Fire Safety Journal 82 (2016) 49–62

Contents lists available at ScienceDirect

Fire Safety Journal

journal homepage: www.elsevier.com/locate/firesaf

Increase in the quality of the prediction of a computational wildfire behavior method through the improvement of the internal metaheuristic

Miguel Méndez-Garabetti [a,b,c,*], Germán Bianchini [a], Paola Caymes-Scutari [a,b], María Laura Tardivo [a,b,d]

[a] Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD), Departamento de Ingeniería en Sistemas de Información, Facultad Regional Mendoza – Universidad Tecnológica Nacional, M5502AJE Mendoza, Argentina
[b] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[c] Instituto Tecnológico Universitario, Universidad Nacional de Cuyo, Centro Universitario, Mendoza, Argentina
[d] Departamento de Computación, Universidad Nacional de Río Cuarto, Río Cuarto, Córdoba, Argentina

ARTICLE INFO

Article history:
Received 29 August 2015
Received in revised form
9 March 2016
Accepted 26 March 2016

Keywords:
Wildfire behavior prediction
Simulation
Uncertainty reduction
Parallel Evolutionary Algorithms
Statistical System

ABSTRACT

Wildfires cause great losses and harms every year, some of which are often irreparable. Among the different strategies and technologies available to mitigate the effects of fire, wildfire behavior prediction may be a promising strategy. This approach allows for the identification of areas at greatest risk of being burned, thereby permitting to make decisions which in turn will help to reduce losses and damages. In this work we present an Evolutionary-Statistical System with Island Model, a new approach of the uncertainty reduction method Evolutionary-Statistical System. The operation of ESS is based on statistical analysis, parallel computing and Parallel Evolutionary Algorithms (PEA). ESS-IM empowers and broadens the search process and space by incorporating the Island Model in the metaheuristic stage (PEA), which increases the level of parallelism and, in fact, it permits to improve the quality of predictions.

1. Introduction

Fire has been a fundamental element in the development of civilizations. Nevertheless, it represents a threat when it spreads without control causing wildfires. Wildfires may be generated by meteorological or human factors, although independently of the cause, it always generates a great impact on biodiversity, landscape, water resources and health (i.e., they generate great losses and are harmful to the environment) [1]. For example, in 2010, the extremely high temperatures that reached a record (in the Russian summer) and the drought in the region caused a large fire that burned one million hectares of forests, killing approximately 53 people [2,3] (in addition, 806 people required medical attention). In January 2014, in General Alvear (Mendoza, Argentina), 200,000 hectares of natural forests were burned due to high temperatures, low humidity and high amount of fuel accumulated in the area [4]. Finally, during February 2015 in Cholila (Chubut, Argentina) 45,000 hectares of native forests were burned, being the worst wildfire in Argentina [5,6].

Currently, there exists a great scientific effort to develop technologies and strategies to reduce the effects caused by wildfires. However, due to the complexity of the phenomenon and its characteristics, there is still a long way to go in order to achieve such a goal. Certainly, one of the most promising tools consists in the development and improvement of the ability to predict the phenomenon behavior. More specifically, this is to determine the future behavior of a wildfire once it has started, permitting to reduce damage and losses in the environment and the population by making decisions based on the areas that are most likely to be reached by the fire. Based on this, we can say that the ability to predict wildfire behavior represents a capacity that will allow us to implement useful tools in the process of firefighting.

However, it must be remembered that the prediction of any natural phenomenon is not an easy task and it takes time, especially, if high quality prediction is desired.

Predicting wildfire spreading consists in determining which will be the direction and speed of fire propagation, the shape of

* Corresponding author at: Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD), Departamento de Ingeniería en Sistemas de Información, Facultad Regional Mendoza – Universidad Tecnológica Nacional, M5502AJE Mendoza, Argentina.
E-mail addresses: mmendez@itu.uncu.edu.ar (M. Méndez-Garabetti), gbianchini@frm.utn.edu.ar (G. Bianchini), pcaymesscutari@frm.utn.edu.ar (P. Caymes-Scutari), lauratardivo@dc.exa.unrc.edu.ar (M.L. Tardivo).

http://dx.doi.org/10.1016/j.firesaf.2016.03.002
0379-7112/

the fire front, flame intensity, etc. Once the fire has started, it is necessary to predict its progress for the near future using as much information as possible about the fire front and the environment in which this occurs (i.e., climatic conditions, terrain conditions, vegetation conditions, fuels, etc.).

It is important to emphasize that wildfire prediction, as any other natural phenomenon, is not an easy task due to the complexity of the models used, the amount of variables involved and the uncertainty that they often exhibit.

In this paper, in Section 2 the issue of uncertainty in prediction systems is described. Next, a brief explanation of the concepts and tools that are used in the development of the methods is analyzed in Section 3. The approach, implementation and methodology of Evolutionary-Statistical System with Island Model (ESS-IM) are explained in Section 4. Finally, the experimental results and comparison are shown in Section 5, and conclusions are provided in Section 6.

2. Uncertainty in wildfire prediction

In wildfire behavior prediction there are different sources of uncertainty that affect the precision of the method. These sources are related to the limitations of the model used, parameters with unknown values, dynamic changes produced by the model, discretization of values, the difficulty to quantify the parameter values in real time, etc. In this section the sources of uncertainty and the existence of uncertainty in classical prediction are briefly discussed.

2.1. Uncertainty sources

The uncertainty concept itself has different meanings and levels that can refer to the lack of knowledge, the lack of certainty, among others. In this context, we refer to the lack of knowledge of the parameters that determine the behavior of the model. Mainly, this kind of uncertainty is usually observed in those variables that present a dynamic behavior. Some examples of this kind of variables are wind speed, wind direction, humidity content in vegetation, etc. These variables strongly affect wildfire behavior and should be measured in real time. To do this, devices such as wireless sensor networks (WSNs) could be of great help [7]. By using this type of sensors in areas affected by fires, we can obtain temperature measurements, wind speed and direction, etc. An example of this is [8] where WSNs are used as a tool for early detection and wildfire monitoring.

While the use of WSN can be a promising tool to reduce uncertainty in the input parameters, this technology can only be used in protected areas where the installation of a sensor network is possible. However, it is not feasible to install an extensive network of WSN in forests worldwide. Therefore, it is necessary to develop and improve methods and tools to address the problem of uncertainty into input parameters.

In our work, we consider that there is no exact set of input parameters to feed the propagation model because it is not possible to know the exact value of each parameter at the beginning of the fire and through the time. Furthermore, in most cases these models cannot be analytically solved and must be solved by applying numerical methods that are only an approximation to reality. Therefore, to make a wildfire behavior prediction with estimated values cannot be considered as reliable.

2.2. Classical prediction

In general terms, the Classical Prediction method consists in evaluating the position of the fire after a certain initial period of time, using any existing fire simulator behavior. A general scheme of this kind of methodology can be observed in Fig. 1. 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 weather data, vegetation, and terrain description (all these data are called input parameters). Each input parameter has a value assigned, and 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}) through a single simulation. Furthermore, it is necessary to say that in any prediction method it is expected the estimated prediction carried out by the simulator to match reality in the best possible way. However, due to the model complexity, the uncertainty in the input parameters, and since the prediction is based on a single simulation, this prediction methodology provides generally predictions that are far from reality. Examples of classical prediction in wildfires are [9–15]. Due to the limitations of Classical Prediction, the development of methods that allow for reducing the uncertainty in order to improve the prediction quality has been necessary.

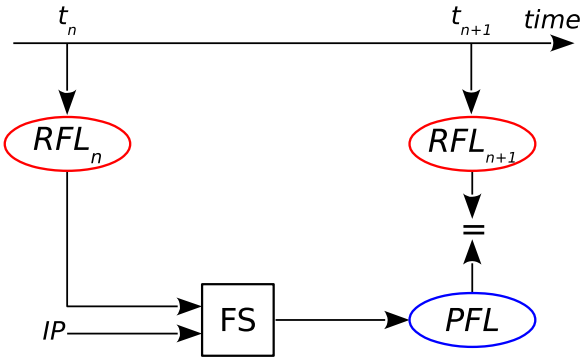


Fig. 1. Classical Prediction: diagram of wildfire propagation (**FS**: Fire Simulator; **PFL**: Predicted Fire Line; **IP**: Input Parameters; $\mathbf{RFL}_n$: Real Fire Line on time n).

2.3. Uncertainty reduction

As we mentioned before, one of the factors that obstruct the classical prediction methods to obtain similar predictions to reality is lack of precision, i.e., the uncertainty in the parameters that feed the model. When this imprecision is present, the prediction capacity of the method is considerably affected, since this is equivalent to feeding the simulator with incorrect values, which usually will produce wrong predictions.

Due to the imprecision of the input parameters and the difficulty to measure them in real time, it is necessary to turn to some technique which will be able to reduce uncertainty, such as the Data Driven Methods (DDM). The DDM consider a large number of values for each parameter, instead of a single value for each parameter. Subsequently, these methods perform a search (i.e., by means of Genetic Algorithm, Taboo Search, Simulated Annealing) to find a set of parameters which describes, in the best possible way, the previous fire behavior that will be used to predict the near future behavior, based on some kind of time and space locality.

In other words, the DDM perform a calibration to obtain these "optimal" values of input parameters. Nevertheless, these methods obtain a single set of values, and for those dynamic parameters, the value found is not generally useful to correctly describe the model behavior. This category of methods is called Data Driven Methods of Unique Solution [16–18].

There is another classification of DDM that works with overlapping cases and combinations of parameters to make

predictions. This category is called Data Driven Methods with Multiple Overlapping Solutions (DDM-MOS). Statistical System for Forest Fire Management (S^2F^2M) [19], ESS [20,21] and ESS-IM [22,23] are included in this category. In addition, DDM are strongly related to Data Assimilation Methods (DAM), which are characterized by incorporating data into a working model [24]. From this point of view, we can say that ESS-IM is a hybrid version between DDM-MOS and DAM, this is because ESS-IM is not aimed at finding a single dataset with specific values. However, the method aims to find a set of values that will improve the final outcome process and also include the incorporation of data at execution time.

To understand the internal working process of these methods, it is necessary to briefly explain the elements on which these methods base their operation, which will be dealt with in the following section.

The aim of this work is to enhance the capabilities of the ESS. ESS is an uncertainty reduction method that has been applied to wildfire spread prediction. Operation of the method is based on statistical analysis [25], parallel computing [26] and Parallel Evolutionary Algorithms [27]. The interaction of these three components has allowed ESS to improve the prediction quality made by his predecessor (S^2F^2M), for more information see [19]. In this paper, following the same idea, we have increased the level of parallelism and modified the internal architecture, with the goal of improving the quality of prediction by incorporating the Island Model [28] to ESS. Fig. 2 shows the evolution of each method and their respective components.

3. Components involved in the DDM-MOS

Each method is constituted by different elements and technologies in order to try to achieve the proposed objectives. This section briefly discusses each of the pillars on which S^2F^2M, ESS and ESS-IM base their operation. These are statistical analysis, parallel computing and Parallel Evolutionary Algorithms.

3.1. Statistical analysis

As we mentioned in the previous section, the methodology presented in this paper bases its operation on certain components, one of which is the statistical analysis. Statistics is a formal science that deals with the collection, presentation, analysis and use of data to take decisions, solve problems, and design products and processes [25]. Thus, statistics permits to look for correlations and dependencies among variables that affect a physical or natural particular phenomenon and the phenomenon itself.

In the challenge dealt with in our work, wildfire behavior prediction involves a set of parameters (also considered as factors or variables) that determine and affect the phenomenon behavior, for example wind speed and direction, moisture in vegetation, slope, among others. These factors feed the prediction model, which aims to determine the fire behavior according to the input values of each variable. Because each parameter can have different values (independently of the other ones), it is necessary to evaluate the effects of each parameter along with the values of the other ones. Therefore, one efficient way to solve this problem, from the statistical point of view, is the factorial experiment design. According to [25] a factorial experiment is one in which the factors are varying together, such as occurs in wildfires with wind speed and direction, and vegetation moisture. In other words, it is a type of experimental design in which every value of one factor is tested in combination with every value of another factors. In general, in a factorial experiment, all possible combinations of values are tested [25].

In our context, each possible combination of parameters values is called a *scenario* (this concept will be will be dealt with in depth in the next sections). Therefore, if we want to determine which portion of a surface of terrain (divided into m cells) will be reached or not by fire, we can define n different scenarios and calculate the *Ignition Probability* (P_{ign}), for each cell as follows:

$$P_{ign}(C) = \frac{n_C}{n} \tag{1}$$

where $P_{ign}(C)$ is the ignition probability for cell C and n_C is the number of scenarios where cell C was burned. To better understand its concept, let us consider the example of Fig. 3a. In such example the number of scenarios is $n=4$, therefore, 4 intermediate

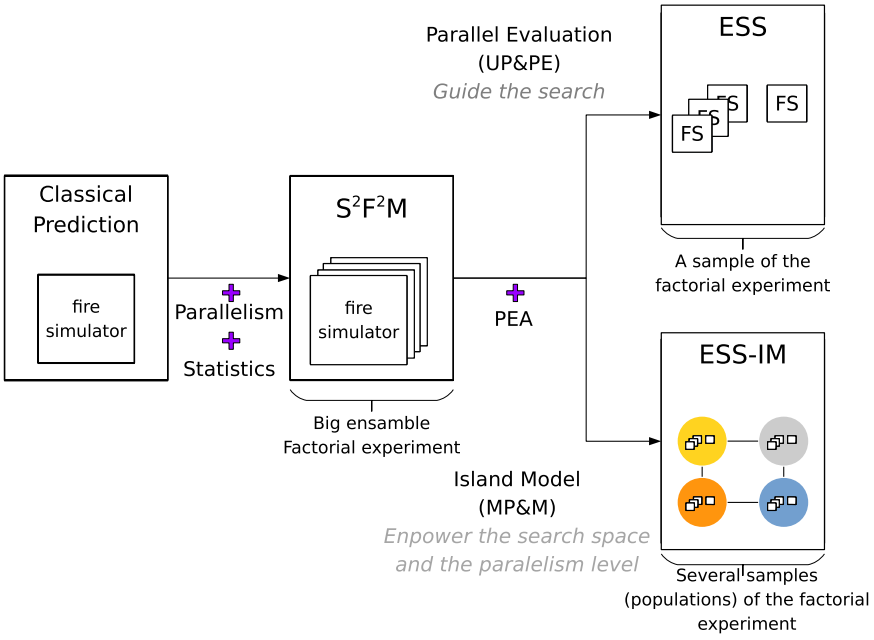


Fig. 2. S^2F^2M, ESS and ESS-IM: components and evolution from the Classical Prediction.

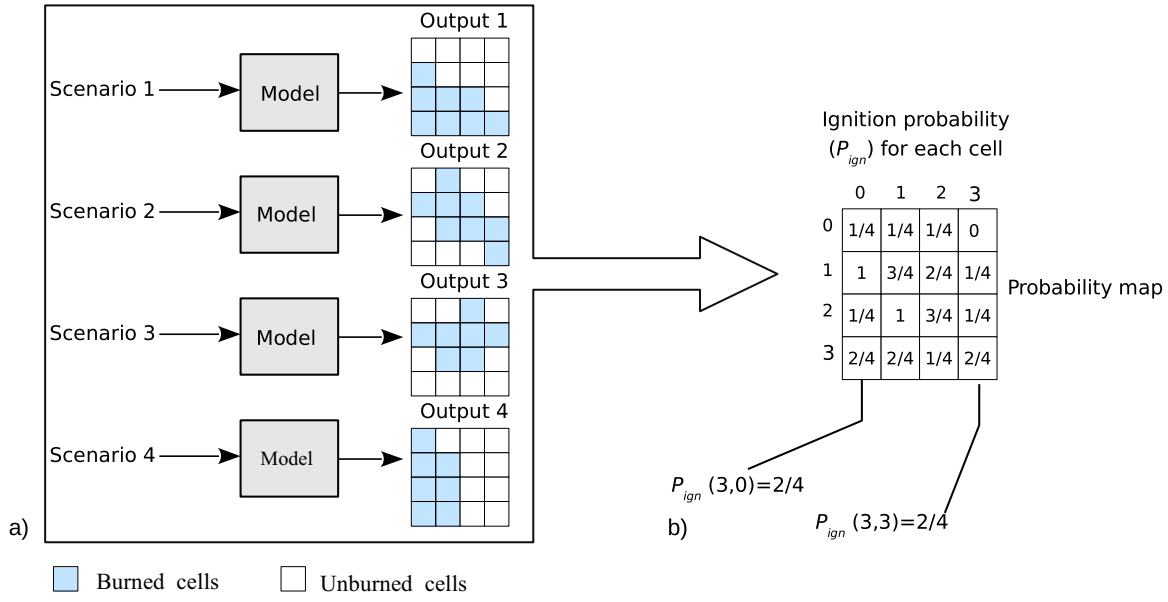


Fig. 3. (a) Example of calculation to build the probability map, and (b) probability map.

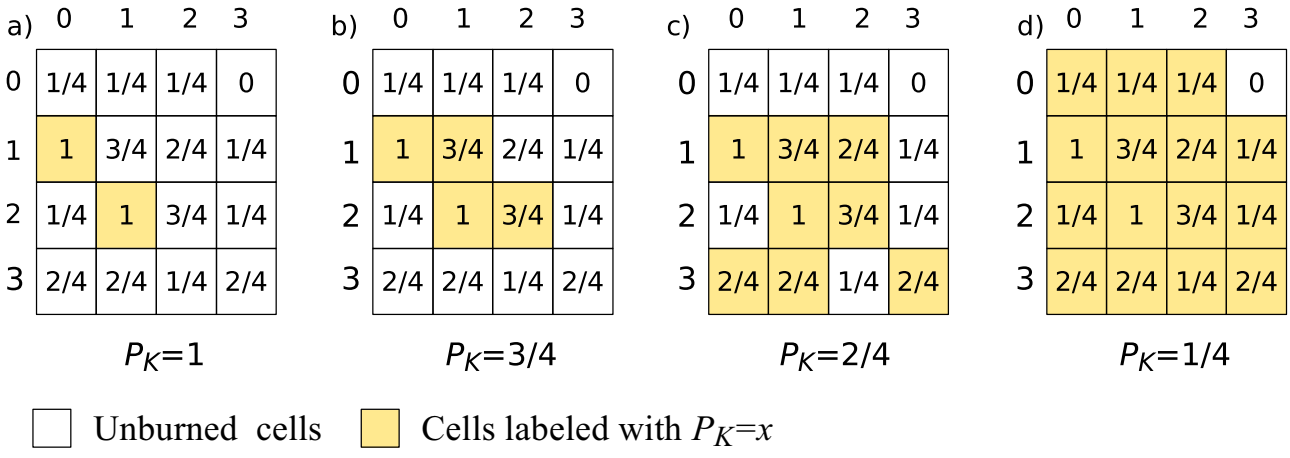


Fig. 4. Four examples of probability map with probabilities: (a) P_K=1, (b) P_K=3/4, (c) P_K=2/4 and (d) P_K=1/4.

outputs are obtained (*outputs 1, 2, 3 and 4*) and considering Eq. (1), we can calculate the ignition probability that takes into account these four scenarios. In this way it is possible to generate the probability map of Fig. 3b. This map, as can be seen, is a matrix with a value associated to each cell that represents the ignition probability of each cell. The set of cells, whose P_{ign} value is higher than or equal to a certain particular value P_K, where $0 \le P_K \le 1$, constitutes what we call the probability map with probability P_K. Fig. 4 shows the graphical representation of this concept.

It is important to emphasize that the probability map includes all the possible combinations of the resulting values of the parameters that exhibit uncertainty. In consequence, for each parameter there is a certain range of values that the parameter can take, as well as an increment value to move within such range. It is also important to remark that a certain number of cases do not constitute significant values to the global result, either they are now redundant, or because they are too far from reality.

3.2. Parallel computing

According to [29] parallel computing, there is more than a strategy to achieve high computational performance: it is a vision of how computation can seamlessly be a scale from a single processor to virtually limitless computing power. Several scientific activities require significant computing power (e.g., Medical Imaging, Oil and Gas, Bioscience, Chemical Engineering, Economics and Financial, Electronic Design, Geosciences, Mechanical Design, Defense and Energy, Weather Forecasting, etc.) in the wake of the large amount of calculations that must be performed to solve problems. Systems of high performance computing (HPC) provide greater computational capacity than a sequential computer as they allow us to share tasks among different processors in order to gain time and cost by performing them [26]. Among the parallel programming paradigms available, we have chosen the *Master–Worker* paradigm [30,31] for this work. In this paradigm, a process (master) generates many sub-problems, which are sent to be solved by someone else (workers), i.e., the master process sends the tasks to be simultaneously performed by each worker and then it return the result to the master. Usually there are no significant dependencies between calculations of worker processes. In this way, by means of cooperation and coordination, it is possible to reduce the execution time of extensive computations, or complex task can be addressed while maintaining a reasonable execution time.

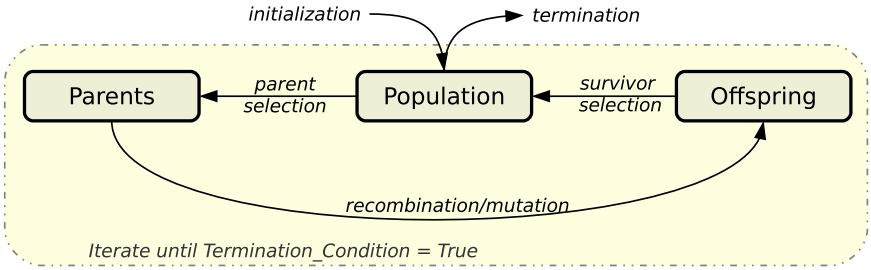


Fig. 5. A schematic diagram of an evolutionary algorithm.

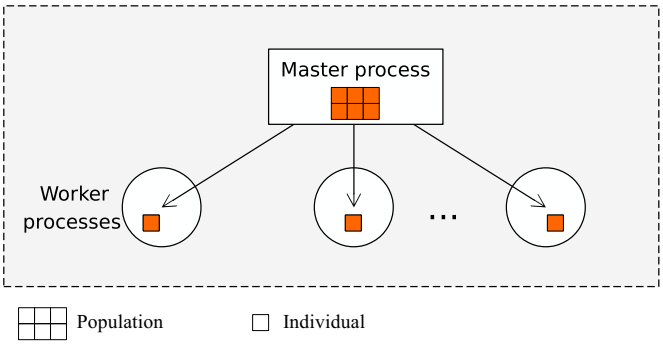


Fig. 6. Master–Worker PEA, the master process applies the genetics operators, and distributes individuals to the workers. The workers only evaluate the fitness of the individuals.

3.3. Parallel Evolutionary Algorithms (PEAs)

Evolutionary Algorithms (EAs) are considered an efficient search method inspired in natural selection and genetics to solve optimization problems [32]. In general terms, the process consists of a series of iterations, each of them called *generation*, where a sample of the searching space (in our case this would be represented by a sample of possible scenarios or parameters values combinations) is organized as a particular set of possible solutions (called *individuals*), which constitutes a *population*. The general EA framework is shown in Fig. 5. The population iteratively evolves mimicking the principles of natural biological evolution and survival of the fittest so that with the aim of the population that converges into a good solution [33]. Basically, the process consists in selecting a sample of parents of the population, which are subjected to different operators to generate the set of offspring. Later, they are introduced to the population replacing individuals with the worst features. In the best of cases, this will be the optimum solution, but in general a value that represents a threshold of what would be an acceptable solution is defined.

EAs may represent solutions in different ways: permutations, binary strings, integers, real numbers, arrays of floating point numbers, etc. [34,35] . Each EA must have a *fitness function* that computes a value for each individual to determine how accurate the solution that the individual represents to the problem is. The EAs must allow that good solutions are more likely to reproduce in order to generate new solutions, which is achieved through genetic operators. The tree classes are *crossover*, *mutation* and *replacement*. When EAs are used in complex problems, these are often implemented in parallel, allowing the reduction of processing times and the increase of the search capability of the algorithm.

Therefore, if the search is performed in parallel, the EAs could provide many possible solutions instead of only one. PEAs are classified into three main groups based on the number of populations involved in the algorithm, the treatment of each population and the genetic operators involved in the intra- or inter-evolution: Unique Population and Parallel Evaluation (UP&PE), Unique Population and Overlapped Neighborhoods (UP&ON), and Multiple Populations and Migration (MP&M) [36]. In this paper, for ESS-IM we have worked with the first scheme, while the predecessor ESS involves the last schemes, which are briefly discussed below.

3.3.1. Unique population and parallel evaluation – UP&PE

In this scheme, the algorithm operates in a single population of individuals where the fitness value of each individual is evaluated in parallel by the workers. Subsequently, the results are sent to the central or master process. In each generation a certain number of individuals, according to their fitness value, are stochastically selected. Next, these selected individuals are modified, by the recombination or mutation, to form a new population. It is important to note that the selection operator is applied considering the entire population. This scheme is also known as master-slave according to [36], which nowadays is considered as Master–Worker paradigm [30,31] (see Fig. 6).

3.3.2. Multiple populations and migration – MP&M

The scheme of multiple populations is also known as “distributed EA” or EA with “island model”, where each island represents a different population of individuals. Unlike the single population scheme, it operates with multiple populations or islands, and genetic operators (mutation and crossover) are applied among individuals of the same population. Furthermore, there appears a new operator named “migration”, which performs the movement of individuals among islands, in order to add diversity and prevent premature convergence or stagnation in local values. In Fig. 7 we can see the characterization of such implementation.

Each of these schemes allows different kinds of parallelism and can have different advantages and disadvantages based on the type of problem. The PEA of ESS corresponds to the UP&PE scheme. And in our latest development, we have implemented ESS with the MP&M scheme, whose results we are called ESS with Island Model (ESS-IM) [22,23].

4. Evolutionary-statistical system with island model – ESS-IM

ESS-IM is a general-purpose parallel uncertainty reduction method that can be applied to different propagation models such as avalanches, landslides, wildfires, among others. In this work, it has been used as a parallel uncertainty reduction method for wildfires spread prediction. ESS-IM uses PEAs to optimize the parameters search that feed the model. In ESS-IM, the input parameters set (or scenarios) are represented by individuals in different populations of a given size, where each individual contains a value for each input parameters. These parameters represent the input variables (e.g., vegetation type, moisture content, speed and wind direction, slope, etc.), which determine the behavior of a wildfire. Besides, the use of PEA to guide the search, ESS-IM uses statistical analysis to calibrate the results.

The PEA in ESS-IM corresponds to a MP&M scheme. The PEA

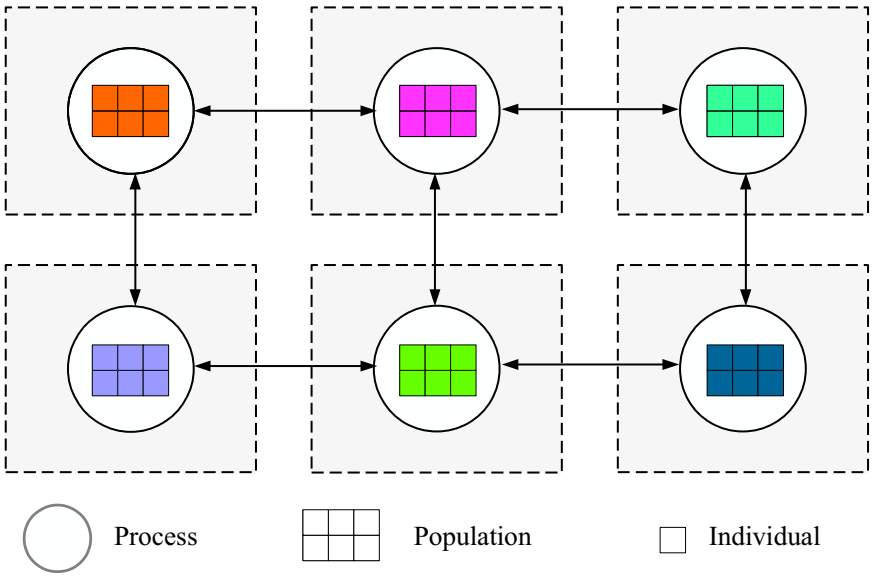


Fig. 7. A schematic diagram of a multiple-population parallel EA.

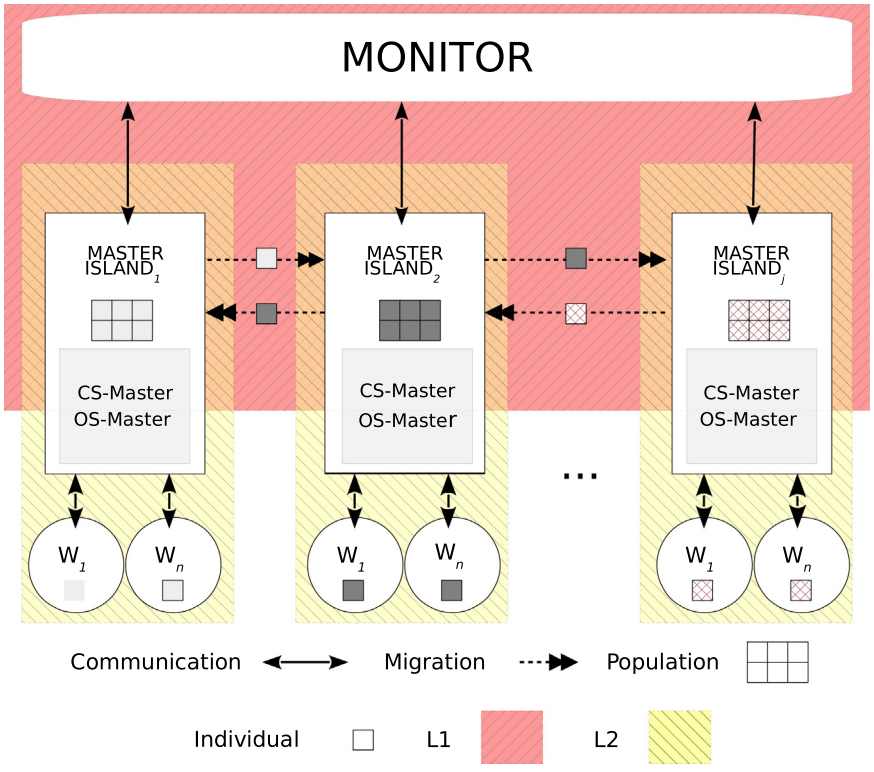


Fig. 8. ESS-IM communication model and population treatment representation. **CS-Master**: Master Calibration Stage, **OS-Master**: Master Optimization Stage.

with MP&M scheme has been implemented in ESS-IM using two parallel levels as can be seen in Fig. 8, each of them called as follow: Level 1 (**Master/WorkerL1** (Master–Worker model [30,31]) or **L1**) and Level 2 (**Master/WorkerL2** or simply **L2**). **L1** consists of a monitor process that is responsible for controlling the entire prediction process through communication with each of the islands. The monitor process carries out the initialization of the islands by sending the necessary information so that each island executes its EA. For his part, **L2** is formed by a master process (responsible for controlling the operation of the island) and *n* workers (who carry out the simulation and the fitness evaluation of the population and the sending of the results to the master). It is important to emphasize that in each execution of ESS-IM there exist *j* concurrent instances of **L2** running in parallel (based on the initialization parameters and the number of processes used). In addition to sending the individuals to the workers (in order to be evaluated in parallel) the master process carries out the evolution of the population. Also, it is responsible for the migration of individuals based on a certain communication topology and the migration parameters configured. Finally, for each simulation step, once all the master processes of each island have sent their results to the monitor process, this one is responsible for carrying out the fire line prediction. Each prediction represents the state of the fire line of the wildfire at that given instant of time.

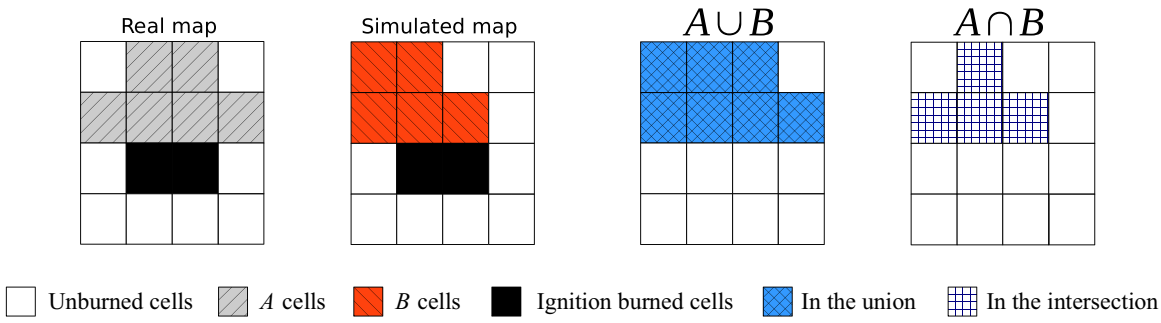


Fig. 9. Graphical representation of fitness function calculation.

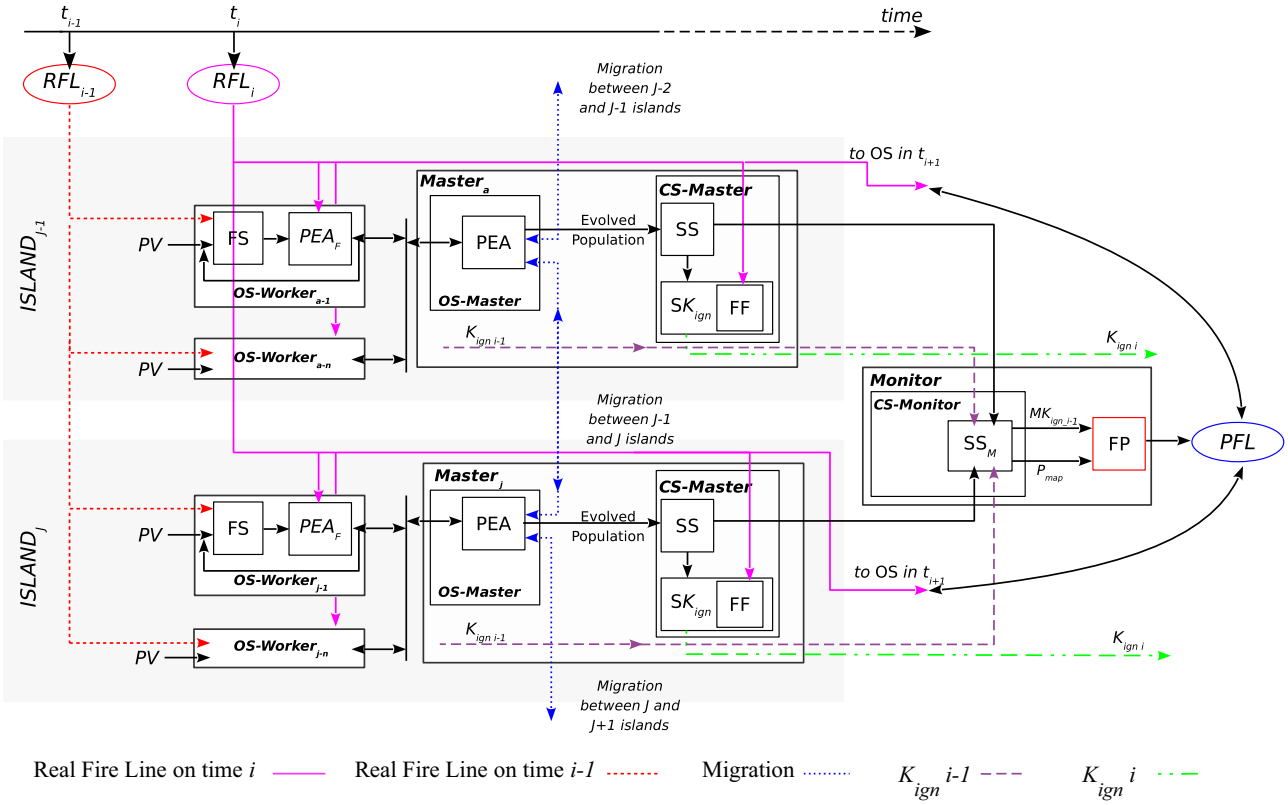


Fig. 10. Evolutionary-Statistical System with Island Model: **FS**: Fire Simulator; **PEA**: Parallel Evolutionary Algorithm; **PEA$_F$**: Parallel Evolutionary Algorithm (fitness evaluation); **OS**: Optimization Stage; **SS**: Statistical Stage; **SK**: Search K_{ign}; **K**$_{ign}$: key value used to make the prediction model; **FF**: Fitness Function; **CS**: Calibration Stage; **SS**$_M$: monitor Statistical Stage; **MK**$_{ign}$: monitor K_{ign}*value*; **FP**: Fire Prediction; **PFL**: Predicted Fire Line; **RFL**$_x$: Real Fire Line on time *x*; **PV**: Parameters Vectors.

4.1. Methodology of the ESS-IM

As we mentioned above, ESS-IM performs the wildfire behavior prediction using two levels of parallelism. On the one hand, a global level called **L1**, where each island evolves in parallel and a Monitor supervises all the prediction process. On the other hand, *j* inner levels **L2**, where the evaluation of the fitness of each individual is performed in parallel.

To better understand the operation of ESS-IM, see Fig. 10. As we can see, the system has two Optimization Stages (**OS-Worker** and **OS-Master**), two Calibration Stages (**CS-Master** and **CS-Monitor**) and a Fire Prediction stage (**FP**). The **OS-Worker** stage performs the evaluation of the population through two internal stages called Fire Simulation (**FS**) and PEA Fitness Evaluation (**PEA$_F$**). The **FS** stage must be fed with real fire line of wildfire at time t_{i-1} (RFL_{i-1}) and the input parameters contained in an individual. The simulation performed in **FS** is carried out by the simulator fireSim (which will be seen as a black box). FireSim is based on the Rothermel model [37,38] and it is implemented with the fireLib library [39]. FireLib encapsulates the Behave fire behavior algorithm [40], which uses an automata cell approach to evaluate wildfire behavior. The terrain is divided into cells and a neighborhood relationship enables to evaluate whether a cell will be burned and what time the fire will reach those cells. The simulator input is an initial ignition map and the output consists in a map of the terrain in which each cell is labeled with its ignition time.

When the **FS** ends the simulation, the resulting maps are inserted into the **PEA$_F$** stage. **PEA$_F$** compares the simulated map with the real map in the time instant t_i (RFL_i) and, depending on the difference between both maps, it determines the fitness value for each individual (i.e., the value will be between 0 and 1, in which a value equal to 1 represents a perfect prediction, and a fitness value equal to 0 would indicate the maximum error). The fitness function is based on the Jaccard-Index [41] and it is described in the following equation:

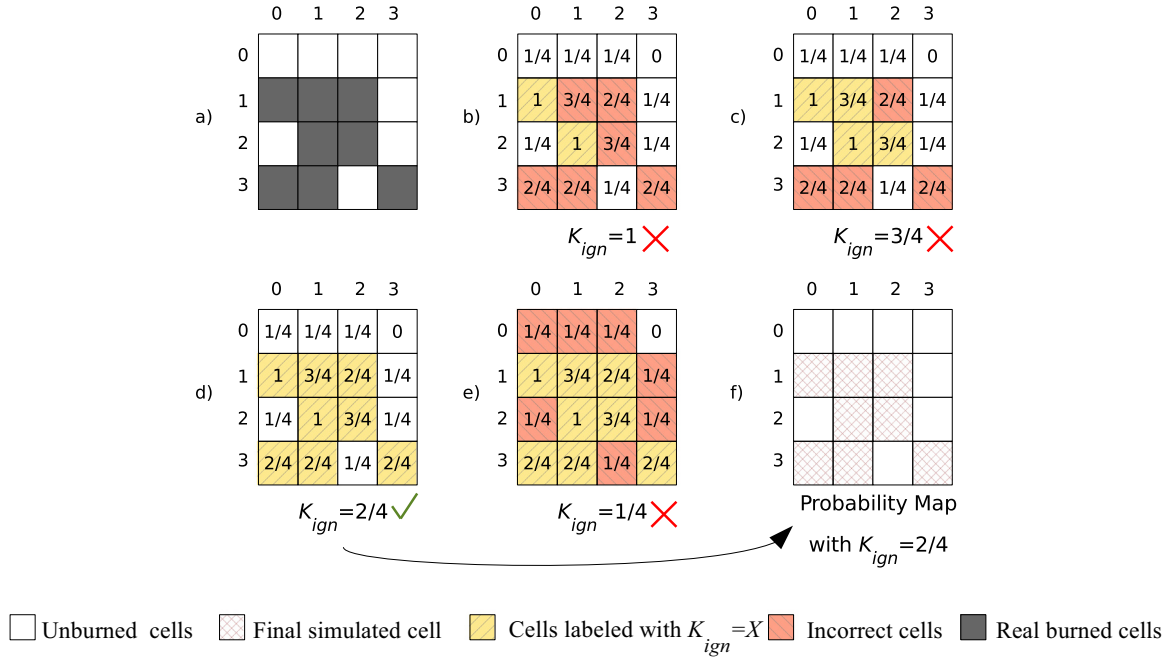


Fig. 11. (a) Burned real map representation and (b)–(f) example of a search of the Key Ignition Value K_{ign}.

$$Fitness = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

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. Fig. 9 shows an example of how to calculate this function for a terrain made up of 4×4 cells. In this case, the fitness function value is $4/7 = 0.5717$.

All the process of **OS-Worker**$_n$ stage is performed in parallel, having as many instances as number of processes involved in the resolution of the problem. All individuals evaluated by **OS-Worker** $_n$ have previously been sent to each worker from the master.

It is important to note that the functions of **PEA**$_F$ stage represent a portion of the complete PEA. The rest of the PEA processing is performed in the master process, more precisely in the **OS-Master** stage. **OS-Master** stage operates on each of the islands and is responsible for carrying out the genetic evolution (operation performed by iterating until the population reaches a certain level of quality). This stage is also responsible for carrying out the selection of individuals to migrate, send, receive, and also is in charge of replacing the migrated individuals in the current population. That is, the creation of the population, individuals' selection, crossover, mutation and communication functions with every worker. Once the population reaches a certain level of fitness, it is introduced into the calibration stage (**CS-Master**), which is also carried out in the master process. Within this stage, the evolved population feeds a sub stage called Statistics System (**SS**). **SS** implements the concept of "Ignition Probability" as seen in Section 3.1, i.e., the output of **SS** (a probability map) is used for dual purpose. On the one hand, the probability maps are used as input to the sub stage **SK** (Search K_{ign}) to search the key ignition value. During this stage, the Fitness Function (**FF**, see Eq. (2)) is used to evaluate the probability map. The K_{ign} value represents the wildfire behavior pattern in a specified time interval, and it is used to make the prediction in the next instant of time (t_{i+1}). In other words, K_{ign} value consists in a particular value of P_{ign}, whose associated probability map provides the best matching with the real fire propagation. The concept of K_{ign} is represented graphically in Fig. 11. On the other hand, the output of **SS** is introduced to **CS-Monitor** in the sub-stage **SS**$_M$ (monitor Statistical Stage) along with the j K_{ign} values calculated by the j islands. **SS**$_M$ stage has three modes of operation: (a) **Best** K_{ign}, (b) **Global** K_{ign} and (c) **First** K_{ign}. **Best** consists in choosing the better K_{ign} of all the islands, task that demands that all the islands operate synchronously. This means that an island will not be able to advance to the next simulation step unless all the islands have finished the current step. **Global** consists in calculating a K_{ign} value based on a statistical map that considers the aggregation of every statistical map generated by each island (P_{map}). Finally, **First** uses the first K_{ign} value found (i.e., uses the K_{ign} values of the island that converges faster). Once the **SS**$_M$ stage calculates or selects which K_{ign} used, this (MK_{ign}) is sent to the fire prediction stage (**FP**). Finally, the **FP** stage performs the fire line prediction **(PFL)** for each simulation step based on the probability map P_{map} and the MK_{ign} value calculated by **SS**$_M$ stage.

5. Experimental results

In this section, we present the results obtained after applying ESS-IM to three instances of controlled burns carried out in the field, particularly in Serra de Lousã (Gestosa, Portugal), where the fires were part of the SPREAD project (Gestosa field experiments 2002 and 2004) [42].

In order to verify whether the increased parallelism in the PEA offers improvements in the prediction quality, the results of the ESS-IM were compared with those produced by ESS and Classical Prediction.

Table 1
Description of the experiments: dimensions, slope, and time step for each experiment.

Experiment	Width (m)	Length (m)	Slope (deg)	Initial time (min)	Increment (min)	End time (min)
A	89	91	21	2.5	2.5	12.5
B	20	30	6	2.0	2.0	10.0
C	89	109	21	2.0	2.0	14

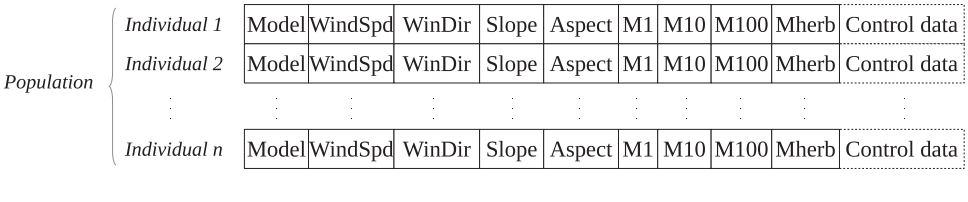


Fig. 12. Graphical representation of an individual and a population.

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, final time and the final value used to discretize comparisons over the time). As we have seen, all cases have got different sizes and the experiment C has different slopes from the other two cases. Another important value to consider is the *initial time*, because ESS and ESS-IM use a calibration step that cannot make predictions in the first instant of time.

It is important to note that ESS and ESS-IM have a non-deterministic behavior, since individuals from each population are generated using stochastically generated seeds. Due to this, we used a set of 30 different seeds, i.e., each experiment was executed 30 times for each method. Therefore, the values obtained are the average of the results for each experiment.

5.1. Parameters configuration

It is important to note that ESS-IM operates with two sets of input parameters: (a) those that feed the wildfires prediction model and (b) the parameters that determine the runtime environment of the PEA and the assigned resources to each island.

5.1.1. Model parameters

The first group of parameters directly affects wildfire behavior simulated by the model. It is important to remember that one possible configuration of these input parameters determines a particular individual in the population of an island. For some of these parameters, certain ranges have been specified, in particular those parameters that present uncertainty (taking into account the knowledge of the experiments in the field and the models of Rothermel [37,38]). Table 2 shows the set of parameters that define an individual and the uncertainty associated ranges, based on Behave Plus 5.0 [43]. Furthermore, in Fig. 12 we can see how a population is composed by different individuals.

5.1.2. Initial configuration parameters

As we mentioned earlier, there are different initial configuration parameters that condition the system prediction operation, and they are mainly related to the performance of the PEA (in addition to the parameters of the serial EA). Each of these parameters (see Table 3) impact in different ways on the results generated by the method, as well as on its performance. This paper does not include any analysis of the impact of these parameters on the prediction quality or on the method's performance. The configuration parameters that have been used for this work are listed in Table 3.

Table 2
Parameters that compose an individual and uncertainty associated ranges.

Parameter	Description	Range	Description/unit
Model	Rothermel fuel model	1–13	Fuel models
WindSpd	Wind speed	0–80	min/h
WindDir	Wind direction	0–360	Degrees clockwise from north
Slope	Surface slope	0–81	Degrees
Aspect	Direction a slope faces	0–360	Degrees clockwise from north
M1	Dead fuel moisture in 1 h	1–60	Percent
M10	Dead fuel moisture in 10 h	1–60	Percent
M100	Dead fuel moisture in 100 h	1–60	Percent
Mherb	Live herbaceous fuel moisture	30–300	Percent

Table 3
Configuration parameters in the experiments.

Parameter	Value
Number of islands	2–6
Size of populations	25–200
Individuals to migrate	16%
Migration criteria	The best individuals
Migration rate	Each 2 generations
Direction of migration	Clockwise
Number of nodes per island	2–6
Communication topology	Ring

5.2. Experiments

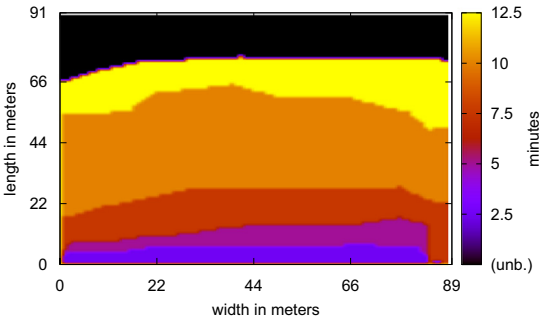
The data used in this section belongs to the prescribed fires which were started in different points on the ground surface. In all of them, the fire was started linearly (Figs. 13a, 14a and 15a). In Figs. 16a, 17a and 18a, we represent time (x-axis) and quality of prediction (y-axis) achieved by each method. It is important to recall that a fitness value equal to 1 indicates a perfect prediction and a value close to 0 indicates a very poor prediction. Similarly, in Figs. 16a 17b and 18b we represent time in the x-axis and in the y-axis the improvement (in percentage) that each method performs in each prediction step is shown, and it is calculated as follows:

$$im_{step}(\%) = \frac{(F(method)_{step} - F(cp)_{step})x100}{E(cp)_{step}} \quad (3)$$

where im_{step} represents the improvement of each prediction step, and $F(method)_{step}$ corresponds to the prediction quality value obtained by each method in each step, $F(cp)_{step}$ corresponds to the fitness value obtained by classical prediction (i.e., without using any uncertainty reduction method), and $E(cp)_{step}$ represents the classical prediction error in each prediction step (i.e., $1 - F(cp)_{step}$). Thus, the im_{step} indicates how much the negative impact of the lack of precision in the input parameters is reducing.

Although the comparison includes ESS, ESS-IM and Classical Prediction, the graphs contain five lines or four boxes because ESS-IM has three modes of operation: **Best**, **Global** and **First**. It is important to remember that the methods compared here (ESS and ESS-IM) use a Calibration Stage (**CS**), therefore, in the first simulation step these methods cannot make any predictions because in such step the first K_{ign} is calculated. To better understand this concept see Fig. 10. As already shown, the prediction is performed by the **FP** stage, which, to make the prediction in time instant $i+1$

(a) Real fire.



(b) Real spread representation.

Fig. 13. Experiment A: (a) real fire during the burns in the Gestosa area and (b) real spread representation.

(at time i), needs to have the K_{ign} of instant $i-1$. Since the beginning of the prediction chain, there is no calculated K_{ign}; therefore, it is not possible to make a prediction in the first time instant. As from the second simulation step, the calibration procedure is overlapped with the prediction phase, without interrupting the subsequent steps. Finally, it has to be observed that, at the time instant i, the $K_{ign}i$ is calculated and it will be used to predict $i+2$ at the time instant $i+1$. The results of each of the experiments are discussed below.

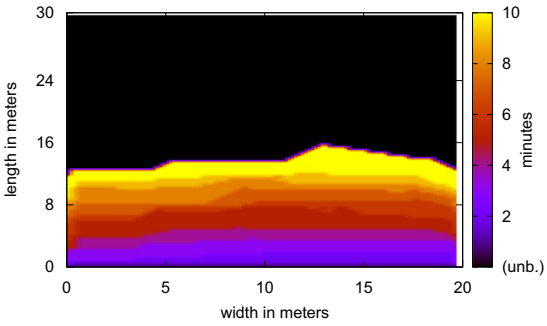
5.2.1. Experiment A

In this experiment each cell was 3.28083×3.28083 ft, the rest of parameters, such as wind conditions and moisture content, were variable (except the slope that is 21°). Of the total duration of the wildfire, the initial value was taken at 2.5 min with increments of 2.5 min. In Fig. 13b we can observe the real fire spread for each prediction step, where the black color (i.e., unb.) represents the cells that have not been reached by fire (i.e., unburned cells).

This experiment is divided into a 4 simulation step (step 1: from 2.5 to 5 min, step 2: from 5 to 7.5 min, step 3: from 7.5 to 10 min and step 4: from 10 to 12.5 min) and 3 prediction steps (step 1: from 5 to 7.5 min, step 2: from 7.5 to 10 min and step 3: from 10 to 12.5 min), this is due to the calibration process at the beginning of the prediction chain. It is important to note that in Fig. 16a x-axis values correspond to the prediction steps in minutes (moments in which the prediction quality of each method is evaluated), for that reason ESS and ESS-IM do not have values in the prediction step corresponding to 5 min instead of the classic method used, because this does not perform calibration of parameters. We can see that in the first two prediction steps (minute 7.5 and minute 10), ESS-IM (**Best** and **Global**) have obtained better results than ESS. However, these values decrease in the last step (12.5 min) and they are reached by the result obtained by ESS. In turn, ESS-IM **First** obtained a very good result in the first step. In the second step, its prediction quality decreases, taking a value close to the one in ESS, and in the last step it reaches the worst performance of the experiment. The highest fitness value (0.845428) is reached by ESS-IM **Best** at time 7.5 (first prediction step). In addition, both ESS-IM and ESS exceed the classical method at all-time points. Meanwhile, the worst performance of ESS-IM is given by ESS-IM **First**, which, although in the first prediction step, the method gets a good quality value, this is lower than ESS-IM **Best** and even ESS-IM **Global**.

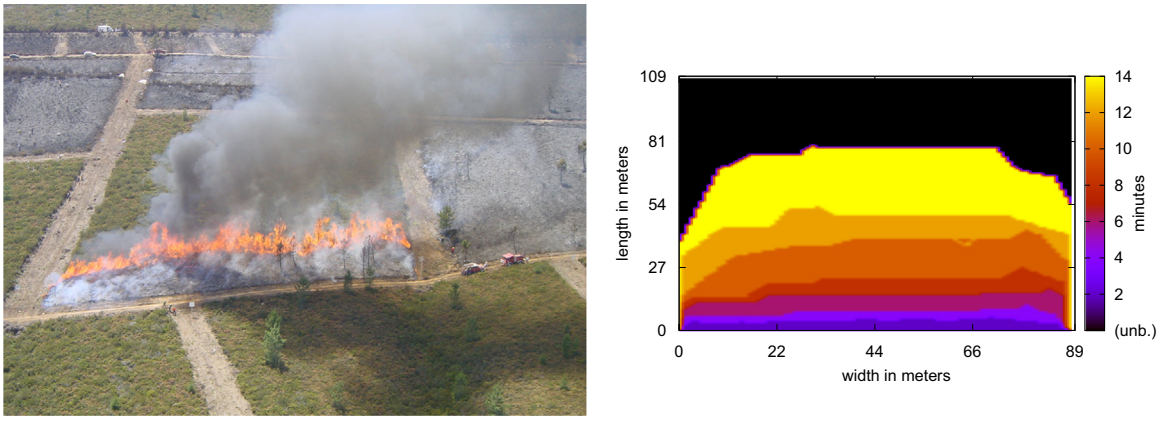
Fig. 16b shows the improvement achieved by each method at each predicting step with respect to the classical prediction; note that the highest improvement value (82.80%) is obtained by ESS-IM **Best** in minute 10, however, the best prediction quality value is 7.5 min, this is because the classic method gets its worst performance in terms of prediction quality in minute 10 (0.0908463). It is important to note that prediction quality increases in ESS-IM over ESS are related to the diversity increased in the populations that constitute the PEA in the optimization stage. Fig. 19 presents the standard deviation of each method for each experiment. As can be seen, ESS-IM **Global** is the method that presents the most

(a) Real fire.



(b) Real spread representation.

Fig. 14. Experiment B: (a) real fire during the burns in the Gestosa area and (b) real spread representation.



(a) Real fire. (b) Real spread representation.

Fig. 15. Experiment C: (a) real fire during the burns in the Gestosa area and (b) real spread representation.

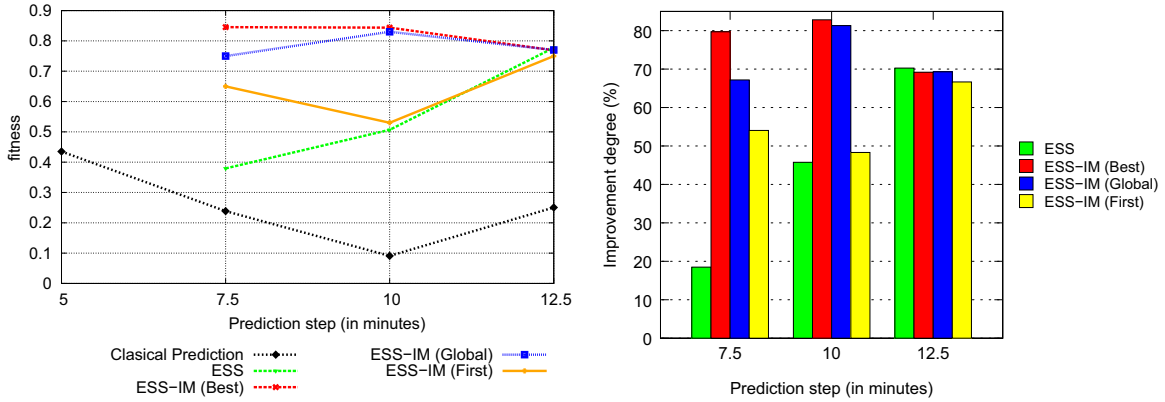
continuous behavior (in Experiment A) followed by ESS-IM **Best**. This is a very important advantage of ESS-IM **Best**, as in addition to generating the best prediction values, which is the main objective sought, this methodology provides a certain level of reliability as the predictions from one step to another do not vary significantly.

5.2.2. Experiment B

Unlike the previous experiment, it has smaller cell and slope values; the cell size is 1.09361 × 1.09361 ft and the slope 6°. The total duration of the fire is less than 2 min (8 min) than experiment A and, because it has the same number of simulation steps (4 steps) and prediction steps (3 steps), the time between each step is also lower (2 min). In Fig. 14b we can observe the real fire spread for each prediction step; it is important to note that the fire has a reduced duration and it does not cover 50% of the land. The prediction quality values are plotted in Fig. 17a and the associated improvement values appear in Fig. 17b. First, it is important to note that all methods have a very similar behavior, which may be due to the short duration of the fire, as well as due to the small number of cells involved in every step of prediction. Although we can see that ESS-IM offers the best quality in all prediction steps, it can also be observed that all methods provide very similar values, with some minor differences. This can also be seen in Fig. 19. The best prediction value is reached in the first prediction step (6 min) by ESS-IM **Best** with a fitness value equal to 0.899909 followed by ESS with a value of 0.8962068. Here it is evident that the diversity increased in the PEA of ESS-IM is not reflected in a significant increase in prediction quality, while in the remainder steps a little more significant difference is observed. In terms of method improvement (Fig. 17b), we can see that in the first prediction step, all methods get values greater than 80%. In the case of ESS-IM and ESS, over 85% of improvement is obtained. In the second step the improvement values are high (75.81%, best value obtained by ESS-IM). And as the last prediction step, although the improvement decreases (64.89%, best value obtained by ESS-IM), fitness values are of good quality: 0.788061 for ESS-IM **Best** and 0.768779 for ESS. It is important to note that although ESS-IM **Best** exceeds very little difference from ESS, the margin of improvement available is very small, therefore, a small increase in quality prediction must be very well accepted.

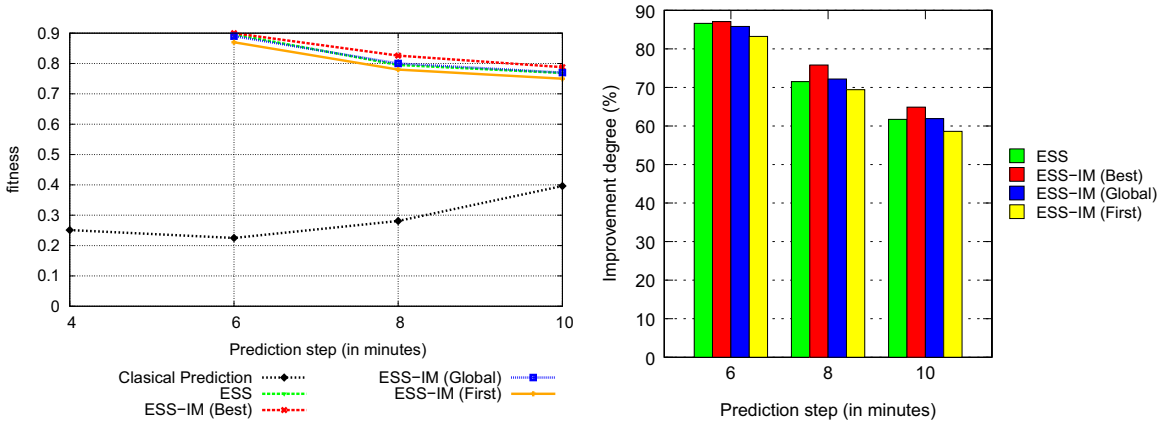
5.2.3. Experiment C

This experiment was set up with a cell size of 3.28083 × 3.28083 ft, with a slope equal to experiment A (21°). It has a total duration of 12 min, being this one of longer duration. The sampling interval was set at 2 min between each step by a total amount of 6 simulation steps and 5 prediction steps from 4 min. In Fig. 15b we can observe the real fire spread for each prediction step. In Fig. 18a we can see the fitness function values generated by each method. In contrast to the previous experiment, here we can see that in the presence of a longer duration fire, ESS-IM **Best**, differentiates even more from the other methodologies obtaining the highest quality levels in all prediction steps. The best prediction value is reached by ESS-IM **Best** in minute 10 with a fitness value equal to 0.785167 followed by ESS-IM **Global** with a value of 0.7585468. An important thing to consider is that,



(a) Fitness comparison. (b) Improvement comparison.

Fig. 16. Experiment A: comparison between the fitness (a) and improvement (b) obtained for each method.



(a) Fitness comparison. (b) Improvement comparison.

Fig. 17. Experiment B: comparison between the fitness (a) and improvement (b) obtained for each method.

although this experiment shows more steps than previous experiments, the prediction quality of ESS-IM **Best** remains constant between 0.73424225 and 0.785167. This is demonstrated in Fig. 19, where we can see that ESS-IM **Best** is the method that has the most continuous behavior. In terms of method improvement (see Fig. 18b), it shows that the best performance is obtained at minute 12 with a value of 73.88%.

5.2.4. Additional considerations

Beyond the previously presented analysis, it is important to stress some details that may be considered in the development or improvement of future methodologies. Although the best prediction quality and uncertainty reduction values have been achieved mostly by ESS-IM **Best**, there exist isolated cases where less computational cost methodologies generate acceptable results. For example, in experiment A (Fig. 16a) in the last prediction step the results obtained by ESS-IM **First** are acceptable compared with those obtained by ESS-IM **Best** or **Global**. Experiment B (Fig. 17a) is a very interesting case for our analysis, as it can be seen at every prediction step with less expensive methods, from a computational point of view, very similar results were obtained with ESS-IM **Best**. Experiment C (Fig. 18a) is no exception to this analysis since the same thing occurs in the experiments A and B. The specific case corresponds to the small difference in prediction quality obtained at minute 8 among all methods, although the ESS-IM **Best** gets the best results, the other methodologies have

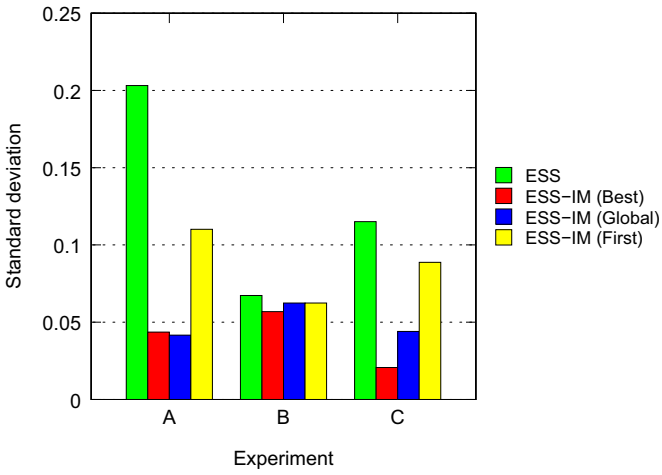
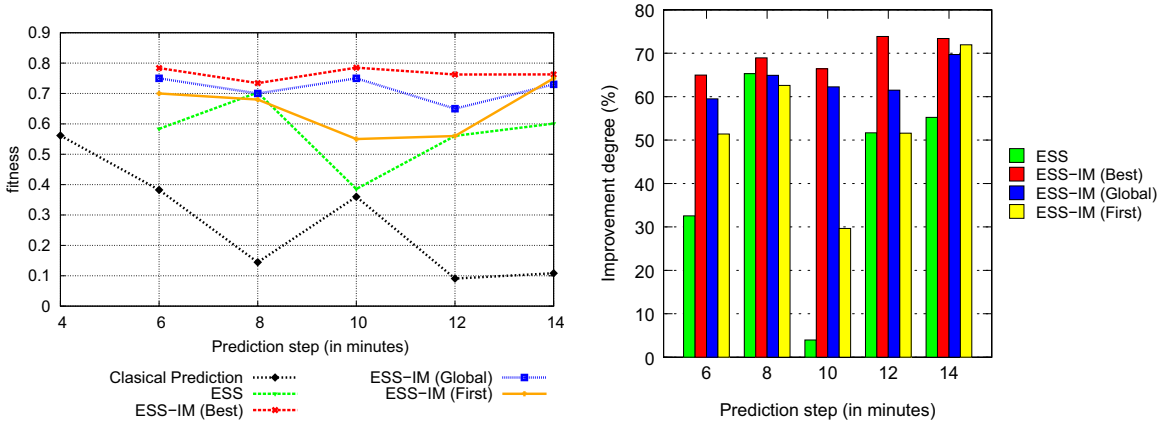


Fig. 19. Variance values of each method and each experiment.

very close values. Based on this, a version of ESS-IM that operates with the three modes in parallel could be considered in the future: **Best**, **Global** and **First**, and by a certain degree of intelligence determine when a solution of lower computational cost can be considered acceptable and advance to the next prediction step reducing the processing time.



(a) Fitness comparison. (b) Improvement comparison.

Fig. 18. Experiment C: comparison between the fitness (a) and improvement (b) obtained for each method.

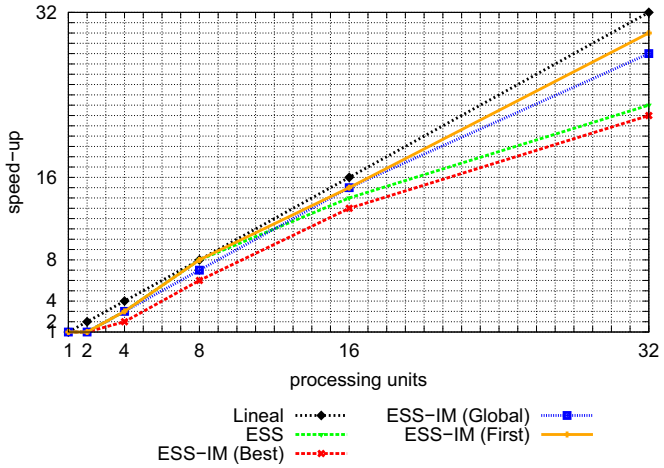


Fig. 20. Speed-up analysis.

5.3. Performance analysis

To assess the performance in terms of computational cost we used the measure known as Speed-Up [44]. This measure is very much used in the parallel computing area and it is defined as the ratio between the time required to solve a problem using a single processing unit and the time required to solve the same problem with x identical processing elements. The calculation is performed as follows:

$$S_P = \frac{T_1}{T_P} \tag{4}$$

where T_1 represents the serial execution time and T_N the parallel execution time with P being processing units.

The experiments were carried out on a cluster Linux with 32 processing units (Intel-Q9550 processors), 4 GB of RAM, Gigabit Ethernet network and under a message passing environment MPI [45]. In Fig. 20 we can observe the results of speed-up analysis of ESS and ESS-IM. The graph corresponds to the use of 1, 2, 4, 8, 16 and 32 processing units. To draw up each curves of speed-up, the executions were carried out on the three experiments and the average value for each configuration of processing units was recorded.

As we can see, the best performance in terms of computing efficiency is obtained by ESS-IM **First**, this is because this method makes prediction with the results of the first island that has ended (i.e., prioritizes finalize in the shortest time possible). Second, it is ESS-IM **Global**, from which a similar performance to ESS-IM **Best** can be expected since both perform almost the same amount of processing. But it is important to note that ESS-IM **Best** has more communications between islands which impacts negatively on performance. Beyond this, it is important to note that, using 16 processing units ESS-IM performance is very similar to other methodologies. Therefore, an effort should be made to improve resource optimization so that the method can offer higher prediction quality and so be the method with the best performance.

It is also important to note that although ESS-IM has only been evaluated with the Rothermel model, the same can be used with other fire behavior models and even with other models with propagation features. This is because ESS-IM is designed in a modular way allowing the change of the core simulation, even maintaining independence of computational cost since ESS-IM can be configured – based on the model requirements and the scenarios size to be simulated – with the amount of required computational resources (when available).

6. Conclusions

This work has focused on a very important problem that requires a quality prediction to minimize the negative effects on the ecosystem and the environment: wildfires. The main objective in the field of forest fire simulation is to create tools to assist in firefighting, whether by fire prevention, fire management, forecasting or firefighting. Based on this, it is vital to develop appropriate forecasting tools to take the right decisions. For this reason, we have developed a method that increases the parallelism level of the Parallel Evolutionary Algorithm (PEA) included in a previous methodology ESS. This method is called Evolutionary-Statistical System with Island Model (ESS-IM), because it incorporates the concept of multiple populations or islands, increasing the diversity and contributing to generating more accurate solutions (i.e., predictions that are closer to reality). ESS-IM is a method to reduce uncertainty in the input parameters of the prediction model, in this case it is applied to predict wildfires spreading. However, due to its modular features, this method could be applied to other propagation models such as floods and avalanches.

In the present work we have taken another step forward in the area of *fire safety*, specifically in the field of wildfire behavior prediction, through the ESS-IM uncertainty reduction method. This kind of tools provides significant contribution to the *fire safety* field, since predicting fire behavior may provide wide range of benefits such as: optimization of resources for firefighting, for evacuation measures, for risk analysis of lands depending on the type of vegetation and topography, among others. Since the prediction results should be obtained in the shortest time possible, the method has been developed in a parallel environment. The performance of the method has been compared with a previous methodology and evaluated according to three parameters: (a) prediction quality, (b) improvement of each method, and (c) computational cost. The results, in terms of prediction quality and improvement, are encouraging, as ESS-IM exceeds its predecessor in more than 90% of evaluations conducted, and although, in terms of computational performance, the results observed are slightly lower, they are overlapped by the gain in prediction quality. Because of the favorable results obtained with ESS-IM, it is necessary to continue working with the aim of optimizing the use of computing resources, for example: analyzing the effect of different communication topologies, evaluating those migration strategies that offer better results, analyzing evolutionary parameters tuning, incorporating graphics processing units (GPUs), among others.

Acknowledgements

This work has been supported by UTN under Project EIUTIME0002169TC, by MINCyT under Project PRH PICT-2008-00242 and MEC-Spain under Contract TIN2014-53234-C2-1-R. The first author would like to thank CONICET for the PhD Grant provided.

References

[1] P. Morgan, C. Hardy, T.W. Swetnam, M.G. Rollins, D.G. Long, Mapping fire regimes across time and space: understanding coarse and finescale fire patterns, Int. J. Wildland Fire 10 (2001) 329–342, http://dx.doi.org/10.1071/WF01032.
[2] United Nations Environment Programme, environment for development (UNEP), Satellite Images Record How Wildfires Have Destroyed One Million Hectares of Forests in Western Russia, ⟨http://na.unep.net/geas/getUNEPPageWithArticleIDScript.php?article_id=64⟩ (accessed 09.04.14).
[3] British Broadcasting Corporation (BBC), Russia Wildfires Still Spreading—50 Dead, ⟨http://www.bbc.co.uk/news/world-europe-10881892⟩ (accessed 09.04.14).
[4] Diario Los Andes: Unas 200.000 hectáreas devoró el fuego en General Alvear,

⟨http://losandes.com.ar/article/unas-200.000-hectareas-devoro-fuego-general-alvear-762711⟩ (accessed 10.03.14).
[5] Diario La Nación: Chubut enfrenta el peor incendio forestal que se haya registrado en Argentina, ⟨http://www.lanacion.com.ar/1771568-chubut-enfrenta-el-peor-incendio-forestal-de-la-historia-de-argentina⟩ (accessed 06.05.15).
[6] Diario Clarín: Incendios forestales en Chubut: finalmente, la lluvia trajo alivio, ⟨http://www.clarin.com/sociedad/chubut-incendios-forestales-lluvia-alivio_0_1331267110.html⟩ (accessed 06.05.15).
[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, Arch. Support. Form. Program. Lang. Oper. Syst. (2000) 93–104.
[8] Y.E. Aslan, I. Korpeoglu, Ö. Ulusoy, A framework for use of wireless sensor networks in forest fire detection and monitoring, Comput. Environ. Urban Syst. 36 (2012) 614–625, http://dx.doi.org/10.1016/j.compenvurbsys.2012.03.002.
[9] P.L. Andrews, C.D. Bevins, R.C. Seli, BehavePlus Fire Modeling System, Version 2.0: User's Guide. General Technical Report, RMRSGTR-106WWW, Department of Agriculture, Forest Service, Rocky Mountain, Research Station, Ogden, UT, 2003.
[10] M.A. Finney, FARSITE: Fire Area Simulator-Model Development and Evaluation. Research Paper, RMRS-RP-4, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT, 1998, 47 pp.
[11] G. Wallace, A numerical fire simulation model, Int. J. Wildland Fire 3 (1993) 111–116, http://dx.doi.org/10.1071/WF9930111.
[12] J. Jorba, T. Margalef, E. Luque, J. Campos da Silva, D.X. Viegas, Parallel approach to the simulation of forest fire propagation, in: Proceedings of the 13 International Symposium Informatik fur den Umweltshutz der Gesellshaft Fur Informatik (GI), 1999, pp. 68–81.
[13] A.M.G. Lopes, M.G. Cruz, D.X. Viegas, FireStation—an integrated software system for the numerical simulation of wind field and fire spread on complex topography, Environ. Model. Softw. 17 (2002) 269–285, http://dx.doi.org/10.1016/S1364-8152(01)00072-X.
[14] L.M. Ribeiro, D.X. Viegas, A.G. Lopes, P. Mangana, P. Moura, Operational application of a decision support tool in fire management in Portugal, For. Ecol. Manag. 234 (2006) S243, http://dx.doi.org/10.1016/j.foreco.2006.08.272.
[15] G.L. Ball, D.P. Guertin, FIREMAP fire and the environment: ecological and cultural perspectives, USDA Forest Service, Knoxville, TN 1991, pp. 215–218.
[16] K. Beven, A. Binley, The future of distributed models: model calibration and uncertainty prediction, Hydrol. Process. 6 (1992) 279–298.
[17] J. Piñol, R. Salvador, K. Beven, Model Calibration and Uncertainty Prediction of Fire Spread, Forest Fire Research & Wildland Fire Safety, on CD-ROM, Millpress, 2002.
[18] B. Abdalhaq, A methodology to enhance the prediction of forest fire propagation (Ph.D. thesis), Universitat Autnoma de Barcelona, Spain, 2004.
[19] G. Bianchini, M. Denham, A. Cortés, T. Margalef, E. Luque, Wildland fire growth prediction method based on multiple overlapping solution, J. Comput. Sci. 1 (2010) 229–237, http://dx.doi.org/10.1016/j.jocs.2010.07.005.
[20] G. Bianchini, M. Méndez-Garabetti, P. Caymes Scutari, Evolutionary-Statistical System for Uncertainty Reduction Problems in Wildfires. XVIII CACIC, XII Workshop de Procesamiento Distribuido y Paralelo (WPDP), 2012, pp. 230–238, ISBN: 978-987-1648-34-4.
[21] G. Bianchini, P. Caymes Scutari, M. Méndez-Garabetti, Evolutionary-statistical system: a parallel method for improving forest fire spread prediction, J. Comput. Sci. 6 (2015) 58–66.
[22] M. Méndez-Garabetti, G. Bianchini, M.L. Tardivo, P. Caymes-Scutari, Predicción del comportamiento de incendios forestales mediante aplicación de modelo de islas a ESS, 9no Encuentro Del. Int. Cent. Earth Sci. (2013) 55, ISBN 978-987-1323-34-0.
[23] 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 (Electronic Notes in Theoretical Computer Science) vol. 314, 2015, pp. 45–60. http://dx.doi.org/10.1016/j.entcs.2015.05.004.
[24] F. Outtier C. Ourtier. Data assimilation concepts and methods, Training Course Notes, ECMWF, 1999.
[25] D.C. Montgomery, G.C. Runger, Applied Statistics and Probability for Engineers, fifth ed., Limusa Wiley & Sons, New York, 2011.
[26] R. Buyya, High Performance Cluster Computing: Programming and Applications, Prentice-Hall, New Jersey, 1999.
[27] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evolut. Comput. 6 (2002) 443–462, http://dx.doi.org/10.1109/TEVC.2002.800880.
[28] H. Mühlenbein, Parallel genetic algorithms, population genetics and combinatorial optimization, in: Lecture Notes in Computer Science, Parallelism, Learning, Evolution, vol. 565, 1991, pp. 398–406. http://dx.doi.org/10.1007/3-540-55027-5_23.
[29] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, Sourcebook of Parallel Computing, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.
[30] A. Grama, A. Gupta, G. Karypis, V. Kumar, Introduction to Parallel Computing, second ed., Pearson, New York, 2003.
[31] T. Mattson, B. Sanders, B. Massingill, Patterns for Parallel Programming, Addison-Wesley, Boston, 2005.
[32] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co. Inc., Boston, 1989.
[33] C. Darwin, On the Origin of Species by Means of Natural Selection, Murray, London, 1859.
[34] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975.
[35] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, Berlin Heidelberg, 2003.
[36] E. Cantú Paz, A survey of Parallel Genetic Algorithms, Calc. Parallèles, Reseaux Et. Syst. Repartis 10 (1998) 141–171.
[37] R.C. Rothermel, A Mathematical Model for Predicting Fire Spread in Wildland Fuels, Research Paper INT-115, US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, (Ogden, UT, 1972.
[38] R.C. Rothermel, How to Predict the Spread and Intensity of Forest Fire and Range Fires, General Technical Report, INT-143, US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, 1983.
[39] C.D. Bevins, FireLib User Manual& Technical Reference, ⟨http://www.fire.org⟩, 2004.
[40] P.L. Andrews, BEHAVE: Fire Behavior Prediction and Modeling Systems—Burn Subsystem, Part 1. General Technical Report, INT-194, US Department of Agriculture, Forest Service, Intermountain Research Station, Odgen, UT, 1986.
[41] R. Real, J.M. Vargas, The probabilistic basis of Jaccard's Index of similarity, Syst. Biol. 45 (3) (1996) 380–385.
[42] D.X. Viegas, (Coordinator) et al., Project Spread—Forest Fire Spread Prevention and Mitigation, ⟨http://www.algosystems.gr/spread/⟩, 2004.
[43] P.L. Andrews, BehavePlus Fire Modeling System, Version 5.0: Variables, General Technical Report, RMRS-GTR-213WWW Revised, Department of Agriculture, Forest Service, Rocky Mountain Research Station, Fort Collins, CO, 111 p, ⟨http://www.fs.fed.us/rm/pubs/rmrsgtr_213.pdf⟩, 2009.
[44] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, Source-Book of Parallel Computing, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2003.
[45] W. Gropp, E. Lusk, A. Skjellum, Using MPI—Portable Parallel Programming with the Message-Passing Interface, second ed., The MIT Press, 1999.