Journal of Computational Science 6 (2015) 58–66

Contents lists available at ScienceDirect

Journal of Computational Science

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

Evolutionary-Statistical System: A parallel method for improving forest fire spread prediction

G. Bianchini [a,*], P. Caymes-Scutari [a,b], M. Méndez-Garabetti [a,b]

[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

ARTICLE INFO

Article history:
Received 3 December 2013
Received in revised form
12 December 2014
Accepted 12 December 2014
Available online 16 December 2014

Keywords:
Parallel Evolutionary Algorithm
Statistical System
Forest fire prediction
High performance computing
Parallel processing

ABSTRACT

Fighting fires is a very risky job, where loss of life is a real possibility. Proper training is essential. Several firemen academies offer courses and programs whose goal is to enhance the ability of fire and emergency services to deal more effectively with fire. Among the tools that can be found in the training process are fire simulators, which are used both for training and for the prediction of forest fires. In many cases, the used simulators are based on models that present a series of limitations related to the need for a large number of input parameters. Moreover, such parameters often have some degree of uncertainty due to the impossibility of measuring all of them in real time. Therefore, they have to be estimated from indirect measurements, which negatively impacts on the output of the model. In this paper we present a method which combines Statistical Analysis with Parallel Evolutionary Algorithms to improve the quality of the model output.

1. Introduction

People have considerable fear of fire (history is replete with disastrous losses). However, fires in areas like, for instance, Western United States are natural and they benefit to the plant communities there. Periodic fires help to clear the forest floor of debris and promote the growth of sturdy, fire-resistant trees [33]. Nevertheless, expanding human populations have intruded on previously uninhabited areas, establishing their own communities in fire-prone zones. Moreover, human activities, such as fire suppression, livestock grazing, and logging, have increased the probability of hotter and more destructive fires [35].

Another example is the Mediterranean areas with dry and warm summers that favour the occurrence of fire ignition and propagation. Every year, millions of hectares are burned in Tropical, Boreal and Mediterranean forests [26], which causes a wide variety of effects, from atmospheric emissions [28], to soil erosion, biodiversity loss and drainage alterations [2]. Reduction of those negative effects of fire requires to rely on tools and methods for the assessment of the fire risk.

Tools like simulators, expose to the trainees to a convincing fire propagation model, where instructors can vary fuel types, environmental conditions, and topography. Responding to these variables, trainees may call for appropriate resources and construct firebreaks, and fundamentally, students can review the results of their decisions in the security of a computer laboratory.

Different propagation models have been developed to predict fire behaviour. They can be classified into empirical, semi-empirical, and physical models [11]. The probable fire behaviour is predicted in empirical models from average conditions and accumulated knowledge obtained from laboratory and outdoor experimental fire or from historical fires. Semi-empirical (semi-physical or laboratory models) are those models based on a global energy balance and on the assumption that the energy transferred to the unburned fuel is proportional to the energy released by the combustion of the fuel; one of the most important among these models is the pioneering work of Rothermel [31,32]. Finally, physical (theoretical or analytical) models are based on physical principles and have the potential to accurately predict the parameters of interest over a broader range of input variables than empirically based models do. Any of these models can be used to develop simulators and tools for preventing and fighting forest fires. Some old and current examples are Behave-Plus [3], FARSITE [14], FIREMAP [4], FireStation [21], WRF-Fire [22], XFire [20].

According to Fons [15] the relevant factors that affect the rate of spread and shape of a forest fire front are the fuel type (type of

* Corresponding author. Tel.: +54 261 5244579; fax: +54 261 5244538.
E-mail addresses: gbianchini@frm.utn.edu.ar (G. Bianchini), pcaymesscutari@frm.utn.edu.ar (P. Caymes-Scutari), miguelmendezgarabetti@gmail.com (M. Méndez-Garabetti).

http://dx.doi.org/10.1016/j.jocs.2014.12.001
1877-7503/

vegetation), humidity, wind speed and direction, forest topography (slope and natural barriers), and fuel continuity (vegetation thickness). Therefore, models require a set of input parameters, including vegetation type, moisture contents, wind conditions, and so on, and they provide the evolution of the fireline in the successive simulation steps. However, the result obtained after the direct application of a simulator (known as Classical Prediction and explained in Section 2) usually differs from reality because of the difficulty of providing accurate input values to the model. Given this uncertainty, we propose an alternative method, that tries to determine the possible fire behaviour based on Statistical Analysis [25] and Parallel Evolutionary Algorithms (PEAs) [24].

The method proposed, called Evolutionary-Statistical System (ESS), is based on S^2F^2M [8,10], an already validated method based on statistics and High Performance Computing (HPC). Different from classical predictions, these two methods make use of different techniques in order to calibrate the set of input parameters, and generate predictions based on a lot of possible cases (each case defined by a different set of input parameters), rather than on a single case. This is why they are classified as Data-Driven methods with Multiple Overlapping Solution. On the one hand, S^2F^2M considers the total set of cases (considering a factorial experiment based on certain ranges for each parameter) to carry out the search of the forest fire behaviour. Unlike the methods of unique solution, S^2F^2M does not make a distinction between the best and worst cases, but it determines the statistical trend of the fireline. On the other hand, the new proposed method, ESS, incorporates the PEA component to guide the search of the solution and reduce the number of cases under study in the statistical stage. In place of considering a factorial experiment (as S^2F^2M does), ESS considers just a sample of possible cases to conform a population which is evolved according to the PEA principles. The results obtained from the evolved population are then submitted to the statistical stage. In this way, it is possible to reach better results with the possibility of finding the solution even in less execution time.

The simulation of the spread of forest fires is a challenge from the computational point of view, given the complexity of the models involved, the need for efficient numerical methods and resource management. In this context, the method presented in this paper is an important tool for the prevention and prediction of forest fires, given that it provides a better prediction of the forest fire behaviour. This is a general method which could be applied on different propagation models (e.g. floods, snow avalanches, landslides, etc.), but in this article we present its application to forest fire spread prediction.

In this paper, we describe the direct use of a simulator (known as classical prediction) in Section 2. Section 3 describes the proposed methodology, implemented in a system called Evolutionary-Statistical System (ESS) [7]. In Section 4 we use a set of real cases of forest fires for evaluating ESS in contrast to S^2F^2M; we also comment on the obtained results related to the execution time and the speedup obtained when we work on a cluster computer. Finally, we present the main conclusions.

2. Classical prediction

Classical prediction approach is depicted in Fig. 1. In this scheme, **FS** corresponds to the underlying fire simulator, which is considered as a black box. **RFL0** is the real fireline at time t_0 (initial fire front), whereas **RFL1** corresponds to the real fireline at t_1. If the prediction process works properly, after executing **FS** (which should be fed with the corresponding input parameters and **RFL0**) the predicted fireline at time t_1 (**PFL**) should coincide with the real fireline (**RFL1**).

As we previously mentioned, models require static parameters (e.g. information about topography), parameters that can change

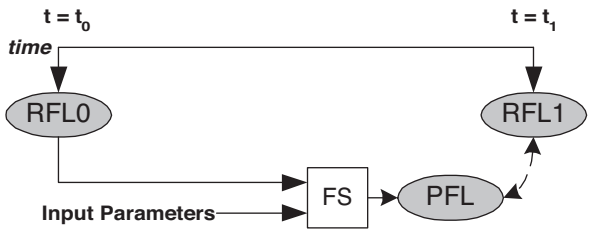

Fig. 1. Diagram of classical prediction of forest fire propagation (FS, Fire Simulator; PFL, Predicted Fireline; RFL*X*, Real Fireline on time *X*).

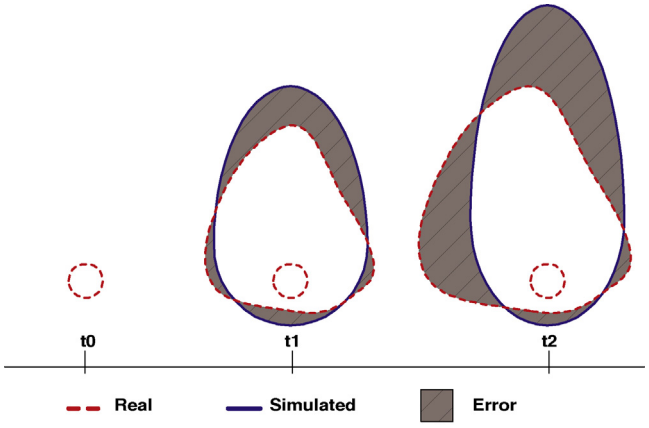

Fig. 2. Error using classical prediction.

very slowly (type of vegetation), parameters that can change frequently (moisture content), and parameters that are completely dynamic (like wind conditions). The precision of these parameters is a very important point in prediction of the behaviour. However, in many cases it is impossible to carry out any type of measurement, particularly in real fire situations, which is a critical situation to feed the simulation model.

Generally, the obtained prediction using this classical approach does not match the reality [5,29]. A simple example can be seen in Fig. 2, where it can be appreciated the error between the real fireline and the predicted fireline. One reason for the discrepancy between real and simulated propagation stems from the difficulty of feeding the model with accurate input values. Uncertainties in the input variables can have a substantial impact on the result errors and should be considered.

In this context, the classical prediction of the fireline behaviour cannot be considered to be reliable for two reasons: on the one hand, the difficulties in making an accurate estimation of the parameters and, on the other hand, the resulting prediction is based on a single simulation, which does not constitute a reasonable basis for making a decision given the uncertainty of the parameters.

3. Evolutionary-Statistical System

In the uncertainty reduction field, we propose a new method which we called Evolutionary-Statistical System (ESS) [7]. Such a method combines the strength of three components: Evolutionary Algorithms (EAs), Statistics and Parallelism. The method generates predictions based on the statistical analysis of a population of cases, in contrast to the single case considered by the classical prediction. The population is constituted by a set of individuals which are also referred as scenarios. Each individual or scenario constitutes a different setting of the input parameters under study (i.e. each individual arises as a combination of the possible values of

input parameters of the model). Given that, in general, each parameter could assume different values belonging to a certain range (for more details see below) the combination of every possible value for every parameter considered leads in a factorial experiment which represent a huge number of scenarios. For efficiency reasons, ESS considers just a sample of such factorial experiment for constituting a population of individuals which is exploited by the evolutionary component. More specifically, in the model of forest fire used in this work, an individual is composed by 9 values corresponding to the 9 parameters used in the forest fire model [31]. These parameters are: fuel model, wind speed, wind direction, slope, aspect (this is degrees clockwise from north, only used for location in a map), 1-h dead fuel moisture (the moisture content of one-hour time lag dead fuels, which are fuels consisting of dead herbaceous plants and roundwood less than about 1/4 in. in diameter), 10-h dead fuel moisture (the moisture content of the 10 h time lag dead fuels, which consist of roundwood 1/4 to 1-inch in diameter and, very roughly, the layer of litter extending from immediately below the surface to 3/4 below the surface), 100-h dead fuel moisture (this value represents the modelled moisture content of dead fuels in the 1–3 in. diameter class) and live herbaceous fuel moisture. Each individual is represented by a vector of numbers in double precision, because to represent these variables is not possible to use binary representation (this structure is the main reason why we talk about an evolutionary system and we do not mention it as a genetic method). It is important to remark that each individual constitutes the input of a particular simulation, and such input values are global to the entire map simulated, i.e. for each simulation the same values are used on each cell in the map.

About the values used for each parameter, these are generated using a uniform distribution within a range bounded by a minimum value and a maximum value. The minimum and maximum values can be mainly obtained from two possible sources: measurements made by sensors, or standard values which, although less accurate, at least allow us to define a threshold to work.

The application of EA aims to work on a population of scenarios where each one could offer useful information to the statistical analysis. In absence of the EA component, it will be necessary to work on a set of scenarios bigger (the complete factorial experiment), and that has the side effect that a certain percentage of cases does not significantly contribute to the global result (either because they are redundant values, or because they are very far from reality) [10].

For his part, the parallel component is the tool to enable the execution of the high number of operations resulting of the other parts of the method.

3.1. Methodology of the Evolutionary-Statistical System

As we previously mentioned, one of the component of ESS are the EAs, which help us to deal with a population of scenarios relevant to the study. We define the concept of scenario as a particular setting of the set of parameters, and in this case, each scenario is represented by an individual in a population of possible solutions. The population is also managed by the statistical component, as we explain in the following.

For each parameter (e.g. wind speed, slope, moisture content, etc.) we define a range, which is used to create random values in a valid interval. These values are used to define the different components of each individual in the population.

Meanwhile, the terrain representation is performed through a grid or a matrix M in which each minimum terrain portion is called a cell. Let suppose n is the total number of scenarios. M constitutes the input for the simulation of each scenario and the output is composed by a matrix M'_s, where s represent a particular scenario and $1 \leq s \leq n$. Considering every M'_s, and for a given time interval, we want to know whether a cell has been statistically burned or not. Remembering that n is the total number of scenarios (or what is the same, the number of individuals in the population) and defining n_A as the number of scenarios in which the cell A was burned, we can calculate the ignition probability of cell A as:

$$P_{ign}(A) = \frac{n_A}{n} \tag{1}$$

This reasoning is applied to every cell in the matrix M. As a consequence, we obtain a matrix M'' as the statistical aggregation of every M'_s. M'' has a value $P_{ign}(X_{i,j})$ associated to each cell $X_{i,j}$ that represents the probability of cell $X_{i,j}$ in M to be statistically caught by the fire. In general terms, we define the probability map with probability P_K as the set of cells whose P_{ign} value is higher than or equal to a certain particular value P_K, where $0 \leq P_K \leq 1$. Once we have obtained the output matrix M'', which includes the aggregation of all the probability maps, the next task consists in comparing the real fire against this matrix, where P_K plays an important role. The objective of such a comparison is to search for a particular value of P_K, 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}). Therefore, the associated map of probability has to accomplish the following condition:

$$\left\{ x : P_{ign}(x) \geq \frac{K_{ign}}{n} \quad \middle| K_{ign} \in N \right\} \tag{2}$$

with n equal to the number of scenarios and $P_{ign}(x)$ varying from K_{ign}/n to 1, i.e. the set of cells (x) which have been burned at least K_{ign} times. In other words, K_{ign} can be seen as the minimum number of times that a cell, in different scenarios, should be reached by fire to accomplish the condition, and this is directly related to P_K (the probability for K_{ign}).

A scheme of ESS is presented in Fig. 3, where different time instants were marked as t_0, t_1 and t_i (being t_i a general time, valid from t_2 to t_m, with m equals to the final time). As can be observed, the system is divided into two general stages: an Optimization Stage (**OS**) that implements the Parallel Evolutionary Algorithm (**PEA** box), and the Calibration Stage (**CS**) that is in charge of the statistical component. **OS** iterates until the population reaches a certain level of quality or until a predetermined number of iterations is reached. For each individual, the simulated map (using **FS**) and the fitness are calculated. Then, every map will be included in the Statistical System (**SS** box). The output of **SS** (a probability map) has a double purpose. On the one hand, the probability maps are used as the input of the **SK** box (Search K_{ign}) to search for the current K_{ign} (the key number used to make a prediction), which will be used at the next prediction time. In this stage, a Fitness Function (**FF**) is used to evaluate the probability map. **FF** takes into account the pattern defined by both fireline shapes (the real fireline for RFL_{i-1} and the simulated one for each P_K value). P_K takes at most n different values (being n the size of population) given that each cell will be burnt at most n times. Therefore, the n comparisons are carried out between the real map defined by RFL_{i-1} and the resulting n maps to find the K_{ign} value that gives better fitness value. On the other hand, the output of **SS** box enters the Fire Prediction box (**FP**). **FP** will be in charge of generating the prediction map taking into account the K_{ign} evaluated at previous time. All this process will be repeated during the execution as the system is fed with new information about the fire situation.

3.2. Parallel evolutionary component

Evolutionary Algorithms (EAs) form a type of heuristic search methods based on a particular algorithmic framework whose main components are the variation operators (mutation and

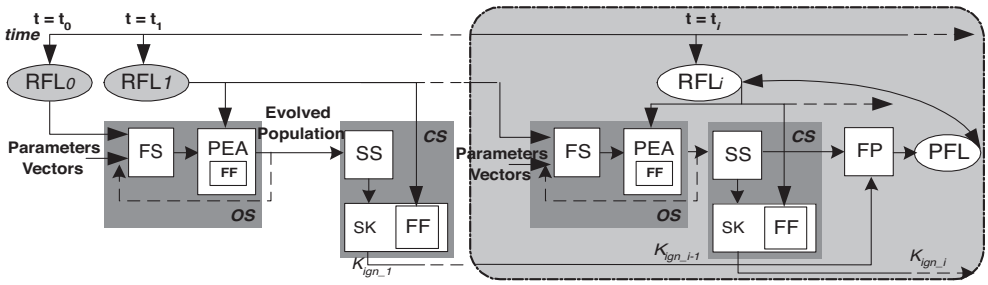

Fig. 3. Diagram of ESS (FS, Fire Simulator; PEA, Parallel Evolutionary Algorithm; OS, Optimization Stage; SS, Statistical System; SK, Search K_{ign}; FF, Fitness Function; CS, Calibration Stage; FP, Fire Prediction; PFL, Predicted Fireline, RFL*X*, Real Fireline on time *X*).

recombination) and the selection operators (parent selection and survivor selection) [24]. On the one hand, the recombination is analogous to reproduction and biological crossover. Recombination is a process of taking more than one parent solutions and producing a child solution from them, with a certain probability between a minimum and maximum values. In the case of EAs, this broadens their search spectrum. On the other hand, mutation is a genetic operator used to maintain genetic diversity from one generation of a population to the next. In this case, this allows a deeper search. Mutation occurs during evolution according to a probability value defined by the user. This probability should be set low. If it is set too high, the search will turn into a primitive random search. The general EA framework is shown in Fig. 4. EAs mimic the concept of natural biological evolution: they operate on a population of potential solutions applying the principle of survival of the fittest [16]. In each iteration EAs update the population through a process of selecting individuals according to the level of fitness for the problem domain (through the fitness function that quantifies this feature) and perform the recombination and mutation of them using operators that mimic natural genetics. Normally, it is expected that this process is leading to the evolution in the population of individuals that have best adapted (or acceptably adapted) to the environment just as happens in natural adaptation.

Given the use of evolutionary algorithms in optimization problems, where they have found very good results [1], we propose the application of this methodology in combination with statistical methods.

In our particular case, the individuals (parents) are selected on the basis of their fitness using the roulette wheel selection. The probability of an individual to be selected increases or decreases as its fitness is greater or smaller than its competitors' fitness. The technique of recombination used is one-point crossover [13], with the cross point randomly defined. Therefore, the two children are composed by one part of each parent. It is important to note the following: The crossover point sets the values of one or more parameters of a father in the first child, and obtains the remaining values from the other parent. The second son is complementary to the first one.

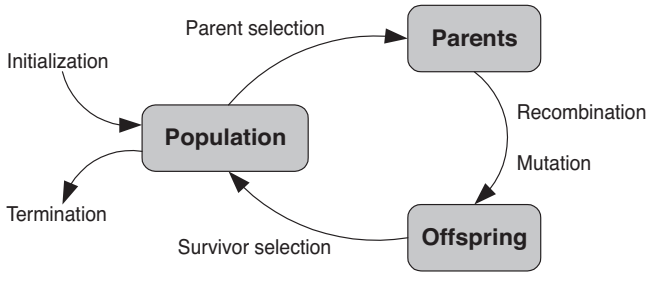

Fig. 4. General framework of an evolutionary algorithm.

Regarding the mutation operator, because the genome is conformed by integer and float genes, the mutation is randomly carried out by changing one randomly selected gene [13]. Of course, when changing the value of the selected gene, the mutation considers the valid domain defined a priori for such a gene.

To evaluate the system response we have used the Jaccard-Index [19,30] as fitness function. We have conceptually partitioned the forest field of the experiments in square cells since the simulator uses an approximation based on cells. The following equation shows the expression:

$$Fitness = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

where *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. This correction in the set of cells through the elimination of the subset of burned cells is used to avoid skewed results.

According to this similarity measurement, a fitness value equal to one corresponds to the perfect prediction because it means that the predicted area is equals to the real burned area. On the other hand, a fitness value equal to zero indicates the maximum error: the experiment did not coincide with reality at all.

The execution of the EA may finalize, for example, after a certain number of generations or when the quality of the population reaches a certain threshold, i.e. when a set of individuals reaches a certain fitness value in an acceptable range for the problem. In such a case, we can stop the evolution of the population though no particular individual has reached the ideal solution.

Evolutionary algorithms are a powerful tool for solving different kinds of problems [27]. However, sometimes this type of methodology iterates for a long time and does not converge or converges to a local optimum. This is one of the reasons why it is interesting to combine the use of evolutionary methods with parallel computing. Furthermore, because of EAs work on a population of individuals, the search for a solution can be performed in parallel, thus providing a number of potential solutions instead of one per unit of time. This scheme is known as Parallel Evolutionary Algorithms (PEAs). According to the amount of populations involved in the algorithm, the treatment and the operators, PEAs can be classified in three broad groups: Single Population with Parallel Evaluation, Single Population with Parallel Application of Operators, and Multiple Populations with Migration. In this work, we consider the first group [12].

In each generation the evaluation of each individual in the population is carried out in parallel. The evaluation of each individual involves not only the simulation corresponding to that individual but also its respective fitness evaluation. After that, multiple individuals are stochastically selected from the current population (depending on their fitness), and they are modified (by

Table 1
Parameter setting for the evolutionary component.

Parameter	Value
Population size	200
Parents	80%
Max. prob. recombination	0.6
Min. prob. recombination	0.2
Probability of mutation	0.5
Max. iterations	200

recombination or by random mutation) to form a new population. The fitness is defined in terms of the genetic representation and measures of quality of the solution represented, as we previously defined in Eq. (3).

3.3. Implementation of ESS

ESS has as goal the parameter optimization to predict real fire behaviour, so that the prediction can be useful in deciding which actions need to be taken in fighting the fire.

Although heuristic optimization techniques like EAs may reduce the time of search, we could still make the execution time faster by applying HPC. Therefore, the abovementioned method has been parallelized in order to reduce the execution time.

The ESS method has been implemented in a system that incorporates a simulation kernel and applies a methodology to evaluate the fitness function. The system has been developed for a PC Linux cluster, using MPI [18] as the message passing library.

With respect to the setting of parameters of the evolutionary component, the values used have been those listed in Table 1, in which the term 'Max. Prob.' refers to the maximum probability, while 'Min. Prob.' to the minimum probability.

3.3.1. The simulator

ESS uses the wildland simulator (fireSim) proposed by Bevins as a simulation core, which is based on the fireLib library [6]. fireLib is derived directly from BEHAVE [4] fire behaviour algorithms for predicting fire spread in two dimensions, but it is optimized for highly iterative applications such as cell –or wave– based fire growth simulation. In particular, this simulator uses a cell automata approach to evaluate fire spread. The terrain is divided into square cells and a neighbourhood relationship is used to evaluate whether a cell will be burned and at what time the fire will reach those cells.

As inputs, fireSim accepts maps of the terrain, vegetation and wind characteristics and the initial ignition map. The output generated by the simulator consists of a map of the terrain in which each cell is labelled with its ignition time.

3.3.2. Parallel version

Because the method uses a sequential simulator as a kernel (fireSim) and commonly it is necessary to do an important number of iterations related to the evolution of the population and the amount of individuals in the population, the method is forced to perform a large number of calculations giving as a result a very time consuming method.

A solution to obtain a more appropriate efficiency has been the use of multiple computational resources working in parallel. Therefore, we applied the Master-Worker paradigm [17,23]. This paradigm is a natural option to this problem because each Worker process requires different data and produces results from its input data without any need for results from other Worker processes (see Fig. 5). In each generation of the evolution the Master successively distributes an individual per Worker; the simulation of the model and the evaluation of fitness function are applied over each individual (tasks carried out by the Workers), returning the results to the Master. This process is repeated until every individual in the population is treated. Finally the Master evolves the population, aggregates the partial results in the statistical step and makes the prediction for each time step.

4. Experimental results

This section shows the results obtained after applying the method on a set of real fire situations. The values obtained from ESS were compared with those found after applying another method already validated (S^2F^2M [8,10]) on the same experiments. Such a method, has been previously contrasted against different methodologies showing good results [8,9].

For the purpose of comparing ESS and S^2F^2M we have used five cases of controlled burns. They were carried out in the field, particularly in a hill of Serra de Lousã (Gestosa, Portugal). The burns were part of the SPREAD project [34]. These experiments were very useful to collect experimental data, to support the development of new

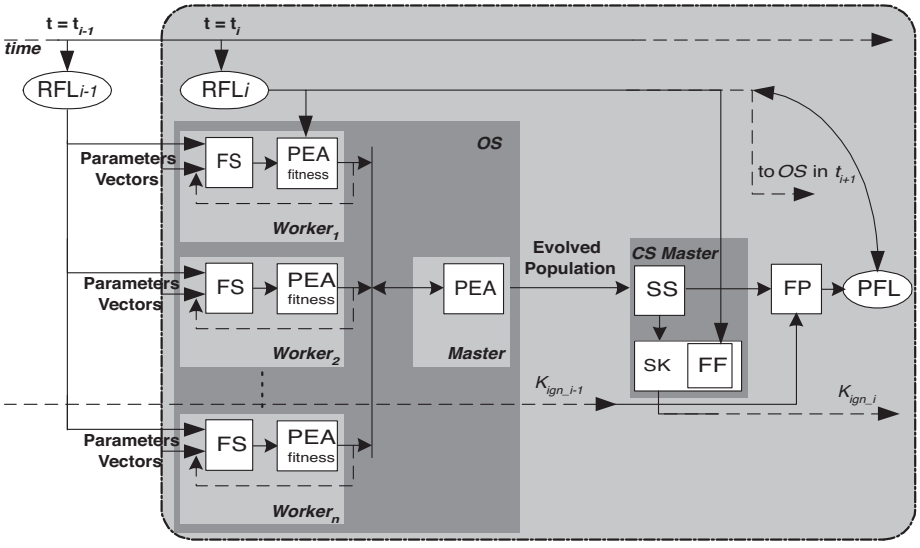

Fig. 5. Diagram of parallel ESS (FS, Fire Simulator; PEA, Parallel Evolutionary Algorithm; OS, Optimization Stage; SS, Statistical System; SK, Search Kign; FF, Fitness Function; CS, Calibration Stage; FP, Fire Prediction; PFL, Predicted Fireline, RFLX, Real Fireline on time *X*).

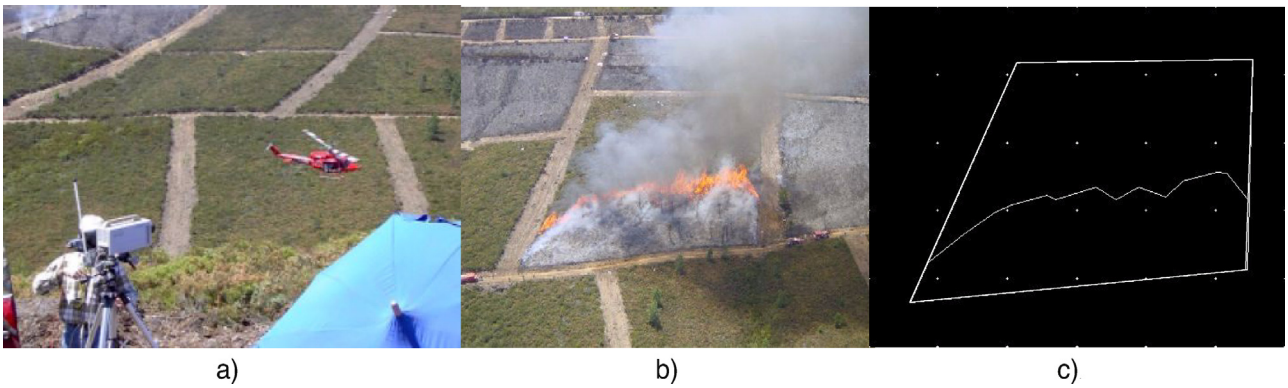

Fig. 6. (a) Obtaining data from forest fire through video cameras. (b) Image capture. (c) Conversion to graphic format.

concepts and models, and to validate existing methods or models in various fields of fire management. We have not included the results of the classical method because the values obtained are low and do not contribute information to this work. In addition, previous studies have shown that the values obtained by applying the statistical method S^2F^2M, surpasses the quality of the prediction achieved by the classic approach [9].

Along the progress of burning, discrete steps were defined to represent the progress of the fire front. Therefore, we consider various time instants t_0, t_1, t_2, ... In Table 2 can be appreciated the characteristics (size and slope) of the terrain used for each experiment. In order to gather as much information as possible about the fire-spread behaviour, a camera recorded the complete evolution of the controlled fires. The videos obtained were analyzed and several images were extracted every certain period of time. From the images, the corresponding fire contours were obtained and converted into a suitable format so they could be interpreted by the methods. A simple scheme of this process can be seen in Fig. 6.

In experiments 1, 2 and 3 the cell size was 1 m^2, and in experiments 4 and 5 the cell size was 0.1112 m^2. The remaining parameters such as wind conditions and moisture content were variable.

4.1. Experiment 1

In accordance to the information already known about the experiment and the models of Rothermel [31] for some of the parameters, certain ranges have been specified (in particular those parameters that exhibit uncertainty). Some part of this information has been measured during the experiment, and the remainder has been taken from standard values used by BehavePlus [3].

In order to be able to compare both prediction methods, we need to set an initial time (t_0) and a certain time step. These values have been fixed to minute 0 (for initial time) and 2 min for time step. Then, predictions are carried out in the minutes 4, 6, 8 and 10, because, in the minute 2, the method is still working in the calibration stage (K_{ign} calculation).

This is a case with linear ignition on the right border. After the application of each method, we obtained the fitness values shown in Fig. 7.

Table 2
Dimensions and slopes of the plots used in the experiments.

Experiment	Width (m)	Length (m)	Slope (°)
1	58	50	21
2	89	91	21
3	95	123	21
4	20	30	6
5	20	30	6

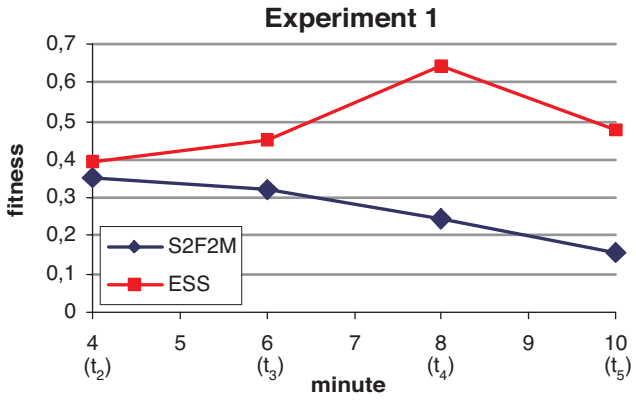

Fig. 7. Comparison between the fitness obtained for each method for the Experiment 1.

4.2. Experiment 2

This is another case where the plot was burned by linear ignition (in this case, on the left border). For this experiment, the intervals for comparison have been defined each 2.5 min. The initial time has been fixed at minute 2.5 and final time has been fixed at minute 12.5. After the application of each method, we obtained the fitness values shown in Fig. 8. We can observe that in both cases the initial prediction reached is not high (0.4 around), but this value is improving towards the end, mainly for ESS (near to 0.8).

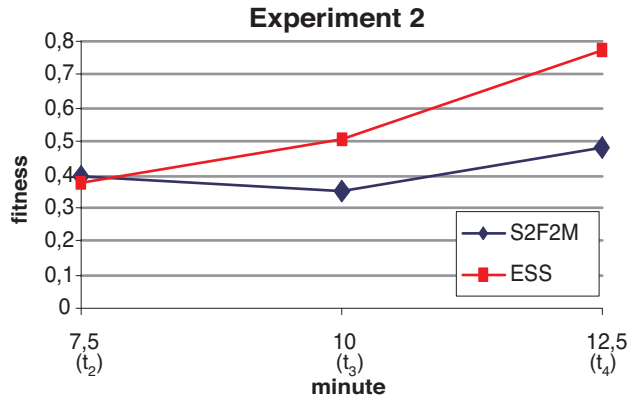

Fig. 8. Comparison between the fitness obtained for each method for the Experiment 2.

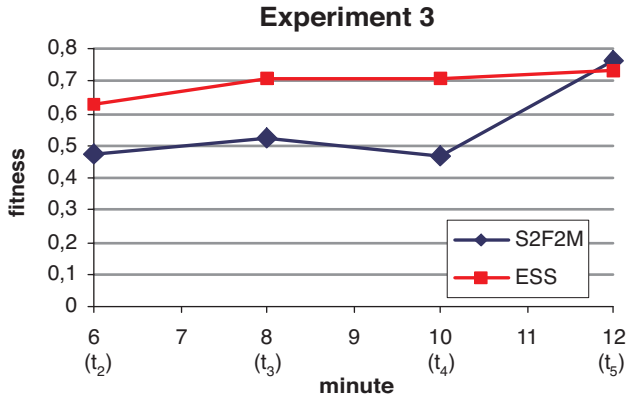

Fig. 9. Comparison between the fitness obtained for each method for the Experiment 3.

4.3. Experiment 3

The Experiment 3 belongs to cases where the fire was originated in a single point. After execution of the methods, the comparison of fitness values found are shown in Fig. 9. We can see that ESS performs better compared to S^2F^2M. However, at certain times (for instance at minute 12), the values found may be similar or even slightly lower than the results achieved by the S^2F^2M method. One possible explanation is a substantial difference between the two methods: although both share the notion of statistical analysis of the results they produce, S^2F^2M is an exhaustive approach that covers the whole range of values for the parameters with uncertainty, which means it analyzes 'all' combinations (obviously considering for each parameter its domain and the discrete step to vary the value along such a domain). On the other hand, ESS uses EAs with small populations, which has proven to be an excellent heuristic, but it does not guarantee finding the ideal values. Therefore, sometimes S^2F^2M can offer higher fitness values (as in the above cases), although overall, ESS provides better results in average.

4.4. Experiment 4

The fourth experiment has a reduced plot size. For this reason we defined a small cell size to increase the number of cells in the plot and in consequence the accuracy. In this case, the experiment belongs to cases of fires started on the left border of the field through pyrotechnic devices in a linear way.

After execution of the methods, the fitness values found are shown in Fig. 10. As we can see, the duration of this experiment was

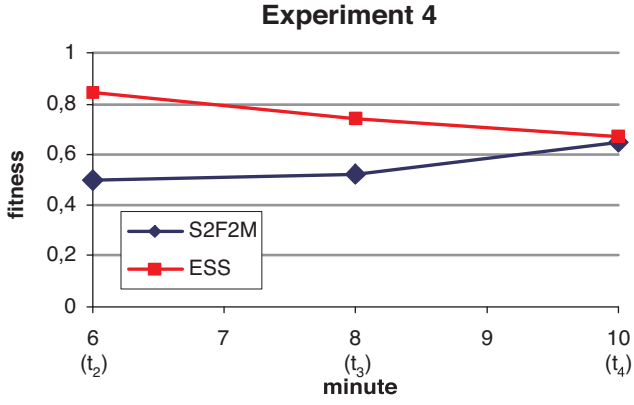

Fig. 10. Comparison between the fitness obtained for the Experiment 4.

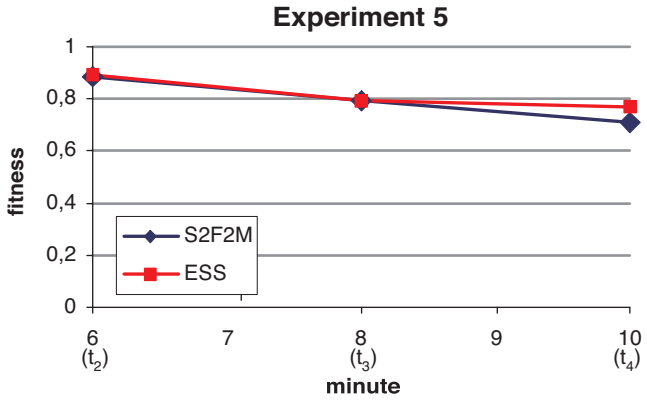

Fig. 11. Comparison between the fitness obtained for the Experiment 5.

very short, possibly due to the wind effect: with a high wind speed, the ratio of spread (ROS) and the flame intensity can become very high. The combination of these factors produces a fast propagation and, therefore, a more dangerous fire.

4.5. Experiment 5

The last experiment, like Experiment 4, has a reduced plot size. For this reason, once again, we defined a small cell size to increase the number of cells in the plot. We defined the dimensions of cells as in previous case (the cell size was set as 0.1112 m^2). In this case, the plot was burned by linear ignition at bottom using pyrotechnic devices, setting the t_0 in minute 2 and time step in 2 min. The fitness values obtained can be observed in Fig. 11.

4.6. Parallelism and speedup

The results were obtained by executing both systems (ESS and S^2F^2M) on a LINUX cluster composed by 16 Intel® Core™ 2 Duo Processor E4600 2.4 GHz, under an MPI environment [18]. Fig. 12 shows the speedup obtained as an average of all experiments.

The straight line represents the ideal speedup. As we can observe, both methods have a speedup relatively good (S^2F^2M a bit better than ESS). For the purposes of a fair comparison, in both cases were performed the same number of simulations. Thus, in addition to the graph, the execution times are also similar (ESS takes on average 10% less execution time). However, ESS may take even less time because the number of iterations depends on when it finds individuals who meet the expected fitness, and this usually happens before in ESS than in S^2F^2M (because S^2F^2M is a deterministic and exhaustive method into the defined ranges, while ESS is a stochastic one). Moreover, the execution time is tied to the expected quality (in the method, given by the threshold of fitness determined by the user), as for instance happens for Experiment 4, where ESS can take around 35 min to find individuals with fitness equal to 0.85, or it can spend 140 min looking for individuals with fitness equal to 0.95. In conclusion, there is a trade-off between time and quality. It depends on the user to set certain parameters to emphasize either the time constraint or the expected quality.

As we commented in Section 3, the parallel version of the method has been designed using the Master-Worker paradigm, due to the data can be divided into independent parts, which gives us no interaction between the Worker processes. This kind of distribution may be one part of the explanation of the speedup factor found. However, we can also see that the associated speedup value for 32 processors has decreased. For this point the Master process has started become a bottleneck because it concentrates different functionalities of the method. Related to this, we are currently working

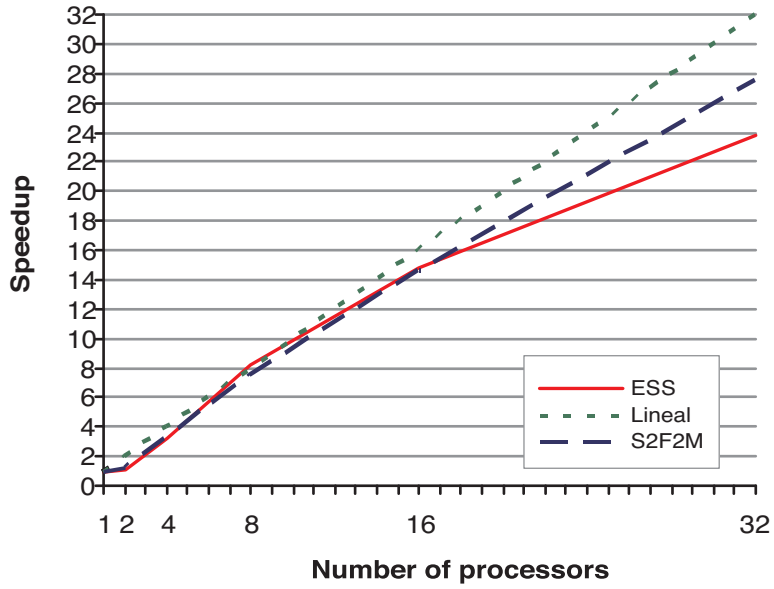

Fig. 12. Speedup against number of processors.

on a new version of ESS which applies the concept of islands in the evolutionary algorithm with the idea to remedy this problem by working with independent populations in order to free the Master process of the management of the whole problem.

Moreover, we started the tuning process of ESS, both for adjust its quality of prediction and its efficiency. On the one hand, we are considering a static tuning for the improvement of the parameters of the method; on the other hand, we consider dynamic and automatic tuning with the aim of adapting the underlying parallel system as the needs for computational power vary with the evolution of the fire and the load of the execution environment.

5. Conclusions

The techniques that combine high performance computing with statistical methods have excellent ability to solve or reduce the problem of uncertainty in input parameters, as in this case it is the prediction of forest fire behaviour. For this reason, it is of great interest the ongoing research on this subject, so as to optimize and evolve on the approaches and methods already developed to maximize the results achieved. We combined the power of the statistical calculation with capabilities provided by Parallel Evolutionary Algorithms, achieving results that improve the original methodology (S^2F^2M) based solely on statistical calculation and high performance computing. Then, from S^2F^2M we have arrived at the concept of Evolutionary-Statistical System (ESS).

Given the costs, risks and obvious difficulties for design multiple fires in real plots to obtain reliable data for experimentation and validation of the methods, the experiments were conducted on five real fires considering different instants of time in each case. In general we could see that the results of predictions are better using ESS. In addition, one important feature is that the method is general enough to be used on different models (floods, avalanches, etc.).

As we commented, in this first approach of ESS, we decided to apply parallelism only in the evaluation of the individuals, with the goal of gradually increase the degree of parallelism to compare the results offered by each alternative of PEAs. Currently, we are tuning the ESS method both to static and dynamic levels, in order to improve the quality of prediction and the quality of the execution in terms of computational resources usage. We are also working in a new version called ESS-IM (Evolutionary-Statistical System with Island Model) which in preliminary experiments is achieving better results than ESS. In addition, we have planned to apply the method on a scheme configured with GPUs to exploit the high level of parallelism offered by this kind of technology.

Acknowledgments

Financial support was provided by UTN under projects UTN1194 and EIUTIME0002169TC, by CONICET under project PIP 11220090100709 and by FONCyT (ANPCyT) under project PRH PICT-2008-00242. M. Méndez-Garabetti was supported by CONICET through an Internal Doctoral Grant.

References

[1] B. Abdalhaq, G. Bianchini, A. Cortés, T. Margalef, E. Luque, Between classical and ideal: enhancing wildland fire prediction using cluster computing, J. Clust. Comput. 9 (3) (2006) 329–343 (special issue on cluster computing in science and engineering).
[2] F.J. Ahern, J.G. Goldammer, C.O. Justice (Eds.), Global and Regional Vegetation Fire Monitoring from Space: Planning a Coordinated International Effort, SPB Academic Publishing, The Hague, The Netherlands, 2001.
[3] P.L. Andrews, C.D. Bevins, R.C. Seli, BehavePlus Fire Modeling System, Version 2.0: User's Guide. Gen. Tech. Rep. RMRS-GTR-106WWW, Dept. of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT, 2003, pp. 132.
[4] G.L. Ball, D.P. Guertin, FIREMAP, in: S.C. Nodvin, T.A. Waldrop (Eds.), Fire and the Environment: Ecological and Cultural Perspectives, Proceedings of an International Symposium. General Technical Report SE-69, Knoxville, TN, USDA Forest Service, Southeastern Forest Experiment Station, Asheville, NC, 1991, pp. 215–218.
[5] K. Beven, A. Binley, The future of distributed models: model calibration and uncertainty prediction, Hydrol. Process 6 (1992) 279–298.
[6] C.B. Bevins, FireLib User Manual & Technical Reference, 2004 http://www.fire.org (accessed May 2012).
[7] G. Bianchini, P. Caymes Scutari, Uncertainty reduction method based on statistics and parallel evolutionary algorithms, in: Proceedings of High-Performance Computing Symposium – 40 JAIIO (HPC 2011, ISSN:1851-9326), 2011, pp. 1–4.
[8] G. Bianchini, M. Denham, A. Cortés, T. Margalef, E. E. Luque, Wildland fire growth prediction method based on multiple overlapping solution, J. Comput. Sci. 1 (2010) 229–237.
[9] G. Bianchini, A. Cortés, T. Margalef, E. Luque, Improved prediction methods for wildfires using high performance computing: a comparison, LNCS 3991 (Part I) (2006) 539–546.
[10] G. Bianchini, Wildland fire prediction based on statistical analysis of multiple solutions (Ph.D. thesis), Universitat Autònoma de Barcelona, Spain, 2006.
[11] L. Bodrožić, C. Stipanicev, M. Šeric, Forest fires spread modeling using cellular automata approach, in: V. Hladky, J. Paralic, J. Vaščak (Eds.), Modern Trends in Control, Equilibria, Košice, Slovakia, 2006, pp. 23–33.
[12] E. Cantú Paz, A survey of parallel genetic algorithms, Calc. Parallèl Reseaux Syst. Repart. 10 (2) (1998) 141–171.

[13] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, 2003, pp. 37–70 (Chapter 3).
[14] M.A. Finney, FARSITE: Fire Area Simulator-model Development and Evaluation. Res. Pap. RMRS-RP-4, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT, 1998, pp. 47.
[15] W. Fons, Analysis of fire spread in light forest fuels, J. Agric. Res. 72 (1946) 93–121.
[16] D.E. Goldberg, Genetic and evolutionary algorithms. Come of age, Commun. ACM 37 (3) (1994) 113–119.
[17] A. Grama, A. Gupta, G. Karypis, V. Kumar, Introduction to Parallel Computing, second ed., Pearson, 2003.
[18] W. Gropp, E. Lusk, A. Skjellum, Using MPI – Portable Parallel Programming with the Message-Passing Interface, second ed., The MIT Press, 1999.
[19] P. Jaccard, Lois de distribution florale, Bull. Soc. Vaudoise Sci. Nat. 38 (1902) 67–130.
[20] J. Jorba, T. Margalef, E. Luque, J. Campos da Silva, D.X. Viegas, Parallel approach to the simulation of forest fire propagation, in: Proc. 13 Inter. Symposium "Informatik fur den Umweltshutz" der Gesellshaft Fur Informatik (GI), 1999, pp. 68–81.
[21] 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 (3) (2002) 269–285.
[22] J. Mandel, J.D. Beezley, A.K. Kochanski, Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011, Geosci. Model Dev. 4 (2011) 591–610.
[23] T. Mattson, B. Sanders, B. Massingill, Patterns for Parallel Programming, Addison-Wesley, 2005.
[24] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, third ed., Springer, 1996.
[25] D.C. Montgomery, G.C. Runger, Probabilidad y Estadística aplicada a la Ingeniería, second ed., Limusa Wiley, 2011.
[26] P. Morgan, C.C. Hardy, T.W. Swetnam, M.G. Rollins, D.G. Long, Mapping fire regimes across time and space: understanding coarse and fine-scale fire patterns, Int. J. Wildland Fire 10 (2001) 329–342.
[27] K.M. Nelson, Applications of Evolutionary Algorithms in Mechanical Engineering, 1997 http://digitool.fcla.edu/dtl_publish/34/12514.html (accessed May 2012).
[28] A. Palacios-Orueta, A. Parra, E. Chuvieco, C. Carmona, Remote sensing and geographic information system methods for global spatiotemporal modelling of biomass burning emissions: assessment in the African continent, J. Geophys. Res. Atmos. 109 (D14S07) (2004) 1–12.
[29] J. Piñol, R. Salvador, K. Beven, Model Calibration and Uncertainty Prediction of Fire Spread, 2002, ISBN: 90-77017-72-0(2002)99-111.
[30] R. Real, J.M. Vargas, The probabilistic basis of Jaccard's Index of similarity, Syst. Biol. 45 (3) (1996) 380–385.
[31] R.C. Rothermel, A mathematical model for predicting fire spread in wildland fuels, Res. Pap. INT-115, US Dept. of Agric., Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, 1972.
[32] R.C. Rothermel, How to predict the spread and intensity of forest fire and range fires. Gen. Tech. Rep. INT-143, US Dept. of Agric., Forest Service, Intermountain Forest and Range Experiment Station, Ogden, UT, 1983.
[33] A.B. Shiflet, G.W. Shiflet, Introduction to Computational Science – Modelling and Simulation for the Sciences, Princeton University Press, 2003.
[34] Spread Project – Forest Fire Spread Prevention and Mitigation. http://www.algosystems.gr/spread/ (accessed October 2013).
[35] Wilderness Society The Ecology of Wildland Fire. http://www.wilderness.org/OurIssues/Wildfire/ecology.cfm?TopLevel-=Ecology (accessed January 2006).

Germán Bianchini was awarded his BSc in Computer Science from Universidad Nacional del Comahue, Argentina, in 2002. In 2004 and 2006, he got the MSc and PhD in Computer Science from Universitat Autònoma de Barcelona (UAB), respectively. His main research interests are focused on parallel and distributed computing and, in particular, how to take advantage of the computational power provided for parallel and distributed systems to enhance different kinds of problems (wildland fire, floods, etc.). Since 2008, he has been professor of the course in "Parallel Computing" at the Universidad Tecnológica Nacional – Facultad Regional Mendoza (UTN-FRM), where he is also leading the Research Laboratory in Parallel/Distributed Computing (LICPaD).

Paola Caymes Scutari received the BSc in Computer Science from Universidad Nacional de Río Cuarto, Argentina, in 2002. In 2005 and 2007, she got the MSc and PhD in Computer Science from Universitat Autònoma de Barcelona (UAB), respectively. Currently, she is the co-director of the LICPaD (Research Laboratory in Parallel/Distributed Computing, Universidad Tecnológica Nacional – Facultad Regional Mendoza, Argentina) where she is working as a CONICET researcher and as a professor of the course in "Parallel Computing". Her main research interest is focused on high performance computing, parallel/distributed applications, evolutionary computing, performance analysis and performance tuning of parallel applications.

Miguel Méndez Garabetti was awarded his BSc in Telecommunication Engineering from Universidad del Aconcagua, Argentina, in 2011. In 2013, he got the MSc in Information Technology from Universidad de León, España. Currently, he is a PhD student of Computer Science at Universidad Nacional de San Luis, Argentina. His current research is related to improving wildfire prediction using high-performance computing, metaheuristics and bioinspired algorithms. Also he is a professor at the Instituto Tecnológico Universitario at Universidad Nacional de Cuyo, and he is a member of the Research Laboratory in Parallel and Distributed Computing (LICPaD) at the Department of Information Systems Engineering, Universidad Tecnológica Nacional, Facultad Regional Mendoza.