parameters of scenarios 1 and 2, related to basic and hybrid functions respectively.

Table 3: Scenario 1 and 2 - Basic and Hybrid Functions

D	Scenario1	Function Name	Scenario2	Function Name	Size Pop	Num. Ger	Max. Evaluation	D*Max_FES	Num Rep
10	**F1**	**Bent Cigar**	**FH1**	**Hybrid 1**	50	200	10.000	100.000	50
10	**F3**	**Zakharov**	**FH4**	**Hybrid 4**	50	200	10.000	100.000	50
10	**F5**	**Rastrigin's**	**FH5**	**Hybrid 5**	50	200	10.000	100.000	50
10	**F15**	**Griewank's**	**FH8**	**Hybrid 8**	50	200	10.000	100.000	50
30	**F1**	**Bent Cigar**	**FH1**	**Hybrid 1**	50	600	30.000	300.000	50
30	**F3**	**Zakharov**	**FH4**	**Hybrid 4**	50	600	30.000	300.000	50
30	**F5**	**Rastrigin's**	**FH5**	**Hybrid 5**	50	600	30.000	300.000	50
30	**F15**	**Griewank's**	**FH8**	**Hybrid 8**	50	600	30.000	300.000	50

Source: Author, 2017.

III.3 FLOWCHART FOR THE TEST

Figure 4 shows through a flowchart the steps used to run the tests for this article.

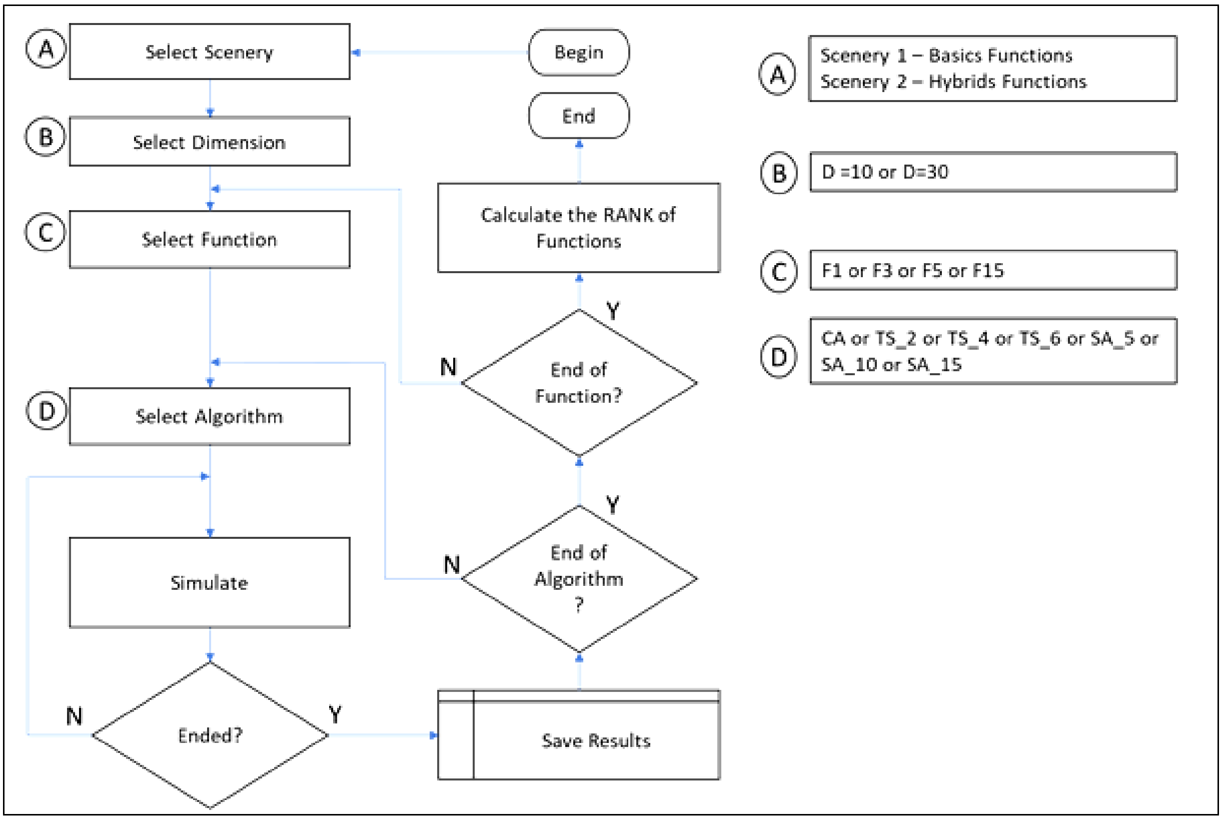


Figure2: Flowchart for the test.
Source: Author, 2017.

IV RESULTS

For all the CAs used, the same pattern of their parameters was maintained, and preliminary tests were carried out with the intention to define the size of the SA variable and the tabu list size of the TS, and after these tests it was defined that three variables of each technique that were shown with more differentiated results would be used to increment new test scenarios. In the variation of the CAs with the SA, the Local Search Energy variable changed from 5, 10 and 15. In the variation of the CAs with the TS, the tabu list variation varied in 2, 4 and 6. In this way, three new test conditions emerge for this hybrid algorithm. Getting, then with seven variations of algorithms. Table 4 presents each of the alternatives.

Table 4: Algorithms for testing

Algorithm	Description
CA	Classical Cultural Algorithm
SA_5	Cultural Algorithm + *Simulated Annealing* (Local Search Energy = 5)
SA_10	Cultural Algorithm + *Simulated Annealing* (Local Search Energy = 10)
SA_15	Cultural Algorithm + *Simulated Annealing* (Local Search Energy = 15)
TS_2	Cultural Algorithm + Tabu Search (Tabu List Size = 2)
TS_4	Cultural Algorithm + Tabu Search (Tabu List Size = 4)
TS_6	Cultural Algorithm + Tabu Search (Tabu List Size = 6)

Source: Author, 2017.

For evaluation, the results of the best values of each simulation were submitted to the tests of Friedman, Friedman Aligned and Quade. The tests were performed with the results of D = 10, D = 30 and with the combination of these two results. The inputs of data for analysis are in table 5.

Table 5: Data entry of D-10 and D-30.

Data-set	CA	TS2	TS4	TS6	SA5	SA10	SA15
10-F1	1,25E-13	1,05E-13	3,90E-14	7,60E-14	1,04E-13	2,50E-14	2,90E-14
10-F3	0	0	0	0	0	0	0
10-F5	1,66E-12	1,74E-12	4,38E-12	4,54E-12	4,46E-11	9,12E-12	1,00E-11
10-F15	0,004932	0,009857	7,87E-12	1,23E-10	0,004932	7,44E-13	0,007396
10-FH1	0,053469	0,066624	0,041932	0,000210	0,126893	0,001537	0,142696
10-FH4	1,069561	1,069561	1,069561	1,069561	1,069561	1,069561	1,069561
10-FH5	0,524180	0,300677	0,383617	0,480464	0,358352	0,588176	0,396050
10-FH8	1,569561	1,569561	1,569561	1,569561	1,569561	1,569561	1,569561
30-F1	6,27E-08	7,25E-08	4,71E-08	1,11E-07	3,65E-08	1,41E-08	3,14E-08
30-F3	0	0	0	0	0	0	0
30-F5	0,000221	1,46E-05	4,83E-05	0,000447	6,07E-05	0,000495	0,002175
30-F15	8,38E-06	4,22E-06	9,90E-06	9,62E-06	5,92E-06	7,03E-06	9,23E-06
30-FH1	0,053469	0,066624	0,041932	0,000210	0,126893	0,001537	0,142696
30-FH4	1,069561	1,069561	1,069561	1,069561	1,069561	1,069561	1,069561
30-FH5	0,524180	0,300677	0,383617	0,480464	0,358352	0,588176	0,396050
30-FH8	1,569561	1,569561	1,569561	1,569561	1,569561	1,569561	1,569561

Source: Author, 2017.

The data of table 5 were submitted to evaluation of the tests treated in this work. The results are shown in Table 6, 7 and 8. The values in red are the best results in each test associated to the algorithm that had the best performance for the set of functions.

Table 6: Results for D-10.

Algorithm	Friedman	Friedman Aligned	Quade
CA	3,375	22,875	3,277778
TS2	4,3125	27,3125	4,416667
TS4	4,25	37,375	4,694444
TS6	4,1875	30,4375	4,472222
SA5	3,6875	23,0625	3,666667
SA10	4,9375	33,4375	4,666667
SA15	3,25	25	2,805556

Source: Author, (2017).

Table 7: Results for D-30.

Algorithm	Friedman	Friedman Aligned	Quade
CA	3,375	23,75	3,25
TS2	5,1875	31,8125	5,430556
TS4	3,625	33,375	4,083333
TS6	3,3125	25,0625	3,513889
SA5	4,6875	33,8125	4,708333
SA10	4,5625	29,3125	4,125
SA15	3,25	22,375	2,888889

Source: Author,(2017).

Table 6 shows the results obtained when only the data set with dimension 10 was applied. In this case we had SA15 as the best-ranked algorithm in the Friedman and Quade tests. In the Friedman Aligned test, the AC obtained a better classification. Table 7 shows the resulting values of the tests for the dataset with dimension 30 in each problem. In this scenario SA15 was the best in all tests.

Table 8: Result for the junction of D-10 and D-30

Algorithm	Friedman	Friedman Aligned	Quade
CA	3,375	45,4375	3,220588
TS2	4,75	58,0625	4,886029
TS4	3,9375	70,5	4,389706
TS6	3,75	54,875	3,974265
SA5	4,1875	57,25	4,238971
SA10	4,75	62,375	4,430147
SA15	3,25	47	2,860294

Source: Author, (2017).

After an individual evaluation of the proposed algorithms with the appropriate problems, it was decided to join the generated databases with dimensions 10 and 30, to verify if the classification of these three tests differs greatly from the previous results. Table 8 shows the result after the submission of these data to the tests discussed here. For the result of the union of the obtained values it was observed that accompanied the simulation of the dimension 10.

V CONCLUSIONS

The use of the Friedman, Friedman Aligned and Quade tests helped to evaluate the variations of the proposed algorithms, avoiding the comparison of pairs of algorithms. The data set of D = 10 and D = 10 + 30 (union of test results of algorithms with domain 10 with results of tests with domain equal to 30), obtained equal results for the best Algorithms (SA15 and AC) for the functions used. However, with the data of D = 30, the result was a single one pointing SA15 as the best classified in the simulations. In relation to the CA with Tabu Search, the best result was found with the tabu list of size 6.

The positional values of the 3 best values found in local searches were added to the CA belief space, specifically in topographic knowledge, thus creating a region of promising results for CA evaluation. In addition, it is noted that the hybridization of CA with local search techniques tends to obtain better results in the solution of multivariate functions. The tests used for this evaluation are easy to implement and robust in results when little is known about the problem.

VI ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support to this research done by UFPA, ITJRSC and ITEGAM.

VII BIBLIOGRAPHIC REFERENCES

[1] D. Bray and S. Lay, "**Computer simulated evolution of a network of cell-signaling molecules**," Biophysical journal, vol. 66, pp. 972-977, 1994.

[2] C. Adami, "**The use of information theory in evolutionary biology**," Annals of the New York Academy of Sciences, vol. 1256, pp. 49-65, 2012.

[3] K. De Jong, D. Fogel, and H.-P. Schwefel, "**Handbook of Evolutionary Computation,**" IOP Publishing Ltd and Oxford University Press, 1997.

[4] A. E. Eiben and J. E. Smith, **Introduction to evolutionary computing** vol. 53: Springer, 2003.

[5] A. Petrowski and S. Ben-Hamida, **Evolutionary Algorithms**: John Wiley & Sons, 2017.

[6] A. Gaspar-Cunha, R. Takahashi, and C. H. Antunes, **Manual de computação evolutiva e metaheurística:** Imprensa da Universidade de Coimbra/Coimbra University Press, 2012.

[7] R. G. Reynolds, "**An introduction to cultural algorithms,**" in Proceedings of the third annual conference on evolutionary programming, 1994.

[8] Z. Wei, B. Yan-ping, and Z. Ye-qing, "**The application of an improved cultural algorithm in grid computing**," in Control and Decision Conference (CCDC), 2013 25th Chinese, 2013, pp. 4565-4570.

[9] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "**Hybrid metaheuristics in combinatorial optimization: A survey**," Applied Soft Computing, vol. 11, pp. 4135-4151, 2011.

[10] J. Derrac, S. García, D. Molina, and F. Herrera, "**A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms**," Swarm and Evolutionary Computation, vol. 1, pp. 3-18, 2011.

[11] M. Friedman, "**A comparison of alternative tests of significance for the problem of m rankings,**" The Annals of Mathematical Statistics, vol. 11, pp. 86-92, 1940.

[12] J. Derrac, S. García, S. Hui, P. N. Suganthan, and F. Herrera, "**Analyzing convergence performance of evolutionary algorithms:** A statistical approach," Information Sciences, vol. 289, pp. 41-58, 2014.

[13] J. Demšar, "**Statistical comparisons of classifiers over multiple data sets,**" Journal of Machine learning research, vol. 7, pp. 1-30, 2006.

[14] R. A. Krohling, R. Lourenzutti, and M. Campos, "**Ranking and comparing evolutionary algorithms with Hellinger-TOPSIS,"** Applied Soft Computing, vol. 37, pp. 217-226, 2015.

[15] P. Moscato and C. Cotta, "**A modern introduction to memetic algorithms**," in Handbook of metaheuristics, ed: Springer, 2010, pp. 141-183.

[16] R. G. Reynolds and Y. A. Gawasmeh, "**Evolving heterogeneous social fabrics for the solution of real valued optimization problems using cultural algorithms**," in Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012, pp. 1-8.

[17] M. Z. Ali, N. Awad, and R. G. Reynolds, "Balancing search direction in cultural algorithm for enhanced global numerical optimization," in Swarm Intelligence (SIS), 2014 IEEE Symposium on, 2014, pp. 1-7.

[18] D. J. A. Silva, "**Algoritmos Culturais com Abordagem Memética e Multipopulacional Aplicados a Problemas de Otimização**," Tese - Doutorado em Engenharia Elétrica - Aplicações Computacionais, Instituto de Tecnologia, Universidade Federal do Pará, Belém-PA, 2012.

[19] M. Gendreau and J.-Y. Potvin, "**Tabu search," in Search methodologies**, ed: Springer, 2014, pp. 243-263.

[20] S. García, D. Molina, M. Lozano, and F. Herrera, "**A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour:** a case study on the CEC'2005 special session on real parameter optimization," Journal of Heuristics, vol. 15, pp. 617-644, 2009.

[21] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, "**Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization,**" 2016.