Evolutionary-Statistical System for Uncertainty Reduction Problems in Wildfires

Germán Bianchini, Miguel Mendez-Garabetti and Paola Caymes-Scutari

Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD) Departamento de Ingeniería en Sistemas de Información, Facultad Regional Mendoza - Universidad Tecnológica Nacional. (M5502AJE) Mendoza, Argentina
gbianchini@frm.utn.edu.ar, miguelmendezgarabetti@gmail.com, pcaymesscutari@frm.utn.edu.ar

Abstract. Fire modelling is used by engineers and scientists to understand and to predict possible fire behaviour. Empirical, semi-empirical, and physical models have been developed to predict wildfire behaviour. Any of these can be used to develop simulators and tools for preventing and fighting wildfires. However, in many cases the models 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 getting all of them in real time. Consequently, these values have to be estimated from indirect measurements, which negatively impacts on the output of the model. In this paper we show a method which takes advantage of the computational power provided by High Performance Computing to improve the quality of the output of the model. This method combines Statistical Analysis with Parallel Evolutionary Algorithms. Besides, we compare this method with a previous version which did not use evolutionary algorithms.

1 Introduction

According to Fons [1] the relevant factors that affect the rate of spread and shape of a wildfire front are the fuel type (type of vegetation), humidity, wind speed and direction, forest topography (slope and natural barriers), and fuel continuity (vegetation thickness). Hence, models require a set of input parameters, including vegetation type, moisture contents, wind conditions, and so on, to provide the evolution of the fire line in the successive simulation steps.

Numerous fire spread models have been proposed following several methods. They can be grouped into empirical, semi-empirical, and physical models [2]. 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 [3], [4]. Finally, physical (theoretical or analytical) models are based on physical principles. All these models can be used to develop simulators and tools for preventing and fighting wildfires. Some examples are Behave-Plus [5], FARSITE [6], FIREMAP [7], FireStation [8], WRF-Fire [9], XFire [10], etc.

Considering this kind of tools, the direct application of a simulator offers a result that usually differs from reality. This happens due to 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 [11] and Parallel Evolutionary Algorithms (PEAs) [12] as optimization method. This method corresponds to an improvement of a previous methodology based on Statistical Analysis and High Performance Computing, which has been modified by the combination with Evolutionary Algorithms to improve the prediction level and reduce the execution time. Furthermore, it is a general method which could be applied on different propagation models (e.g. floods, snow avalanches, landslides, etc.), but here we only present its application to wildfire prediction.

The remaining sections of this paper are organized as follow: we describe the predecessor of the current method, called Statistical System for Forest Fire Management or S^2F^2M [13], [14], in section 2; section 3 describes the basic concepts of Parallel Evolutionary Algorithms and the new methodology, implemented in a system called Evolutionary-Statistical System (ESS) [15]. In section 4 we compare both methods using a set of real cases of wildfires. Section 5 shows the results obtained, related to the execution time and the speedup obtained when we work on a cluster computer. Finally, we present the main conclusions and future work in section 6.

2 Statistical System for Forest Fire Management

The Statistical System for Forest Fire Management (S^2F^2M) [13], [14], [16], works considering a large number of scenarios, where a certain setting of the input parameters values of the simulator defines a particular scenario. S^2F^2M finds out a pattern of behaviour of the model without performing a specific analysis of each scenario. All the possible scenarios are discretely generated considering a certain domain by a factorial experiment [11] and the model is evaluated with each set of values. The evaluation of the model considering each scenario outputs a ignition map based on cells. The results (i.e. the ignition maps) are combined to determine the trend in the behaviour of the model, adjusting to the current observation of it. The pattern found is then taken to predict the next step.

This method requires a large number of operations, and therefore is very time demanding. For this reason, we applied a parallel computing scheme for its implementation to reduce the execution time. Given that the processing of the scenarios is not depending on each other, we applied the Master-Worker paradigm [17], [18]: a main processor calculates each combination of parameters and send them to a set of Workers. These Workers carry out the simulations in parallel and return the partial results to the Master, which aggregates all these individual results in each iteration. This aggregation constitutes the Statistical Stage. Also, the Master process is responsible for the remaining prediction technique, which involves the use of a key value (K_{ign}) obtained in the previous time step and the generation of the K_{ign} for the next time step. The K_{ign} value establishes the ignition probability of the cells to be considered in the statistical map that allow for the best approximation of the real fireline. For more details, see [13] and [14].

Fig.1 shows a general diagram of the complete system. In general, for every *i* from 2 to *n*, both the prediction operation for time t_i and the calibration stage to obtain the K_{ign} to be used in time t_{i+1} will overlap at time t_i. As can be observed, the output generated by the **SS** box (Statistical Stage) is used for a double purpose. On the one hand, the probability maps are used as an input of the **SK** box (Search K_{ign}) to search for the current K_{ign}, which will be used at the next prediction time. In this stage, a Fitness Function (**FF**) is used to evaluate the probability map. On the other hand, the output of **SS** box enters the Fire Prediction box (**FP**), which will be in charge of generating the prediction map taking into account the K_{ign} evaluated at previous time. This process will be repeated during the execution as the system is fed with new information about the fire situation. The time period that goes from t_0 to t_1 is a particular case: the process of prediction needs a calibration stage at the beginning to firstly obtain a K_{ign} value to start up the prediction chain.

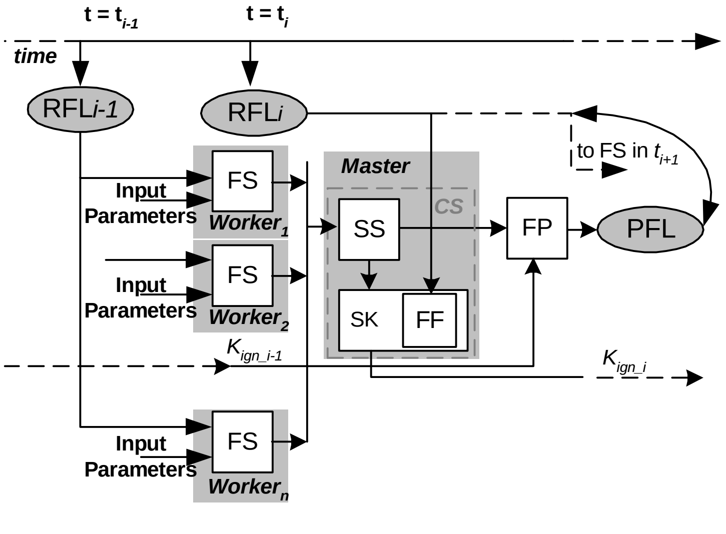

Fig. 1. Detailed diagram of S^2F^2M (FS: Fire Simulator; CS: Calibration Stage; SS: Statistical Stage; SK: Search K_{ign} Stage; FF: Fitness Function; FP: Fire Prediction; PFL: Predicted Fire Line; RFL*X*: Real Fire Line at time *X*)

3 Parallel Evolutionary Algorithms

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 recombination) and the selection operators (parent selection and survivor selection). The general EA framework is shown in Fig. 2. EAs mimic the concept of natural biological evolution: they operate on a population of potential solutions applying the principle of survival of the fittest [19]. In each iteration EAs create a new set of approaches 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 a recombination 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.

The execution of the PEA may finalize, for example, after a certain number of generations or when the quality of the population reaches a certain threshold.

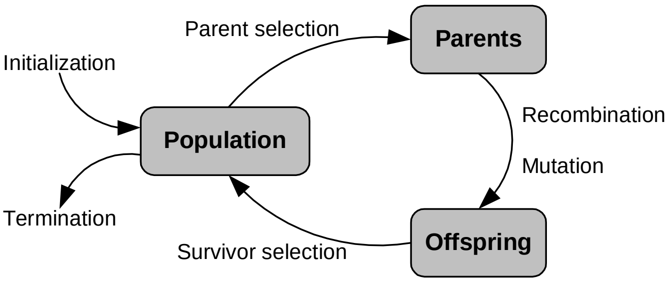

Fig. 2. General framework of an evolutionary algorithm.

Evolutionary algorithms are a powerful tool for solving different kinds of problems [20]. 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 combine the use of evolutionary methods with parallel computing. Furthermore, because of EAs work on population of individuals, the search for a solution can be performed in parallel, thus providing a number of potential solutions instead of one. 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 [21].

In each generation the fitness of each individual in the population is evaluated in parallel. Multiple individuals are stochastically selected from the current population (depending on their fitness), and they are modified (by 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.

Given the use of evolutionary algorithms in optimization problems, where they have found very good results [22], we propose the application of this methodology in combination with statistical methods, as discussed in the following subsection.

3.1 Evolutionary Statistical System (ESS)

The modification and improvement of the statistical method discussed in section 2 has resulted in a new method that combines the strength of four components: uncertainty reduction, statistics, evolutionary algorithms and parallelism, which is why the new method has been called Evolutionary Statistical System (ESS) [15]. The improvement of the S^2F^2M method is related to the introduction of features of PEAs in the calibration step. As we seen, the statistical phase of the methodology includes all the results of a series of cases that arise as some combination of the possible resulting values (within valid ranges) of the parameters that exhibit uncertainty. Clearly, there is a certain percentage of cases that do not contribute significant values to the global result, whether they are now redundant, or because they are too far from reality (and thus, could be considered as negative cases that ultimately degrade the result provided by the method). To solve this problem is that we have decided to apply PEAs.

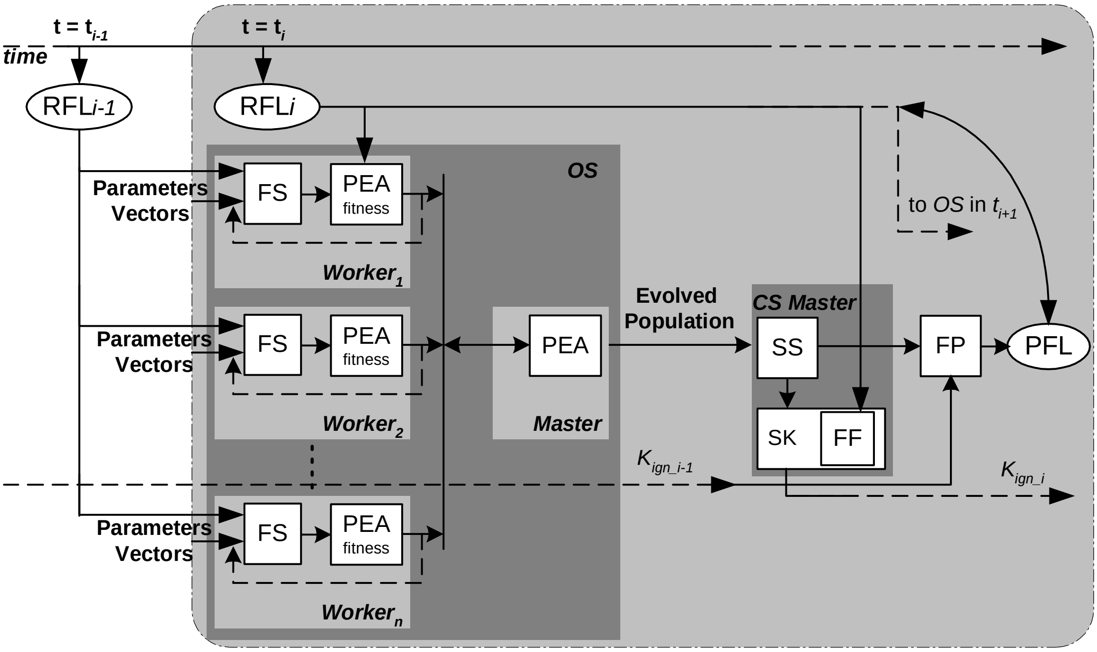

Fig. 3. Diagram of ESS (FS: Fire Simulator; PEA: Parallel Evolutionary Algorithm; OS: Optimization stage; SS: Statistical System; SK: Search K_{ign}; FF: Fitness Function; CS: Calibration stage; FP: Fire Prediction; PFL: Predicted Fire Line, RFL*X*: Real Fire Line on time *X*)

As can be observed in Fig. 3, the system is divided in 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 part of the method. **OS** iterates until the population reaches a certain level of quality. For each individual **FS** and the fitness are calculated in parallel. This is why there are two kinds of **PEA** boxes: the architecture of the ESS is based on the Master-Worker paradigm [17], [18]. In each iteration, the Master 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 **PEA** Master evolves the population. These aggregated results constitute the input of the Statistical System (**SS** box). Similarly to S^2F^2M, 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} (a 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. 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 The fitness function

It is necessary to define a criterion for evaluating the individuals and to compare the prediction resulting from each method with the real situation. For this purpose we have defined a fitness function. Since the simulator uses an approximation based on cells, the fitness function is defined as a quotient over the cells of the prediction map obtained from the parameters of each individual. Equation (1) shows the expression:

$$Fitness = \frac{\#\,cells\,\mathrm{I}\;-\#\,ignitionCells}{\#\,cells\,\mathrm{Y}-\#\,ignitionCells} \qquad \mathbf{(1)}$$

where *#cells*∩ represents the number of cells in the intersection between the simulation results and the real map, *#cells*U is the number of cells in the union of the simulation results and the real situation, and *#ignitionCells* represents the number of burned cells before starting the simulation.

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

4 Experimentation

The main goal of the performed experiments has been to check the effectiveness and to compare the proposed method in order to estimate the performance gain achieved by the introduction of PEAs in the ESS.

The proposed methods have been studied using four cases of controlled burns. They were made in the field, particularly in a hill of Serra de Lousã (Gestosa, Portugal). Fig. 4 shows three specific moments of the experiment 4 and also shows the delineation of the plots under study. The burns were part of the SPREAD project [23]. These experiments have been very useful to collect experimental data, to support the development of new concepts and models, and to validate existing methods or models in various fields of fire management.

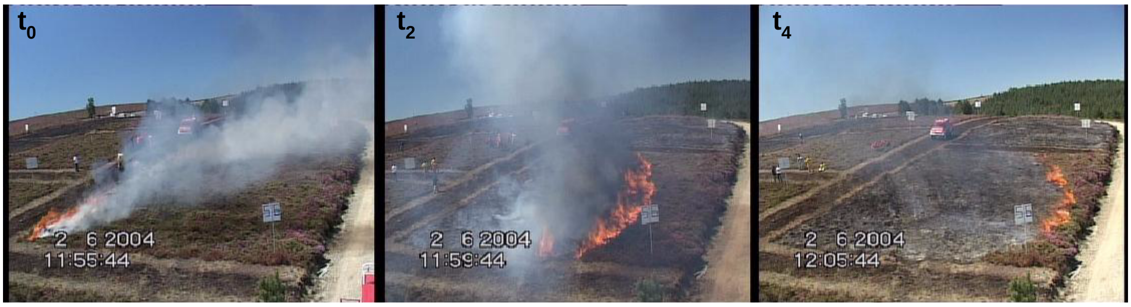

Fig. 4. Different times in a real fire during the burns in the area of Gestosa (Portugal).

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..., etc. In Table 1 can be appreciated the characteristics (size and slope) of the land 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 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. In experiments 1 and 3 the cell size was 1 m^2, and in experiments 2 and 4 the cell size was 0.1108 m^2. The remaining parameters such as wind conditions and moisture content were variable.

Table 1. Dimensions and slopes of the four plots used in experiments.

Experiment	Width in meters	Length in meters	Slope in degrees
1	58	50	21
2	20	30	6
2	89	91	21
4	20	30	6

4.1 Experiment 1

In accordance to the information already known about the experiment and the models of Rothermel [4] 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 [5].

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 minutes for time step.

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. 5.

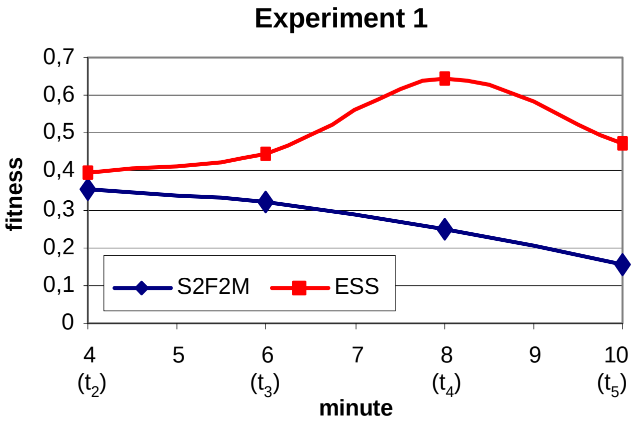

Fig. 5. Comparison between the fitness obtained for each method for the experiment 1.

4.2 Experiment 2

The second 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. The duration of this experiment was very short, possibly due to the wind effect: with a high wind speed, the ROS (ratio of spread) and the flame intensity can become very high. The combination of these factors produces a fast propagation and, therefore, a more dangerous fire. 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 minutes. The fitness values obtained can be observed in Fig. 6.

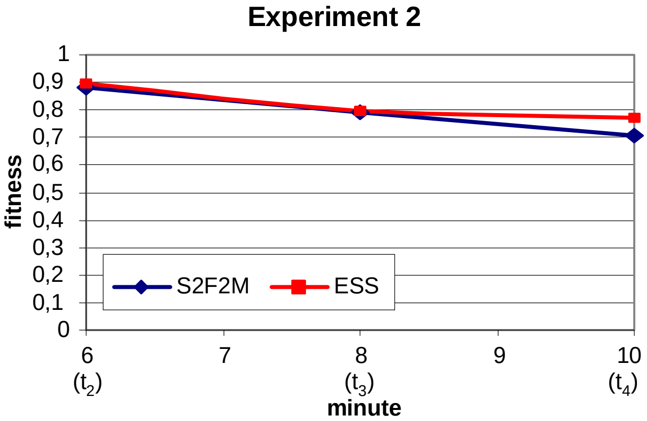

Fig. 6. Comparison between the fitness obtained for each method for the experiment 2.

4.3 Experiment 3

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 as 2.5 minutes. 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. 7. We can observe that in both cases the initial prediction reached is not high, but this value is improving towards the end, mainly for ESS.

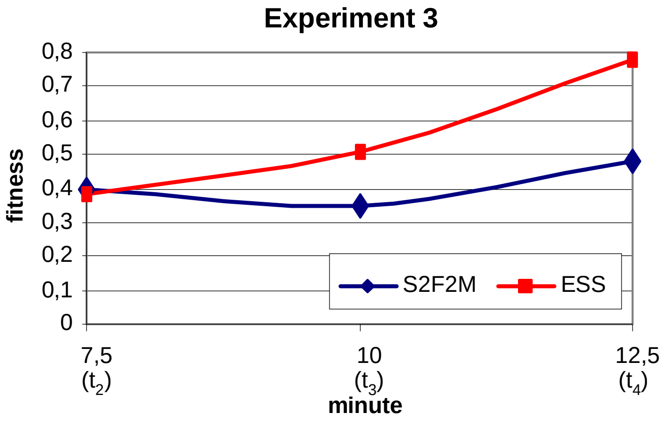

Fig. 7. Comparison between the fitness obtained for each method for the experiment 3.

4.4 Experiment 4

The last experiment, as experiment 2, 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 experiment 2 (the cell size was set as 0.1108 m^2). In this case, the plot was burned by linear ignition on the left border (Fig. 4 shows the time instants t_0, t_2 and t_4, which give an idea both of the dimensions of the plot and the evolution of the fire on it).

After execution of the methods, the fitness values found are shown in Fig. 8.

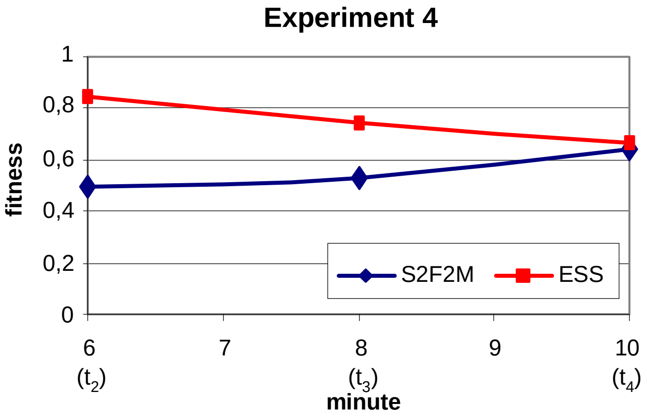

Fig. 8. Comparison between the fitness obtained for each method for the experiment 4.

It is important to highlight that, due to S^2F^2M gives a deterministic output, the shown values are the results of a single execution. In the case of the ESS, as it works with a population of randomly created individuals, the values shown in the figures are the average of ten executions.

5 Performance gain

As we have seen in Sections 2 and 3, both methods operate on a parallel scheme, thus the experimentation results were obtained by executing both systems on a LINUX cluster. The cluster consist of 12 processors AMD64 with 2G RAM and Gigabit Ethernet 1000 Mbps, and working under an MPI environment. The performance gain has been analyzed using the measure known as Speedup [17].

Figure 9 shows the values obtained as an average of all experiments, where the identity line represents the linear case (or ideal Speedup).

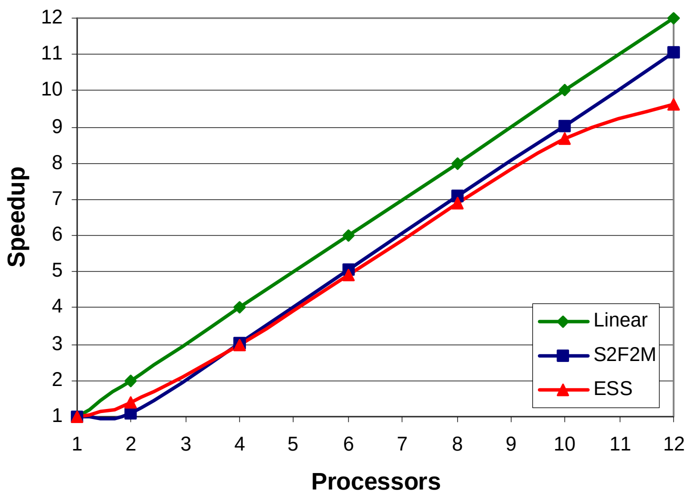

Fig. 9. Speedup for both methods.

Both methods have a speedup relatively good (S^{2}F^{2}M a bit better than ESS, at least between the use of 1 and 10 processors.). Furthermore, the execution times are also similar (ESS takes on average 10% less execution time). However, in actual executions, ESS usually takes even less time because in principle, the number of iterations depends on when it finds individuals who meet the expected fitness, and this usually happens before in ESS that in S^{2}F^{2}M. For example, for Experiment 4, ESS can take around 35 minutes to find individuals with fitness equal to 0.85, or it can spend 140 minutes looking for individuals with fitness equal to 0.95 (in the case of S^{2}F^{2}M, the execution time is 39 minutes). In conclusion, there is a trade-off between time and quality, and depends on the user to configure certain parameters to emphasize either the time restriction or the expected quality.

6 Conclusions and Future Work

A very important part of the land use management is the wildfire management: wildfire models allow us to plan firefighting and fire avoiding strategy in advance. There are several fire model software systems available on the market and the decision on which one to use depends on the area of application and intended use. However, the use of models has certain limitation: they require input data to be accurate, but such data often have some degree of uncertainty. In this paper we proposed a novel approach to the uncertainty reduction problem.

As we have seen, the techniques that combine high performance computing with statistical methods have excellent abilities to solve or reduce the problem of uncertainty in input parameters. 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 actually improve the original methodology S^{2}F^{2}M based solely on statistical calculation and high performance computing. Both methods have been described throughout the present work. They correspond to methods to reduce uncertainty in the input parameters, in this case applied to the prediction of wildfires spread. Furthermore, both methods are general enough to be used on different models. Thus, the combination of evolutionary computation, parallelism and uncertainty reduction is a promising option for tackling various Grand Challenge Problems, as in this case it is the prediction of wildfire behaviour.

The future work will be focused in two different issues related to the improvement of the ESS method. On the one hand, in this paper we presented a first approach of ESS, which applies a scheme of Single Population with Parallel Fitness Evaluation. Our goal is gradually increase the degree of parallelism to compare the results offered by each alternative (i.e. Single Population with Parallel Application of Operators and Multiple Populations with Migration). Each one of the parallelizing methods has some advantages and drawbacks, in general related to the trade-off between the complexity of the parallelization and the performance benefits. On the other hand, further study should focus on the analysis and tuning of the method to obtain the best possible results and compare it with other methods and among them considering their different alternatives.

Acknowledgments. This work has been supported by CONICET under project PIP 11220090100709, by UTN under project UTN1194 and by ANPCyT under project PICT PRH 2008-00242.

References

1. Fons W.: Analysis of fire spread in light forest fuels, J. Agric. Res. 72, 93--121 (1946)
2. Bodrožic, L., Stipanicev, C., Šeric, M.: Forest fires spread modeling using cellular automata approach. Modern trends in control. (eds V. Hladky, J. Paralic, J. Vašcak). pp. 23--33 (Košice, Slovakia: equilibria) (2006)
3. Rothermel, R.C.: 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)
4. Rothermel, R.C.: 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)
5. Andrews, P.L., Bevins, C.D., Seli, R.C.: BehavePlus fire modeling system, version 2.0: User's Guide. Gen. Tech. Rep. RMRS-GTR-106WWW. Ogden, UT: Dept. of Agriculture, Forest Service, Rocky Mountain Research Station. pp. 132 (2003)
6. Finney M.A.: FARSITE: Fire Area Simulator-model development and evaluation. Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. pp. 47 (1998)
7. Ball, G.L., Guertin, D.P.: FIREMAP, in Nodvin, S. C. and Waldrop, T. A., Fire and the Environment: Ecological and Cultural Perspectives: Proceedings of an International Symposium. Knoxville, TN. USDA Forest Service, Southeastern Forest Experiment Station, Asheville, NC. General Technical Report SE-69, pp. 215--218 (1991)
8. Lopes, A.M.G., Cruz, M.G., Viegas, D.X.: FireStation - An integrated software system for the numerical simulation of wind field and fire spread on complex topography. Environmental Modelling & Software, 17 (3), pp. 269--285 (2002)
9. Mandel, J., Beezley, J.D., Kochanski, A.K.: Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE 2011, Geoscientific Model Development (GMD) 4, pp. 591--610 (2011)
10. Jorba, J., Margalef, T., Luque, E., Campos da Silva, J., Viegas, D.X.: Parallel Approach to the Simulation of Forest Fire Propagation. Proc. 13 Inter. Symposium “Informatik fur den Umweltshutz” der Gesellshaft Fur Informatik (GI). pp. 68--81 (1999)
11. Montgomery, D.C., Runger G.C.: Probabilidad y Estadística aplicada a la Ingeniería. Limusa Wiley (2002)
12. Michalewics, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Third, Revised and Extended Edition. Springer. (1999)
13. Bianchini, G., Denham, M., Cortés, A., Margalef, T., Luque, E.: Wildland Fire Growth Prediction Method Based on Multiple Overlapping Solution, Journal of Computational Science, 1 Issue 4, pp. 229--237 (2010)
14. Bianchini, G., Cortés, A., Margalef, T., Luque, E.: Improved prediction methods for Wildfires using High Performance Computing: A comparison, LNCS 3991, Part I, pp. 539--546 (2006)
15. Bianchini, G., Caymes Scutari, P.: Uncertainty Reduction Method Based on Statistics and Parallel Evolutionary Algorithms. Proceedings of High-Performance Computing Symposium - 40 JAIIO (HPC 2011, ISSN: 1851-9326) pp. 1--4 (2011)
16. Bianchini, G., Denham, M., Cortés, A., Margalef, T., Luque E.: Improving forest-fire prediction by applying a statistical approach. Forest Ecology and Management. (234, supplement 1) (2006) S210.
17. Grama A., Gupta A., Karypis G., Kumar V.: Introduction to Parallel Computing. Second Edition. Pearson (2003)
18. Mattson, T., Sanders, B., Massingill, B.: Patterns for Parallel Programming. Addison-Wesley (2005)
19. Goldberg, D.E.: Genetic and evolutionary algorithms. Come of age. Communications of the ACM, 37 (3) pp. 113--119 (1994).
20. Nelson, K.M.: Applications of evolutionary algorithms in mechanical engineering, http://digitool.fcla.edu/dtl_publish/34/12514.html
21. Cantú Paz, E.: A survey of Parallel Genetic Algorithms. Calculateurs Parallèles, Reseaux et Systems Repartis, 10(2), pp. 141--171. (1998)
22. Abdalhaq, B., Bianchini, G., Cortés, A., Margalef, T., Luque, E.: Between Classical and Ideal: Enhancing Wildland Fire Prediction Using Cluster Computing. Journal of Cluster Computing Special Issue on cluster computing in science and engineering. Vol 9, Num 3. pp. 329--343.(2006)
23. Viegas, D.X. (Coordinator) et al., Project Spread - Forest Fire Spread Prevention and Mitigation (2004). http://www.algosystems.gr/spread/