INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 2, Pages 432–448

URGENCY VERSUS ACCURACY: DYNAMIC DATA DRIVEN APPLICATION SYSTEM FOR NATURAL HAZARD MANAGEMENT

ANDRÉS CENCERRADO, ROQUE RODRÍGUEZ, ANA CORTÉS, AND TOMÀS MARGALEF

Abstract. This work faces the problem of quality and prediction time assessment in a Dynamic Data Driven Application System (DDDAS) for predicting natural hazard evolution. Natural hazard management is undoubtedly a relevant area where systems modeling and numerical analysis take a great prominence.

Modeling such systems is a very hard problem to tackle. Besides, the results obtained by simulators usually don't provide accurate information, mostly due to the underlying uncertainty in the input parameters that define the actual environmental conditions at the very beginning of the simulation. For this reason, we have developed a two-stage prediction strategy, which, first of all, carries out a parameter adjustment process by comparing the results provided by the simulator and the real observed hazard evolution. It has been demonstrated that this method improves notably the quality of the predictions. Furthermore, we have designed data injection techniques that allow us to take advantage from real-time acquired information, so that our strategy fits the DDDAS paradigm.

Nevertheless, because of the urgent nature of the systems we deal with, it is also necessary to assess the time incurred in applying the above mentioned strategy, in order for it to be useful and applicable in a real emergency situation. In this sense, we have developed a new methodology for prediction time assessment under this kind of prediction environments, based on Artificial Intelligence techniques.

In this research work, we have chosen forest fires as a representative study case, although the exposed methods can be extrapolated to any kind of natural hazard.

Key words. DDDAS, Data Uncertainty, Natural Hazard Management, Prediction Quality, Prediction Time Assessment

1. Introduction

A natural hazard is an unexpected or uncontrollable natural event of unusual intensity that threatens people's lives or their activities. Unfortunately, the losses caused by natural hazards are increasing dramatically, mostly due to the rapid increase in human population. Therefore, in order to mitigate the tragic consequences of such disasters, it is interesting to be able to make urgent decisions while the natural catastrophe is taking place. For this purpose, many interdisciplinary research has been carried out to provide models/simulators to the community for evaluating in advance the natural hazard evolution. However, model-related issues aside, many simulators lack precision on their results because of the inherent uncertainty of the data needed to define the state of the system environment. This uncertainty is due, basically, to the difficult in gathering precise data at the right places where the disaster is taking place. So, in many cases, the simulators have to work with interpolated, outdated, or even absolutely unknown data values.

In this kind of environmental systems, the use of knowledge extraction techniques from the data collected from different sources (e.g. meteorological stations) would

Received by the editors February 1, 2011.

2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

This research has been supported by the MICINN-Spain under contract TIN2007-64974.

be suitable, in order to improve the accuracy of the predictions, as well as to speed up the simulations. However, as it is widely discussed in [3], when designing such data mining processes, it is critical to take into account the fact that the results we will obtain are affected by the underlying uncertainty in the data.

To overcome the just mentioned input uncertainty problem, we have developed a two-stage prediction strategy, which, first of all, carries out a parameter adjustment process by comparing the results provided by the simulator and the real observed disaster evolution. Then, the underlying simulator is executed taking into account the adjusted parameters obtained in the previous phase, in order to predict the evolution of the particular hazard for a later time instant. A successful application of this method mainly depends on the effectiveness of the adjustment technique that has been carried out. In this sense, our research group has developed several solutions for input parameters optimization, all of them characterized by an intensive data management: use of statistical approach based on exhaustive exploration of previous fires databases [8], application of evolutionary computation [14], calibration based on domain-specific knowledge [32], and even solutions coming from the merge of some of the above mentioned [29]. Since all these approaches perform the calibration stage in a data driven fashion, they all match the Dynamic Data Driven Application Systems paradigm [12, 13, 15].

In particular, we have developed this prediction scheme using forest fire as a study case and it has been demonstrated that the above mentioned adjustment techniques contribute to improve the quality of the fire spread prediction.

Another key point to be considered when dealing with an ongoing disaster is the time incurred in providing evolution prediction results. While a natural catastrophe is taking place, it is necessary to make urgent decisions to effectively fight against it. Many times there exist several constraints that make arise the question of how and where to execute our prediction system, depending on the available resources we have. Consequently, we come up with the *urgency-accuracy* binomial.

There exist diverse factors that may affect both precision of the results and time invested to get them. The power of computational resources is an aspect that has important influence, so are the intrinsic features of both the simulator and the adjustment technique chosen to face the environmental hazard.

For this reason, we introduce a new methodology to characterize each element of the proposed DDDAS prediction process, with the aim of being able to design a tool for prediction time assessment during an emergency management. This work is part of a more ambitious project, which consists of determining in advance, how a certain combination of natural hazard simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality.

This paper is organized as follows. In the next section, an overview of the two-stages DDDAS for forest fire spread prediction is given. In Section 3, we expose in detail our developed solutions for quality enhancement and prediction time assessment, as well as how this framework could be generalized to any natural hazard. In Section 4, the experimental study is reported and, finally, the main conclusions are included in Section 5.

2. DDDAS and Natural Hazard Evolution Prediction

In the field of physical systems modeling, there exist several simulation and forecasting tools for mitigating damages caused by natural hazards [6, 17, 27, 24, 18, 25, 1, 31], based in some physical or mathematical models.

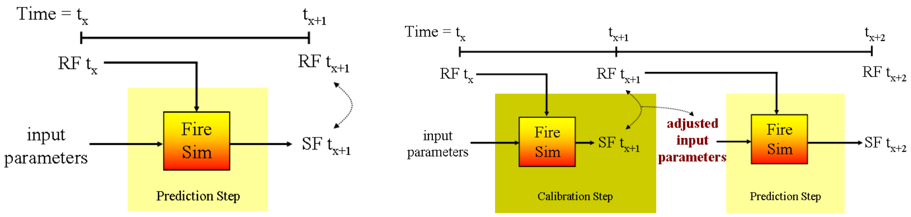
These simulators need certain input data, which define the characteristics of the environment where the catastrophe is taking place, in order to evaluate its future propagation. This data usually consists of the current state of the hazard, and the specification of the variables that define the environment. Some of this data could be retrieved in advance and with noticeably accuracy, as, for example, the topography of the area. However, there is some data that turns out very difficult to obtain with reliability. For instance, getting an accurate fire perimeter is very complicated because of the difficulties involved in obtaining, at real time, images or data about this matter. Other kind of data sensitive to imprecisions is that of meteorological data. These restrictions concerning uncertainty in the input parameters, added to the fact that these inputs are set up only at the very beginning of the simulation process, become an important drawback, because as the simulation time goes on, variables previously initialized could change dramatically. This may mislead results of simulations. Therefore, we need a system capable of dynamically obtain real time input data in those cases that is possible, in order to overcome this disadvantage. For this reason, we rely on the so-called *Dynamic Data Driven Applications Systems* (DDDAS) paradigm.

DDDAS is a paradigm whereby application/simulations and measurements become a symbiotic feedback control system. DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process. An optimal framework for a reliable DDDAS for Natural Hazard Mangement must consider, among others, the following issues: real-time data assimilation strategies for being further injected into the running system; the ability to dynamically couple models from different disciplines; steering strategies for automatic adjusting either models or input data parameters and to have access to enough computational resources to be able to obtain the prediction results under strict real-time constraints. Some current work on this area could be found in [26, 5]. Our current research consists of the first steps towards a DDDAS for Natural Hazard Mangement, where our main efforts are oriented to take advantage of the computing power provided by High Performance Computing (HPC) systems to, in the one hand, propose computational data driven steering strategies to overcome input data uncertainty and, on the other hand, reducing the execution time of the whole prediction process in order to be reliable during real-time crisis.

As a main case study, we apply our developed techniques and methodologies to the case of wildland fires. Our proposal consists of performing a forest fire spread prediction based on a two stages prediction scheme: calibration stage and prediction stage. The DDDAS bases are included in the calibration stage by assimilating the actual propagation of forest fire and using such information for input parameter calibration. Taking advantage of the computer power provided by HPC systems, several strategies such as Evolutionary Algorithms or Statistical Analysis are used to explore a huge number of input parameters combinations (called scenarios). However, building a DDDAS for forest fire prediction also involves to take care about the aspect of the data acquisition. By means of our proposed strategy, our system is able to dynamically acquire new data at real time. These issues are discussed in the next subsections.

2.1. Two-Stage Forest Fire Spread Prediction.

As it is summarised in Figure 1(a), the classic way of predicting forest fire behaviour takes the initial state of the fire front (RF = real fire) as input as well as the input parameters given for some



(a) Classic prediction

(b) Two-stage prediction

FIGURE 1. Prediction Methods

time t_x. The simulator then returns the prediction (SF = simulated fire) for the state of fire front at a later time t_{x+1}.

Comparing the simulation result SF from time t_{x+1} with the advanced real fire RF at the same instant, the forecasted fire front tends to differ to a greater or lesser extent from the real fire line. One reason for this behaviour is that the classic calculation of the simulated fire is based upon one single set of input parameters afflicted with the before explained insufficiencies. To overcome this drawback, a simulator independent data-driven prediction scheme was proposed to optimize dynamic model input parameters [2]. Introducing a previous calibration step as shown in Figure 1(b), the set of input parameters is optimized before every prediction step. The solution proposed come from reversing the problem: how to find a parameter configuration such that, given this configuration as input, the fire simulator would produce predictions that match the actual fire behavior. Having detected the simulator input that better describes current environmental conditions, the same set of parameters, could also be used to describe best the immediate future, assuming that meteorological conditions remain constant during the next prediction interval. Then, the prediction becomes the result of a series of automatically adjusted input configurations.

This two-stage fire prediction methodology reduces the negative impact of input parameters uncertainty. We divided the two-stage strategies into two categories: the Single Solution methods and the Multiple Solutions methods.

The Single Solution scheme uses a Genetic Algorithm (GA) in the calibration stage. In particular, the evolution operations applied in the GA are driven according to the observed real fire behavior. Population is formed by a set of scenarios, and each scenario - an individual - represents a group of simulator parameters. Each parameter is considered a gene of an individual for the GA, which will be evolved by applying the data driven GA until a good single solution is reached. A good solution occurs when the scenario to be introduced in the simulator generates a propagation map akin to that of the real fire to the time interval of the adjustment step. This single solution will then be used in the prediction phase as input parameters to the fire simulator [16].

An implementation of the Multiple Solution method is the Statistical System for Forest Fire Management (S^2F^2M), where thousands of possible scenarios are evaluated. In this case, the information extracted from the calibration stage consists of a statistical integration of the results obtained from the complete set of simulations, which will be compared to the real fire propagation to obtain a probabilistic propagation value. This value will be used in the prediction stage in order to consider not only one possible scenario but the complete set [8].

2.2. Real-Time Data Injection. Another key point when dealing with DDDAS for emergencies is the ability to inject real-time data at execution time in order to provide better forest fire spread prediction. For this reason, we have also developed and design strategies for real-time data gathering and injection, in order to establish a methodology from which DDDAS applications running in distributed environments can take benefit [10]. The proposal raises a great challenge, as there are many aspects to take into account. Our main goal is to provide an external-incoming-data assimilation schema in order to allow obtaining needed information in real time from not (necessarily) trusted platforms, which means to deal with certain key issues:

- To acquire new data from external sources as soon as it is available, to provide more efficicient application executions.
- To provide data sources with access to the execution platforms, which, security issues aside, also means to know which resources are being used by the application.
- To adapt current applications to be able for accepting new incoming data at execution time.
- To be as less intrusive with the execution environment as possible.

We have studied strategies to satisfy these points based on the use of interposition agents, which allow exploring and browsing external file systems in a non-intrusive way and without the need of having special privileges in the environment the users execute their applications, even when it is about a distributed-computing environment. Thereby, we can establish a methodology for solving the previously mentioned problems and allowing many HPC applications to fit in the DDDAS paradigm. Figure 2 shows the basic architecture for this scheme.

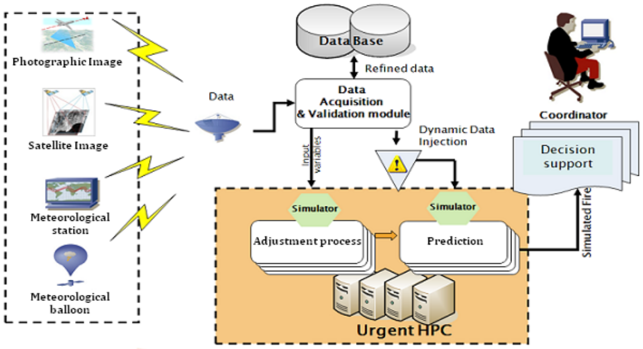


FIGURE 2. Current *Forest Fire Spread Prediction System*

However, building a DDDAS for forest fire prediction implies several extra aspects. These aspects are: the capability of incorporating real time data during simulation execution; Urgent HPC: the ability of turning a HPC system into a high priority HPC system and, finally, the capability to easily interact with different physical models, such as the interaction between fire and atmosphere. Because of these reasons, we propose a new method to reduce execution time, by applying a combination of the above mentioned DDDAS models, which is described in the next section.

3. Quality Enhancement and Time Assessment in Natural Hazard Evolution Prediction

As stated in Section 1, we have to effectively deal with the *urgency-accuracy* binomial, in order to perform a successful management of the hazard.

As regards the accuracy of the results, we subsequently present a prediction system that fits the DDDAS paradigm. This system takes advantage from the ability of real-time data acquisition, in order to improve the predictions.

Nevertheless, as for the urgency part of the binomial, we must be able to satisfy strict time constraints in the prediction process, but paying special attention to minimize the looses on results' fidelity. For this purpose, we also introduce a new methodology to characterize each element of the proposed DDDAS prediction process, with the aim of being able to design a tool for prediction time assessment during an emergency management. This work is part of a more ambitious project, which consists of determining in advance, how a certain combination of natural hazard simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality.

3.1. Quality Enhancement: SAPIFE^3. One common assumption in the methods mentioned in Section 2.1 is that the environmental conditions are stable throughout the adjustment and calibration steps. However, this condition actually does not happen all the time and is, in fact, the biggest source of simulation errors of those methods. For this reason, new techniques had to be introduced to overcome this disadvantage. We propose a new technique called SAPIFE^3 - this is the spanish acronym for *Adaptive System for Fire Prediction Based in Statistical-Evolutive Strategies* (Sistema Adaptativo para la Prediccin de Incendios Forestales basados en Estratgias Estadístico-Evolutivas) [29], which joins the advantages of the two-stages described in the previous section.

This new approach is able to reduce the number of total scenarios to simulate, from a number such as hundreds of thousands to some hundreds, by optimizing the set of scenarios through the use of a Genetic Algorithm. This method reduces execution time, and improves dynamic adaptation applying a combination of the DDDAS models. This is feasible thanks to the *Data Acquisition and Validation* module (see Figure 2), which gathers all information regarding the fires environment, such as weather, topography and terrain composition data (the combustible).

SAPIFE^3 runs in two phases. In the first one, the Genetic Algorithm (GA) is used, and in the second one, the Statistical method is applied over the population obtained from the GA. In the case of the Genetic Algorithm, tasks are the product of the total amount of scenarios and the number of generations G. In the Statistical method, only the last population produced in the previous stage is processed.

This system has been developed so that it allows new data insertion between the GA process and the Statistical process, if it is detected an important change in the environmental conditions. This feature is the responsible of the improvement of predictions quality, as it will be exposed in Section 4.

3.2. Prediction Time Assessment. A key point to be considered when dealing with an ongoing disaster is the time incurred in providing evolution prediction results. In order to be useful, any evolution prediction of an ongoing hazard must be delivered as fast as possible for not being outdated. For this purpose, we introduce a new methodology to characterize each element of the proposed DDDAS prediction process, with the aim of being able to design a tool for prediction time assessment during an emergency management. As in the case of the quality aspect, we have used forest fire spread prediction as study case.

In order to approach the problem in an organized way, in this paper we first introduce how to characterize the core of the DDDAS for the case of forest fire spread prediction. As it is well known, the execution time of the underlying simulator depends on the specific setting of the input parameters. For this reason, decision trees were used to obtain an upper bound for the simulator execution time by previously classifying it according to the input parameter setting. The proposed classification scheme has been carried out considering two different fire spread simulators in order to validate the classification strategy with different setup conditions.

This goal is oriented to provide the personnel in charge of taking decisions about how to face an ongoing emergency, with intelligent tools able to evaluate, in advance, how a certain combination of simulator, computational resources, adjustment strategy, and frequency of data acquisition will perform, in terms of execution time and prediction quality. In order to bound the problem, we work under certain assumptions:

- We focus on those emergencies where the corresponding simulators present high input-data sensitivity.
- We assume scenarios where the computational resources are dedicated. Currently, we are working on adapting tools that allow urgent execution of tasks in distributed-computing environments, e.g. SPRUCE [7].
- We rely on the two-stage DDDAS prediction strategy.

Taking into account these premises and bearing in mind the scheme shown in figure 3, we can define three levels of prediction time assessment: Simulator level assessment (SLA), Adjustment level assessment (ALA) and Prediction level assessment (PLA).

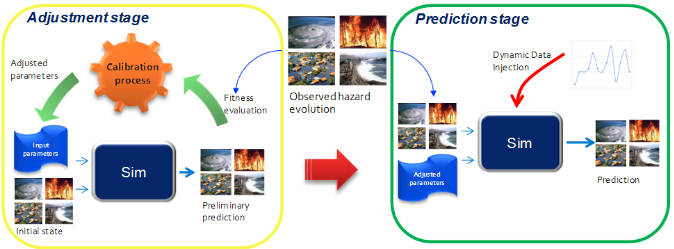


FIGURE 3. General two-stages DDDAS for natural hazard prediction evolution

3.2.1. Simulator level assessment (SLA). Prediction time assessment at this level must be done independently on the underlying simulator (natural hazard) and the particular setting of their input parameters. The main objective at this level is to define a simulator-independent methodology to determine a clustering classification of the simulator execution time, where each cluster has associated an upper bound for the execution time depending on the values of the input parameters. This process is carried out in an *off-line* way and will be widely explained later on in this paper. Since this characterization process depends on the executable platform, different simulator characterizations will be performed for each available computational resource.

3.2.2. Adjustment level assessment (ALA). This level corresponds to estimate the prediction time increase due to the calibration strategy used in the *Adjustment stage*. As we have previously mentioned, there exist several calibration strategies that have been demonstrate to be useful for improving the prediction quality of a hazard evolution. Each one of this optimization schemes must be modeled independently of each other because the way of performing is quite different. As it could be observed in figure 3, there is a thight relation between the results obtained at SLA with this level because SLA is inside ALA, therefore, ALA is directly proportional to SLA.

3.2.3. Prediction level assessment (PLA). At this level one can rely either on dynamic data injection to the system or not. A pure DDDAS will take into account data injection at real time and this is the way that the DDDAS for forest fire spread prediction has been designed in its advanced form. However, in a preliminary version, the dynamic data injection was not considered and it was based in the working hypothesis that the environmental conditions keep constant from the calibration stage to the prediction stage. For this reason, the PLA methodology has been designed in a two step fashion, first of all we will determine a standard methodology for the prediction stage without real time data injection and, afterwards, the PLA's characterization will be performed, taking into account data gathering frequency and data source. The aim consists of reaching the capability to determine the probability distribution that indicates which percentage of prediction improvement has *historically* been obtained in the cases where the data was acquired with a certain frequency, and from a certain data sources. This characterization level, as in SLA, relies on a massive sadistic study.

It is important to notice that in the characterization of the simulator, we focus on the execution time as a "classification criteria", whereas the quality of prediction is the factor taken into account when characterizing the adjustment stage (ALA). This is because the quality of the initial prediction given by the simulator has no influence over the final prediction. Nevertheless, the execution time of each calibration technique is directly proportional to the execution time of the simulator. Hence, in order to estimate both accuracy of prediction and time needed to perform it, the study of these aspects is carried out in this way. In the next section, an empirical study concerning the method followed for the Simulator Level Assessment is detailed and the obtained results are analyzed.

4. Experimental Study

In the previous section, it has been exposed our developed strategies to tackle both the quality enhancement and prediction time assessment problems. Subsequently, we present experimental studies that validate these proposals, and expose how the application of them turns out very beneficial in order to face real emergency situations.

4.1. Quality Enhancement through Dynamic Data Injection. In order to test our DDDAS forest-fire propagation prediction system, we performed a series of postmortem experiments based in the conditions of the Horta de Sant Joan fire (Tarragona-Spain) on July 2009. Figure 4 shows the extension of this fire, where the environmental conditions were quite dynamic, showing suddenly changes in wind speed and wind direction. There are several weather stations in the region, property of the Meteorological Service of Catalonia. Data for the next experiments were gathered from station "D8 Horta de Sant Joan (D8HSJ)" weather station, located at latitude 4046'20.52"N and longitude 017'54.29"E, inside the area affected by the fire.This station was chosen because it monitors humidity, wind speed and wind direction every thirty minutes. We also used the MODIS Hotspot detection system [22], which allows fire data to be visualized into Google Earth using KML languaje [23], so it is possible to verify the situation of the fires. Figure 4 also shows the location of the D8 Sant Joan weather station.

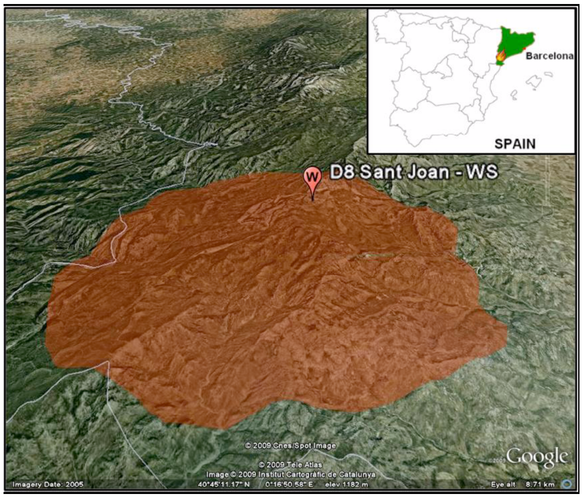


FIGURE 4. Sant Joan Fire view using Google Earth

The experiments, reported in this section take into account the changes in both wind direction and wind speed, recorded by the above mentioned weather station and two different approaches of the proposed fire prediction method were used:

- SAPIFE^3, where wind data is always reinserted.
- SAPIFE$^3_{rt}$, which includes the possibility to set a speed and direction change threshold between the adjustment and prediction steps.

The main objective of these experiments were to demonstrate the benefits of DDDAS for forest fire prediction, specially when environmental conditions are quite dynamic showing suddenly changes in wind speed and wind direction.

As it was described in section 3.1, SAPIFE^3 is composed by the Genetic Algorithm (GA) and the Statistical method. In the proposed experiment, the real-time data injection is done after the GA stage and just at the beginning of the statistical method. The particular configuration for this experiment can be checked in Table 1.

Slope and vegetation model are assumed to be known, therefore they are set up as constants inputs for all experiments and schemes. As it has been previously described, the measurement of wind speed and wind direction are available, not only at the very beginning of the fire, but also every 30 minutes (recorded by D8 weather station). Although these data availability, the only dynamic data driven prediction schemes that can take advantage of their such information are SAPIFE^3 and SAPIFE$^3_{rt}$ because of ability to receive real-time data at execution time.

Figure 5 depicts the prediction error provided by each method once the prediction stage has finished. In this stage, the right y axis shows wind speed in the different prediction time steps, and the left y axis shows error level, and the bottom x axis shows the interval for prediction stage in real local time, the upper x axis

Table 1. SAPIFE^3 Configuration

Parameter	Value
Model	4
sizePopulation	2500
numGenerations	5
elitism	20
crossoverProbability	0.2
mutationProbability	0.01
slope	5
aspect	0

shows the time when data of wind speed is collected. For instance, the first prediction provide by the proposed prediction scheme will be delivered after 60 minutes from he beginning of the whole process, being the first 30 minutes of execution for warmup, when the system calibrates itself with the first two sets of data from the meteorological stations, the one from time 0 and the subsequent one from time 30. This warmup phase only happens once, and then results will be delivered in intervals of 30 minutes.

In this comparison, we included the FireLib results representing the classical prediction method. It is important to notice that, although we are depicting time intervals that exactly last 60 minutes, in fact, the prediction results are provided before reaching the end of the corresponding interval time. However, we can not evaluate the goodness of the obtained prediction until reaching the end of the underlying time interval. That is the reason we plot as prediction interval the exact times.

For example, in the step 12 to 1 p.m., the system will deliver the prediction fire behavior for time 1 p.m. when the second wind value is read at time 1:30 p.m. or in another words, how will be the fire behavior between 12 p.m. and 1 p.m. However, the real prediction validation will be performed only when fire propagation will reach time 1 p.m. The same happen at each prediction step as shown in figure 5.

An immediate conclusion obtained from observing figure 5 is that FireLib prediction results are for all time intervals the worst. This fact states that the classical prediction scheme, where no dynamic data driven approach is included, is a clear drawback of such a scheme. $SAPIFE^3$ and $SAPIFE^3_{rt}$ are shown to be the bests. The ability of injecting real-time data allows, for the case studied, to keep bounded the error ratio below 30% even in presence of drastic wind changes.

If we observe wind behavior, we can see that it suffers from extreme change on three specifics periods of time 3 p.m. to 4 p.m., and 5 p.m. to 6 p.m., and 7 p.m. to 8 p.m. These changes are taken into account by $SAPIFE^3$ and $SAPIFE^3_{rt}$ when performing the prediction stage. This fact represents a big advantage, because of this change will generate an increase in fire spread velocity that will not be considered otherwise. Consequently, $SAPIFE^3_{rt}$ achieves the best error ratio than $SAPIFE^3$. Besides, we can see that the improvement over $FireSim$ is more than 40%.

In the time period 2 p.m. to 3 p.m., $SAPIFE^3$ is the one who better performs. This happened because the wind conditions keep quite similar between the adjustment and prediction phases. This turns the individual found between 1:00 p.m. and 1:30 p.m. to be very good also for the period 1:30 p.m. to 2:00 p.m. However, $SAPIFE^3_{rt}$'s results are very close to it, even in those stable conditions. Therefore,

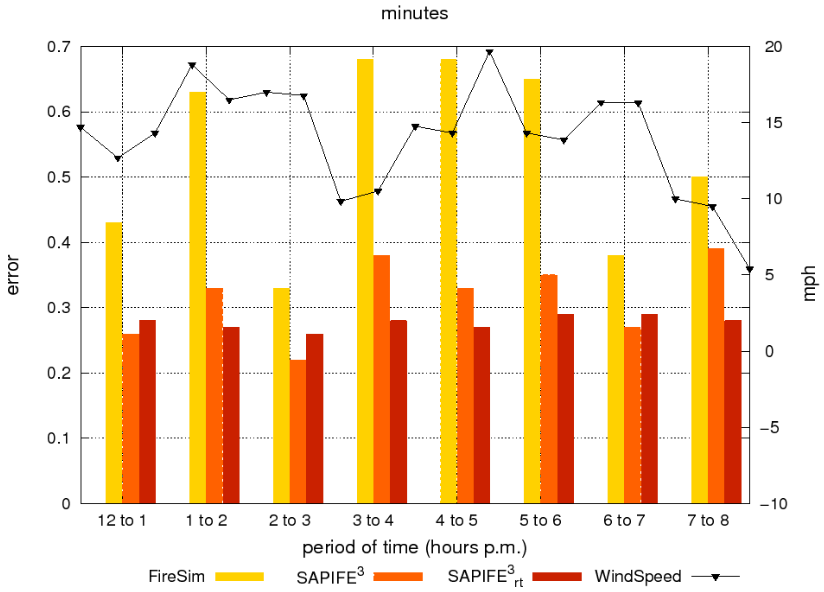


Figure 5. Comparison between methods

this parameter was not changed. This predicted better than $SAPIFE^3$, which is negatively affected by wind's parameters insertion. In fact its changes were subtle until period of time 3 p.m. to 4 p.m., when again, they change a lot. This affects seriously the prediction of all methods except for $SAPIFE^3_{rt}$, which are able to get results with error ratios about 30%.

4.2. Prediction Time Assessment via Decision Trees Classification. As stated above, the fact of having well characterized each simulator we deal with, in terms of execution time, becomes crucial to validate the proposed methodology.

This matter may be tackled by means of taking the strategy of carrying out large sets of executions of the underlying simulator, and then analyzing its behavior from the obtained results. However, this fact may not be trivial in certain cases. While it is easy to detect that the application presents a high sensitivity to certain input parameters, even in an intuitive way, some of them produce a behavior of the simulator that turns out hard to predict. Figures 6 and 7 show examples of each case, respectively. In the former, one can observe that the dimension of the map to be simulated has a direct influence on the execution time (as it was bound to happen), whereas, in the latter, it can be noticed that the relation between execution time and wind direction is not so clear (this *anomaly* is reported in [19]), and even it becomes odder when combining variations in wind direction with variations with vegetation type.

Currently, this characterization is fulfilled by means of carrying out large sets of executions (on the order of tens of thousands) counting on different initial scenarios (different input data sets), and then, applying knowledge-extraction techniques from the info they provide. We record the execution times from the experiment, and then we establish a classification of the input parameters according the elapsed times they produced. At this moment, we are capable to apply machine learning

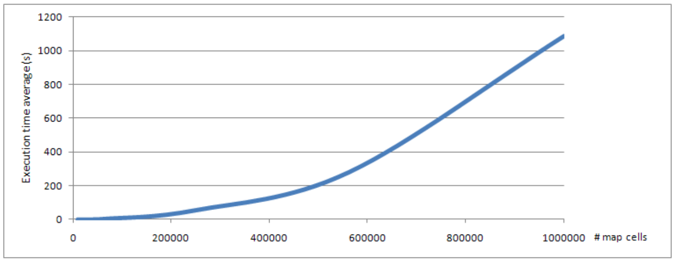


Figure 6. Execution time as a function of number of cells.

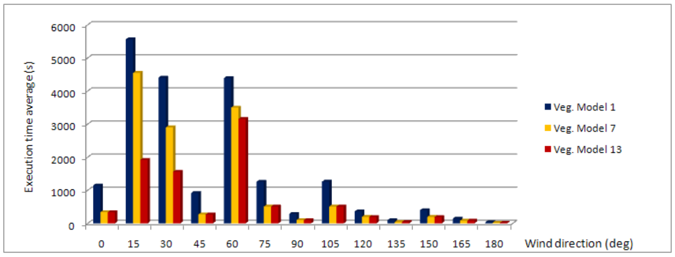


Figure 7. Variations in execution time according to variations in wind direction and vegetation type.

techniques to determine classification criteria and, therefore, given a new set of input parameters, to be able to estimate how much the execution will last.

This fact highlights the need to base on complex criteria in order to successfully classify the input data sets according to the execution time they will cause. Consequently, we rely on the field Artificial Intelligence to reach such an objective. Specifically, this experimental study shows the results obtained from the use of decision trees as classification technique.

4.2.1. Test bed description. For validation purposes, we have used two different forest fire spread simulators in the experimental study: HFire [27] and fireLib [18]. HFire is a spatially explicit fire spread model that was developed for modeling fires in chaparral environments in 2001. FireLib is a C function library for predicting the spread rate and intensity of free-burning wildfires, developed in 1996. Both of them are based on the Rothermel fire model [30] to determine the direction and magnitude of the maximum rate of spread. Because of the specific features of each simulator, the simulated scenario slightly differs in each case as are subsequently listed:

- HFire:
  - Domain: The domain studied in the case of HFire was the Santa Monica Mountains National Recreation Area (SMM) in southern California, which topografy details are provided in [20].
  - Simulation duration: In the case of HFire a 10-day simulation was carried out in every execution.
  - Ignition point: When using HFire, it was approximately the ignition point of the well known 1996 Calabasas fire in California (also provided in [20]).
- FireLib:

– Domain: For the characterization of fireLib, an artificial 1001x1001 cells map was used (cells width and height: 100 feet). In both cases, the indicated topography remained constant for all the executions.
– Simulation duration: FireLib simulations end once the fire reaches one edge of the map.
– Ignition point: The ignition point in the case of fireLib was the central cell of the map.

Apart from these peculiarities, the rest of input parameters were the same in both cases. Specifically, Table 4.2.1 shows the assigned probability distributions for each type of input. As regards wind speed and direction, the chosen distributions and their associated parameters were the ones used in [11], based on statistical analysis of data from weather stations in the area of SMM. The vegetation models correspond to the 13 standard Northern Forest Fire Laboratory (NFFL) fuel models [4].

Input	**Distribution**	μ,σ	**Min,Max**
Vegetation model	Uniform	—	1,13
Wind Speed	Normal	12.83,6.25	—
Wind Direction	Normal	56.6,13.04	—
Dead fuel moisture	Uniform	—	0,1
Live fuel moisture	Uniform	—	0,4

Table 2. Input parameters distributions description.

Once established the distribution of each input parameter, a set of 38750 different combinations of input data sets was generated, and the simulations of each scenario for each simulator were performed.

As regards the computational platform, all the experiments carried out in this work were done on a cluster of 32 IBM x3550 nodes, each of which counting on 2 x Dual-Core Intel Xeon CPU 5160, 3.00GHz, 4MB L2 cache memory (2x2) and 12 GB Fully Buffered DIMM 667 MHz, running Linux version 2.6.16.

4.2.2. Preliminary conclusions. Figure 8 depicts the histogram obtained from the execution of the test bed using HFire simulator. As it can be observed, there exist some execution time intervals which assemble most of the execution instances. Nevertheless, the important matter concerning these results is that HFire shows a very regular behavior, so the whole execution time interval is short enough to discard a classification process. Hence, when using HFire as a fire spread simulator, we can assume the worst case (executions will last approximately 29 seconds) for the characterization of the whole prediction process.

This behavior contrasts with the one obtained from the fireLib simulator. As one can see in Figure 9, the variance on the simulation time is very noticeable. The great majority of the executions are located under the 2500 seconds threshold, but

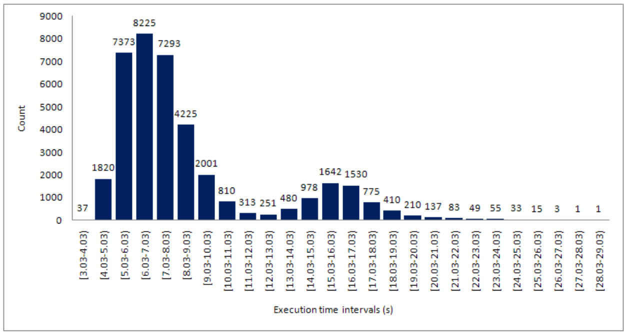


Figure 8. Histogram of execution times using HFire.

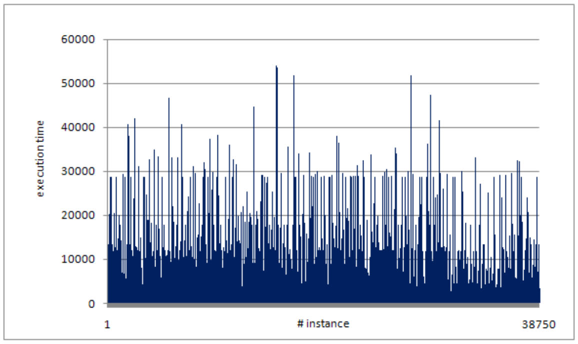


Figure 9. Execution times using fireLib.

there were several executions that lasted more than 30000 seconds, and even more than 50000 seconds.

From the point of view of emergency prediction, it is crucial to have the question of execution time under control, so we may deal with cases that drastically slow down the prediction process. An elapsed time prediction for a simulator execution with an error on the order of thousands of seconds would be prohibitive, so, from cases like this one, there arises the need to be able to predict how the simulator is going to behave and, therefore, the need to use an efficient classification technique.

4.2.3. Empirical evaluation. In order to respond to this need, the experimental study carried out in this work consisted of use decision trees as the classification method, to be able to estimate, in advance, the execution time of fireLib, given a new unknown set of input parameters.

The decision trees used in this research were the generated by the C4.5 algorithm [28], specifically, the J48 open source Java implementation of the C4.5 algorithm in the Weka [21] data mining tool. The data obtained from the 38750 executions was used as a training set, and 1000 new instances were generated (according to the distributions shown in Table 4.2.1) to be used as a test set.

The number of classes, and the execution time intervals they represent, were determined taking into account where our work is framed, i.e. the intervals chosen for each class are those that in a real emergency situation would matter (it has no sense, for example, to classify by intervals of 10 seconds when predicting forest fire spread). They are:

- Class A: ET $\leq$ 900 seconds.
- Class B: 900 seconds $<$ ET $\leq$ 1800 seconds.
- Class C: 1800 seconds $<$ ET $\leq$ 3600 seconds.
- Class D: 3600 seconds $<$ ET $\leq$ 7200 seconds.
- Class E: 7200 seconds $<$ ET.

Where *ET* stands for execution time.

The results of the application of decision trees to the test set are summarized in Table 4.2.3. Here, one of the main aspects to highlight is the prominence of the main diagonal, which means that perfect matches are predominant over the whole set of predictions. Furthermore, one can notice that the values decrease as one moves away from the main diagonal. Indeed, the worst possible cases (predict A when the real class is E, and vice-versa), never happened.

Figure 10 shows the absolute values of the number of predictions that totally hit the real class, as well as the absolute values where the prediction had an accuracy determined by the distance between classes. A *Distance X* accuracy means that there are X-1 classes between the predicted class and the real class.

The most noticeable aspect when analyzing this graphic is that if we consider *Distance 1* as a good prediction accuracy, then the results obtained present a 96.8% of satisfactory classifications.

		Predicted Class				
		A	B	C	D	E
Real Class	A	669	14	4	2	0
	B	17	72	9	4	0
	C	2	12	72	12	4
	D	5	6	14	24	5
	E	0	3	2	12	36

TABLE 3. Correspondence between real and predicted classes.

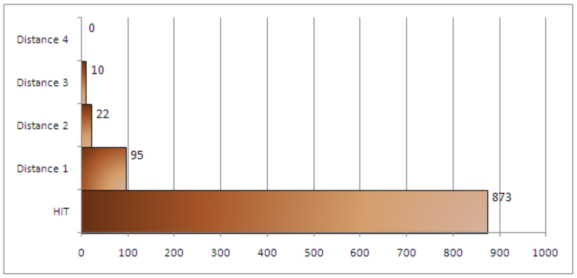


FIGURE 10. Classification accuracy.

5. Conclusions

In this paper, we have described the main lines of our current research, which consists of a first step toward a DDDAS for Wildland Fire Prediction. Our main efforts are oriented to take advantage of the computing power provided by High

Performance Computing (HPC) systems to, in the one hand, propose computational data driven steering strategies to overcome input data uncertainty and, on the other hand, reducing the execution time of the whole prediction process in order to be reliable during real-time crisis.

We presented a two-stage prediction method, which fits the DDDAS paradigm, that allows to improve the quality of evolution prediction in different natural hazards. Particularly, we presented a DDDAS for forest fire spread prediction with real time data injection. We performed a series of experiments based on the behavior of wind speed and wind direction in a real case that produced human losses.

This way, we showed the importance of the DDDAS systems for forest fire prediction, and how they can improve the fire simulators' output, when conditions are dynamic and changes are sudden. We also observed that data insertion at real time can improve the prediction results significantly.

Furthermore, we have designed a methodology to assess the urgency-accuracy binomial in each particular case. This methodology can be extrapolated to any DDDAS for predicting natural hazards evolution, which uses the two-stage prediction scheme as a working framework. As it is well known, the execution time of a particular simulator depends on the specific setting of the input parameters. However, as it has been exposed, it becomes hard to predict how certain variations on certain input parameters would affect the execution time. In this work, we approach such a challenge by means of Artificial Intelligence and Data Mining techniques. Particularly, we present how we deal with simulators characterization by means of the use of decision trees as classification technique.

The obtained results demonstrate that the use of decision trees as classification strategy is suitable for this research, obtaining up to 96.8% of satisfactory classification prediction, which represents a great advance, and allows us to tackle the subsequent steps of the proposed methodology.

References

[1] Aberson, S.D., *Five-day tropical cyclone track forecasts in the North Atlantic basin*, Weather and Forecasting, Volume 13, pp. 1005–1015. 1998.

[2] Abdalhaq, B., *A methodology to enhance the Predction of Forest Fire Propagation*, PhD Thesis dissertation. Universitat Autònoma de Barcelona (Spain). June 2004.

[3] Aggarwal, C.C. and Yu, P.S., *A Survey of Uncertain Data Algorithms and Applications*, IEEE Transactions on Knowledge and Data Engineering, Volume 21(5), pp. 609–623, 2009.

[4] Albini, F.A., *Estimating wildfire behavior and effects.* Gen. Tech. Rep. INT-GTR-30. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station. 1976.

[5] Allen, G., *"Building a Dynamic Data Driven Applications System for Hurricane Forecasting."* ICCS 2007, Part I, LNCS 4487, pp. 1034-1041.

[6] Andrews, P.L., *BEHAVE: Fire Behavior prediction and modeling systems - Burn subsystem*, part 1. General Technical Report INT-194. Odgen, UT, US Department of Agriculture, Forest Service, Intermountain Research Station. 1986.

[7] Beckman, P., Nadella, S., Trebon, N. and Beschastnikh, I., *SPRUCE: A System for Supporting Urgent High-Performance Computing*, Grid-Based Problem Solving Environments, Volume 239/2007, pp. 295–311. 2007.

[8] Bianchini, G., Cortés, A., Margalef, T. and Luque, E., *Improved Prediction Methods for Wildfires Using High Performance Computing A Comparison*, LNCS, Volume 3991, pp. 539–546, 2006.

[9] 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, Volume 1, Issue 4, pp. 229–237. Ed. Elsevier Science. 2010.

[10] Cencerrado, C., *"Real-time Data Flow Management for DDDAS-based Applications under Distributed Computing Environemnts"*, MSc Thesis. Universitat Autònoma de Barcelona (Spain), July 2009.

[11] Clark, R.E., Hope, A.S., Tarantola, S., Gatelli, D., Dennison, P.E. and Moritz, M.A., *Sensitivity Analysis of a Fire Spread Model in a Chaparral Landscape*, Fire Ecology, Volume 4(1), pp. 1–13. 2004.

[12] Darema, F., *"Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements."* ICCS 2004, LNCS 3038, Springer Berlin / Heidelberg, pp. 662-669.

[13] Darema, F., *Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems*, Proceedings of the IEEE, Volume 93(3), pp. 692–697. 2005.

[14] Denham, M., Cortés, A. and Margalef, T., *Computational Steering Strategy to Calibrate Input Variables in a Dynamic Data Driven Genetic Algorithm for Forest Fire Spread Prediction*, Lecture Notes in Computer Science, Volume 5545(2), pp. 479–488, 2009.

[15] *"Dynamic Data Driven Application Systems homepage."* http://www.dddas.org. Acceded on November 2007.

[16] Denham M., Cortés A., Margalef T. and Luque E., *Applying a Dynamic Data Driven Genetic Algorithm to Improve Forest Fire Spread Prediction*. ICCS 2008, Part III, LNCS 5103, pp 36-45. Krawkow, Poland, June 2008.

[17] 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, 1998.

[18] FIRE.ORG - Public Domain Software for the Wildland fire Community. http://www.fire.org.

[19] fireLib User Manual and Technical Reference (online). http://www.fire.org/downloads/fireLib/1.0.4/doc.html.

[20] HFire Fire Spread Model homepage. http://firecenter.berkeley.edu/hfire/.

[21] Holmes, G., Donkin, A., and Witten, I.H., *Weka: A machine learning workbench*, Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. pp. 357–361. 1994.

[22] Justice, C.O, Giglio, L., Korontzi, S., Owens, J., Morisette, J.T., Roy, D., Descloitres, J., Alleaume, S., Petitcolin, F., Kaufman, Y., *The MODIS fire products*, *Remote Sensing of Environment*, ELSEVIER., vol. 83, pp. 244-262, 2002.

[23] KML Support homepage , "KML Tutorial," Nov, 2008; http://code.google.com/apis/kml/documentation/mapsSupport.html.

[24] Lopes, A., Cruz, M. and Viegas D. *FireStation - An integrated software system for the numerical simulation of fire spread on complex toography*. Environmental Modelling and Software 17(3), pp. 269–285. 2002.

[25] Madsen, H. and Jakobsen, F., *Cyclone induced storm surge and flood forecasting in the northern Bay of Bengal*, Coastal Engineering, Volume 51, Issue 4, pp. 277–296. 2004.

[26] Mandel, J., Beezley, J., Bennethm, L., Chakraborty, S., Coen, J., Douglas, C., Hatcher, J., Kim, M. and Vodacek, A., *"A Dynamic Data Driven Wildladnd Fire Model."* ICCS 2007, LNCS 4487, Springer Berlin / Heidelberg, pp. 1042-1049.

[27] Morais, M., *Comparing spatially explicit models of fire spread through chaparral fuels: a new algorithm based upon the Rothermel fire spread equation*. PhD Thesis, University of California, USA. 2001.

[28] Quinlan, J.R., *Improved use of continuous attributes in c4.5*, Journal of Artificial Intelligence Research, Volume 4, pp. 77–90. 1996.

[29] Rodríguez, R., Cortés, A. and Margalef, T., *Injecting Dynamic Real-Time Data into a DDDAS for Forest Fire Behavior Prediction*, Lecture Notes in Computer Science, Volume 5545(2), pp. 489–499, 2009.

[30] Rothermel, R.C., *How to Predict the Spread and Intensity of Forest and Range Fires*, USDA FS, Ogden TU, Gen. Tech. Rep. INT-143, pp. 1–5. 1983.

[31] Weber, H.C., *Hurricane Track Prediction Using a Statistical Ensemble of Numerical Models*, Monthly Weather Review, Volume 131, pp. 749-770. 2003.

[32] Wendt, K., Cortés, A., and Margalef, T., *Knowledge-guided Genetic Algorithm for input parameter optimisation in environmental modelling*, Procedia Computer Science 2010, Volume 1(1), International Conference on Computational Science (ICCS 2010), pp. 1361–1369.

Computer Architecture and Operating Systems Department, Escola d'Enginyeria, Universitat Autònoma de Barcelona, Bellaterra, 08193 Barcelona, Spain

E-mail: acencerrado@caos.uab.es and dario.rodriguez@caos.uab.es, and ana.cortes@uab.es and tomas.margalef@uab.es