

Accelerating Wildland Fire Prediction on Cluster Systems*

Baker Abdalhaq, Ana Cortés, Tomás Margalef, and Emilio Luque

Departament d'Informàtica, E.T.S.E, Universitat Autònoma de Barcelona, 08193-Bellaterra
(Barcelona) Spain
baker@aows10.uab.es,
{ana.cortes,tomas.margalef,emilio.luque}@uab.es

Abstract. Classical prediction fire schemes do not match the real fire propagation, basically, because of the complexity of the physical models involved, the need for a great amount of computation and the difficulties of providing accurate input parameters. We describe an enhanced prediction scheme, which uses recent fire history and optimization techniques to predict near future propagation. The proposed method takes advantage of the computational power offered by distributed systems to accelerate the optimization process at real time.

1 Introduction

Wildland fire is an important hazard from the ecological, economical and social points of view. Considerable effort from different ambits (physics, biology, computer science...) are often joined in order to develop fire propagation models that can help fire prevention and mitigation. Classical prediction schemes rely on these fire propagation simulators, however, in most of cases, the results provided by simulation tools do not match real propagation. One of the most common sources of deviation in fire simulation spread from that of real-fire propagation is imprecision in input model parameters. A way of overcoming this input parameter uncertainty during real fires lies in exploiting to the limit all available data about the recent fire history in order to predict its near-future evolution [1]. We propose an enhanced prediction scheme, which, basically, works as follows. Once the initial fire line and a real propagation after a certain time interval are available, an optimization technique will be applied in order to determine a set of input parameters, providing the best match between simulation and real-fire behavior. These values will then be used to predict fire behavior in the next interval. Whilst feed-back from the field is supplied, this process is repeated so as to correct the prediction by using the available information. This enhanced prediction scheme reflects the dynamics of the fire environment (wind, moisture content, etc.), since the real data is updated periodically.

* This work has been supported by the *Comisión Interministerial de Ciencia y Tecnología* (CICYT) under contract TIC2001-2592 and by the European Commission under contract EVG1-CT-2001-00043 SPREAD.

Parameter optimization and prediction must be carried out faster than real-time propagation so that the prediction can be useful in deciding which actions need to be taken in tackling the emergency. However, the number of parameters is quite large and the resulting search space for the optimization strategy becomes enormous. It is not, therefore, feasible to assess the whole search space, which needs to be reduced by applying certain techniques. The current state of the art in the computational field offers the required background that should be applied. On the one hand, evolutionary computing is a well-established field with several techniques in the literature that are widely accepted (for example, Genetic Algorithms [2]). These techniques can be applied to guiding the search over the whole space, so that only certain cases are tested. On the other hand, computing systems based on parallel and distributed platforms offer the required computing power to apply these techniques and to provide successful results in an acceptable time. Typically, these techniques work in an iterative way by improving the obtained solution at each iteration. Therefore, a clear way of saving time consists of increasing the convergence speed of the optimization technique. For this purpose, we propose applying a sensitivity analysis to the input parameters in order to assess their impact on output and, consequently, to determine which parameters are worth spending time on tuning and which are not, maintaining the latter at an estimated value. In order to be more effective in tuning the most sensitive parameters, we also propose introducing a certain degree of knowledge during the optimization process. This knowledge will consist of limiting the range of the tuned parameters around an estimated value (which may be the real measurement) for those parameters.

The rest of this paper is organized as follows. Section 2 introduces the parallel implementation of the enhanced prediction scheme. Section 3 is devoted to the sensitivity analysis carried out. Section 4 reports on the experimental study carried out, and the results obtained. Finally, section 5 presents the main conclusions to this work.

2 Parallel Wildland Fire Prediction Method

The parallel enhanced prediction method is a step forward with respect to the classical methodology. It introduces the idea of applying an optimisation scheme to calibrate the set of input parameters of a given fire simulator, with the aim of finding an “optimal” set of inputs, which improves the results provided by the fire-spread simulator. Figure 1 shows how the proposed prediction method works. The FS box corresponds to the underlying fire simulator (ISStest in our case [3]). This simulator needs to be fed with some input parameters, which will form the input parameters set to be optimised. In our particular case, the number of input parameters required are 9. In section 3, we will describe in more detail each one of these parameters. In order to evaluate the goodness of the results provided by the simulator, a fitness function (FF box) has been included, which determines the degree of matching between the predicted fire line and the real fire line. Both FS and FF boxes will be evaluated for a large group of input parameter sets in order to exploit the inherent parallelism of some optimisation techniques (OPT box). The optimisation process will be repeated until a “good” solution is found or until a predetermined number of iterations has been reached. At that point, an “optimal” set of inputs should be found, which will be used

as the input set for the fire simulator, in order to obtain the position of the fire front in the very near future. Obviously, this process will not be useful if the time incurred in optimizing is superior to the time interval between two consecutive updates of the real-fire spread information. For this reason, it is necessary to accelerate the optimization process as much as possible.

For optimization purposes, we used the optimization framework called BBOF (Black-Box Optimization Framework) [10], which consists of a set of C++ abstract-based classes that must be re-implemented in order to fit both the particular function being optimized and the specific optimization technique. BBOF uses the master-worker programming paradigm working in an iterative scheme. In particular, since the evaluation of the function to be optimized for each guess is independent, the guesses can be identified as the work (tasks) done by the workers. The responsibility for collecting all the results from the different workers and for generating the next set of guesses by applying a given optimization technique will be taken on by the master process. For the purposes of master-worker communication, BBOF uses the MPICH (Message Passing Interface) library [9]. In our case, the master will execute a GA [2] as optimisation strategy and will also be in charge of distributing the guesses of the candidate solutions to the workers. The workers themselves will execute, the underlying fire-spread simulator and will also provide the prediction error (evaluation of the fitness function).

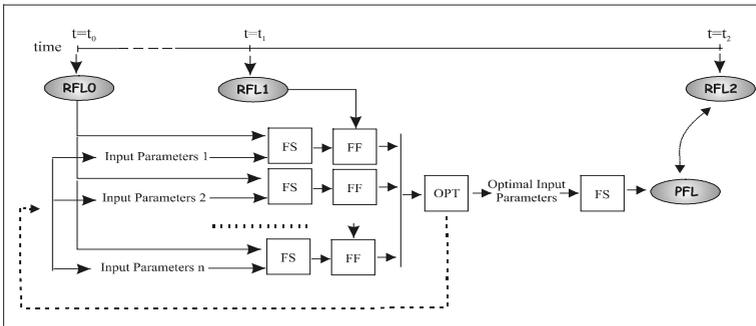


Fig. 1. Enhanced wildland fire prediction method on a master-worker scheme.

Since our objective consists of finding the combination of input parameters that minimizes the deviation of the simulator prediction from the real scenario as fast as possible, we need to compare the simulated fire lines against the real fire line and, according to the results of this comparison, assign a quality measurement to the underlying scenario. Each fireline describes a burned area. To compare the simulated and the real firelines we used the area of the XOR between the real and simulated burned areas (FF box). This XOR includes the areas that are burned in one of the propagations but not in the other one. If there is a perfect match between the real and simulated fire, this XOR is zero.

The optimization technique (GA) will be iterated until either a preset number of iteration is executed (1000 in our case). In each iteration a generation size of 20 individuals have been used.

3 Sensitivity Analysis

Sensitivity Analysis (SA) classically aims to ascertain how the model depends upon the information fed into it (input model/simulator parameters)[5]. The method we used here is based on nominal range sensitivity analyses, which are also known as local sensitivity analysis or threshold analysis [5]. Basic nominal sensitivity analysis evaluates the effect on the model output exerted by individually varying only one of the model inputs across its entire range of possible values, while holding all other inputs at their nominal or base-case values. The difference in the model output due to the change in the input variable is referred to as the sensitivity or swing weight of the model to that particular input variable, in that given case. However, there may be interdependencies among the parameters. The nominal sensitivity analysis must therefore be repeated for each parameter for all possible cases and combinations of all the other parameters.

The sensitivity of the parameters, in our case, depends on the fire propagation model used in the core of the objective function. For a generic study, we studied the effect of the parameters of the model in one dimension on propagation speed, thus the wind has only one scalar value, which is the speed of the wind in the direction of the fire propagation, however, in the experimental study both wind speed and direction has been considered because the fire simulator works with a two dimensional space.

Let V_{ik} be the effect of varying factor i from its minimum to its maximum (difference of the speed of the minimum and the speed of the maximum) at case k . The total effect of parameter i is defined as follows:

$$V_i = \sum V_{ik} \quad (1)$$

where k is all the possible cases (combinations of input factors). Thus, V_i will be our index of sensitivity for the parameter i . This index not only reflects the effect of the parameter but also the effect of its range. Higher parameter ranges mean greater uncertainty in the measurement of that parameter.

Table 1 outlines each one of these parameters and their corresponding minimum and maximum values according to [6], also showing the calculated index. Using the value of the index, we can classify input parameters by their sensitivity. This table shows that the two most important parameters are the load parameters (W0, β); the third is wind speed (U), followed by humidity (Mf). The parameters with weakest effect are metal content (St, Se) and heating content (h). This result concurs with the results obtained by [7], which also uses the Rothermel set of equations as a forest fire propagation model.

Table 1. Ranges used to calculate the sensitivity index

parameter	W0	β	σ	St	Se	Mx	Mf	h	U
Min	0,1	0,01	315	0,001	0,0001	0,1	0	18571429	0
Max	4	0,11	11500	0,08	0,07	0,4	Mx	22000000	15
Index	0,77	0,86	0,56	0,03	0,16	0,28	0,61	0,13	0,71

Since sensitivity analysis implies a high number of simulations, we have also used the master/worker programming paradigm to evaluate all sensitivity indexes.

4 Experimental Study

The experimental study was carried out on a Linux cluster composed of 21 PC's with Intel Celeron processor 433 MHz, each one with 32 MB RAM, connected to a Fast Ether Net 100 Mb.

To properly evaluate the XOR area after executing the ISSstest simulator for each guess, we need to have a reference fire line for comparison. For this purpose, a synthetic fire line was obtained by setting the values of all inputs to certain known values. We assumed homogeneous vegetation through the whole terrain, which consisted of a flat area. Once the synthetic real fire line was obtained, it was dismissed and was only used as a comparative member for calculating the XOR during the optimization process.

As we have commented, a GA was used as the optimization technique. Since the genetic algorithm has random factors, the global optimization process was performed 10 times and all results were averaged. All the values reported in this study therefore correspond to the mean values of the corresponding 10 different experiments conducted.

As we have commented, we are interested in calibrating the parameters that have a greater sensitivity index while we do not know their real values. Since calibrating the parameters that have little effect on the result will not improve the simulator results significantly, it is not worth to spent processing time on tuning them. We suppose that fewer parameters to be optimized will make the convergence faster and, at the same time, fixing certain unimportant parameters to a given value with a reasonable error will not deviate the optimization process too far from the global minimum, because of the reduction of the search space size.

This experiment is designed to observe the effect of removing the parameters that have a small sensitivity index on the convergence of the optimization process.

As estimated values for the parameters that are to be fixed, we have their real values plus 10% of its full range. Table 2 shows the real value of the less sensitive parameters, and their corresponding estimated values, when applying this estimation error (10%).

Table 2. The real and estimated values of the fixed parameters.

Parameter	Real value	Estimated value
St	0.04	0.04799
h	18971429	19314270
Se	0.02	0.0269
Mx	0.3	0.33

Figure 2(a) shows the convergence of the optimizing process by reducing the number of optimized parameters. Each curve differs from the other by omitting one parameter each time. We applied statistical hypothesis testing [8] to the results in order to asses whether or not the two observed behaviors can be considered statistically different. We can observe in figure 3(a) that at iteration 500, in general, for each case (fixing certain number of parameters) the corresponding mean value lays inside the confidence intervals of the other cases. That means that there is no statistical evidence that the differences between the means due to fixing some parameters at that iteration are relevant. However, figure34(b) shows that the means

have a trend to diverge one from the other at iteration 1000. In the same way we have applied the statistical hypotheses testing for the iterations 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000. We found that there is no statistical difference between the means before iteration 500; consequently, it is irrelevant to discuss the behavior of the curves during the first phase of the optimization process. However, at iteration 1000, the results show a statistical difference between optimizing all parameters as opposed to fixing 1, 2, 3 and 4 parameters because the mean value for the case '10 parameters' clearly lays out of the other confidence intervals. Furthermore, we can observe that, as the number of parameters fixed increases, a statistical difference also appears (i.e., 8 and 9 versus 6 and 7). The mean values of the objective function (XOR area) at the end of the optimization process (iteration 1000) is shown in figure 2(b). As we can see, the objective function for the case of '6 parameters' is one third of the mean value obtained for the case of '10 parameters'. Table 3 shows the search space reduction for the cases plotted in figure 2(a). As we can observe, the search space reached a size reduction of one third from the '10 parameters' case to the '6 parameters' case. This reduction is directly related to the improvement obtained in the objective function (one third, approximately, in both cases).

Therefore, since the statistical study has shown that the convergence improvement for the case of '6 parameters' is not a matter of chance, we can conclude that reducing the search space is a good policy to speed up the optimization convergence.

However, we should bearing in mind that, these results are obtained using an error of estimation equal to 10%. If the error is greater, the practice of fixing the value of the parameters to estimated values will not be good. This method therefore assumes a "good" estimation of the real parameter value.

Table 3. Search space size for different number of parameters.

N° parameters	10	9	8	7	6
Search space size	1E+77	1E+69	1E+61	1E+53	1E+45

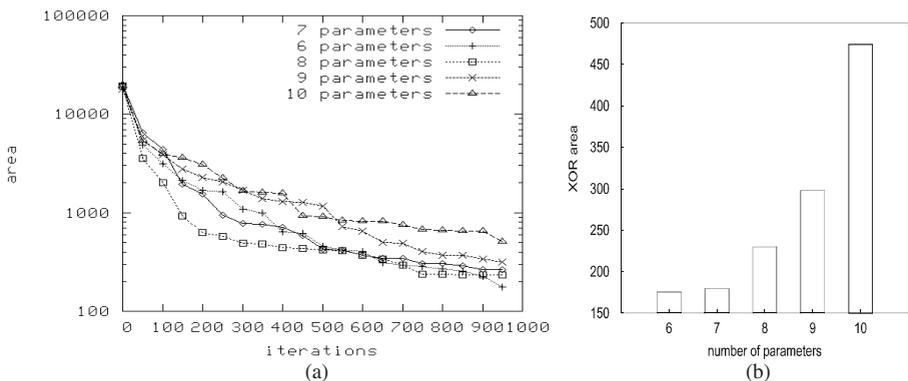


Fig. 2. Optimization Convergence Changing the Number of Parameters (a) and the mean value of the XOR area at iteration 1000 (b)

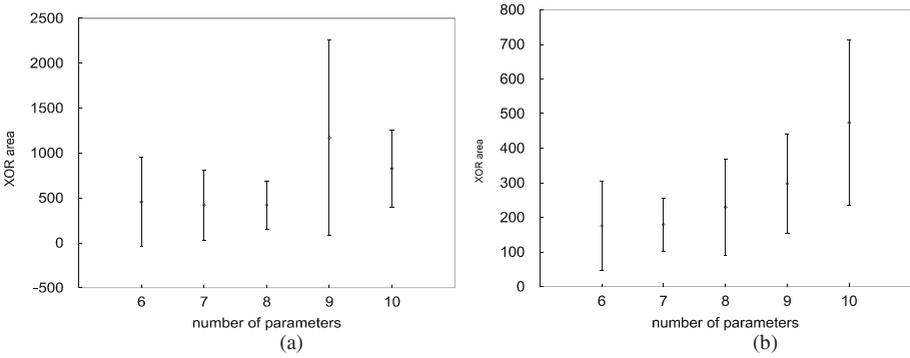


Fig. 3. 95% confidence intervals at iteration 500 (a) and 1000 (b).

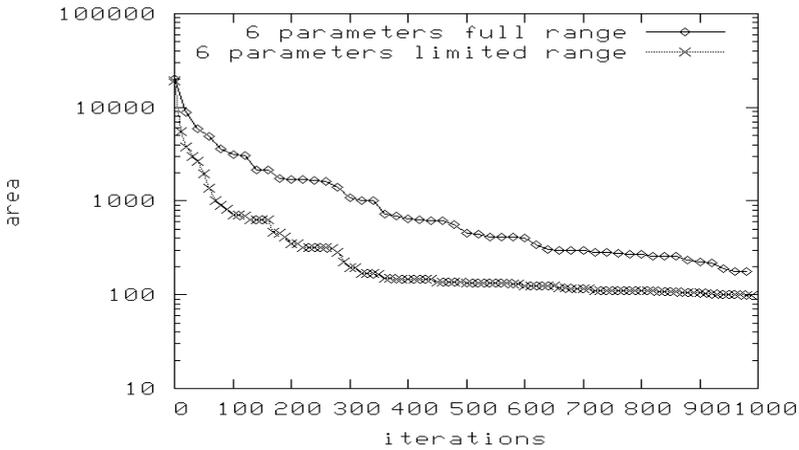


Fig. 4. Optimization convergence comparison using both the full and limited ranges.

Once we have observed that fixing 4 parameters to a certain estimated value provides a considerable improvement in optimization convergence, we focus on this case to introduce a certain degree of knowledge of the optimized parameters in order to further improve such convergence. We assume that we have some knowledge about the limits within which a parameter can vary, therefore it is not necessary to search within its full possible range. For the purpose of this experiment, we limited the range of the parameter to 15% above and below its “known value” so as to simulate the expected range. Figure 4 shows the optimization convergence when optimizing 6 parameters using either their full range or a limited searching range. As we can observe, cutting the range of the parameters significantly accelerates optimization convergence. Although from the figure it seems that, at iteration 1000, both situations provide similar results, the limited range at the end of the optimization process provides an objective function (XOR area) equal to 98.71, on average, whereas the final value is 175.47, using the full range.

5 Conclusions

One of the most common sources of fire spread simulation deviation from real fire propagation is imprecision in input simulator parameters. This problem can be approached by applying an evolutionary optimization such as the Genetic Algorithm so as to calibrate input-simulator parameters. Since this approach is a time-demanding task, we have proposed a global sensitivity analysis to accelerate optimization convergence. This technique reduces the search space screened by fixing the less sensitive parameters to an estimated value and by focusing optimization on the most sensitive parameters. We have also reduced the range of each optimized parameter by introducing some degree of knowledge of each of them. This was considered by limiting the variation of these parameters around a known value (field measurement). Both techniques were carried out on a Linux cluster composed of 21 PC's. We used a master/worker programming paradigm, where the master and worker processes communicate with each other using MPI. The results show that, combining both accelerating strategies, the convergence improvement obtained is quite significant.

References

1. Baker Abdalhaq, Ana Cortés, Tomàs Margalef, Emilio Luque, "Optimization of Fire Propagation Model Inputs: A Grand Challenge Application on Metacomputers". LNCS 2400, pp. 447-451. (2002).
2. Coley David A.: "An Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific, 1999.
3. Jorba J., Margalef T., Luque E., J. Campos da Silva Andre, D. X Viegas "Parallel Approach to the Simulation Of Forest Fire Propagation". Proc. 13 Internationales Symposium "Informatik fur den Umweltschutz" der Gesellschaft Fur Informatik. pp. 69-81, (1999)
4. Rothermel, R. C., "A mathematical model for predicting fire spread in wildland fuels", USDA FS, Ogden TU, Res. Pap. INT-115, (1972).
5. Satelli, A., K. Chan, M. Scott, Editors. "Sensitivity analysis". John Wiley & Sons publishers, Probability and Statistics series. (2000).
6. André, J. "Uma Teoria Sobre a propagação de Frentes de Fogos Florestais de Superficie". PhD. Dissertaion. Coimbra. (1996).
7. Salvador, R., Piñol, P, Tarantola, S. And Pla, E. "Global Sensitivity Analysis and Scale Effects of a Fire Propagation Model used Over Mediterranean Shrub lands". Elsevier, Ecological Modelling 136 pp. 175-189, (2001).
8. Wadsworth, Harrison M. "Handbook of statistical methods for engineers and scientists", McGraw.Hill, Inc. (1990).
9. W. Gropp and E. Lusk and N. Doss and A. Skjellum, "A high-performance, portable implementation of the MPI message passing interface standard", "Parallel Computing", volume22-6, pp.789-828, sep,1996.
10. Baker Abdalhaq, Ana Cortés, Tomàs Margalef, Emilio Luque: "Evolutionary Optimization Techniques on Computational Grids.", International Conference on Computational Science, pp. 513-522, (1) 2002.