Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 2018–2027

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

A comparative study of evolutionary statistical methods for uncertainty reduction in forest fire propagation prediction

María Laura Tardivo[1,2,3], Paola Caymes-Scutari[1,2], Germán Bianchini[1], Miguel Méndez-Garabetti[1,2], Andrés Cencerrado[4], and Ana Cortés[4]

[1] LICPaD. Departamento de Ingeniería en Sistemas de Información, Facultad Regional Mendoza. Universidad Tecnológica Nacional. Argentina.
{pcaymesscutari;gbianchini}@frm.utn.edu.ar

[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
mmendez@mendoza-conicet.gob.ar

[3] Departamento de Computación. Universidad Nacional de Río Cuarto, Argentina.
lauratardivo@dc.exa.unrc.edu.ar

[4] HPCA4SE research group, Universitat Autònoma de Barcelona, Spain
acencerrado@caos.uab.es, ana.cortes@uab.es

Abstract

Predicting the propagation of forest fires is a crucial point to mitigate their effects. Therefore, several computational tools or simulators have been developed to predict the fire propagation. Such tools consider the scenario (topography, vegetation types, fire front situation), and the particular conditions where the fire is evolving (vegetation conditions, meteorological conditions) to predict the fire propagation. However, these parameters are usually difficult to measure or estimate precisely, and there is a high degree of uncertainty in many of them. This uncertainty provokes a certain lack of accuracy in the predictions with the consequent risks. So, it is necessary to apply methods to reduce the uncertainty in the input parameters. This work presents a comparison of ESSIM-EA and ESSIM-DE: two methods to reduce the uncertainty in the input parameters. These methods combine Evolutionary Algorithms, Parallelism and Statistical Analysis to improve the propagation prediction.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Forest Fire Prediction, Statistical analysis, Evolutionary Algorithms, Islands model, High Performance Computing

1 Introduction

Forest fires are a critical problem worldwide. Although these fires are a necessary element to maintain the balance of the ecosystems, they can become a major threat to the population

1877-0509
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.252

and the biodiversity under certain environmental conditions, such as high temperatures and drought. In January 2017 eleven people have died in the blazes of the Chile's worst forest fire in recent history, which have devastated more than a million acres.

Either natural or anthropogenic, the effects of large forest fires can be catastrophic and have the power to devastate thousands of hectares of vegetation and to kill many people and animals year by year. Fires also alter the availability of natural resources, with the consequent impact on the daily lives of surrounding communities.

In the process of fire-fighting, the actions taken by the security forces are complemented with tools, systems and/or strategies such as fire simulators, firewalls, evacuation strategies, technology for fire suppression, etc. These strategies are used at different stages: prevention, detection, monitoring and forecasting of fires. The forecasting stage is an essential task to allow useful and efficient fire management. Fire forecast tries to predict the spread of a wildland fire already started, so as to know in advance the areas that present greater risk of being affected, making it possible to take early decisions. Predicting the spread of wildfire consists in determining which will be the direction and rate of fire spread, the shape of the fire front, the flame intensity, etc. To achieve a good prediction, it is necessary to use as much information as possible about the fire front and the environment in which the fire occurs.

Currently, there is a strong interest among the scientific community in this kind of natural phenomenon, and determining their behavior represents a great challenge. The main goal is to provide models to explain and predict the spread of fire on the ground. They can be used to develop simulators and decision-making tools. However, the models must be fed by all the input parameters used to represent the various dynamic factors that determine the fire behavior. Unfortunately, it is very unlikely to know exactly the value of all these factors. This lack of accuracy or uncertainty in the value used by the model negatively affects the quality of the prediction. The development of computational methods for uncertainty reduction can improve the knowledge about the input parameter values, in order to perform more realistic predictions.

In this work, two methods to improve the wildland fire propagation prediction are described. These methods are framed within the Data Driven Methods with multiple overlapped solutions, and are based on the use of Evolutionary Strategies, Parallelism and Statistical Analysis. This paper focuses on a comparative analysis of the two methods in terms of quality of prediction.

This work is organized as follows. Section 2 describes two types of prediction methods. Section 3 includes a brief description of the evolutionary strategies used by the methods under comparison, gives some detail of their parallel design and describes the methods ESSIM-EA and ESSIM-DE. Section 4 presents the experimentation and the obtained results. Finally, section 5 summarizes the main conclusions and future work.

2 Prediction Methods

In general terms, the Classical Prediction method for wildland fire behavior consists in evaluating the position of the fire after a certain initial period of time, using any existing fire simulator. A general scheme of this kind of methodology can be observed in the left upper part of Figure 1 (green rectangle). As can be seen, the simulator (**FS**) is fed by two sets of data: the real fire line of the wildfire at time t_n (RFL_n), generally represented by a map that shows the burned area where the fire started, and the information that describes the environment on which the fire spreads, such as vegetation, terrain description and weather data (this set of parameters are the input parameters). One scenario is represented by a particular set of parameter values. This set of values, along with RFL_n, is used by **FS** to make the prediction of the fire line (**PFL**) for the next time instant (t_{n+1}) corresponding to that particular scenario. So, each scenario includes a

value for each one of the input parameters (wind speed, wind direction, temperature, humidity and moisture contents of dead and live fuel). Examples of classical prediction in wildfires are [2] and [7]. Although this methodology has been widely used as classical prediction method, it provides generally predictions that are far from reality due to input parameter uncertainty.

With the focus on the uncertainty problem, we are geared toward 2-stage or Data-Driven Methods. In these methods, the fire evolution is observed during a time interval and after that the input parameters are calibrated to determine the values of the parameters that best reproduce the observed behavior. The obtained values for the parameters are then used to determine the future evolution of the fire. So, there is a parameter calibration stage and a prediction stage. In this methods the parameter uncertainty is reduced by calibrating the values of the parameters according to the observed behavior. This scheme is represented in Figure 1.

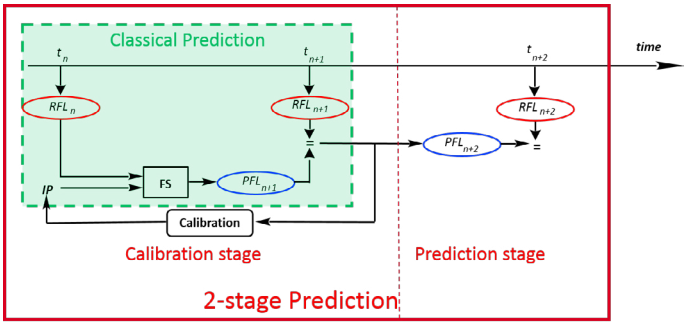

Figure 1: Classical and 2-stage Prediction methods (**FS**: Fire Simulator;**IP**: Input Parameters; $\mathbf{RFL}_n$: Real Fire Line on time n; $\mathbf{PFL}_n$: Predicted Fire Line on time n)

In this data-driven methods, two main approaches can be distinguished. On the first one the prediction is based on a single set of parameters and is the result of one particular simulation. Different approaches and algorithms, mainly evolutionary algorithms, has been used to determine the set of parameters [1], [6], [5]. On the second one, the calibration and the prediction consider a statistical combination of several sets of parameters to estimate the fire propagation. These methods are called Data-Driven Methods with Multiple Overlapped Solutions (DDM-MOS). The methods compared in this work are framed within this approach.

Considering the calibration stage it is necessary to define a fitness function that measures the similarity between the real propagation and the propagation predicted by each scenario. In this work, the fitness function considered is based on the Jaccard-Index [12] and is described in Equation 1, in which A represents the set of cells in the real map without the subset of burned cells before starting the simulations, and B represents the set of cells in the simulation map without the subset of burned cells before starting the simulation. So, the fitness value will be between 0 and 1. A value equal to 1 represents a perfect prediction, and a fitness value equal to 0 would indicate the maximum error.

$$Fitness = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

2.1 Data-Driven Methods with Multiple Overlapped Solutions (DDM-MOS)

In these methods, a certain number of scenarios is considered. For each scenario, the fire propagation prediction from t_n to t_{n+1} is evaluated. All the maps obtained by the simulation of each scenario are aggregated to determine the trend behavior so that a value is associated to each cell. This value represents the probability of each cell to be burned (P_{ign}, defined as n_A/n where n is the total number of scenarios, and n_A is the number of scenarios in which the cell A was burned). The output is the matrix including the ignition probability of each cell. This matrix considers the contribution of each scenario, taking into account all the total number of scenarios.

The set of cells whose P_{ign} value is higher than or equal to a certain particular value P_K constitutes the probability map with probability P_K. An example of the aggregation process and the probability maps is shown in Figure 2 (a).

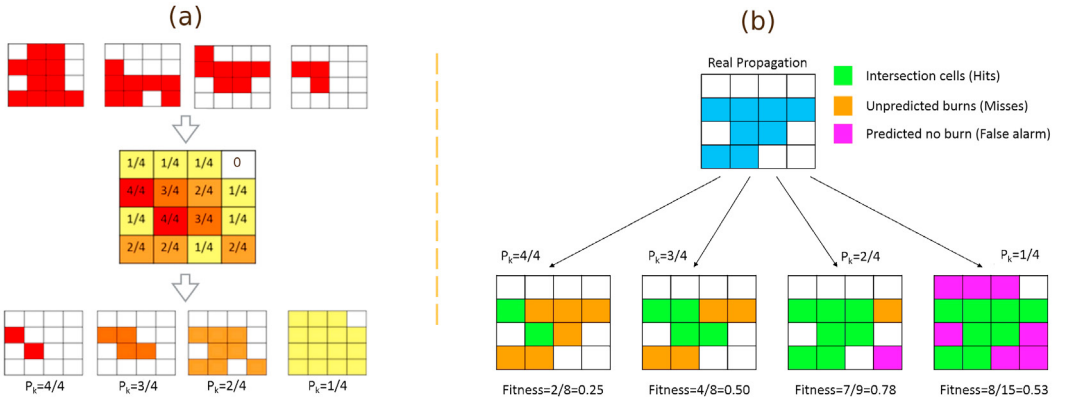

Figure 2: (a) Matrix aggregation and Probability maps. (b) Fitness function evaluation for each probability map.

The next step consists in comparing the real fire propagation against each probability map. The objective of this comparison is to search for a particular value of P_{ign}, whose associated probability map provides the best matching with the real fire propagation. In other words, we are interested in finding what we refer to as a Key Ignition number (K_{ign}) to determine the fire behavior pattern. The value found, $\mathbf{K}_{ign}$, is used to make the prediction for the next time instant (t_{i+1}). To better illustrate the K_{ign} value searching process, a graphical representation is depicted in Fig. 2 (b). In this example, the real burned map is made up of 4x4 cells. It is shown on the upper part of the figure. The four matrices in the lower part represent the fitness corresponding to each probability map P_K. In these matrices, the predicted and burned cells (hits) are shown in green, the burned cells that were not predicted (misses) in orange and the predicted but not burned (false alarms) in purple. The fitness values reached considering $P_k = 4/4$, $P_k = 3/4$, $P_k = 2/4$ and $P_k = 1/4$ are, respectively, 0.25, 0.50, 0.78 and 0.53. So, the best matching with the real burned map is the probability map having $P_K = 2/4$, with a fitness value of 0.78 and, therefore, the K_{ign} value is 2.

To predict the fire propagation for the next time interval from t_{n+1} to t_{n+2}, the propagation prediction for each scenario is calculated, the aggregation map is generated and the prediction is given by the probability map with $P_k = 2/4$. So, this approach can be exploited by using

multiple scenarios, where for each parameter a range of values and a resolution step is specified. These results in thousands of simulation runs to get reliable probability maps, which introduces an important overhead. To overcome these difficulties, a combined Evolutionary-Statistical approach can be defined. The following section presents a description of this strategy.

3 Evolutionary Computing and Parallel Design: ESSIM

In the Evolutionary-Statistical approach the statistical analysis is carried out on a lesser amount of scenarios, which are the result reached by an evolutionary search algorithm. So, a population with a certain amount of individuals (each individual is an scenario) is generated and evolved for a certain number of iterations. The evolved population is used to carry out the statistical analysis. The main idea of using the evolutionary approach is to guide a random population towards a certain search space region with "good" individuals and then apply the statistical method over this reduced set of scenarios. In the next, we describe the main elements of evolutionary-statistical methods.

3.1 Evolutionary Computing

The methods compared in this work use as search algorithms two different evolutionary strategies: Evolutionary Algorithms and Differential Evolution.

Evolutionary Algorithms (EAs) are considered efficient search methods that can be applied to optimization problems. In general, the method consists of a series of iterations, called generations, in which a group of possible solutions, called individuals, conforms a population. Each individual is evaluated (i.e., run) to obtain a result that allows us to assess its suitability as a solution for the problem. Each individual has a particular performance or fitness associated with the optimization problem considered; this performance is calculated using the function that mathematically describes the problem to be optimized. Then, the population evolves through generations simulating principles of natural selection and survival of individuals, by applying a set of operations such as selection, mutation, crossover and replacement.

The **Differential Evolution (DE)** algorithm is a population-based stochastic optimizer proposed by Price and Storn in 1995 [13]. It starts to explore the search space by generating a population of individuals. An individual is defined as a D-dimensional vector, whose initial values are randomly obtained based on the limits defined by the user and according to the nature of the problem. The main idea of the method is to use vectors difference in order to modify the population vector. After initialization, DE performs in sequence three vector operations over each of the population individuals: mutation, crossover and selection. The *mutation* operation applies vector differences between the existing population members, for determining both the degree and direction of the new individuals produced. In the mutation operation a parameter F named *scaling factor* is involved, and controls the difference amplification between involved individuals, and it is also used to avoid stagnation in the search process. The mutant vector obtained after mutation phase and the current population individual are involved in the *crossover operation*, generating a new vector denominated “trial vector". The constant $Cr \in [0, 1]$, denominated *crossover factor*, is a parameter of the algorithm defined by the user. Cr is used to control the values fraction that are copied from the mutant vector. The *selection* phase determines which individuals will be part of the next generation. The objective function of each trial vector is evaluated and compared with the objective function value for its counterpart in the current population. If the trial vector has less or equal objective function target value (for minimization problems) it will replace the current vector in the next generation

population. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation.

3.2 Parallel Design

An important feature of any prediction tool is the provision of short-term responses. This goal can be achieved by using high performance computing tools. Since every evaluation of individuals is independent of each other, these methods are highly parallelizable. One of the classical parallelization design consists in a *Master/Worker* pattern [8]. In a distributed implementation, each process is usually located into a separate computing unit, which results in a Parallel Evolutionary Strategy (PES).

Regarding the population evolution, several parallel techniques have been designed and developed to improve the quality of the solution and reduce the execution time [11], which could be considered for any PES. The methods compared in this work are framed within the *multiple sub-populations* scheme, or Island Model: the processes are organized in "islands", in which a sub-population is separately evolved. The islands are managed by a master process, which carries out the evolutionary actions, but delegates the evaluation of the objective function to worker processes. The islands exchange individuals occasionally in a "migration" operation, promoting the diversity in each island.

3.3 ESSIM-EA and ESSIM-DE

The development of the two methods compared in this work is based on the ESS method (Evolutionary-Statistical System) [4], a DDM-MOS for uncertainty reduction. ESS uses Evolutionary Algorithms (EAs) as search strategy, having a unique population scheme applying parallel evaluation to optimize the search parameters that feed the model. Each individual consists of a set of values that represent the input parameters of the model (e.g., moisture content in vegetation, vegetation type, wind direction and speed, the terrain slope, etc.), in other words, they represent different *scenarios* of the input parameters. ESS incorporates a Statistical Analysis module to determine the trend of the fireline in each simulation step, and a EA component is used to guide the search of the solution and reduce the number of cases under study in a statistical stage.

In recent years, the ESS method has been studied in depth and various improvements have been proposed in order to achieve a better quality approach [9, 10]. As a result, the ESS Island Model (ESSIM) happened to be a very suitable method, where the addition of the Island Model and a migration phase improved significantly the quality of the predictions. In this work, we analyze two variants of ESSIM: ESSIM-EA and ESSIM-DE. Both methods use PES to optimize the parameters search that feed the model. In ESSIM-EA, EA stands for *Evolutionary Algorithm*, i.e., the evolutionary phase follows the steps of a standard EA (initialization, selection, crossover, mutation, evaluation), whilst in ESSIM-DE the PES used is the Differential Evolution (see Section 3.1).

4 Experimentation and results

ESSIM-EA and ESSIM-DE were applied to five instances of prescribed burns carried out in a hillside of Serra de Lousã (Gestosa, Portugal). The size of each plot was defined according to the available surface, to the characteristics of the land and to the project requirements. It is important to remark that the size of each plot has limited dimensions due to the need to control

Table 1: Description of the experiments: dimensions, slope, time step and ignition type

Exp.	Width (m)	Lenght (m)	Slope (deg)	Initial time (min)	End time (min)	Increment (min)
A	89	109	21	2.0	14.0	2.0
B	95	123	21	2.0	12.0	2.0
C	60	90	6	2.0	10.0	2.0
D	89	91	21	2.5	12.5	2.5
E	75	126	19	3.0	9.0	1.0

all risk factors related to the fire. For each of the experiments we have defined discrete time steps representing the advance of the fire front. The number of time steps is not the same in each experiment since all plots correspond to different dimensions. Table 1 shows the details of each experiment (i.e., dimensions, slope, initial time and final time).

As mentioned earlier, the two methods focus its operation on parallel evolutionary algorithms. Since the individuals of each population are generated using random seeds, each experiment was executed 30 times with the aim of determining an average of results. The two methods have the same parameters setting for all common features related to the operational configuration of the PES: an amount of 5 islands, 7 workers for each island and a population size of 200. They only differed in that the amount of migrants in the case of ESSIM-DE was 20% of population, whereas in ESSIM-EA it was only the best individual. The termination criterion in the evolutionary process consist in achieving a fitness threshold of 0.7. This threshold value was established taking into account a fitness value that represents an acceptable quality of prediction. The parameters for ESSIM-DE also include those necessary for the configuration of Differential Evolution: Cr=0.3, F=0.9 and binomial crossover.

The experiments were executed on a Linux cluster with 44 processing units distributed over 10 quad-core CPUs with 4 GB of RAM and Gigabit Ethernet and using MPICH.

4.1 Performance Analysis

Figure 3 shows the average fitness values obtained by each method in each prediction step. Axis x represents the prediction steps according to progress of the fire front, and y axis represents the average fitness value obtained from 30 executions. Figure 4 depicts the distribution of the 30 executions obtained by each method in each prediction step, represented by boxplot charts. The labels in the x axis indicate the corresponding method method ("EA" stands for ESSIM-EA, "DE" stands for ESSIM-DE) and prediction step.

In experiment A (Fig. 3 (a)) it is possible to see that ESSIM-EA reaches average fitness values near to 0.8 in all the steps. ESSIM-DE results are far from the results provided by ESSIM-EA, specially from step 8. Analysing the boxplot graphs, it is possible to see that the 75% of the fitness values obtained (1st, 2nd and 3rd quartile) are superior to 0.65 approximately. Also, it is possible to observe that there may be significant differences between ESSIM-EA and ESSIM-DE for the prediction steps 10, 12 and 14 (Fig. 4 (a)). In order to confirm such assumption, it was necessary to carry out the non-parametric statistical test Wilcoxon rank sum test. The analysis suggest that ESSIM-EA is the best method in this experiment, achieving better average fitness values and low distribution of the obtained results.

Experiment B (Fig. 3 (b)) is a very interesting case for our analysis. It can be seen at steps 6 and 8 that ESSIM-DE obtains the best fitness values. However, the quality obtained in

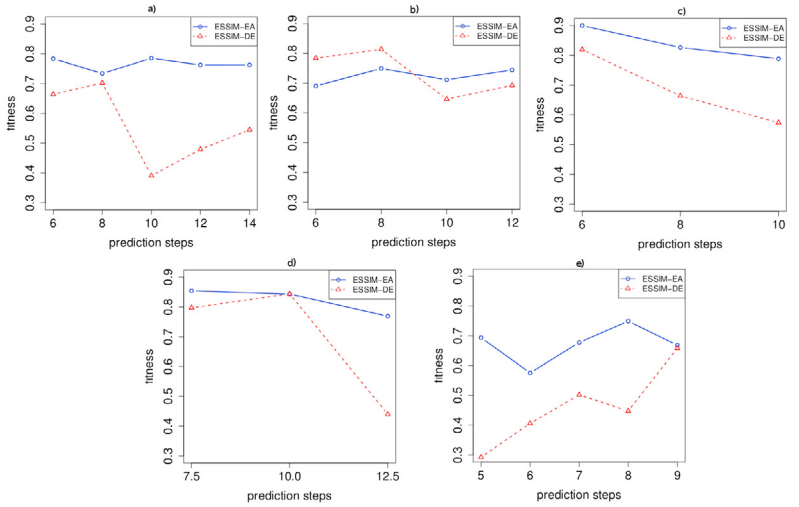

Figure 3: Average fitness values for each method and time step. Experiments A-E.

the last two steps are below the averages ESSIM-EA. But even when ESSIM-DE obtains lower quality of results for the last two steps, it is important to analyse the runtime performance of the method in this experiment. Table 2 shows the average execution times for the experiments using each method. It is possible to see that ESSIM-DE achieves a reduction in the execution time compared with the other method. This is an example where we emphasize that low computational cost methodologies may generate acceptable results, i.e., the fitness values obtained with ESSIM-DE for the last two steps can be considered as acceptable results due to the gains in execution time, achieving a short-term prediction. In addition, it is possible to see that ESSIM-DE achieves a low distribution of the obtained fitness value (Fig. 4 (b)). The analysis suggest that ESSIM-DE can be considered as the best method in this experiment.

In experiment C (Fig. 3 (c)) the analysis suggest that ESSIM-EA is the best method, achieving average fitness values over 0.8 in all the prediction steps. ESSIM-DE reaches an acceptable quality in step 6, but the results fall to values lower than 0.7 in the last two steps.

In experiment D (Fig. 3 (d)), the average fitness values achieved by ESSIM-EA and ESSIM-DE are higher than 0.8 for the first two steps, but the fitness obtained with ESSIM-DE decreases to lower values on step 8. It can be seen that the 75% of the results obtained by ESSIM-EA are superior to 0.8 for the two first steps. This feature is further evidence to suggest that ESSIM-EA is the best method in this experiment, achieving average fitness values over 0.75.

The last experiment is depicted in Fig. 3 (e). In this case, the simulation consists in six prediction steps. It is possible to see that ESSIM-EA outperforms ESSIM-DE in all the prediction steps. Although ESSIM-DE obtains less quality of results, it is possible to see that there is a growing tendency, achieving good quality in the last step. Moreover, in this experiment is important to remark that ESSIM-DE achieves an important reduction in the execution time, of around 70% the time of ESSIM-EA.

Focusing on the results obtained by each method, it is possible to see that the prediction quality varies from one step to another. It can be seen that the fitness value increases or decreases depending on the step that is being analysed. The prediction of the fire line on a certain time instant uses a pattern of the fire behaviour obtained in a calibration stage (CS)

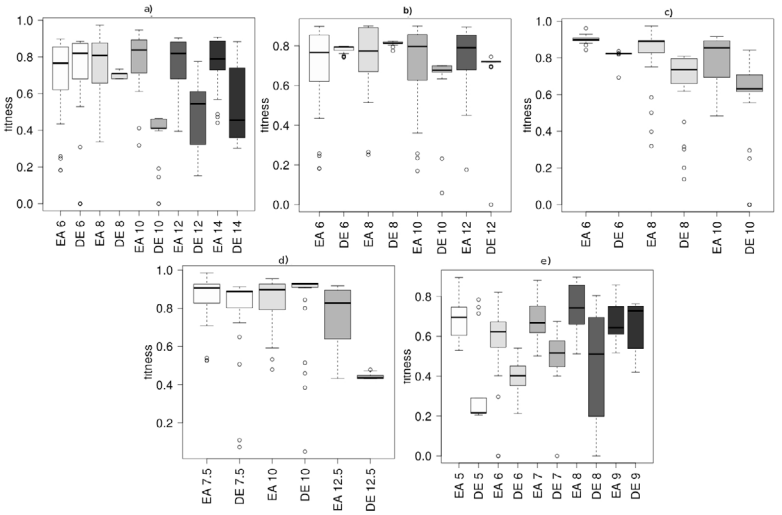

Figure 4: Distribution of the results for each method and time step. Experiments A-E.

Table 2: Average execution time for each experiment.

Experiment	ESSIM-EA	ESSIM-DE
A	00:57:20	**00:37:48**
B	01:01:15	**00:49:05**
C	00:50:10	**00:27:49**
D	01:18:08	**00:43:42**
E	02:11:38	**00:41:20**

of the previous time instant, with the underlying assumption of spacial and temporal locality of the fire front. However, the environmental conditions could drastically vary over the time, e.g. a noticeable change in the wind speed or direction may alter the form of the fire front. For this reason, the predicted fire line of each simulation step must be analysed independently from the predictions obtained in next time steps. Therefore, the uncertainty on the input parameter values is reduced in an independent way each simulation step.

As mentioned above, Table 2 shows the average execution time of the 30 executions with different seeds for the two methods. It is possible to see that ESSIM-DE obtains less execution time than ESSIM-EA, achieving reduction from 10% to more than 70% approximately. This reduction in the execution time is related to the characteristic of DE of obtaining earlier convergence of the threshold fitness value. Although, ESSIM-DE does not achieve a better quality of prediction considering the whole population (with some exceptions), in all the steps.

5 Conclusions

This paper presents two methods for the uncertainty reduction in the input parameter values applied to wildland fire prediction. The methods use statistical analysis to determine the trend

of the fireline at each simulation step and to calibrate the results. They also use evolutionary algorithms to optimize the parameters search that feed the model, and high performance computing tools to improve the wildfire prediction quality and the response time, obtaining short-term solutions. The analysis of results suggest that ESSIM-EA find better results compared against ESSIM-DE in terms of quality of prediction, but ESSIM-DE achieves a significant reduction in the execution time. As future work, we plan to develop a method combining the benefits provided by the methods compared in this work.

Acknowledgment

This work has been supported by UTN under projects EIUTIME0003939TC and EIUTNME0003952, and MINECO-Spain under Contract TIN2014-53234-C2-1-R. The first author would like to thank CONICET for the PhD Grant provided.

References

[1] B. Abdalhaq, A. Cortés, T. Margalef, E. Luque, "Enhancing wildland fire prediction on cluster systems applying evolutionary optimization techniques", *Future Generation Computer Systems*, 21, 1, pp. 61-67, 2005.

[2] P.L. Andrews, "Current status and future needs of the BehavePlus Fire Modeling System", International Journal of Wildland Fire, 23,1, pp. 21-33, 2013.

[3] C.D. Bevins, "FireLib User Manual & Technical Reference", http://www.fire.org, 2004.

[4] G. Bianchini, P. Caymes Scutari, M. Méndez Garabetti, "Evolutionary-Statistical System: A parallel method for improving forest fire spread prediction", *J. of Computational Science*, 6, pp. 58-66, 2015.

[5] A. Cencerrado, R. Rodríguez, A. Cortés, T. Margalef, "Urgency versus accuracy: Dynamic data driven application system for natural hazard management", *International Journal of Numerical Analysis and Modeling*, 9, 2, pp. 432-448, 2012.

[6] M. Denham, K. Wendt, G. Bianchini, A. Cortés, T. Margalef, "Dynamic Data-Driven Genetic Algorithm for forest fire spread prediction", *J. of Computational Science*, 3, pp. 398-404, 2012.

[7] M.A Finney, "FARSITE: Fire Area Simulator-model development and evaluation", Res. Pap. RMRS-RP-4 Revised, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2004.

[8] T. Mattson, B. Sanders, B. Massingill, PATTERNS FOR PARALLEL PROGRAMMING, Addison-Wesley Professional, chapter 5, pp. 143-152. 2004.

[9] M. Méndez Garabetti, G. Bianchini, M.L. Tardivo, P. Caymes Scutari, "Comparative Analysis of Performance and Quality of Prediction Between ESS and ESS-IM", *ENTCS*, 314, pp. 45-60, 2015.

[10] M. Méndez Garabetti, G. Bianchini, M.L. Tardivo, P. Caymes Scutari, "Increase in the quality of the prediction of a computational wildfire behavior method through the improvement of the internal metaheuristic", *Fire Safety Journal*, 82, pp. 49-62, 2016.

[11] E.G. Talbi, METAHEURISTICS: FROM DESIGN TO IMPLEMENTATION, John Wiley & Sons, Hoboken, New Jersey, 2009.

[12] R. Real, J.M. Vargas, The probabilistic basis of Jaccard's Index of similarity, Syst. Biol. 45 (3) (1996) 380–385.

[13] R. Storn, K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11, 1997, pp. 341–359.