

Comparative Analysis of Performance and Quality of Prediction Between ESS and ESS-IM[★]

Miguel Méndez-Garabetti^{a,b,1}, Germán Bianchini^a,
María Laura Tardivo^{a,b,c} and Paola Caymes-Scutari^{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).*

^c *Departamento de Computación, Universidad Nacional de Río Cuarto.
(X5804BYA) Río Cuarto, Córdoba, Argentina.*

Abstract

Wildfires cause major damage and losses around the world. Such damages range from human and economical losses to environmental ones. Therefore, having models to predict their behavior can be a key element in the process of firefighting. In this paper, we present a comparative study between two methods we have developed. Both methods use Statistical Analysis, Parallel Evolutionary Algorithms and High Performance Computing, respectively named: Evolutionary-Statistical System (ESS) and Evolutionary-Statistical System with Island Model (ESS-IM). In this study, we have compared these two methods in terms of quality of prediction and performance in the parallel environment.

Keywords: Wildfires Prediction, Speed-up, Evolutionary Algorithms, High Performance Computing.

1 Introduction

Fire used appropriately has provided immeasurable benefits to humanity. However, we usually observe instances in which fires have spread out of control burning large areas of vegetation in different continents around the world [22]. This type of phenomenon is known as Wildfire.

[★] This work has been supported by UTN under project EIUTIME0002169TC and by ANPCyT under project PRH PICT-2008-00242. The first author would like to thank CONICET for the PhD grant.

¹ Email: miguelmendezgarabetti@gmail.com



Fig. 1. Voracious fire occurred in Valparaiso, Chile in April 2014.

Unfortunately, human-caused wildfires occur more frequently than those started by natural causes [19]. Furthermore, climatic conditions such as the summer heat benefit its spread, thus increasing danger and hampering fire fighting.

The negative effects of wildfires range from human loss to damage to flora, fauna, air, property and aquatic ecosystems and can all be measured in terms of money as they impact on economy. As an example we can mention the unfortunate recent wildfire in Valparaiso (Chile, April 2014) where the fire burned more than 2,900 homes, causing, at the time this paper, 15 human deaths, about 500 injured people and more than 12,000 homeless and evacuees (Figure 1). That is why, having a model capable of predicting wildfire behavior is of great interest in the process of firefighting, since this will let us distribute fire suppression resources more efficiently, thus reducing losses, damages and costs. Nevertheless, the development of a model for predicting wildfire behavior is affected by a significant drawback called uncertainty. Uncertainty is associated to the variables that intervene and affect fire behavior, i.e. uncertainty appears from the moment difficulties quantifying some variables in real time arise. For example, wind speed and moisture content in vegetation are some of the parameters fire behavior depends on, and often these values cannot be obtained in real time. Therefore, such parameters are represented by estimates based on reference values, indirect measurements or sampling, which reduce the accuracy of the input parameters affecting the quality of prediction of the model.

In every prediction system, the results must be generated before those produced by the phenomenon in reality. Therefore, it is crucial to get the model's output in the shortest time so that predictions can be used to make some kind of preventive decision. Taking into account this requirement and its inherent complexity in any predictive process, it is common to observe a tendency to implement such systems in parallel/distributed environments.

It is important to emphasize that the prediction system response time is a very

relevant feature, as well as the quality of the predictions. In this sense, the use of High Performance Computing [10] (HPC) allows for a great reduction in the processing time. However, this tool should operate with techniques or methods that optimize the model's performance in order to improve the quality of the prediction, i.e., predictions that differ the least possible from reality.

This paper focuses on a comparative analysis (in terms of quality and performance prediction) between two versions of the general uncertainty reduction method called Evolutionary-Statistical System [3,9] (ESS), which can be applied to different phenomena with propagation characteristics. ESS is based on the results of a previous research called Statistical System for Forest Fire Management (S^2F^2M) [7,8]. ESS uses to operate: HPC, Statistical Analysis [20] and Parallel Evolutionary Algorithms (PEAs) [1,11]. In this paper, the method has been applied to wildfires behavior prediction. Having this purpose in mind, we used a fire spread simulator based on Rothmel model [24]. In [4] and [9] ESS obtained better results than S^2F^2M . Besides, in [6] and [7], this method was compared to other methodologies in which it obtained better performance and, consequently, in this paper a comparison is made only between ESS and ESS-IM.

Currently, ESS has two operating versions and one more is under development. In the first implementation of the method, the PEA has used a scheme of Unique Population and Parallel Evaluation [11]. In the second operational version of the method, the level of parallelism of PEA has been increased with a scheme of Multiple Populations and Migration [11,21] called ESS with Island Model [17,18] (ESS-IM). Finally, there is an under development version of ESS that uses Differential Evolution (DE) [23] as an optimization method, called Differential Evolutionary-Statistical System [16] (DESS).

In the next section a brief description and classification of PEAs related to implemented versions of the ESS are provided. Next, a detailed description of the operation mode of ESS and ESS-IM is presented. The details of the experiments performed to carry out the comparative study are also provided, the work environment is described, and then the results are presented. Finally, conclusions and future work are presented.

2 Parallel Evolutionary Algorithms

As mentioned above, both methods in this paper used PEA as an optimizing technique. Therefore, in this section a brief description of the operation of this metaheuristics is presented together with two of its implemented variants.

Evolutionary metaheuristics are optimizing algorithms that use a set of candidate solutions (usually called population) to create new search points within a space of solutions. These methods are often inspired by nature elements, such is the case of Evolutionary Algorithms (EAs), which are based on the natural evolution theory of survival of the fittest [12]. EAs have been successfully applied to solve different optimizing problems in several areas of science.

The general mechanism of EAs is a process of iterations called generations, in

which candidate solutions, known as individuals constitute a population. In each generation, the algorithm applies to the population members the principle of natural selection and survival of the fittest by operators with the aim of approximating to an optimal solution with its elements.

Each individual has a fitness value that represents the quality of the solution. In addition, the EAs must guarantee that individuals with higher fitness values are more likely to reproduce, so they can take part in generating new solutions. This is achieved through the application of the aforementioned operators (crossover, mutation and replacement).

When the problem to be solved with EAs requires considerable execution time (usually because of the complexity of the problem treated), they are usually implemented in a parallel or in a distributed manner. The parallel implementations of EAs can reduce the processing times and also, depending on the treatment of the solutions, increase the search capability of the algorithm. Next, two alternative parallelization of EAs are briefly described: 1) Unique Population and Parallel Evaluation and 2) Multiple Populations and Migration, both were used in ESS and ESS-IM, respectively. These implementations allow us to work with different levels of parallelism and may have different advantages and disadvantages depending on the problem to be dealt with.

2.1 PEA with Unique Population and Parallel Evaluation

In this parallelization model, also known as Master/Worker (as its parallel implementation matches the paradigm Master/Worker [15]), the PEA operates entirely with individuals belonging to a single population in which the individuals' fitness evaluation is carried out in parallel. In each generation, a certain number of individuals is selected based on their fitness value; these are subsequently modified by recombination or mutation, to make up a new population. It should be noticed that the operators are applied according to the entire population.

2.2 PEA with Multiple Populations and Migration

Unlike the previous model, this scheme works with multiple populations and it is often called EA with Island Model, in which each island represents a population of individuals. As the operators (mutation and crossover) are applied among individuals of the same population, this model of parallelization implies that individuals used by operators must belong to the same island or population. Thus, this implementation would be equivalent to running n instances of a PEA with unique population since the islands would evolve independently. In order to avoid this effect, a new operator called "migration" is used in the model of multiple populations. The migration process performs exchange of individuals among different islands in order to add diversity and reduce the probability of premature convergence or stagnation in local optima.

3 ESS and ESS-IM: Operation Description

As we mentioned above, this work corresponds to a comparison of two versions of the Evolutionary Statistical System: ESS and ESS-IM, applied to wildfires behavior prediction. Therefore, the detailed descriptions of each of these versions are discussed in this section.

3.1 ESS: Evolutionary Statistical-System with Unique Population

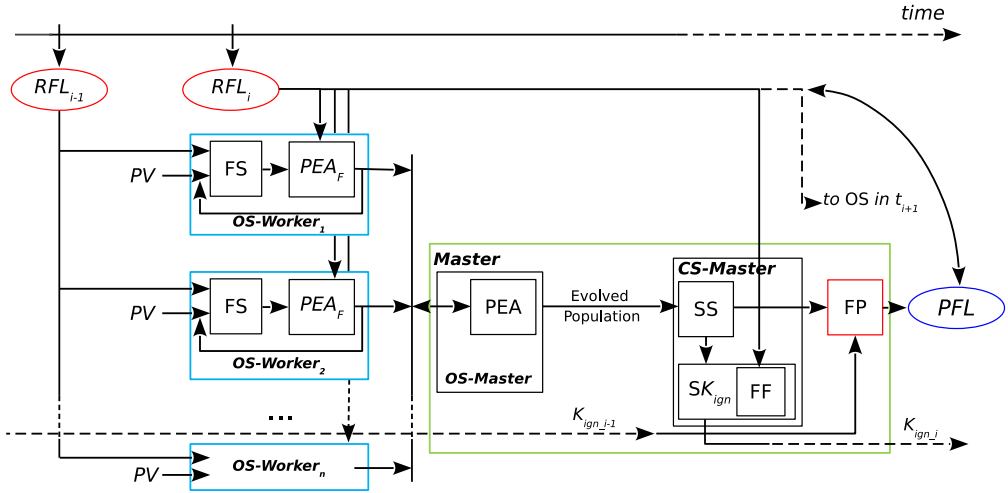


Fig. 2. Evolutionary-Statistical System: **FS**: Fire Simulator; **PEA**: Parallel Evolutionary Algorithm; **PEA_F**: Parallel Evolutionary Algorithm (fitness evaluation); **OS**: Optimization Stage; **SS**: Statistical Stage; **SK**: Search K_{ign} ; K_{ign} : key value used to make the prediction model; **FF**: Fitness Function; **CS**: Calibration Stage; **FP**: Fire Prediction; **PFL**: Predicted Fire Line; **RFL_x**: Real Fire Line on time x ; **SS**: Statistical Stage; **PV**: Parameters Vectors.

ESS is a general method for uncertainty reduction that uses PEAs with a unique population scheme applying parallel evaluation to optimize the search parameters that feed the model. The input parameters in ESS are represented by individuals of a given population. Each individual consists of a set of values that represent each of the input parameters of the model (e.g., moisture content in vegetation, vegetation type, wind direction and speed, the terrain slope, etc.). As can be seen in Figure 2, Optimization Stage (**OS-Worker**) is carried out by the workers nodes. This stage performs the fitness evaluation of individuals through two internal sub-stages called Fire Simulation (**FS**) and Fitness Evaluation of PEA (**PEA_F**). **FS** must be fed with a real fire line at time t_{i-1} (**RFL_{i-1}**) together with the input parameter vector (**PV**). When **FS** finishes the simulation of individuals, the result of each simulation is entered into the stage **PEA_F** to compare the simulated map with the real map at time t_i (**RFL_i**). Here, the fitness value for each individual is calculated considering the difference between the simulated map and the real map. Clearly, the execution time of the method depends on the n number of instances of **OS-Worker** that can be performed in parallel, which actually depends on the number of processing units available. It should also be noticed that all individuals

previously processed by **OS-Worker** must have been sent from the master node to the respective workers. The master node also performs the remaining operations of the PEA (i.e., population generation, individuals selection, crossover, and mutation) and communication functions with each of the workers (tasks performed through the **OS-Master** stage). Once the population reaches a certain level of fitness, it is entered into the Calibration Stage (**CS-Master**). At this stage, the evolved population feeds a sub-stage called Statistics Stage (**SS**, see [7,8]). The output of **SS** is a probability map that is used in the sub-stage **SK_{ign}** (search key ignition **K_{ign}** value) to determine the fire behavior pattern. The value found, **K_{ign}**, is used to make a prediction for the next time instant (t_{i+1}). The fitness evaluation of the probability map is carried out in the stage that implements the Fitness Function (**FF**). In addition, the output of **SK_{ign}** is combined to the probability map provided by **SS** to generate the prediction (**PFL**) at the Fire Prediction stage (**FP**).

As can be seen, ESS is based on a Master/Worker model [15], in which the master process distributes individuals to the workers in each iteration. The workers are responsible for the parallel fitness evaluation of individuals and then for returning the results to the master. After, the whole population is evaluated, the master node evolves the population, includes the partial results and performs the prediction for the next time step.

3.2 ESS-IM: Evolutionary Statistical-System with Island Model

ESS-IM was developed with the aim of improving the quality of prediction of ESS by increasing the diversity of the individuals generated by the evolutionary component of the method. This objective was achieved through the use of multiple populations, using the scheme known as Island Model. Due to the characteristics of the model, ESS-IM has been implemented in a two-level architecture: *L1* and *L2* (it can also be seen as a hierarchical algorithm), such as Figure 3 shows.

In Level 1 (*L1*) the algorithm responds to an evolutionary coarse-grained multi-population algorithm. This level consists of a monitor node responsible for controlling the whole prediction process by communicating with each of the islands. The monitor node carries out the initialization of the islands by sending the necessary information to each of them to perform the simulation in parallel. Meanwhile, the low level *L2* responds to a Master/Worker model as it is composed of a master node (in charge of controlling the operation of the island) and n workers. As in ESS, the workers are responsible for evaluating the fitness of the population. However, in each execution of ESS-IM are executed in parallel j instances of *L2*.

In ESS-IM, master node sends the individuals to the workers, it performs the evolution of the population and it is also responsible for the individuals' migration. The functions performed by the master node of each island are dependent on the configured communication topology. From the performance point of view, this depends on the settings of the predetermined migration parameters. Some of these parameters are: the number of islands, of individuals per island, of individuals to migrate, migration criteria, migration frequency, communication topology (see Table 1). Once all the masters have sent their results, the monitor node carries out

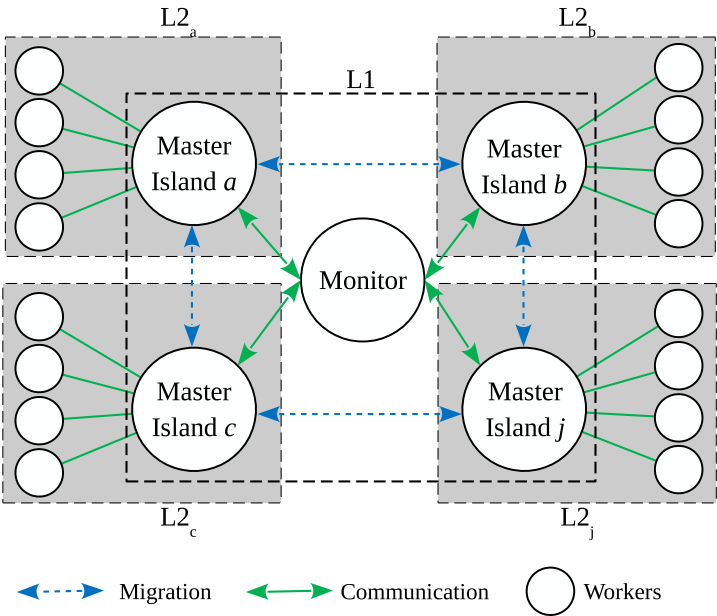


Fig. 3. Hierarchical architecture in ESS with Island Model. Two level communication scheme $L1$ and $L2$.

the prediction of the fire line for each simulation step.

| Parameter | ESS | ESS-IM |
|----------------------------------|-----|--------|
| Threshold Fitness | X | X |
| Number of Iterations | X | X |
| Number of Individuals | X | X |
| Number of Individuals per Island | – | X |
| Number of Islands | – | X |
| Number of Individuals to Migrate | – | X |
| Migration Criteria | – | X |
| Migration Frequency | – | X |
| Number of Workers | X | – |
| Number of Workers per Island | – | X |
| Communication Topology | – | X |

Table 1
Configuration parameters for each method.

Both levels of ESS-IM ($L1$ and $L2$) work with some degree of overlapping in each master. As it may be seen in Figure 4, the Calibration Stage (**CS-Master**) performs communication functions with the monitor node (in $L1$) and the Optimization Stage (**OS-Master**) does the same with the workers (in $L2$). The detailed operation of the two ESS-IM levels can be summarized in two Optimization Stages (**OS-Worker**

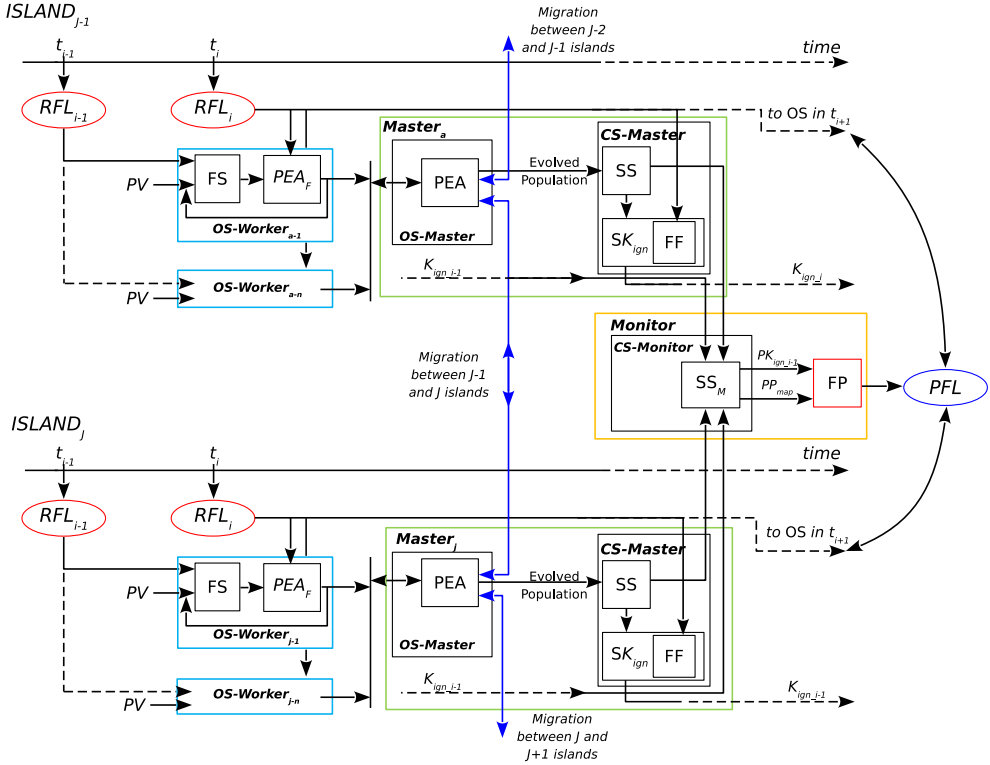


Fig. 4. Evolutionary-Statistical System with Island Model: **FS**: Fire Simulator; **PEA**: Parallel Evolutionary Algorithm; **PEA_F**: Parallel Evolutionary Algorithm (fitness evaluation); **OS**: Optimization Stage; **SS**: Statistical Stage; **SK**: Search K_{ign} ; K_{ign} : key value used to make the prediction model; **FF**: Fitness Function; **CS**: Calibration Stage; **FP**: Fire Prediction; **PFL**: Predicted Fire Line; **RFL_x**: Real Fire Line on time x ; **PV**: Parameters Vectors; **SS_M**: Statistical Stage (monitor); **PP_{map}**: Probability Map

and **OS-Master**), two Calibration Stages (**CS-Master** and **CS-Monitor**) and one Fire Prediction stage (**FP**).

The **OS-Worker** stage performs the evaluation of the population through the fitness function that is part of **PEA** (**PEA_F**). In each island, there is an **OS-Master** stage, which is responsible for carrying out the evolution of the population and performing the individuals' migration (a process that includes: selecting, sending, receiving and replacing individuals). **CS-Master** stage performs the same functions as **ESS**, except that the **SS** output is sent to **CS-Monitor** along with the j K_{ign} values calculated by the j islands. This stage features three operating modes: **CS-Monitor-1**, **CS-Monitor-2** and **CS-Monitor-3**. Finally, the **FP** stage performs the fire line prediction (**PFL**) for each simulation step based on the probability map and the K_{ign} value calculated by **CS-Monitor**. Each of the operating modes has advantages and disadvantages in both quality and performance prediction:

CS-Monitor-1: In this mode, the best K_{ign} value obtained for each of the island is selected. This implies that those islands that have finished with the current simulation step cannot proceed to the following step until all the islands have completed the current step. This scheme prioritizes the quality of prediction of the method by penalizing performance, since there may be times when full island nodes

are idle. It is worth mentioning that each island will be composed of at least two processing units. However, the islands usually use a greater number of nodes, and therefore, the number of idle processors or cores in a cluster can be considerable.

CS-Monitor-2: In this mode, the K_{ign} value is assessed based on a statistical map generated from the resulting statistical maps of each island. This configuration also implies a global synchronization as it is necessary to wait until all islands have sent the results to the monitor node.

CS-Monitor-3: This configuration intends to provide the best possible performance, since the prediction is performed using the K_{ign} value of the island that finishes first (i.e., the value found in the shortest time). In terms of quality of prediction, this modality provides different results since they depend on the characteristics of the island population that finish in less time.

It is important to note that both ESS and ESS-IM have different initialization parameters that determine the operation of the method, both as in quality prediction and as execution time. Each of these parameters are listed in Table 1, where “x” or “–” means that a parameter is present or not in the method. As can be observed the majority of these are related to the PEA. Although this work does not analyze in detail the influence of each of these parameters on the method behavior, in [2] work it was carried out the tuning of the parameters of ESS, and in future researches the effects of these parameters in ESS-IM will be analyzed.

4 Experiments Design

The experimentation has been designed to assess the quality of performance and prediction achieved when executing the two methods on a parallel environment.

4.1 Quality of Prediction Assessment

The quality of prediction of ESS and ESS-IM has been compared by using three real case studies corresponding to controlled fires conducted in different lands located in Portugal (more precisely in Serra Lousã, Gestosa), all of them belonging to the SPREAD project [25]. Each test case has certain duration in minutes, a slope expressed in degrees and an area in square meters. The total duration of each wildfire has been divided into smaller time periods called simulation steps; Table 2 shows the detailed information for each test case.

| Case | Width (m) | Length (m) | Area (m^2) | Slope (d) | Initial Time | Increase | End Time |
|------|-----------|------------|----------------|-----------|--------------|----------|----------|
| 1 | 60 | 90 | 5.400 | 6 | 2,0 | 2,0 | 10,0 |
| 2 | 89 | 109 | 9.701 | 21 | 2,0 | 2,0 | 14,0 |
| 3 | 95 | 123 | 11.685 | 21 | 4,0 | 2,0 | 12,0 |

Table 2
Experiments: size, slope and details of every simulation step.

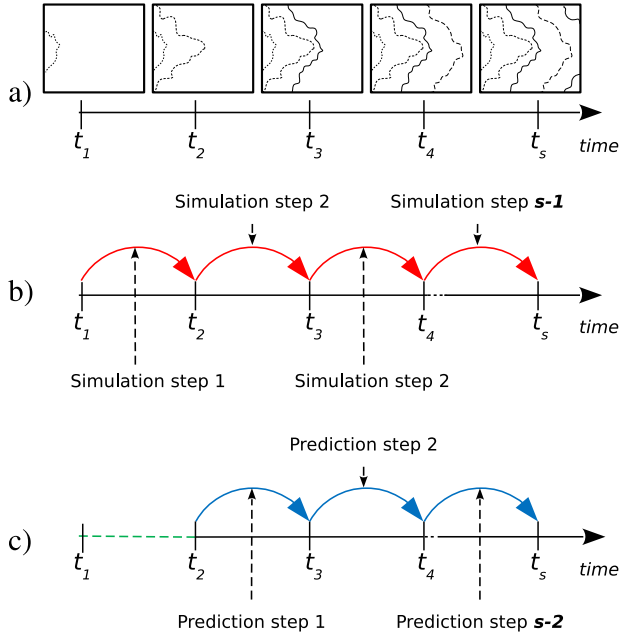


Fig. 5. a) Wildfire progression representation, b) simulation steps and c) prediction.

4.1.1 Simulation and Prediction Steps

It is important to highlight the difference between a simulation step and a prediction step: for that, see Figure 5. Here, we can see that, for a wildfire of specific duration, there exist t_s representative samples of the fireline progression, therefore, $s-1$ simulation steps and $s-2$ prediction steps. This is because these methods require a calibration phase of input parameters, thus they cannot make predictions instantaneously (i.e., during the first simulation step ranging between t_1 and t_2). The first simulation step is used to perform the initial calibration of the input parameters. After carrying out this calibration, the system is able to make the first prediction for the second simulation step, that is equivalent to the first prediction step (i.e., between t_2 and t_3). Therefore, for each test case, $s-2$ quality of prediction values are obtained, one for each prediction step. Thus, comparison of quality is carried out in each prediction step of the experiments.

4.1.2 Fitness Function for Quality of Prediction Assessment

The quality of prediction of each method is assessed by means of a fitness function which quantifies the difference between the resulting map of the prediction and the real map. Such function is based on cells as the maps involved in both methods work with this type of representation. It is defined as follows:

$$(1) \quad \text{Fitness} = \frac{(\#Cells \cap - \#BurnedCells)}{(\#Cells \cup - \#BurnedCells)}$$

in which $\#Cells \cap$ represents the number of cells in the intersection between the simulated map and the real map, $\#Cells \cup$ is the number of cells present in the

union of the resulting map of the simulation and the real maps, and $\#BurnedCells$ represents the number of burned cells before starting the simulation (i.e., corresponding to the initial state of the wildfire in each prediction step). The resulting value of (1) will be between 0 and 1, in which a value equal to 1 corresponds to a perfect prediction, since it means that the predicted area is equal to a real burned area. Moreover, a fitness value equal to 0 would indicate the maximum error because the prediction is completely different from reality.

4.2 Performance Assessment

Regarding performance assessment of both methods, these have been analyzed using the measure known as Speed-Up [13], which 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 in a parallel/distributed environment with x identical processing elements.

Equation (2), shows how to calculate speed-up, in which $t(1)$ represents the serial execution time and $t(N)$ the parallel execution time with N processing units.

$$(2) \quad S = \frac{t(1)}{t(N)}$$

Both methods were executed on a Linux cluster with 32 processing units, under the MPI [14] and Gigabit Ethernet environment.

4.3 Execution of the Experiments

As mentioned earlier, ESS and ESS-IM focus its operation in parallel evolutionary algorithms, this implies that both methods have non-deterministic behavior, because the individuals of each population are generated using stochastically generated seeds. Therefore, in order to make a fair comparison between both methods, a set of 30 different seeds were used. That is to say, the three cases of real fires were executed 30 times each for each method. The experiments were performed in this way with the aim of determining an average of results for each experiment and each method.

5 Results

This section shows the results obtained after evaluating both methods, in terms of quality of prediction as well as performance aspects.

5.1 Quality of Prediction Analysis

The results, in terms of quality of prediction, are shown in Figures 6, 7 and 8 for experiments 1, 2 and 3 respectively. It is important to remember that the depicted results in the above mentioned figures show the average of 30 runs. These were carried out with random seeds for each experiment, but with the same initial seeds for both methods, so that the comparison results are as fair as possible.

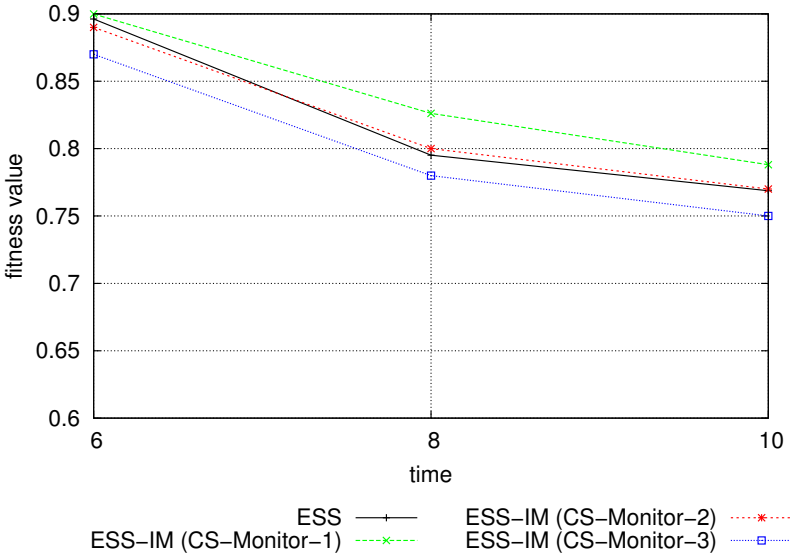


Fig. 6. Experiment 1. Prediction quality analysis.

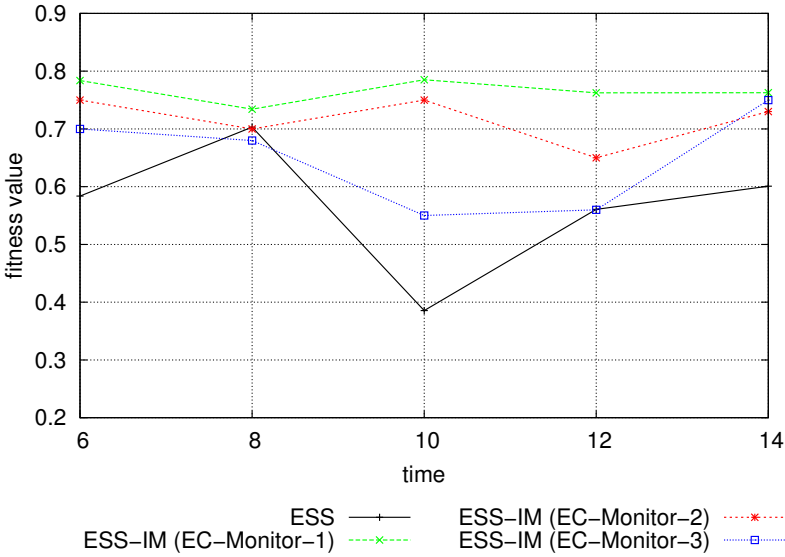


Fig. 7. Experiment 2. Prediction quality analysis.

In each of these figures, axis x represents the progress time of the fire according to the prediction steps (see Figure 5) and the axis y , represents the fitness value achieved for each prediction step as mentioned above in Section 4.1.1.

In general terms, we can see that ESS-IM, mainly in the **CS-Monitor-1** mode, surpasses the results provided by ESS, with some exceptions: in experiment 1 (Figure 6), ESS surpasses ESS-IM in the first prediction step, in **CS-Monitor-2** and **CS-Monitor-3** modes. In the second step, it only surpasses **CS-Monitor-3**, like in the last prediction step. In the second experiment (Figure 7), ESS obtains bet-

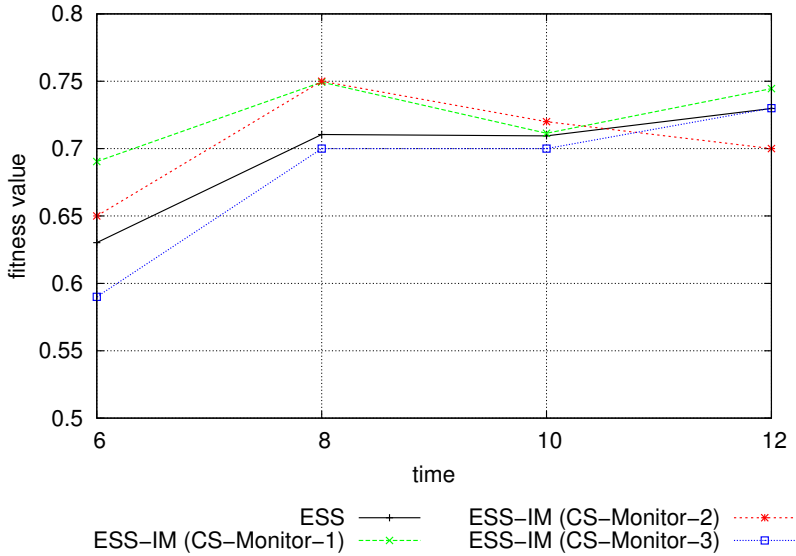


Fig. 8. Experiment 3. Prediction quality analysis.

ter quality of prediction than ESS-IM in **CS-Monitor-3** mode, only in the second prediction step. And finally, in the third experiment (Figure 8), ESS gets better results than ESS-IM (in **CS-Monitor-3** mode) throughout the whole experiment. In the 10th minute, ESS gets the same performance as **CS-Monitor-1** and the second best performance in the last prediction step.

5.2 Performance Analysis

The Figure 9 shows the result of speed-up analysis of each methodology described above. The graph corresponds to the use of: 1, 2, 4, 8, 16 and 32 processing units in the simulation of a particular case. To draw up each curves of speed-up, the same numbers of simulations were performed considering the average value for each case.

In general terms, we can see that all the methods show very good result up to the use of 16 processing units. From there on, two modes of ESS-IM (**CS-Monitor-2** and **CS-Monitor-3**) continue with the same trend, showing very good performance, close to the ideal speed-up, in contrast with ESS-IM in **CS-Monitor-1** mode and ESS. Although the performance benefit obtained by the latter mentioned method decreases when working with 32 processes, it is an important factor to consider according to the results obtained in quality of prediction by these methods. We can say that ESS-IM in **CS-Monitor-1** mode and ESS offer a “very good” performance, since the quality of prediction levels are higher than the others. Besides, by using a larger number of processing units, it was possible to reduce the execution time. Evidently we must make some effort to try to improve the use of resources of these methods, thus, making a better use of them, in other words, obtaining performance gains.

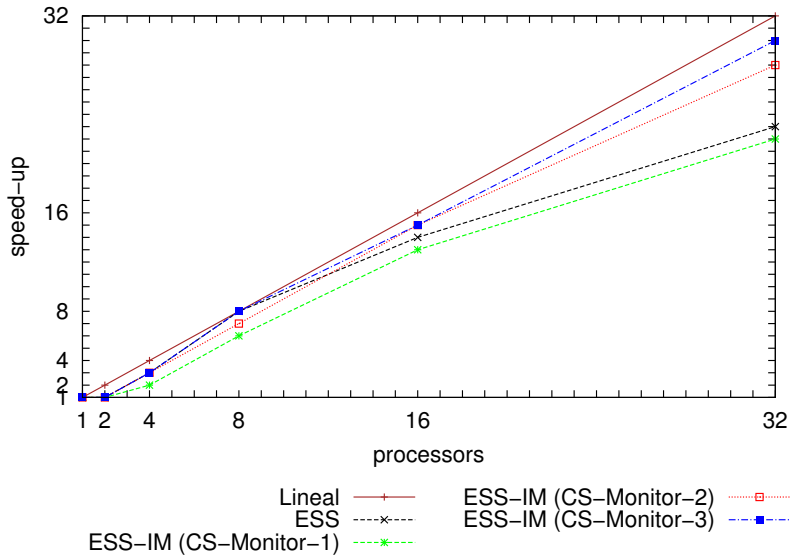


Fig. 9. Speed-up analysis.

5.3 Threats to Validity

By nature, there are threats to validity in any empirical study of this type. In this section, some threats and the way in which they have been addressed are discussed. It should be noticed that whenever a comparative study is made, it is of crucial importance to ensure that it is as reliable as possible. Therefore it is necessary to analyze such threats to ensure that the conclusions are not wrong [26]. One of the important factors is associated with the inherent stochastic behavior of PEA used in both methods. Therefore, for each method were carried out 30 runs for each experiment, just as discussed earlier. Another potential source of bias is presented in the parameter settings used in each method, and the possibility that it may favor or harm the performance of one or both methods. In this case, we decided to take a standard set of parameters for both ESS and ESS-IM so as not to benefit any particular method.

6 Conclusions

Wildfires, like any other phenomenon that may generate some kind of natural disaster, are considered highly dangerous. For this reason, it is of great interest to predict their behavior in order to minimize losses and damages caused by them. Since the simulation of this kind of phenomena is often affected by the uncertainty in the input parameters that feed the model, it is advisable to incorporate uncertainty reduction features aiming at make predictions with the least possible degree of error. Moreover, for such predictions to be used for making decisions that are part of the process of firefighting, such results should be provided in the shortest possible time. To achieve this aim, typically high performance systems that reduce the processing time are usually used. In this study, we have evaluated and compared two methods

for uncertainty reduction that have been developed and applied to wildfires behavior prediction. One of them is called Evolutionary Statistical-System (ESS) and the other ESS with Island Model (ESS-IM), which has three modes of operation, depending on the internal treatment of the probability maps.

It is important to say that such analysis has been carried out in terms of quality of prediction and performance benefits in a parallel/distributed environment. The results show that the new implementation, ESS-IM in **CS-Monitor-1** mode, provides great improvement in quality of prediction. However, this result is slightly overshadowed by the decreased speed-up obtained in comparison with other operation modes of ESS-IM, which prioritize the processing time to the quality of prediction.

Further studies will focus on considering alternatives to optimize the use of computational resources, with the aim of improving the performance gains in the mode that provides the best quality of prediction levels, i.e. ESS-IM in **CS-Monitor-1** mode. In addition, other metaheuristics will be evaluated, as is the case of DESS, which is an ESS version under development that incorporates Differential Evolution as optimization method.

References

- [1] Alba E. and Tomassini M., *Parallelism and evolutionary algorithms*, Evolutionary Computation, IEEE Transactions on **6** (5) (2002), pp. 443–462.
- [2] Bianchini G. and Caymes Scutari P., *Calibration of the parameters of ESS system for Forest Fire prediction*, XXXIX Conferencia Latinoamericana en Informática, CLEI 2013 Volume II. Club Puerto Azul, Naiguatá, Venezuela, (2013), pp. 102–111.
- [3] Bianchini G. and Caymes Scutari P., *Uncertainty Reduction Method Based on Statistics and Parallel Evolutionary Algorithms*, Proceedings of High-Performance Computing Symposium - 40 JAIIO. HPC 2011, Córdoba, Argentina, (2011), pp. 1–4.
- [4] Bianchini G. and Caymes Scutari P., *Métodos Basados en Computación de Alto Rendimiento para Predecir el Comportamiento de Incendios Forestales*, Séptimo Encuentro del International Center for Earth Sciences (E-ICES7). Malargüe, Argentina, (2011), pp. 28–36.
- [5] Bianchini G., Cortés A., Margalef T. and Luque E., *Improved prediction methods for Wildfires using High Performance Computing: A comparison*, Lecture Notes in Computer Science **3991** (2006), pp. 539–546.
- [6] Bianchini G., Cortés A., Margalef T. and Luque E., *Uncertainty Reduction Method Based on Distributed Computing Applied to Forest Fire Prediction*, 38 JAIIO High Performance Computing (HPC 2009). Mar del Plata, Argentina, ISSN 1851-9326 (2009), pp. 99–113.
- [7] Bianchini G., Denham M., Cortés A., Margalef T. and Luque E., *Wildland Fire Growth Prediction Method Based on Multiple Overlapping Solution*, Journal of Computational Science **1** (4) (2010), pp. 229–237.
- [8] Bianchini G., Denham M., Cortés A., Margalef T. and Luque E., *Improving forest-fire prediction by applying a statistical approach*, Forest Ecology and Management **234**(S1) (2006), pp. 210.
- [9] Bianchini G., Méndez-Garabetti M. and Caymes Scutari P., *Evolutionary-Statistical System for Uncertainty Reduction Problems in Wildfires*, XVIII CACIC, XII Workshop de Procesamiento Distribuido y Paralelo (WPDP). Bahía Blanca, Argentina, (2012), pp. 230–238.
- [10] Buyya R., “High Performance Cluster Computing: Architectures and Systems,” Prentice Hall, 1999.
- [11] Cantú Paz E., *A survey of Parallel Genetic Algorithms*, Calculateurs Parallèles, Réseaux et Systems Repartis **10**(2) (1998), pp. 141–171.
- [12] Darwin C., “On the Origin of Species by Means of Natural Selection,” London, Murray, 1859.

- [13] Dongarra J., Foster I., Fox G., Gropp W., Kennedy K., Torczon L. and White A., “Source-book of Parallel Computing,” Morgan Kaufmann Publishers, 2003.
- [14] Gropp W., Lusk E. and Skjellum A., “Using MPI - Portable Parallel Programming with the Message-Passing Interface,” The MIT Press, 1999.
- [15] Mattson T., Sanders B. and Massingill B., “Patterns for Parallel Programming,” Addison-Wesley, 2005.
- [16] Méndez-Garabetti M., Tardivo M. L., Bianchini G. and Caymes-Scutari P., *Predicción del Comportamiento de Incendios Forestales mediante un Método de Reducción de Incertidumbre basado en HPC y Evolución Diferencial*, XVI Workshop de Investigadores en Ciencias de la Computación (WICC 2014), Tierra del Fuego, Argentina, (2014), pp. 690–694.
- [17] Méndez-Garabetti M., Bianchini G., Tardivo M. L. and Caymes-Scutari P., *Evolutionary-Statistical System with Island Model for Forest Fire Spread Prediction*, HPCLatAm 2013, Session: Evolutionary Computation & Scheduling, Mendoza, Argentina, (2013), pp. 61–64.
- [18] Méndez-Garabetti M., Bianchini G., Tardivo M. L. and Caymes-Scutari P., *Predicción del comportamiento de incendios forestales mediante aplicación de modelo de islas a ESS*, 9no Encuentro del International Center For Earth Sciences (E-ICES9), Malargüe, Argentina, (2013), pp. 55.
- [19] Moreno Rodríguez J. M., Rodríguez-Urbieto I., Zavala G. and Martín M., *Peligro de incendio en Castilla-La Mancha bajo escenarios de clima futuro*, Seguridad y Medio Ambiente **114** (2009), pp. 32–42.
- [20] Montgomery D. C. and Runger G. C., “Applied Statistics and Probability for Engineers,” Limusa Wiley & Sons, 2010.
- [21] Mühlenbein H., *Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization*, Parallelism, Learning, Evolution, Lecture Notes in Computer Science **565** (1989), pp. 398–406.
- [22] Omni P. N., “Forest Fires: A Reference Handbook (Contemporary World Issues),” ABC-CLIO, 2005.
- [23] Price K., Storn R. and Lampinen J., “Differential Evolution: A Practical Approach to Global Optimization,” Springer, 2005.
- [24] 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.
- [25] Viegas D. X., “Project Spread - Forest Fire Spread Prevention and Mitigation,” (2004), URL: <http://www.algosystems.gr/spread/>.
- [26] Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B. and Wesslén A., “Experimentation in Software Engineering – An Introduction,” Kluwer Academic Publishers, 1999.