

Computational Steering Strategy to Calibrate Input Variables in a Dynamic Data Driven Genetic Algorithm for Forest Fire Spread Prediction^{*}

Mónica Denham, Ana Cortés and Tomás Margalef

Departament d' Arquitectura de Computadors i Sistemes Operatius, Universitat Autònoma de Barcelona, 08193 - Bellaterra (Barcelona) Spain

Abstract. This work describes a Dynamic Data Driven Genetic Algorithm (DDDGA) for improving wildfires evolution prediction. We propose an universal computational steering strategy to automatically adjust certain input data values of forest fire simulators, which works independently on the underlying propagation model. This method has been implemented in a parallel fashion and the experiments performed demonstrated its ability to overcome the input data uncertainty and to reduce the execution time of the whole prediction process.

1 Introduction

Forest fires are part of natural balance in our planet but, unfortunately, during last years the number of forest fires had increased in an alarming way. The high number of this kind of disasters break the natural balance that forest fire means.

Nowadays, people is arduously working on this problem in order to avoid and to reduce forest fires damages. As results of this effort there exist different kind of studies, strategies and tools used to prevent fires, to define risk areas and to reduce the fire effects when a disaster occurs.

Forest fire simulators are a very useful tool for predicting fire behavior, simulators allow us to know the fire progress, intensity, spread area, flame length, etc. Nowadays, there exist several forest fire simulators [7], which may differ in inputs, outputs, fire model implemented, fire type (crown, surfaces or underground fires), etc.

A forest fire simulator needs to be fed with data related to the environment where fire occurs such as terrain main features, weather conditions, fuel type, fuel load and fuel moistures, wind conditions, etc. However, it is very difficult to exactly evaluate the real time values of these parameters for different reasons. There are certain parameters that change through time such as air and fuel humidities. Environmental conditions are also affected by the fire itself due to its elevated temperatures, fires could generate very strong gust of winds as well, etc. The lack of accuracy of the input parameter values adds uncertainty to the

^{*} This work is supported by the MEC-Spain under contracts TIN 2007-64974

whole method and it usually provokes low quality simulations [1].

Thus, in order to achieve high simulation quality, our application is held at Dynamic Data Driven Application Systems (DDDAS) paradigm [4] [5] [9]. In particular, our prediction system explores multiple fire propagation scenarios (different combinations of the input parameters values) dynamically adapting those scenarios according to observed real fire evolution. By the observation of real fire progress, certain input parameter values are steered in order to reduce the whole search space achieving a response time reduction. Consequently, steering the parameter values will improve its value accuracy improving predictions quality as well.

Moreover, in order to reduce response time, we also had developed our application using a parallel solution (master/worker programming paradigm).

This work is organized as follow. Next section describes the proposed dynamic data driven forest fire prediction methodology compared to the classical prediction scheme. Section 3 is focused to describe the Calibration Stage of the proposed prediction methodology and two steering strategies are described. Experimental results are shown in section 4 and, finally, main conclusions and future work are reported in section 5.

2 Forest Fire Spread Prediction

Traditionally, forest fire prediction (figure 1 (a)) is carried out using a forest fire simulator (FS), a set of input parameters (slope, vegetation, dead fuel humidity, live fuel humidity, wind characteristics) and the state of the fire front at a given instant time t_i (called RFt_i : Real Fire for instant t_i). Using this information, the predicted fire line for a time t_{i+1} is obtained (SFt_{i+1} : Simulated Fire for instant t_{i+1}). This method consumes very few resources in terms of time and computation power (it performs just one simulation using the unique scenario).

However, this simplicity suffers from a very important drawback: usually the predictions obtained are far from the real fire spread due to simulator underlying uncertainty and the quality of the unique scenario.

In order to improve the prediction accuracy, we include a stage called Calibration Stage that will be executed before the classical prediction scheme (called Prediction Stage, figure 1 (b)). Within this new stage, we used a Genetic Algorithm (GA) for evolving a set of different scenarios in order to improve their values accuracy (each scenario is an individual of a given GA population).

At Calibration Stage, the beginning fire line (RFt_i) and a set of parameter values are used to obtain the simulated fire line for instant t_{i+1} (SFt_{i+1}). This simulated fire line and the real fire line (SF and RF both for instant t_{i+1}) are compared and the result of this comparison is used as feedback tuning information to improve parameter values accuracy. This process is executed for all population individual, through a prefixed number of generations. Once Calibration Stage ends, the Prediction Stage takes place where the best parameter's set values found in the Calibration Stage is used to feed the simulator and to obtain the final prediction for the next time instant (SFt_{i+2}) [1].

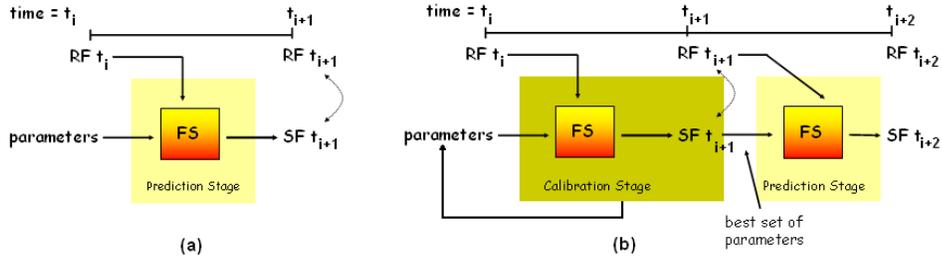


Fig. 1. (a) Classical method for forest fire prediction. (b) Two stage method for forest fire prediction.

Since in the Calibration Stage the original random values setting for the GA is updated to be driven by the observed real fire behavior, it could be referred as a Dynamic Data Driven Genetic Algorithm (DDDGA). In the subsequent section, the proposed DDDGA strategy is described.

3 Parallel Dynamic Data Driven Genetic Algorithm

Classical GAs [8] are inspired in evolutionary theory where a population of individuals is evolved generation by generation with the objective of improving population individuals characteristics. Some operations are applied to obtain each of the generations: *selection* (including elitism), *crossover* and *mutation*.

Through our application frame, the goodness of a given individual (scenario) is determined by real and simulated fire maps comparison (SFt_{i+1} and RFt_{i+1} in figure 1(b)) where maps differences determine the error of the simulation. Therefore, GA main goal is to minimize this error.

In this work, we use the forest fire simulator *fireLib* [3] [7] [10] for surface fires. Some of *fireLib* input parameters are: terrain slope (direction and inclination), wind speed and wind direction, moisture content of live fuel, moisture content of dead fuel (at three different times: 1 hour, 10 hours and 100 hours), vegetation type, etc. In particular, vegetation is modeled by considering the vegetation models defined in [2].

Vegetation type and slope are the most static parameters, therefore, their values will be considered static and known for each prediction process. Consequently, the DDDGA will only consider for evolving purposes the remainder parameters: wind direction and wind speed, moisture content of live fuel and the three moisture content of dead fuel.

In a previous work [6], the authors introduced the data assimilation process needed for this two stage prediction method and an analytical steering strategy for the Calibration Stage was also described.

It is well known that slope and wind are the two main features to determine fire progress. Thus, during most fire model implementations slope and wind factors are composed in order to obtain fire progress direction and velocity. Through

Calibration Stage, we dispose of fire progress from instant t_i to t_{i+1} (figure 1 (b)), then we analyze this fire progress and we obtain fire direction and velocity (the real fire spread characteristics for instant t_{i+1}).

Taking into account these facts and knowing slope characteristics (as we had mentioned in a previous paragraph), we could combine slope and real fire spread in order to obtain wind values, those which are necessary for achieving the observed fire spread [6]. For this purpose, an analytical steering strategy for the Calibration Stage was introduced in [6]. This approach shows good results reducing the error function and improving Calibration and Prediction Stages results. However, the main drawback exhibited by this steering method was its strong dependency to the underlying simulator. In order to overcome this penalty, we had developed a steering strategy called computational method, which has no dependence of the underlying fire simulator.

Next section will introduce the computational steering strategy main characteristics and, in subsequent sections, a comparative study of both analytical and computational steering strategies will be presented.

3.1 Computational Steering Strategy

The main advantage of the proposed computational steering strategy is its fire simulator independence. For this purpose, the underlying fire simulator is used as a black-box from which the only available information is the input/output data used/generated. Based on a complete set of fire spread information obtained from both real historical and synthetic fires, one generates a complete database of fire evolutions with the corresponding environmental conditions. This database information will be used for the DDDGA (Calibration stage) to discover the "ideal" wind values (wind speed and wind direction) based on a key search obtained from the real observed fire spread.

Figure 2 shows an example of how the DDDGA works under this computational steering strategy. Let's assume that the observed real fire at instant t_{i+1} exhibits a rate of spread equal to 20 fpm and a spread direction equal to 45° . Furthermore, the slope is known and corresponds to 45° (1 radian), and the observed fuel model is 7 [2]. The database register selected by the method is shown with a circle in the figure. This register has a wind speed equal to 9 fpm and its direction is 45° azimuth. These wind values will be used to define a subrange through the whole parameter valid range and when *mutation* takes place, the wind values will be assigned using a random value limited by the new subrange (taking into account database cases incompleteness).

In order to validate the experimental results when the proposed computational steering strategy is applied, we have compared its results (computational method results) against analytical steering strategy results. Although we expected that analytical method performs better than the computational strategy, we wanted to demonstrate that the proposed method could reach a good performance prediction despite of not being aware of the underlying simulator model used.

Taking into account that fire simulation is the most time consuming task,

model	slope	spread direction	speed of spread	distance	wind direction	wind speed
7	0.99	0.00	82.84	180.44	0.00	21.00
7	0.99	0.00	82.84	180.44	0.00	24.00
7	0.99	0.00	82.84	180.44	0.00	27.00
7	0.99	0.00	82.84	180.44	0.00	30.00
7	0.99	0.00	24.55	78.73	45.00	0.00
7	0.99	0.00	23.69	65.61	45.00	3.00
7	0.99	0.00	16.65	39.36	45.00	6.00
7	0.99	45.00	21.39	64.95	45.00	9.00
7	0.99	45.00	31.21	92.79	45.00	12.00
7	0.99	45.00	41.71	129.91	45.00	15.00

Fig. 2. Data stored in Computational Method data base.

we had proposed a master/worker solution for our parallel GA. Thus, master process holds the population, and it distributes the individuals among the worker processes and then, it receives the evaluated individuals (that means, each individual and its error). In turn, the worker processes receive a group of individuals (called chunk). For each individual of the received chunk, the worker process executes the fire spread simulation and evaluates the error function. When the worker finalizes with one chunk, it sends back to the master process the evaluated information. Then, the worker will keep waiting until another chunk of individuals is being received. This process is repeated until the whole population has been evaluated. Afterward, the master process applies genetic operators over the evaluated population in order to minimize the error value.

4 Experimental Results

As we had mentioned, two important key point in fire spread prediction are, on the one hand, to obtain the prediction results as fast as possible and, on the other hand, to provide simulation results as precise as possible. These two characteristics are essentials for having useful forest fire spread predictions. Thus, our experimentation covers these two topics: time reduction using parallel computing and error reduction when a dynamic data driven option is applied. Next sections describe how each experiment has been performed.

4.1 Parallel GA Performance Evaluation

The first experiment deals with application scalability. We had fixed a population size of 512 individuals (all generations have the same number of individuals), and we varied the number of workers from 1 to 31. A real map of 110 x 110 m^2 cells of $1m^2$ was used in this case, and we tested the Calibration Stage for a unique 8 minutes interval time. Figure 3 shows the time reduction when we execute the parallel Calibration Stage using computational steering method and no guided GA. In vertical axis we can see time (in seconds) and horizontal axis shows the number of workers used. The execution platform is a homogeneous PC cluster

composed by 32 nodes. This cluster uses a queue system in order to guarantee exclusive access to the required resources.

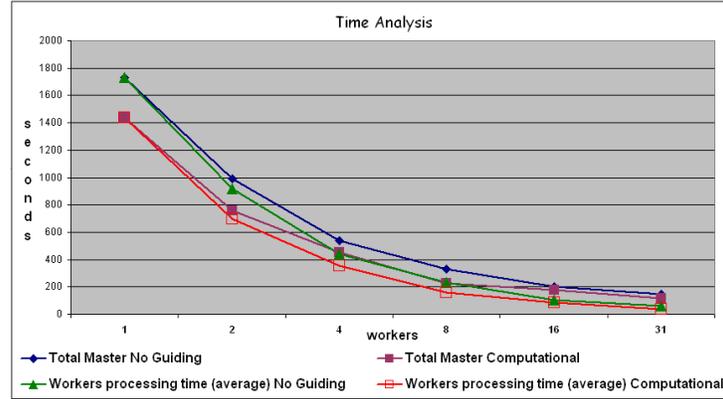


Fig. 3. Execution time varying the number of workers from 1 to 31.

The application described in this work deals with a real problem, and the results provided by the proposed prediction scheme could be used for improving fire fighting actions. The final users of this application may be institutions such as civil protection and fire fighters, usually, can not directly access to big computational resources during real time hazards. From the results shown in figure 3, we can see that our application has an appropriate scalability when the number of workers is increased. In the next section, we report the experiments performed in order to evaluate the proposed steering methods.

4.2 Dynamic Data Driven GA Evaluation

In this section we are going to evaluate the benefit of applying a dynamic data driven GA for fire spread prediction purposes. We are going to compare the analytical method with the computational method applying the prediction methodology without considering any external data. In this work, we present some results using one real fire and two synthetic fires (simulation results). Figure 4 shows the fire front evolution through time (2 minutes intervals) for the three experiments.

In all cases, we had used a populations size of 50 individuals with random values at the beginning. Each population was evolved 5 times (error reduction is insignificant after the first 3 or 4 evolutions). The depicted results are the average of 5 different executions for each case using different initial populations. The initial populations were created by random values but the remainder time lapses previous evolved populations (at Calibration Stage) were used. Moreover, we use the best individual obtained after executing the Calibration Stage as input in the Prediction Stage.

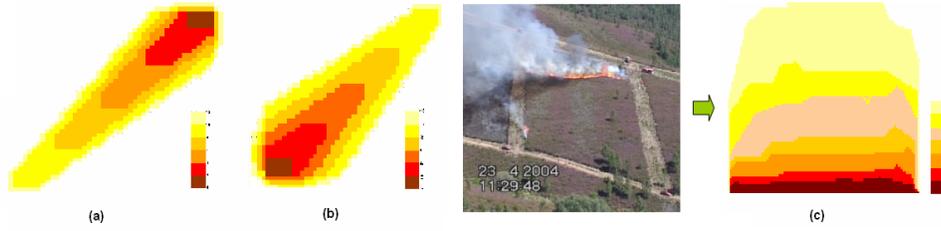


Fig. 4. (a) Experiment 1: synthetic fire case. (b) Experiment 2: synthetic fire case. (c) Experiment 3: real fire case.

Experiment 1: Synthetic Case Experiment 1 concerns with the map shown in figure 4(a) ($109.0 \times 89 \text{ m}^2$, cells of 1m^2). The terrain had 18° slope and the vegetation fuel model was set up to model 7 [2]. Figures 5(a) and (b) show the Calibration and Prediction Stages results respectively.

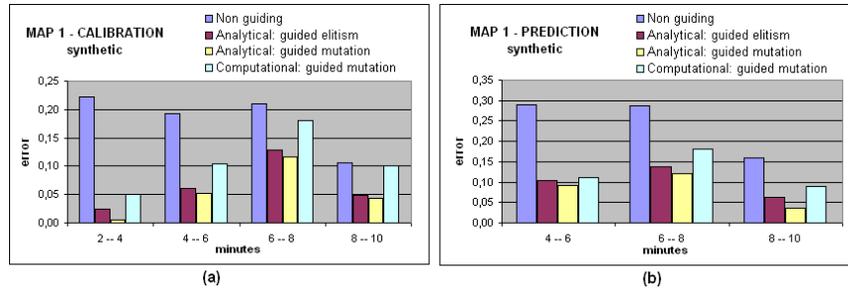


Fig. 5. Experiment 1: (a) Calibration Stage and (b) Prediction Stage results.

For both stages (Calibration and Prediction), the error (difference between the real fire spread and the simulated fire obtained) has been significantly reduced whatever dynamic data driven methods was used. Since one of our goals was to use the analytical method as a validation element of the computational method, we have analyzed in more detail each method behavior. From an immediate analysis of graphics 5(a) and 5(b), we detect a clear similarity along all data driven methods. However, it is also remarkable that analytical method provides better results than the computational method as it was expected. Nevertheless, the error difference between the two analytical methods and the computational method keeps, on average, bounded by 25% for all interval times, therefore, we can conclude that the analytical results validate the computational behavior. Prediction Stage results show errors slightly higher than Calibration Stage, however this is an expected result because we are using the best individual obtained at the Calibration Stage to obtain the prediction for a later time interval at the Prediction Stage.

Experiment 2: Synthetic Case Another synthetic map was used to carry out the second test (figure 4(b)), $33,22 \times 27,12 \text{ m}^2$ size map, cells of 1 f^2 . In this case, we consider 27° slope and the same vegetation as in the first burning case.

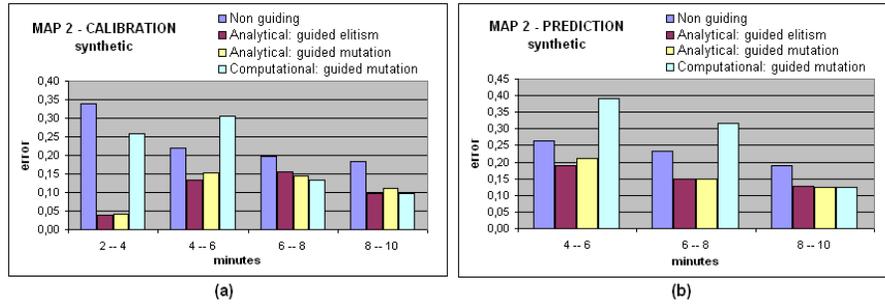


Fig. 6. Experiment 2: (a) Calibration Stage and (b) Prediction Stage results.

Figure 6 shows the second experiment results. At Calibration Stage (figure 6 (a)) we can see that analytical method reduces the error in all cases. However, computational method has an unexpected behavior in steps 2-4 and 4-6 at Calibration Stage and 4-6, 6-8 at Prediction Stage. In order to understand this unexpected behavior we analyze each individual execution for those particular situations.

At Calibration Stage we observed that when computational method was used, the resulting error values of each population were similar. However, when no steering method were applied, some populations had generated high errors and another ones had generated very small errors. Thus, total average was under influence of high errors as well as small ones. Since the Prediction Stage results depend on Calibration Stage best individual quality, the behavior observed in the Calibration Stage was reflected in the Prediction Stage. Therefore, when we apply any of the proposed methods, we are avoiding to depend on “lucky” of choosing good random values. It is important to take into account that in disasters problems having stable algorithms means an important improvement if we can still guarantee good results.

In order to determine benefits of applying any steering methods and taking into account calibration stage (where steering methods are applied), we compare total error for each method (no guiding total error means 100%). When computational method was applied, error reduction was about 15,9% approximately, analytical method (guided elitism option) error reduction was about 55,3% and analytical method (guided mutation option) error reduction was about 52,1% approximately (all lapses time average). For the same reasons of experiment 1, we note that computational method behavior is validated by analytical method behaviors.

Experiment 3: Real Case The last experiment is a real fire (figure 4(c)). The fire analyzed in this experiment corresponds to a plot of $89 \times 109 \text{ m}^2$, 1 m^2 cell size. The terrain was 18° slope and fuel type was equal 7. This burn has been extracted from a set of prescribe burns performed in the frame of an European project called SPREAD (Gestosa, 2004).

The results obtained for this experiment are shown in figure 7. The first 2 time intervals at Calibration Stage results were similar through all methods, there were not significant differences between different configurations, however, this stable behavior changes from that point to the end. We can see that in experiment 3, the errors were larger compared to the previous experiment errors. This error improvement could be considered as an expected behavior because of being dealing with a real fire instead of a synthetic fire. The spread behavior, in this case, was not uniform because of vegetation, variable wind conditions, the fire itself, and so on. This non uniform environment is harder to be reproduced by any fire spread simulator. In this case, error reduction was, on average, for the computational method 23,5%, for the analytical method (guided elitism option) about 13,5% and, finally, for the analytical method when mutation was guided is on 12,1%.

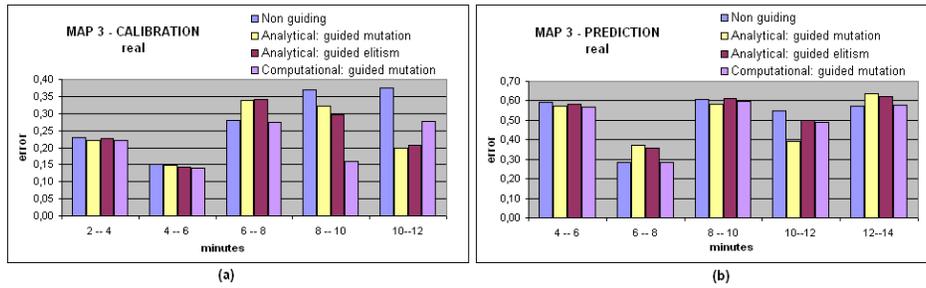


Fig. 7. Experiment 3: (a) Calibration Stage results and (b) Prediction Stage results.

5 Conclusions and Future Work

A Parallel Dynamic Data Driven Genetic Algorithm was proposed for forest fire spread prediction. This application deals with response time restrictions and prediction accuracy requirements. From the experimental results we could determine that our master/worker scheme was appropriate to take advantages of parallel computing in order to reduce the forest fire prediction response time.

On the other hand, in order to improve prediction accuracy, a Dynamic Data Driven GA was proposed where real fire progress was used for adapting the scenarios used by the method according to the observed real fire spread. A new steering method called Computational Steering method has been proposed,

which main feature is being independent on the underlying fire simulator becoming an universal method for calibrating the input parameters of any fire spread simulator.

Three cases of study were presented and their results had shown that the inclusion of the dynamic data driven systems bases in the Calibration Stage improves the quality of the propagation predictions. Furthermore, since the analytical and the computational methods have a similar behavior that is bounded by a constant difference (around 25%), we can conclude that the computational method behavior does not exhibit unexpected characteristics. Thus, computational method development is validated by the analytical method results through our specific domain.

Although our main objectives are real burning maps, synthetic cases help us to validate the proposed methods. These cases prove that dynamic data driven GA improves the final results by reducing search space and avoiding simulations with individuals that because of their characteristics provide low quality simulations.

References

1. Abdalhaq B.: A methodology to enhance the Prediction of Forest Fire Propagation. Ph.D Thesis. Universitat Autònoma de Barcelona (Spain). June 2004.
2. Anderson H. E.: Aids to Determining Fuel Models For Estimating Fire Behavior. Intermountain Forest and Range Experiment Station Ogden, UT 84401. General Technical Report INT.122. (1982)
3. Bevins C. D.: FireLib User Manual & Technical Reference. <http://www.fire.org> Accessed on January 2006.(1996)
4. 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. (2004)
5. Douglas, C. C.: Dynamic Data Driven Application Systems homepage. <http://www.dddas.org>. Accessed on October 2008.
6. Denham M., Cortés A., Margalef T., 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.
7. FIRE.ORG - Public Domain Software for the Wildland fire Community. <http://www.fire.org>. Accessed on May 2007.
8. Koza J.: Genetic Programming. On the programming of computers by means of natural selection. Massachusetts Institute of Technology. Cambridge, Massachusetts 02142. The MIT Press. (1992)
9. Mandel J., Beezley J., Bennethm L., Chakraborty S., Coen J., Douglas C., Hatcher J., Kim M. and Vodacek A.: A Dynamic Data Driven Wildland Fire Model. ICCS 2007, LNCS 4487, Springer Berlin / Heidelberg, pp. 1042-1049. (2007)
10. Rothermel R. C.: A mathematical model for predicting fire spread in wildland fuels. USDA FS, Ogden TU, Res. Pap. INT-115. (1972)