Optimization of Fire Propagation Model Inputs: A Grand Challenge Application on Metacomputers*

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. Forest fire propagation modeling has typically been included within the category of grand challenging problems due to its complexity and to the range of disciplines that it involves. The high degree of uncertainty in the input parameters required by the fire models/simulators can be approached by applying optimization techniques, which, typically involve a large number of simulation executions, all of which usually require considerable time. Distributed computing systems (or metacomputers) suggest themselves as a perfect platform to addressing this problem. We focus on the tuning process for the *ISStest* fire simulator input parameters on a distributed computer environment managed by Condor.

1 Introduction

Grand Challenge Applications (GCA) address fundamental computation-intensive problems in science and engineering that normally involves several disciplines. Forest fire propagation modeling/simulation is a relevant example of GCA; it involves several features from different disciplines such as meteorology, biology, physics, chemistry or ecology. However, due to a lack of knowledge in most of the phases of the modeling process, as well as the high degree of uncertainty in the input parameters, in most cases the results provided by the simulators do not match real fire propagation and, consequently, the simulators are not useful since their predictions are not reliable. One way of overcoming these problems is that of using a method external to the model that allows us to rectify these deficiencies, such as, for instance, optimization techniques. In this paper, we address the challenge of calibrating the input values of a forest fire propagation simulator on a distributed computing environment managed by Condor [1] (a software system that runs on a cluster of workstations in order

* This work has been supported by MCyT-Spain under contract TIC2001-2592, by the EU under contract EVG1-CT-2001-00043 and partially supported by the Generalitat de Catalunya- Grup de Recerca Consolidat 2001SGR-00218. This research is made in the frame of the EU Project SPREAD - Forest Fire Spread Prevention and Mitigation.

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 447–451.

to harness wasted CPU cycles from a group of machines called a Condor pool). A Genetic Algorithm (GA) scheme has been used as optimization strategy. In order to evaluate the improvement provided by this optimization strategy, its results have been compared against a Pure Random Search.

The rest of this paper is organized as follows. In section 2, the main features of forest fire propagation models are reported. Section 3 summarizes the experimental results obtained and, finally, section 4 presents the main conclusions.

2 Forest Fire Propagation Model

Classically, there are two ways of approaching the modeling of forest fire spread. These two alternatives essentially differ from one other in their degree of scaling. On one hand, we refer to local models when one small unit (points, sections, arcs, cells, ...) is considered as the propagation entity. These local models take into account the particular conditions (vegetation, wind, moisture, ...) of each entity and also of its neighborhood in order to calculate its evolution. On the other hand, as a propagation entity, global models consider the fire line view as a whole unit (geometrical unit) that evolves in time and space.

The basic cycle of a forest fire simulator involves the execution of both local and global models. On the basis of an initial fire front and simulating the path for a certain time interval, the result expected from the simulator is the new situation of the real fire line, once the said time has passed. Many factors influence the translation of the fire line. Basically, these factors can be grouped into three primary groups of inputs: vegetation features, meteorological and topographical aspects. The parameter that possibly provides the most variable influence on fire behavior is the wind [2]. The unpredictable nature of wind caused by the large number of its distinct classes and from its ability to change both horizontal and vertical direction, transforms it into one of the key points in the area of fire simulation. In this work, we focus on overcoming wind uncertainty regardless of the model itself and of the rest of the input parameters, which are assumed to be correct. The *ISStest* forest fire simulator [3], which incorporates the Rothermel model [4] as a local model and the global model defined by André and Viegas in [5], has been used as a working package for forest fire simulation.

3 Experimental Study

The experiments reported in this section were executed on a Linux cluster composed of 21 PC´s connected to a Fast Ether Net 100 Mb. All the machines were configured to use NFS (Network File System) and the Condor system; additionally, PVM were installed on every machine. The *ISStest* forest fire simulator assumes that the wind remains fixed during the fire-spread simulation process; consequently, it only considers two parameters in quantifying this element: wind speed (w_s) and wind direction

(w_d). We refer to the two-component vector represented by $\theta = (w_s \quad w_d)$ as a *static* wind vector. However, in order to be more realistic, we have also considered a different scenario where in which the wind vector changes over time. The new wind vector approach will be referred to as a *dynamic* wind vector and is represented as follows:

$$\theta = \left(w_{s0} \quad w_{d0} \quad w_{s1} \quad w_{d1} \quad w_{s2} \quad w_{d2} \quad \cdots \quad \cdots \quad w_{s(t-1)} \quad w_{d(t-1)}\right) \tag{1}$$

where t corresponds to the number of wind changes considered. In order to tune these values as closely as possible to their optimum values, a Genetic Algorithm (GA) [6] as optimization technique has been applied. We also conducted the same set of experiments using a Pure Random approach to optimize the *wind* vector parameters in order to have a reference point for measuring the improvement provided by GA. The real fire line, which was used as a reference during the optimization process, was obtained in a synthetic manner for both the *static* and *dynamic* scenarios. Furthermore, we used the Hausdorff distance [7], which measures the degree of mismatch between two sets of points, in our case the real and simulated fire line, to measure the quality of the results.

For optimization purposes, the Black-Box Optimization Framework (BBOF) [8] was used. BBOF was implemented in a plug&play fashion, where both the optimized function and optimization technique can easily be changed. This optimization framework works in an iterative fashion, moving step-by-step from an initial set of guesses about the vector θ to a final value that is expected to be closer to the optimal vector of parameters than were the initial guesses. This goal is achieved because, at each iteration (or evaluation) of this process, the preset optimization technique (GA or Pure Random) is applied to generate a new set of guesses that should be better than the previous set.

We will now outline some preliminary results obtained on both the *static* and *dynamic* wind vector scenarios.

3.1 Static Wind Vector

As is well known, GA's need to be tuned in order to ensure maximum exploitation. Therefore, previous to the fire simulation experimental study, we conducted a tuning process on the GA, taking into account the particular characteristics of our problem. Since the initial set of guesses used as inputs by the optimization framework (BBOF) were obtained in a random way, we conducted 5 different experiments and the corresponding results were averaged. Table 1 shows the Hausdorff distance, on average, obtained for both strategies (GA and Random). As can be observed, GA provides considerable improvement in results compared to the case in which no optimization strategy has been applied.

In the following section, we will outline some preliminary results obtained on the *dynamic* wind vector scenario.

Table 1. Final Haussdorf distance (m.) obtained by GA and a Pure Random scheme under the static wind vector scenario.

Algorithm	Genetic	Random
Hausdorff dist. (m)	11	147,25
Evaluations	200	200

3.2 Dynamic Wind Vector

Two different experiments were carried out in order to analyze the *dynamic* wind vector scenario. In the first study, the wind changes were supposed to occur twice, the first change after 15 minutes with the second change coming 30 minutes later. Therefore, the vector to be optimized will include 4 parameters and is represented by: $\theta = (w_{s1} \quad w_{d1} \quad w_{s2} \quad w_{d2})$. In the second case, three change instants have been considered, each separated from the next by 15 minutes. Consequently, the vector to be optimized will be: $\theta = (w_{s1} \quad w_{d1} \quad w_{s2} \quad w_{d2} \quad w_{s3} \quad w_{d3})$. In both cases, the optimization process was run 5 times with different initial sets of guesses and, for each one, 20000 evaluations had been executed. Table 2 shows the Hausdorff distance, on average, for GA and Random strategies and for both dimensions setting for the *dynamic* wind vector. We observe that the results obtained when the vector dimension is 6 are worse than those obtained for dimension 4. Although the number of evaluations has been increased by two orders of magnitude with respect to the experiment performed when the wind vector was considered as *static*, the results are considerably poorer in the case of the *dynamic* wind vector.

As can be observed in table 2, GA provides a final Hausdorff distance, on average, which, in the case of a tuned vector composed of 4 components, is five times better than that provided by the Random approach, which represents the case in which no external technique is applied. In the other tested case (6 components), we also observed improvements in the results. Therefore, and for this particular set of experiments, we have determined that GA is a good optimization technique in overcoming the uncertainty input problem presented by forest fire simulators. Since the improvement shown by this approach is based on the execution of a large number of simulations, the use of a distributed platform to carry out the experiments was crucial.

Table 2. Final Haussdorf distance (m.) obtained by GA and a Pure Random scheme under the dynamic wind vector scenario for 4 and 6 vector dimensions and after 20000 objective function evaluations

Parameters	4	6
Random	97.5	103.5
Genetic	18.8	84.75

4 Conclusions

Forest fire propagation is evidently a challenging problem in the area of simulation. Uncertainties in the input variables needed by the fire propagation models (temperature, wind, moisture, vegetational features, topographical aspects...) can play a substantial role in producing erroneous results, and must be considered. For this reason, we have provided optimization methodologies to adjust the set of input parameters for a given model, in order to obtain results that are as close as possible to real values.

In general, it has been observed that better results are obtained by the application of some form of optimization technique in order to rectify deficiency in wind fields, or in their data, than by not applying any method at all. The method applied in our experimental study was that of GA. In the study undertaken, we would draw particular attention to that fact that, in order to emulate the real behavior of wind once a fire has started, and in order to attain results that can be extrapolated to possible future emergencies, a great number of simulations need to be carried out. Since these simulations do not have any response-time requirements, these applications are perfectly suited to distributed environments (metacomputers), in which it is possible to have access to considerable computing power over long periods of time

References

1. M. Livny and R. Raman. High-throughput resource management. In Ian Foster and Carl Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kauffmann, (1999)
2. Lopes, A.,: Modelaçao numérica e experimental do escoamento turbulento tridimensional em topografia complexa: aplicaçao ao caso de um desfiladeiro, PhD Dissertation, Universidade de Coimbra, Portugal, (1993)
3. Jorba J., Margalef T., Luque E., J. Campos da Silva Andre, D. X Viegas: Parallel Approah to the Simulation Of Forest Fire Propagation. Proc. 13 Internationales Symposium "Informatik fur den Umweltshutz" der Gesellshaft Fur Informatik (GI). Magdeburg (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. André, J.C.S. and Viegas, D.X.,: An Unifying theory on the propagation of the fire front of surface forest fires, Proc. of the 3nd International Conference on Forest Fire Research. Coimbra, Portugal, (1998).
6. Baeck T., Hammel U., and Schwefel H.P.: Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation, Vol. 1, num.1 (April 1997) 3–17
7. Reiher E., Said F., Li Y. and Suen C.Y.: Map Symbol Recognition Using Directed Hausdorff Distance and a Neural Network Classifier. Proceedings of International Congress of Photogrammetry and Remote Sensing, Vol. XXXI, Part B3, Vienna, (July 1996) 680–685
8. Abdalhaq B., Cortés A., Margalef T. and Luque E.: Evolutionary Optimization Techniques on Computational Grids, In Proceeding of the 2002 International Conference on Computational Science LNCS 2329, 513-522