ESSIM-EA applied to Wildfire Prediction using Heterogeneous Configuration for Evolutionary Parameters

Miguel Méndez-Garabetti[1,2], Germán Bianchini[1], Paola Caymes-Scutari[1,2], María Laura Tardivo[1,2,3], and Verónica Gil-Costa[2,4]

[1] Laboratorio de Investigación en Cómputo Paralelo/Distribuido, (LICPaD) UTN-FRM, Mendoza, Argentina.
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).
[3] Departamento de Computación, Facultad de Ciencias Exactas, Físico-Químicas y Naturales (UNRC), Córdoba, Argentina.
[4] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) Universidad Nacional de San Luis - Ej. de los Andes 950 - San Luis, Argentina.
mmendez@mendoza-conicet.gob.ar,gbianchini@frm.utn.edu.ar, pcaymesscutari@frm.utn.edu.ar,lauratardivo@dc.exa.unrc.edu.ar,gvcosta@unsl.edu.ar

Abstract. Wildfires devastate thousands forests acres every year around the world. Fire behavior prediction is a useful tool to cooperate in the coordination, mitigation and management of available resources to fight against this type of contingencies. However, the prediction of this phenomenon is usually a difficult task due to the uncertainty in the prediction process. Therefore, several methods of uncertainty reduction have been developed, such as the Evolutionary Statistical System with Island Models based on Evolutionary Algorithms (ESSIM-EA). ESSIM-EA focuses its operation on an Evolutionary Parallel Algorithm based on islands, in which the same configuration of evolutionary parameters is used. In this work we present an extension of the ESSIM-EA that allows each island to select an independent configuration of evolutionary parameters. The heterogeneous configuration proposed, at the island level, with the original methodology in three cases of controlled fires has been contrasted. The results show that the proposed ESSIM-EA extension allows to improve the quality of prediction and to reduce processing times.

Keywords: wildfire prediction, HPC, uncertainty reduction, metaheuristics.

1 Introduction

Wildfires are considered the natural phenomenon that causes most damages and losses worldwide. As a recent fact we can cite the great fire that occurred in the provinces of La Pampa, Río Negro and Buenos Aires between December 2016 and January 2017, where fire consumed more than 2.5 million acres [1]. This kind of situations often brought about when there are long periods of drought, low humidity, high temperatures and considerable winds. In this context it can be very useful to have tools or methods that minimize the negative effects caused by wildfires. Such is the case of the widlfire behavior prediction, where we intend to determine with some time in advance the future behavior of the fire. It allows to identify the areas that are most at risk of being caught by fire, and therefore, to plan the efficient use of available resources. The behavior of a wildfire depends on several variables or factors, which are not usually known at the time of the fire.

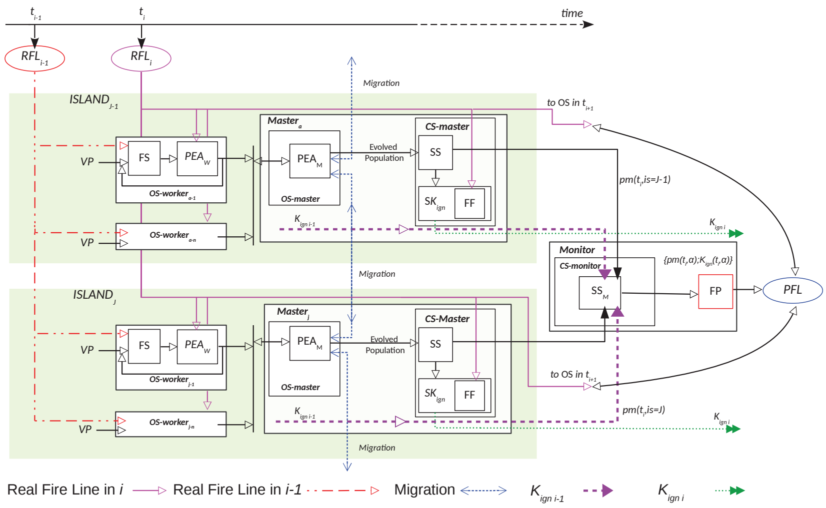

Fig. 1: ESSIM-EA: **FS**: Fire Simulator; **PEA**: Parallel Evolutionary Algorithm; **PEA**$_F$: PEA (Fitness Evaluation); **OS**: Optimization Stage; **SS**: Statistical Stage; **SK**$_{ign}$: search K_{ign}; **K**$_{ign}$: Key Ignition Value; **FF**: Fitness Function; **CS**: Calibration Stage; **FP**: Fire Prediction; **PFL**: Predicted Fire Line; **RFL**$_x$: Real Fire Line on time x; **PV**: input Parameters Vector; **SS**$_M$: Statistical Stage in monitor process; pm: probability map.

Due to this, the use of classical prediction tools does not usually allow to obtain good quality results, being necessary to resort to uncertainty reduction methods that allow to minimize the lack of precision in the system's input. By classical prediction we mean the use of a single instance of the model or simulator which feed each parameter with direct values, whether measured or estimated, but not including pre-processing or calibration data. The Evolutionary Statistical System with Island Model based on Evolutionary Algorithms (ESSIM-EA) [2], is a general uncertainty reduction method that has been successfully applied in wildfire behavior prediction. ESSIM-EA uses Parallel Evolutionary Algorithms (PEAs) under a multi-population parallelization scheme based on islands [3]. In [2] ESSIM-EA obtained improvements in the quality of prediction in contrast to previously developed methodologies. Subsequently, in [4], ESSIM-EA underwent two complementary studies of static tuning [5] of the evolutionary parameters involved. In the tuning work [4], it was possible to increase the quality of prediction for some instants of the evaluated fires, without being able to determine a single optimum parameters configuration for all the fires. It is important to mention that the quality of prediction, for a particular fire, is calculated in discrete instants of time called prediction steps. Based on these results, this work presents an extension of the ESSIM-EA method where the islands use a heterogeneous configuration of the evolutionary parameters, allowing to obtain, for each prediction step, the result generated by the best parameter configuration. In this work this version is called ESSIM-EAh to differentiate it from the homogenous configuration version.

The Section 2 describes the ESSIM-EA method operation, together with the proposed heterogeneous configuration of evolutionary parameters. Then the details of the experiments are given, the working environment is described, and the results obtained are presented below in Section 3. Finally, in Section 4, the conclusions and the future work are presented.

2 ESSIM-EA: method description

ESSIM-EA has been successfully applied as an uncertainty reduction method in wildfire behavior prediction. As mentioned above, ESSIM-EA uses Parallel Evolutionary Algorithms under an island-based parallelization scheme as the optimization technique with dual master-worker hierarchy. Once the method is running, three different types of processes are executed: a) monitor process, b) master process and c) process worker. The arrangement and communication relationship between these processes can be seen in Fig. 1. The method operation begins in the monitor process, it is responsible for sending the initial information to each island. This initial communication includes: a) the Real Fire Line (RFL_x), b) the Value Ranges (VR) for each input parameter, and c) the PEA Parameter Setting values (PS_{PEA}). It is important to mention that the monitor process will have in memory an array with known parameter configurations, which will be used by the different islands (PS_{PEA}). Since each island can use a different evolutionary parameter configuration, these values (PS_{PEA}) are selected by the monitor and sent to the masters of each island (this is explained in more detail in section: 2.1).

Once each master has received the initial information, it can start the parallel evolutionary algorithm (PEA_M). Each master initiates a different population and applies the evolutionary operators using the parameters configuration previously received from the monitor process. The evaluation of individuals is carried out by workers (OS_{Worker}), which simulate their behavior in the Fire Simulation stage (FS).

Table 1: Parameter settings and ranking. (a) FT: fitness threshold, NIt: number of iterations, NI: number of individuals, NMI: number of migrated individuals, MF: migration frequency, CP: crossing probability and PM: probability of mutation. (b) Ranking of parameter configurations.

(a)

Configuration	FT	NIt	NI	NMI	MF	CP	PM
No. 0	**0,7**	**200**	**200**	**5**	**5**	**60**	**0,5**
No. 1	0,78	373	68	3	2	44	0,7
No. 2	**0,75**	**419**	**194**	**17**	**4**	**37**	**1,7**
No. 3	**0,74**	**198**	**175**	**3**	**7**	**19**	**1,1**
No. 4	**0,71**	**302**	**41**	**11**	**10**	**42**	**0,3**
No. 5	**0,81**	**172**	**25**	**9**	**3**	**22**	**2**
No. 6	0,56	359	125	4	9	27	2,1
No. 7	**0,68**	**129**	**145**	**20**	**7**	**24**	**2,2**
No. 8	0,69	117	91	5	7	58	1,7
No. 9	**0,59**	**429**	**187**	**7**	**5**	**30**	**0,4**
No. 10	0,57	144	51	6	4	28	0,6
No. 11	**0,65**	**249**	**97**	**19**	**8**	**10**	**0,9**
No. 12	**0,73**	**454**	**39**	**9**	**12**	**40**	**0,35**
No. 13	**0,84**	**465**	**110**	**12**	**3**	**13**	**1,5**
No. 14	**0,62**	**263**	**15**	**18**	**6**	**53**	**1,4**
No. 15	0,64	328	62	16	3	17	1,3
No. 16	**0,8**	**293**	**123**	**13**	**9**	**33**	**0,5**
No. 17	**0,67**	**486**	**162**	**15**	**6**	**56**	**1,1**
No. 18	**0,82**	**234**	**139**	**9**	**5**	**48**	**2,4**
No. 19	0,77	398	37	8	9	39	1,8
No. 20	**0,161**	**304**	**84**	**14**	**1**	**45**	**2,4**

(b)

Rank	Case A	Case B	Avg.
1°	No. 0	No. 0	No. 0
2°	No. 5	No. 9	No. 9
3°	No. 18	No. 20	No. 18
4°	No. 3	No. 18	No. 3
5°	No. 14	No. 3	No. 14
6°	No. 13	No. 11	No. 7
7°	No. 7	No. 17	No. 4
8°	No. 9	No. 7	No. 20
9°	No. 4	No. 2	No. 12
10°	No. 16	No. 14	No. 2
11°	No. 12	No. 4	No. 5

In the simulation of each individual, the actual fire front of the fire at t_{i-1} (RFL$_{i-1}$), together with the values of each individual (or vector parameters, VP) in the simulator, is evaluated. At the end of the simulation, the difference between the simulated map and the actual map for that subsequent time instant t_i (RFL$_i$) is contrasted. The result of this comparison is the fitness value of each individual, which is used by the PEA. Subsequently, the results of the individuals evaluated are sent to their respective master, where the PEA stage is in charge of storing the partial results, evolve the population and migrate the individuals to the neighboring islands. This procedure is repeated until reaching the threshold of fitness or the maximum number of iterations, both previously established. At this time the calibration step begins to operate, i.e., the evolved population is sent to the Statistical Stage (SS) of the master process Calibration Stage (CS$_{Master}$). SS is responsible for generating a probability map, which is created by considering the simulated maps of each individual. The probability map generated by each island in t_i is used to calculate a pattern of fire behavior, called Key Ignition Value (K_{ign}). This value is calculated in the sub-step search for K_{ign} (SK$_{ign}$). It is important to note that each island generates in t_i a *key ignition value* and a *probability map*, which are sent to the monitor process (CS$_{Monitor}$). The monitor statistical stage (SS$_{Monitor}$) evaluates the pairs of data received by all islands by selecting those that offer the best fitness value for each prediction step. These data pairs are sent to the Fire Prediction stage (FP) to generate the Predicted Fire Line (PFL).

2.1 Heterogeneous Configuration of Evolutionary Parameters

From the point of view of configuration of evolutionary parameters, an island-based PEA can be implemented using a homogeneous or heterogeneous configuration. Homogeneous configuration is that all islands use the same configuration of evolutionary parameters, i.e. the same values of number of individuals, probability of crossing and mutation, etc., executing exactly the same algorithm with the exception of the populations initialized independently. In contrast, the heterogeneous configuration uses different parameter configurations on each island, whether obtained by known or calibrated configurations, even using random values within preset ranges. The work developed in [6], shows that combining multiple search threads with heterogeneous parameter configurations increases the robustness of the search engine, allowing to explore the space of solutions more efficiently and facilitating the obtaining of better results.

The PEA's capacity depends on certain parameters, such as population size, selection criteria, total number of generations, frequency of migration, among others. Determining the best values for each parameter is usually not an easy task, since there is no universal configuration of parameters and these are usually related to the problem's characteristics. Therefore, in order to obtain a suitable parameter configuration, a calibration or tuning study of the parameters [5] is necessary. Based on this, to determine the parameter configurations to be used, the results obtained in [4] were taken as a starting point, where a static calibration study was carried out on a group of evolutionary parameters of ESSIM-EA. This calibration consisted of evaluating 20 parameter configurations (from No. 1 to No. 20 in Table 1.a) generated through a Latin Hypercube Design [7] (LHD), along with an additional experiment obtained on the basis of a classical parameter configuration (configuration No. 0 in Table 1.a). Each parameter configuration was evaluated in two test cases corresponding to controlled fires. The metric used to perform the ranking was the quality of prediction for each simulation step, the average of each of them and the total execution time. The results allowed to determine those configurations that offered good quality of prediction values, without being able to determine a single configuration, since the results varied by each step of prediction. In order to use the best configurations (in terms of quality of prediction), a ranking was used to reduce the number of LHD configurations and to select those that offered better results.

The ranking was elaborated starting from the results obtained in [4] for each configuration of parameters of the Table 1.a. The classification of the parameter configurations was done for the two case studies (columns Case A and Case B in Table 1.b) and for the average of both (column Avg. in Table 1.b). The generated ranking can be seen in the Table 1.b. The values in the Case A, Case B, and Avg. columns indicate the parameter setting number of the Table 1.a. Therefore, of the 21 initial configurations, we were able to reduce, in a ranking of 11 positions, to a total of 15 final configurations, which have been highlighted in bold in Table 1.a). It is important to note that the ranking configurations of the Table 1.b are those that ESSIM-EAh maintains at the moment of starting the execution of the method to be used by each of the islands. The selection of each

Table 2: Description of the case studies: dimension (m), slope (degrees), initial time, increment and final time (min).

Case	Width	Height	Slope	Initial Time	Increment	Final Time
I	89	109	21	2	2	14
II	89	91	21	2.5	2.5	12.5
III	60	90	6	2	2	10

configuration is done randomly in the master process of each island (previously all of the configurations are sent from the monitor process), this selection is carried out at runtime.

3 Experimentation and Results

The performance of the methodology has been evaluated by two metrics that allow us to evaluate important aspects of the behavior of the method: *(a)* the quality of prediction and *(b)* the total time consumed. The quality of prediction is a value calculated using a fitness function. This function compares the real fire map with the predicted map (simulated behavior), determining a numerical value that indicates the accuracy of the prediction. This value, called fitness value indicates the degree of accuracy of this prediction. These values are within the range {0,1} where 1 represents a matching prediction with reality and 0 represents a totally wrong prediction. The second metric is to evaluate the total time elapsed for each experimen, this time will depend on the island with the most expensive parameter configuration. At the moment it has not been evaluated how to reduce the idle time in which each island waits for the completion of the rest of the islands, this is part of the future work.

3.1 Experiments

For the experiments, three cases of controlled burns were used in Serra de Louçã, (Gestosa, Portugal).

For each experiment, discrete time intervals have been defined which represent the progress of the fire front where the quality assessment is carried out. Since the terrains have different dimensions, the number of time steps is not the same in all experiments. The characteristics of each of them (i.e., dimension, slope, initial time and duration) are described in Table 2. Due to the nondeterministic behavior of the method, 20 executions of each experiment were performed using the same set of seeds. For each metric the average of them was considered. It is important to note that both ESSIM-EA and ESSIM-EAh cannot perform predictions for the first instant of time (e.g. Experiment I, minute 4), since the calibration step is performed at that time.

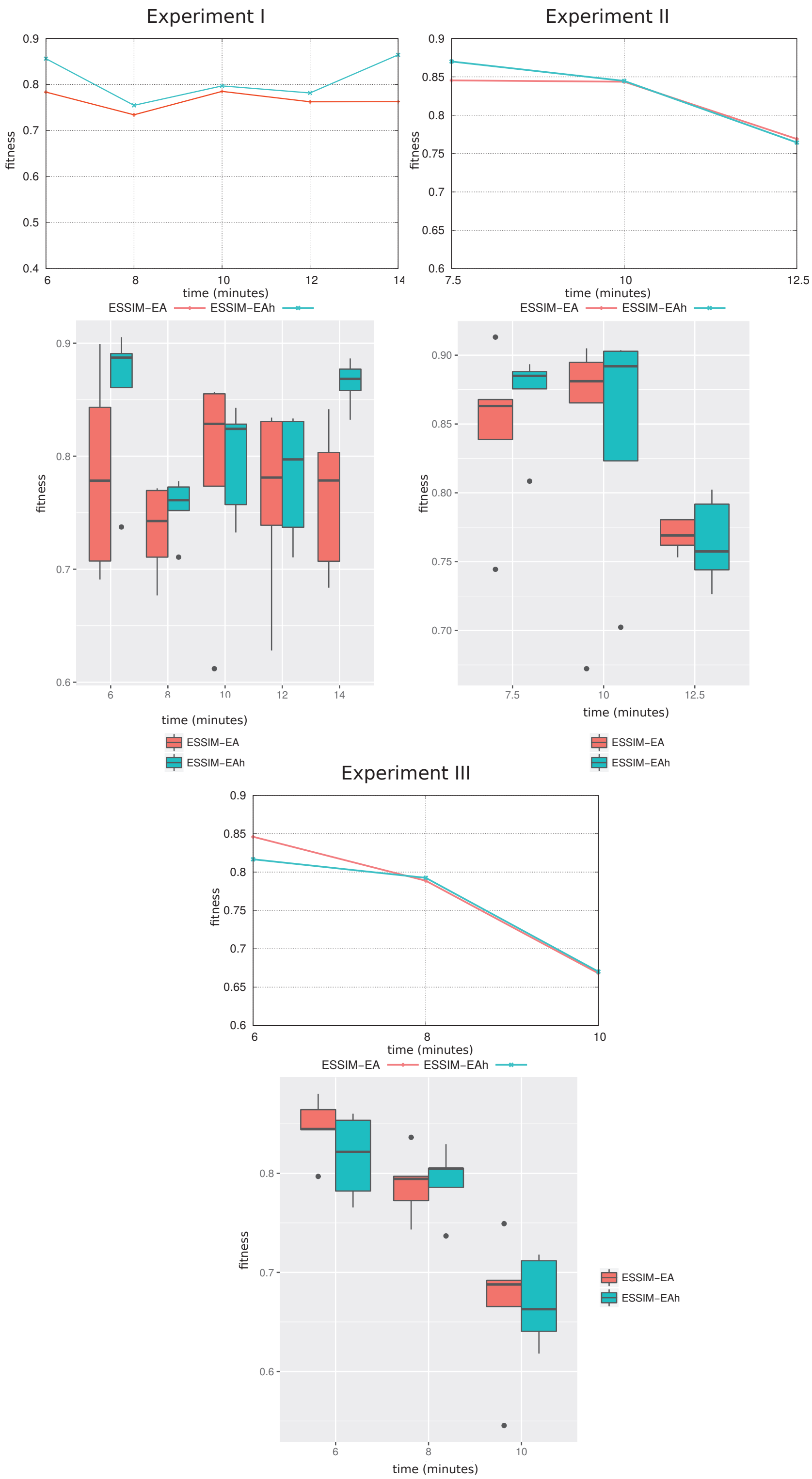

Fig. 2: Experiments I, II and III: Comparison between ESSIM-EA and ESSIM-EAh. Above: Comparison of quality of prediction. Below: Boxplots.

3.2 Results

The results obtained, both in terms of prediction quality and in processing time, have been compared with those obtained by ESSIM-EA using a homogeneous parameter configuration. The first one refers to the original implementation of ESSIM-EA based on a homogeneous configuration of evolutionary parameters, and ESSIM-EAh to the heterogeneous configuration version. It is important to note that the parameter configuration used by ESSIM-EA in these experiments corresponds to the configuration No. 0 of the Table 1.a, while in ESSIM-EAh each island operates using some of the 15 configurations generated by the ranking. In Fig. 2 the quality of prediction has been plotted for the N prediction steps of each experiment. This quality of prediction has been generated from the average of 20 executions. Also, in the lower part of the same figure (Fig. 2), we can see the distribution of these results, using boxplots diagrams. The experiments were carried out in a Linux cluster with 32 processing units (Intel-Q9550 Quad Core 2.83 GHz processors), 4GB RAM, Gigabit Ethernet network under an MPI environment.

Experiment I: this experiment (see Fig. 2), shows that the heterogeneous implementation exceeds the quality of prediction obtained by ESSIM-EA throughout the experiment. The greatest difference is observed in the first and last instants of time, with an average fitness value greater than 0.85. If we analyze the results considering the box diagrams (see Fig. 2), we can observe that in the time instants for which ESSIM-EAh exceeds ESSIM-EA (minutes 6 and 14), the distribution of results obtained by ESSIM-EAh has a smaller deviation. It can also be seen that in the 1 step (minute 6), about 99% of the results is above 0.85. And in the last step (minute 14) just over 75% is on the same value. In the step 2 (minute 8) a similar behavior can be observed, since although the difference in average is smaller, ESSIM-EAh has a minimum deviation concentrating near the 99% of the results in the range: 0.75 and 0.78. Finally, in Table 3 we can observe the amount of time consumed by each method for each experiment. In this table, the executions that offer the best performance have been highlighted in bold, where ESSIM-EA obtains a final difference of 24 less minutes compared to ESSIM-EAh. In this experiment, this difference is mainly related to the value assigned to the parameter that determines the maximum number of iterations (NIt, in Table 1.a).

Experiment II: in this experiment the largest difference is obtained in favor of ESSIM-EAh in the first prediction step (7.5 minute) with an average value equal to 0.8700898 (Fig. 2). However in the subsequent steps, minute 10 and 12.5, the differences are negligible of the order of 10^{-3} being best ESSIM-EAh. Observing the boxplot, we can see that the performance of ESSIM-EAh exceeds ESSIM-EA. This is so because, at minute 7.5, the 99% of the results are above 0.87, except for an outlier with a value equal to 0.808473. In the rest of the prediction steps (minutes 7.5 and 10), although the quality of prediction average is similar in both implementations, it is observed that the homogeneous configuration offers a more limited distribution of results, indicating a better performance

Table 3: Average execution time.

Experiment	ESSIM-EA	ESSIM-EAh
I	**00:57:20**	00:81:30
II	01:37:54	**00:38:17**
III	01:04:13	**00:31:40**

of ESSIM-EA. The best results obtained correspond to those generated by the parameter configurations number 13, 4, 3 and 7 of Table 1.a.

If we evaluate the amount of total time consumed (Table 3) we can observe that in this experiment, ESSIM-EAh obtains a very significant gain regarding ESSIM-EA, with a difference of 59 minutes. The reduction of time obtained can be related to a greater speed of convergence of the evolutionary algorithm. This could be facilitated by the use of parameter configurations that allow the method to find good solutions with less effort. If this is the case, the maximum amount of iterations would not matter since the fitness threshold would be reached in a smaller number of iterations. To be able to determine this behavior with certainty, we should make a detailed analysis of the evolution of the different populations in each one of the islands.

Experiment III: this has a total area of 5400m^2. Although it has an equivalent amount of prediction steps to the previous experiment, it has a propagation velocity of the upper fire front, so the total simulation time is 8 minutes. When evaluating the results in terms of the quality of prediction average (Fig. 2) we can observe a similar behavior to the previous experiment, i.e., a significant difference in the first prediction step, in favor of ESSIM-EA, and minimal differences in the following steps. Looking at the box diagram we can see that the performance of ESSIM-EA is more stable than ESSIM-EAh as it concentrates about 99% of results in ranges of the order of 10^{-1}. Finally, the amount of time consumed (Table 3), shows a quite similar behavior to that of the Experiment II. Although here the gain is considerably lower, ESSIM-EAh reduces resource utilization by more than 50%. While the parameter configurations that offer the best results are 13, 9, 0 and 3 from the Table 1.a.

4 Conclusions

In this paper, a new implementation of the Evolutionary Statistical System with Islands Model based on Evolutionary Algorithms (ESSIM-EA) uncertainty reduction method has been presented. The original methodology of the method consists in using an evolutionary algorithm parallelized following an island-based scheme, operating with a unique configuration of evolutionary parameters. This work present the design and evaluation of a new implementation of this method, which consists in equipping ESSIM-EA with the ability to operate each island with different parameter configurations, i.e. with a heterogeneous configuration of evolutionary parameters. The implementation was based on results obtained in previous works, which consisted of studies of static tuning of the evolutionary parameters. In these works it was possible to determine a series of configurations that allowed to improve the capacity of the method at certain instants of each fire. In this way, the new heterogeneous implementation uses these *known configurations*, and allows each island to select one of them in order to improve the performance of the method both in quality of prediction and in terms of resource utilization. The experimentation was carried out in three cases of controlled fires, evaluating the quality of prediction and the time consumed in each execution. The comparison of results was performed between the heterogeneous implementation (ESSIM-EAh) and the original methodology (ESSIM-EA). The results show that the heterogeneous configuration achieves improvements in both prediction quality and processing time. Although the increase in prediction quality is observed to a greater extent in one of the three experiments, and in the rest this increase is minimal, the new implementation manages to improve the prediction quality in 81% of the predicted steps. In addition to significantly reducing processing time in two of the three cases. As future work will evaluate alternatives to improve the use of resources (avoiding idle resources), will also implement this concept of "heterogeneous configuration of parameters" in a method that uses multiple metaheuristics.

References

1. Infobae, "Los incendios afectan más de un millón de hectáreas en La Pampa, Río Negro y Buenos Aires", http://www.infobae.com/sociedad/2017/01/05/sergio-bergman-los-incendios-afectan-mas-de-un-millon-de-hectareas/. 2017.
2. M. Méndez-Garabetti, G. Bianchini, P. Caymes-Scutari, M. L. Tardivo, "Increase in the quality of the prediction of a computational wildfire behavior method through the improvement of the internal metaheuristic", Fire Safety Journal, Volume 82, ISSN 0379-7112, pp. 49-62, May. 2016.
3. H. Mühlenbein, Parallel genetic algorithms, population genetics and combinatorial optimization, pages 398–406. Springer Berlin Heidelberg, 1991. doi:10.1007/3-540-55027-5_23.
4. M. Méndez Garabetti, G. Bianchini, P. Caymes Scutari, M. L. Tardivo, ESS-IM applied to Forest Fire Spread Prediction: Parameters Tuning for a Heterogeneous Configuration. 2016 35th International Conference of the Chilean Computer Science Society (SCCC), ISBN: 978-1-5090-3338-6 (USB), pp. 514-525
5. E. Talbi, Metaheuristics: From Design to Implementation, John Wiley and Sons, 2009.
6. E. Alba, F. Luna, A.J. Nebro, and J.M. Troya. Parallel heterogeneous genetic algorithms for continuous optimization. Parallel Computing, 30(5- 6):699–719, 2004.
7. R. L. Iman y W.J. Conover, "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment," Communications in Statistics, Part A. Theory and Methods, Vol. 17, 1980, pp. 1749-1842.
8. R. Real, & J.M. Vargas. (1996). The Probabilistic Basis of Jaccard's Index of Similarity. Systematic Biology, 45(3), 380–385. http://doi.org/10.1093/sysbio/45.3.380