Universitat Autònoma de Barcelona

Escola d Enginyeria

Departament d Arquitectura de Computadors i Sistemes Operatius

Multi-core Hybrid Architectures Applied to Forest Fire Spread Prediction

Thesis submitted by Tomàs Artés Vivancos in fulfillment of the requirements for the doctoral degree from Universitat Autònoma de Barcelona advised by Dr. Ana Cortés Fité.

Barcelona (Spain), July 23th, 2015

Multi-core Hybrid Architectures Applied to Forest Fire Spread Prediction

Thesis submitted by Tomàs Artés Vivancos in fulfillment of the requirements for the doctoral degree from Universitat Autònoma de Barcelona. This work has been developed under RD 1393/2007 (adapted to RD 99/2011) in the High Performance Computing doctoral program and presented to the Computer Architecture & Operating Systems Department at the Escola d Enginyeria of Universitat Autònoma de Barcelona. This thesis was advised by Dr. Ana Cortés Fité.

Advisor signature Author signature

Acknowledgements

Gracias Laura, por empezar una nueva familia con un estudiante de doctorado a pesar de lo que ello conlleva. A ella agradezco cada una de sus acciones para ayudarme y ensenarme durante todos estos anos, nunca los olvidaré. Ella me convenció a adoptar a Tedi, que sin sus carinosos y felinos aranazos matinales esta tesis no hubiese sido posible. Aún me pregunto quién le puso ese nombre y el porqué, Tedi. También debo agradecer la aportación de la tortuga sin nombre que tengo a mi lado, que hace ruido al chocar contra los cristales de la tortuguera manteniéndome despierto en momentos como este.

Esta tesis no se habría llevado a cabo sin mi familia. Y no lo digo porque es lo que toca en esta situación. Gracias a mis padres, hermanos, tíos, abuelos, etc. que me han orientado y alejado de mi sueno de infancia, ser vendedor ambulante de pizzas. Querido lector, créame cuando le digo que mi familia y mi pareja han hecho mucho por mi. Todos ellos no han parado de darme ejemplo desde que los conozco, y doy fe de que no se dan por vencidos.

Ahora que ya sabe como he podido acabar realizando estudios universitarios y porque esta tesis, a su pesar, no es un conjunto de apetitosos ingredientes mediterráneos, creo conveniente que sea consciente del soporte recibido durante la elaboración de la misma.

Este trabajo se ha realizado dentro de un grupo de investigación dirigido por Ana y Tomàs. Ambos me han introducido en el intrincado campo de la investigación. Imagine el esfuerzo que han realizado para dar lugar al presente trabajo y aliarse con mi familia para mantenerme alejado de la venta ambulante de pizzas. Muchas gracias a todos los miembros del *Fire Team* por cada uno de los esfuerzos que me han dedicado.

Es bien conocida la importancia de un buen ambiente laboral. Aunque en el departamento de la universidad no tengamos toboganes, sala de juegos o una fábrica de chocolate, gozamos de un excelente ambiente laboral gracias a lo mas importante, las personas. Por este motivo, agradezco a todos los profesores, técnicos, administrativos, *CAOS Girls* y a la *PIF Crew* cada momento agradable al que han contribuido con su esfuerzo e iniciativa. Cabe destacar también, la gran aportación que hacen aquellos estudiantes de doctorado que vienen de otros países. A menudo nos olvidamos de lo difícil que es estar alejado de tu tierra natal aguantando estupideces de otros estudiantes indígenas del departamento y mantener un carácter agradable y positivo.

Muchas gracias a todos los amigos que hicieron mi estancia de doctorado más amena, ya sea por hacerme reir a distancia o por su agradable visita a Italia.

I would like to express my gratitude to the Joint Research Centre and the Institute for Environment and Sustainability for give me the chance to work with them during the PhD. stay. Specially, those who made my stay more comfortable, constructive and pleasant.

Abstract

Large forest fires are a kind of natural hazard that represents a big threat for the society because it implies a significant number of economic and human costs. To avoid major damages and to improve forest fire management, one can use forest fire spread simulators to predict fire behaviour. When providing forest fire predictions, there are two main considerations: accuracy and computation time.

In the context of natural hazards simulation, it is well known that part of the final forecast error comes from uncertainty in the input data. For this reason several input data calibration methods have been developed by the scientific community. In this work, we use the Two-Stage calibration methodology, which has been shown to provide good results. This calibration strategy is computationally intensive and time-consuming because it uses a Genetic Algorithm as an optimization strategy. Taking into account the aspect of urgency in forest fire spread prediction, we need to maintain a balance between accuracy and the time needed to calibrate the input parameters. In order to take advantage of this technique, we must deal with the problem that some of the obtained solutions are impractical, since they involve simulation times that are too long, preventing the prediction system from being deployed at an operational level.

This PhD Thesis exploits the benefits of current multi-core architectures with the aim of accelerating the Two-Stage forest fire prediction scheme being able to deliver predictions under strict real time constraints. For that reason, a Time-Aware Core allocation (TAC) policy has been defined to determine in advance the more appropriate number of cores assigned to a given forest fire spread simulation. Each execution configuration is obtained considering the particular values of the input data needed for each simulation by applying a dynamic decision tree. However, in those cases where the optimization process will drive the system to solutions whose simulation time will prevent the system to finish on time, two different enhanced schemes have been defined: Re-TAC and Soft-TAC. Re-TAC approach deals with the resolution of the simulation. In particular, Re-TAC finds the minimum resolution reduction for such long simulations, keeping accuracy loss to a known interval. On the other hand, Soft-TAC considers the GA s population size as dynamic in the sense that none individual will be killed for over passing the internal generations deadline, but it will be keep executing and the population size for the subsequent GA s generation is modified according to that.

All proposed prediction strategies have been tested with past real cases obtaining satisfactory results both in terms of prediction accuracy and in the time required to deliver the prediction.

Keywords

Forest fire spread prediction, model coupling, wind field model, multi-core systems, MPI, OpenMP, hybrid applications.

Resumen

Los incendios forestales son un tipo de catástrofe natural que representa un gran reto para sociedad debido a sus elevados costes económicos y humanos. Con el objetivo de evitar los costes derivados de dicho desastre natural y mejorar la extinción de éstos, los simuladores de propagación de incendios se pueden utilizar para intentar anticipar el comportamiento del incendio y ayudar a conseguir una extinción del incendio más eficiente y segura. Cuando se propociona una predicción de la propagación de un incendio forestal existen dos elementos clave: la precisión y el tiempo necesario para computar la predicción.

Bajo el contexto de la simulación de desastres naturales, es bien conocido que parte del error de la predicción está sujeta a la incertidumbre en los datos de entrada utilizados por el simulador. Por esta razón, la comunidad científica ha creado distintos métodos de calibración para reducir la incertidumbre de los datos de entrada y así mejorar el error de la predicción. En este trabajo se utiliza una metodología de calibración basada en dos etapas que ha sido probada en trabajos previos con buenos resultados. Este método de calibración implica una necesidad considerable de recursos computacionales y eleva el tiempo de cómputo debido al uso de un Algoritmo Genético como método de búsqueda de los mejores datos de entrada del simulador. Se debe tener en cuenta las restricciones de tiempo bajo las que trabaja un sistema de predicción de incendios. Es necesario mantener un equilibrio adecuado entre precisión y tiempo de cómputo utilizado para poder proporcionar una buena predicción a tiempo. Para poder utilizar la técnica de calibración mencionada, se debe solucionar el problema que representa que algunas soluciones sean inviables debido a que implican tiempos de ejecución muy largos, lo que puede impedir que se pueda dar respuesta a su debido tiempo en un supuesto contexto operacional.

La presente Tesis Doctoral utiliza las arquitecturas multi-core con el objetivo de acelerar el método de calibración basado en dos etapas y poder proporcionar una predicción bajo tiempos de entrega que se darían en un contexto real. Por esta razón, se define una política de asignación de núcleos basada en el tiempo disponible de ejecución . Esta política de asignación asignará un número determinado de recursos a una determinada simulación previamente a ser ejecutada. La política de asignación se basa en árboles de decisión creados con los parámetros de simulación utilizados. Sin embargo, se proponen dos métodos para aquellos casos donde el algoritmo genético tienda a crear individuos cuyo tiempo de ejecución provocan que sea imposible acabar la calibración a tiempo: Re-TAC y Soft-TAC. La propuesta ReTAC utiliza la resolución de las simulaciones para solucionar el problema. En concreto, Re-TAC trata de encontrar la mínima reducción de la resolución que permita que aquellas simulaciones que son demasiado largas puedan ser ejecutadas manteniendo la precisión bajo control. Por otro lado, Soft-TAC utiliza poblaciones de tamano dinámico. Es decir, los individuos no se matan al alcanzar el límite de timepo de ejecución asignado a una generación del Algoritmo Genético, sino que se permite la ejecución simultanea de individuos de distintas generaciones haciendo que el tamano de la población sea dinámico.

Todas la estrategias de predicción propuestas han sido probadas con casos reales obteniendo resultados satisfactorios en términos de precisión y de tiempo de cómputo utilizado.

Palabras clave

Predicción de propagación de incendios forestales, acoplamiento de modelos, modelador de vientos, sistemas multi-core, MPI, OpenMP, aplicaciones híbridas.

Resum

Els incendis forestals són un tipus de desastre natural que representa un gran repte per a la societat a causa dels seus elevats costos econòmics i humans. Amb l objectiu d evitar els costos derivats d aquest desastre natural i millorar l extinció dels mateixos, els simuladors de propagació d incendis es poden utilitzar per intentar anticipar el comportament de l incendi i ajudar a aconseguir una extinció de l incendi més eficient i segura. Quan es propociona una predicció de la propagació d un incendi forestal existeixen dos elements claus: la precisió i el temps necessari per computar la predicció.

Sota el context de la simulació de desastres naturals, és ben conegut que part de l error de la predicció està subjecta a la incertesa en les dades d entrada utilitzades pel simulador. Per aquesta raó, la comunitat científica ha creat diferents mètodes de calibratge per reduir la incertesa de les dades d entrada i així millorar l error de la predicció. En aquest treball s utilitza una metodologia de predicció basada en dues etapes que ha estat provada en treballs previs amb bons resultats. Aquest mètode de calibratge implica una necessitat considerable de recursos computacionals i eleva el temps de còmput a causa de l ús d un Algorisme Genètic com a mètode de cerca de les millors dades d entrada del simulador. S ha de tenir en compte les restriccions de temps sota les quals treballa un sistema de predicció d incendis. És necessari mantenir un equilibri adequat entre precisió i temps de còmput utilitzat per poder proporcionar una bona predicció a temps. Per poder utilitzar la tècnica de calibratge esmentat, s ha de solucionar el problema que representa que algunes solucions siguin inviables ja que impliquen temps d execució molt llargs, fet que pot impedir que es pugui donar resposta a temps en un suposat context operacional.

La present Tesi Doctoral utilitza les arquitectures multi-core amb l objectiu d accelerar el mètode de predicció basat en dues etapes per poder proporcionar una predicció sota temps de lliurament que es donarien en un context real. Per aquesta raó, es defineix una política d assignació de nuclis basada en el temps disponible d execució. Aquesta política d assignació assignarà un nombre determinat de recursos a una determinada simulació prèviament a ser executada. La política d assignació es basa en arbres de decisió creats amb els paràmetres de simulació utilitzats. No obstant això, es proposen dos mètodes per a aquells casos on l Algorisme Genètic tendeix a crear individus el temps d execució dels quals provoquen que sigui impossible acabar el calibratge a temps: ReTAC i SoftTAC. La proposta ReTAC utilitza la resolució de les simulacions per solucionar el problema. En concret, ReTAC tracta de trobar la mínima reducció de la resolució que permeti que aquelles simulacions que són massa llargues puguin ser executades mantenint la precisió sota control. D altra banda, SoftTAC utilitza poblacions de grandària dinàmica. És a dir, els individus no es maten en arribar al límit de temps d execució assignat a una generació de l AG, sino que es permet l execució simultanea d individus de diferents generacions de l algorisme genètic.

Totes les estratègies de predicció proposades han estat provades amb casos reals obtenint resultats satisfactoris en termes de precisió i de temps de còmput utilitzat.

Paraules clau

Predicció de la propagació d incendis forestals, acoblament de models, modelador de vents, sistemes multi-core, MPI, OpenMP, aplicacions híbrides.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Forest fires in Europe

Wildland fires are a kind of natural hazard that every year causes environmental, economical and human losses. In the European continent, forest fires show a clear seasonal behaviour and an occurrence pattern that affects the sames zones of Europe every year. Figure 1.1 shows the number of forest fires occurrences since 2009 in Europe, sorted by country and month of the occurrence. As can be observed, the main fire season starts around April or May and the critical period of the season goes from June to September. The countries with the higher number of forest fires are those located at the Mediterranean area such as Portugal, Spain, Greece, Italy and, in north Africa, Algeria. Another way of showing the impact of forest fires in Europe is shown in figures figure 1.2 where the total burned area from 2009 to 2014 in the European countries is depicted.

To determine the fire risk and fire spread potential on a given area taking into account the underlying vegetation and the meteorological conditions, several indexes have been developed. The most used indexes are *Fire Weather Index* [1] and the *Haines Index* [2]. Figures 1.3a and 1.3b show the values of both indexes, respectively, over Europe for a particular date. If figure 1.1 is compared to figure 1.3a and 1.3b, it can be extracted a clear correlation with the depicted data, what determines the degree of accuracy of these indexes.

Consequently, at European level, there exist a large research community focused on contributing to mitigate the effects of forest fire from different points of view. The complexity of this phenomenon requires to approach any solution as a multidisciplinary task that implies scientists from different fields. Therefore, *Computational Science*

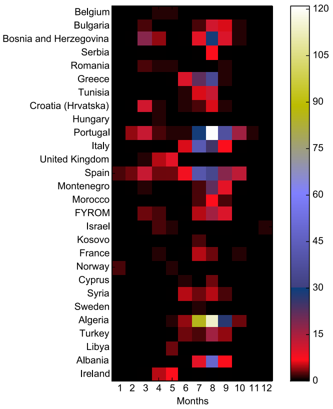


Figure 1.1: Ocurrence of Forest Fires in Europe from 2009 until 2014.

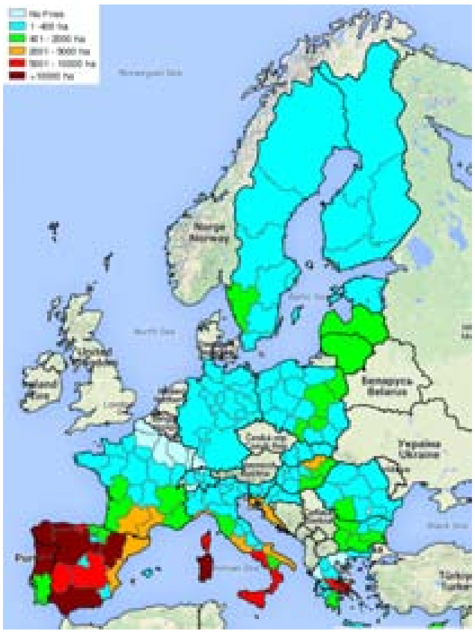
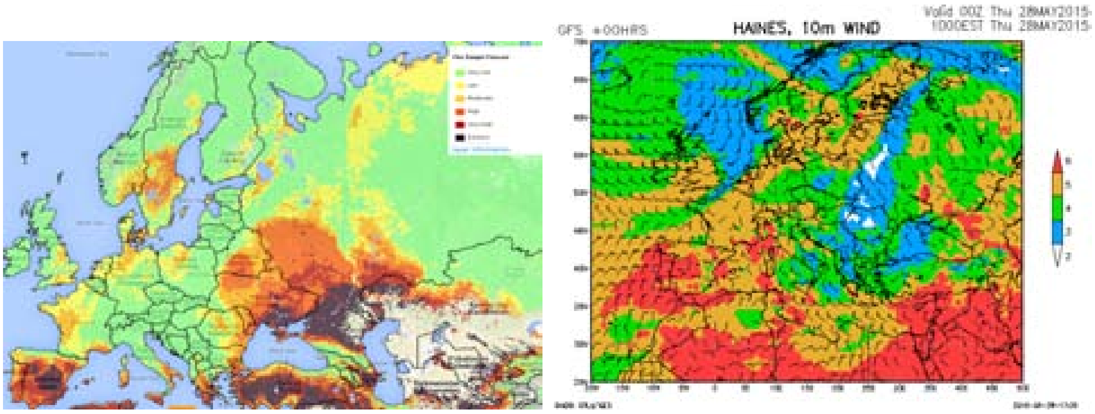


Figure 1.2: Burned areas by forest fires in Europe from 2009 until 2014.

arises as the scientific field that could help to develop instruments oriented to tackle this natural hazard.



(a) Fire Weather Index (b) Haines Index

Figure 1.3: Fire risk and spread potential indexes.

1.2 Computational Science and High Performance Computing

The scientific community has expended great efforts in modeling wildfire behaviour as accurately as possible to aid wildfire analysts during an ongoing hazard [3][4][5][6]. However, forest fire spread prediction is a complex and multidisciplinary field where physics, meteorology, forestry, computer science and geographical information systems among others fields are needed. This fact implies that a well coordinated team is crucial to improve forest fire spread prediction systems. This way of doing science corresponds to the so called Computational Science [7] (see figure 1.4).

In this new field, mathematics and scientist are able to model complex phenomena, which now can be computed using numerical methods and high performance computers. This fact enables the possibility to model some experiments that are very expensive or even impossible to do in reality [8][9]. So, there are some phenomena such as natural disasters, which represents hard society damages that can be modeled and simulated using Computational Science in order to predict the behavior of the phenomena [10]. When fighting against natural disasters, the fact of knowing the phenomena behavior beforehand, would have a direct effect in evacuation tasks, decision support systems, cost damages, etc. For that reason Computational Science represents an improvement for the society.

Simulation systems, which needs inputs and provides outputs, can describe the desired experiment with a certain error. The output error must be specified among the different steps of the creation of the simulation system and input data collection.

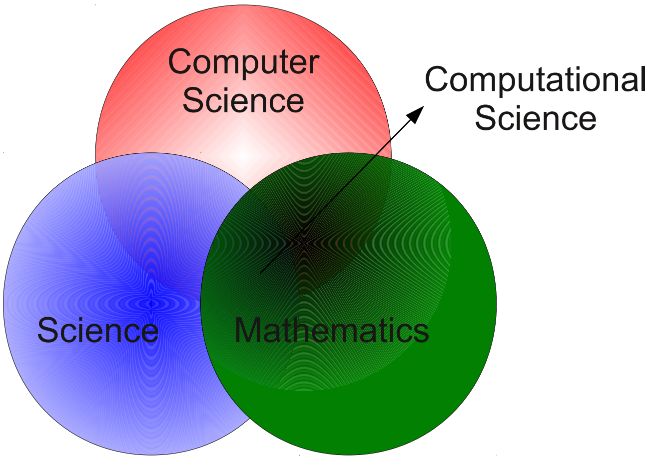


Figure 1.4: Computational Science defined as an intersection between Mathematics, Science and Computer Science.

Simulating the experiment with different inputs scenarios provides new data which can produce new knowledge.

One of the main advantages of Computational Science is that allows to simulate some scenarios that are not possible in reality due to different constraints. Simulation can be found in crash car tests [11], weather prediction [12][13], product prototype tests, social behavior, natural disasters such as hurricanes [14], tsunamis [15], floods [16] and others. The limitations of simulation resides in the computing performance capability, time available to run simulation and the reliability of the input data collection. So, simulation is able to model experiments that are difficult or impossible to do in reality and provides new valuable knowledge. Then, simulation is a powerful tool to use in natural disasters like forest fires [17].

But, despite the huge effort invested in forest fire knowledge discovering, there are still a huge gap between science and operational prediction systems. One problematic point when dealing with natural phenomena simulations, such as forest fire spread simulation, is input data uncertainty. The input data required for any existing forest fire spread simulator consists of a set of files (land use, fire perimeters from remotely sensed sources, meteorological data), which should be appropriately conflated. The high degree of uncertainty introduced in the process due to inexact input data results in a certain degree of error in the delivered predictions. Thus, different approaches have been developed to reduce input data uncertainty such as applying ensemble strategies, reducing the input parameters effects [18], and applying a Kalman filter to certain input variables to tune their values [19] among others. For that reason, calibration methods are relevant in order to provide an accurate spread prediction.

When such calibration techniques are applied the computational needs and the execution time increase significantly. However, the time constraints to deliver a spread predictions remains. Thus, it is mandatory to use high performance computing resources to provide an accurate prediction on time.

1.3 Basic Prediction Scheme

To understand the requirements of forest fire spread prediction, let's analyze how the phenomenon should be described from the computational point of view. A forest fire can be defined by a burning perimeter at a certain time. As the forest fire propagates, different perimeters can describe the fire front evolution, so the fire propagation can be seen as a set of perimeters, which describe the arrival position of the fire front at different time steps. This kind of evolution representation is shown in figure 1.5 where a 2-dimensional map evolution is depicted over a yellow background, which represents the elevation map. Other properties that could be obtained from the fire front are Fire Intensity (FLI), Flame Length (FML) or the Rate of Spread (ROS), which describe the speed of the front.

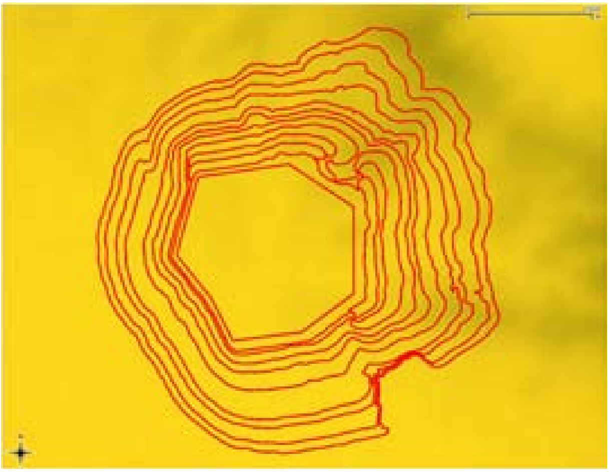


Figure 1.5: Projection of fire perimeters in the terrain, bidimensional representation.

Any basic prediction system requires a 2-dimensinal forest fire spread simulator to obtain the fire front evolution. In this work, the forest fire spread simulator used is FARSITE [20][21][22]. FARSITE needs a set of input parameters to describe the environmental conditions where the fire simulation takes place. These inputs are the ones related to the topography, vegetation, meteorological properties and the initial fire front. Furthermore, the time interval to be simulated, the precision required in the outputs, as well as the computation precision must be specified. The input data that depends on space is expressed by maps, that can be represented by raster or vectorial formats. Raster maps divide the surface in a grid composed by cells where each cell has a value that defines a certain characteristic in that map position. For example, the DEM (Digital Elevation Map) that describes the elevation of the terrain is represented using a raster map because it is mandatory to have an elevation value for each cell. On the other hand, FARSITE uses vector files in those cases where not all cells have relevant information. An example of that case is the initial ignition perimeter where only a subset of the cells are affected. The fire front (perimeters) information could be stores using different formats such as points, lines or polygons. Following, data maps required by FARSITE are briefly described:

- Raster Maps:
  - Elevation file: This file describes the elevation at each cell providing in this way the shape of the surface.
  - Slope file: This information is needed during the fire simulation and it is better to be computed previous to the simulation. One can obtain this file from the elevation map using GIS software such as Miramon [23], Arcview [24], Quantum GIS [25] which uses GDAL [26].
  - Aspect file: It is a similar case to slope, but this file contains the direction of the slope per cell. It is also precomputed using GIS software.
  - Canopy file: Also a raster map, which provides information about the percentage of the cell that is covered by canopy trees.
  - Fuel file: Contains the the kind of fuel in each cell. This information is very important due to the close dependency between rate of spread and the fuel properties.
- Vector Files:
  - Initial ignition: This input describes the polygon that represents the initial fire. Typically, this input data is needed to be pre-processed because it can be provided in different raster files formats.

- Fuel moistures: It is a value that commonly is not available in some regions. It is provided in a file where there are all the fuel models with the different moistures to certain times such as 1hour, 10hour, and 100hour.

- Weather data: This file contains different meteorological information fields at different time intervals, such as:

  Maximum and Minimum temperatures

  Maximum and Minimum humidity

  Cloud cover data

  Wind speed

  Wind direction

Normally, the above listed maps are obtained from different sources in different time scales, however, at a simulation/prediction time, all needed maps should be overlapped in a single file. To generate a correct overlapped file (called LCP file), all required maps must be well located, with the same spatial reference system, same resolution, and using the units specified by the simulator. Furthermore, it is convenient to reduce the executed map size just to the zone of interest, due to memory and time constraints, specially, when considering a wind field model. This pre-processing map task is a non dismissible procedure that should be properly established because prediction results depends on it. The whole process is summarized in figure 1.6.

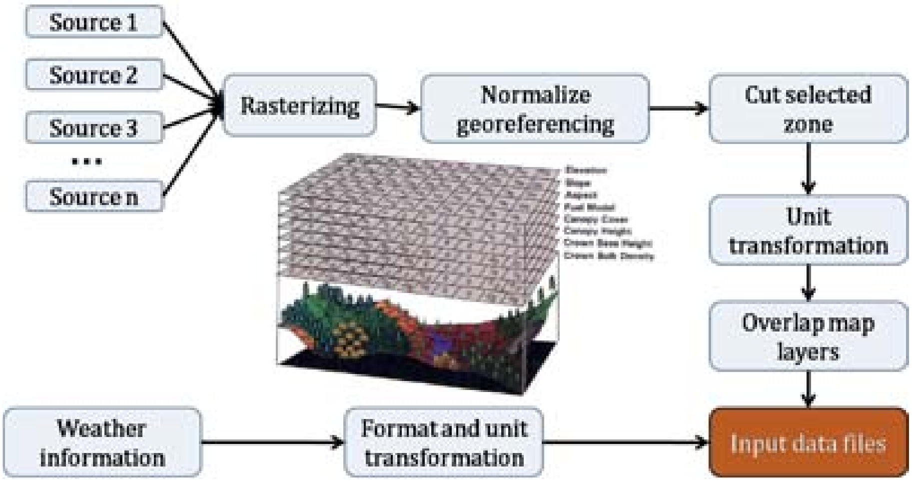


Figure 1.6: Map pre-processing flow for a forest fire spread prediction systems.

The results provided by the system will typically be organized in different files, each one including different data related to the fire behaviour such as: fire time arrival, final fire front intensity, fire perimeters at a certain time step, etc.

As it has been mentioned, fire behavior depends on different parameters being the most sensitive ones: wind speed and wind direction, fuel features and the slope of the terrain. Obviously, the most dynamic parameter of those is the wind. Therefore, one should take a special attention to its behaviour. Wind speed and wind direction are directly affected by the terrain characteristics, specially when dealing with large wildland fires. Thus, it is crucial to consider wind variations due to the topography. For that reason, a wind field model should also be included in any forest fire spread prediction system to compute the values of the wind speed and wind direction at ground level given a certain coarse meteorological wind data. In this work, the selected wind field modeler has been WindNinja [27][28] because it could be directly coupled to FARSITE.

Coupling models to the forest fire spread system has a positive impact in the accuracy of the simulation. However, the execution time of the prediction system increases. Therefore, the whole prediction could spent such a long time that the resulting prediction could be outdated at the time to be used. Thus, despite the complexity of the resulting forest fire spread prediction system, it should be able to deliver a propagation forecast under strict real time constraints. Otherwise, the obtained results will be useless.

1.3.1 Classical Forest Fire Prediction: Drawbacks

The classical prediction scheme, that has been described in the previous sections, needs a considerable number of precise inputs. In real emergency cases, all this data is needed in a short time lapse. The set of input data parameters is not always available at the time of the event or it is not enough reliable, therefore, in real cases, some input data must be obviated or approximated. Consequently, simulation input data stems from a certain error what directly impacts the accuracy of the outputs (see Figure 1.7).

Due to the uncertainty in the input parameters, the Classical Forest Fire Prediction was enhanced by adding a calibration stage before doing the effective prediction, in order to obtain a set of input parameters that best reproduce the near past. This set of calibrated input data will be used to perform the forest fire spread prediction for the near future. The Two-Stage scheme is widely described in the next section.

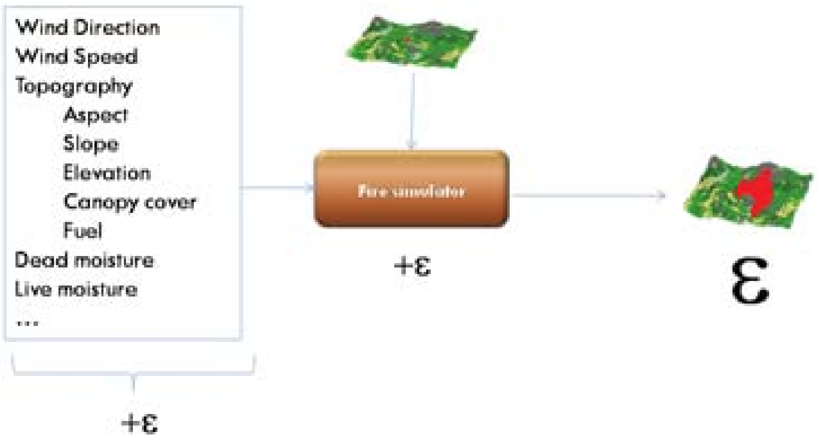


Figure 1.7: Error sources in any forest fire spread system that directly affects the accuracy of the results.

1.4 Two-Stage Forest Fire Spread Prediction

The Two-Stage method is based on dividing the prediction time horizon into two different stages (see figure 1.8). These two stages are subsequently described:

- Calibration Stage: The interval time defined as the one used for calibration purposes ($[t_i \ \ t_{i+1}]$) determines two time instants where the real fire perimeter are obtained. These two fire front positions are used by the calibration strategy, a Genetic Algorithm (GA) in this case, to search for those unknown input values [29][30][31].
- Prediction Stage: Once a calibrated input parameters set is obtained, it will be used in the prediction stage to forecast the fire behaviour during the interval time $[t_{i+1} \ \ t_{i+2}]$.

As it has been mentioned, the Two-Stage prediction scheme uses a GA to calibrate values of the input parameters. The initial GA population is a set of individuals (different configurations of the input parameters values, also called scenarios), where each input value is randomly obtained within the corresponding variation rang. After that, all individuals are simulated.

Therefore, one comes out with a set of forecast fire perimeters and, for each one an error function is evaluated. This error is obtained from compared the real fire propagation with the predicted perimeters using the symmetric difference. Then, the individuals are sorted according to the obtained error and, the GA operators such as elitism, selection, mutation and crossover are applied to obtain a new set of individuals, called second generation. Subsequently, the process is repeated again. The GA works in an iterative way until a preset number of generations is achieved. The GA has the advantages of being an optimization technique that skips local minimum driving the system to the best zone of the search space. However, GAs are computational intensive and, therefore, parallel strategies should be used to accelerate the convergence and being able to reach the preset prediction deadline. In the next section, the first parallel approach of the Two-Stage prediction scheme is introduced.

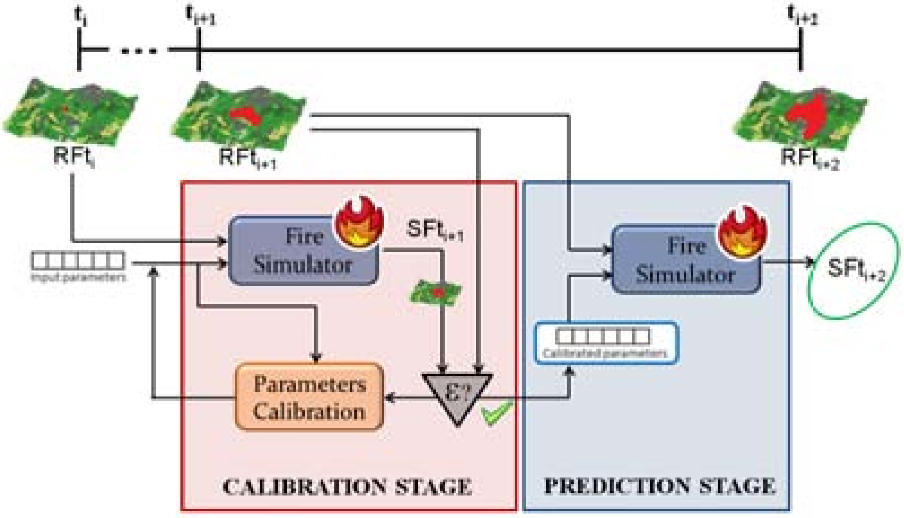


Figure 1.8: Two-Stage prediction scheme.

1.4.1 MPI Master Worker approach

The Two-Stage method described in section 1.4, provides more accurate predictions but it requires a huge number of forest fire spread simulation executions because of the number of individuals involved in the GA population and the number of iterations done in the Calibration stage. This important workload increases the execution time of the prediction system. Taking into account that for each individual the fire simulator must be run, the computational cost of the whole process highly increases. Thus, the method becomes non-viable under a serial programming technique. For this reason, parallel programing strategies are introduced using Message Passing Interface (MPI) and the Master\Worker programming paradigm as it is shown in Figure 1.9.

As it is stated in [32], the execution time of a single fire spread simulation hardly depends on the particular settings of the input parameters. The execution time can range from few seconds to more than 10 minutes for a single simulation. This difference can lead the prediction system to a high execution time, where several worker

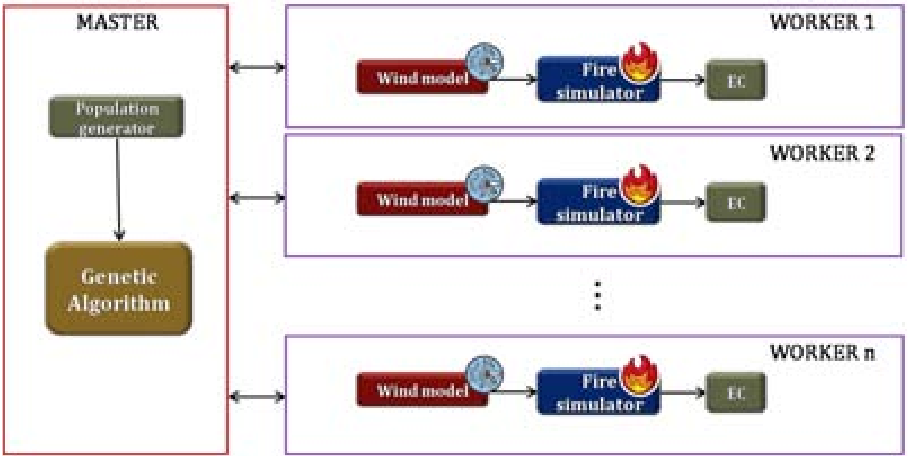


Figure 1.9: Two-Stage prediction method under a Master\Worker structure.

processes remain idles for a long time (see Figure 1.10). Therefore, the main goal of this work, consist of exploiting the multi-core capabilities of current systems in order to reduce the workers execution time. This will be firstly done by means of applying OpenMP to the worker in order to obtain a hybrid scheme for forest fire spread prediction, that provides an accurate prediction in the preset time deadline. Furthermore, different time-aware core allocation policies had been developed to accomplish the strict deadlines constraints of the problem.

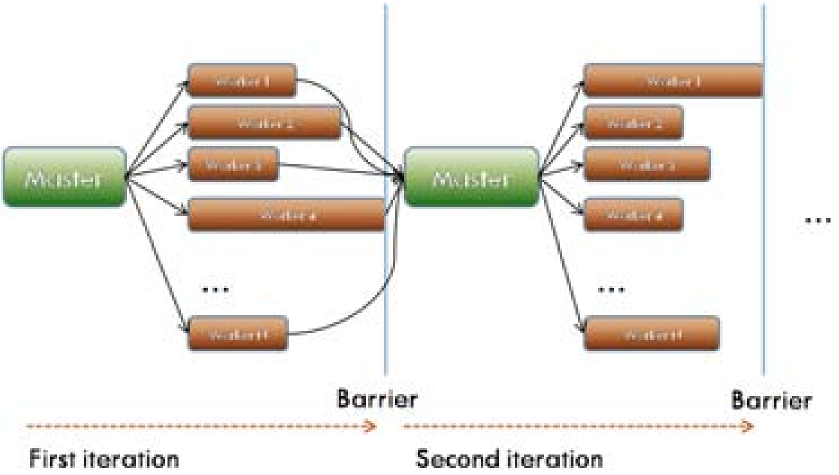


Figure 1.10: Idle workers caused by the barrier per generation in the Master\Worker structure.

1.5 Compendium of publications

This PhD Thesis is presented as a compendium of publications. Each one of the contributions are introduced in chapter 2. These contributions have been divided in two categories: published contributions and contributions under review. The first group is conformed by those papers that are already published and have been approved by the PhD Doctoral Commission to be enough to start the PhD dissertation process. The contributions under review are those papers that were under review process at the moment of initiating the PhD dissertation procedure. All papers (published and under review) have an unifying thread, which is directly related to the main contribution of this Thesis, *"defining strategies to exploit multi-core architectures in order to execute forest fire spread prediction under strict real-time constrains without penalizing the prediction accuracy"*.

1.6 Outline

In chapter 2, each one of the contributions are summarized. All contributions are sorted according to the chronological enhancements in the proposed strategies. In particular, an hybrid MPI-OpenMP approach for the Two-Stage prediction method is defined. In order to cope with the strict time deadlines, different core allocations policies have been proposed: TAC (Time-Aware Core allocation), Re-TAC (Resolution based TAC) and Soft-TAC where a dynamic size populations scheme for the GA is proposed. Each strategy improvement is oriented to enables the GA the capacity of exploring all search space despite of being the good solutions too time consuming.

Chapter 3, describes the work done during the pre-doctoral stay at the Joint Research Center-Ispra (Italy). The main goal of this stage was to deploy a prototype of an operational forest fire spread system that considers the requirements defined in this Thesis.

Chapter 4 summarizes the main conclusions of this Thesis and, chapter 5 includes an overview of the open research lines and future work.

Finally, Appendix A includes all published contributions, Appendix B the contributions under review and Appendix C includes an example of the prediction reports delivered by the operational system deployed at the JRC-Ispra (Italy).

Chapter 2

Contributions

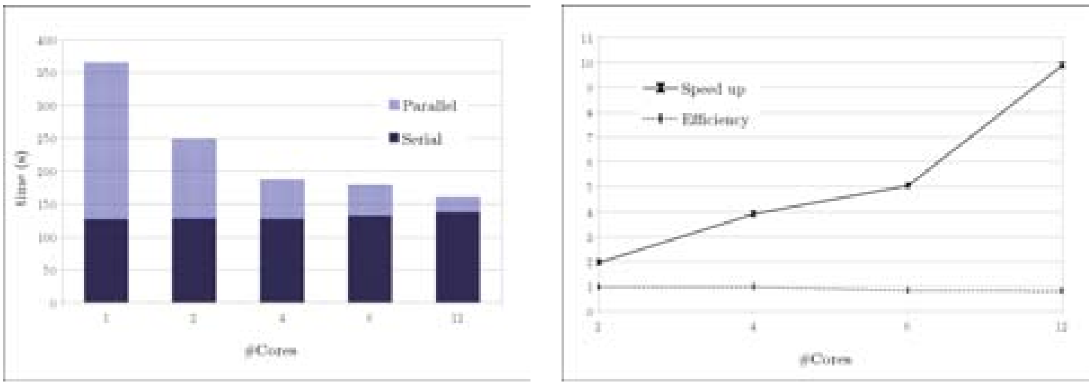
2.1 Published contributions

In this chapter, all published papers that have been accepted as a compendium of publications for this PhD dissertation, are summarized following their unifying threading.

2.1.1 Relieving the Effects of Uncertainty in Forest Fire Spread Prediction

> Artés, T., Cencerrado, A., Cortés, A., & Margalef, T. (2013). Relieving the effects of uncertainty in forest fire spread prediction by hybrid MPI-OpenMP parallel strategies. *Procedia Computer Science*, 18, 2278-2287.

When dealing with natural hazards prediction, the execution time of the simulators becomes crucial. The goal of this work consists of accelerating the execution time of the whole calibration method focusing on the potential parallelism of the simulation core, FARSITE. The forest fire spread propagation simulator was profiled [33][34], detecting the most consuming time parts of the code. With the idea of using the parallel programing inside of the node in the MPI implementation of the GA, OpenMP pragmas were used to achieve such a goal. Most of the OpenMP pragmas were inside the function CrossCompare() which represents the 60% of the execution time when the execution is long enough. Figure 2.1(a) and 2.1(b) show the speedup and the efficiency, respectively, of the parallelized part of the code.



(a) Execution time of the parallel implementation of FARSITE by indication which part is executed either in parallel or in serial

(b) speedup and efficiency of the parallel part

Figure 2.1: Parallel FARSITE.

As it can be observed, the proportion of the potential parallel time is not constant for all executions, for that reason, a set of individuals generated during an execution of the GA was run using 1 and 4 cores. Figure 2.2 shows the improvement in the execution time when using 4 cores and depending on its serial execution time.

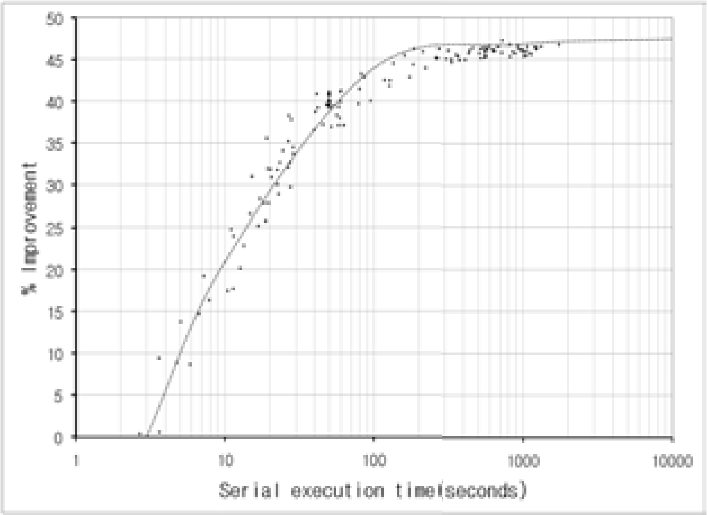


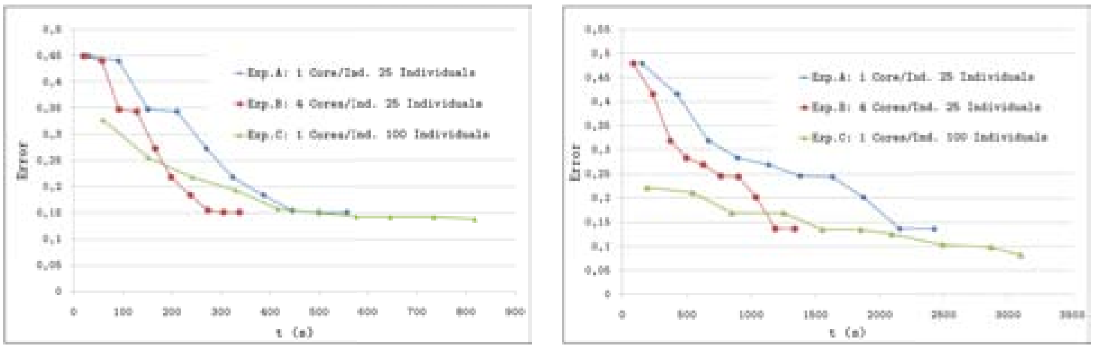
Figure 2.2: Execution time improvement compared to serial time

In order to analyze the effects of the parallel version of FARSITE in the calibration process, three different experiments were proposed:

- Experiment A: the execution of the GA was carried out using 10 iterations and 25 individuals. Thus, the required amount of cores was 25 cores.

- Experiment B: This experiment uses the populations generated by the previous experiment trying to reproduce the same conditions. The GA was executed for 10 iterations and the population size was 25 individuals. This time, instead of using 1 core per individual, 4 cores were used. Consequently, 100 cores were needed to run the GA.

- Experiment C: This last experiment does not try to reproduce the same conditions using the same populations generated for every iteration of the experiment A. The main goal of this approach was to use the same amount of resources that in experiment B, 100 cores, consequently, the population size of the GA was increased to 100 individuals. This fact could improve the final prediction error and, furthermore, it performs faster.



(a) Error/time graph of a low workload GA execution.

(b) Error/time graph of a high workload GA execution

Figure 2.3: Two different executions of the three experiments of the GA.

Figure 2.3 depicts the error and execution time for the three different executions of the GA. When comparing experiment A and B, it can be seen that the errors are obviously the same, because their are using the same intermediate populations in the GA, but the experiment B is almost two times faster. If experiment B and C are compared with high and low workloads, the increased population experiment C starts with a lower error than B, but it takes more time to evolve to the next population. Thus, it appears an area where experiment B, is getting better error than experiment C. This last fact is more visible in the low workload case (see figure 2.3(a)).

2.1.2 Core Allocation Policies on Multicore Platforms to Accelerate Forest Fire Spread Predictions

> Artés, T., Cencerrado, A., Cortés, A., & Margalef, T. (2014). Core Allocation Policies on Multicore Platforms to Accelerate Forest Fire Spread Predictions. *In Parallel Processing and Applied Mathematics, PPAM 2013*, 151-160. Springer Berlin Heidelberg.

Up to know, the efforts of the described work was focused on accelerating the calibration process of the Two-Stage system by defining a hybrid MPI-OpenMP calibration scheme. However, no mention to how many cores assign to a given individual has been done. This contribution describes the proposal of a preliminary policy to allocate different numbers of cores per individual. The first key point that appears when trying to implement such a task, is the missing knowledge about the execution time of each individual beforehand. The allocation policy proposed relies on a previous work [35][36], which describes a methodology to characterize a given simulation kernel [37]. The kernel is executed for a certain area scenario with a representative numbers of individuals recording its parameters and the execution time of each one. Following, the histogram of such executions can be created (see figure 2.4). The next step, consists of dividing the histogram in "classes" using local minimums as class' borders.

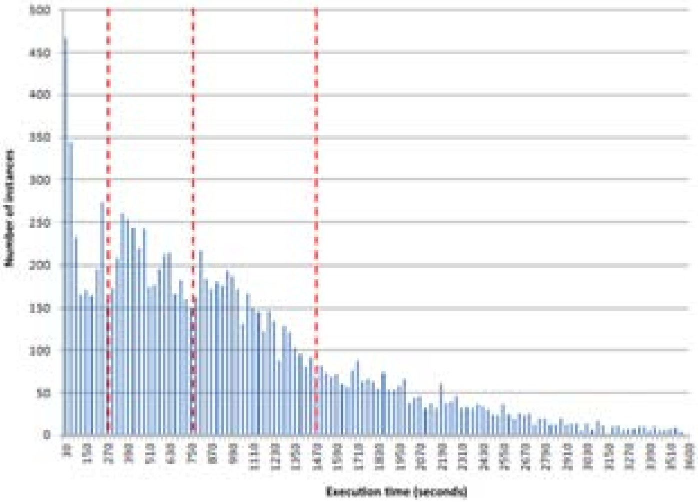


Figure 2.4: Histogram of FARSITE executions divided into four different classes.

Thus, 4 different classes are considered:

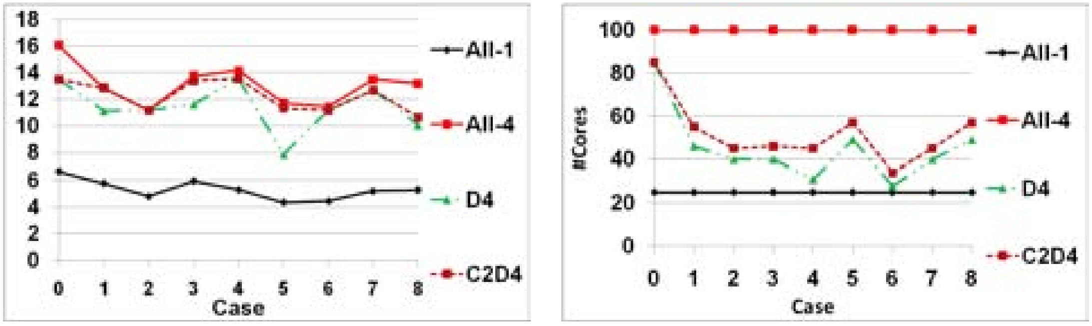
- Class A: $T_{Ex} \leq 270s$

- Class B: $270s < T_{Ex} \leq 750s$
- Class C: $750s < T_{Ex} \leq 1470s$
- Class D: $1470s < T_{Ex} \leq 3600s$

All the individuals of the histogram are labeled with their corresponding class. Next, a C4.5 algorithm [38][39] available in Weka [40] is used to generate a decision tree using the individuals with their associated input parameters and the corresponding class labels. Finally, each time a new individual is generated, it can be classified before its execution by using the obtained decision tree and assigning to it a class label.

Thus, the implementation of the GA [18] is modified to incorporate the decision tree and also policies to determine the number of cores to assign to each individual. In this section four different core allocation policies are proposed:

- All-1: One core for all the individuals.
- C2D4: Two cores for C individuals and four cores are used for D individuals.
- D4: Four cores are used for individuals which belong to D class.
- All-4: All individuals uses four cores.



(a) Speedup of 10 GA executions for the proposed core allocation policies (b) Core usage during 10 GA executions for the proposed core allocation policies

Figure 2.5: Two different executions of the three GA experiments.

Every proposed policy is executed 10 times and the GA executions are done using 25 individuals and 10 generations. As depicts in figure 2.5(a), the policy All-4 provides the best speedup. The two policies that rely on the classification method of the individuals are quite near to the best speedup, being the policy C2D4 the best one. When looking at the core usage, in terms of number of cores used, the policy C2D4 compared with D4 does not use so many cores. In conclusion, C2D4 seems to be the best policy reaching almost the same speedup that All-4 and using about half of the cores.

2.1.3 Relieving Uncertainty in Forest Fire Spread Prediction by Exploiting Multicore Architectures

> Cencerrado, A., Artés, T., Cortés, A., & Margalef, T. (2015). Relieving Uncertainty in Forest Fire Spread Prediction by Exploiting Multicore Architectures. *Procedia Computer Science*, 51, 1752-1761.

Core allocation policies introduced in the previous contribution, have shown to play an important role in the reduction of the execution time of the forest fire spread system. In this section the results obtained when using this core allocation policy in two different computing clusters, are reported. Table 2.1 summarizes the obtained classes boundaries for each tested system. This experiment has been repeated 50 times to take into account the stochastic part of the GA.

	Class A	Class B	Class C	Class D
IBM: 2 threads	73	209	433	1054
IBM: 4 threads	58	162	332	938
Dell: 1 thread	105	360	675	1617
Dell: 2 threads	78	250	493	1121
Dell: 4 threads	67	229	424	998
Dell: 8 threads	54	193	307	734

Table 2.1: Upper boundaries (in seconds) for each class according to each combination [computer platform, number of threads]

Figure 2.6 shows the results obtained in terms of GA execution time for each one of the 50 experiments when applying the core allocation policy and the previous MPI version. As can be observed, in figure 2.6, the mean execution time is substantially reduced. However, there are certain cases where the execution time is not significantly reduced.

This fact happens in those cases where there is a low number of individuals classified as C or D. If the individuals that determines the generation elapsed time are not C or

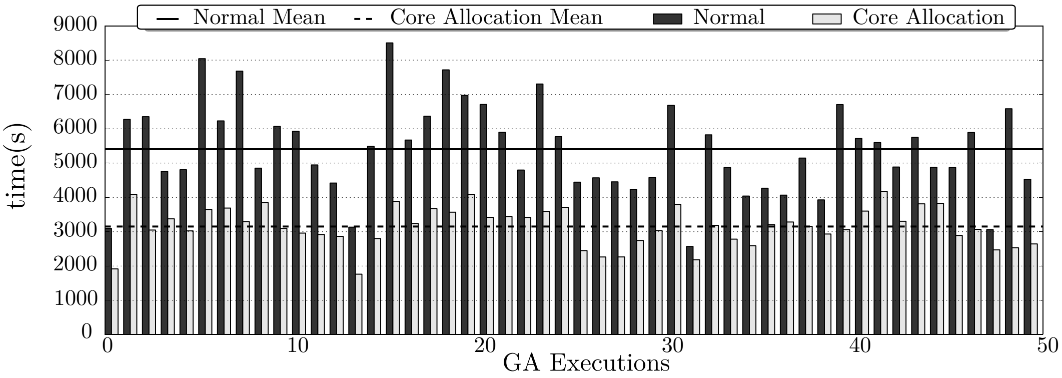


Figure 2.6: Execution time of 50 runs of the GA using the basic core allocation policy.

D, the execution time is not significantly reduced. Figure 2.7 shows the distribution of class C and D individuals for all 50 experiments at each GA generation. As can be observed, there are certain situations (for example populations 8, 31 and 47 for instance) where the number of individuals C and D is not relevant and, therefore, the core allocation policy does not introduce any reduction in the total execution time.

Hence, this contribution shows that core allocation works in different cluster architectures and, in most of the cases, it reduces the execution time of the Calibration stage.

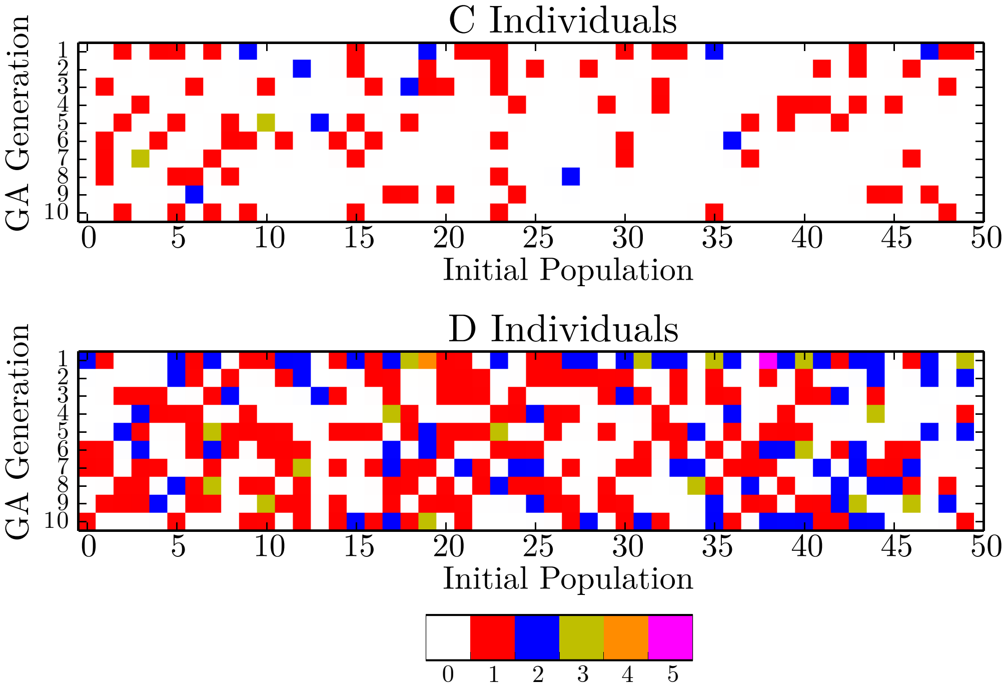


Figure 2.7: Number of individuals C and D for each generation of the 50 executions of the GA.

2.1.4 Time-aware Multi-threaded Genetic Algorithm for Accelerating a Forest Fire Forecast System

> Artés, T., Cencerrado, A., Cortés, A., & Margalef, T. (2014). Time-Aware Multi-threaded Genetic Algorithm for Accelerating a Forest Fire Spread Forecast Systems. *14th International Conference on Computational and Mathematical Methods in Science and Engineering.* Conference Proceedings, 103 - 114.

From the contribution described in section 2.1.2, one concludes that the class division criteria based on local minimums was not enough accurate to uniforme the execution time of the different individuals. Thus, the classification was modified to achieve this goal by analyzing in more detail the effects of the OpenMP parallelization at individual level (see figure 2.8).

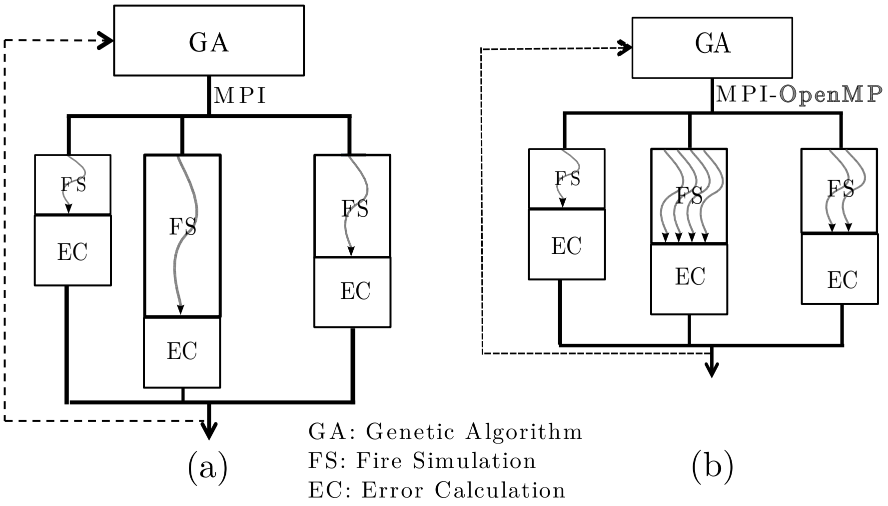


Figure 2.8: Two different versions of the GA: one level parallelims using MPI (a) and a MPI-OpenMP version which tries to uniforme execution time of all individuals (b).

For that reason, FARSITE was profiled and parallelized reaching a potential of 60% of the serial time that could be executed in parallel. Thus, the expected parallel time could be expressed as follows:

$$t_{par}(N_{Cores}) = 0.4 * t_s + \frac{0.6}{N_{Cores}} * t_s \tag{2.1}$$

From Equation 2.1, it can be concluded how many cores are needed (N_{Cores}) to reduce a given execution time t_s to a new execution time defined by $t_{par}(N_{Cores})$.

Class	Cores	Time limits
A	1	$0 < t_s \leq tmax_{Gen}$
B	2	$tmax_{gen} < t_s \leq 1.42 * tmax_{Gen}$
C	4	$1.42 * tmax_{Gen} < t_s \leq 1.81 * tmax_{Gen}$
D	8	$1.81 * tmax_{Gen} < t_s \leq 2.1 * tmax_{Gen}$

Table 2.2: Time limit classes and numbers of cores used for a time constraint of $tmax_{gen}$.

Table 2.2 shows the number of cores needed for a given time limit when the maximum time assigned to a GA generation is $tmax_{Gen}$.

The individuals with an expected execution time longer than $2.1 * tmax_{Gen}$ are replaced by individuals of class A. Applying the proposed Time-Aware Core Allocation policy (TAC), the execution time of all workers is more uniforme. Figure 2.9 shows, for a particular case, the execution time traces for all workers without applying the TAC strategy. As it can be observed, the final execution time is more than 30485 seconds. On the other hand, figure 2.10 shows the execution time traces when applying the TAC policy. In this case, the final execution time of the GA reduced almost three times to 10121 seconds.

Despite this time reduction, the main concern in both cases is the efficiency. Next contribution was focused to solve this problem.

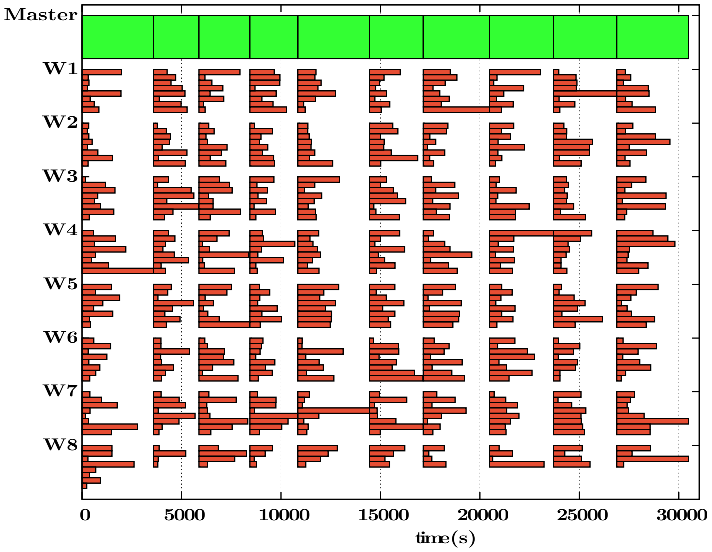


Figure 2.9: Traces of the GA based on MPI parallel approach when 1 cores is assigned to each individual.

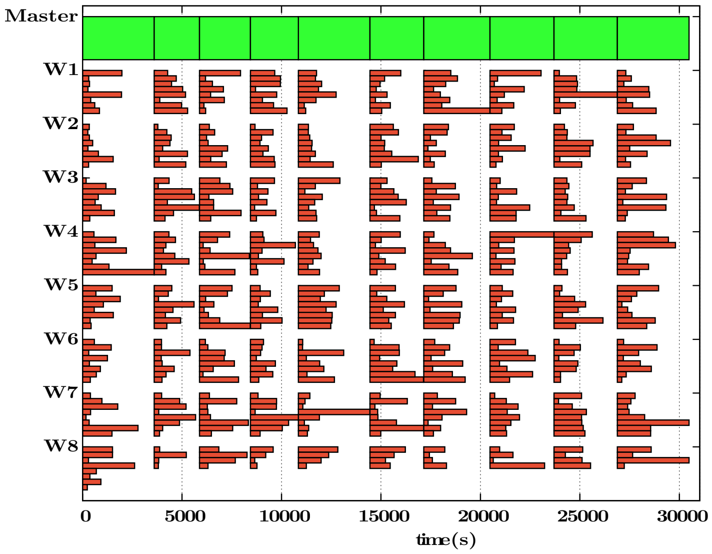


Figure 2.10: Traces of the GA based on an MPI-OpenMP approach

2.1.5 Enhancing computational efficiency on forest fire forecasting by time-aware Genetic Algorithms

> Artés, T., Cencerrado, A., Cortés, A., & Margalef, T. (2015). Enhancing computational efficiency on forest fire forecasting by time-aware Genetic Algorithms. *The Journal of Supercomputing*, 71(5), 1869-1881.

As it has been shown, using the basic TAC strategy penalizes the efficiency of the GA parallel execution. In order to overcome this drawback, the implementation was modified to use non-blocking MPI routines. This way, an on demand Master/Worker scheme was developed. This approach was oriented to reduce workers s idel time keeping them as busy as much as possible.

The same initial population used in the previous section was reused to run this new approach but using half of the resources and the same time constraints were applied. The obtained traces for both executions can be seen in figures 2.11 and 2.12 respectively.

The analysis of the efficiency for both versions is not an obvious task. Only the initial population is the same, the subsequent populations would be different because of the random element of the GA. This time, the populations are not saved and used for both versions because those individuals that reach the time limit per generation were killed and consequently not used in the GA. So, using less resources could raise the number of killed individuals and get a worse final calibration error. Figure 2.13

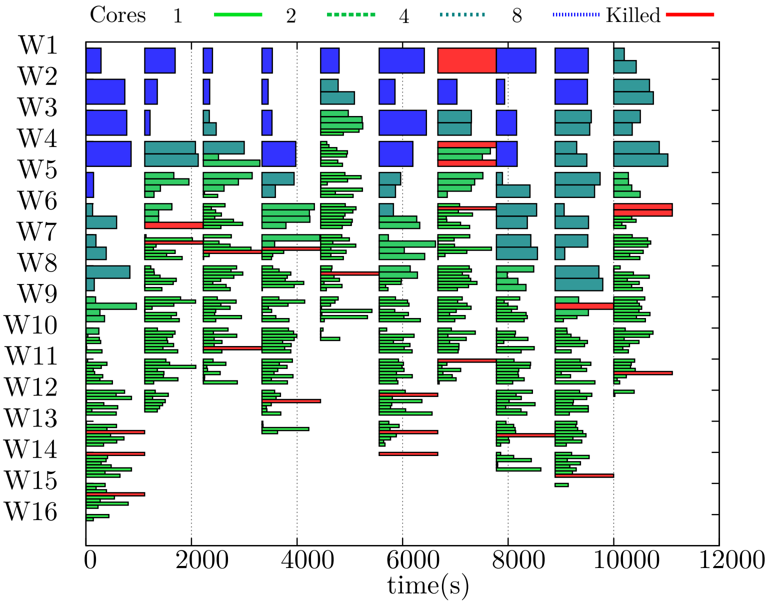


Figure 2.11: Trace obtained using TAC.

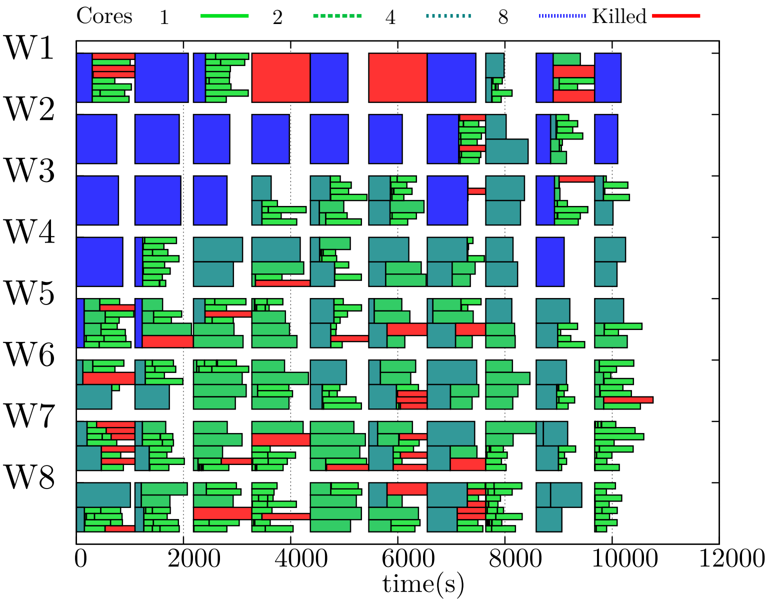


Figure 2.12: Traces obtained using the Efficient TAC approach.

depicts the error obtained using TAC and TAC with Master/Worker on demand. The difference between both versions is not so relevant. In mean terms, both errors are quite similar.

The efficiency was computed estimating the expected serial time of the GA using 1 cores per individual. As this execution will take days to compute, this time is estimated using the proportion of parallel execution of FARSITE and suposing a serial execution

Figure 2.13: Error obtained for 10 different executions of the GA with TAC and the efficient TAC.

without MPI. The killed individuals are also used to compute this value, but using the maximum time assigned to this individual. Instead of comparing the calibration error, figure 2.14 shows the efficiencies obtained with both versions. The efficiency obtained using the efficient TAC is significantly higher than the basic TAC.

Figure 2.14: Efficiency obtained for 10 different executions of the GA with TAC and the efficient TAC.

In conclusion, with this modification the achieved efficiency was almost twice the efficiency obtained with the previous version of TAC, and the error is quite similar for almost all the executions. In fact, this contribution increases the throughput of individuals per generation for a given worker. That means that providing more resources the space that the GA could explore is wider and, consequently, the error of the GA could be significantly reduced.

2.1.6 Towards a Dynamic Data Driven Wildfire Behavior Prediction System at European Level

> Artés, T., Cencerrado, A., Cortés, A., Margalef, T., Rodríguez-Aseretto, D., Petroliagkis, T., & San-Miguel-Ayanz, J. (2014). Towards a Dynamic Data Driven Wildfire Behavior Prediction System at European Level. *Procedia Computer Science*, 29, 1216-1226.

When dealing with real scenarios, the data sources are crucial. This contribution was done in collaboration with the Joint Research Centre (JRC)-Ispra in Italy. In this section, a study case is used as a first experiment to determine all required data sources, their original formats and the transformations that must be performed to be able to run the Two-Stage prediction method.

The data provided by the JRC could be organized in the following four categories:

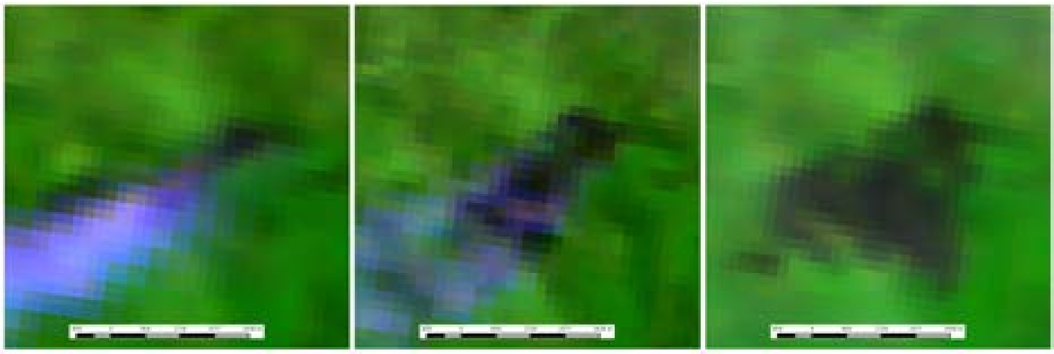
- Topography: the data is a Digital Elevation Map (DEM). This data is a product of the Advanced Spaceborn thermal Emission and Reflection Radiometer (ASTER) instrument on board of NASA Terra satellite. The DEM resolution is 30m.

- Fuel map: This is a raster map which constains the predominant kind of vegetation for each cell. There are 42 different fuels which are converted to the 13 fire behaviour fuel models defined by Anderson [41].

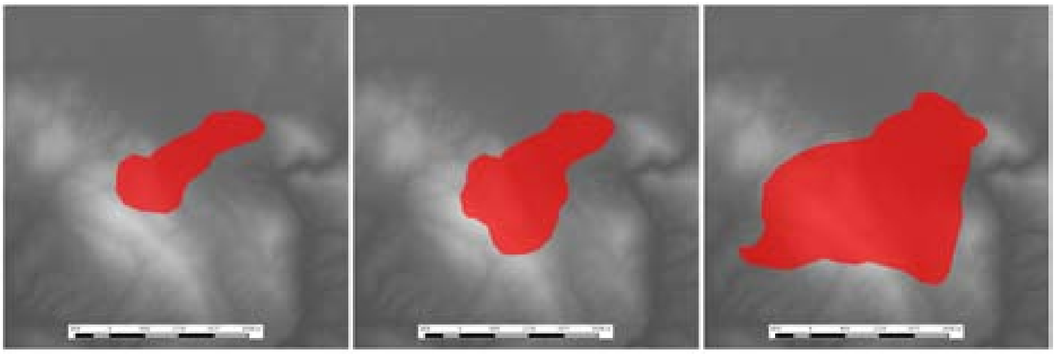
- Meteorological data: the meteorological data data used is from the European Centre for Medium-Range Weather Forecasts (ECMWF) high-resolution global model. The horizontal resolution is 16km with a time-step of 3 hours.

- Fire perimeters: EFFIS (European Forest Fire Information System) [42] includes a module of burned Area Map. This module uses as input the Moderate Resolution Imaging Spectroradiometer (MODIS) sensors in Aqua and Terra satellites. Thus, EFFIS provides the most reliable perimeter describing the burned area, at least, once a day for each active fires in Europe.

The study case was a fire that took place in Greece during the summer of 2011 in the region of Arkadia. The initial perimeter is the one observed by Terra satellite on August 26th at 09:43 pm. The next perimeter corresponds to the same day and it was captured at 11:27 pm. Finally, the last observation was done on August 27th at 08:49 am. The MODIS image and burned area are shown in figure 2.1.6.



(a) MODIS images



(b) Vectorial shapes as a result of MODIS images.

Figure 2.15: MODIS images and their corresponding extracting shapes

The Two-Stage prediction scheme was applied to the Arcadia case, but considering four different configurations of the spatial and temporal variation of the input parameters:

- Homogeneous Spatial and Homogeneous Temporal(HSHT): All data was considered constant in time and space over the map.
- Homogeneous Spatial and Variable Temporal(HSVT): the data was considered constant in space, but the meteorological data was used considering the time-step of the 3 hours set up by the ECMWF.

- Variable Spatial and Homogenenous Temporal(VSHT): The wind speed and wind direction were computed using WindNinja wind field model [43][44][45], in order to consider their variation due to the terrain characteritics. However, no variation through time was included.

- Variable Spatial and Variable Temporal(VSVT): In this case, WindNinja was used for considering the wind spatial variation and the variation in time was also incorporated by using the 3 hours time-step data provided by ECMWF.

Since the Two-Stage prediction scheme has a steering strategy in the Calibration stage to tune the input parameters values that is driven by the observed real fire propagation, it fits the definition of Dynamic Data Driven Application Systems (DDDAS) [46][47]. The DDDAS-VSVT image in figure 2.16 depicts a spread prediction when the VSVT configuration is used. In this case, the obtained prediction overestimates the fores fire spread, but it clearly describes the three main spread direction of the fire. To overestimate the fire spread prediction is not a relevant concern because of the human action. This fact is not considered for simulation purposes, that is, the delivered prediction is a freely burning fire.

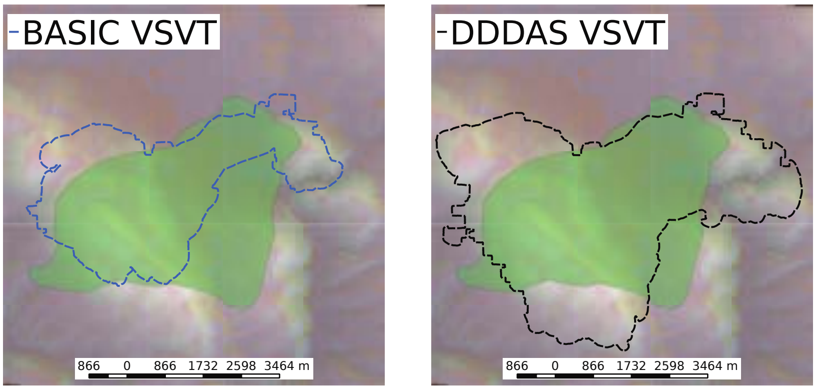


Figure 2.16: Results obtained using the non calibrated VSVT and the Two-Stage based VSVT.

2.2 Contributions under review

In this section, two not already published works are included. The main contribution of both works consists of two further TAC enhancements oriented not to penalize the optimization search when the good solution resides in a slow individual.

2.2.1 TAC with resolution adjustment

TAC has been proven to be a useful approach to fit the execution time of the calibration stage in the required response time. However, when the good solution is a long one, it could happen that a non negligible number of individuals shall be killed or even not executed. For that reason, it is mandatory to tackle such problem to keep the calibration accuracy lower bounded.

Figure 2.17 shows the TAC behaviour for a high workload case. As it can be observed, most of the individuals are killed during the calibration stage. Therefore, the calibration results can be distorted due to this malfunction.

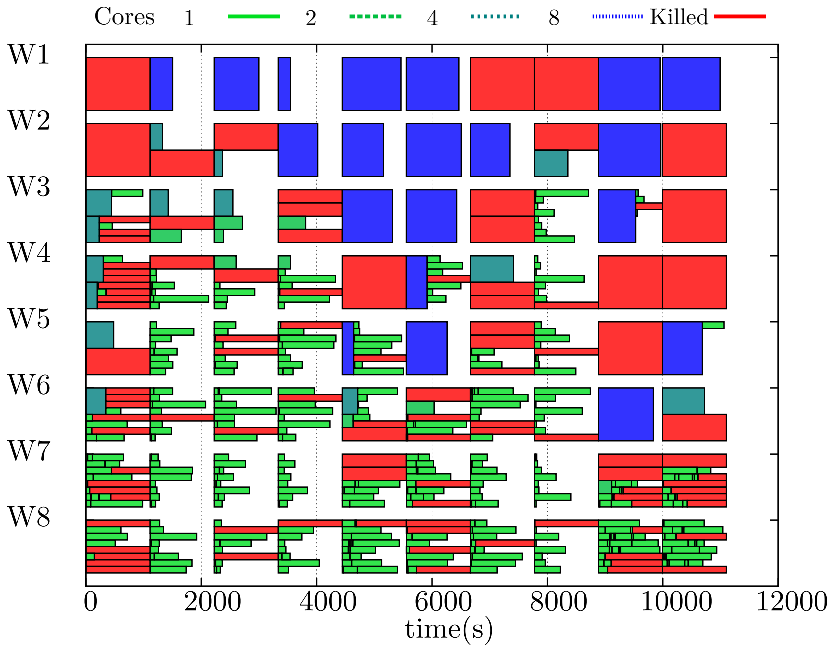


Figure 2.17: Trace of the TAC-Two-Stage prediction scheme with a generation time of 1080 seconds as limit. The GA has a population size of 64 individuals and 10 generations. The parallel MPI-OpenMP implemented used 8 workers where each worker has 8 available cores.

This is caused because TAC is removing part of the search space of the GA where one possible solution may lie in.

Figure 2.18 describes the proportion of individuals of each class at each population of the GA when executing without TAC and no time restrictions. There is a clear trend in the GA to generate E individuals, that could be because of the good spread results are the ones provided by this kind of individuals. When an individual provides good results, the GA would select this individual for crossover and elitism (the individuals is included in the next generation without changes). Therefore, TAC needs to be improved to work with high workloads cases. This contribution tackles this problem modifying the simulation resolution of long individuals (Re-TAC). Figure 2.19 shows the relation between the execution time and the resolution used. Both graphs show that using a resolution of 60m, most of the D and E individuals are executed in less than 100 seconds.

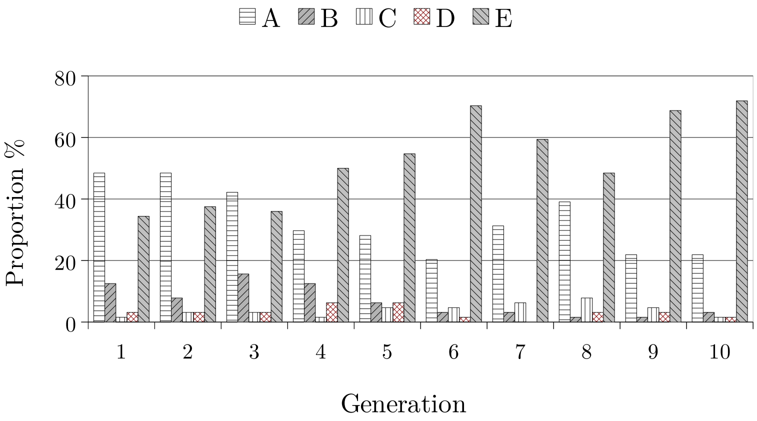
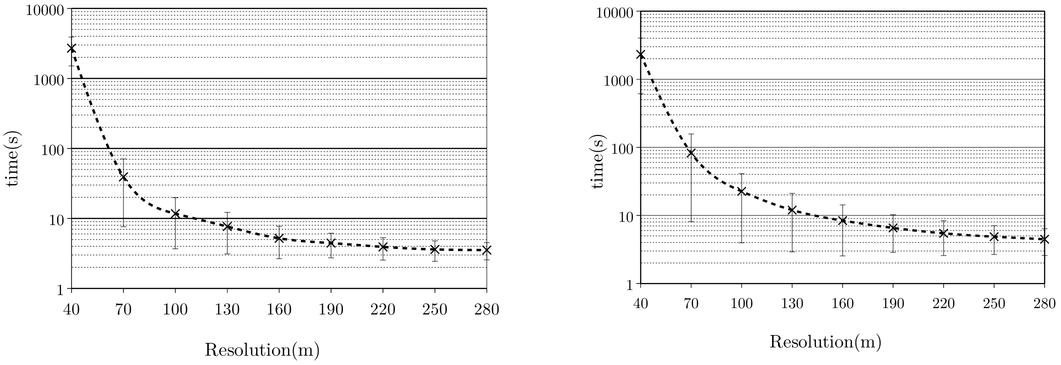


Figure 2.18: Proportion of individuals of each class for every population generation during a GA execution.



(a) D individuals.

(b) E individuals.

Figure 2.19: Execution time when changing the resolution of individuals D and E.

Nevertheless, the potential parallel time of FARSITE execution could be affected if the resolution is changed. Figure 2.20 shows how the parallel time varies when the resolution of individuals D and E are changed. When the resolution is lower than 40m (higher number, for example 70m, 100m) the parallel time is below 50%. In such situation is better to run different individuals in different cores than execute sequential runs of individuals using more than one core for each run. Thus, one core could be assign to the coarse individuals.

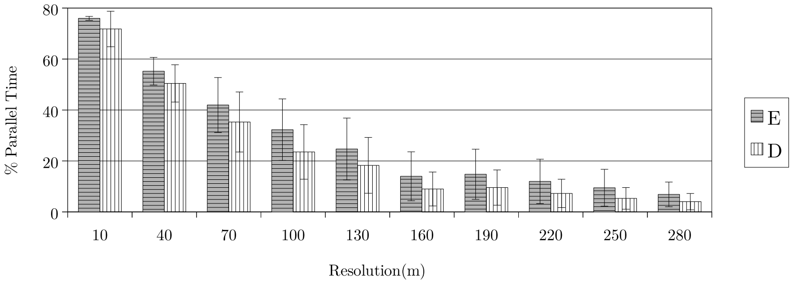


Figure 2.20: Parallel time variation related with the resolution modification.

When using such criteria, TAC with coarse resolution (ReTAC), the execution traces show that few individuals are killed (see figure 2.21). Therefore, Re-TAC avoids local minimums that TAC strategy could not skeep and provides betters results in terms of accuracy of the prediction (see figure 2.22).

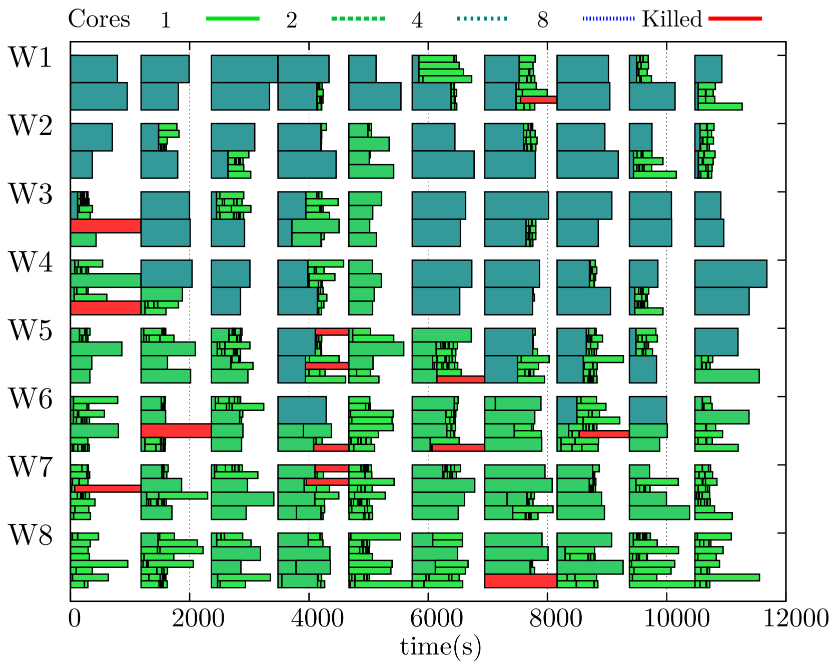


Figure 2.21: Trace of the Re-TAC-Two-Stage prediction scheme with a generation time of 1080 seconds as limit. The GA has a population size of 64 individuals and 10 generations. The parallel MPI-OpenMP implemented used 8 workers where each worker has 8 available cores.

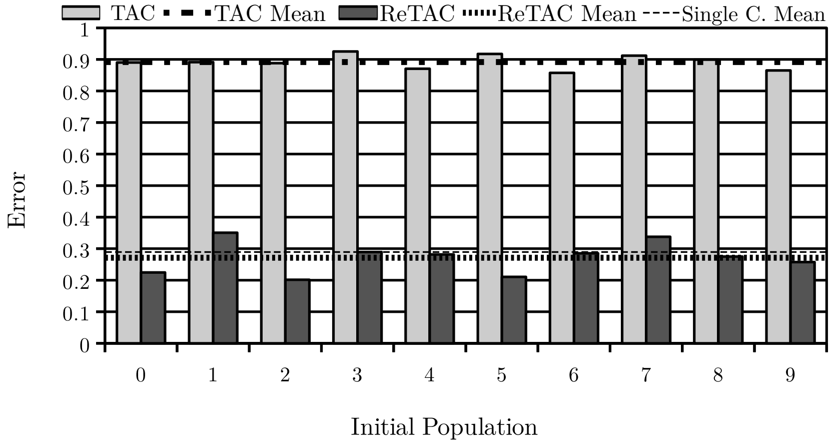


Figure 2.22: Error when comparing TAC and ReTAC using the same initial population for 10 different experiments.

2.2.2 Soft generation time deadline

The previous approach to deal with high workload fire cases relies on information about how the simulation execution time varies when changing the resolution. These information could be collected during the characterization process needed for the individual classification process. However, this implies an increment of the computation time of the characterization. This section introduces a different approach that is independent from the classification method.

The problem when TAC is used with high workload cases, lies in the hard deadline per generation. This deadline is computed dividing the remaining time between the pending generations. This deadline is caused because of the GA algorithm. The Master can not continue to the next generation without having all the fitness results of the individuals. Figure 2.23 shows that issue.

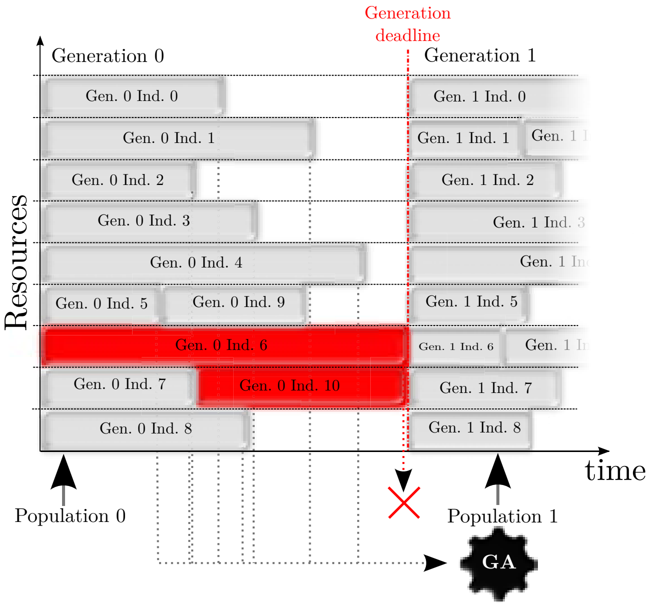


Figure 2.23: TAC deadline per generation that causes a high number of killed indivuals when working with high workloads.

Since the GA could be considered a bioinspired algorithm, this generation deadline is not very faithful with the reality. For that reason, the generation hard deadline is removed. The parallel implementation of the GA is modified to remove this deadline and be able to work with a variable population size and individuals of different generations. The scheme of this approach is depicted in figure 2.24.

Table 2.3 lists the time limits of the classes when the deadline for the calibration stage is set up to 3 hours. As it is shown in the table all E individuals, with an expected execution time greater than 2273 seconds, are not executed, they are replaced by A individuals when using the TAC policy. The behaviour of the three different approaches proposed to fit in the preset Two-Stage execution time are summarized in table 2.4.

The three different approaches have been executed with 10 different initial populations. The best error obtained for each version and the mean of each version are

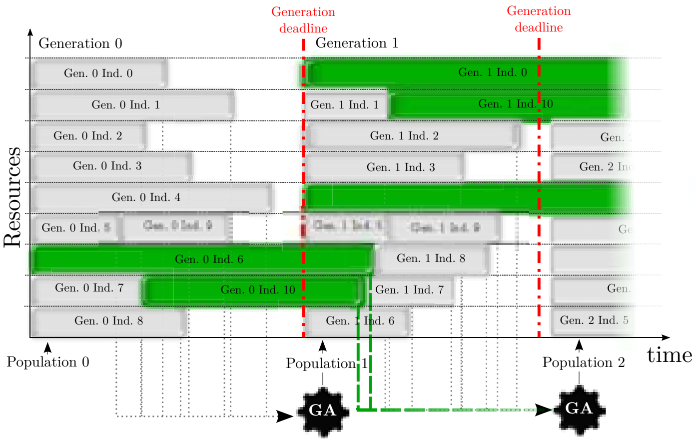


Figure 2.24: Proposed approach to improve the result of the GA with high workloads avoiding a high proportion of killed individuals.

Class	Time limits (sec.)	Cores
A	$0 < t_s \leq 1080$	1
B	$1080 < t_s \leq 1547$	2
C	$1547 < t_s \leq 1966$	4
D	$1966 < t_s \leq 2273$	8
E	$t_s \geq 2273$	0

Table 2.3: Time limits with 1080sec time available at first generation.

Approach	$t_{exe} > 1080sec$	Replacement
HR	killed	Yes
HNR	killed	No
SNR	Executed	No

Table 2.4: Proposed approaches with real time restrictions for the first generation.

shown in Figure 2.25. In this figure, the previous result is corroborated. The replacement of E individuals by faster A individuals affects calibration when comparing HR with HNR. In addition, SNR improves with respect to HNR. HNR tries to execute E individuals, but seems to kill most of them, unlike SNR. The mean error generated by each approach describes a trend that indicates the significance of the E individuals in some scenarios.

The actual behaviour of the execution of the GA, for one particular case (Case 9 in Figure 2.25), using the three approaches (HR, HNR, SNR), is shown in Figures 2.26, 2.27 and 2.28, respectively. It can be observed that, when class E individuals are replaced by class A individuals, these class A individuals fill up most of the cores and the efficiency is very low. Actually, in Figure 2.26, which shows the execution

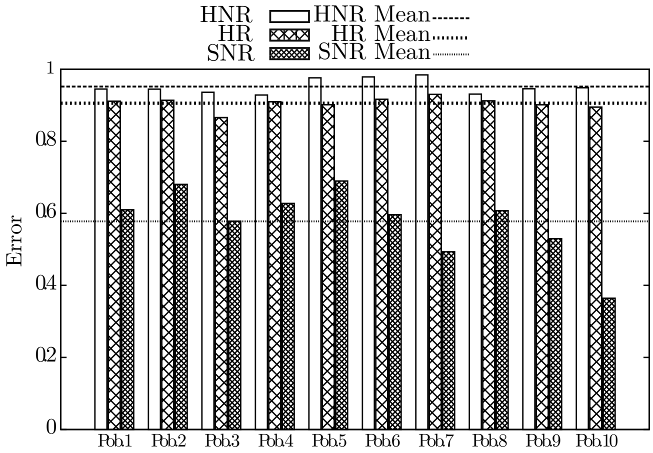


Figure 2.25: Best error obtained during the calibration of the four versions of each initial population.

trace of the hard deadline with replacement scheme, there are iterations of the GA with idle cores. This means that there are not enough individuals to take advantage of the computing resources available. However, there are several individuals that are killed because they take too much time. This means that computing resources are not used in the best possible way, execution time is not reduced and calibration is not very successful.

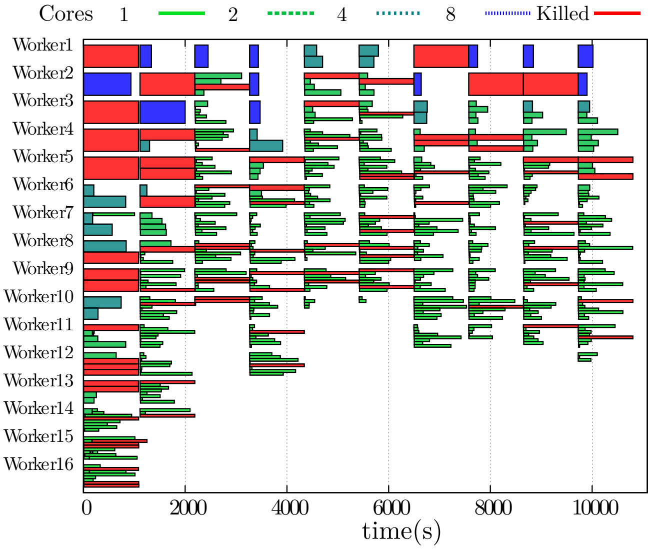


Figure 2.26: Trace of GA execution of 9th initial population with HR

In Figure 2.27, which shows the trace of the execution of the hard deadline without replacement, there are many individuals that are killed (red slots) because the iteration time limit has been reached. This means that the work carried out executing such individuals is not productive, since the individuals are not completed. So, actual efficiency is also very low, considering the red slots as non-productive computation. As mentioned above, the calibration result is not significantly different from the one reached by applying the HR scheme.

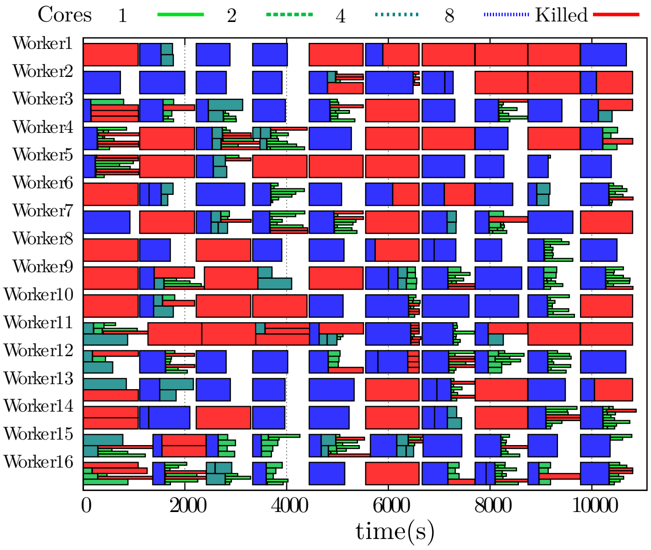


Figure 2.27: Trace of GA execution of 9th initial population with HNR

Finally, in the trace obtained applying the SNR scheme, it is observed that class E individuals are normally executed and, when the iteration time is reached, the GA is applied considering the number of completed individuals. In this case, there are almost no idle cores and only those individuals executing when the total time limit is reached are killed. In this case, the calibration reached is significantly better because those individuals in the E zone enrich the population and allow for the exploration of different areas of the search space.

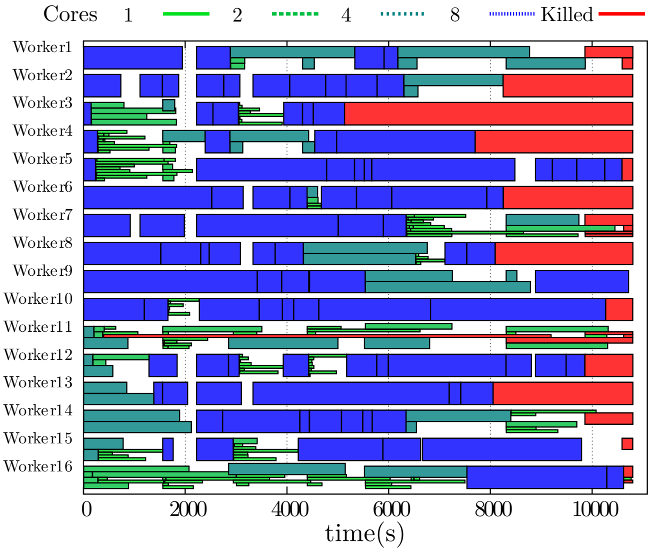


Figure 2.28: Trace of GA execution of 9th initial population with SNR

Chapter 3

A First Approach of an Operational Forest Fire Spread Prediction System

There is a lot work done about forest fire spread modelling. Nevertheless, when talking with firefighters about fore fire spread simulation tools the conclusion is that there is not an operational system that they can use. Such kind of systems require a lot of data, which comes from different sources and formats. However, any tool that should be used as a decision support system by the forest fire fighters, should be user friendly and without complex execution commands.

This section is devoted to explain the development of a forest fire spread prediction systems, which main goal is to provide a fire spread prediction without human interaction. Such task was carried out in the Institute for Environment (IES) and Sustainability in the Joint Research Centre (JRC) of the European Commission, in Ispra (Italy). The possibility of working in that centre was a chance to develop a forest fire spread simulator at European level. In order to develop the system, the same data sources mentioned in the section 2.1.6 were used.

The IES group is in charge of the European Forest Fire Information System (EFFIS). The EFFIS team deals with pre-fire and post-fire data. This task goes from computing fire danger forecast, monitor fire events, vegetation regeneration, fire detection to the carbon emission assessment. To do such tasks, they use the data usually required for forest fire spread modelling. Thus, the main goal of the stage was to include the forest fire spread module to EFFIS (figure 3.1).

One of the most important EFFIS tasks is the detection and monitoring of large

Figure 3.1: Services provided by EFFIS remarkin the place where the forest fire spread module should be coupled.

forest fire events. The time at what a fire is detected, will determine the start up time of the forest fire spread prediction. This first version of the operational system does not use the Two-Stage prediction scheme. The main reason for that fact is the computing resources required and, on the other hand, the initial approach was consider only for testing the basic prediction approach.

Next section shall describe the process required for run a simulation when the fire event is detected. The process is divided in two sections: the first one describes the static data that is needed for FARSITE and WindNinja and, the second block depicts the process to deal with meteorological data which changes during the simulation time.

3.1 Static Map Data

When a fire is detected, the first perimeter describing the burned area is computed by the EFFIS team and is immediately uploaded to the EFFIS database. Afterwards, the deployed system is capable to make a query about such fire, then the centroid of the area, which is going to be used as spatial reference, is computed. With this spatial reference, the system determines the Universal Transverse Mercator (UTM) [48] zone with the World Geodetic System 84 (WGS84)[49][50]. This step is mandatory because FARSITE is a 2-dimensional spread simulator that works with projected data.

Following, the DEM and fuel map are clipped using a window of $1600km^2$ $(40kmx40km)$ centered in the previously mentioned centroid. This size has been established from the analysis of the bigger forest fires happened in Europe since 2009. In this way, a DEM and fuel map clip of the same size are obtained. Next, slope, aspect and direction are computed from the DEM file using GDAL libs for Python. Finally, the FARSITE LCP for the give fire ocurrence is generated. The entire process is depicted in Figure 3.2.

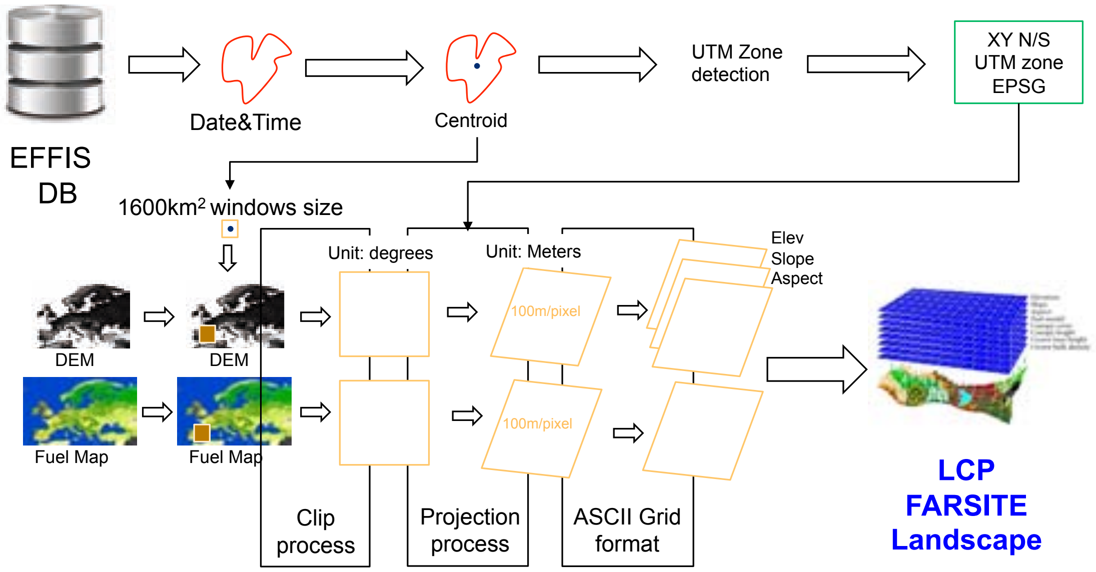


Figure 3.2: Scheme of the process carried out to generate the FARSITE LCP.

3.2 Dynamic Meteorological Data

As it has been mentioned in section 3.1, the centroid of the event is obtained from the initial fire perimeter. Afterwards, using this centroid and the GRIB API [51] provided by the ECMWF for Python, the four nearest points of the 2-dimensional mesh can be pinpointed. For each nearest point, a distance is computed and used as weight for interpolation that is carried out for each time-step. This pinpointing process is done for each variable since there is a GRIB file for each variable. The parameters that are requested to the GRIB files are temperature, dew point, pressure, and wind vectors. Then, WindNinja is applied to obtain the wind fields for the different time-steps required for the time horizon of the simulation (24 hours in the preliminary tests).

The rest of FARSITE files such as weather file, gridded wind file and ignition file are created. The scheme of the process is shown in Figure 3.3.

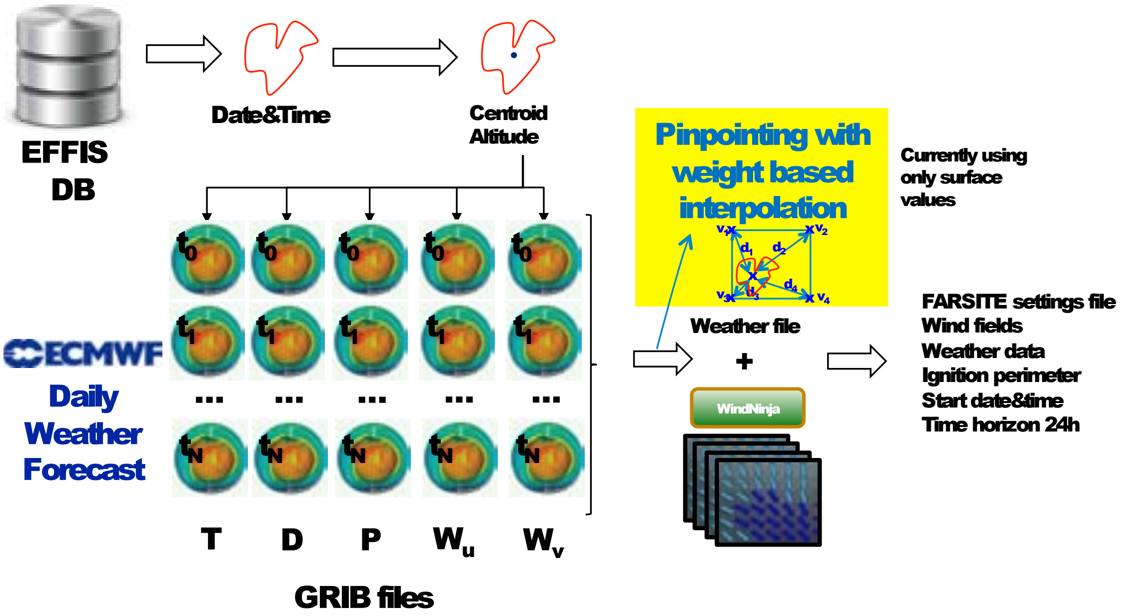


Figure 3.3: Scheme of the process carried out to generate the FARSITE weather file, wind fields and the ignition file.

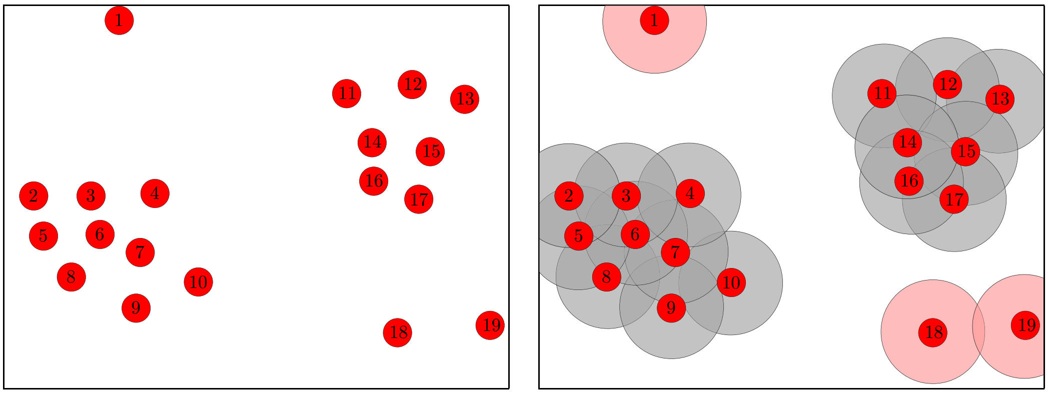
3.3 Hotspots and Fire Perimeters

EFFIS team generates the perimeters that depict the burned area with a high degree of accuracy. However, as it is explained in [52], there are some limitations in the current method. For instance, if a considerable amount of clouds is presented in the MODIS images [53][54], EFFIS's operators could not provide the burned area. For that reason, some perimeters are currently not available in the EFFIS database. In addition, large fires use to produce pyrocumulus. That fact causes that the perimeters of these fires were not always available in a dayly bases. EFFIS database is oriented to storage the burned area, so they can wait until the fire stops and there are no clouds to get some clear MODIS images. However, for operational simulating purposes, the fire perimeters are required in, at least, daily bases. For that reason, during the stay, an algorithm that builds fire perimeters was developed. The algorithm uses the MODIS Fire Anomalies product also called hotspots. Such product is a set of points over a global map, where each point represents a fire anomaly at a give instant of time. The area of interest where the fire is taking place is selected and the hotspots of the fire are filtered [55]. The filtering process selects the hotspots with two different criteria: first, only the MODIS hotspots fire anomalies [56] with a confidence superior to 80% are selected. Secondly, those hotspots, which do not have a number of points inside a radius are discarded. This steps corresponds to figures 3.4(a) and 3.4(b).

However, it is necessary to create clusters of these points because sometimes, forest fires happen at the same time in the same area. Therefore, a set of graphs are created using the nearest neighbor graph (NNG) with the hotspots as nodes. This method is shown in figure 3.4(c) until figure 3.4(g).

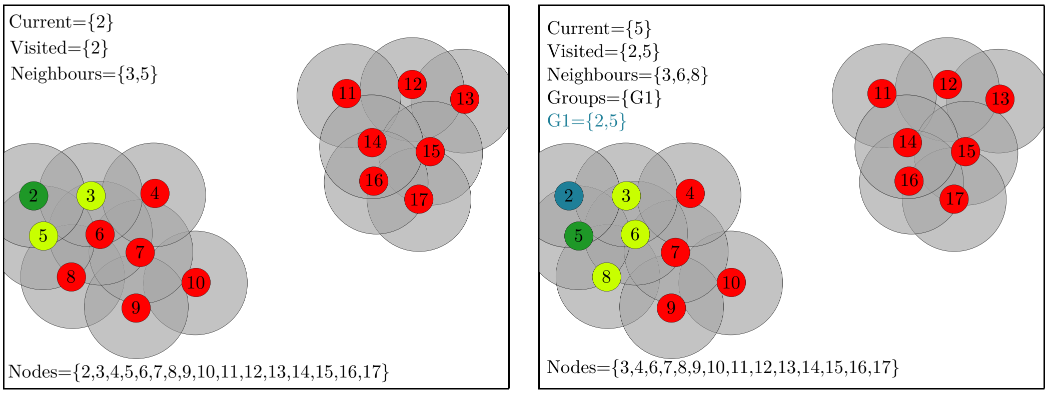
Right after, a Delaunay triangulation [57][58] is done for each graph on the area of interest. Afterwards, the edges of the set of triangles, whose length is less than a value called alpha, are discarded. This last process is done in a loop searching for the higher value of alpha that does not divide a set of triangles in multiple and not connected groups. Finally, all the edges are discarded excepting the external edges of the polygon. This last two steps are depicted by Figure 3.4(h) and 3.5.

In conclusion, with this algorithm the fire perimeters can be obtained using hotspots as inputs. The time stamp assigned for the whole perimeter is the last hotspot of the perimeter. This algorithm needs as parameters the radius and the minimum number of neighbours used in the density barrier as well as the maximum distance to create a distance based graph and the maximum lenght of the edge for the Delaunay triangulation. When playing with such parameters appears the most interesting part, the algorithm could be used to detect the active fire fronts. In addition, only the last hotspots of the day could be used, so the output of the algorithm would be the last active fire fronts. The only concern of this method raises by the resolution of the hotspots, which is about 1km.



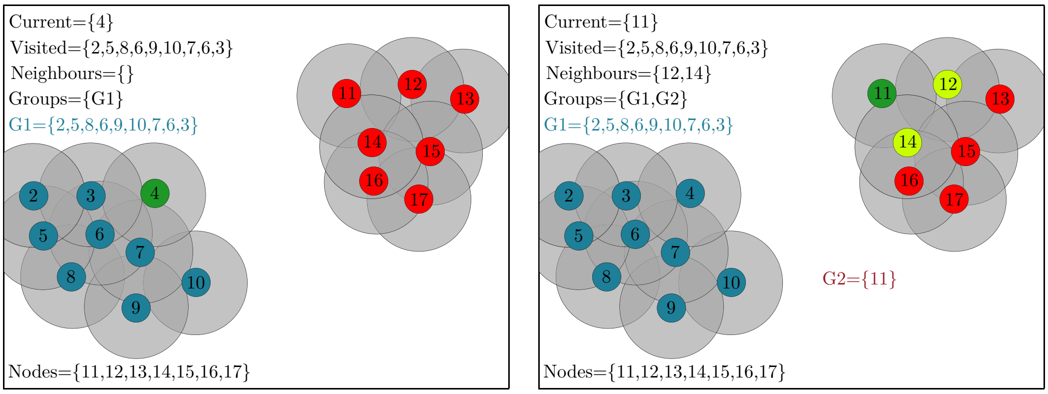
(a) Raw hotspots.

(b) Filter by density.



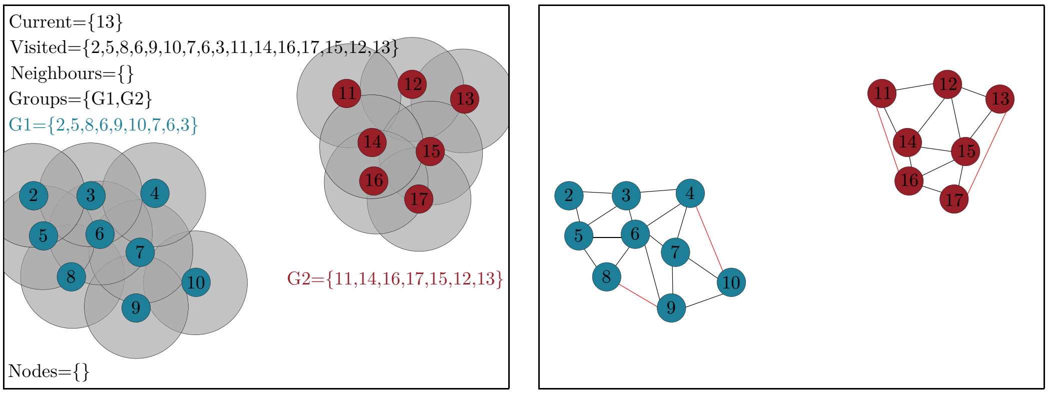
(c) Clustering.

(d) Clustering.



(e) Clustering.

(f) Clustering.



(g) Final groups.

(h) Removing edges after Delaunay triangulation.

Figure 3.4: Scheme of hotspots algorithm to obtain fire perimeters.

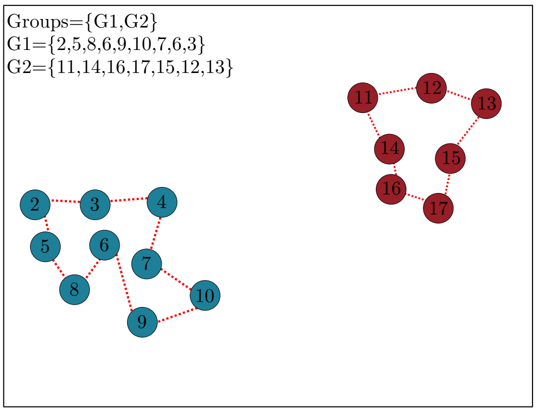


Figure 3.5: Output of the hotspots algorithm to obtain fire perimeters.

3.4 Preliminary Results

Once the first version of the operational forest fire spread prediction system was developed, the system was tested using past large fires without available daily perimeters from EFFIS [59], but using the hotspots algorithm described in the previous section. Following, the results for some of the tested cases are going to be explained.

3.4.1 Sardegna (Italy) 2009

The first tested case happened in Sardegna, Italy on 23/07/2009. The EFFIS database only provides the starting date of the fire (23/07/2009). The final day was detected applying the hotspots algorithm (25/07/2009). Figure 3.6 shows the hotspots detected during the duration of the forest fire. Furthermore, the detected active fire areas using the hotspots algorithm is depicted in daily bases. Then, the system fetched the required meteorological data (figure 3.7) and computed the corresponding wind fields using WindNinja (figure 3.8).

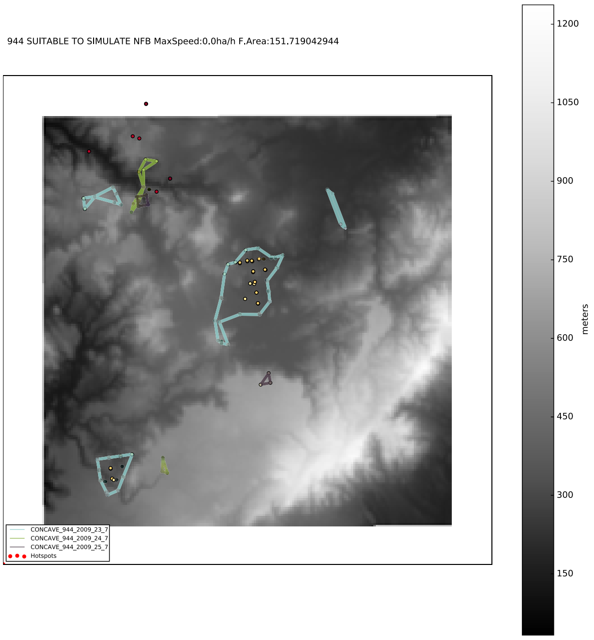


Figure 3.6: Hotspots and active fire zones. Elevation in grayscale at background.

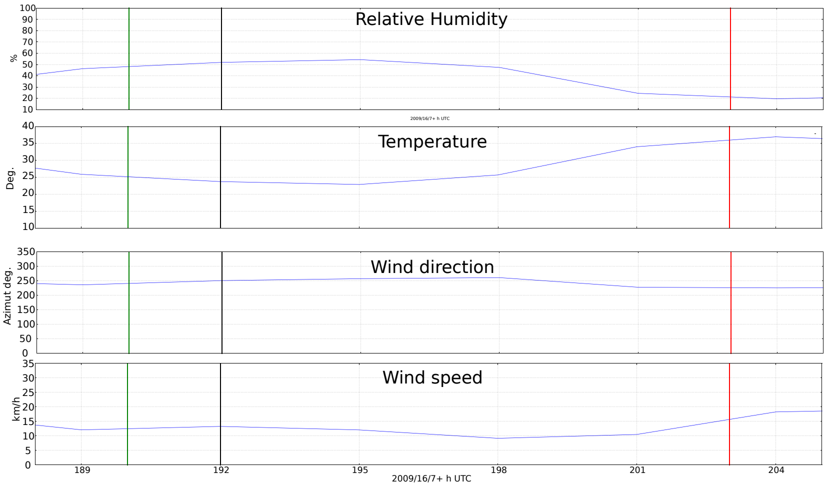


Figure 3.7: Meteorological data included in the summary of the simulation.

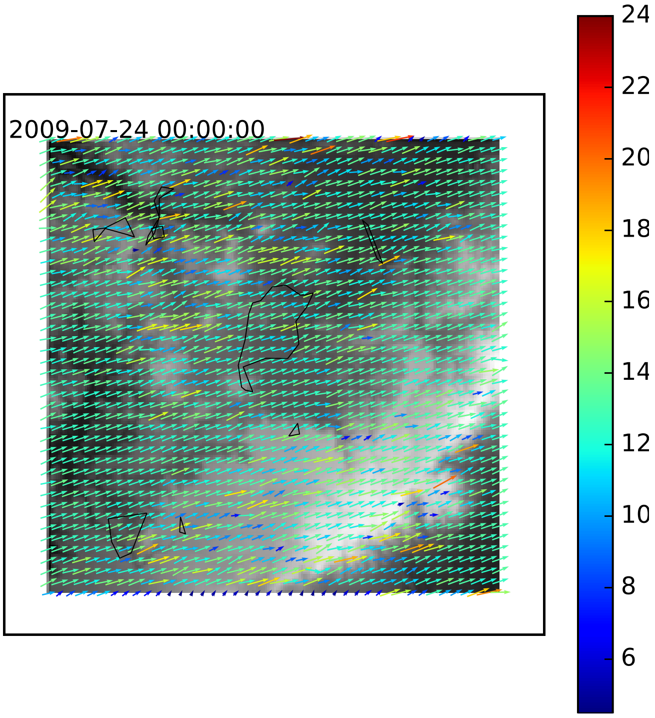


Figure 3.8: Wind fields used for hour 0 and 3 at 24/07/2009.

When looking at the output of the simulation in figure 3.9, it can be seen that four different fire fronts are simulated. None of them have a broaden spread. Nevertheless, the perimeter at north-east has a critical zone where the fire is spreading faster than on the rest of active fire fronts. The reason of such behaviour could be the topographical characteristic that causes high wind speed when the wind comes from south-west.

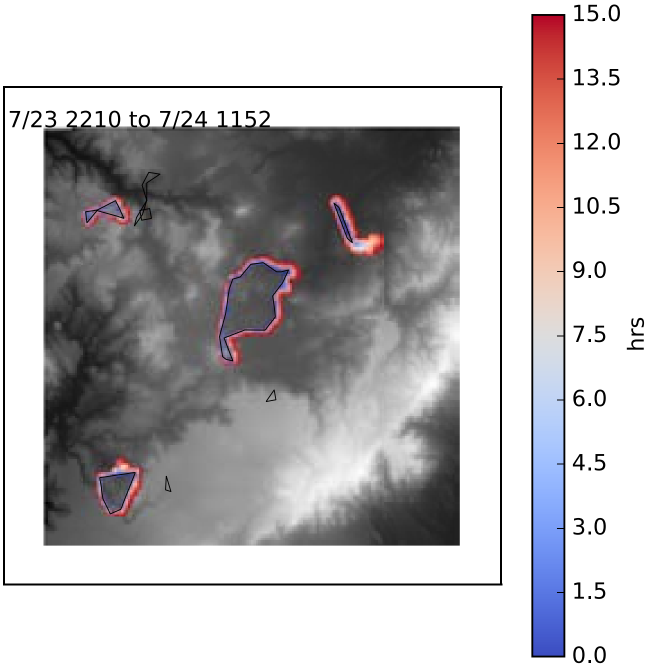


Figure 3.9: Forest fire spread simulation with a time horizon of 24h.

3.4.2 Valencia (Spain) 2012

This case is one of the biggest forest fires that had happened in Europe. This case took place in Valencia, Spain on June 28th, 2012. The creation of the daily perimeters was impossible because of the combination of the pyrocoumulus and clouds (see figure 3.10). Only a final perimeter obtained days after the fire was available. Thus, there was no fire evolutions and the case was not suitable to be simulated.

(a) 28/06/2012 (b) 29/06/2012 (c) 30/06/2012 (d) 01/07/2012

Figure 3.10: Corrected Reflectance MODIS images during the forest fire in Valencia.

The algorithm to obtain daily perimeters using hotspots was applied to the Valencia fire. This case is ideal for the algorithm because of the large burned area. Figure 3.11

shows the result provided by the algorithm. As it can be observed, using the hotspots analysis, one comes out with one perimeter for each day.

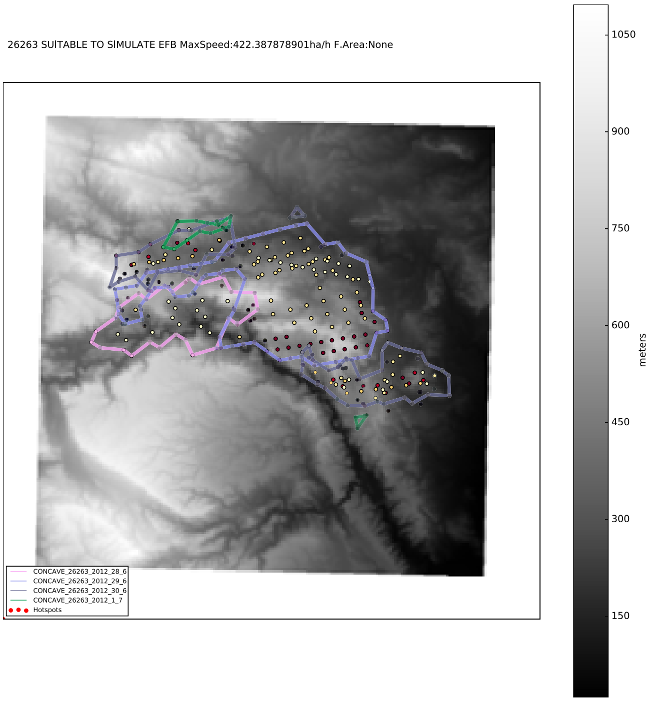


Figure 3.11: Hotspots and active fire zones for the Valencia fire.

Once the perimeters were obtained, the next test was to use the operational system to predict the fire evolution. In this case, the meteorological data used was analysis weather data instead of forecasted data. Figure 3.12 shows the results for the first day simulation. The fire seems to spread in three different directions. The first one is to the south where there is a river and a little lake, which are not reflected in the fuel map. Secondly, there is a fire spread direction to the north, where the firefighters where working with the aerial resources. The last fire spread direction is to east. This direction is expected to be the most dangerous, the wind is blowing, in mean terms, through this direction.

The prediction results for the second day are shown in figure 3.13. Finally, the third day of simulation is reported in figure 3.14.

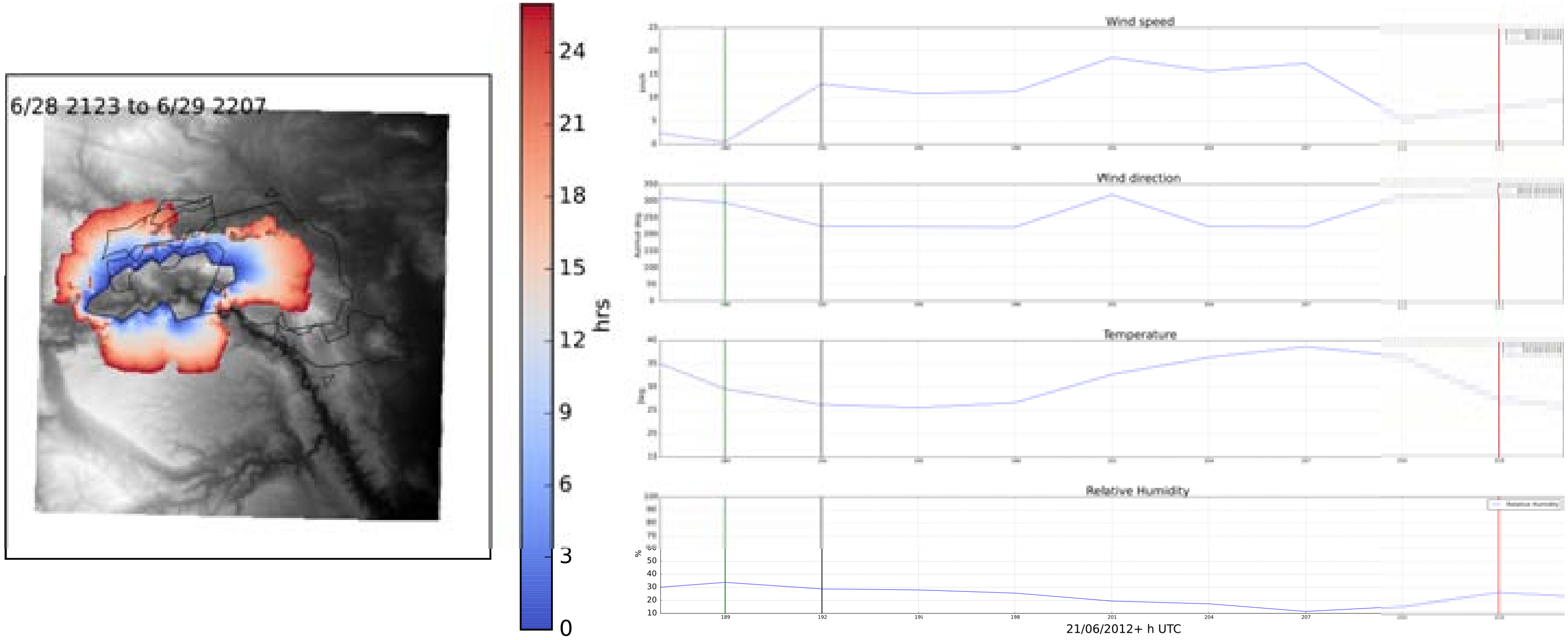


Figure 3.12: Valencia first day simulation from 21:23 to 22:07, and the meteorological data during the simulation.

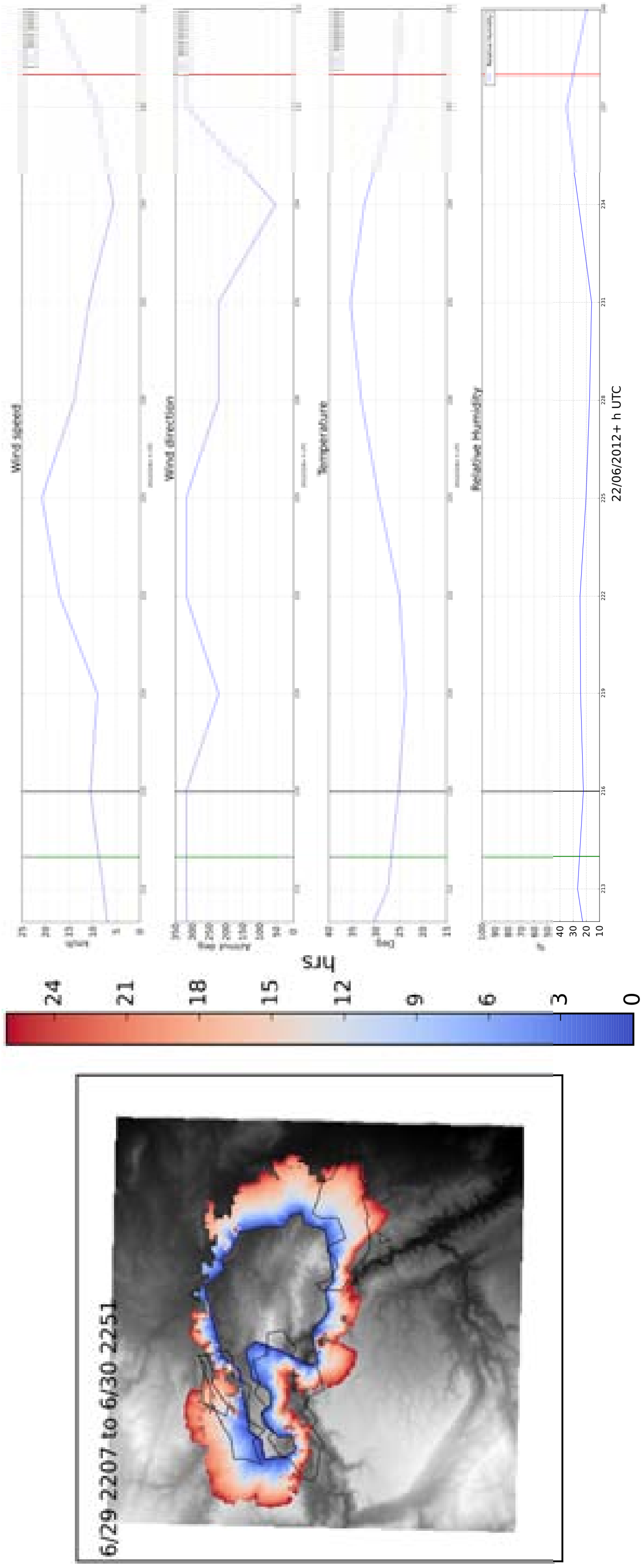


Figure 3.13: Valencia first day simulation from 22:07 to 22:51, and the meteorological data during the simulation.

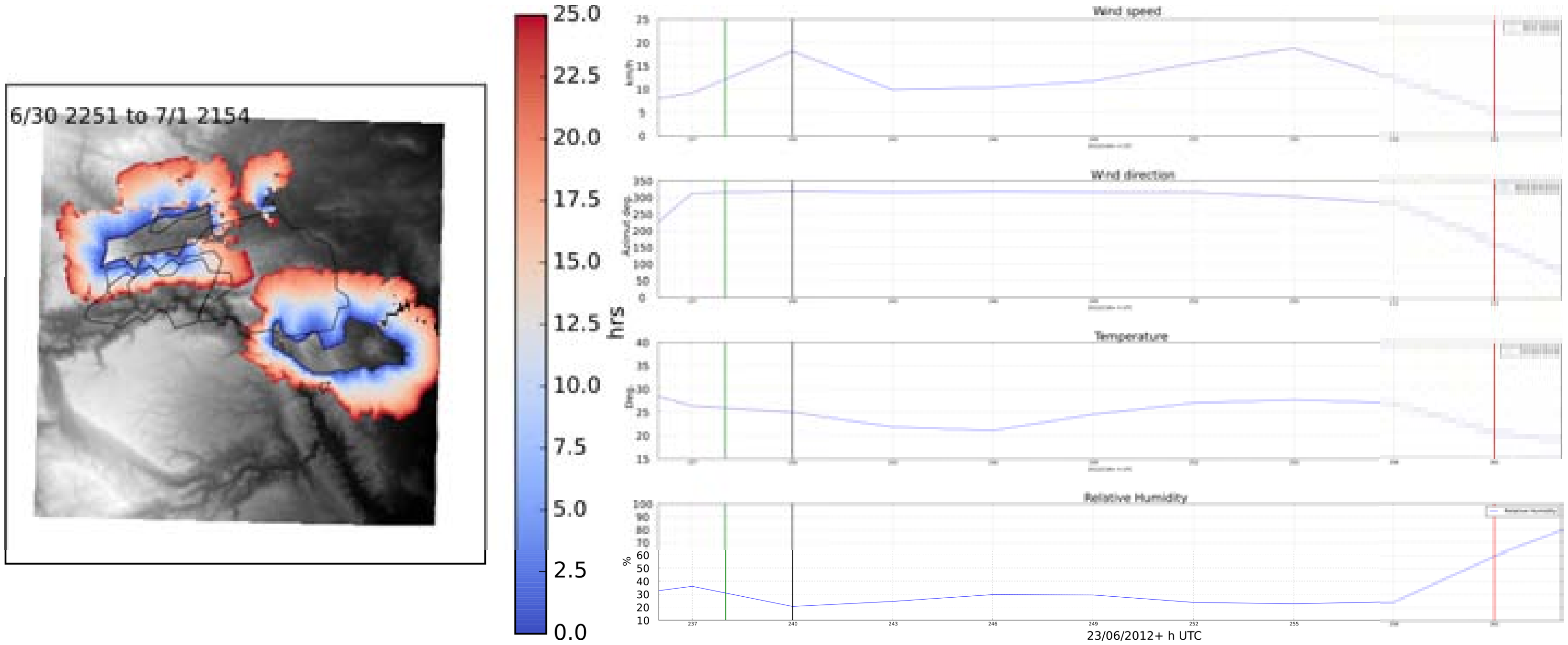
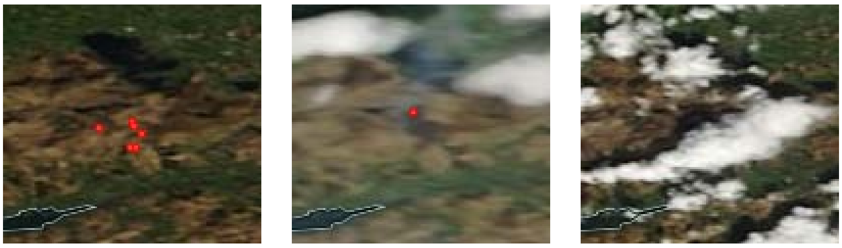


Figure 3.14: Valencia third day simulation from 22:51 to 21:54, and the meteorological data during the simulation.

3.4.3 Muckros (Ireland) 2015

The examples previously shown are past cases retrieved from the EFFIS database. The example reported in this section, is a real time test case because the prediction system was run during the ongoing hazard. This forest fire took place on April 9th, 2015, in Muckross (Ireland). The weather forecast used in this case was the HIRES [60] from ECMWF that was available on April 10th, 2015 after noon. The execution of the system took less than 10 minutes. The summary generated is shown in the Appendix B. The summary shows a minor fire spread. The temperature was expected to be low, mostly lower than 10 degrees. In addition, the relative humidity was over 65% during most of the time. The wind speed varies between 10km/h and 15km/h. Besides, when looking to the wind fields, the topography reduced the wind speed in the fire area. Thus, the prediction produced by the system seems reasonable taking into account the input parameters values.

Once the fire was extinguished and using the MODIS near real-time world viewer [61], the fire evolution could be obtained using the corrected reflectance and the hotspots. The evolution of this images can be seen in figure 3.15. From the first image, it seems that the fire was active on April 9th, one day before before the only available perimeter in EFFIS. The next day, only one hotspot is detected, but it seems that the fire was active because of the smoke emission generated by the fire. The subsequent day, April 11th, there is no hotspots neither smoke columns in the image. Therefore, by the MODIS images, it seems that fire started on April 9th and it finished on April 11th, with a minor fire activity from 10th to 11th of April.



(a) 09/04/2015 (b) 10/04/2015 (c) 11/04/2015

Figure 3.15: Corrected Reflectance MODIS images during the forest fire in Muckross, Ireland.

Chapter 4

Conclusions

In this Thesis a forest fire spread prediction system that works under strict real time constraints is proposed. The simulation kernel used in the system is FARSITE. Because of the input uncertainty involved in the models used to describe this phenomenon, the prediction method called Two-Stage is used because it deals with this problem by including a input parameters calibration stage, previously to the prediction. Such calibration approach requires high performance computing resources and increases the final execution time of the system. This Thesis proposes different approaches to accelerate the calibration stage of the prediction system by exploiting the multi-core architectures.

For that reason, FARSITE was profiled and parallelized using OpenMP pragmas. This fact opens the chance to reduce the unbalancing problem of the MPI implementation of the GA and speedup the calibration method. The obtained results show that using parallelism inside the nodes to accelerate the time spent in computing one GA generation, provides better results than increasing the population size.

Different core allocation policies are proposed using a method that estimates the execution time for a given scenario beforehand. Detect and speedup these individuals that take longer is not an obvious task, since the method used to know the execution time could deliver some wrong classified individuals. The results show that providing cores, not only to the slower individuals, but also to next individual class, helps to maintain the execution time and it does not increase significantly the number of used cores.

Following, the study done in section 2.1.3 analyzes an allocation policy using an heterogeneous cluster. After repeating the experimentation with 50 different initial populations, it can be concluded that there is direct relation between the proportion of individuals for each class and the final speedup. Thus, the policy is working well and the hybrid version is reducing the time in such cases where the occurrences of slow individuals is high.

Although the final execution time is being reduced, the calibration method must accomplish with a given deadline in a real context. For this reason, a time-aware core allocation policy (TAC) based on the parallelism of the simulation kernel is described in Section 2.1.4. The implemented TAC is able to provide a calibrated parameters set at time. It also reduces the unbalancing problem between workers but requires more available resources.

In order to achieve a better use of the resources using TAC, the MPI implementation of the GA is modified using non-blocking communications implementing a Master/Worker on demand. So the master process will provide more individuals as soon as the workers became idle. As it is observed in section 2.1.5, under this approach, the efficiency is significantly increased without having relevant consequences on the error (fitness function) at the calibration stage.

However, to include the TAC policy into the system, causes a misbehaviour of the GA. That is, since TAC eliminates from the calibration process those individuals that overpass the present time deadline, it could happen that these killed individuals were the ones that drives the search to a good solution. Therefore, this search area was not properly explore. In order to overcome this problematic situation, two different approaches were proposed. The first one is based on taking benefit of the classification method. So, since one can estimates beforehand the execution time of a given individual, when a long one is detected, instead of killing it, its resolution will be modified to achieve a reasonable execution time. This method depends on the classification process and increases the computational time needed for the characterization process. Therefore, another approach was proposed. This proposal relies on the bioinspired nature of the GA. The main contribution consists of relaxing the internal deadline for each GA generation. Thus, instead of killing those individuals that have not finished at the generation expiration time, one keeps them running. Therefore, the population size is changed to be dynamic and it will be continuously adapting until the global calibration deadline has been reached. Both proposals have been proven to provide good results, however the resolution based method could be considered aggressive in terms on simulation reliability and increases the execution time of the characterization. The second method that removes the hard deadline per generation shows also a good calibration error. Although the final calibration is not so good as the resolution based method, the second one is not modifying the resolution of some individuals. So finally, one comes out with different approaches to achieve real time forest fire spread prediction such as: TAC (Time-Aware Core allocation), Re-TAC (Resolution TAC) and Soft-TAC (soft deadlines TAC).

This PhD Thesis applies for the international mention because a five months stay has been done in the Institute for Environment and Sustainability at the Joint Research Centre (JRC-Ispra) in Italy. This centre provided all data related to real past forest fire cases. This data is saved in the EFFIS (European Forest Fire Inforamtion System) database. The stage represent a chance to analyze the behaviour of the Two-Stage calibration method executed with different configurations. Furthermore, during the pre-doctoral stage, an operational forest fire spread prediction was proposed, deployed and tested. The most important data source used in this work is the EFFIS database, which contains the fire burned areas, the fire anomalies (hotspots), the fuel map, which describes the fuel over all Europe and the meteorological data coming from the ECMWF. The operational systems puts the data sources and the simulation kernel all together. In addition, an algorithm to create burned area or even detect active fire fronts is proposed. Such method makes possible to obtain the fire evolution of large fires from those burned areas that were not available.

Besides, two different versions are implemented. The first one is focused on the already happened cases, useful to analyze and improve the results of the first approach of the operational system. The second one is the operational system itself. In this work, results from both of them are shown. Although being a very preliminary version, the preliminary tests arise good results and a reasonable prediction behaviour.

Chapter 5

Open Lines

The results obtained in this long-term research work are fully satisfactory and give rise to new challenges. The proposed time-aware core allocation policies aim to reduce the execution time of the calibration stage of the Two-Stage prediction scheme. The steering strategy applied is a Genetic Algorithm what relies on a fitness function (called error in our case). One pending study is a deep analysis of the fitness function used in the GA. The current fitness function based on the symmetric difference, penalises false positives in excess. Keeping in mind that the system is blind to human actions that could affect to fire spread, the overestimation must be less penalized than the underestimation. So, the proposal of a new error function could be of a great interest.

This thesis describes an implementation of a time-aware core allocation which has been proven. However the task to estimate the required resources needed for a given scenario is still a pending work. It is not an obvious task, the initial proportion of individuals of different classes could be estimated if the characterization has been done creating randomly a big amount of individuals. Besides, the GA could evolve creating a trend for some individual class. Thus, the proposed methodologies could try to improve the computational efficiency but does not completely fix that issue because of the not predictable behaviour of the GA.

As ita has been described, when working with high workload cases one comes out with time problems. Two core allocation policies had been proposed to overcome this issue, the resolution based method (Re-TAC) and the version with a soft deadline per generation (Soft-TAC). However, the best option could be a combination of both strategies determining a criteria for selecting one or the other.

After developing a preliminar version of an operational prediction system, it appears several points to improve. One pending task is the development of a database which tries to collect the calibrated fuel parameters of each forest fire occurrence in Europe. This could lead to a work of data mining to extract useful information for outcoming fire cases. In addition, the data interchange between different elements of the operational system must be changed. For instance, WindNinja could be initialized by using a netcdf file created from the GRIB files available instead of interpolating the nearest points of the GRIB mesh a use the domain average initilitzation of WindNinja. Something similar happens with the rest of the meteorological data. FARSITE is not using gridded data, except the WindNinja wind fields. Differents approaches could be developed to such task improving the meteo data that uses FARSITE.

A secondary task is a study of the algorithm that creates fire perimeters from hotspots. Such method could be improved to avoid posible clustering mistakes. In addition, the parameters could be adjusted to the fire size and, then, use the algorithm to detect only fire fronts instead of burned area.

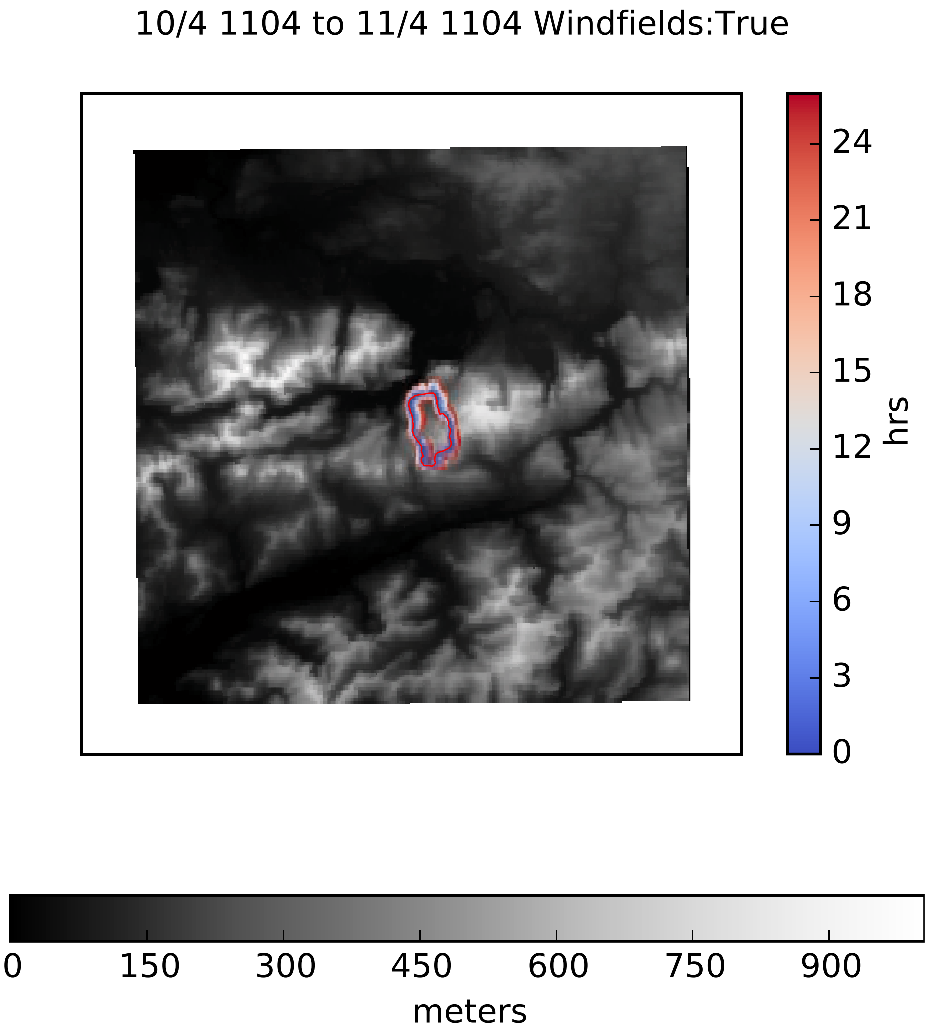
Finally, the operational system has been implemented using a 2-dimensional spread model because of the computational requirements of the atmosphere models. However, it is well know that atmosphere affects to the fire spread and viceversa. During the PhD. stay, we come out with some cases that could not be predicted using a non-coupled model as FARSITE. So, its mandatory to find some factors such as atmosphere instability, gusty winds or wind jet streams that could help just to detect if the prediction could be wrong. In addition, the atmosphere coupled fire spread models could be extremely useful to study the effects of such factors over the fire.

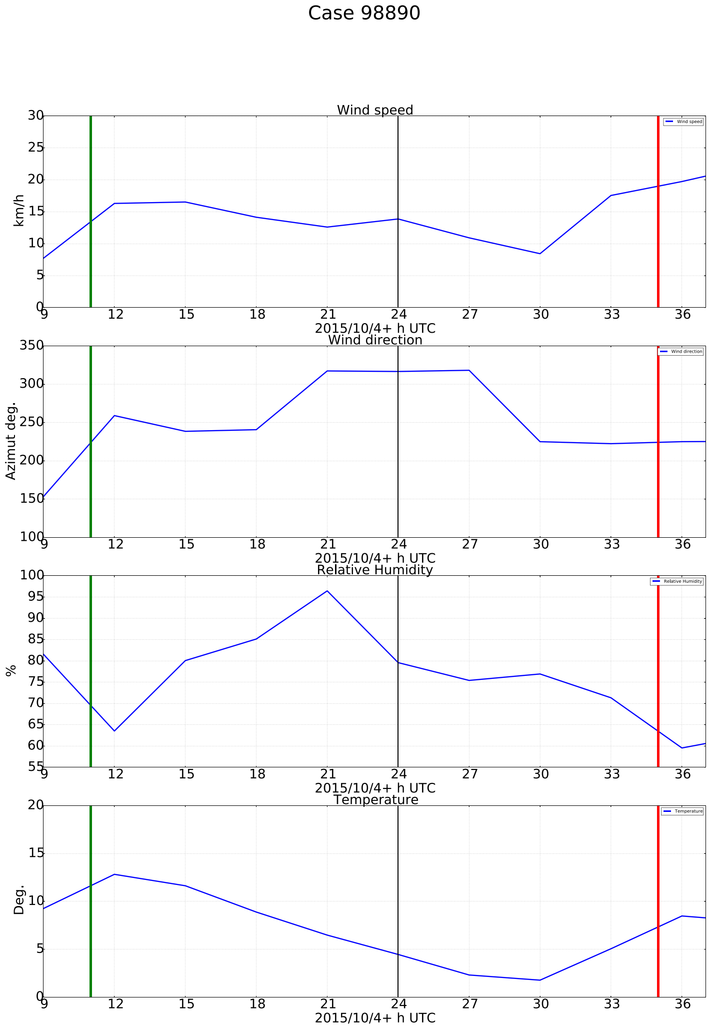
Appendices

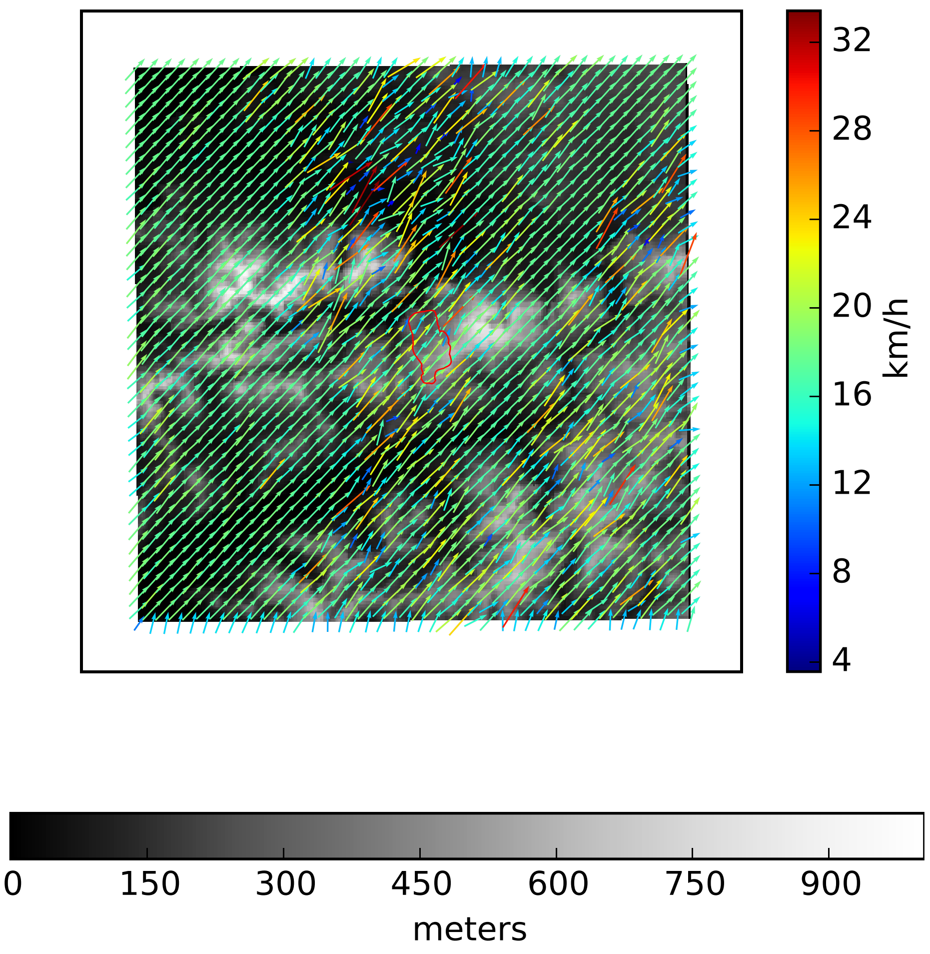
Appendices

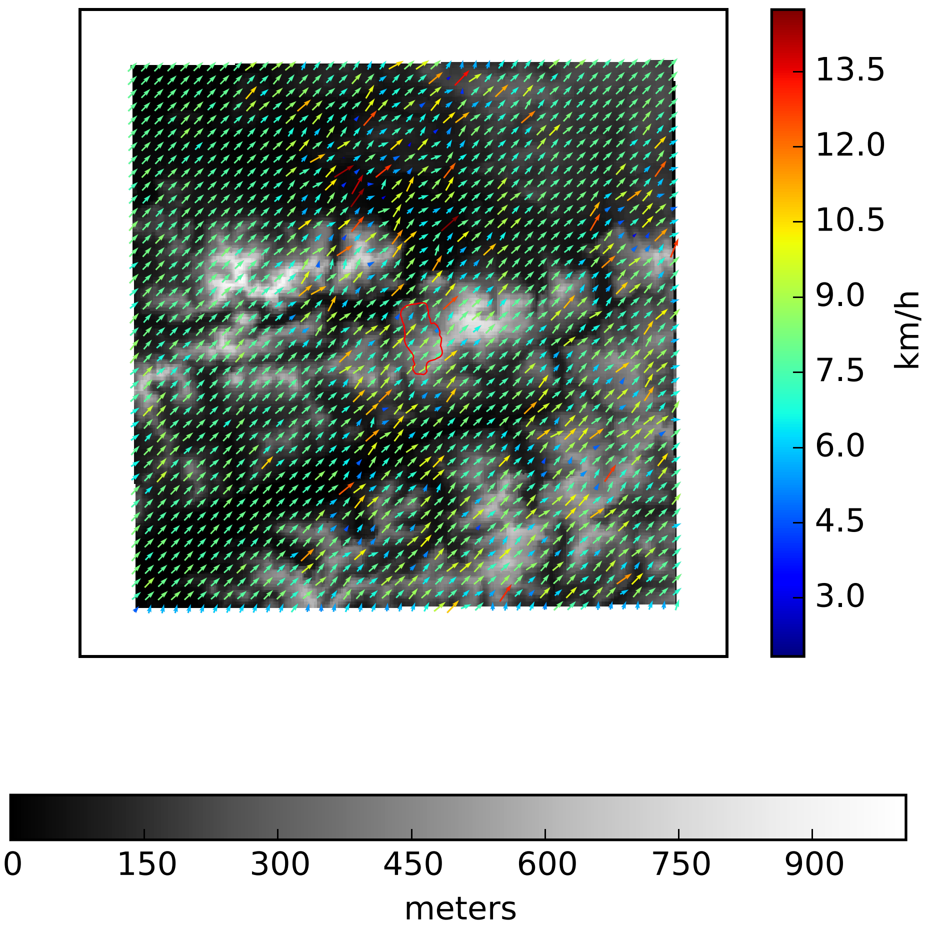
A Published Contributions

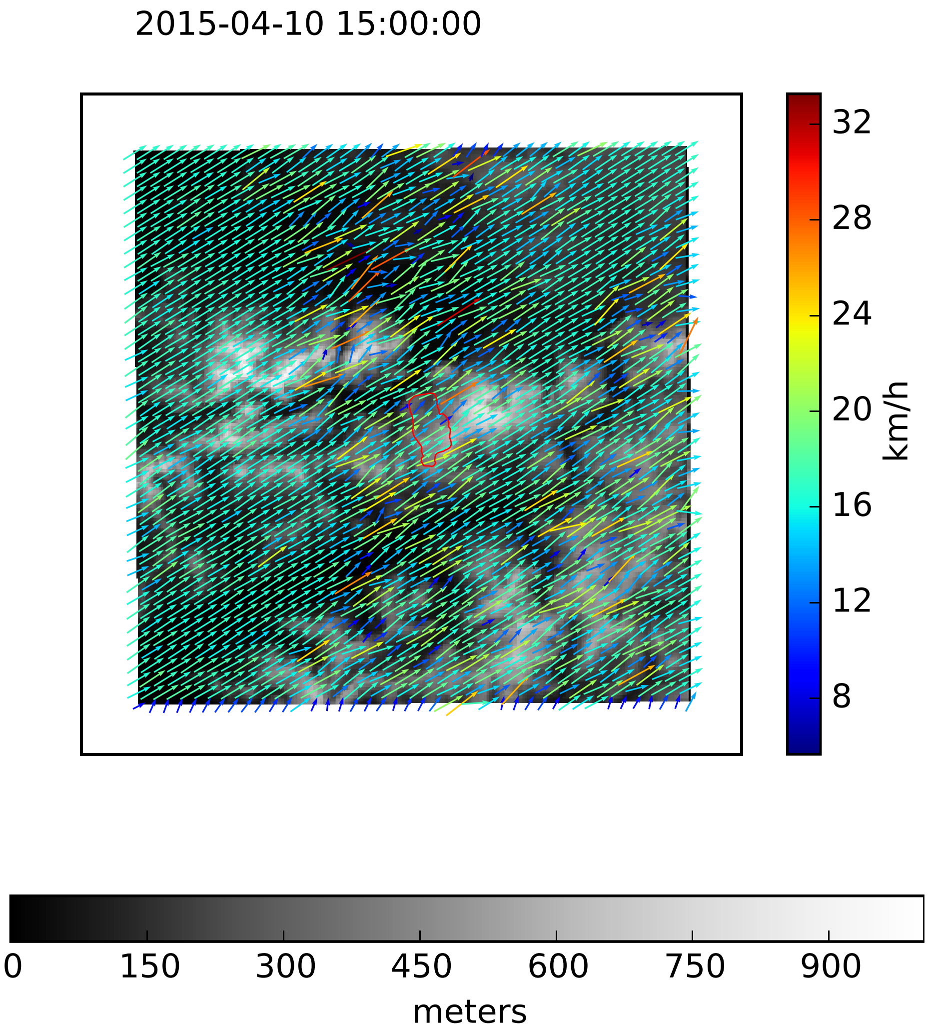
B Summary Generated by the Operational System in a real case.

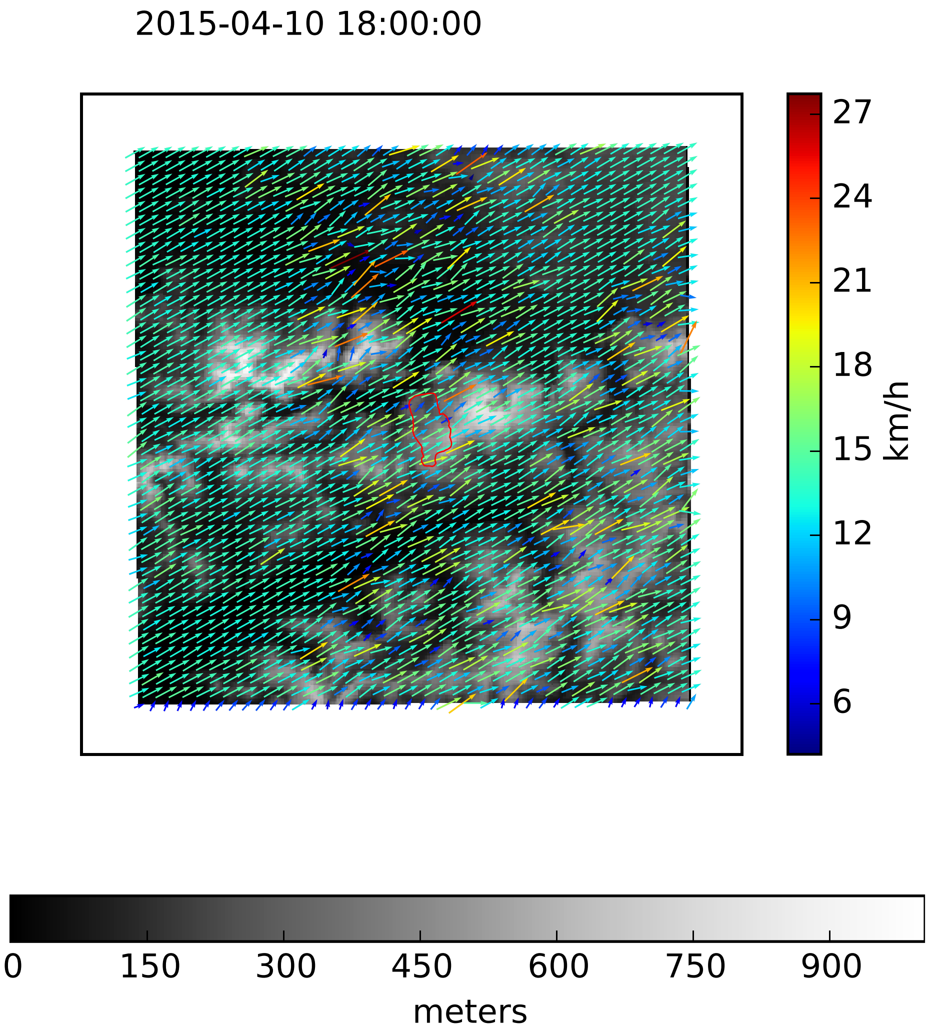

10/4 1104 to 11/4 1104 Windfields:True
hrs
0
3
6
9
12
15
18
21
24
meters
0
150
300
450
600
750
900

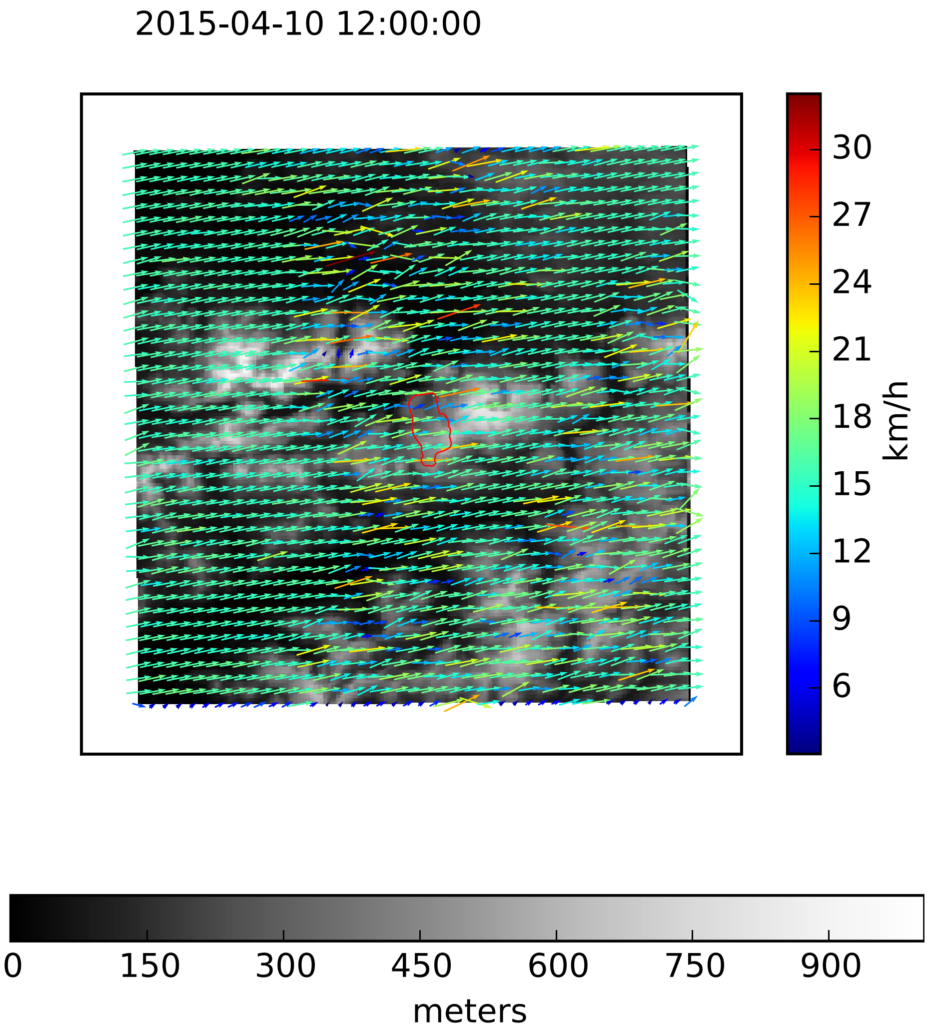

Case 98890
Wind speed
km/h
0
5
10
15
20
25
30
Wind direction
Azimut deg.
100
150
200
250
300
350
Relative Humidity
%
55
60
65
70
75
80
85
90
95
100
Temperature
Deg.
0
5
10
15
20
9
12
15
18
21
24
27
30
33
36
2015/10/4+ h UTC

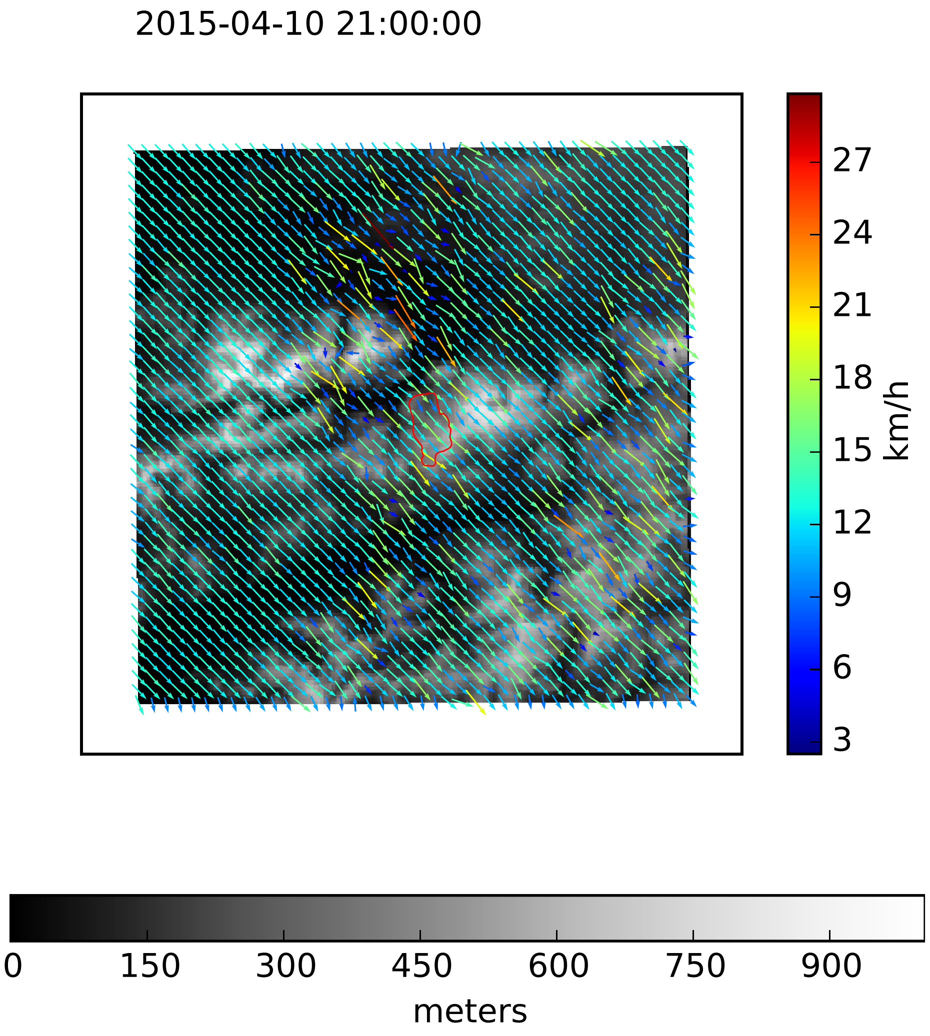

2015-04-11 09:00:00


2015-04-11 06:00:00

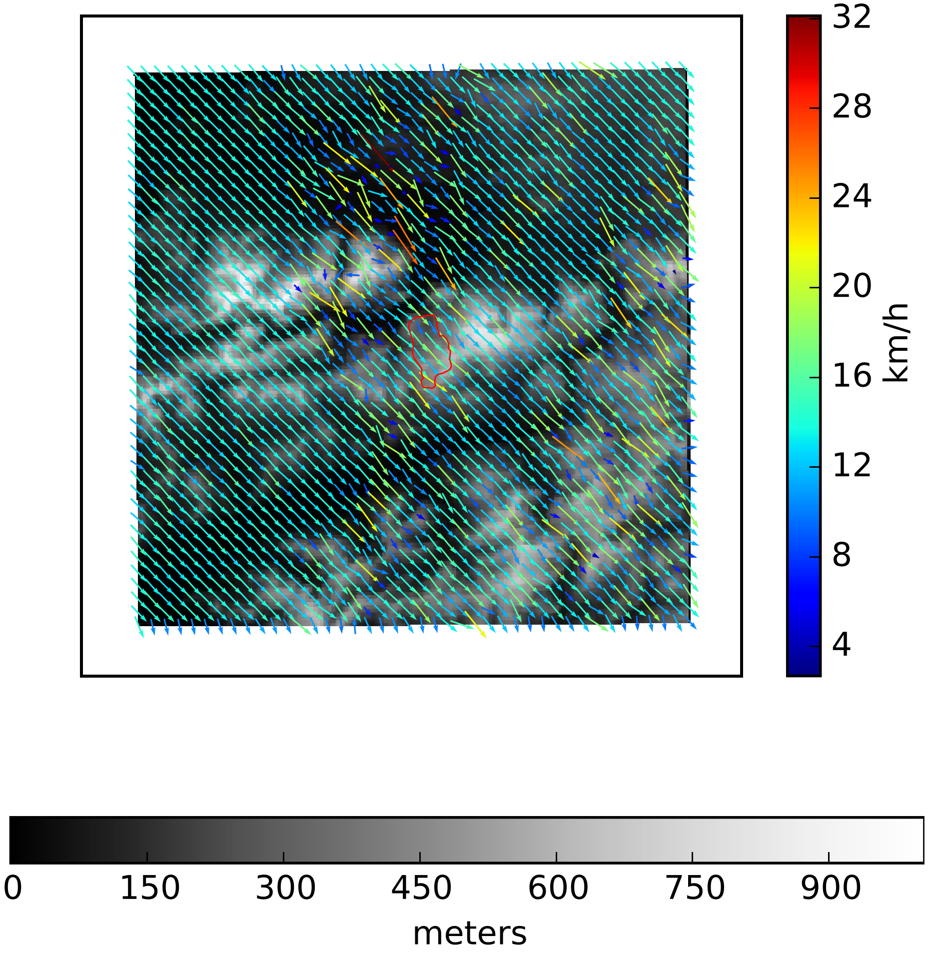

2015-04-10 15:00:00
32
28
24
20
16
12
8
km/h
0
150
300
450
600
750
900
meters


2015-04-10 18:00:00
27
24
21
18
15
12
9
6
km/h
0
150
300
450
600
750
900
meters

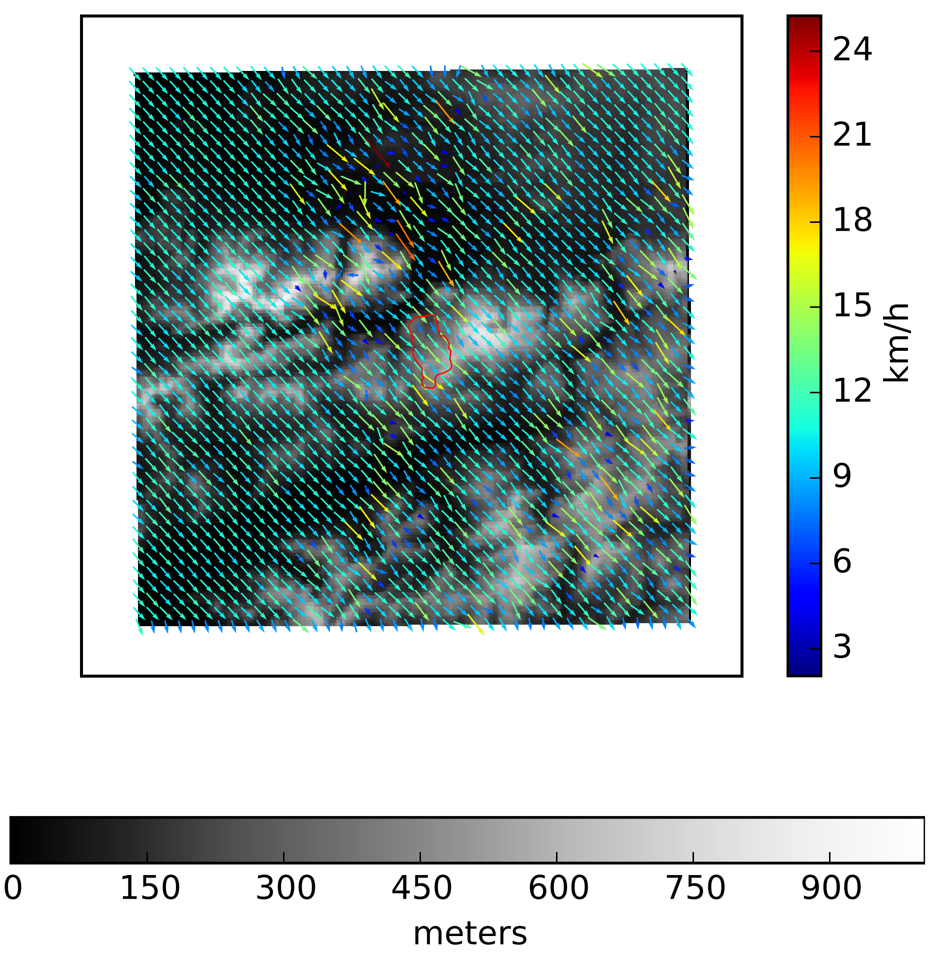

2015-04-10 12:00:00
30
27
24
21
18
15
12
9
6
km/h
0
150
300
450
600
750
900
meters


2015-04-10 21:00:00
27
24
21
18
15
12
9
6
3
km/h
0
150
300
450
600
750
900
meters

2015-04-11 00:00:00



2015-04-11 03:00:00



Bibliography

[1] CE Van Wagner, Petawawa Forest, et al. Development and Structure of the Canadian Forest FireWeather Index System . In: *Can. For. Serv., Forestry Tech. Rep.* Citeseer. 1987.

[2] Brian E Potter, Julie A Winkler, Dwight F Wilhelm, Ryan P Shadbolt, and Xindi Bian. Computing the low-elevation variant of the Haines index for fire weather forecasts . In: *Weather and Forecasting* 23.1 (2008), pp. 159 167.

[3] Lewis Ntaimo, Xiaolin Hu, and Yi Sun. DEVS-FIRE: towards an integrated simulation environment for surface wildfire spread and containment . In: *Simulation* 84.4 (2008), pp. 137 155.

[4] Seth H Peterson, Marco E Morais, Jean M Carlson, Philip E Dennison, Dar A Roberts, Max A Moritz, David R Weise, et al. Using HFire for spatial modeling of fire in shrublands . In: (2009).

[5] S Yassemi, S Dragićević, and M Schmidt. Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour . In: *Ecological Modelling* 210.1 (2008), pp. 71 84.

[6] Janice L Coen, Marques Cameron, John Michalakes, Edward G Patton, Philip J Riggan, and Kara M Yedinak. WRF-Fire: coupled weather wildland fire modeling with the weather research and forecasting model . In: *Journal of Applied Meteorology and Climatology* 52.1 (2013), pp. 16 38.

[7] Richard A. Tapia, Cynthia Lanius, Cassandra M. Mc Zeal, and Teresa A. Parks. *Computational Science: Tools for a Changing World.* Ed. by Rice University. URL: `http://ceee.rice.edu/Books/CS/index.html`.

[8] Jesse Ruan, Raed El-Jawahri, Chai Li, Saeed Barbat, and Priya Prasad. Prediction and analysis of human thoracic impact responses and injuries in cadaver impacts using a full human body finite element model . In: *Stapp car crash journal* 47 (2003), p. 299.

[9] David T Butry, ED Mercer, Jeffrey P Prestemon, John M Pye, and Thomas P Holmes. What is the price of catastrophic wildfire? In: *Journal of Forestry* 99.11 (2001), pp. 9 17.

[10] Chad M Shafer, Andrew E Mercer, Charles A Doswell III, Michael B Richman, and Lance M Leslie. Evaluation of WRF forecasts of tornadic and nontornadic outbreaks when initialized with synoptic-scale input . In: *Monthly Weather Review* 137.4 (2009), pp. 1250 1271.

[11] J King Foster, James O Kortge, and Michael J Wolanin. *Hybrid III-a biomechanically-based crash test dummy*. Tech. rep. SAE Technical Paper, 1977.

[12] John Michalakes and Manish Vachharajani. GPU acceleration of numerical weather prediction . In: *Parallel Processing Letters* 18.04 (2008), pp. 531 548.

[13] J Michalakes, J Dudhia, D Gill, T Henderson, J Klemp, W Skamarock, and W Wang. The weather research and forecast model: software architecture and performance . In: *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology*. Vol. 25. Reading, UK. 2004, p. 29.

[14] P. J. Vickery. *HAZUS-MH hurricane model methodology. I: hurricane hazard, terrain, and wind load modeling.*

[15] V. V. Titov. *Implementation and testing of the method of splitting tsunami (MOST) model*. 1997.

[16] P.D Bates and A.P.J De Roo. A simple raster-based model for flood inundation simulation . In: *Journal of Hydrology* 236.1 2 (2000), pp. 54 77. ISSN: 0022-1694. DOI: `10.1016/S0022-1694(00)00278-X`. URL: `http://www.sciencedirect.com/science/article/pii/S002216940000278X`.

[17] AMG Lopes, MG Cruz, and DX Viegas. FireStation an integrated software system for the numerical simulation of fire spread on complex topography . In: *Environmental Modelling & Software* 17.3 (2002), pp. 269 285.

[18] Mónica Denham, Kerstin Wendt, Germán Bianchini, Ana Cortés, and Tomàs Margalef. Dynamic data-driven genetic algorithm for forest fire spread prediction . In: *Journal of Computational Science* 3.5 (2012), pp. 398 404.

[19] Jan Mandel. *Efficient implementation of the ensemble Kalman filter*. University of Colorado at Denver and Health Sciences Center, Center for Computational Mathematics, 2006.

[20] M. A. Finney. *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.

[21] R C Rothermel. A mathematical model for predicting fire spread in wildland fuels . In: *Director* INT-115 (1972). Ed. by Usda Forest ServiceEditors, p. 40. URL: `http://www.srs.fs.usda.gov/pubs/32533`.

[22] Mark Arnold Finney et al. FARSITE: Fire area simulator: model development and evaluation . In: (2004).

[23] X. Pons. MiraMon . In: *Sistema de Información Geográfica y software de Teledetección, Centre de Recerca Ecològica i Aplicacions Forestals, CREAF. Bellaterra. ISBN* 84 (2000), pp. 931323 4.

[24] W.L. Gorr and K.S. Kurland. *GIS Tutorial: Workbook for ArcView 9*. GIS Tutorial. Esri Press, 2008. ISBN: 9781589482050. URL: `http://books.google.es/books?id=ZLGzr4li8U8C`.

[25] L.M. Surhone, M.T. Tennoe, and S.F. Henssonow. *Quantum GIS*. VDM Publishing, 2010. ISBN: 9786131406409. URL: `http://books.google.es/books?id=P_0kkgAACAAJ`.

[26] Frank Warmerdam. The geospatial data abstraction library . In: *Open Source Approaches in Spatial Data Handling*. Springer, 2008, pp. 87 104.

[27] Jason Forthofer, Kyle Shannon, and Bret Butler. 4.4 Simulating Diurnally Driven Slope Winds with WindNinja . In: *USDA Forest Service, Rocky Mountain Research Station, Missoula, MT* (2009).

[28] J Forthofer and B Butler. Windninja . In: *US Forest Service Joint Fire Science Program* (2013).

[29] Mónica Denham, Ana Cortés, Tomàs Margalef, and Emilio Luque. Applying a Dynamic Data Driven Genetic Algorithm to Improve Forest Fire Spread Prediction . In: *Computational Science – ICCS 2008*. Ed. by Marian Bubak, Geert van Albada, Jack Dongarra, and Peter Sloot. Vol. 5103. Lecture Notes in Computer Science. 10.1007/978-3-540-69389-5_6. Springer Berlin / Heidelberg, 2008, pp. 36 45. ISBN: 978-3-540-69388-8.

[30] Kerstin Wendt, Ana Cortés Fité, and Universitat Autònoma de Barcelona. Departament d Arquitectura de Computadors i Sistemes Operatius. Efficient knowledge retrieval to calibrate input variables in forest fire prediction . In: (2008).

[31] Kerstin Wendt, Ana Cortés, and Tomàs Margalef. Knowledge-guided Genetic Algorithm for input parameter optimisation in environmental modelling . In: *Procedia Computer Science* 1.1 (2010). ICCS 2010, pp. 1367 1375. ISSN: 1877-0509. DOI: `10.1016/j.procs.2010.04.152`. URL: `http://www.sciencedirect.com/science/article/pii/S1877050910001535`.

[32] Andrés Cencerrado, Ana Cortés, and Tomàs Margalef. Genetic Algorithm Characterization for the Quality Assessment of Forest Fire Spread Prediction . In: *Procedia Computer Science* 9 (2012). Proceedings of the International Conference on Computational Science, ICCS 2012, pp. 312 320. ISSN: 1877-0509. DOI: `10.1016/j.procs.2012.04.033`. URL: `http://www.sciencedirect.com/science/article/pii/S1877050912001548`.

[33] Susan L Graham, Peter B Kessler, and Marshall K McKusick. Gprof: A call graph execution profiler . In: *ACM SIGPLAN Notices* 39.4 (2004), pp. 49 57.

[34] Karl Furlinger and Michael Gerndt. ompP: A profiling tool for OpenMP . In: *OpenMP Shared Memory Parallel Programming*. Springer, 2008, pp. 15 23.

[35] Andrés Cencerrado, Ana Cortés, and Tom`as Margalef. On the way of applying urgent computing solutions to forest fire propagation prediction . In: *Procedia Computer Science* 9 (2012), pp. 1657 1666.

[36] Andrés Cencerrado Barraqué, Ana Cortés Fité, et al. Methodology for time response and quality assessment in natural hazards evolution prediction . In: (2014).

[37] ME Tryby, BY Mirghani, GK Mahinthakumar, and S Ranji Ranjithan. A solution framework for environmental characterization problems . In: *International Journal of High Performance Computing Applications* 24.3 (2010), pp. 265 283.

[38] Ross Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[39] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[40] Stephen R Garner et al. Weka: The waikato environment for knowledge analysis . In: *Proceedings of the New Zealand computer science research students conference*. Citeseer. 1995, pp. 57 64.

[41] Hal E Anderson. Aids to determining fuel models for estimating fire behavior . In: *The Bark Beetles, Fuels, and Fire Bibliography* (1982), p. 143.

[42] Jesús San-Miguel-Ayanz, P Barbosa, G Schmuck, G Libertà, and J Meyer-Roux. The european forest fire information system . In: *AGILE*. Vol. 2003. 2003, 6th.

[43] J. Forthofer, K. Shannon, and B. Butler. 4.4 Simulating Diurnally Driven Slope Winds with WindNinja . In: (2009).

[44] J. ForthoferA, K. ShannonA, and B. ButlerA. Initialization of high resolution surface wind simulations using NWS gridded data . In: *Notes* (2010).

[45] Brian Potter and Bret Butler. Using wind models to more effectively manage wildfire . In: *Fire management today* 62 (2009), pp. 40 46.

[46] Frederica Darema. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements . In: *Computational Science - ICCS 2004*. Ed. by Marian Bubak, Geert van Albada, Peter Sloot, and Jack Dongarra. Vol. 3038. Lecture Notes in Computer Science. 10.1007/978-3-540-24688-6_86. Springer Berlin / Heidelberg, 2004, pp. 662 669. ISBN: 978-3-540-22116-6.

[47] Frederica Darema. Grid computing and beyond: The context of dynamic data driven applications systems . In: *Proceedings of the IEEE* 93.3 (2005), pp. 692 697.

[48] E Grafarend. The optimal universal transverse Mercator projection . In: *Geodetic Theory Today*. Springer. 1995, pp. 51 51.

[49] B LOUIS Decker. *World geodetic system 1984*. Tech. rep. DTIC Document, 1986.

[50] Michael J Merrigan, Everett R Swift, Robert F Wong, and Joedy T Saffel. A refinement to the World Geodetic System 1984 reference frame . In: *Proceedings of the ION-GPS-2002*. 2002.

[51] Software.ecmwf.int. *GRIB API - GRIB-API - ECMWF Confluence Wiki*. 2015. URL: `https://software.ecmwf.int/wiki/display/GRIB/GRIB+API`.

[52] Jesús San-Miguel-Ayanz, Ernst Schulte, Guido Schmuck, Andrea Camia, Peter Strobl, Giorgio Liberta, Cristiano Giovando, Roberto Boca, Fernando Sedano, Pieter Kempeneers, et al. Comprehensive Monitoring of Wildfires in E urope: The European Forest Fire Information System (EFFIS) . In: (2012).

[53] CO Justice, Louis Giglio, S Korontzi, J Owens, JT Morisette, D Roy, J Descloitres, S Alleaume, F Petitcolin, and Y Kaufman. The MODIS fire products . In: *Remote Sensing of Environment* 83.1 (2002), pp. 244 262.

[54] NASA FIRMS. MODIS hotspot/active fire detections . In: *Book Modis Hotspot/Active Fire Detections* (2012).

[55] N Akkiraju, H Edelsbrunner, M Facello, P Fu, EP Mucke, and C Varela. Alpha shapes: definition and software . In: *Proceedings of the 1st International Computational Geometry Software Workshop*. 1995, pp. 63 66.

[56] Stijn Hantson, Marc Padilla, Dante Corti, and Emilio Chuvieco. Strengths and weaknesses of MODIS hotspots to characterize global fire occurrence . In: *Remote Sensing of Environment* 131 (2013), pp. 152 159.

[57] Wikipedia. *Delaunay triangulation — Wikipedia, The Free Encyclopedia*. [Online; accessed 15-June-2015]. 2015. URL: `%5Curl%7Bhttps://en.wikipedia.org/w/index.php?title=Delaunay_triangulation&oldid=662807396%7D`.

[58] Sébastien Loriot Naceur Meskini and Sylvain Pion. Python, Scilab and Ipe interfaces to CGAL . In: ().

[59] Jesús San-Miguel-Ayanz, Jose MC Pereira, Roberto Boca, Peter Strobl, Jan Kucera, and Anssi Pekkarinen. Forest fires in the European Mediterranean region: mapping and analysis of burned areas . In: *Earth observation of wildland fires in Mediterranean ecosystems*. Springer, 2009, pp. 189 203.

[60] Thomas Jung, MJ Miller, TN Palmer, P Towers, N Wedi, D Achuthavarier, JM Adams, EL Altshuler, BA Cash, JL Kinter Iii, et al. High-resolution global climate simulations with the ECMWF model in Project Athena: Experimental design, model climate, and seasonal forecast skill . In: *Journal of Climate* 25.9 (2012), pp. 3155 3172.

[61] NASA. *NASA FIRMS NRT MODIS Near real-time Hotspot / Active Fire Detections MCD14DL data set. Available on-line.* `https://worldview.earthdata.nasa.gov/`. [Online; accessed 13-April-2015]. 2015.