



Universitat Autònoma de Barcelona

Escola Tècnica Superior d'Enginyeria
Departament d'Arquitectura de Computadors
i Sistemes Operatius

**Predicción de la Evolución
de los Incendios Forestales
Guiada Dinámicamente por los Datos**

Tesis doctoral presentada por **Mónica Malen Denham**, para optar al grado de **Doctor** por la Universitat Autònoma de Barcelona, bajo la dirección de la Dra. Ana Cortés Fité.

Barcelona, Julio de 2009

Predicción de la Evolución de los Incendios Forestales Guiada Dinámicamente por los Datos

Tesis doctoral presentada por Mónica Malen Denham, para optar al grado de Doctora por la Universitat Autònoma de Barcelona. Trabajo realizado en el Departamento de Arquitectura de Computadores y Sistemas Operativos de la Escuela Técnica Superior de Ingeniería de la Universidad Autónoma de Barcelona, dentro del programa de Doctorado en Informática, opción A: “Arquitectura de Computadores y Procesamiento Paralelo” bajo la dirección de la Dra. Ana Cortés Fité.

Barcelona, julio 2009

Director:

Dra. Ana Cortés Fité

El mundo

Un hombre del pueblo de Neguá, en la costa de Colombia, pudo subir al alto cielo.

A la vuelta, contó. Dijo que había contemplado, desde allá arriba, la vida humana. Y dijo que somos un mar de fueguitos.

—El mundo es eso —reveló—. Un montón de gente, un mar de fueguitos.

Cada persona brilla con luz propia entre todas las demás.

No hay dos fuegos iguales. Hay fuegos grandes y fuegos chicos y fuegos de todos los colores. Hay gente de fuego sereno, que ni se entera del viento, y gente de fuego loco, que llena el aire de chispas. Algunos fuegos, fuegos bobos, no alumbran ni queman; pero otros arden la vida con tantas ganas que no se puede mirarlos sin parpadear, y quien se acerca, se enciende.

Eduardo Galeano
(El Libro de los Abrazos)

Agradecimientos

Siempre serán pocas las palabras para expresar el agradecimiento a los que brillan, arden y viven la vida con ganas, haciendo con una mirada, con una sonrisa, lo que miles de palabras pueden.

A mi familia, gracias por todo el amor, toda la alegría y todo el apoyo. Gracias a mis viejos, a mis hermanas y a mis sobrinos hermosos que son una fuente de alegría constante. A toda mi familia, este trabajo es dedicado a todos ellos.

Gracias a los personajes argentinos que siempre estuvieron ahí, Agus, Arito y Astor Piazzolla.

A todos los personajes que conocí acá, especialmente a Kury, Gore, Lali y Nico, gracias por tantos aprendizajes, emociones y momentos.

A todos mis amigos, gracias por todo el apoyo y cada momento compartido.

A Maru, gracias por tantas cosas.

Gracias a todo el Departamento, y en especial a Ana Cortés y a Tomás Margalef, por toda la guía y el cuidado. Y al resto de pirómanos: Germán, Darío, Kerstin, Andrés, gracias por los momentos de terapia y toda la ayuda. Los recordaré siempre con mucho cariño.

GRACIAS a todos, han hecho que estos años sean una experiencia hermosa.

Índice general

1. Introducción	1
1.1. Visión General	1
1.2. Lucha Contra los Incendios Forestales	3
1.3. Características de los Incendios Forestales	5
1.3.1. Tipos de Incendios Forestales	8
1.4. Ciencia Computacional	9
1.5. <i>Dynamic Data Driven Application Systems</i>	12
1.6. Contribución de la Tesis	14
1.7. Organización del Trabajo	14
2. Simulación de Incendios Forestales: Modelos, Técnicas y Herramientas	17
2.1. Características Generales	17
2.2. Modelos de Comportamiento del Fuego	22
2.2.1. Modelos Físicos	22
2.2.2. Modelos Físicos-Estadísticos	23
2.2.3. Modelos Estadísticos	23
2.2.4. Modelos Probabilísticos	23
2.2.5. Caso de Estudio: Modelo de Rothermel	23
2.3. Técnicas de Simulación	26
2.3.1. <i>Bond Percolation</i>	26
2.3.2. Propagación elíptica	27
2.3.3. Autómatas Celulares	29
2.4. Simuladores del Comportamiento del Fuego	36
2.4.1. BehavePlus	36
2.4.2. <i>Clarke Cellular Automaton System</i>	38
2.4.3. DYNAFIRE	38
2.4.4. EMBYR	39
2.4.5. FIREMAP	39
2.4.6. WILDFIRE	40

2.4.7.	NEXUS	40
2.4.8.	FARSITE	41
2.4.9.	FOFEM	42
2.4.10.	FireStation	42
2.4.11.	<i>fireLib</i>	44
2.4.12.	Resumen	47
3.	Predicción de la Evolución de los Incendios Forestales Guiada por los Datos	51
3.1.	Predicción Clásica	52
3.2.	Fuentes de Error de la Predicción Clásica	53
3.3.	Predicción en Dos Etapas	55
3.3.1.	Etapas de Ajuste	56
3.3.2.	Etapas de Predicción	58
3.4.	Fuentes de Error de la Predicción en Dos Etapas	59
4.	Algoritmo Genético Guiado Dinámicamente por los Datos	61
4.1.	Algoritmos Genéticos	62
4.1.1.	Operadores Genéticos	64
4.1.2.	Función de <i>Fitness</i>	70
4.2.	Algoritmo Genético Paralelo	72
4.3.	Etapas de Ajuste Guiada por los Datos	80
4.4.	Obtención del Conocimiento	81
4.5.	Determinación de la Dirección e Intensidad de Máxima Propagación del Fuego.	87
4.6.	Utilización del Conocimiento para Sintonizar los Valores de los Parámetros	90
4.6.1.	Método Computacional	92
4.6.2.	Método Analítico	97
4.7.	Inyección de datos	102
4.7.1.	Método Computacional	103
4.7.2.	Método Analítico	103
5.	Resultados Experimentales	107
5.1.	Introducción	107
5.2.	Características de la Implementación	107
5.3.	Predicción Clásica vs Predicción en Dos Etapas	110
5.4.	Caso de estudio 1	112
5.5.	Caso de estudio 2	116
5.5.1.	Análisis de convergencia	121
5.6.	Caso de estudio 3	122
5.7.	Caso de estudio 4	125

6. Conclusiones y Líneas Abiertas	129
6.1. Conclusiones y Observaciones	129
6.2. Líneas Abiertas	136

Apéndices

A. Fórmulas Trigonómicas del Modelo del Fuego	139
A.1. Obtención de la Dirección Donde Ocurre la Máxima Velocidad de Propagación e Intensidad del Fuego	139
B. Obtención del Ángulo y Velocidad del Viento	143

Capítulo 1

Introducción

1.1. Visión General

Los bosques y áreas forestadas son una fuente necesaria para mantener la vida en nuestro planeta. Árboles y plantas juegan un papel muy importante para todo tipo de expresión de vida alrededor del planeta y su calidad.

Los bosques son el medio ambiente natural para miles de especies diferentes de animales, plantas y árboles. Cada vez menos, pero también son el medio ambiente donde aún habitan grupos de personas, las cuales, generación tras generación, han vivido con la naturaleza, respetándola, amándola y viviendo de una forma sana e inteligente.

Actualmente, muchas de estas áreas son consideradas como recursos económicos en todo el mundo. Existen áreas consideradas “reservas verdes” dada la importancia de los bosques en el mantenimiento de la calidad del aire. Por otro lado, la madera extraída de los árboles es muy usada en diversos campos: producción de calor (gracias a su combustión), construcción, producción de papel, manufacturación de muebles, etc.

Los incendios forestales son parte del balance ecológico natural del planeta. Este tipo de incendio permite naturalmente la regeneración de árboles y plantas, esto es, viejas poblaciones de árboles y plantas son destruidas por el fuego, dejando el suelo en condiciones favorables para el nacimiento de nueva y tal vez productiva vegetación.

La combinación de diversos factores provocan el comienzo de fuegos y la devastación de quizás bosques enteros como así también de áreas forestadas. Temporadas secas, combustibles secos, tormentas eléctricas, volcanes en erupción, terremotos, son distintos factores que pueden provocar incendios forestales.

Mientras la cantidad de este tipo de incendios se mantenga naturalmen-

te bajo, o dentro de los rangos naturales, éstos pueden ser una oportunidad para nueva vida, manteniendo la flora y la fauna dentro del contexto natural.

Pero lamentablemente durante los últimos años el número de incendios forestales se ha incrementado de una forma alarmante. El gran número de este tipo de incendios rompe el balance que naturalmente ellos significan.

En [20] se muestra la gravedad del año 2007 con respecto a este tipo de incendios:

“ ...Apenas comenzada la temporada de incendios forestales, los datos provisionales indican que en 2007 ya han ardido 3376 km^2 , frente a un total de 3585 km^2 en 2006, siendo julio de 2007 uno de los peores meses que se recuerdan en este aspecto. Los datos proceden del Sistema Europeo de Información sobre Incendios Forestales (EFFIS en su sigla en inglés), que gestiona la Comisión y facilita alertas rápidas de riesgo de incendio y evaluaciones de daños.

Tras las alertas de elevado riesgo de incendio forestal que se emitieron a finales de junio en países como Grecia y Chipre, en la segunda quincena de julio se produjo un espectacular incremento de incendios y zonas quemadas en Bulgaria, Croacia, Grecia e Italia. Por medio de imágenes por satélite se situaron todos los incendios que afectaban a una zona de más de 50 hectáreas sólo en estos cuatro países y dejaban un total de 2229 km^2 de tierras calcinadas. Todavía deben incorporarse en el EFFIS las cifras correspondientes a otros países afectados por grandes incendios, como Turquía y Albania...”

Por otro lado, cada año entre el año 2000 y el año 2005, en África se han perdido más de 4 millones de hectáreas, debido a fuegos y a la conversión de áreas forestadas a tierras de cultivo.

Durante octubre del 2007, California fue escenario de 23 incendios forestales. Un millón de habitantes fueron evacuados, 1500 casas fueron destruidas, mientras que 56000 estuvieron bajo amenaza del fuego. El fuego provocó 17 muertes y 16 bomberos han salido lastimados tratando de combatirlo [44]. La figura 1.1 muestra los incendios ocurridos a nivel mundial en el período 30-07-07 al 08-08-07 [32].

Estos son sólo algunos ejemplos que nos permiten apreciar que el número de incendios forestales ha crecido de forma preocupante en los últimos años. Además, muchos de estos incendios son provocados por personas, de-

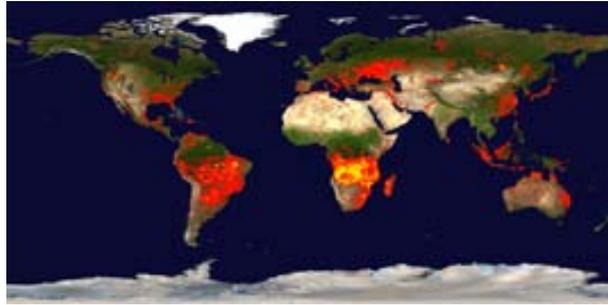


Figura 1.1: Fuego durante los primeros días de agosto del 2007.

bido a descuidos (fogatas mal apagadas, colillas de cigarrillos, botellas de vidrio, etc.), negligencia (incendios provocados adrede, quemas prescritas que crecen y se convierten en fuegos incontrolables, venganzas, etc.), quemas para convertir un área forestada en áreas aptas para cultivo, etc., etc. Este último caso, es realmente desalentador, pues cuando las tierras se condenan al cultivo masivo, ya sea monocultivo como así también pluricultivo, se suele arruinar las tierras, sacándoles todos los nutrientes y además de esto, envenándola con agroquímicos, pesticidas, etc., logrando que tierras que antes eran fuente de vida se conviertan en pocos años en desiertos, dejando sólo pobreza a sus alrededores [24].

Algunas consecuencias directas de estos incendios y de la pérdida de áreas forestadas, son la muerte de distintas especies de animales y de vegetales, pérdida de tierras ricas en nutrientes y minerales, contaminación del aire, inundaciones provocadas por la erosión de la tierra, contaminación de cursos de agua cercanos como así también de napas subterráneas, muertes de animales y seres humanos, etc.

A su vez, algunas consecuencias indirectas de todos estos problemas son la propagación de enfermedades y pestes, hambre, extinción de especies de animales y de vegetales, etc. Todas estas consecuencias, directas e indirectas amenazan la calidad de vida sobre el planeta.

1.2. Lucha Contra los Incendios Forestales

Actualmente existe mucho esfuerzo por parte del hombre en la lucha contra este tipo de desastres. Como resultados de este esfuerzo existen distintos estudios, planes, estrategias, herramientas, etc., usadas para prevención de incendios, definición de áreas de riesgo, para reducir los efectos del fuego, mejorar la seguridad de las personas implicadas, etc.

Principalmente, todo este trabajo se puede dividir en:

- **Prevención:** como en muchas otras áreas, la información es una de las medidas preventivas más eficientes. Mantener a la población informada y con conocimientos sobre riesgos de incendios en un área determinada, definir y dar a conocer planes de prevención, informar a la gente sobre qué hacer ante una alarma de fuego, ante un incendio (de pequeña o gran magnitud), definir planes de evacuación, etc.
- **Extinción del fuego:** mejorar la efectividad de la lucha contra el fuego, reducir los daños causados por el fuego, mejorar la toma de decisiones durante este tipo de desastres, etc.
- **Análisis de riesgos:** elaboración de mapas de riesgo basados en factores humanos, condiciones meteorológicas, características topográficas, datos históricos, etc. Evaluación y determinación de zonas de riesgo por temporadas.

Llevar a cabo estas tareas no es un trabajo fácil, pues se está trabajando con factores meteorológicos, topográficos, humanos (comportamiento de poblaciones), comportamiento del fuego, etc. Todos estos elementos tienen su propio patrón de comportamiento, y además, existen interacciones entre ellos, los cuales afectan dichos comportamientos de forma significativa, resultando un sistema de comportamiento heterogéneo y complejo.

A través de los años se han ido creando y perfeccionando un amplio conjunto de herramientas que ayudan al hombre en la lucha contra el fuego. El correcto uso y efectividad de estas herramientas son críticos para reducir la propagación del fuego, reducir los daños ocasionados, evitar pérdidas materiales, económicas, reducir el riesgo para las personas afectadas por el fuego (pobladores de la zona, bomberos), etc.

Afortunadamente, en la actualidad existen diversas herramientas informáticas usadas en este campo de trabajo. Tal vez las herramientas informáticas más utilizadas son los simuladores del comportamiento del fuego. Existen diversos simuladores [6] [9] [21] [22], los cuales pueden variar en el modelo de fuego implementado, tipo y cantidad de entradas, formato y cantidad de salidas, nivel de conocimiento por parte del usuario, tipo de fuego que simulan, etc. Un factor común de estos simuladores es que necesitan la descripción del entorno en donde se desarrolla el fuego, entonces, a partir de ese entorno, se obtiene la simulación de cómo se desarrollaría el fuego bajo dichas condiciones específicas (condiciones climatológicas, topográficas, vegetación, etc.)

Con estos simuladores, se puede obtener cómo se propagaría el fuego bajo determinadas condiciones del entorno, en un espacio y tiempo específicos.

Disponer de cómo avanzaría el fuego antes de que esta propagación realmente suceda, es útil para los puntos descritos anteriormente:

- **Prevención:** conocer cómo se prepagaría el fuego en un determinado sitio, es útil para definir planes de evacuación, definir el rol de las personas involucradas en el lugar del desastre (pobladores, bomberos, etc.), definir dónde disponer de cortafuegos, etc.
- **Extinción del fuego:** conocer cómo se propagará el fuego en un incendio permite mejorar la eficacia de las decisiones y acciones tomadas para reducir el fuego y las consecuencias de los incendios.
- **Análisis de riesgos:** las simulaciones se utilizan para definir mapas de riesgo y riesgos de ignición. Esto es muy útil en situaciones de riesgo como pueden ser largas temporadas secas en climas cálidos y calientes.

Por otro lado, una predicción del avance del fuego es realmente útil si:

- la predicción está disponible antes de que el fuego real alcance la zona que se está prediciendo. Además,
- la predicción realmente describe el comportamiento del fuego. Esto significa, necesariamente la predicción debe tener buena calidad, se debe aproximar a la realidad, toda predicción necesariamente debe ser confiable.

En este trabajo tenemos en cuenta estos dos aspectos, los cuales han determinado ciertas características de la aplicación propuesta.

1.3. Características de los Incendios Forestales

El fuego es una reacción química, producida por la ignición y combustión de materiales. Tres elementos son necesarios para que dicha ignición y combustión sucedan: combustible para arder, aire para la contribución de oxígeno y una fuente de calor para elevar el combustible a la temperatura de ignición (figura 1.2). Reduciendo uno de estos tres elementos, el fuego puede ser extinguido [11].

Luego, para que un fuego tenga continuidad, es necesario agregar la reacción en cadena (figura 1.3, [43]): reacción mediante la cual la combustión se mantiene sin necesidad de mantener la fuente principal de ignición. Sin esta última sólo se tiene el fenómeno llamado incandescencia.

El fuego comienza en un punto específico y se propaga por el terreno debido a tres fenómenos (figura 1.4, [49]):

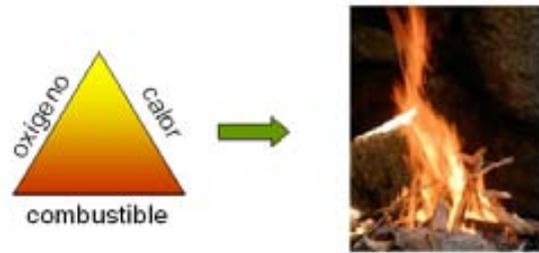


Figura 1.2: Elementos del “triángulo del fuego”.

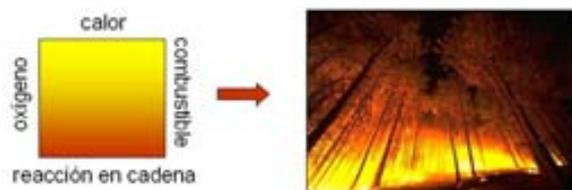


Figura 1.3: Elementos del “cuadrado del fuego”.

- **Conducción:** es la única forma de transferencia de calor de los sólidos. Es la transmisión del calor desde una región con mayor temperatura a otra región con menor temperatura, aumentándola en esta última (se cree que es debido al movimiento de electrones libres que transportan energía cuando existe una diferencia de temperaturas).
- **Convección:** movimiento de un fluido debido exclusivamente a la no uniformidad de la temperatura de dicho fluido: el fluido más caliente y menos denso asciende, mientras que el menos denso y más frío desciende. Este movimiento transfiere calor de una parte del fluido a otra parte del mismo.
- **Radiación:** a diferencia de la conducción y convección, la radiación es el intercambio de calor entre sustancias que no tienen que estar necesariamente en contacto, sino que pueden estar separadas por un vacío. Es la energía emitida por un cuerpo, cuanto mayor es la temperatura de la sustancia, mayor es la energía emitida.

El fuego se produce en 2 fases: precalentamiento y combustión: los materiales son precalentados mediante la transmisión del calor, y luego, cuando alcanzan la temperatura de ignición, la combustión de dicho material ocu-

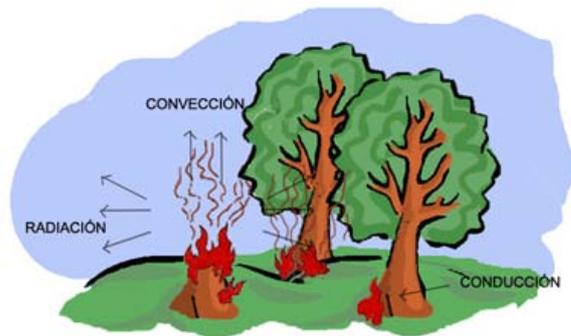


Figura 1.4: Formas en que el calor se propaga.

rre, no siendo necesaria otra fuente de calor.

Aunque estas características se mantienen, cuando hablamos de incendios forestales el comportamiento físico del fuego depende también de otros factores. Algunos ejemplos son: la topografía (relieve, pendiente, exposición, altura), la vegetación (ubicación, forma y tamaño, compactación, continuidad, carga, composición química, contenido de humedad, densidad) y la meteorología (temperatura, humedad relativa, viento, precipitaciones, nubes) [43]. Estas características, suelen ser la entrada en muchos de los simuladores de incendios forestales actuales.

En la actualidad, muchos de los modelos matemáticos del fuego (por



Figura 1.5: Triángulo del fuego en incendios forestales.

ejemplo, el definido por Rothermel [38]), asumen un crecimiento del fuego (*ROS: Rate of Spread*) dependientes sólo de las condiciones externas (topográficas y meteorológicas, figura 1.5). Esto es, a condiciones externas limitadas y uniformes en el tiempo, el *ROS* se mantendrá también uniforme en el tiempo.

Sin embargo, en trabajos tales como [14] [43] y [42] se muestra que el

histórico del fuego, es decir, todo el crecimiento del fuego, influirá en su futura propagación. Esto se debe principalmente a la convección y radiación de calor que produce el fuego mismo, modificando las propiedades del combustible potencial (un ejemplo muy fácil de ver es el aceleramiento en la pérdida de humedad de los combustibles no quemados), de las características de las llamas, etc.

De esta forma, se considera un elemento más en el triángulo mostrado en la figura 1.5. En estos trabajos se muestra que el *ROS* del frente del fuego depende de cómo ha crecido el incendio, por lo tanto, el *ROS* también depende del tiempo (figura 1.6).



Figura 1.6: Cuadrado del fuego en incendios forestales, donde se incorpora el tiempo como elemento.

1.3.1. Tipos de Incendios Forestales

Dependiendo de dónde ocurre el fuego, los incendios forestales se pueden definir como:

- Incendios de superficie: ocurren al ras del suelo, quemando combustibles contiguos en la superficie del terreno, quemando hierba, pasto, matorrales, arbustos y demás vegetación menor. El fuego avanza rápidamente.
- Incendios de copas: fuegos que se propagan en ramas altas y en las copas de los árboles. El fuego puede alcanzar velocidades muy altas, especialmente en días de viento [46].
- *Spotting fires*: este tipo de fuego sucede cuando hojas y ramas ardientes son arrastradas por el viento o por “explosiones” del combustible (propias del proceso de combustión) y comienzan focos de fuegos distantes. De esta manera, el fuego puede saltar carreteras, ríos, incluso pueden pasar cortafuegos.

- Fuegos subterráneos: ocurren bajo la superficie del suelo, quemando raíces y materia orgánica.

A pesar de que dos o más tipos de incendios pueden ocurrir simultáneamente, cada uno de ellos tiene su propio patrón de comportamiento, pues factores influyentes tales como el tipo de combustible, presencia de viento, pendiente, cantidad de oxígeno, humedad, etc. varían significativamente dependiendo de su localización: bajo la tierra, superficie del terreno o a altura, en las copas de árboles.

Típicamente, los simuladores tratan con sólo uno de estos tipos de incendios. La complejidad del comportamiento físico del fuego y todos los factores que interactúan con él, más las limitaciones en la modelización e implementación que significan las actuales herramientas de modelado y herramientas informáticas, hace conveniente la implementación de un sólo tipo de propagación a la vez.

Richard Rothermel, plantea en [39] la siguiente pregunta: “*Puede un incendio forestal realmente ser predecido?*” y la respuesta que se da a continuación es la siguiente: “*Esto depende de la precisión pretendida en la respuesta*”.

El estudio de este problema nos permite ver que esta respuesta es acertada, pues dada la complejidad del problema (comportamiento del fuego), la cantidad de elementos que actúan (topografía, meteorología, vegetación, tiempo, llamas de fuego), sus interrelaciones e influencias, el microclima generado por el fuego y sus consecuencias, etc., etc., resultan en un entorno de trabajo muy complejo, donde es necesario un importante compromiso entre calidad de predicción y tiempo de respuesta.

1.4. Ciencia Computacional

Durante todos los tiempos el ser humano ha buscado encontrar explicaciones a distintos problemas, fenómenos (naturales como así también artificiales), y constantemente se ha topado con las limitaciones tanto del propio conocimiento y capacidad de comprensión del ser humano como así también las limitaciones tecnológicas, religiosas, éticas, morales, etc. Dependiendo del momento y del lugar, el ser humano ha buscado las respuestas a muchas de sus inquietudes en la naturaleza, la religión, la mitología, el misticismo, etc.

En los últimos tiempos, y principalmente en el mundo occidental, el método científico es el que ha prevalecido y el que actualmente se está utilizando. Partiendo de observaciones, conocimientos previos (premisas asumidas como “verdaderas”, las bases del conocimiento), experimentación,

formulación de tesis y demostraciones para estas tesis, son las herramientas utilizadas desde la concepción del método científico hasta nuestros días (desde Aristóteles, cambiando a través del tiempo gracias a influencias religiosas, éticas, etc.). Pero en los últimos siglos, la ciencia a dado pasos realmente importantes y cada vez más usuales gracias al continuo y veloz avance tecnológico.

En diversas áreas de estudio tales como la astronomía, medicina, física, matemáticas, etc., la tecnología ha sido un aspecto fundamental en el avance de dichas ciencias. Por ejemplo, el uso de microscopios en la medicina moderna ha permitido pasar de estudiar los órganos, sus funciones e interrelaciones al estudio de los tejidos, las células, sus estructuras internas, componentes, etc. Desde la física antigua, se ha intentado encontrar las “partes fundamentales” que componen la materia, se logró observar que las partículas están compuestas de átomos, los átomos tienen sus núcleos y electrones, los nucleos atómicos sus protones y neutrones y así se seguirán hallando más componentes. Estos son sólo ejemplos, pero demuestran que los avances en la ciencia van de la mano del avance en la tecnología utilizada en cada una de estas áreas.

En el marco tecnológico, el avance de la informática es un gran paso que ha marcado y modificado a todo tipo de emprendimiento. En este momento un computador (o varios) es una herramienta imprescindible en todo tipo de área de trabajo, científico, comercial, empresarial, industrial, etc.

En la actualidad se puede observar que a través del uso de computadoras se dan soluciones a distintos tipos de problemas. Muchos de ellos son el resultado de la implementación de algún modelo de algún fenómeno (natural o artificial). Pero, para lograr esto, se debe contar con conocimiento en el dominio del problema, luego, se necesita un importante conocimiento para definir un modelo que lo represente de forma correcta y completa, y por último, se necesita de la tecnología adecuada para poder implementar el modelo para su posterior uso.

Como se puede observar, este trabajo es naturalmente interdisciplinario, existe una fuerte interacción entre las distintas partes para lograr solucionar problemas de forma eficaz y correcta utilizando sistemas informáticos. La Ciencia Computacional es la ciencia que interrelaciona estos aspectos: usa conocimiento y herramientas matemáticas y tecnológicas para diseñar herramientas que serán usadas en una computadora para solucionar problemas científicos [10] [18]. La figura 1.7 muestra la interacción entre las distintas áreas implicadas.

Frente a un problema, es imprescindible el estudio del dominio de dicho problema para su correcta comprensión, valoración y definición, su correcto análisis, diseño del modelo que represente lo que queramos solucionar del

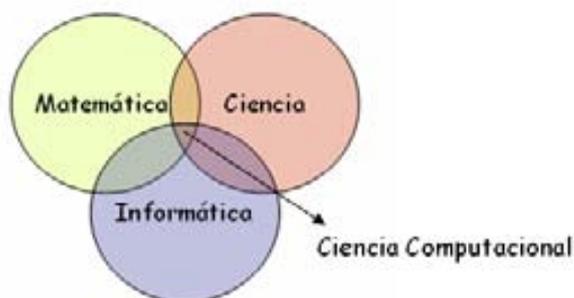


Figura 1.7: Interacción entre los distintos campos de ciencias.

problema (todo o partes), verificación de correctitud, completitud, viabilidad de dicho modelo (aquí son necesarias herramientas informáticas de diseño, visualización, etc.). Una vez definido el modelo que represente el problema, y se verifique su utilidad, se implementa utilizando alguna herramienta informática, se definen entradas, salidas (y sus formatos), recursos necesarios para su uso, pre y post condiciones, formas de visualizar entradas y salidas, etc., siendo estos, sólo algunos de los pasos de estos procesos. Aquí se hace visible la necesidad de la interacción entre expertos de las distintas áreas, y cómo todas las personas involucradas deben, idealmente, adquirir conocimientos del resto de las mismas. Esto no siempre es posible, pueden existir diversas limitaciones que afectan en el resultado final de un sistema informático.

Teniendo en cuenta los problemas con los que en los últimos años el ser humano se está enfrentando, es imprescindible el buen trabajo en la Ciencia Computacional. Para ver sólo algunos ejemplos de esto, es suficiente observar el sitio *Top 500* [45], donde se listan los supercomputadores más rápidos del momento (se actualiza cada 6 meses), y en la descripción de muchos de ellos, se puede ver que han sido creados y diseñados para solucionar algún problema específico (simular fenómenos naturales, análisis de secuencias de adn, síntesis de proteínas, etc.). Teniendo en cuenta la magnitud, el dinero, los recursos utilizados en estos supercomputadores, se verifica la importancia de las interacciones y del trabajo en conjunto que debe existir entre las áreas que involucra la Ciencia Computacional.

1.5. *Dynamic Data Driven Application Systems*

El paradigma *Dynamic Data Driven Application Systems (DDDAS)* fue creado por la doctora Frederica Darema. Las principales características de este paradigma son claramente descritas en el trabajo [16], las cuales se resumirán en los párrafos siguientes.

La idea que dio origen a este nuevo paradigma ha surgido en los años 80. En dicho momento, la motivación principal que deriva en este nuevo paradigma fue el de acelerar el procesamiento de las aplicaciones usando datos adicionales en lugares específicos durante el cómputo, mejorar la precisión de muestras estadísticas y acelerar la convergencia a regiones de interés, e inversamente, controlar los procesos de medición.

Estas motivaciones definen las principales características de este paradigma: sistemas con la habilidad de incorporar dinámicamente datos adicionales en tiempo de ejecución, y también, habilidad para guiar dinámicamente los procesos de medición.

En la actualidad, el análisis y la predicción precisa del comportamiento de un sistema complejo es difícil. Existen aplicaciones y sistemas de simulación pero que todavía no describen de forma correcta dichos tipos de sistemas. La elaboración de modelos complejos producen simulaciones que divergen o fallan al predecir el comportamiento real del sistema evaluado. Esto es acentuado en procesos con límites de tiempo real.

Aplicaciones que dinámicamente incorporen nuevos datos (almacenados o medidas *on line*) prometen análisis más precisos, predicciones de mayor calidad, controles más exactos y salidas más confiables. Estas son características consideradas por este paradigma.

Según palabras de la autora, este paradigma puede ser visto como una metodología para superar la falta de completitud en los modelos, y la capacidad de mejorar dichos modelos utilizando información adicional (en tiempo de ejecución, la información adicional puede ser utilizada para mejorar el modelo, seleccionando sus partes, refinándolo, etc.)

Adicionalmente, el paradigma incentiva la habilidad de una aplicación de controlar y guiar el proceso de medición, haciéndolo de esta forma más efectivo. Esta capacidad es conveniente en casos donde las medidas son difíciles de realizar, costosas o críticas en tiempo. La habilidad de guiar el proceso de medición y enfocar selectivamente en un subconjunto del espacio de medición puede mejorar la efectividad y eficiencia, llegando a reducir costos, tiempo de recolección o mejorar la calidad de los datos recolectados.

En la década de los años 80 la tecnología requerida para estos tipos de propósitos, era inadecuada o directamente inexistente, recién en los años

90 este paradigma se vuelve potencialmente viable. Aunque en un principio dicho paradigma nace para aplicaciones científicas y de ingenirías, experiencias posteriores mostraron ser adecuado para procesos de administración, industriales, etc.

Para satisfacer los nuevos requerimientos que este paradigma implica, un trabajo interdisciplinario entre investigadores y desarrolladores debe ser llevado a cabo en 4 áreas distintas [16]:

- **Aplicaciones:** las aplicaciones requieren que las simulaciones acepten datos en tiempo de ejecución y ser dinámicamente guiadas por estos datos. Para manejar la inyección de datos, se necesitan modelos que describan el sistema en diferentes niveles de detalle, y la habilidad de invocar dinámicamente esos modelos en función a los datos de entrada.
- **Algoritmos matemáticos:** se requieren avances en algoritmos matemáticos existentes o el desarrollo de nuevos algoritmos matemáticos, que sean estables y tengan propiedades de convergencia robusta bajo entrada dinámica de datos.
- **Software:** tecnología e infraestructura necesaria para soportar el cómputo dinámico, comunicación y requerimiento de datos de las aplicaciones *DDDAS*, interfaces con sistemas de medición, tolerancia a fallos y calidad de servicio para asegurar flujo de datos y disponibilidad de recursos. Este tipo de sistemas emplearán ambientes con plataformas heterogéneas, como sensores empotrados (recolección de datos), ambientes de simulación de alto rendimiento distribuido y plataformas para pre y post procesamiento de datos y visualización.
- **Mediciones:** es necesario el desarrollo de interfaces a dispositivos físicos e interfaces de aplicación/medición junto con capacidad para manejar los sistemas de medición, sensores y actores que son recursos adicionales. La combinación de aplicaciones y mediciones requerirán nuevas aproximaciones para sistemas de manejo de datos que soporten diferentes esquemas de nombres y ontologías (esquema conceptual), servicios de información, visualizaciones, etc.

En la actualidad existen diversas aplicaciones que se están desarrollando dentro del marco de este paradigma. Existen algunos ejemplos que nos alientan a desarrollar nuestro trabajo dentro del marco que el paradigma *DDDAS* nos ofrece, como por ejemplo, los trabajos desarrollados en [15], [30], [31], [4].

1.6. Contribución de la Tesis

En trabajos previos ([10], [1]) se ha desarrollado un método de predicción en dos etapas que mejora la calidad de las predicciones clásicas del avance del fuego. Estos métodos trabajan con la hipótesis de que las características ambientales presentes en el momento de un incendio, se mantendrán constantes (al menos similares) durante un intervalo de tiempo suficientemente largo para permitir mejorar la calidad de las predicciones.

Teniendo en cuenta las características del problema que se trata de resolver, se propone el desarrollo de una aplicación dentro del paradigma *DDAS* ([16], [17]) que tiene como principales objetivos:

- Mejorar la calidad de las predicciones de la propagación de los incendios forestales.
- Reducir el tiempo de predicción, esto puede ser: acelerar el tiempo de respuesta, como así también, en la misma cantidad de tiempo, aumentar la calidad de predicción.
- Reducir el impacto de la hipótesis de partida (características de las condiciones medioambientales) mediante la inyección dinámica de conocimiento.

Teniendo en cuenta estos objetivos, se ha desarrollado una aplicación para la predicción de la evolución del fuego la cual será presentada y explicada en los próximos capítulos.

1.7. Organización del Trabajo

En el segundo capítulo de esta Tesis, los distintos modelos de simulación de incendios forestales serán descritos, como así también las técnicas de simulación y se ejemplificará mostrando resumidamente las principales características de los simuladores actuales. Este capítulo muestra el estado del arte actual de esta problemática.

El tercer capítulo, muestra la predicción clásica de incendios forestales, presentando sus ventajas y sus problemas. A continuación, se presenta la predicción en dos etapas, método que supera los problemas encontrados en la predicción clásica. Pero este método de predicción en dos etapas también presenta algunos problemas, también presentados en este capítulo.

Durante el capítulo 4 se muestran las propuestas para solucionar los problemas de la predicción en dos etapas. Estas mejoras son la contribución principal de este trabajo, y son ampliamente desarrollados durante este capítulo.

En el capítulo 5, se muestra la experimentación más representativa de este trabajo, mostrándose las características de cada experimentación, los resultados obtenidos y un análisis de estos resultados.

Por último, en el capítulo 6 se exponen las conclusiones que el desarrollo de este trabajo permitió llegar, haciéndose hincapié en los objetivos planteados. En dicho capítulo se describen también las principales líneas abiertas de trabajos futuros.

Capítulo 2

Simulación de Incendios Forestales: Modelos, Técnicas y Herramientas

El objetivo de este capítulo es describir distintos modelos que han sido desarrollados para representar la propagación del fuego. Estos modelos difieren para cada tipo de propagación (superficie, copa, *spotting*, etc) ya que las condiciones donde se desarrolla el fuego son muy distintas, por lo tanto, su comportamiento resulta también muy diverso.

Además, se describirán las principales características de varios de los simuladores actuales, con el objetivo de describir distintas formas en que los modelos de comportamiento del fuego y técnicas de simulación son implementados.

2.1. Características Generales

El interés por disponer de modelos que describan la propagación del fuego aumenta día a día, procediendo estos intereses de distintos sectores: científicos, ambientalistas, ecologistas, gobierno, dueños de tierras, productores agrícolas, madereras, papeleras, etc.

Es importante notar la complejidad que tiene modelar el comportamiento del fuego. Se puede conocer el proceso químico que se produce al arder cierta sustancia, siendo viable la simulación de este proceso, pero cuando lo que se intenta modelar es el comportamiento del fuego en un bosque, la complejidad del problema crece desproporcionalmente.

En un incendio forestal, están involucrados muchísimos combustibles,

donde estos varían en cantidad, en tamaño, en contenido de humedad, etc. Además de la diversidad en los combustibles, también influyen las condiciones ambientales, variando la humedad relativa ambiente, la cobertura de nubes, las condiciones variantes del viento (dirección, velocidad, etc), la topografía del terreno puede variar espacialmente, la pendiente varía tanto en inclinación como así también en la dirección, la acción que el calor, el humo y todo el microclima generado por el incendio tiene en el crecimiento de las llamas y la línea del fuego, etc. Es importante notar también que todas estas propiedades varían en tiempo y espacio, con lo que la información disponible puede llegar a ser, en muchos casos, obsoleta, incompleta, inconsistente, etc.

Todos los problemas mencionados anteriormente tienen influencia en el comportamiento del fuego, además, muchos de estos factores tienen su propio patrón de comportamiento y las interacciones entre estos factores pueden desencadenar comportamientos difíciles de predecir, modelar, simular, etc.

Los modelos de comportamiento de fuego utilizados en la actualidad suelen ser específicos para distintos tipos de fuegos: fuegos de superficie, de copa, *spotting* y aceleración de fuego. Estos comportamientos son abstracciones de un proceso de combustión tridimensional de múltiples combustibles, con su correspondiente transferencia de calor [21]. En la mayoría de los casos, este espacio tridimensional se representa en 2 dimensiones para representar la propagación del fuego en el espacio.

En los siguientes apartados, se enumeran algunos aspectos importantes que influyen en la propagación del fuego y que deben considerarse antes de describir los modelos de propagación.

Combustible, Clima y Topografía

El tipo de combustible, el clima y la topografía del sitio donde se desarrolla el incendio determinan cómo será su propagación. Estos factores harán que el fuego no se propague y se extinga rápidamente o, por el contrario, harán que el fuego crezca y el incendio se propague por varias hectáreas [11] [13] [35].

La cantidad y tipo de combustible inflamable que rodea un incendio se conoce como carga. Ésta es medida por la cantidad de combustible disponible por unidad de área (casi siempre se mide en toneladas por acre). Cuanta más carga, el fuego se propagará más rápido y con mayor intensidad.

Otra de las características que influye en la forma de propagación es el tamaño del material que conforma el combustible. Si se tiene un combustible pequeño, este es más fácil de precalentar y de quemar que un material grande el cual tarda más tiempo en ambos procesos. Esto se debe a la rela-

ción entre la superficie total del material y su volumen. Cuanta mayor sea su superficie con respecto al volumen, el material es más fácil de quemar. Al contrario, cuanto mayor es el volumen con respecto a su superficie, más tardará en quemarse.

La humedad del combustible también afecta a la velocidad en que el material es quemado. Cuanta mayor humedad más tiempo tarda la fuente de calor en secar el combustible y hacer que alcance su temperatura de ignición. Un material con alto contenido de humedad absorbe el calor del fuego haciendo más difícil su propagación.

Temporadas de muchas precipitaciones ayudan a que el fuego no se propague ya que el combustible tendrá alto contenido de humedad. Por el contrario, las temporadas secas son una gran amenaza ya que aumenta las probabilidades de que se produzca un incendio como así también ayuda en la rápida propagación de sus llamas.

Otro factor que afecta es cómo se encuentra el combustible en el escenario. Si se tiene mucho combustible junto, este es más difícil de calentar ya que mantiene mejor la humedad, y además existe menos oxígeno necesario para que el fuego exista. Por el contrario, si el combustible está esparcido, es más fácil de precalentar (por el fuego y el humo) y existe más oxígeno para favorecer el desarrollo del fuego.

Cómo influye la temperatura en el ambiente es directo: a mayores temperaturas se tienen combustibles más calientes y también más secos.

Otro factor con gran importancia en determinar la forma en que avanza el frente del fuego en un incendio es el viento. Cuanta mayor intensidad tenga el viento, más rápida será la propagación ya que el viento empuja las llamas acelerando la propagación. Un viento lo suficientemente intenso determina la dirección de propagación. Además, el viento puede arrastrar restos de combustible encendido y crear otros focos de incendios (*spotting*), el viento ayuda a secar combustible potencial, provee de más oxígeno al fuego, etc.

A pesar de que dependiendo de su intensidad puede ser el parámetro con mayor importancia en la propagación del incendio, también es muy impredecible: el viento de por sí es un factor cambiante, pues sus ráfagas rara vez son constantes en tiempo e intensidad. A esto se le agrega que en los incendios suele generarse un microclima que crea fuertes vientos en su interior, de forma de ráfagas intensas y de torbellinos.

La pendiente del lugar de desarrollo del incendio también afecta a la velocidad y dirección de propagación. El fuego se propaga más rápidamente a favor de la pendiente (subiendo la pendiente). Además, el humo y el calor suben también por la pendiente, precalentando el combustible y haciéndolo más fácil de quemar.

La figura 2.1 muestra esquemáticamente qué factores favorecen a la

propagación del fuego y cuales no lo favorecen (los tres primeros, hacen referencia a características propias del combustible).

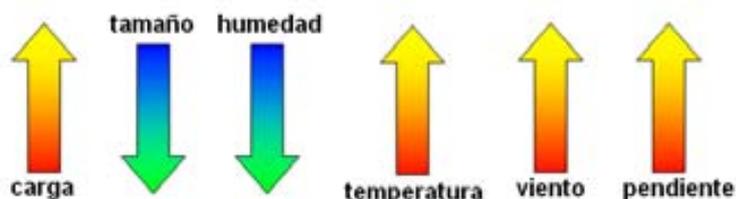


Figura 2.1: Factores que influyen en la propagación del fuego.

Modelos de Combustible

Los modelos de propagación de fuego utilizan las características del entorno donde se desarrolla el fuego para determinar principalmente dos cosas: comportamiento del fuego e índices de riesgo de incendio [5].

A pesar de que en los bosques o áreas forestadas suele encontrarse gran diversidad de especies vegetales, es común que haya un tipo de material que predomine y que sea el que determine cómo se propagaría el fuego en caso de incendio. Los modelos de combustible intentan representar estos materiales para simplificar el uso de los modelos matemáticos de propagación del fuego.

Desde la década de los años 70 se han propuesto modelos para describir los distintos tipos de combustible [2] [5]. Actualmente se utilizan 13 modelos de combustible distintos, divididos a su vez en 4 clases: pastos, arbustos, leñosos y ramas. Estas 4 clases de combustible, al variar sus características, propagan el fuego de distinta forma. Esto es principalmente por la carga de los combustibles y por la distribución a lo largo de los distintos tamaños de partículas.

Estos 13 modelos describen las características de los combustibles que, junto con otros factores, ayudan a determinar la intensidad y el índice de propagación del fuego. Estas características son utilizadas como entradas en los modelos matemáticos de comportamiento del fuego y definen para cada tipo: carga, altura, razón entre volumen y superficie, contenido de humedad del combustible herbáceo vivo y combustible ya muerto, contenido de humedad para la cual dicho combustible ya no propaga, cantidad de silicio, etc.

La figura 2.2 muestra los 3 tipos de combustible del grupo pastos, la

figura 2.3 muestra los modelos del grupo arbustos, la figura 2.4 el grupo leñosos y el grupo 2.5 donde preponderan las ramas.

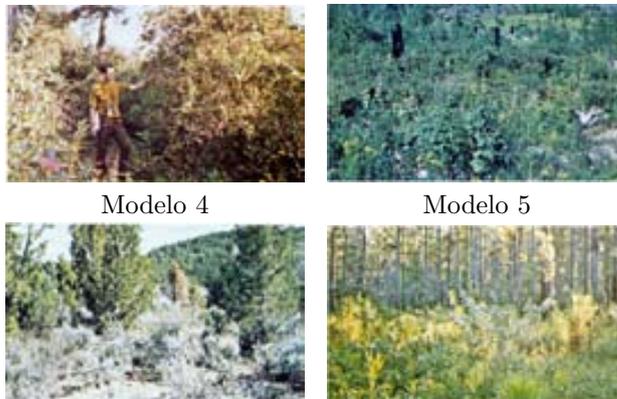


Modelo 1

Modelo 2

Modelo 3

Figura 2.2: Modelos de combustible: grupo pastos.



Modelo 4

Modelo 5

Modelo 6

Modelo 7

Figura 2.3: Modelos de combustible: grupo arbustos.

Dada la importancia del combustible en la propagación del fuego, estos modelos se usan como entrada en muchos de los modelos y simuladores desarrollados.

La mayoría de los sistemas de simulación de incendios forestales combina un *modelo* subyacente de comportamiento del fuego con alguna *técnica de simulación*. Categorizando los distintos modelos y las distintas técnicas de simulación se pueden encontrar similitudes y diferencias entre los distintos simuladores [3]. En la próxima sección se describirán los distintos modelos de comportamiento del fuego, y a continuación se presentarán las técnicas de simulación más utilizadas para la implementación de dichos modelos.

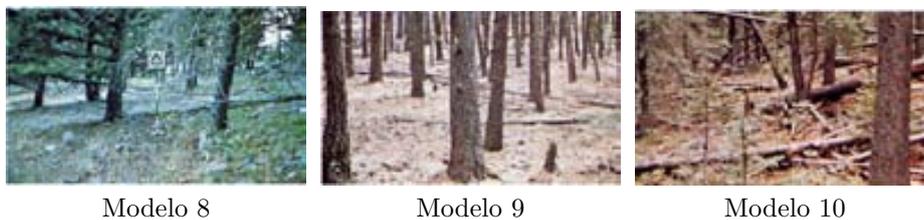


Figura 2.4: Modelos de combustible: grupo leñosos.

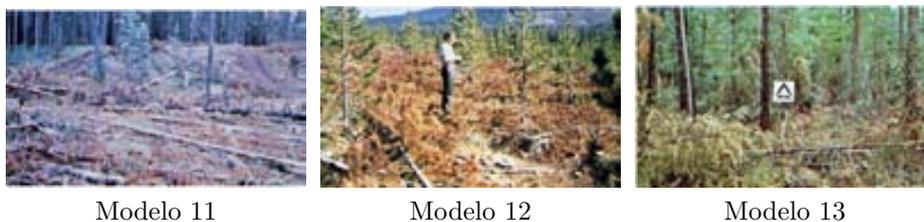


Figura 2.5: Modelos de combustible: grupo ramas.

2.2. Modelos de Comportamiento del Fuego

Los modelos de comportamiento del fuego simulan el comportamiento del fuego (características tales como *ROS*, intensidad del fuego, altura de llama), usando datos específicos del lugar donde se desarrolla el fuego (clima, características del terreno, tipo de combustible, etc). La propagación del fuego libera energía desde la combustión y transporta parte de la misma a combustible adyacente no quemado, calentándolo hasta el punto de ignición. Diferentes modelos de propagación del fuego usan diferentes modelos predictivos para cuantificar este proceso de ignición. En las próximas subsecciones describiremos cuatro modelos de predicción diferentes: físico, físico-estadístico, estadístico y probabilístico [3].

2.2.1. Modelos Físicos

Estos modelos predicen la propagación del fuego basándose en las características físicas de la combustión. En estos modelos se consideran los 3 modos de transferencia de calor: conducción, convección y radiación (descritos en el capítulo anterior).

A pesar de que se han desarrollado diversos modelos físicos, su utiliza-

ción se hace muy difícil debido a la gran cantidad de datos detallados que requieren.

2.2.2. Modelos Físicos-Estadísticos

Estos modelos combinan la teoría física con correlaciones estadísticas para generar fórmulas que describen el comportamiento del fuego. Un ejemplo muy utilizado es el modelo de Rothermel, el cual se explicará en la sección 2.2.5 como caso de estudio. Este modelo es muy utilizado en muchos simuladores, por eso se dedicará una sección especial para describirlo (el simulador utilizado en este trabajo implementa sus ecuaciones).

2.2.3. Modelos Estadísticos

Estos son modelos que utilizan un conjunto de ecuaciones que son derivadas a partir de distintos fuegos utilizados como tests. Estas ecuaciones predicen parámetros del fuego tales como: *ROS*, consumo del combustible, intensidad de la línea del fuego, etc.

Como estos modelos no se basan en procesos físicos, sino que en tests, su efectividad en la predicción está limitada a condiciones similares a los fuegos incluidos como tests.

2.2.4. Modelos Probabilísticos

Estos modelos se basan en tablas de contingencia más que en características físicas o ecuaciones estadísticas. En estos modelos cada variable del ambiente (como tipo de combustible, humedad del combustible, viento) es asignado a una de varias categorías discretizadas. Luego, se utilizan las tablas de contingencia para simular las probabilidades de propagación del fuego desde un punto hacia el próximo.

Este modelo tampoco se basa en características físicas del proceso de combustión, por lo que es aplicable sólo bajo condiciones similares para los que fue desarrollado. Estos modelos son generalmente utilizados para simular la ignición y simular las probabilidades de propagación para una secuencia de fuegos hipotéticos sobre terrenos, y no para predecir la propagación de un fuego específico.

2.2.5. Caso de Estudio: Modelo de Rothermel

Richard Rothermel ([39]) ha definido un modelo de comportamiento del fuego (modelo físico-estadístico) asumiendo que el mismo avanza sobre superficies con combustible fino y continuo: pastos, suelo cubierto de acículas

de pino u hojas, arbustos, ramas caídas, etc.

El modelo de fuego desarrollado por Rothermel evalúa la energía generada por el fuego, la transmisión de calor desde el fuego al combustible próximo y la energía absorbida por el combustible. En este modelo se consideran combustibles vivos y muertos, sus contenidos de humedad (estos afectan tanto la energía generada como la energía absorbida). También se consideran los efectos del viento y de la pendiente en la transferencia del calor, la carga, la compactitud del combustible, el tamaño de las partículas que lo componen, etc. Todos estos factores determinan la forma y la velocidad en que el fuego se propaga.

Aunque son factores que influyen en el avance y comportamiento del fuego, este modelo no considera ni modela: fuego de copas, torbellinos de fuego, *spotting*, etc.

El modelo de Rothermel es uno de los modelos más utilizados para predicción del comportamiento del fuego. La mayoría de los simuladores de comportamiento del fuego basan sus cálculos en este modelo [10]. Sus operaciones calculan el índice de máxima propagación y la intensidad de reacción del fuego conociendo ciertas propiedades del combustible y del ambiente donde se desarrolla el incendio.

Las ecuaciones que componen el modelo desarrollado por Rothermel requieren la descripción del combustible, incluyendo: altura, carga, porcentaje de combustible muerto, humedad de extinción, contenido de calor, relación entre superficie y volumen del combustible, contenido mineral, contenido de silicio y densidad de partículas. También requiere datos del ambiente, incluyendo: velocidad del viento a media altura de la llama, pendiente y contenido de humedad del combustible (vivo y muerto) [41].

El modelo de Rothermel calcula y proporciona el vector de máxima propagación del viento. La ecuación 2.1 muestra la fórmula propuesta por Rothermel para modelizar la propagación del fuego. Dicha fórmula calcula el índice de propagación del fuego sobre combustible uniforme según sigue:

$$R = \frac{I_r \xi (1 + \Phi_w + \Phi_s)}{P b^\epsilon Q i g} \quad (2.1)$$

donde,

R	índice de propagación del frente del fuego (pies/minutos).
I_r	intensidad de reacción - índice de energía liberada del frente del fuego ($Btu/pies^2/min$).
ξ	índice de flujo de propagación (intensidad de reacción, la cual calienta partículas de combustible adyacente para arder).
Φ_w	efecto del viento.
Φ_s	efecto de la pendiente.
Pb	cantidad de combustible.
ε	medida de proporción del combustible parcial que es calentado a temperaturas de ignición en el momento que comienza la ignición.
Qig	cantidad de calor requerido para quemar una libra de combustible (pre-ignición).

Esta ecuación es una relación entre el calor recibido por el combustible con respecto al calor necesario para llevar este combustible a la temperatura de ignición. Rothermel determina esta velocidad mediante el empleo de ecuaciones obtenidas a partir de datos experimentales (usando técnicas de correlación estadística y en varios casos ajustes mediante la incorporación de parámetros empíricos). Esto implica una simplificación del modelo que le da validez en un margen de condiciones específicas (relacionadas con la disposición del combustible y características medio ambientales) [52].

Como se mencionó anteriormente, el viento y la pendiente determinan la dirección en que avanza el fuego. Esto se ve reflejado en las ecuaciones del modelo de Rothermel donde se determina que la dirección de máxima propagación del fuego resulta de componer el efecto del viento y de la pendiente como se muestra en la ecuación (2.2):

$$\Phi = \Phi_w + \Phi_s \quad (2.2)$$

donde Φ_s es la dirección de máxima pendiente y su módulo es el *efecto de la pendiente* y Φ_w es la dirección del viento y su módulo es el *efecto del viento*. A su vez, el efecto de la pendiente está determinado según muestra la ecuación 2.3:

$$\Phi_s = 5,275 \beta^{-0,3} \tan \omega^2 \quad (2.3)$$

la ecuación 2.4 muestra el efecto del viento:

$$\Phi_w = C (3,281 v)^B (\beta / \beta_{op})^{-E} \quad (2.4)$$

donde v es la velocidad del viento, β es la razón de compactación del manto de combustible, β_{op} es la razón de compactación óptima, ω es la pendiente (en radianes), y los coeficientes C, B y E están en función del tamaño

de las partículas que componen el manto de combustible [2] [5] [38] [41]. Entonces, el modelo define la dirección de máxima propagación y el *ROS* en esta dirección a partir de la composición de la dirección y los grados de inclinación de la pendiente y de la dirección y la velocidad del viento.

Este modelo conforma el núcleo básico de la mayoría de los simuladores de incendios forestales más utilizados [10].

Una vez vistos los diferentes modelos de comportamiento del fuego, se describirán a continuación las distintas técnicas de simulación.

2.3. Técnicas de Simulación

Todos los simuladores utilizan, además del modelo del comportamiento del fuego subyacente, una técnica de simulación que representa la propagación del fuego en el terreno [3]. Cada técnica de simulación difiere de otras en cómo representan el terreno y el proceso de propagación.

El terreno puede ser visto como una malla de cajas o celdas discretas, donde la propagación del fuego de una celda a otra está gobernada por un conjunto específico de reglas o por probabilidad de ocurrencia. Otra forma, es modelar la línea de fuego por funciones matemáticas sobre un terreno visto como un medio continuo. En las próximas subsecciones describiremos tres técnicas de simulación: *bond percolation*, propagación elíptica y autómatas celulares, poniendo mayor énfasis a las últimas dos, ya que son las que más se utilizan en los simuladores actuales.

2.3.1. *Bond Percolation*

Esta técnica representa el terreno como una malla de celdas cuadradas, triangulares o rectangulares. El fuego propaga desde una celda a otra celda vecina que contenga combustible quemable.

El usuario puede ajustar la probabilidad de propagación para una dirección de propagación debido a factores como viento, topografía y diferencias entre los tipos de combustibles. Si muchas celdas contienen combustible sin quemar y la probabilidad de propagación es alta, entonces, el fuego se propaga a través de la malla.

Esta técnica debe ser “sintonizada” ajustando las probabilidades de tal manera que el fuego modelado propague de forma comparable a un fuego real bajo condiciones de clima y combustibles similares.

Debido a que esta técnica no se basa en las características físicas de la propagación, la efectividad de la simulación es limitada a condiciones similares para las que la técnica fue sintonizada.

2.3.2. Propagación elíptica

Esta técnica proyecta el terreno como un medio continuo y no como una malla de celdas. Considerando la propiedad de que bajo condiciones uniformes el crecimiento del fuego en dos dimensiones es elíptico, entonces, esta aproximación implementa el crecimiento del fuego utilizando las propiedades de comportamiento del fuego junto a las propiedades matemáticas de las elipses ([21] y [3]).

Teniendo en cuenta esta propiedad, Anderson et al. [5] propone definir puntos espaciados regularmente en el perímetro de la línea de fuego donde cada uno representa un fuego pequeño independiente que se propaga de forma elíptica, donde el tamaño y forma de cada elipse están determinados por las condiciones locales. El perímetro de la línea de fuego (total) a cada paso de tiempo es definida por todas las pequeñas elipses.

Para este grupo de elipses se han desarrollado ecuaciones diferenciales que describen la propagación del fuego para combustible heterogéneo, topografía no uniforme y clima variable.

Esta forma de representar el crecimiento del fuego evita distorsiones encontradas en otras implementaciones, como las que observaremos en la próxima sección cuando se describan autómatas celulares.

En esta implementación, se utiliza el principio de Huygens, donde el frente del fuego es propagado como un polígono en continua expansión. El polígono es definido como una serie de vértices (2 dimensiones, puntos en las coordenadas X, Y). El número de vértices crece al crecer el frente del fuego a medida que transcurre el tiempo (expansión del polígono).

Richards (1990, 1995) implementa el principio de Huygens como método para simular el crecimiento de una línea de fuego en incendios forestales [21]. Las ecuaciones diferenciales del modelo describen la expansión de una onda elíptica desde una serie de vértices que definen los bordes del fuego. El principio de Huygens asume que cada vértice puede servir como fuente de una expansión elíptica independiente (figura 2.6 [21]).

Para determinar cuál será la propagación desde cada vértice, se necesita saber: (1) la orientación del vértice en el frente del fuego en términos de componentes diferenciales x_s, y_s , (2) la dirección de máximo ROS (llamada θ , el vector viento-pendiente resultante, en radianes azimuth), y (3) la forma de un fuego elíptico determinado por las condiciones locales de dicho vértice en términos de las dimensiones a, b, c (figura 2.7, [21]). A partir de estas entradas, las ecuaciones propuestas por Richards (1990), computan las diferenciales X_t y Y_t de la propagación ortogonal de dicho vértice:

$$X_t = \frac{a^2 \cos \theta (x_s \sin \theta + y_s \cos \theta) - b^2 \sin \theta (x_s \cos \theta - y_s \sin \theta)}{(b^2 (x_s \cos \theta + y_s \sin \theta)^2 - a^2 (x_s \sin \theta - y_s \cos \theta)^2)^{\frac{1}{2}}} + c \sin \theta \quad (2.5)$$

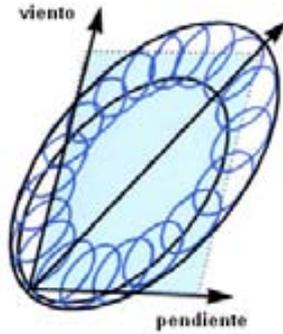


Figura 2.6: Crecimiento elíptico del fuego.

$$X_t = \frac{-a^2 \sin \theta (x_s \sin \theta + y_s \cos \theta) - b^2 \cos \theta (x_s \cos \theta - y_s \sin \theta)}{(b^2 (x_s \cos \theta + y_s \sin \theta)^2 - a^2 (x_s \sin \theta - y_s \cos \theta)^2)^{\frac{1}{2}}} + c \cos \theta \quad (2.6)$$

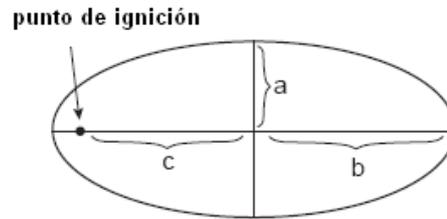


Figura 2.7: Dimensiones utilizadas para computar el crecimiento elíptico del fuego.

Estas ecuaciones han sido originalmente desarrolladas para terrenos planos (sin pendiente). En este tipo de terreno, el sistema de coordenadas horizontal se proyecta sin modificaciones sobre el terreno. Pero cuando el terreno tiene pendiente, esto no es así. Las entradas de las ecuaciones 2.5 y 2.6 (x_s , y_s y θ) son almacenadas en el ordenador en forma de plano horizontal, y deben transformarse a superficie. Del mismo modo, las salidas X_t y Y_t deben ser transformadas desde el plano del terreno al plano horizontal

para ser almacenadas efectivamente.

La expansión de la línea del frente del fuego, representada por la expansión de la elipse, es determinada calculando el *ROS* y dirección de cada vértice y multiplicándolo por el tiempo transcurrido (tiempo de cada paso de simulación). La normal de la dirección y *ROS* del vértice se determina mediante la máxima propagación y su dirección por una transformación elíptica.

El cálculo de la propagación de cada vértice del frente del fuego se asume independiente de los demás.

2.3.3. Autómatas Celulares

Esta técnica representa el terreno como una malla de celdas, donde cada celda contiene un conjunto de valores (pendiente, tipo de combustible, orientación, etc.). Cada celda comienza en un estado inicial, y la probabilidad de propagar el fuego a cada celda en la malla es determinado por el mismo conjunto de reglas para todas las celdas [3].

El usuario utiliza parámetros (tipo de combustible, humedad del combustible, características del clima y viento) para determinar la propagación por la malla. Como las reglas que gobiernan la propagación pueden estar basadas en aspectos físicos del proceso de ignición, esta técnica puede ser aplicada a un gran rango de condiciones [3].

Este tipo de implementación asume condiciones uniformes en secciones pequeñas del terreno y, discretizando el tiempo, el fuego se propaga sobre la malla haciendo viajar el fuego desde una celda a sus 8 celdas vecinas [21] (asumiendo una malla rectangular, existen distintas variaciones, que ejemplificaremos en secciones siguientes). Se iteran sucesivos pasos, siempre comenzando desde la celda con menor tiempo de ignición, propagando así el fuego hacia las celdas vecinas (figura 2.8).

Al propagar el fuego desde una celda a sus 8 celdas vecinas, la forma final del fuego queda fuertemente distorsionada. Existen distintas aproximaciones, algunas intentan reducir esta distorsión incrementando el número de vecinos de cada celda, modificando la forma de las celdas, modificando el tamaño de las celdas a medida que el fuego avanza, etc.

Las cuatro próximas subsecciones muestran muy brevemente ejemplos de distintas aproximaciones que proponen mejorar los autómatas celulares a partir de considerar cambios en la malla que representa el terreno.

Autómata Celular con Celdas Hexagonales

Una variación a considerar celdas cuadradas es descrita en el trabajo [41], donde se utilizan celdas hexagonales en toda la malla.

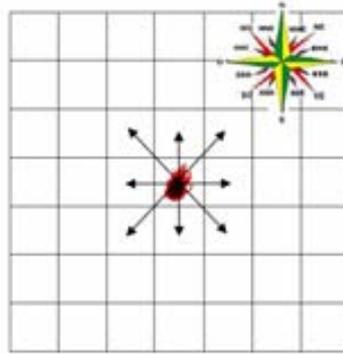


Figura 2.8: Propagación del fuego utilizando autómatas celulares.

Principalmente, se aprovechan las características de representar el avance del fuego mediante una malla de celdas, donde cada celda tiene un estado (el cual depende de su estado local como así también de la interacción con las celdas vecinas), y a su vez, la propagación del fuego es una función de transición (idéntica para todas las celdas). Entonces, el autómata celular evoluciona cambiando el estado de las celdas a pasos discretos de tiempo, según la función de transición.

Con el objetivo de evitar falsas simetrías producidas por mallas de celdas cuadradas, se propone un modelo basado en celdas hexagonales (como se muestra en la figura 2.9, [41]).

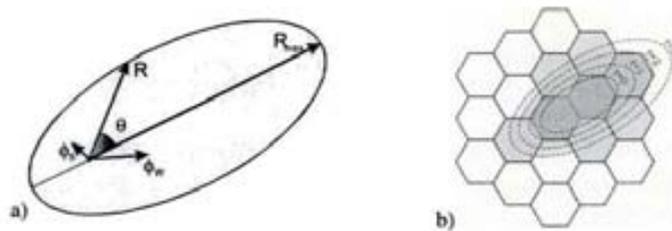


Figura 2.9: a) Área elíptica de propagación del fuego. b) Patrón de propagación entre vecinos hexagonales.

En esta propuesta, el estado de cada celda explica todas sus características, las relevantes a todo el sistema completo así como también la

información relativa al espacio correspondiente a dicha celda. Cada característica es individualizada como un subestado (constante en toda la celda). Al trabajar con subestados dentro de las celdas, la función de transición es dividida en distintos componentes. En esta implementación, se reconocen 2 componentes: transformación interna (dependiente de los subestados de la celda) e interacciones locales, dependiendo de los estados de las celdas vecinas.

En este trabajo, se muestra el uso de este sistema con un ejemplo de un fuego real, ocurrido en Sardinia en agosto del 1998. Los resultados son satisfactorios, el formato final de la simulación no se ve degradado por el formato elegido para las celdas. Aunque este ejemplo ha dado buenos resultados, observan la dificultad de tener todos los datos de entrada necesarios para la simulación, y esperan validar este sistema con diversos combustibles, distintas características morfológicas y más condiciones de clima.

Autómata Celular Utilizando una Malla Multiresolución Dinámica

Esta es otra aproximación definida en [25]. En dicho trabajo, se propone un espacio de celdas multidimensional, esto es, el tamaño (resolución) de las celdas varían dinámicamente y adaptativamente a medida que la simulación evoluciona. Las celdas que están cerca del frente del fuego cambian a una resolución mayor (tamaño de celda menor) y las celdas lejanas al frente del fuego tienen una resolución menor (tamaño mayor de la celda). Entonces, esta aproximación permite una simulación en alta resolución sin crear un modelo de alta resolución desde el principio.

En este trabajo, se menciona que la resolución elegida para las celdas afecta la precisión con la que se representa lo que realmente hay en el sitio representado por la celda (topografía, combustible, etc). En consecuencia, la resolución afectará el resultado total de la simulación. Entonces tener alta resolución permite tener las condiciones espaciales mejor representadas, pero, trabajar con muchas celdas pequeñas afecta directamente el cómputo requerido para la simulación.

Como primer paso en este trabajo, se utiliza un espacio de celdas de dos resoluciones: resolución baja (celdas grandes) y celdas de alta resolución (celdas pequeñas). Durante la simulación, si es necesario, una celda de baja resolución es reemplazada por cuatro celdas de alta resolución.

Cada celda tiene un estado, y dicho estado cambia a medida que avanza el fuego (algunos de los estados son: no quemada, ardiendo, quemada, etc. que corresponde con el modelo DEVS *Fire Spread Model* [Ntamo et al., 2004]).

Un problema importante que surge al trabajar con distintas resoluciones

de celdas es el de consistencia. Para asegurar consistencia, las celdas están conectadas entre ellas por puertos de entrada/salida. Para mantener consistencia cuando una celda de baja resolución se divide en 4 de alta resolución, estas nuevas celdas mantienen el estado de la celda padre. El tamaño de las nuevas celdas corresponde a la mitad de su dimensión (alto y ancho). El comportamiento de la celda (quemada, no quemada, ardiendo), se mantiene de una celda de baja resolución a sus celdas hijas. Los atributos del ambiente de cada celda, que son obtenidos de fuentes como GIS o fuentes de información climatológica, se vuelven a obtener desde aquí con la precisión disponible, usando mecanismos de agregación o disgregación cuando es necesario. Por último, los atributos referentes a la ignición (intensidad del fuego, velocidad del fuego y su dirección), son atributos que no se asignan al inicializar la celda, se asignan sólo cuando la celda es alcanzada por el fuego. Dado que un cambio de resolución ocurre sólo antes de la ignición o una vez que ya está quemada dicha celda (en tiempo de ejecución), los atributos referentes a la ignición no plantean problemas de consistencia.

La figura 2.10 muestra los resultados presentados, donde se compara la propagación utilizando el mapa entero con celdas de resolución baja (a), celdas con resolución alta (b) y por último, la propuesta de cambiar dinámicamente la resolución de las celdas.

Se puede ver que el área quemada cuando se utiliza multiresolución, es igual que cuando el mapa está dividido en celdas todas de alta resolución. Entonces, en este esquema se evita trabajar con todas celdas de alta resolución (evitando la penalización que implica el tiempo de cómputo) y se logran simulaciones similares a esta opción.

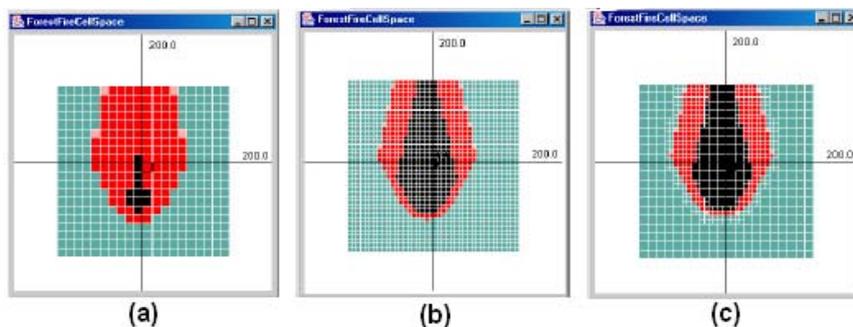


Figura 2.10: (a) Simulación utilizando celdas de baja resolución. (b) Simulación utilizando celdas de alta resolución y (c) Simulación utilizando celdas de distantes resoluciones.

Autómata Celular Utilizando Geometría *Taxicab*

Otra aproximación que intenta evitar las distorsiones generadas por la representación del terreno en autómatas celulares la encontramos en el trabajo descrito en [12].

En este trabajo se describen distintas fuentes de error para las predicciones (las cuales se enumerarán más adelante en el capítulo 3). Pero una fuente de error mencionada es la discretización en celdas cuadradas de los mapas, como así también distintos factores del entorno del fuego. Esto da como resultado una simulación del avance del fuego errónea debido al cómputo de distancias, por ejemplo.

Las simulaciones basadas en autómatas celulares, prácticamente describen el avance del fuego haciendo viajar el fuego desde una celda a sus 8 vecinas, calculando el *ROS* y la distancia para determinar el momento preciso en que el fuego alcanza el centro de la celda. En este trabajo se habla de la distancia euclidiana y de la distancia autómatas, la cual difiere de la euclidiana por discretizar las distancias y obtener la sumatoria de las distancias vecino a vecino en vez de la distancia euclidiana de un punto a otro (figura 2.11 (a)).

Se hace referencia entonces a la geometría *taxicab* (alusión a la distancia recorrida por un taxi en una ciudad recorriendo las calles para llegar de un punto a otro, figura 2.11 (b)), donde la distancia es la suma de las diferencias (absolutas) de sus coordenadas. Pero en el contexto de simulación de propagación de incendios, a esta opción se le agregan las diagonales para la representación de la relación de una celda con sus 8 vecinos (figura 2.11 (c)).

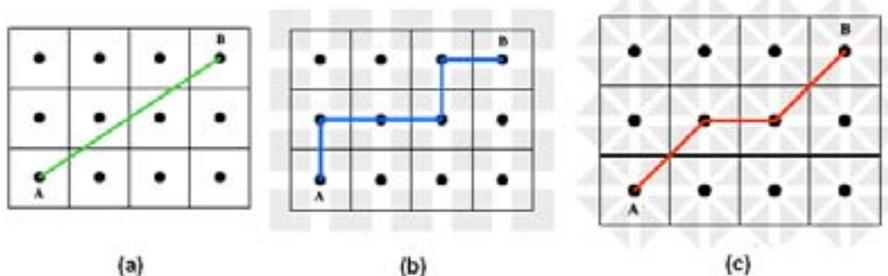


Figura 2.11: (a) Distancia euclidiana entre los puntos A y B. (b) Distancia entre los puntos A y B utilizando geometría *taxicab*. (c) Distancia entre los puntos A y B utilizando geometría *taxicab* extendida.

Pero teniendo en cuenta el formato elíptico del avance del fuego, y utilizando la geometría *taxicab*, se observan las distorsiones en el formato del mapa mostradas en la figura 2.12.

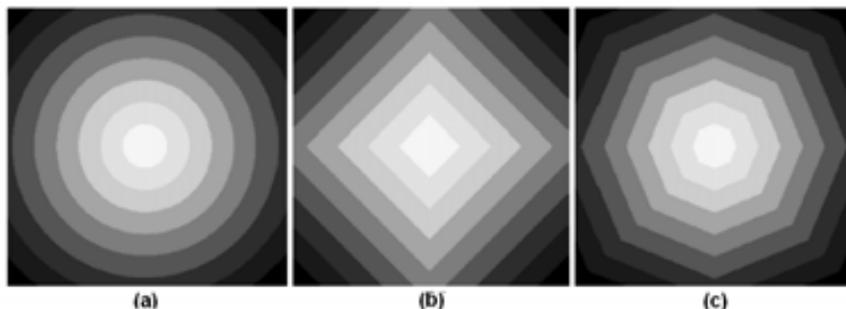


Figura 2.12: Representación de líneas circulares utilizando (a) distancias euclidianas, (b) *taxicab* y (c) *taxicab* extendido.

Entonces, para evitar estas distorsiones, se utilizan algunas reglas que dependen de los cambios o similitudes existentes entre las condiciones y propiedades de cada celda (el tipo de vegetación, pendiente, viento, etc.). Así, si entre dos puntos que se quieren unir las diferencias entre sus propiedades no superan un umbral y se asume que son uniformes, entonces se utiliza la distancia euclidianas entre dichos puntos. Si las propiedades no son uniformes, se considera un nuevo punto de referencia, pasando el lugar donde ocurre el cambio (haciendo que el punto de referencia y el segundo punto tengan características homogéneas). Luego, se suman las distancias en relación a este nuevo punto de referencia. Pero si los cambios ocurren celda a celda, como suele suceder en la realidad, se realiza una nueva mejora, donde se consideran dos distancias nuevas y estas distancias son calculadas, una teniendo en cuenta la primera ley (celdas con condiciones homogéneas) y la segunda distancia teniendo en cuenta la segunda ley (celdas no homogéneas).

La geometría *taxicab* extendida explica y demuestra que hay una diferencia significativa entre la distancia euclidianas y la distancia utilizando geometría *taxicab*. Estas diferencias deberían ser consideradas por los programadores al desarrollar sus soluciones.

Simulador de Incendios Forestales con Celdas Irregulares

En los trabajos [26] y [27], se describe un simulador basado en la transferencia de calor. En dichos trabajos se intenta evitar los problemas derivados

de la representación geométrica del terreno, asumiendo una representación del mismo a través de un espacio dividido en celdas irregulares (figura 2.13 [26]). Estas celdas tienen formas y tamaños irregulares para evitar la influencia geométrica en la forma final del fuego que resultan de utilizar un espacio de celdas rectangulares y homogéneas.

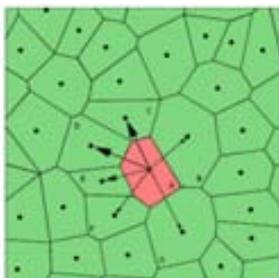


Figura 2.13: Disposición del terreno dividido en celdas irregulares.

Cada una de las celdas representa una porción del terreno, y cada una tiene sus propios parámetros de estado: tipo de combustible, carga y humedad, temperatura y estado de ignición. El simulador se basa en eventos, los cuales son bastante sencillos: cuando una celda alcanza la temperatura de ignición, cambia del estado *flammable* (estado que recibe calor y lo convierte en un cambio de temperatura) al estado *burning* (donde el combustible es convertido en calor). Cuando todo el combustible en una celda es consumido, cambia el estado a *burnt*. La cantidad de calor transmitida (frecuencia de transmisión de calor) depende del gradiente de la temperatura y de las propiedades de los *links* de comunicación entre las celdas. La conductividad de calor de un *link* de comunicación es dinámica y está influenciada por la pendiente y dirección y velocidad del viento. La ignición de una celda ocurre cuando recibe suficiente calor como para arder.

Las celdas son creadas a partir de puntos distribuidos irregularmente. Alrededor de cada punto, se construye un polígono que contiene todo el territorio que está más cerca de ese punto que de cualquier otro punto en el espacio. Los vecinos de una celda son las celdas con las que comparte algún borde. Esta representación del terreno es estática, no cambia durante la simulación, pero lo que cambia es el estado de las celdas, capturando el comportamiento dinámico del fuego.

En este trabajo se presentan distintos resultados, los cuales muestran que la representación del terreno por celdas irregulares logra evitar las dis-

torsiones en el formato del fuego producidas por mallas de rectángulos regulares. En este trabajo se hace alusión a que en distintos trabajos se ha intentado reducir la distorsión aumentando la cantidad de vecinos, por ejemplo. Pero al utilizar celdas cuadradas que discretizan el avance del fuego, el error producido por las celdas involucradas es siempre el mismo y se van magnificando (sumando), en cambio, utilizando celdas irregulares, los errores producidos entre las distintas celdas son distintos, los cuales se van cancelando a medida que el fuego avanza.

2.4. Simuladores del Comportamiento del Fuego

En la actualidad, existen diversos simuladores de comportamiento del fuego. En las secciones siguientes, se describirán brevemente las principales características de los simuladores más utilizados actualmente y otros más antiguos y tal vez no tan utilizados [22] ([3]). En particular, se describirá más ampliamente el simulador *fireLib* ya que es el simulador utilizado como motor de simulación en este trabajo.

Los distintos simuladores pueden diferenciarse en el tipo de incendio que simulan (superficial, copa, *spotting*, etc.), si incluyen uno o varios de estos tipos de incendios, el modelo de predicción (físicos, físicos-estadísticos, estadísticos o probabilísticos), la técnica de simulación (elíptico, *bond percolation* o autómatas celulares), además de las características de sus entradas y salidas, utilización de interfaces gráficas o entornos de visualización, requerimientos de conocimiento del dominio por parte del usuario, etc.

Es útil tener en cuenta que algunos simuladores han sido desarrollados hace más de 15 años (por ejemplo, el simulador DYNAFIRE, desarrollado en 1991), y que la tecnología disponible en cada momento condiciona las posibilidades del tratamiento del fuego, entradas, salidas, etc., de cada simulador. A continuación se describirán las principales características de diversos simuladores, concluyendo este análisis con una tabla resumen de la información expuesta a lo largo de estas secciones.

2.4.1. BehavePlus

BehavePlus es una actualización y extensión de BEHAVE *Fire Behavior Prediction System* [6]. Anteriormente, BEHAVE era un sistema que consistía de 5 programas. Dos de ellos eran empleados para calcular la predicción del fuego desde una entrada suministrada por el usuario. Otro programa proveía al usuario de tablas que eran útiles en la planificación de fuegos. Y otros dos programas eran utilizados para desarrollar un mode-

lo de combustible cuando los 13 modelos de combustible estándar ([5]) no eran adecuados. Los modelos de combustible definidos por el usuario eran salvados en archivos para su uso por los otros 3 programas. El desarrollo de BEHAVE en 5 programas se debía principalmente a la limitada capacidad de los recursos de la época [8].

La extensión BehavePlus es un simulador que opera bajo el entorno Windows, en cualquier PC de escritorio. BehavePlus es una colección de modelos que describen el comportamiento del fuego, su entorno y sus efectos.

BehavePlus provee los medios para modelar el comportamiento del fuego (ROS y distancia de *spotting*), efectos del fuego (tales como altura de la brasa y mortandad de árboles) y el ambiente del fuego (como la humedad del combustible y el factor del viento). Estas entradas son introducidas directamente por el usuario, para obtener como salida gráficos, tablas y diagramas que describen las principales características de la propagación resultante.

Para todos los cálculos se asume que las condiciones son uniformes y



Figura 2.14: BehavePlus

constantes para el período de tiempo que dura la simulación.

A pesar de que se recomienda el uso a usuarios experimentados o al menos familiarizados con el problema, tiene la opción de utilizar una serie de ventanas que permiten introducir los valores de los parámetros de entrada como así también mostrar la salida de una forma amigable y fácil de utilizar.

Tras una simulación, BehavePlus muestra las relaciones entre humedades, viento, pendiente e índice de propagación de forma gráfica. Este sistema produce gráficos, tablas, diagramas y puede ser utilizado por diversas aplicaciones para manejo del fuego (figuras 2.15 (a) y (b)).

Actualmente BehavePlus cuenta con numerosos módulos [7], los cuales, por ejemplo, calculan: ROS del fuego de superficie, intensidad de la línea de fuego y longitud de llama, intensidad de reacción y calor por unidad de área, transición del fuego de superficie a copas, ROS del fuego de copas,

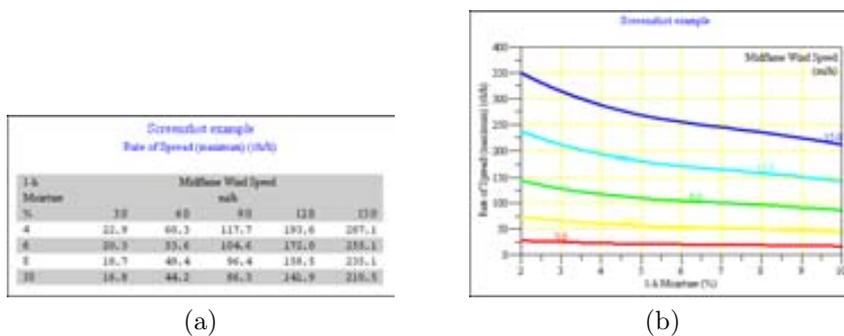


Figura 2.15: Salidas típicas de BehavePlus: (a) Tabla de salida de BehavePlus y (b) Gráfico del ROS, con humedad muerta 1h, y velocidad del viento a media llama como variables.

área y perímetro del fuego de copas, zonas de seguridad, perímetro y forma del fuego, distancia máxima de *spotting*, probabilidad de mortalidad de árboles, probabilidad de ignición, etc.

2.4.2. Clarke Cellular Automaton System

Este simulador realiza simulaciones de propagación de fuegos potenciales y del comportamiento de extinción del fuego. El sistema puede simular el comportamiento de un fuego simple bajo condiciones constantes como así también condiciones variables. Los resultados se utilizan para medir el riesgo de incendios potenciales.

Los datos de entrada se obtienen a partir de sensores remotos, modelos digitales de elevación (DEMs), y condiciones ambientales locales. Se utilizan mapas GIS para describir el combustible presente y las características del terreno. Otras entradas incluyen temperatura, humedad relativa, humedad del combustible y una tabla de velocidad y dirección del viento. Las medidas y los mapas que se obtienen como salida permiten identificar áreas de alto riesgo de incendio [3].

2.4.3. DYNAFIRE

DYNAFIRE utiliza el modelo de predicción BEHAVE y como técnica de simulación implementa la opción de autómatas celulares. Para su correcta ejecución se utiliza pMAP (*Spatial Information Systems*) y los cálculos se realizan sobre el terreno representado por una malla de celdas, las cuales tienen condiciones constantes dentro de la celda pero pueden variar de una celda a otra.

Las entradas que utiliza incluyen datos GIS: capas que describen el combustible, la elevación, la pendiente, la orientación del terreno y cauces de agua. Otras entradas que son adicionales son el clima diurno, humedad del combustible, humedad relativa del ambiente, velocidad y dirección del viento, humedades del combustible vivo y muerto a 10 y 100 horas [3].

Las salidas de este simulador incluyen mapas del perímetro del fuego (por hora), intensidad de la línea del fuego, promedio de *ROS* y dirección de la propagación del fuego.

2.4.4. EMBYR

Este simulador no fue diseñado para predecir el comportamiento de un incendio particular, sino que es un modelo probabilístico que intenta predecir patrones de fuegos potenciales de grandes incendios. Realiza estos cálculos a partir de variaciones del combustible del lugar y patrones de clima del área estudiada.

Utiliza la técnica de simulación *bond percolation* y las probabilidades de ignición en las celdas vecinas están basados en datos empíricos. Además se simula la distancia de *spotting*, el nacimiento de otros fuegos a partir de combustible ardiente, etc.

Este sistema se basa en datos empíricos, donde el usuario especifica una tabla de probabilidades de propagación para varios tipos de combustible bajo una de las tres condiciones de humedad: mojado, intermedio o seco. Las probabilidades de propagación son ajustadas introduciendo un factor que incluye una de 3 categorías de velocidad del viento y una de las 8 posibles direcciones.

La salida de este simulador consiste en un patrón de fuego de uno o más fuegos potenciales. Estos mapas pueden ser usados para evaluar el posible impacto de futuros fuegos en un área determinada. Debido a la gran cantidad de posibles combinaciones de especies de vegetales que habitan un bosque o un área forestada, las distintas edades de estas especies, las condiciones de humedades variantes, etc., todo esto hace que generar mapas de probabilidades empíricas o tablas de datos sea difícil de lograr [3].

2.4.5. FIREMAP

Utiliza autómatas celulares para simular fuegos de superficie considerando condiciones no uniformes del terreno y combustible heterogéneo. Utiliza el modelo BEHAVE para predecir dirección, velocidad e intensidad de fuegos de superficie.

Necesita como entradas ficheros *raster* provenientes de GIS, que incluyan: modelo de vegetación (alguno de los 13 modelos definidos en [5]), ele-

vacación, pendiente, orientación del terreno, etc. También se puede especificar las humedades del combustible o las mismas se pueden calcular usando las ecuaciones de BEHAVE. Además se pueden utilizar datos del viento (dirección y velocidad), temperatura y humedad relativa.

Las salidas son de la misma resolución que la utilizada en los datos de entrada. Los mapas que este simulador ofrece como salidas, incluyen datos de *ROS*, intensidad de la línea de fuego, altura de llama, calor por unidad de área e intensidad de reacción [3].

2.4.6. WILDFIRE

Este sistema es un simulador simple para comportamiento del fuego que utiliza la técnica de simulación de crecimiento elíptico del fuego.

Tiene una interfaz gráfica para especificar el combustible y los datos del terreno. Se pueden especificar barreras como rutas, puntos o cauces de agua. El sistema permite un terreno con pendiente variable. Las condiciones del clima son consideradas como uniformes, excepto el viento que puede cambiar en el tiempo.

Las salidas de este simulador consisten en mapas del perímetro del fuego en los pasos de tiempo especificados por el usuario y la intensidad de la línea de fuego final [3].

2.4.7. NEXUS

NEXUS es una aplicación que relaciona modelos de predicción de fuegos de copa y fuegos de superficie para computar índices de fuegos potenciales de copa. NEXUS es útil para estudiar dicho riesgo y determinar formas alternativas de tratarlos. Esta aplicación incluye herramientas visuales que son útiles para entender la interacción entre incendios de copa y de superficie.

En un comienzo, NEXUS se trataba de una hoja Excel (figura 2.16) pero en el año 2003 se convirtió en un programa por sí mismo. Está orientado a usuarios familiarizados con BEHAVE o BehavePlus, usuarios que posean conocimiento sobre comportamiento del fuego y sobre modelos de fuego de copas.

La versión 2.0 de NEXUS es una aplicación de análisis de riesgo de fuegos de copa, que se puede utilizar para comparar el fuego de copa potencial para diferentes lugares, y comparar los efectos de distintos tratamientos de combustibles en fuegos de copa.

salidas de este simulador para la toma de decisiones.



Figura 2.17: FARSITE: *Fire Area Simulator*.

2.4.9. FOFEM

FOFEM (*First Order Fire Effects Model* [37]) modela los efectos directos o indirectos que son consecuencia de los incendios. Establece los efectos secundarios de los incendios (mortalidad de árboles, consumición de combustible, humo y calor del suelo).

Es útil para planificar de mejor forma los incendios prescritos ya que permite conocer cuáles son los efectos y consecuencias de la realización de dichas quemaduras. Permite conocer los valores de humedad del combustible, determinar la superficie que debería ser quemada un día determinado para no superar los límites de emisión de gases, evaluar efectos de fuegos, comparar resultados de distintas acciones, etc.

Los modelos *First Order Fire Effect* son los que conciernen las consecuencias inmediatas y también indirectas del fuego. Estos efectos forman una importante base para predicción de efectos secundarios tales como regeneración de árboles, cambios en la productividad del lugar, etc., pero estos efectos a largo plazo generalmente implican la interacción con diversas variables (clima, uso de animales, insectos y enfermedades) y éstos no son predecidos por este sistema.

2.4.10. FireStation

FireStation ([29]) es un simulador de incendios forestales que trabaja sobre topografía compleja. El simulador implementa un modelo semi-empírico (basados en correlaciones empíricas, obtenidas a partir de datos experimentales), que toma como entrada la pendiente del terreno, parámetros que

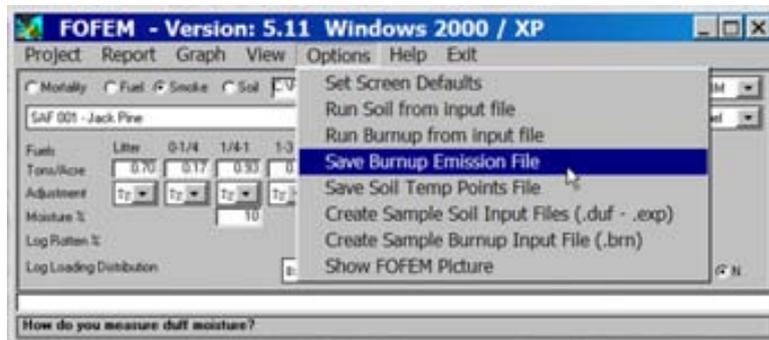


Figura 2.18: Una ventana de la versión para Windows XP de FOFEM.

describen las propiedades del combustible, y la dirección y velocidad del viento. La evolución del fuego se propaga utilizando el modelo elíptico de propagación.

Además, este sistema incorpora dos modelos de viento, que predicen su velocidad y su dirección, basándose en observaciones locales tomadas de estaciones meteorológicas. Dicha incorporación es una importante contribución para mejorar la calidad de las simulaciones.

Este simulador tiene interfaz gráfica, facilitando su utilización y mejorando la legibilidad de sus salidas. Está desarrollado con *Microstation*, ambiente que provee interfaces amigables y útiles herramientas de desarrollo, además ofrece herramientas gráficas en tres dimensiones para la visualización de mapas u otros procesos de visualización.

Este simulador está implementado con autómatas celulares, pero en este caso, el terreno es dividido en celdas pero no sólo se consideran 8 vecinos, sino que se consideran 16 vecinos tal como muestra la figura 2.19 (a) ([29]). La división del terreno en celdas, tiende a lograr predicciones donde el fuego avanza menos que lo que avanza en la realidad (figura 2.19 (b)). Esto se puede mejorar en parte aumentando el número de celdas vecinas, pero esta aproximación tiene como contrapartida que, en cada paso, se propaga el fuego a celdas espacialmente más distantes, incrementando el error en el caso de que el terreno no tenga características uniformes.

Además de las simulaciones de propagación del fuego a nivel local, este sistema incorpora un modelo de índice de riesgo a nivel regional y nacional.

Incluye dos modelos para simular el viento, uno de los cuales es utilizado cuando se necesitan resultados rápidos: cuando existen restricciones de tiempo real se utiliza NUATMOS y en el caso de que se utilice el sistema para, por ejemplo, tareas de planificación, donde el tiempo de respuesta

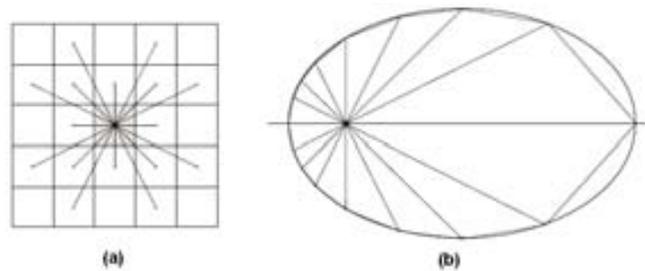


Figura 2.19: FireStation: (a) Disposición de las 16 celdas consideradas celdas vecinas. (b) Comparación entre la forma ideal del fuego y la obtenida en este simulador

no es primordial, se pueden realizar cálculos más largos y, en este caso, se utiliza CANYON.

El modelo NUATMOS toma como entrada velocidad y dirección del viento en distintos puntos discretizados del terreno, luego se computan los datos del viento para todos los puntos del espacio (extrapolando e interpolando los datos conocidos), posteriormente esta velocidad es ajustada siguiendo un método de análisis de variación.

A su vez, CANYON utiliza coordenadas en tres dimensiones, teniéndose en cuenta velocidad y presión. Se modelan las turbulencias, los efectos del terreno (o de la vegetación), y aunque CANYON tiene en cuenta los efectos termales, estos no se tienen en cuenta en simulación de incendios forestales (se considera el viento a media altura de llama).

2.4.11. *fireLib*

fireLib es una librería desarrollada en C para la simulación de incendios forestales basados en los algoritmos de BEHAVE para predecir la propagación del fuego en 2 dimensiones. Calcula el índice de propagación, intensidad, longitud de llama y altura de brasa para incendios de superficie.

fireLib contiene 13 funciones, pero con 4 de ellas es suficiente para crear un simulador simple y funcional. Este simulador recibe como entradas el mapa con el frente de fuego inicial (lo llamamos instante t_i) y los parámetros de entrada que aportan toda la información necesaria para la predicción, obteniéndose como resultado el mapa del incendio simulado para un instante de tiempo posterior a t_i (instante t_{i+1} , figura 2.20).

Los parámetros que recibe el simulador son:

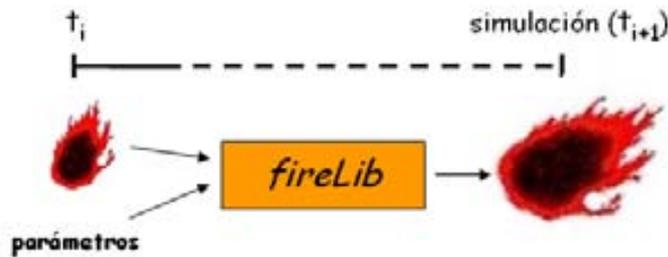


Figura 2.20: *fireLib*: entradas y salidas.

- Número de modelo de combustible (uno de los 13 modelos definidos anteriormente en este mismo capítulo).
- Contenido de humedad del combustible muerto (pasadas 1 hora, 10 horas y 100 horas).
- Contenido de humedad del combustible herbáceo vivo.
- Velocidad y dirección del viento.
- Pendiente del terreno.
- Orientación de la pendiente.

Este simulador utiliza la técnica de simulación autómatas celulares para la propagación del fuego. Los mapas representan el terreno donde se desarrolla el incendio y se dividen en celdas cuadradas, donde cada celda es etiquetada con el instante de tiempo en que el fuego alcanza el centro de la misma (un valor nulo si la celda no ha sido quemada).

Si el tipo de combustible encontrado en el terreno no concuerda con ninguno de los 13 modelos de combustible propuestos en [5], *fireLib* permite crear un tipo de combustible nuevo y definir todas sus características para que sea más representativo del tipo de combustible real. Para esto se debe especificar un catálogo de combustible nuevo, el modelo de combustible, y las partículas que conforman dicho combustible. Existen distintos métodos para facilitar la creación de un modelo nuevo, la especificación de sus características y su posterior utilización. En este caso, hay que especificar todos los datos que se encuentran ya preestablecidos en los 13 modelos de combustibles mencionados anteriormente.

En el momento de una simulación, el simulador itera sobre cada una de las celdas del mapa propagando el fuego de una celda a otra hasta que

la intensidad del fuego no es suficiente como para propagarse a una celda vecina o se llega a un borde del terreno.

Básicamente, el proceso de simulación puede resumirse en la descripción de 4 pasos [9]. Estos pasos se realizan por cada celda del mapa desde la cual sea posible la propagación del fuego hacia sus celdas vecinas. Los pasos que se computan se resumen en la tabla 2.1:

Paso	Entradas	Salidas
1. Combustible	Características del combustible	Características generales del combustible, carga, densidad, altura, etc.
2. Humedad	Humedad del combustible	Índice de propagación sin viento ni pendiente. Intensidad de reacción, humedad de extinción del combustible vivo, etc
3. Viento y Pendiente	Dirección y velocidad del viento, dirección e inclinación del terreno	Índice de máxima propagación y su dirección
4. Dirección	Dirección de máxima propagación	Intensidad de propagación, longitud de llama, altura de brasas para cada una de las 8 direcciones principales.

Tabla 2.1: Pasos de la simulación utilizando *fireLib*.

1. En el primer paso, se establecen y se asignan los valores relacionados con el tipo de combustible. A pesar de que en la naturaleza se pueden tener distintos tipos de combustibles en el mismo área, en este simulador el tipo de combustible se considera uniforme tanto espacial como temporalmente (cierto grado de variabilidad ya está considerada en los modelos de combustibles utilizados).
2. En el segundo paso, se tiene como salida el índice de propagación sin considerar todavía la influencia del viento y de la pendiente. En este paso, lo que determina la intensidad de propagación es el tipo de combustible y el contenido de humedad de dicho combustible. Los datos del combustible se obtuvieron en el paso anterior y los contenidos de humedades son las entradas de esta función.
3. En el tercer paso, se obtiene el índice de máxima propagación (ROS) y la dirección en la que ocurre esta máxima propagación. Para obtener estos 2 datos, es necesario considerar la intensidad y la dirección del viento como así también, la inclinación y la dirección de la pendiente. Aunque la pendiente se considera también uniforme en todo el terreno y a lo largo de toda la duración de la simulación, recién se utiliza en este paso porque el modelo de propagación propuesto por Rothermel

modela el efecto producido por la pendiente de la misma forma que el efecto producido por el viento, combinándolos y tratándolos en conjunto para determinar el ángulo en que el fuego alcanza su mayor velocidad de propagación.

4. En el paso 4 se obtiene la intensidad y el índice de propagación para cada una de las 8 direcciones principales (N, NO, O, SO, S, SE, E y NE) [9]. En este paso se obtiene en qué momento el fuego llega a cada una de las celdas vecinas (si es que llega) dependiendo de la intensidad de propagación, de la dirección de máxima propagación, del crecimiento elíptico del modelo y la dirección y distancia en que se encuentra el fuego con cada una de las celdas vecinas.

Dado que este es el simulador utilizado en el presente trabajo, en capítulos posteriores se volverán a mencionar ciertas características del mismo.

2.4.12. Resumen

La tabla 2.2 muestra, de forma resumida, las principales características de la mayoría de simuladores descritos anteriormente. Para un análisis más exhaustivo de los mismos se puede consultar el estudio comparativo desarrollado por Dorothy Albright y Bernard N. Meisner en [3].

Como se puede notar, una propiedad en común de todos estos simuladores es la importancia de sus datos de entrada. Esta importancia radica en que las propiedades del entorno donde se desarrolla el fuego determinará su forma, velocidad, etc., todos los aspectos que los simuladores intentan calcular. Entonces, si las condiciones ambientales se encuentran mal descritas, la simulación no será útil ya que no resultarán valores acordes al entorno donde sucede el fuego.

Además, la mayoría de estos simuladores necesitan múltiples datos de entrada, algunos con formatos específicos, pero podemos ver que estos datos son una abstracción que representa un aspecto de la realidad. Estas abstracciones muchas veces las asumimos como válidas, como completas, pero que distan mucho de serlo.

Entonces, al entender que los datos de entrada en estos sistemas son una fuente de incertidumbre, optamos por elegir una metodología de trabajo que no intenta mejorar modelos ni técnicas de simulación, sino que, asumiendo esta fuente de incertidumbre, intentaremos hallar los mejores parámetros de entrada para que el ambiente donde se desarrolla el fuego resulte bien representado y las simulaciones usando estos sistemas aumenten su calidad.

Es destacable que esta metodología que intenta mejorar la calidad de los datos de entrada, es aplicable a cualquier simulador (los componentes del

sistema propuesto intentan ser lo más independientes posible). Obviamente, cambiar el simulador implica cambiar los datos de entrada y de salida, pero la metodología propuesta es fácilmente adaptable a las características de los simuladores disponibles en cada momento.

Simulador	Componentes		Idealto para	Entradas		Salidas	Plataforma y software
	Modelo de predicción	Técnica de Predicción		GIS	Adicional		
Clarke Cellular Automation System (Clarke et al 1993)	Probabilístico	Autómata Celular	Simular riesgo de ignición, medición y patrones de ignición	Vegetación, elevación, humedad de combustible	Temperatura, humedad relativa, velocidad y dirección del viento	mapa de riesgo	UNIX con compilador C e Interfaz XWindows
DYNAMIRE (Kalabokidis et al. 1991; Hay 1991)	Físico-estadístico	Autómata Celular	Simulación de fuegos superficiales de intensidad baja a moderada	Tipo de combustible, elevación, pendiente, orientación	temperatura, humedad relativa, humedad del combustible, dirección y velocidad del viento	mapas del perímetro del fuego, intensidad del fuego, media del índice de propagación	PC con MS-DOS y pMAP
EMBYR (Hargrove et al. 1995)	Probabilístico	Bond Percolation	Simular patrones de ignición	Vegetación clasificada por especies y edad	Humedad del combustible, dirección y velocidad del viento	Mapa de la ignición final	UNIX con compilador FORTRAN
FARSITE (Finney 1993)	Físico-estadístico	Propagación elíptica	Simulación de la propagación y de comportamiento de fuegos	Tipos de combustible, elevación, pendiente, orientación, cobertura de copas	Temperatura, humedad relativa, dirección y velocidad del viento, características de las copas	mapa de comportamiento del fuego, y perímetro del fuego	PC con Windows NT o 95
FIREMAP (BaU y Gwertin 1992)	Físico-estadístico	Autómata Celular	Simulación de fuegos superficiales de intensidad baja a moderada	Tipo de combustible, elevación pendiente y orientación	Temperatura, humedad relativa, humedad del combustible, dirección y velocidad del viento	Mapas de: propagación, intensidad de la línea del fuego, longitud de llama, calor por unidad de área, intensidad de reacción, perímetro del fuego	UNIX con PROMAP
WILDFIRE (Walker 1993)	Físico-estadístico	Propagación elíptica	Simulación de fuegos superficiales de intensidad baja a moderada	Tipo de combustible, elevación	Dirección y velocidad del viento	Mapas de: perímetro del fuego e intensidad del fuego	PC con MS-DOS
FireStation (Lopes, Cruz, Viegas 2002)	Físico-estadístico	Autómata Celular	Simulación de fuegos superficiales de intensidad baja a moderada	Tipo de combustible, elevación	Temperatura, humedad relativa, humedad del combustible, lecturas del viento de estaciones meteorológicas	Mapas de: dirección de viento, velocidad del viento, índice de propagación, intensidad del fuego, altura de la llama, calor por unidad de área, intensidad de reacción, ,perímetro del fuego	PC con Windows, Microstation software

Tabla 2.2: Resumen de las características de los simuladores expuestos.

Capítulo 3

Predicción de la Evolución de los Incendios Forestales Guiada por los Datos

Los incendios forestales son el resultado de la conjunción de diferentes aspectos físicos, ambientales, climáticos, topográficos y agregándose lamentablemente, factores humanos también. En las simulaciones se deben tener en cuenta todos estos factores, para realmente tener como resultado una simulación que represente el avance del fuego de una forma correcta.

Desafortunadamente, son múltiples los problemas encontrados en las simulaciones de incendios forestales, resultando, la mayoría de las veces, en simulaciones imprecisas. A partir de esta premisa, se desarrollará en este capítulo distintas aproximaciones para lograr mejorar la calidad de las simulaciones.

Estos métodos no intentan modificar ni mejorar un simulador, sino que intentan mejorar la calidad de los datos de entrada en los que se basa cada simulación con el objetivo de generar simulaciones más próximas a la evolución real del fuego. El *framework* utilizado en este trabajo es independiente del simulador utilizado, por lo que se puede cambiar y utilizar el simulador que más convenga en cada momento (haciendo los cambios pertinentes en las entradas y salidas para la adaptación a los formatos requeridos por cada simulador).

3.1. Predicción Clásica

La mayoría de los simuladores de incendios forestales utilizan la descripción del entorno del fuego, esto es, datos de la topografía, del clima, de la vegetación, etc., junto con la descripción del frente del fuego inicial para poder generar como salida la evolución del frente del fuego en un instante de tiempo posterior. De este modo, una predicción de propagación de un incendio forestal puede ser fácilmente obtenida utilizando un simulador. La figura 3.1 muestra, esquemáticamente, cómo sería una predicción clásica utilizando un simulador cualquiera.

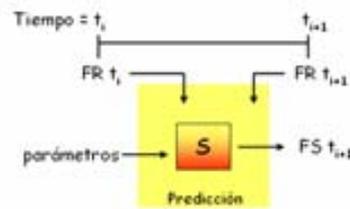


Figura 3.1: Método de predicción clásica.

En este tipo de predicción, se utiliza el estado de la línea de fuego inicial (FRt_i , Fuego Real en el instante t_i) como entrada del simulador (en la figura, S representa al Simulador), más el conjunto de parámetros que, dependiendo del simulador tienen menor o mayor nivel de detalle, describen el entorno donde sucede el fuego. Utilizando entonces estas entradas en el simulador, se obtiene la línea de fuego en un instante de tiempo posterior ($FS t_{i+1}$, Fuego Simulado en el instante t_{i+1}). Se puede apreciar que este proceso de predicción es muy simple de implementar y de utilizar. Además, requiere muy pocos recursos de procesamiento, ya que sólo es necesario ejecutar una simulación para obtener la predicción. Los datos de entrada dependen del simulador, pero sólo se necesitaría un único conjunto de datos de entrada para realizar dicha predicción.

Pero en la realidad, se reconoce que la salida de los simuladores no suele ser precisa debido a diversos motivos. En la siguiente sección se estudiarán distintas fuentes de error que determinan la poca calidad en las simulaciones de incendios forestales utilizando los simuladores disponibles en la actualidad.

3.2. Fuentes de Error de la Predicción Clásica

Existen muchas fuentes de error cuando se trata de simular el progreso del fuego en incendios forestales. En la bibliografía es habitual encontrar problemas importantes que se tratan de superar de distintas formas. Sólo a modo de ejemplo, describiremos algunos de dichos problemas mencionados en los trabajos [1], [10] y [12].

En el desarrollo del trabajo [1] se mencionan diversas fuentes de incertidumbre en la predicción de incendios forestales utilizando simuladores. A continuación se presentarán algunas de las mismas.

Una fuente de error son los modelos mismos. Los modelos intentan simplificar los fenómenos que modelan sin perder las características principales de dichos fenómenos. Modelos muy simples pueden perder precisión, mientras que modelos demasiado complicados pueden ser impracticables. Entonces, es importante entender cuáles son los aspectos del sistema que el modelo pretende describir, y cuáles son las limitaciones del modelo resultado de la simplificación.

Otro problema son las limitaciones matemáticas: las ecuaciones matemáticas de un modelo son resueltas, habitualmente, mediante soluciones numéricas. Usualmente, los modelos matemáticos contienen complicadas ecuaciones diferenciales que necesitan aproximaciones numéricas para ser resueltas. Estos modelos entonces, tienen errores de aproximación. La discrepancia entre la respuesta real y la respuesta obtenida con un cálculo práctico persiste independientemente de la computadora utilizada.

Otro problema radica en las limitaciones que implica la traducción de los métodos numéricos a un lenguaje de programación. Estos lenguajes y el procesador utilizado tienen cada uno sus propios límites en la representación de datos, en particular en la precisión de la representación numérica. En un ordenador, los números se almacenan en espacios de un número limitado de bits, lo que significa una representación que no es real. La precisión depende del sistema numérico utilizado y de la cantidad de bits utilizados para el almacenamiento.

En este mismo trabajo ([1]), también se distinguen problemas que tienen relación con la precisión de los datos de entrada de los simuladores, por ejemplo, muchos parámetros no pueden ser medidos directamente, sino que se deben estimar de otras medidas (indirectas). Otros parámetros pueden ser medidos en ciertos puntos y luego los demás puntos deben ser interpolados a partir de los primeros.

Otra importante fuente de error es el viento, este parámetro es totalmente dinámico, cambiante en el tiempo y más aún en el entorno de un incendio forestal, lo que hace necesario el uso de algún modelo que permita predecir el comportamiento del viento.

Otra fuente de error son los mapas utilizados como entradas. En el caso de autómatas celulares, estos mapas están compuestos de celdas, entonces, los datos de cada celda son en sí promedios de las características de las celdas, valores predominantes, etc. Para aumentar la precisión en la representación de estos mapas, es posible disminuir el tamaño de las celdas, lo que implica aumentar en gran medida el costo de procesamiento. Por otro lado, los mapas que representan características del terreno son actualizados periódicamente, por lo que es normal trabajar con datos medidos en el pasado los que, tal vez, sean datos obsoletos.

Por otro lado, en el trabajo [12], también se enumeran algunos problemas con respecto al uso de simuladores, de sus entradas y de sus salidas. En este caso, se hace referencia a que es frecuente que las herramientas de simulación no son usadas ni interpretadas de forma adecuada. El uso inexperto, incompleto e interpretaciones sencillas de los resultados, comúnmente derivan en evaluaciones no reales, conllevando esto a una progresiva falta de confianza sobre estas herramientas informáticas.

Otra fuente de error descrita en este trabajo, está relacionada con la complejidad de la información geográfica que se debe manejar. Actualmente existe mucha información de este tipo y están disponibles distintos sistemas de información geográfica (GIS) que intentan facilitar el uso de dicha información. Pero esto es sólo una representación de la realidad, con muchas limitaciones que uno debe considerar en el momento de su utilización.

Además, muchos de los simuladores actuales utilizan mapas que dividen el terreno en celdas cuadradas. Esta solución tiene problemas debido a la interpretación discretizada de la realidad, y la segmentación del espacio en formas geométricas (formato de la celda, muchas veces cuadradas), lo que implica cuantificar el espacio de solución. Esto puede llevar a errores cuando se calculan distancias, cuando se calcula la forma de un fuego utilizando autómatas celulares, etc.

Cabe mencionar que el uso de archivos *raster* no está limitado a simuladores que utilizan el modelo de simulación autómata celular. Por ejemplo FARSITE [21], que utiliza el modelo de crecimiento elíptico del fuego, utiliza ficheros *raster* para la información referente a la topografía (elevación, pendiente, orientación, modelo de vegetación y cobertura de copa [2]). Un fichero *raster*, utiliza una malla de celdas cuadradas para representar al terreno.

También en [10] se enumeran diversas fuentes de error en el uso de simuladores de incendios forestales. En este trabajo se mencionan los problemas que significa la falta de precisión en los parámetros de entrada a partir de la naturaleza de los mismos. La mayoría de los simuladores utilizan diversos parámetros, algunos de los cuales son considerados estáticos (como la topografía del suelo), otros, que varían lentamente (como por ejemplo el

tipo de vegetación que actuará de combustible) y otros que varían de forma muy dinámica (como la velocidad y la dirección del viento). Entonces, la precisión de estos datos de entrada es indispensable para las simulaciones, y hay casos en que ni siquiera se pueden llegar a tomar medidas de los mismos.

En este trabajo también se valoran los límites que implican las soluciones numéricas a distintos modelos y los límites que también derivan de la abstracción de una parte de la realidad (modelo).

Resumiendo lo anteriormente expuesto, se puede concluir que existen principalmente dos tipos de problemas que generan la mayor fuente de errores en la predicción de incendios forestales:

- Las abstracciones que todo modelo implica, así como su implementación, y
- la falta de precisión en los parámetros de entrada.

Utilizando un simulador que implementa el modelo matemático de Rothermel, y asumiendo las limitaciones del modelo y de su implementación, en este trabajo se proponen distintos métodos con el objetivo de disminuir la incertidumbre que implican los parámetros de entrada.

3.3. Predicción en Dos Etapas

Al principio de este capítulo se mostró cómo sería una predicción clásica para incendios forestales utilizando un simulador cualquiera. Este método es muy simple, muy poco costoso en requerimientos de cantidad y formato en los datos de entrada (un único conjunto de valores), como así también de requerimiento de cómputo (tiempo de CPU), ya que se ejecuta una única simulación. Entonces, como es posible ver en la figura 3.1, el resultado de la predicción depende del resultado de una única simulación utilizando un único conjunto de parámetros de entrada.

Así, toda la calidad de la predicción depende de la calidad de este único conjunto de parámetros de entrada. Teniendo en cuenta todos los problemas relacionados con la imprecisión de los parámetros de entrada, y teniendo en cuenta también, la dificultad de la disposición de los valores correctos para estos parámetros, podemos ver que es muy difícil, hasta incluso puede resultar imposible, tener una buena predicción utilizando este esquema.

Para disminuir este problema y mejorar la calidad de las predicciones, en este trabajo se utiliza el esquema definido en [1] donde se agrega una etapa previa a la fase de predicción. En esta nueva etapa se intenta mejorar la calidad de los parámetros de entrada utilizados en la predicción. Este

esquema es ilustrado en la figura 3.2. En dicha figura se pueden ver las entradas y salidas para cada una de las etapas que componen este esquema.

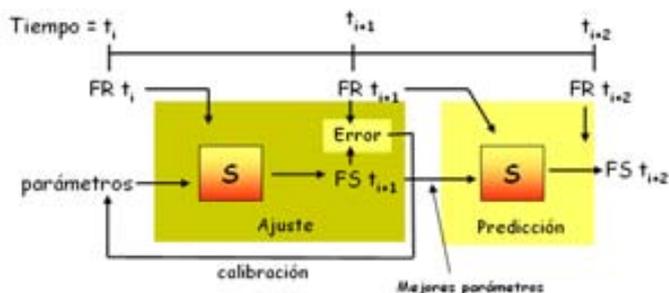


Figura 3.2: Método de predicción en dos etapas.

En las próximas secciones se describen detalladamente estas dos etapas que forman el método aquí utilizado. En el presente trabajo se incluyen mejoras al método de predicción basados en dos etapas, las cuales se expondrán en el siguiente capítulo.

3.3.1. Etapa de Ajuste

La etapa de ajuste se ejecuta previamente a la etapa de predicción e intenta superar el principal problema que tiene la predicción clásica: mejorar la calidad de los parámetros de entrada utilizados para la predicción.

En esta etapa se utilizan múltiples combinaciones de valores para los parámetros de entrada, y en un proceso de retroalimentación se mejoran los mismos con la intención de mejorar las simulaciones obtenidas.

En esta aplicación, se utilizan diversos parámetros, y cada uno de ellos tiene su propio rango de variación. Entonces, teniendo en cuenta todas las posibles combinaciones de estos valores para los parámetros de entrada, se puede concluir que nos enfrentamos a un problema de una búsqueda dentro de un espacio de búsqueda muy grande. Este tipo de problemas implica que dicha búsqueda se puede convertir en un cuello de botella de la aplicación entera.

En este trabajo se propone realizar una búsqueda eficiente en dicho espacio de búsqueda. Esta búsqueda está contenida en la etapa de ajuste (ya que se intenta ajustar, sintonizar los parámetros de entrada).

Específicamente, en nuestra aplicación y en función del simulador utilizado, consideramos los valores de 8 parámetros de entrada: tipo de combustible, pendiente del terreno, humedad del combustible vivo, humedades del

combustible muerto para tres momentos diferentes y velocidad y dirección del viento.

Para el tipo de combustible se utiliza uno de los 13 modelos definidos como estándares de vegetación ([2]) comentados ya en un capítulo previo. Estos modelos estándares evitan la especificación para cada simulación de todas las propiedades del combustible (pues se encuentran predefinidas en dichos modelos). Así, en nuestra aplicación se utilizará como entrada un número del 1 al 13 el cual especifica un modelo en particular.

La pendiente se describe con dos valores, uno define los grados de inclinación (en radianes para el simulador *fireLib*) y el otro, la orientación de dicha pendiente. Estas dos propiedades sirven para describir la topografía encontrada en el terreno.

Por otro lado, el viento es caracterizado por su dirección y por su velocidad. Cabe mencionar que en estudios pasados ([1]), se comprobó que una vez superada cierta velocidad del viento la influencia en la propagación del fuego es siempre la misma, lo que acota superiormente el rango de variación del viento para nuestro sistema. Más adelante veremos la importancia de este parámetro en nuestra aplicación.

Por otro lado, el contenido de humedad del combustible vivo debe ser secado por el calor del fuego para que el combustible pueda arder. Por lo que esta humedad influenciará sobre todo en la velocidad de propagación del fuego.

Existen 3 parámetros más que definen la humedad del combustible muerto: pasadas 1 hora, 10 horas y 100 horas. Además, existe un parámetro definido dentro de los modelos estándares de combustible que es la humedad de extinción. Esta humedad marca el límite para el cual el fuego no propaga, haciendo que se extinga. Entonces, el estudio de las humedades de extinción de cada uno de los modelos, nos ha permitido definir el límite superior de variación de la humedad del combustible muerto pasado 1 hora, pues cuando esta humedad sobrepasaba la humedad de extinción el fuego no propagaba.

Cabe destacar en este punto que existen otros parámetros de entrada que ya están predefinidos dentro de los modelos de combustible estándares.

De todos estos parámetros de entrada que utiliza el simulador *fireLib*, se puede observar que las características de la pendiente (inclinación y orientación) y el modelo de vegetación, son los parámetros más estáticos del entorno, los que varían menos en tiempo y espacio en el microclima generado por el incendio. Además, se los puede observar a simple vista, por lo que su determinación puede ser más simple que la de los demás parámetros.

Dadas estas características, hemos decidido contemplar tanto las características de la pendiente como el tipo de vegetación como características conocidas y constantes para nuestro sistema. De esta forma, estos paráme-

tros se conocen antes de realizar las simulaciones en las etapas de calibración y de predicción.

De esta forma, durante la etapa de ajuste se buscará la mejor combinación de valores para los restantes 6 parámetros: dirección y velocidad del viento, contenido de humedad del combustible muerto (1hr, 10hrs y 100hrs) y contenido de humedad del combustible vivo. Entonces, se considerarán la combinación de todos los posibles valores que pueden adoptar estos 6 parámetros (variando cada uno en su rango de valores válidos).

Además, para realizar la búsqueda de una forma eficiente se utilizará un algoritmo genético que recorrerá el espacio de búsqueda formado por los valores de entrada para los parámetros e intentará buscar la combinación que minimice el error producido en las simulaciones.

De esta forma estamos supliendo la deficiencia vista en el método clásico de predicción, donde se consideraba sólo un conjunto de valores para los parámetros de entrada y para los cuales no existía ningún tipo de preprocesamiento.

Como se puede ver en la figura 3.2 la salida del simulador (FSt_{i+1}) es comparada con la línea de fuego real para ese mismo instante de tiempo ($F Rt_{i+1}$). El resultado de esta comparación es lo que se tiene en cuenta en este proceso de retroalimentación: el error cometido en la simulación realimentará al sistema para mejorar los valores de los parámetros de entrada.

La salida de esta etapa de ajuste es un único individuo que contiene los valores para los parámetros de entrada que mostraron la mejor simulación en toda la evolución del algoritmo genético.

Además, en este trabajo se guiará dinámicamente este proceso de búsqueda. Se utilizará conocimiento para inyectar mejores valores para los parámetros de entrada, guiando la evolución del algoritmo genético. De esta forma, intentaremos guiar al algoritmo hacia zonas con individuos “de buena calidad”, convergiendo así de una forma más rápida a buenas soluciones y evitando evaluar individuos que no darán buenos resultados por tener valores que no representan de forma correcta el entorno medioambiental donde se desarrolla el fuego.

3.3.2. Etapa de Predicción

Como se puede ver en la figura 3.2, la salida de la etapa de ajuste es un único conjunto de parámetros. Este único conjunto de parámetros cumple con la condición de que es el mejor hallado en la etapa de ajuste.

Si los valores de dicho individuo han sido los mejores para la etapa de ajuste quiere decir que son valores para los parámetros que describen de forma correcta el escenario donde se desarrolla el fuego (para el instante de tiempo t_{i+1}). Luego, se asume que estos parámetros seguirán describiendo

de forma correcta el ambiente del fuego para el próximo instante de tiempo, de forma que la predicción para el instante t_{i+2} se basa en parámetros correctos (característica conocida como límite de predictabilidad, [1]).

Lo anteriormente expuesto es válido aunque se pueden tener en cuenta ciertas consideraciones. Por un lado, se asume que los factores ambientales permanecerán constantes (o, al menos, similares) de un instante al próximo, característica que puede suceder como no. Otro factor es que no interesa qué valores adoptaron cada uno de los parámetros de entrada, sólo importa que hayan resultado en una buena simulación.

A pesar de estas 2 características, durante la experimentación se mostrará que no invalidan la hipótesis de trabajo que este esquema en dos etapas agrega.

Entonces, en la etapa de predicción se utiliza como entrada el mejor conjunto de parámetros de la etapa de ajuste y el estado del fuego en el instante de tiempo t_{i+1} para predecir que sucederá en el instante de tiempo siguiente. Sólo en el caso de que se disponga de la información, se puede comparar el mapa predicho (FSt_{i+2}) con el mapa real para ese instante ($F Rt_{i+2}$).

Una vez analizados los componentes de este esquema, en el próximo capítulo se detallarán la implementación del mismo, y la guía mediante la inyección dinámica de datos de entrada que constituyen la principal aportación de este trabajo.

3.4. Fuentes de Error de la Predicción en Dos Etapas

En trabajos anteriores ([1], [10]), se observa y estudia los errores de la predicción clásica de incendios forestales y se propone una predicción en dos etapas para mejorar la calidad de las predicciones. Un error de la predicción clásica radica en que al basarse toda la predicción en un único conjunto de parámetros de entrada, si no se dispone de los valores correctos, la predicción puede ser errónea.

Considerando esta fuente de error, se propuso agregar una etapa previa (etapa de ajuste o calibrado) donde se mejoran los valores para estos parámetros de entrada y se los utiliza en un tiempo posterior para la predicción. Esta alternativa, se basa en la hipótesis de que las condiciones medioambientales se mantendrán constantes durante el tiempo que impliquen dichas etapas.

Pero a esta mejora, se le encuentran algunos posibles errores:

- Esta predicción en dos etapas realiza una búsqueda “*normal*” (sin

ningún tipo de guía o pre conocimiento) en un extenso espacio de búsqueda, lo cual requiere tiempo y tal vez la búsqueda pueda resultar ineficiente.

- Debido a la hipótesis de trabajo, durante la etapa de calibrado no se tiene en cuenta posibles variaciones en las condiciones ambientales, lo que puede influir en la etapa de predicción por usar los valores hallados para un instante de tiempo posterior.

Considerando estas posibles fuentes de error del método de predicción en dos etapas propuesto, las principales aportaciones de esta Tesis son un paso para solucionar estas carencias ya que:

1. se elimina parte de la aleatoriedad guiando la etapa de calibrado hacia espacios de búsqueda más óptimos en base al conocimiento inyectado.
2. se tiene en cuenta posibles variaciones en las condiciones ambientales ya que se inyecta información en la etapa de calibrado proveniente del análisis de la evolución real del incendio, el cual es un reflejo de las condiciones ambientales reales.

El próximo capítulo presentará las mejoras propuestas a este método de predicción de dos etapas que son la principal contribución de este trabajo.

Capítulo 4

Algoritmo Genético Guiado Dinámicamente por los Datos

Como ya se ha comentado en el capítulo anterior, nuestra aplicación para la predicción de los incendios forestales está compuesta de dos etapas. Durante la etapa de ajuste o sintonización se realiza una búsqueda de valores para los parámetros de entrada del simulador que generen una buena simulación. De esta forma, ajustamos los parámetros de entrada para que sean correctos, disminuyendo de esta forma la incertidumbre y errores que dichos parámetros puedan producir en las simulaciones.

En este capítulo se describen las principales características del algoritmo genético utilizado durante la etapa de ajuste. El objetivo de dicho algoritmo genético es realizar de forma eficiente la búsqueda de valores correctos para los parámetros de entrada en el espacio de búsqueda generado por las distintas combinaciones que estos valores generan.

Se explicarán las principales características de los algoritmos genéticos, mostrando en algunos casos distintas alternativas de implementación, pero dando mayor énfasis a las características de nuestra propia implementación.

¿Qué es un individuo en nuestra aplicación? ¿Cómo está definida la función de *fitness*? y más preguntas que surgen de la implementación de un algoritmo genético para la predicción de incendios forestales son contestadas en este capítulo.

Además, se realiza un análisis de aspectos del paralelismo que definen importantes características de nuestra aplicación paralela.

4.1. Algoritmos Genéticos

Los algoritmos genéticos son algoritmos que transforman un conjunto de individuos (población inicial, donde cada individuo está asociado a su valor de *fitness*), en una nueva población (siguiente generación) usando operaciones acordes al principio creado por Darwin de reproducción y supervivencia de los individuos mejor adaptados a su ambiente y acorde además a las operaciones genéticas de ocurrencia natural (particularmente recombinación o cruzamiento sexual) [28].

Una definición útil para la palabra inglesa *fitness* es simplemente: *la condición de ser adecuado*. Entonces, es directo deducir que esta condición depende del contexto para el que se evalúa *ser adecuado* (o no serlo). Debido a esto, cada problema tendrá su evaluación o función de *fitness* específica (en este capítulo se expondrá cuál es la función de fitness utilizada en esta aplicación).

Los algoritmos genéticos establecen una analogía entre el conjunto de soluciones de un problema, llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena llamada *cromosoma*. Los símbolos que forman la cadena son llamados *genes*. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud (*fitness*). Las siguientes generaciones (nuevos cromosomas), llamada descendencia, se forman utilizando dos operadores, de cruzamiento y de mutación [47]. La figura 4.1 extraída del mismo sitio (ver referencia [47]) grafica los principales pasos de los algoritmos genéticos.

En nuestro caso particular de simulación de incendios forestales, un individuo o cromosoma, estará formado por una cadena de valores, donde cada uno de estos valores (gen), es uno de los parámetros de entrada que utiliza el simulador de incendios forestales. Entonces, un individuo en nuestra aplicación será una asignación de valores específica para:

- Contenido de humedad del combustible muerto 1hr (M1)
- Contenido de humedad del combustible muerto 10hrs (M10)
- Contenido de humedad del combustible muerto 100hrs (M100)
- Contenido de humedad del combustible vivo (M_{herb})
- Dirección del viento
- Velocidad del viento

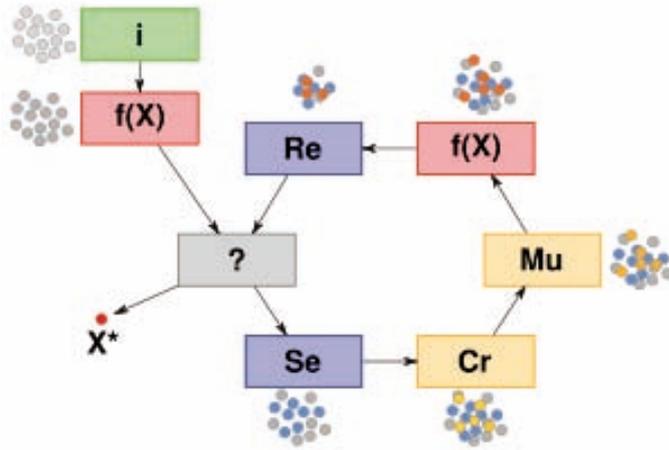


Figura 4.1: Pasos del algoritmo genético: i: inicialización, $f(x)$ evaluación, ?: condición de n, Se: selección, Cr: Cruzamiento, Mu: mutación, Re: reemplazo, X^* : mejor solución.

donde cada uno de estos valores varía en su rango válido de variación. La tabla 4.1 describe los rangos de variación de los parámetros de entrada utilizados en nuestra aplicación. En dicha tabla, cabe destacar que el simulador trabaja con la velocidad del viento como pies por minuto, realizando las conversiones necesarias para convertir las millas por hora en pies por minuto. Como se ha mencionado anteriormente, el modelo de vegetación y las características de la pendiente se asumen como conocidas. De todas formas, en la tabla 4.2 se muestran los valores de variación de dichos parámetros.

Entonces, una vez de nido un cromosoma o individuo en nuestro sistema, se utilizará un algoritmo genético para mejorar la calidad de estos parámetros de entrada, es decir, ajustar estos valores con el objetivo de obtener buenas simulaciones cuando se los compara con el fuego real en un instante de tiempo posterior.

Entonces, una población es un conjunto de individuos cuyo tamaño es un valor predeterminado y configurable al inicio del procesamiento. Durante todas las generaciones, la cantidad de individuos de cada generación se mantiene constante.

Una vez de nido el formato de los individuos, en las próximas secciones se enumerarán las operaciones propias del algoritmo genético.

Parámetro	Valor mínimo	Valor máximo	Unidad
M1	0.0	0.4	lb agua / lb combustible
M10	0.0	1.0	lb agua / lb combustible
M100	0.0	1.0	lb agua / lb combustible
M_{herb}	0.0	4.0	lb agua / lb combustible
Dir. viento	0.0	360.0	grados
Vel. viento	0.0	20.0	millas por hora

Tabla 4.1: Rangos de variación de los valores de los parámetros de entrada.

Parámetro	Valor mínimo	Valor máximo	Unidad
Modelo de vegetación	1	13	
Pendiente	0.0	1.0	radianes
Orientación	0.0	360.0	grados

Tabla 4.2: Rangos de variación de los parámetros de entrada conocidos en nuestra aplicación.

4.1.1. Operadores Genéticos

Un algoritmo genético comienza con una población inicial, la cual, suele tener valores aleatorios asignados a sus genes (salvo excepciones donde se comienza con algún tipo de conocimiento). Esto permite comenzar el algoritmo con diversidad en los genes, lo que permite evaluar distintas zonas del espacio de búsqueda al principio de la evolución.

La evolución de un algoritmo genético se puede ver como una búsqueda inteligente en un espacio de búsqueda determinado. En el caso de que dicho espacio de búsqueda sea grande (como es nuestro caso), un algoritmo de este tipo, ayuda a elegir individuos “buenos” para evaluar, y no cualquier individuo que, debido a sus características, tengan *fitness* muy bajo, no siendo útiles para la aplicación. Entonces, utilizar este tipo de algoritmo permite realizar la búsqueda de forma inteligente y eficiente en nuestro método de dos etapas [1].

Utilizando operadores genéticos se logra que los genes de los individuos con alto *fitness* sobrevivan durante las generaciones, es decir, las características que determinan que un individuo esté bien adaptado a su ambiente sean heredadas de padres a hijos. En nuestro caso, que un individuo esté bien adaptado a su ambiente significa que el resultado de la simulación

sea correcto. En los próximos párrafos se explicará cómo se logra esto.

Los operadores genéticos clásicamente utilizados en este tipo de algoritmos son: *selección*, *cruzamiento* y *mutación*. Cada uno de estos operadores tiene distintas variantes, donde, principalmente, la eficiencia de cada variante depende del dominio de ejecución.

Para cada una de las operaciones, se expondrán distintas variantes de las mismas y se especificará la adoptada en este trabajo.

Selección - Elitismo

En la selección se eligen los individuos padres que se utilizarán para formar a los hijos que serán parte de la nueva población. Esta operación se realiza de forma tal que los individuos mejores adaptados al ambiente tienen mayor probabilidad de ser elegidos. Las características de los individuos seleccionados que hacen que dicho individuo esté bien adaptado al ambiente son transmitidas a los hijos.

Existen distintas formas de realizar esta operación, algunas de las técnicas más utilizadas son:

- Basada en el rango: en este esquema se mantiene un porcentaje de la población para la siguiente generación. Se ordenan los individuos por sus *fitness* y los peores M individuos son reemplazados por descendencia de los mejores con algún otro individuo de la población.
- Rueda de ruleta: se crea una *rueda* con todos los individuos de la población, donde cada uno está representado de forma proporcional a su *fitness*. Si estas proporciones hacen que un individuo domine la población, se realiza algún tipo de escalado. Dentro de esta rueda, se eligen 2 individuos y se cruzan. Con esta técnica es posible elegir el mismo individuo como padre (luego de esta operación se realizan otras que garantizan cierto grado de variabilidad en los individuos).
- Selección de torneo: se eligen aleatoriamente S individuos y el que tiene mayor puntuación se reproduce y su descendencia sustituye al que tiene menor puntuación.

En este trabajo se eligió la segunda opción: rueda de ruleta. Entonces, la población es ordenada según el *fitness* de cada individuo y se seleccionan los padres teniendo más probabilidad de ser elegidos los individuos con mayor *fitness*.

A esta operación se le puede agregar la opción llamada “elitismo”, donde los j ($j > 0$) mejores individuos de la población son seleccionados y forman

parte directamente de la nueva generación, sin realizar ninguna otra operación sobre los mismos (cruzamiento o mutación). La figura 4.2 muestra el caso de utilizar un elitismo de 2 individuos.

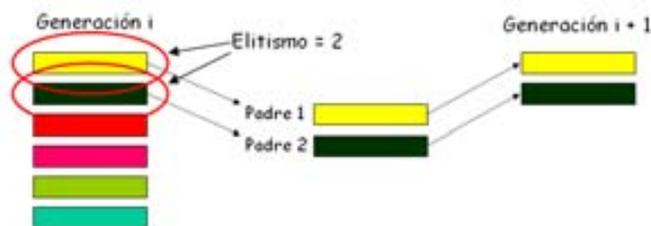


Figura 4.2: Algoritmo genético: elitismo.

La figura 4.3 muestra cómo sería una selección donde se seleccionan el primer y tercer individuo (asumiendo que los individuos están ordenados de mayor *fitness* a menor *fitness* desde arriba hacia abajo).



Figura 4.3: Algoritmo genético: selección.

A continuación se ejemplifica cómo sería un posible procedimiento del algoritmo de la rueda de ruleta para el ejemplo ilustrado anteriormente. Supongamos que el valor de *fitness* está en el rango 0..1 y los individuos de la generación *i* tienen los valores de *fitness* que se muestran en la figura 4.4 (dentro de cada uno de los individuos). Entonces, la rueda de la ruleta tendría un aspecto tal como se muestra en la figura (lado derecho): cada individuo está en la rueda de forma proporcional a su *fitness*.

Una vez que se obtiene esta rueda de ruleta, se elige un número aleatorio en el rango [0..1] y entonces, utilizando la rueda de la ruleta se van acumulando los números que aparecen. Una vez que el número aleatorio es alcanzado o superado, ese individuo es seleccionado.

Para las figuras anteriores, si el primer número aleatorio que sale es el 0.25, entonces, como el primer individuo ya lo supera, ese individuo es el seleccionado (padre 1). Se elije otro número aleatorio (pues se seleccionarán dos padres) y, suponiendo que el número elegido es el 0.61, entonces, se selecciona el individuo rojo, ya que es en donde la acumulación de valores supera el número aleatorio.

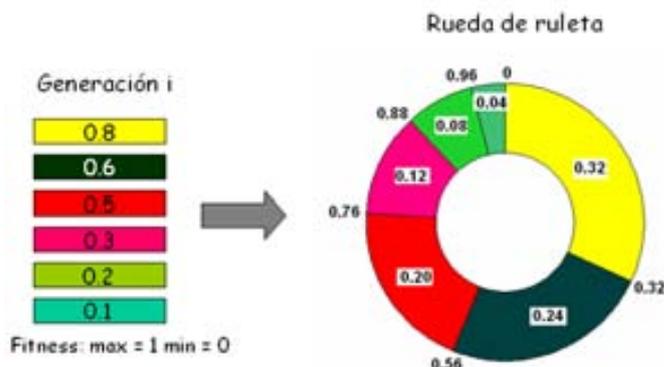


Figura 4.4: Configuración de la rueda de la ruleta para la operación de selección.

Cruzamiento

El cruzamiento consiste en el intercambio de las características de dos individuos, mediante el intercambio de sus partes. Una buena solución está constituida por buenos bloques: la operación de cruzamiento es la encargada de mezclar bloques buenos que se encuentren en diversos progenitores, y que serán los que den a los hijos buen *fitness*. La idea es que sólo los buenos bloques se perpetúen (selección) y poco a poco se generen buenas soluciones.

Algunas técnicas de cruzamiento son:

- Cruzamiento n-puntos: los dos individuos se cortan por n puntos y se intercambian las características situadas entre los puntos. Lo más habitual es usar uno o dos puntos. En la figura 4.5 se muestra un cruzamiento de un punto. El cruzamiento se suele realizar bajo una cierta probabilidad.
- Cruzamiento uniforme: se generan dos patrones aleatorios de ceros y unos y se intercambian las características donde ambos patrones

tienen un uno. O bien se genera un número aleatorio por cada característica y si este número supera una probabilidad se intercambian dichas características de ambos individuos padres.

- Cruzamiento especializado: en algunos casos, aplicar cruzamiento de forma aleatoria da lugar a individuos que tienen características inválidas, en este caso hay que aplicar el cruzamiento de forma que genere siempre individuos o soluciones válidas.

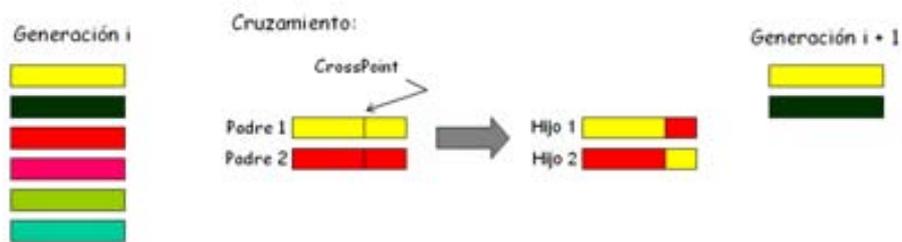


Figura 4.5: Algoritmo genético: cruzamiento.

De estas distintas opciones, en nuestra aplicación se ha optado por el cruzamiento de 1 punto (figura 4.5).

Mutación

En la evolución natural de cualquier especie, una mutación es un fenómeno poco común, siendo muchas veces algo letal, pero en promedio, contribuyen a la diversidad genética de una especie. En un algoritmo genético tiene la misma función y tiene una frecuencia muy baja (como ocurre en la naturaleza).

La mutación es un mecanismo que genera diversidad, y por lo tanto, ayuda cuando un algoritmo genético está estancado, pero también es cierto que, si se abusa de ella, reduce el algoritmo genético a una búsqueda aleatoria. No es conveniente abusar de la mutación en un algoritmo genético habiendo otras formas de generar diversidad, como son aumentar el tamaño de la población o garantizar la aleatoriedad de la población inicial.

La mutación implica variar alguna característica del individuo hijo para así poder obtener individuos con propiedades distintas a la de sus padres. De esta forma es posible evaluar individuos que tengan características diferentes. Esto se suele hacer para evitar caer en máximos o mínimos locales.

La mutación se realiza por cada una de las características de los nuevos

individuos: por cada característica, si se supera cierta probabilidad se modifica dicho valor (figura 4.6).

Para la mutación se especifica una probabilidad y por cada gen se realiza la mutación si y sólo si se supera esta probabilidad. El valor asignado en la mutación está dentro del rango válido para la característica mutada (para evitar generar individuos inválidos).

Entonces, se repiten estas operaciones (selección, cruzamiento y mutación) hasta lograr una población nueva con la misma cantidad de individuos que la generación anterior (como la figura 4.7 muestra).

En este punto es posible resumir que el algoritmo genético itera la aplicación de estas tres operaciones sobre una población hasta un número determinado de veces o hasta alcanzar individuos cuyas simulaciones superen un *fitness* preestablecido.

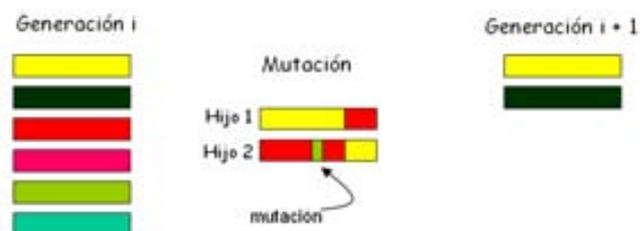


Figura 4.6: Algoritmo genético: mutación.



Figura 4.7: Algoritmo genético: siguiente generación.

Entonces, el algoritmo genético, busca valores que sean máximos o mínimos en todo el espacio de búsqueda (dependiendo de la función de *fitness*).

Si hay máximos o mínimos locales, las operaciones que garantizan algún grado de diversidad intentan evitar que el algoritmo quede en alguno de ellos.

4.1.2. Función de *Fitness*

Como ya se ha mencionado, el algoritmo genético busca maximizar la calidad de los individuos, lo cual depende de la función de *fitness* que se evalúe en cada individuo de la población. La función de *fitness* dependerá del dominio del problema, por lo que será definida por sus características.

En nuestra aplicación, el objetivo es encontrar buenas simulaciones, es decir, simulaciones con buena calidad, que representen el avance real del fuego, tanto en la etapa de ajuste como así también en la etapa de predicción. Entonces, nuestra función de *fitness* debe ser tal que refleje la calidad de las simulaciones.

Todos los mapas utilizados en nuestra aplicación, tanto en la etapa de ajuste como en la de predicción, comparten el mismo formato: son una malla de celdas (en nuestro caso cuadradas), donde cada celda está etiquetada con el instante de tiempo en que el fuego alcanza el centro de la misma. En la aplicación, los mapas que representan el terreno, el mapa inicial, los mapas resultantes de las simulaciones y el mapa real, todos tienen las mismas propiedades, el mismo tamaño y la misma cantidad de filas y de columnas.

Entonces, en nuestra función de error se aprovechará este formato para comparar los mapas celda a celda, calculando la cantidad de celdas distintas (quemada en un mapa y en el otro no) para definir el error cometido en la simulación.

Para poder definir el error, se utilizarán el mapa del fuego inicial (instante t_i), el mapa del incendio real en un instante de tiempo posterior (t_{i+1}), y el mapa de la propagación simulada para el mismo instante de tiempo que el anterior (t_{i+1}).



Figura 4.8: Fuego inicial (*FI*) Fuego Real (*FR*) y Fuego Simulado (*FS*).

Sea FI_{ij} la celda del mapa inicial en la fila i , columna j (la misma forma se utilizará para todos los mapas: FR_{ij} , FS_{ij}) y sea $Ign(FR_{ij})$ una función que es *verdadera* cuando la celda ij del mapa FR está quemada y *falso* caso contrario (la función Ign vale para todos los mapas: FI , FR y FS).

Inicio es la cantidad de celdas que pertenecen al fuego inicial (instante t_i) del cual se comienza la propagación (ecuación 4.1).

Real es la cantidad de celdas quemadas en el mapa real (instante t_{i+1}) (ecuación 4.2).

$$Inicio = \{c_{ij} \mid Ign(FI_{ij}) = verdadero\} \quad (4.1)$$

$$Real = \{c_{ij} \mid Ign(FR_{ij}) = verdadero\} \quad (4.2)$$

Entonces se definen: la *unión* como el número de celdas quemadas en uno o en ambos mapas (ecuación 4.3 y figura 4.9).

$$\cup = \{c_{ij} \mid Ign(FR_{ij}) \vee (Ign(FS_{ij}))\} \quad (4.3)$$

La *intersección* como el número de celdas quemadas en ambos mapas (ecuación 4.4 y figura 4.10).

$$\cap = \{c_{ij} \mid Ign(FR_{ij}) \wedge (Ign(FS_{ij}))\} \quad (4.4)$$



Figura 4.9: Operación de *unión* entre los mapas real y simulado.

Una vez definidas estas cantidades, se define el *error* como:

$$Error = \frac{(\cup - Inicio) - (\cap - Inicio)}{Real - Inicio} \quad (4.5)$$

En esta función no se considera el fuego inicial (se resta siempre *Inicio*) ya que estas celdas que pertenecen a este mapa coinciden en ambos mapas

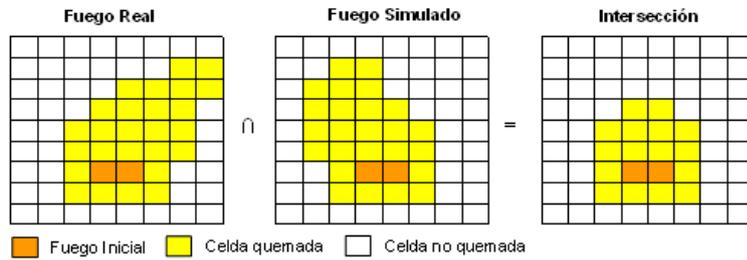


Figura 4.10: Operación de *intersección* entre los mapas real y simulado.

(comienzo de la simulación) y sería contarlas como *aciertos* cuando no son celdas que el simulador haya predicho. De todas formas, en el numerador de la ecuación 4.5 se puede ver que se cancelan estos dos valores *Inicio*, y esto es correcto dado el significado de las cantidades que participan en la función, pero se deja esto en la ecuación para simplificar su comprensión.

La función de *error* determina las celdas equivocadas (numerador en la ecuación 4.5) y estas celdas equivocadas se las hace relativas a la cantidad de celdas quemadas realmente (denominador en la ecuación 4.5). Esta ecuación describe el error cometido en la simulación. El algoritmo evolutivo intenta minimizar dicha función.

En una sección previa, se hace referencia a la función de *fitness normalizada*, la cual depende del significado de dicha función, de la implementación de las operaciones del algoritmo genético como así también de su objetivo: encontrar un máximo, un mínimo, etc.

En nuestro caso, se intenta minimizar el error, pero, como se implementa la selección a través de la rueda de la ruleta (explicada anteriormente), esta selección necesita ordenar la población de mejores individuos a peores (recordemos que los mejores, son los de error mínimo). Esta relación entre *fitness* y error, y la implementación de la selección a través del algoritmo de rueda de la ruleta, determina la ordenación de los individuos de máximo *fitness* a mínimo *fitness*. Esto se implementó definiendo el *error complemento* (E^c): diferencia entre el error de un individuo y el error más grande encontrado en la población.

4.2. Algoritmo Genético Paralelo

Los algoritmos genéticos son altamente paralelizables y, dependiendo de las características del problema y de las características de la arquitectura utilizada, se pueden optar por distintas formas de implementación. A conti-

nuación se mostrarán todos los criterios que han sido discutidos y utilizados para decidir paso a paso las principales características de nuestra aplicación.

Para la resolución de nuestro problema hemos elegido el paradigma de programación paralela *master/worker*, donde se tiene un proceso *master* que distribuye el trabajo entre los procesos *workers*, recolecta los resultados y aplica los operadores genéticos. La figura 4.11 muestra esquemáticamente cómo es la funcionalidad de cada componente de la aplicación y cómo son las comunicaciones entre los mismos:

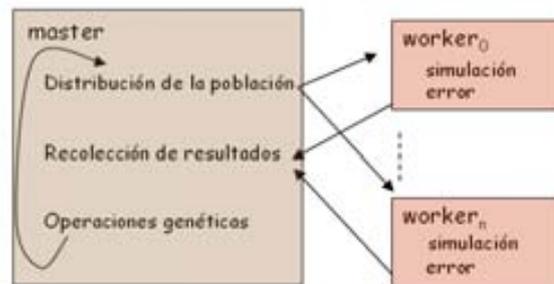


Figura 4.11: Algoritmo genético utilizando el paradigma master/worker

División del Trabajo

Existen distintas formas de paralelizar una aplicación: se puede optar por la división de datos o la división de funcionalidad. En nuestro trabajo se dividirán los datos, de forma tal que la evaluación de las poblaciones se harán en paralelo. Esto es, dada la población inicial, el proceso *master* divide los individuos en grupos (llamados *chunks*) y los distribuye entre los procesos *worker*. Cada *worker* realiza la simulación utilizando el simulador *fireLib* y evalúa la función de error explicada anteriormente.

Las figuras que siguen a continuación, son una ejecución de la aplicación entera, utilizando una población de 512 individuos la cual es evolucionada 5 veces (figura 4.12). Se ha instrumentado el código con las instrucciones de monitorización de la librería *MPE_Open_Graphics* [33]. Para simplificar el ejemplo, en esta ejecución se utilizó sólo 1 *worker*, el cual recibe toda la población en un único *chunk* para su evaluación.

En esta figura se muestran 2 barras horizontales: la superior (etiquetada como “0”) es la ejecución de la tarea *master* y la inferior es la ejecución del único proceso *worker* utilizado (etiquetado como “1”).

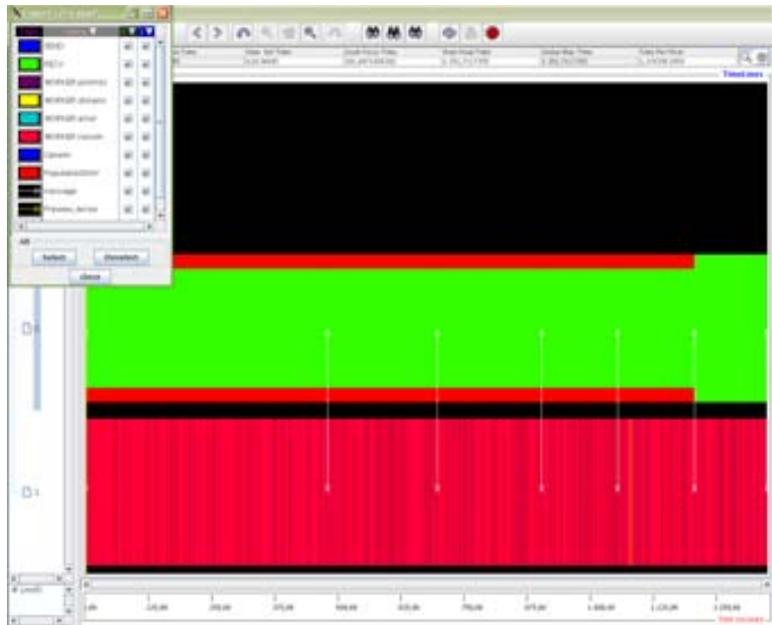


Figura 4.12: Ejecución de la etapa de ajuste, evolucionando la población 5 veces. Se grafican los dos procesos participantes: el proceso *master* (barra superior) y el proceso *worker* (barra inferior).

Como se puede ver, esta aplicación se sincroniza en la recepción de todos los individuos de la población. Para que el proceso *master* aplique los operadores genéticos entre los individuos de la población, todos los individuos deben ser previamente recibidos por dicho proceso. Al utilizar un sólo proceso *worker*, se puede ver cómo, básicamente, el proceso *master* está casi todo el tiempo esperando por la recepción del mensaje de este único proceso *worker*. Cuando se utiliza más de un proceso *worker*, el proceso *master* realiza más comunicaciones por lo que no pasa tanto tiempo ocioso. Se puede notar que la aplicación de los operadores genéticos no requieren tiempo significativo: en esta figura no se alcanza a distinguir el tiempo empleado en dichos operadores, las flechas que representan la recepción y envío de *chunks* están solapadas.

A partir de este gráfico, se analizará qué sucede en el proceso *worker*. A continuación distintas zonas de interés serán ampliadas y, para facilitar la comprensión de las figuras, se han reemplazado las referencias originales de las gráficas de *MPI_Open_Graphics*, dejando sólo las referencias que interesan analizar (traducidas al castellano) y ampliando sólo la zona donde

se ilustra la tarea del proceso *worker*.

El análisis de las siguientes figuras, permite observar que la mayoría del tiempo empleado por el proceso *worker* es utilizado para realizar las simulaciones de la propagación del fuego con los distintos individuos. Con el fin de observar en más detalle, se utiliza el mismo tipo de gráfico, pero se amplía el análisis de zonas más pequeñas.

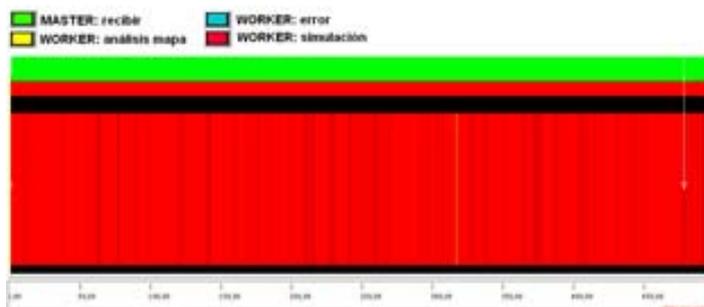


Figura 4.13: Primer evolución de la población.

En la figura 4.13 se puede observar el procesamiento del proceso *worker* durante la primera evolución. Dicha evolución finaliza en la flecha blanca, la cual representa el envío del *chunk* evaluado desde el proceso *worker* al proceso *master*. A su vez, en el gráfico se solapa dicha flecha con la que representa la comunicación del *chunk* para la siguiente generación desde el proceso *master* al *worker*.

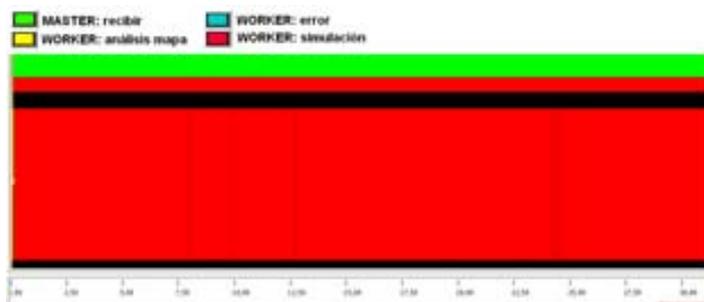


Figura 4.14: Sólo 30 segundos de la evolución.

El análisis de estas figuras (4.12, 4.13, 4.14 y 4.15) permiten detectar que el mayor tiempo de procesamiento de la aplicación es la realización de las simulaciones. Ni el análisis del mapa (en amarillo) ni el cálculo de



Figura 4.15: Simulación, análisis del mapa simulado y cálculo del error para un individuo.

error (en celeste) alcanzan a ser significativos cuando se los compara con el tiempo que pasa el proceso *worker* simulando la propagación del fuego.

Aunque no son significativos los tiempos de cómputo de las operaciones genéticas ni el análisis del mapa ni el cálculo de error no se ha descuidado su implementación, buscando siempre minimizar los tiempos de ejecución obteniendo así una implementación correcta y eficiente.

Entonces, en nuestra aplicación la distribución de trabajo se efectuará repartiendo los individuos de las poblaciones entre los procesos *workers*, distribuyendo entre los mismos la carga de trabajo. La siguiente sección describirá la forma en que se realiza dicha distribución.

Balaceo de Carga - Granularidad

Dado que el método de predicción propuesto utiliza un algoritmo genético, por definición, se dispondrá de una población inicial compuesta de múltiples individuos, la cual se evolucionará un número predeterminado de veces (iteraciones del algoritmo genético). En cada generación, se evalúan todos los individuos de la población, esto quiere decir, se realiza la simulación, se analiza el mapa y se calcula el error con cada uno de ellos. Por otro lado, se sabe que la simulación es la tarea que más requerimiento de cómputo tiene en esta aplicación.

Un posible problema que puede penalizar el cómputo en paralelo de esta aplicación son las comunicaciones ya que todos los procesos *workers* se comunican con el proceso *master* para la recepción y envío de individuos. De esta forma, la comunicación de todos los procesos *workers* con el proceso *master* puede llegar a convertirse en un cuello de botella de la aplicación.

Para evitar este posible problema, el envío de los individuos se realizará de a grupos de individuos (los *chunks* mencionados anteriormente).

De esta forma, se aumenta la granularidad de la aplicación y se disminuye la cantidad total de comunicaciones.

Entonces, cada proceso *worker* recibe un *chunk* y evalúa los individuos del *chunk* recibido (simulación + cálculo del error). Una vez que dicho proceso evalúa todos los individuos del *chunk*, lo retorna al proceso *master* (retornando cada individuo con su correspondiente error). El proceso *master* almacena los individuos del *chunk* recibido (ya evaluado), y si existen más *chunks* sin evaluar todavía, envía al proceso *worker* que acaba de retornar su *chunk* uno nuevo para su procesamiento.

De esta forma, la distribución se hace “bajo demanda”: a pesar de que todos los *chunks* tienen el mismo tamaño (cantidad de individuos), el tiempo de procesamiento que requiere cada uno de los *chunks* puede variar principalmente por dos motivos:

- Heterogeneidad del cluster: si estamos trabajando con un cluster heterogéneo, los procesadores pueden tener distintas velocidades de procesamiento, lo que implica que unos puedan tardar más que otros en evaluar los individuos.
- Evaluación de cada individuo: recordando que cada individuo es una asignación específica de valores a las variables que describen el medio ambiente donde el fuego se desarrolla, entonces, dependiendo de estos valores, el fuego puede no propagar, propagar rápido por parte de la superficie, propagar lento por parte de la superficie, o puede propagar a distintas velocidades por toda la extensión del terreno. Esto quiere decir que cada individuo requiere su propio tiempo de ejecución, que puede ser muy distinto a los tiempos de los demás individuos.

Al realizar la distribución bajo demanda, se intenta evitar posibles desbalances de carga debido a los dos puntos anteriores.

Un aspecto importante a tener en cuenta es que cada una de las simulaciones son independientes entre sí. Esto hace que se puedan evaluar en cualquier orden, sin interferir esto en el correcto avance de la ejecución de la aplicación.

Otras Opciones

Dependiendo de la aplicación, sus objetivos y sus restricciones, de la arquitectura utilizada, etc., la paralelización puede ser realizada de diferentes maneras.

Otra forma de paralelizar este problema consiste en dividir el mapa entre los distintos procesos, paralelizando así la propagación de todo el frente del fuego. De esta forma, cada *worker* propaga el fuego en el territorio del mapa que le haya sido asignado, o bien, propaga los puntos de la línea

del fuego asignados. Esta forma de paralelización plantea dos problemas importantes:

- Cómo se resuelve la transición del fuego desde un proceso a otro cuando el fuego alcanza el borde del territorio propio de un proceso *worker*, y
- cómo evitar desbalanceo de carga entre los procesos debido al avance del fuego: el territorio “activo” (transición de no quemado a quemado), implica procesamiento en la simulación, mientras que en territorios por donde el fuego no ha llegado o ya se ha extinguido (territorio “inactivo”), no existe la necesidad de computar.

Sólo para poder comparar distintas aproximaciones, en [51] se describe una aplicación sobre simulaciones de incendios forestales de grandes escalas. En este trabajo se describe otra forma de paralelizar el problema y además se tratan de forma clara y extensa la solución a los dos puntos observados anteriormente. En los dos párrafos que siguen se explicará de forma muy simple las principales características de cómo se solucionan estos aspectos.

En este trabajo, se tratan de minimizar las comunicaciones haciendo que todos los procesos conozcan todo el mapa completo, y cada uno conoce qué región le es asignada para calcular. Luego de que cada proceso *worker* propaga el fuego en sus celdas (las necesarias), envía a un proceso coordinador sólo las celdas quemadas (con esto evitan muchas comunicaciones).

Por otro lado, el desbalanceo de carga se soluciona mediante un proceso de “detección y reparación”: cuando el desbalanceo supera un umbral, se invocan las funciones de reparación. El desbalanceo depende del mínimo y máximo tiempo de cómputo observado entre los distintos procesadores. Un nodo dedicado a esto, calcula el desbalanceo y determina la carga ideal que debería tener cada proceso *worker*. Este cálculo se solapa con el trabajo de cada uno de los procesos *workers*. Luego, cada uno de éstos lee qué región le es asignada. Mediante esta solución, el balanceo de carga depende de las celdas en actividad (quemándose).

Este trabajo es un ejemplo de cómo se solucionan ambos aspectos, soluciones efectivas, pero que tienen la necesidad de agregar coordinadores o nodos especiales para su resolución. Aunque los trabajos de los distintos procesos se puedan solapar, la complejidad del sistema final aumenta a medida que se agregan procesos específicos para cada tarea. Dividir el terreno es una solución válida (y muy utilizada) cuando se tratan incendios sobre áreas muy extensas.

Existen muchas formas de solucionar estos problemas y cada solución tendrá sus puntos fuertes y puntos débiles, dependiendo esto del tipo y formato de los datos de entrada disponibles, de las salidas deseadas, del tipo de arquitectura para el que se programa, etc.

Pseudocódigos del Algoritmo Paralelo

Una vez vistas distintas propuestas y explicadas algunas de las características más importantes de nuestra aplicación, se mostrará el pseudocódigo de los procesos más importantes de nuestra aplicación:

Algoritmo 4.2.1 Pseudocódigo del proceso *master*

```
1: obtener población inicial
2: while (generaciónActual < cantidadEvoluciones)
3:   evaluar población
4:   aplicación de operadores genéticos en la población
5:   generaciónActual + 1
6: end while
7: evaluar población
8: obtener mejor individuo
```

Del algoritmo 4.2.1, hay dos funciones que merecen ser descritas para una mejor comprensión del algoritmo: función *evaluar población* (línea 3 del algoritmo) y función *aplicación de los operadores genéticos* (línea 4 del mismo algoritmo):

Algoritmo 4.2.2 Pseudocódigo de la función evaluar población

```
1: chunksRecibidos  $\leftarrow$  0
2: for cada worker
3:   enviar chunk al worker
4: end for
5: while (chunksRecibidos < total chunks)
6:   recibir chunk
7:   chunksRecibidos + 1
8:   if (chunksRecibidos < total Chunks) then
9:     enviar chunk al worker
10:  end if
11: end while
```

A su vez, la aplicación de los operadores genéticos implica realizar los pasos mostrados en el algoritmo 4.2.3.

Todos los pseudocódigos anteriores corresponden al proceso *master*. En el pseudocódigo 4.2.4 se muestra el pseudocódigo de los procesos *workers* que se ejecutan en paralelo.

En cada pseudocódigo se han evitado poner pre y post condiciones para no dificultar la lectura de los mismos.

Una vez explicadas las principales características de nuestra aplicación

Algoritmo 4.2.3 Pseudocódigo del proceso aplicación de los operadores genéticos en la población

```
1: padre1 ← selección
2: padre2 ← selección
3: cruzamiento(padre1, padre2, hijo1, hijo2)
4: mutación(hijo1)
5: mutación(hijo2)
```

Algoritmo 4.2.4 Pseudocódigo de los procesos workers

```
1: leer mapa inicial
2: leer y analizar mapa final
3: recibir chunk
4: while (existen chunks a evaluar)
5:   for cada individuo del chunk
6:     simular
7:     calcular el error
8:   end for
9:   enviar chunk evaluado al master
10:  recibir nuevo chunk
11: end while
```

paralela, en las próximas secciones se expondrán las mejoras propuestas a este código: la guía el algoritmo genético a través de la inyección dinámica de datos. Intentaremos, a través de conocimiento sobre el progreso del fuego real, agregar información para sintonizar la búsqueda realizada por el algoritmo genético.

4.3. Etapa de Ajuste Guiada por los Datos

En las próximas secciones se detalla la etapa de ajuste, que es donde se agregan mejoras propuestas al método de predicción en dos etapas.

Dichas mejoras se basan en la definición de una aplicación donde se implementa un *Dynamic Data Driven Genetic Algorithm*, esto es, un algoritmo genético al cual se le inyectan datos con la intención de guiar el proceso de búsqueda. La inyección de datos se basa en conocimiento obtenido de datos reales y tiene como objetivo acelerar la convergencia a buenas soluciones durante el proceso de búsqueda.

Para esto se proponen 2 métodos de inyección dinámica de datos, uno creado para su utilización independientemente del simulador de incendios utilizado (al cual se lo utiliza como caja negra). El segundo método que se

ha propuesto es utilizado para verificar la correctitud de los resultados del primero. Ambos tienen diferencias importantes las cuales serán explicadas más adelante en este mismo capítulo.

Existen diversos trabajos donde es posible encontrar distintas formas de guiar y mejorar los algoritmos genéticos. Estas sintonizaciones dependen del dominio de la aplicación, pero suelen ser significativas las mejoras que dichos algoritmos permiten, tanto en las características propias del algoritmo genético y sus características como así también en la sintonización del espacio de búsqueda del algoritmo.

Como ya se ha mencionado, el algoritmo genético trabaja con un amplio espacio de búsqueda, intentando encontrar buenos individuos y, a su vez, evitar la evaluación de individuos con características que no logren mejorar la solución del problema (salvo excepciones, gracias a la diversidad de características garantizada por el algoritmo).

En nuestra aplicación, se utiliza este método en la etapa de ajuste para sintonizar y mejorar los parámetros de entrada para la etapa de predicción.

En este momento, surgen preguntas como, ¿es el algoritmo genético la forma más eficiente de realizar la búsqueda? ¿se puede mejorar la búsqueda dentro de este algoritmo?

Trabajos anteriores han demostrado que estos algoritmos son adecuados para este problema (por ejemplo, el análisis realizado en [1]). Las características de nuestra aplicación, del método de predicción en dos etapas, y, las características de la etapa de ajuste en particular, nos permiten proponer mejoras en nuestra aplicación.

Durante la etapa de ajuste, se utiliza el mapa del avance real del fuego hasta un instante t_{i+1} posterior al tiempo de comienzo de ignición del incendio (t_i). Este mapa es el avance real del fuego, por lo que se lo puede utilizar como fuente de conocimiento para mejorar la búsqueda del algoritmo genético.

Entonces, basándonos en el análisis del avance del fuego, inyectaremos conocimiento para lograr que nuestro algoritmo genético converja de forma más rápida a mejores soluciones.

En las próximas secciones se expondrá cómo se realiza el análisis del mapa del avance real del fuego, qué conocimiento es extraído del mismo y cómo. Esto se analizará en las siguientes secciones, y posteriormente, se desarrollarán dos métodos empleados para la utilización de este conocimiento.

4.4. Obtención del Conocimiento

Durante el capítulo 3 se ha presentado el método clásico de predicción como así también el método de predicción en dos etapas. La figura 3.2 mues-

tra el esquema de este método, donde se puede ver claramente la utilización del mapa del avance del fuego real desde el instante considerado como inicio del fuego (t_i) hasta el próximo instante de tiempo considerado (t_{i+1}).

En este trabajo, se analizará el avance del frente del fuego y este estudio se utilizará para: 1) determinar la forma del fuego, y 2) para determinar sus principales características: dirección, velocidad y distancia recorrida. Además, el conocimiento aportado por la propagación real del fuego, implica tener en cuenta los cambios ambientales relacionados con estos parámetros.

En este trabajo se diferencian dos formas típicas de propagación del fuego: fuegos que crecen de forma elíptica y fuegos que crecen con un frente ancho (lineal).

Si el foco del incendio se encuentra sobre un terreno con pendiente o existe viento (o ambos) que puedan llegar a influir sobre el fuego, es normal que el fuego propague de forma elíptica, donde existe un sector del frente del fuego que alcanza una mayor velocidad de propagación (figura 4.16 (a) y (b)).

Otro caso es que el fuego avance de forma más o menos pareja en todo un frente que se caracteriza por ser un frente ancho. Este caso se ilustra en la figura 4.16 (c) y es el que definimos como fuego “lineal”.



(a) Fuego elíptico 1 (b) Fuego elíptico 2 (c) Fuego lineal

Figura 4.16: Formatos típicos de propagación del fuego.

Las características del fuego nos ayudarán a agregar conocimiento al algoritmo genético utilizado en este trabajo. Entonces, necesitamos que la descripción del mismo sea lo más correcta posible. Esto no siempre es fácil porque se intenta describir el comportamiento de un fenómeno totalmente complejo, dinámico y dependiente de muchos factores (como ya se ha mencionado en los primeros capítulos).

A continuación, se describirá en detalle cómo se realiza el análisis del mapa del fuego real.

Supongamos que el fuego hasta el instante t_{i+1} tiene el aspecto que muestra la figura 4.17 (a), entonces, analizando el contenido de las celdas

del mapa (instante en que la celda es alcanzada por el fuego), se determinan las celdas que pertenecen al frente del fuego inicial y se almacenan para su uso posterior (esto es, instante t_i , figura 4.17 (b)). De la misma forma, se determinan y almacenan las celdas pertenecientes a la línea de fuego final correspondiente al instante t_{i+1} , tal como muestra la figura 4.17 (c).

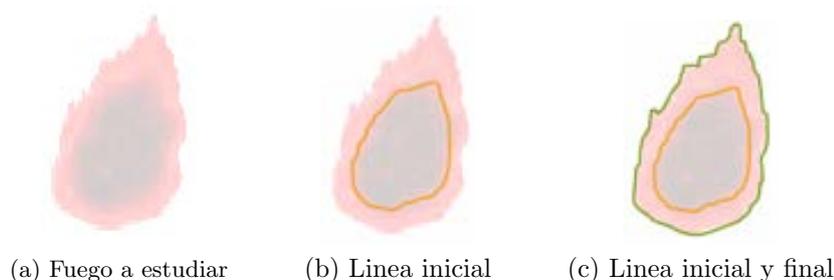


Figura 4.17: Análisis del formato del fuego.

Este análisis resulta directo debido a la representación de los mapas del terreno hechos en nuestra aplicación. Los terrenos están representados por matrices de celdas, entonces, las celdas pertenecientes al frente del fuego para el instante t_i son aquellas que tienen tiempo de llegada del fuego igual o menor a t_i y que tienen al menos una celda vecina que cumpla alguna de estas condiciones: tiempo de ignición mayor a t_i o celda no quemada. Si la celda en análisis tiene algún vecino que cumple estas condiciones, se la considera perteneciente a la línea de fuego inicial y se la almacena. De forma muy similar se arma la línea de fuego final, sólo que considerando el tiempo t_{i+1} . Cabe destacar que para cada celda cuadrada, se consideran sus ocho celdas vecinas (autómata celular).

Una vez que se obtuvieron y almacenaron las dos líneas de fuego, se utiliza un algoritmo que adopta la idea general de la distancia de Hausdorff [48] y se lo implementa según nuestras necesidades.

Aunque los mapas de la aplicación estén almacenados como vectores unidimensionales, el tratamiento lógico de los mismos es el de arreglos bidimensionales, correspondiéndose entonces con los conceptos de autómatas celulares. A su vez, las líneas de fuego son almacenadas como un conjunto de puntos (x, y) , donde a través de estas coordenadas, se accede a la posición en el mapa como así también al tiempo de ignición alocado en cada celda.

El algoritmo 4.4.1 intenta descubrir las características principales del fuego. Por un lado, el formato, y luego, la dirección, la velocidad y la distancia recorrida que describan de mejor forma su comportamiento global.

Para describir el mapa, por cada celda de la línea de fuego final (esto

Algoritmo 4.4.1 Pseudocódigo del proceso para la obtención de las características del avance del fuego.

```
1: for  $celda_n$  de la línea de fuego  $t_{i+1}$ 
2:   velocidadMínima  $\leftarrow$  valorAlto
3:   for  $celda_m$  de la línea de fuego  $t_i$ 
4:     distanciaActual  $\leftarrow$  distancia( $celda_m$ ,  $celda_n$ )
5:     tiempoTranscurrido  $\leftarrow$  diferenciaTiempos( $celda_m$ ,  $celda_n$ )
6:     velocidadActual  $\leftarrow$   $\frac{distanciaActual}{tiempoTranscurrido}$ 
7:     if (velocidadActual  $\leq$  velocidadMínima) then
8:       velocidadMínima  $\leftarrow$  velocidadActual
9:       almacenamiento de las características de la propagación desde
         $celda_n$  hasta  $celda_m$ 
10:    end if
11:  end for
12:  almacenamiento de las características de la propagación desde  $celda_n$ 
    hasta la celda de menor velocidad en 3 vectores: dirección, distancia
    y velocidad
13:  if (velocidadMínima  $\geq$  velocidadMáxima) then
14:    velocidadMáxima  $\leftarrow$  velocidadMínima
15:    almacenamiento de los datos de las celdas  $celda_m$  y  $celda_n$ 
16:  end if
17: end for
```

es, en el instante t_{i+1}) se busca la celda de la línea de fuego inicial (instante t_i), para la que se observa menor velocidad (figura 4.18). En este caso, al tener el tiempo constante (por cómo son armados los frentes del fuego), la celda que muestre menor velocidad, será la de menor distancia, camino mínimo que recorre el fuego desde su ubicación en t_i hasta t_{i+1} .

En la línea 12 del algoritmo 4.4.1, se puede ver que se almacenan

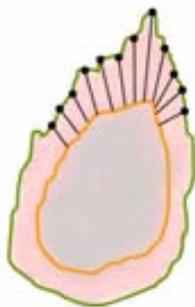


Figura 4.18: Análisis del mapa mediante una adaptación del algoritmo de Hausdorff.

los datos de cada par de celdas donde se observó la menor velocidad de propagación. El almacenamiento de estos datos, permitirá más adelante la descripción del formato del fuego y sus principales características.

En un primer momento, sólo se consideraba el máximo de los mínimos (líneas 13, 14 y 15 del algoritmo 4.4.1) que encontraba este algoritmo, pero diversos mapas, permitieron observar que esta no era una descripción correcta para todos los casos de propagación.

Por ejemplo, en casos como muestra la figura 4.19, donde el avance del fuego es lineal, es normal que suceda que el máximo de los mínimos se encuentre en un borde, y entonces, que toda la descripción del mapa quede determinada por lo que sucede entre dichas celdas.

El comportamiento del fuego en las dos celdas que muestra la figura anterior, no describe el comportamiento del fuego y esta descripción errónea seguramente derivará en una mala guía de los parámetros (ya que guiaremos hacia un comportamiento equivocado del fuego real).

En este caso, una mejor descripción del mapa sería una medida que describa el avance de todo el frente del fuego, y no solamente lo que sucede entre dos celdas. El caso de mapas lineales, hizo definir el análisis del fuego en base al formato del mismo: primero se determina si la forma del fuego es elíptica o lineal y luego, a partir de su forma, se considera el comportamiento de todo el avance del fuego o sólo las celdas donde se observa el

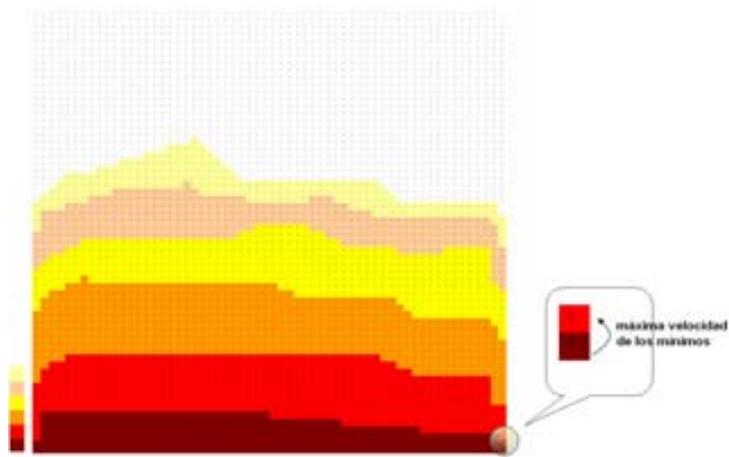


Figura 4.19: Avance del fuego lineal, donde todo el avance queda descrito por dos celdas contiguas.

máximo punto de propagación (más adelante en esta misma sección se hace una descripción más amplia de estos cálculos).

Los datos almacenados en el algoritmo 4.4.1 (línea 12) son la cantidad de ocurrencias de cada dirección observada, de cada distancia y de cada velocidad. Luego, estas muestras son analizadas usando medidas descriptivas (de centralización y de dispersión), para determinar si es un fuego lineal o elíptico:

- Fuego lineal: poca desviación en la muestra de direcciones y poca desviación en la muestra de distancias. Esto quiere decir, el fuego de muchas celdas avanza con el mismo sentido, a velocidad constante.
- Fuego elíptico: gran desviación en la muestra de direcciones y gran desviación en la muestra de distancias. Esto ocurre cuando el fuego avanza de forma elíptica, en la “cabeza” del incendio el fuego avanza rápido, mientras que en todas las otras direcciones avanza más lento. Esto hace una muestra con muchas direcciones de avance distintas a diferentes velocidades.

En nuestro caso, hemos tomado como medidas de centralización, la media y la moda de cada muestra (direcciones, velocidades y distancias), y luego, se trabaja con la medida que muestre menor desviación de estas dos, y justamente sus desviaciones determinan el formato del mapa (según el análisis del mapa en función a su forma).

Una vez definido el formato, se utilizan distintos criterios para describir el mapa.

Si el fuego es elíptico, se considera el punto de máxima velocidad de propagación, su dirección, velocidad y distancia para describir el comportamiento del fuego real.

Por otra parte, si el fuego es lineal, se consideran ambas medidas de centralización, la media o la moda (la que menos desviación generó).

Hasta aquí se ha descrito cómo se analiza el mapa del avance del fuego para obtener conocimiento útil para guiar nuestro algoritmo genético. Pero antes de explicar cómo se utiliza este conocimiento, debemos introducirnos en algunos detalles del modelo de Rothermel y sus cálculos. La próxima sección aborda estos temas, para posteriormente, explicar cómo se combina todo esto para guiar nuestro algoritmo genético.

4.5. Determinación de la Dirección e Intensidad de Máxima Propagación del Fuego.

Por razones físicas del fuego, el mismo es influenciado por diversas variables del entorno y del combustible. Pero se reconoce que la mayor influencia en la forma final del fuego la ejercen la pendiente y el viento. De todos los factores influyentes, son estos dos los que afectan en mayor medida al avance del fuego [1].

En el caso de la pendiente, el fuego avanza con mayor rapidez cuando se encuentra en un terreno con pendiente (a favor de la misma). Por otro lado, el viento influye en la evolución del fuego por diferentes causas, por ejemplo, empuja las llamas sobre la superficie del terreno, aporta más oxígeno, etc.

De forma muy simplificada y esquemática para facilitar la comprensión, en las siguientes figuras se ilustra cómo se componen las características del viento y de la pendiente para determinar hacia qué dirección el fuego propaga con máxima intensidad. Como ya se ha mencionado, sabemos que el fuego propaga en todas direcciones dependiendo de las características del terreno y las características propias del fuego (intensidad, calor que irradia, etc). Pero en los simuladores, se suele buscar la dirección de máxima intensidad de propagación (ROS) y, a partir de esta dirección e intensidad, se calcula la propagación para el resto de las direcciones.

En las figuras 4.20 (a) y (b) se puede observar cómo se combinan el efecto que produce el viento y el efecto de la pendiente (características que son obtenidas a partir de los datos de entrada del simulador). La figura

4.20 (a) muestra el caso más complejo de resolver: el incendio se desarrolla sobre un terreno con pendiente y viento avanzando en distinta dirección. Entonces, en este caso, se combinan las características de estos dos factores y se determina la dirección hacia la que el fuego propagará con su máxima intensidad (figura 4.20 (b)).

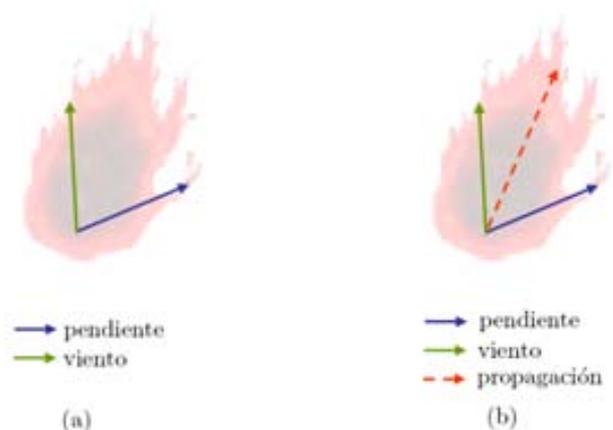


Figura 4.20: (a) El efecto de la pendiente y del viento determinan la propagación del fuego (línea punteada en (b)).

La ecuación del modelo de Rothermel que resuelve esta relación es $\Phi = \Phi_w + \Phi_s$ (ecuación 2.2 del capítulo 2 y reescrita en este capítulo, ecuación 4.7). El simulador utilizado en este trabajo (*fireLib*), implementa esta ecuación en la función :

```
Fire_SpreadWindSlopeMax(catalog, model, windFpm, windDeg,
                        slope, aspect)
```

que es donde se combinan los efectos del viento y de la pendiente. En dicha función, se definen 6 situaciones diferentes: no hay propagación o hay baja intensidad de reacción, no hay viento ni pendiente, presencia de viento sin pendiente, presencia de pendiente sin viento, existen viento y pendiente y el viento avanza a favor de la pendiente, y por último, hay presencia de viento y pendiente y ambos avanzan en distintas direcciones.

Entonces, en todos los casos excepto el último es “directo” definir la dirección del fuego, pues dicha dirección será definida en función de un sólo factor, y no de la combinación de ambos. En el último caso, es el único caso donde se combinan los efectos del viento y de la pendiente.

Algoritmo 4.5.1 Pseudocódigo para la determinación de la dirección de máxima propagación.

Require: Fuel_Spread0 \leftarrow propagación sin viento ni pendiente, depende del individuo

Require: Fuel_PhiSlope \leftarrow factor de la pendiente, depende de su inclinación y del manto de combustible

Require: Fuel_PhiWind \leftarrow factor del viento, depende de su velocidad y del manto de combustible

```
1: {combinación de los efectos de la pendiente y del viento}
2: phiEw  $\leftarrow$  Fuel_PhiSlope + Fuel_PhiWind
3: {Situación 1: no hay propagación o intensidad de fuego}
4: if (Fuel_Spread0() < Límite) then
5:   spreadMax  $\leftarrow$  0 {intensidad de la propagación}
6:   azimuthMax  $\leftarrow$  0 {dirección de máxima propagación}
7:   {Situación 2: no hay viento, no hay pendiente}
8: else if (phiEw < Limite) then
9:   spreadMax  $\leftarrow$  Fuel_Spread0()
10:  azimuthMax  $\leftarrow$  0
11:  {Situación 3: viento sin pendiente}
12: else if (Slope < Limite) then
13:   spreadMax  $\leftarrow$  Fuel_Spread0() * (1. + phiEw)
14:   azimuthMax  $\leftarrow$  dirViento
15:   {Situación 4: pendiente sin viento}
16: else if (Viento < Límite) then
17:   spreadMax  $\leftarrow$  Fuel_Spread0() * (1. + phiEw)
18:   azimuthMax  $\leftarrow$  direcciónPendiente
19:   {Situación 5: viento avanza a favor de la pendiente}
20: else if (direcciónPendiente == direcciónViento) then
21:   spreadMax  $\leftarrow$  Fuel_Spread0() * (1. + phiEw)
22:   azimuthMax  $\leftarrow$  direcciónPendiente
23: else
24:   {Situación 6: viento cruza la pendiente}
25:   ángulo  $\leftarrow$  direcciónPendiente - direcciónViento
26:   slpRate  $\leftarrow$  Fuel_Spread0() * Fuel_PhiSlope()
27:   wndRate  $\leftarrow$  Fuel_Spread0() * Fuel_PhiWind()
28:   x  $\leftarrow$  slpRate + wndRate * cos(angulo)
29:   y  $\leftarrow$  wndRate * sin(angulo)
30:   rv  $\leftarrow$   $\sqrt{x^2 + y^2}$ 
31:   spreadMax  $\leftarrow$  Fuel_Spread0() + rv
32:   al  $\leftarrow$   $\frac{\arcsin(\frac{fabs(y)}{rv})}{rv}$ 
33:   azimuthMax  $\leftarrow$  al + direcciónPendiente
34: end if
```

El algoritmo 4.5.1 muestra cómo se determina la dirección de propagación (variable *azimuthMax*) en cada uno de los casos mencionados anteriormente. En estas líneas se muestra cómo se implementa el modelo de Rothermel en el simulador *fireLib*. Este pseudocódigo es una abstracción resultado del estudio del modelo y de su implementación en dicho simulador. El código es mucho más extenso y exhaustivo. Sin embargo, en esta sección se muestra sólo una abstracción de dicho código, donde muchas cosas se asumen y excluyen (para lograr claridad en la explicación del método).

Desde la línea 1 a la 22 se resuelven los 5 primeros casos mencionados anteriormente, donde la dirección y la velocidad de máxima propagación son determinadas por un sólo factor (por el viento o por la pendiente). Luego, desde la línea 25 a la 33 del mismo algoritmo, se resuelven las ecuaciones que determinan la combinación de ambos factores. Estos son los pasos que realiza el simulador, y los mismos nos servirán en el momento en que se describa el Método Analítico propuesto (sección 4.6.2).

Una vez explicado cómo se analiza el mapa para describir el avance del fuego real, y una vez introducidos los principales conceptos de cómo determina el modelo y el simulador la dirección de máxima velocidad de propagación, se explicará cómo se combinan estas características para guiar nuestra aplicación.

4.6. Utilización del Conocimiento para Sintetizar los Valores de los Parámetros

Una vez explicado cómo el modelo y el simulador obtienen la dirección de máxima propagación del fuego a partir del viento y de la pendiente, se explicará en qué consiste nuestro método que utiliza conocimiento de la propagación real del fuego para guiar al algoritmo genético.

En nuestra aplicación se considera que las características de la pendiente del terreno son conocidas. Entonces, se conoce la inclinación de la pendiente y su orientación (ver figura 4.21 (a), la pendiente representada a través de un vector). A su vez, mediante el análisis y descubrimiento de las características del avance del fuego real (descrito en la sección 4.4 de este capítulo), se obtiene la dirección y velocidad del avance del fuego real en el instante t_{i+1} (figura 4.21 (b)).

En este punto, se puede observar que se conocen 2 de los tres vectores que mostramos anteriormente en la figura 4.20 (b). En la sección anterior se describió cómo, combinando el viento y la pendiente, se encuentra la dirección de máxima propagación del fuego. Ahora, proponemos combinar la pendiente (conocida) con la dirección y velocidad de máxima propagación del fuego (obtenida a partir del análisis del mapa) para descubrir el valor

del viento todavía desconocido.

Entonces, esta vez, combinando la pendiente y las características del avance del fuego, se obtiene cuál sería el viento ideal, para que, en el simulador, cuando sea combinado con la pendiente, se logre la propagación observada en t_{i+1} (figura 4.21 (c)).

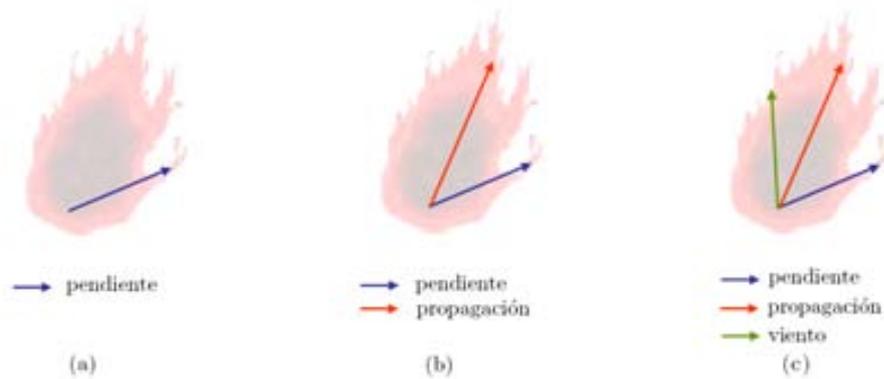


Figura 4.21: (a) y (b) Datos conocidos: pendiente y propagación del fuego. (c) Obtención de las características del viento ideales.

Entonces, usaremos los datos del viento descubiertos para inyectar este conocimiento durante la evolución del algoritmo genético.

De esta forma, nuestra aplicación es una aplicación guiada dinámicamente por los datos, por el conocimiento aportado por el propio avance real del fuego. Estamos frente a un factor dinámico como es el viento, pero inyectamos conocimiento para suplir la incertidumbre que puede llegar a introducir la imprecisión de este dato de entrada que es determinante en el formato del fuego (por ende, de la calidad de las simulaciones y predicciones).

Se han propuesto 2 métodos para descubrir las principales características del viento y luego, inyectar este conocimiento en la aplicación. Estos métodos son: el Método Computacional y el Método Analítico. El Método Analítico fue creado para verificar el correcto funcionamiento del Método Computacional. En las próximas secciones se describirán ambos métodos, detallándose sus principales características. Entonces, utilizando alguno de estos métodos, se obtendrán los datos del viento y luego se inyectarán durante el método genético para guiar la búsqueda.

4.6.1. Método Computacional

El Método Computacional es un método desarrollado para inyectar conocimiento de forma dinámica en el algoritmo genético. Para ello se utilizará información almacenada en archivos de texto plano, configurando nuestra base de conocimiento. Esta base de conocimiento nos proporcionará los valores del viento que queremos obtener.

En la base de conocimiento se describen distintos casos de incendios. Esta información puede ser datos de incendios reales, datos históricos de fuegos sucedidos y estudiados posteriormente o pueden ser incendios obtenidos a partir de simulaciones obtenidas con algún simulador. Cualquiera de los casos es válido para nuestra aplicación.

Un factor importante de la base de conocimiento es la completitud de la misma. Es importante tener la mayor cantidad de casos de incendios posibles, ya que la calidad de los datos inyectados dependerá fuertemente de la completitud de esta base de conocimiento.

De todas formas, como es difícil garantizar esto, asumimos una falta de completitud en la misma, lo cual definirá un aspecto importante durante la inyección de datos que será comentado más adelante en este mismo capítulo (sección 4.7: Inyección de Datos).

El Método Computacional se basa en buscar el caso más parecido al incendio real (incendio a predecir), para el instante t_{i+1} . Dada la pendiente observada en el entorno del incendio, se busca la pendiente más parecida en la base de conocimiento, y el caso de incendio con propagación más similar al observado en el instante t_{i+1} . Una vez hallado el fuego que mostró el comportamiento más parecido en la base de conocimiento, se consideran los valores del viento de dicho incendio como los valores ideales para el caso en estudio.

Formato de los Datos en la Base de Conocimiento

A continuación se detallan los datos almacenados y utilizados en nuestra aplicación. Cada incendio de nuestra base de conocimiento está descrito por:

- Modelo de combustible: uno de los 13 modelos descritos en el capítulo 2.
- Pendiente: grados de inclinación del terreno.
- Dirección de propagación: dirección donde ocurre la máxima velocidad de propagación del fuego observada en el incendio.
- Velocidad de propagación: velocidad observada en la dirección de máxima propagación.

- Distancia: distancia recorrida por el fuego.
- Dirección del viento: dirección del viento presente en ese incendio.
- Velocidad del viento: velocidad del viento presente en ese incendio.

A modo de ejemplo, la figura 4.22 muestra algunos posibles registros de la base de conocimiento (donde la pendiente hace referencia a la inclinación del terreno y se muestra en radianes).

modelo	pendiente	dirección	velocidad	distancia	dirección viento	velocidad viento
7	0.99	0.00	82.84	180.44	0.00	21.00
7	0.99	0.00	82.84	180.44	0.00	24.00
7	0.99	0.00	82.84	180.44	0.00	27.00
7	0.99	0.00	82.84	180.44	0.00	30.00
7	0.99	0.00	24.55	78.73	45.00	0.00
7	0.99	0.00	23.69	65.61	45.00	3.00
7	0.99	0.00	16.65	39.36	45.00	6.00
7	0.99	45.00	21.39	64.95	45.00	9.00
7	0.99	45.00	31.21	92.79	45.00	12.00
7	0.99	45.00	41.71	129.91	45.00	15.00
7	0.99	45.00	49.74	162.39	45.00	18.00
7	0.99	45.00	56.35	176.31	45.00	21.00
7	0.99	45.00	61.60	236.62	45.00	24.00
7	0.99	45.00	65.71	213.43	45.00	27.00
7	0.99	45.00	68.91	250.54	45.00	30.00
7	0.99	0.00	24.55	78.73	90.00	0.00
7	0.99	0.00	15.04	59.05	90.00	3.00
7	0.99	45.00	30.53	106.71	90.00	6.00
7	0.99	45.00	30.06	115.99	90.00	9.00
7	0.99	45.00	18.49	64.95	90.00	12.00
7	0.99	90.00	21.15	52.49	90.00	15.00
7	0.99	90.00	29.37	111.54	90.00	18.00

Figura 4.22: Formato de los registros de la base de conocimiento.

En este punto vale la pena considerar algunos detalles, consideraciones que fueron estudiadas en el momento de la implementación y utilización de este método.

Por un lado, vale la pena meditar unos instantes en la enorme cantidad

de posibles casos que deberían estar incluidos en estas tablas. Por un lado, en estas tablas no se especifican todos los factores que influyen en un incendio (falta por ejemplo, las humedades del combustible vivo, muerto, etc.). Entonces, este conocimiento es una abstracción, que intenta, utilizando las características más influyentes e importantes, representar distintos comportamientos de fuegos.

Por otro lado, es posible observar que no se incluye en estas tablas la orientación de la pendiente. Este factor es muy importante, ya que determina la dirección del fuego.

Es fácil notar que en la propagación del fuego, no importan las direcciones absolutas del viento y de la pendiente, sino su relación. En nuestra aplicación (y en el simulador utilizado), se trabaja con la diferencia de dichas direcciones, esto es, ángulos relativos y no absolutos. El simulador utilizado también trabaja de esta forma, y se ve reflejado en la línea 25 del algoritmo 4.5.1.

Es por esta razón, que se toma la dirección del viento como relativa a 0, donde 0 es la orientación de la pendiente (origen).

En la próxima sección se explicará cómo se ha obtenido la base de conocimiento utilizada en este trabajo y en la sección subsiguiente, se explicará cómo se utiliza la información recolectada de la base de conocimiento para guiar el avance del algoritmo genético.

Creación de la Base de Conocimiento para Nuestra Experimentación

En nuestro caso, los datos de la base de conocimiento son fuegos sintéticos, obtenidos utilizando el simulador *fireLib* y analizando la evolución del fuego a partir del análisis de los frentes del fuego explicado anteriormente.

Existen diversas formas de obtener esta base de conocimiento, pero en nuestro caso, hemos realizado diversas simulaciones tratando de abarcar todo el espacio de estudio, discretizando los distintos datos de tal forma que todas las zonas están representadas. Entonces, cada variable (parámetro de entrada del simulador) se discretizó de la siguiente forma:

- Modelo: modelos de combustibles estándar: del modelo 1 al 13.
- Pendiente: de 0° a $57,3^\circ$, pasos de $6,3^\circ$ (de 0 a 1 radian, paso de 0,11 radianes).
- Dirección del viento: de 0° a 360° , con pasos de 45° (considerando las 8 direcciones principales de autómatas celulares).
- Velocidad del viento: 0 a 30 millas por hora, variando cada 3 millas por hora.

Las humedades utilizadas en estos casos, fueron estudiadas modelo por modelo (cada modelo de combustible) y se utilizaron valores medios de los mismos, para que no dificulte la propagación del fuego, ni tampoco, por el contrario, acelere su propagación de forma indiscriminada. La pendiente se varía hasta $57,3^\circ$ dado que el comportamiento del fuego es similar para pendientes con mayor inclinación. Algo similar ocurre con el viento, cuando este supera una determinada velocidad, la influencia del mismo sobre el fuego es la misma.

Como se ha mencionado anteriormente, se consideran direcciones relativas: se asume la dirección de la pendiente como origen, y el verdadero significado de la dirección de la propagación real de las tablas es el ángulo que forma con esta dirección origen. Lo único que resta es considerar que a la dirección del viento retornada por la rutina se le suma los grados de la dirección de la pendiente para retornar la dirección del viento acorde a dicho escenario.

En este punto, es útil considerar algunas particularidades del viento y de la pendiente. Como hemos dicho, se considera que las direcciones tanto del viento como de la propagación son relativas a la de la pendiente (origen). Entonces, tener un viento con dirección 180° , es casi imposible, pues es naturalmente extraño estar en presencia de un viento que vaya en contra de la pendiente. Y así no sólo con esta dirección, sino también con un rango más amplio cercano a este valor.

De todas formas, para no perder completitud consideramos también la inclusión de estos valores en nuestra base de conocimiento.

Para implementar la base de conocimiento, en nuestra aplicación simplemente se han utilizado ficheros planos dada la facilidad de su utilización y disponibilidad. Existen ya ampliaciones a este trabajo donde se consideran las ventajas de implementar esta base de conocimiento utilizando un motor de base de datos [50].

Utilización de la Base de Conocimiento para Descubrir los Valores del Viento

Antes de la aplicación del algoritmo genético, se utiliza el método computacional para obtener los datos del viento que sean óptimos para, en combinación con la pendiente, lograr una propagación similar a la observada en la realidad. A continuación, graficaremos con un ejemplo el procedimiento realizado para extraer de la base de conocimiento los datos del viento requeridos.

Se tiene un mapa con modelo de vegetación 7 (arbustos con muchas ramas) y pendiente de 1 radián ($57,3$ grados), y el mapa muestra una pro-

pagación a 45 grados (azimuth) y velocidad de 20 pies por minuto, entonces, según los datos de la figura 4.22, la rutina seleccionará el registro mostrado en la figura 4.23.

A continuación se utilizará la dirección y la velocidad del viento pro-

modelo	pendiente	dirección	velocidad	distancia	dirección viento	velocidad viento
7	0.99	0.00	82.84	180.44	0.00	21.00
7	0.99	0.00	82.84	180.44	0.00	24.00
7	0.99	0.00	82.84	180.44	0.00	27.00
7	0.99	0.00	82.84	180.44	0.00	30.00
7	0.99	0.00	24.55	78.73	45.00	0.00
7	0.99	0.00	23.69	65.61	45.00	3.00
7	0.99	0.00	16.65	39.36	45.00	6.00
7	0.99	45.00	21.39	64.95	45.00	9.00
7	0.99	45.00	31.21	92.79	45.00	12.00
7	0.99	45.00	41.71	129.91	45.00	15.00

Figura 4.23: Registro seleccionado por el método.

puestos, pero no como valores exactos para guiar la búsqueda, sino que se utilizarán para acotar el rango por donde varían los valores de dirección y velocidad del viento en la operación de mutación del algoritmo genético. Esta mutación, tomará un valor aleatorio en un rango con centro igual a los valores retornados por la rutina y amplitud acotada por un valor pre-determinado.

Se considera que en la base de datos no es posible tener todas las combinaciones de pendiente, viento y comportamiento del fuego que se puedan dar en la realidad (resultan ser un número muy elevado de combinaciones). Esta falta de completitud en la base de conocimiento define que se utilicen estos valores para definir un nuevo subrango de variación.

La figura 4.24 (a) muestra el rango de variación de la dirección del viento que, sin guiar, puede tomar cualquier valor dentro del rango 0..360 grados. En la figura 4.24 (b) se muestra cómo se reduce el espacio donde la dirección del viento puede variar cuando el algoritmo genético es guiado. Este valor es configurable pero en nuestras experimentaciones, se toma un rango de ± 5 grados a partir del valor retornado por la rutina.

Algo similar sucede con la velocidad, en este caso, el rango original es de 0..30 millas por hora y el rango acotado queda definido entre los valores ± 2 a partir del valor retornado por la rutina implementada. Esto se ilustra en las figuras 4.25 (a) y (b).

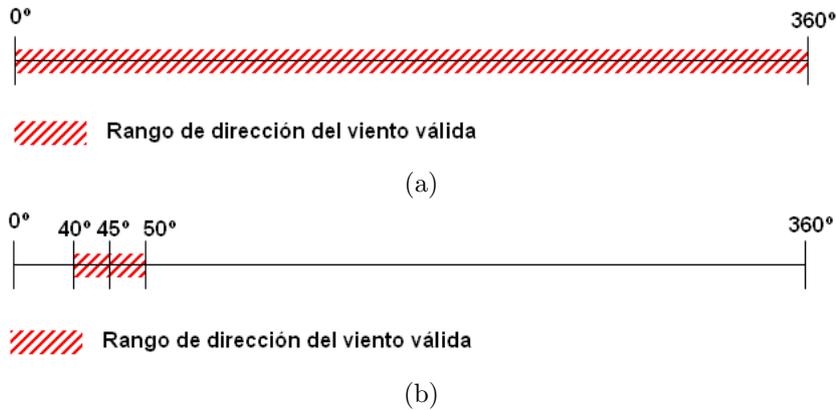


Figura 4.24: (a) Rango de variación de la dirección del viento sin guiar la búsqueda. (b) Rango acotado por el Método Computacional.

Como se puede ver en las figuras 4.24 (b) y 4.25 (b) acotar los rangos de variación de estos dos parámetros es útil ya que reduce el espacio de búsqueda, acelerando la convergencia hacia individuos que generarán simulaciones más cercanas a la realidad.

Una característica importante de este método es que es totalmente independiente del simulador y del marco de la aplicación donde se lo utiliza. Y a la inversa, también la forma de implementar este Método Computacional no está restringida ni al simulador ni a la aplicación que la contenga. Por esto, se puede ver este método como una caja negra, posibilitando así su modificación sin tener que modificar el resto de la aplicación, y a la inversa, se puede modificar su entorno sin ser necesario considerar cambios dentro de este método.

Esta característica es muy importante ya que permite a la aplicación “evolucionar” también, por ejemplo, cambiar el simulador, cambiar el lenguaje de programación, etc. También el Método Computacional puede mejorar independientemente de la aplicación [50].

Para verificar la correctitud del Método Computacional propuesto, se ha implementado el Método Analítico, el cual será explicado en las próximas secciones.

4.6.2. Método Analítico

En este caso, hemos estudiado y considerado la forma en que el simulador implementa las ecuaciones del modelo de Rothermel, y se utilizará es-

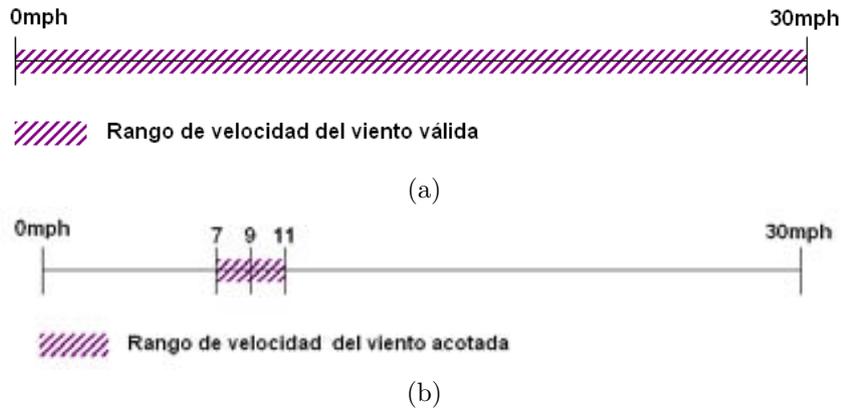


Figura 4.25: (a) Rango de variación de la velocidad del viento sin guiar la búsqueda. (b) Rango acotado por el Método Computacional.

te procedimiento de la forma más conveniente para descubrir los valores del viento ideales para cada uno de los individuos de la población que se esté evolucionando.

Entonces, se han agregado cálculos en el simulador, los cuales utilizan resultados parciales del mismo para retornar los datos que son útiles en nuestro método. Este método realiza estos cálculos para cada uno de los individuos: dependiendo de cómo propaga cada individuo, retorna el viento ideal que debería tener dicho individuo para lograr el avance real del fuego.

Entonces, dadas sus características se espera que este método sea más preciso que el Método Computacional, pero como es fácil notar, es totalmente dependiente del simulador. Por este motivo, no se lo puede ver como caja negra, y cualquier cambio en la aplicación, es probable que acarree cambios en el método.

Basado en el modelo de Rothermel, el simulador *fireLib* utiliza la dirección y velocidad del viento y los grados de inclinación y la orientación de la pendiente para determinar la dirección de máxima propagación del fuego. En el capítulo 2 se han presentado algunas de las ecuaciones de este modelo. A continuación se reescriben dos de ellas para lograr mayor claridad en la explicación del método:

$$R = \frac{I_r \xi (1 + \Phi_w + \Phi_s)}{Pb^\epsilon Qig} \quad (4.6)$$

La ecuación 4.6 determina el índice de propagación del fuego en base a la energía calórica liberada por el frente del fuego y la requerida por los com-

bustibles para arder. La ecuación 4.7 es la composición de las características del viento y de la pendiente que, juntas, determinan la dirección en la que ocurre la máxima propagación del fuego:

$$\Phi = \Phi_w + \Phi_s \quad (4.7)$$

En esta última ecuación, Φ es el vector resultante de la dirección de máxima propagación del fuego. Es el resultado de la combinación de los efectos del viento a altura de media llama (Φ_w) y el efecto de la pendiente (Φ_s) [21].

De una forma esquemática y simplificada, en la figura 4.26 (a) y (b) se muestra cómo se componen las características de la pendiente y del viento para determinar cómo se propaga el fuego (implementación de las ecuaciones 4.6 y 4.7).

Ahora bien, ¿cómo resuelve el simulador esta parte de las ecuaciones del modelo? Sea Φ_w el efecto del viento y β su dirección, Φ_s el efecto de la pendiente y α su orientación (figura 4.26 (a)). Entonces, se suman ambos vectores para componer un tercer vector que determina el ángulo de máxima propagación y el índice de propagación (en la figura 4.26 (b) δ y max respectivamente). Este nuevo vector tiene altura determinada por (ecuación 4.8):

$$altura = y = \text{sen}(\alpha) * \Phi_s + \text{sen}(\beta) * \Phi_w \quad (4.8)$$

y base igual a (ecuación 4.9):

$$base = x = \text{cos}(\alpha) * \Phi_s + \text{cos}(\beta) * \Phi_w \quad (4.9)$$

y el ángulo δ definido por estos dos vectores. En el Apéndice A se muestra paso a paso cómo se llega a estas ecuaciones.

El estudio exhaustivo del simulador nos permitió descubrir que estas ecuaciones se resuelven en diversas líneas de una operación del simulador. Esta operación es la que descubre la dirección e intensidad del fuego en la dirección de máxima propagación, justamente descrita anteriormente en el algoritmo 4.5.1 en este mismo capítulo. En dicho algoritmo, las líneas 25 a 33 resuelven estas ecuaciones.

Pero, como hemos dicho, en el simulador no se trabaja con direcciones absolutas, sino relativas (diferencia entre la orientación de la pendiente y la dirección del viento, línea 25 del algoritmo 4.5.1). La orientación de la pendiente es el origen, por lo que α es 0. Entonces, las ecuaciones que se implementan son:

$$altura = y = \text{sen}(\beta) * \Phi_w \quad \text{ya que } \text{sen}(\alpha) = 0 \quad (4.10)$$

y:

$$base = x = \Phi_s + \text{cos}(\beta) * \Phi_w \quad \text{ya que } \text{cos}(\alpha) = 1 \quad (4.11)$$

La ecuación 4.10 es implementada por la línea 29 del algoritmo 4.5.1 y la ecuación 4.11 es implementada por la línea 28 del mismo algoritmo. Es útil notar ahora que β entonces ya no es la dirección del viento, sino la diferencia entre ésta y la orientación de la pendiente.

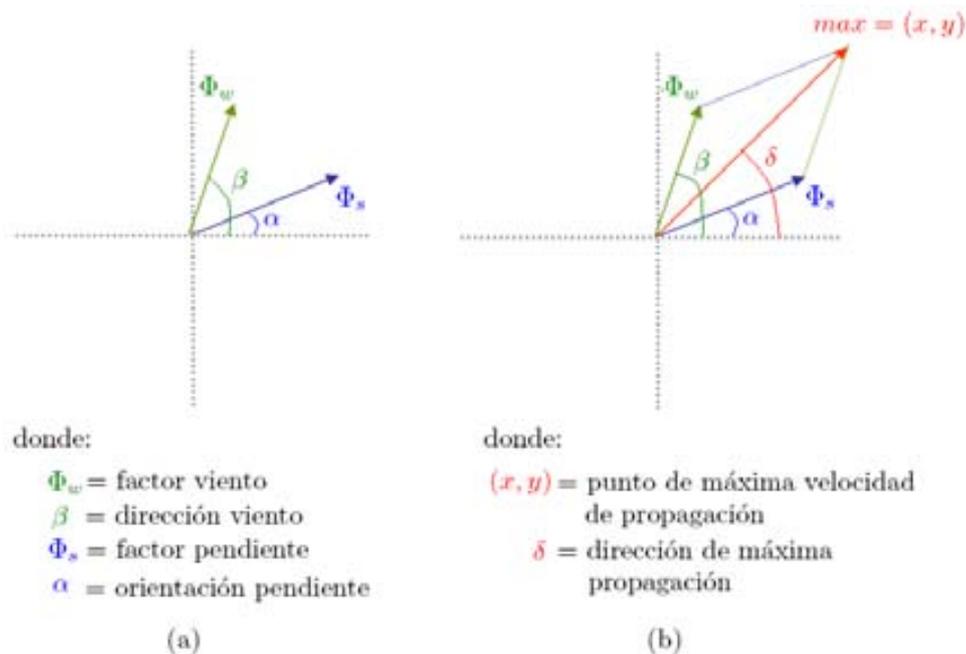


Figura 4.26: (a) Vectores del viento y de la pendiente. (b) Composición del vector de propagación del fuego a partir de la combinación del viento y de la pendiente.

Entonces, ¿cómo se utiliza esto para definir nuestro Método Analítico? Como muestra la figura 4.27 (a) en el instante t_{i+1} , momento en que se realizará la etapa de ajuste, se dispone de los datos de la pendiente y, luego del análisis del mapa real, se dispone del conocimiento de la propagación del fuego en t_{i+1} . A partir de estos datos, sabiendo cómo se obtiene el ángulo de máxima propagación (figura 4.26 (b)) se puede determinar el efecto y la dirección del viento necesarios para llegar a la propagación real (figura 4.27 (b)).

Entonces, a partir de las ecuaciones 4.8 y 4.9, es posible despejar los datos del viento, ya que se conocen los datos de la pendiente (Φ_s y α) y

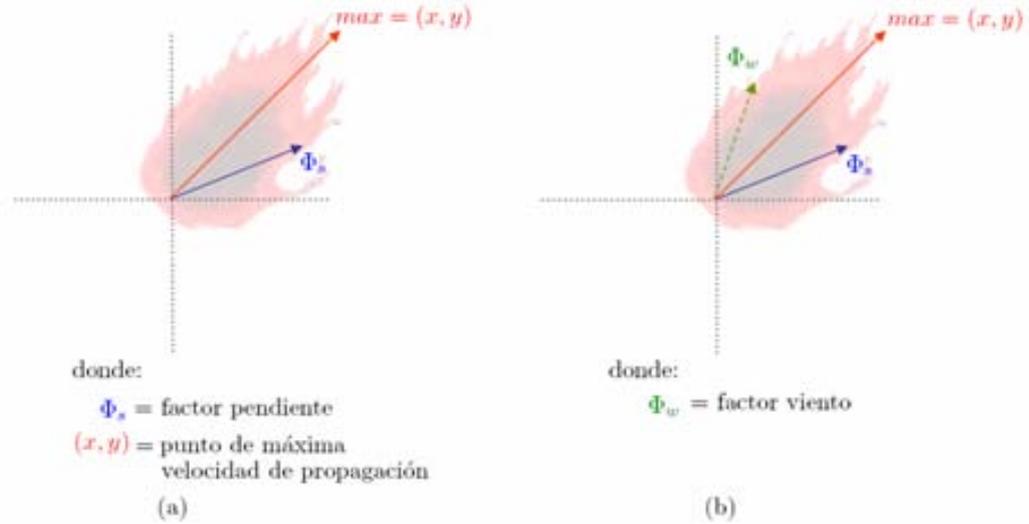


Figura 4.27: (a) Conocimiento disponible: datos de la pendiente y propagación real. (b) Obtención de las características del viento ideal.

los datos de la propagación (x , y y δ). Así, se despejan y se obtienen las características del viento: su efecto (Φ_w , ecuación 4.12) y su dirección (β , ecuación 4.13):

$$\Phi_w = \frac{x - \Phi_s \cos(\alpha)}{\cos(\beta)} \quad (4.12)$$

$$\beta = \arctan\left(\frac{y - \Phi_s \sin(\alpha)}{x - \Phi_s \cos(\alpha)}\right) \quad (4.13)$$

De esta forma, obtenemos las características del viento que serán utilizadas para guiar las operaciones del algoritmo genético. El valor de Φ_w , será utilizado para obtener la velocidad del viento (en millas por hora) que se debería utilizar como parámetro para llegar al efecto del viento Φ_w en la simulación. La secuencia de pasos para llegar a los resultados anteriores se encuentran explicados en el Apéndice B.

Las figuras y las ecuaciones están esquematizadas y simplificadas. Es importante considerar que tanto el efecto del viento como de la pendiente están atenuados por factores que dependen del tipo de combustible, humedades del combustible muerto, humedad del combustible vivo, etc. El

cálculo de las ecuaciones (4.12) y (4.13) está representado trigonométricamente, sin incluir los cálculos que hace el simulador al modelizar a través de vectores estos fenómenos (viento, pendiente y propagación del fuego). Además no está incluida la propagación sin considerar viento y pendiente que también influyen en la propagación final obtenida.

En resumen, el Método Analítico utiliza fórmulas y ecuaciones que realiza el simulador, pero en el orden y forma convenientes para obtener los datos del viento necesarios para guiar la búsqueda durante las evoluciones del algoritmo genético.

En este método, al igual que en el Método Computacional, el análisis del mapa de propagación real se realiza una sola vez al comienzo de la etapa de ajuste. Pero el cálculo del viento ideal en el Método Analítico se hace por cada uno de los individuos, aprovechando de esta forma resultados parciales (por ejemplo, propagación del individuo sin viento y sin pendiente) para determinar los valores del viento para cada individuo en particular. Estos valores del viento calculados, se almacenan con cada uno de los individuos como atributos del mismo.

Mientras que el Método Computacional considera una posible falta de completitud de la base de conocimiento, el Método Analítico busca, para cada uno de los individuos, los valores ideales que el viento debería tener para propagar como el fuego real. El Método Analítico entonces promete ser más preciso que el Método Computacional.

Por otro lado, el Método Analítico realiza operaciones propias dentro de las funciones del simulador *fireLib*. Esto lo hace totalmente dependiente del simulador, siendo imposible su uso con cualquier otro simulador (por más que también se base en las ecuaciones del modelo de Rothermel).

Por estas características se asume que el comportamiento de ambos métodos es comparable, y aunque se espera mayor precisión en los resultados del Método Analítico, los mismos serán utilizados como comprobación y verificación del correcto funcionamiento del Método Computacional.

La última sección de este capítulo contemplará distintas formas de realizar la guía en nuestro algoritmo genético. Para ambos métodos (Computacional y Analítico) se han implementado distintas configuraciones para realizar dicha guía, con el propósito de evaluar distintos caminos que puedan mejorar los resultados.

4.7. Inyección de datos

Una vez explicados los métodos propuestos para guiar de forma dinámica el algoritmo genético, se expondrán las distintas formas en que el conocimiento es inyectado en dicho algoritmo durante la etapa de ajuste.

Principalmente, hemos desarrollado dos alternativas para inyectar este conocimiento.

El primero, surge a partir de la idea de intentar disminuir aleatoriedad en las operaciones del algoritmo genético, agregando valores para los parámetros que sabemos que serán útiles, en vez de valores aleatorios.

Recordando las operaciones del algoritmo genético, en la mutación se “cambia” el valor de un gen por otro valor aleatorio válido. Entonces, una forma de guiar al algoritmo sería instrumentar esta operación de mutación para disminuir la aleatoriedad y forzar la asignación de valores útiles.

De esta forma, durante la mutación, se cambian los valores de todos los parámetros de forma habitual, excepto los valores de velocidad y dirección del viento, que son guiados por los valores hallados como ideales para estos mismos (esto es, utilizando el Método Computacional o el Método Analítico). Durante los próximos párrafos se describirá primero cómo se inyectan los datos en el Método Computacional y luego en el Método Analítico.

4.7.1. Método Computacional

La figura 4.28 ilustra cómo se utiliza el método computacional en nuestra aplicación: antes de comenzar con la evolución de la población, se analiza el mapa real y se obtienen las principales características de la evolución del mapa (parte (a) en la figura). Luego, se busca en la base de conocimiento el incendio con las características más parecidas a la propagación real y se obtienen los valores del viento observado en dicho incendio (b). Posteriormente, los valores del viento son almacenados, para que, durante la evolución de la población (c), cada vez que tiene lugar la operación de mutación para la dirección y la velocidad del viento, se utiliza un valor aleatorio dentro del subrango acotado por los valores del viento ideales encontrados en la base de conocimiento (d).

4.7.2. Método Analítico

La inyección de conocimiento ocurre diferente cuando se utiliza el Método Analítico. La figura 4.29 ilustra esquemáticamente los pasos de la utilización de dicho método.

En este caso, también se analiza el mapa real antes de comenzar con la evolución de la población inicial. Entonces, se obtiene el punto de máxima propagación del fuego real (punto (x, y)). La figura 4.29 (a) muestra este paso.

Entonces, durante la evolución, en el momento en que la población es evaluada, se obtiene la simulación con un determinado individuo (parte (b)

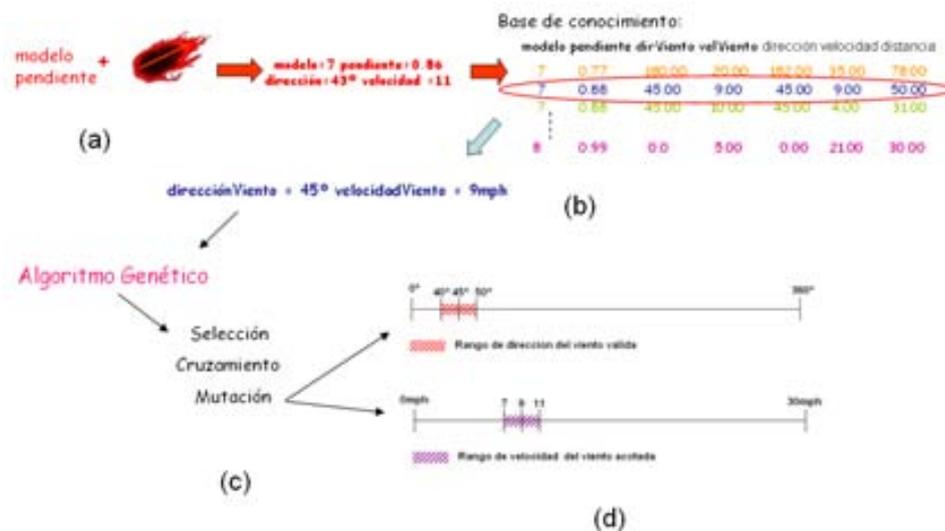


Figura 4.28: Operación del Método Computacional.

de la figura), y luego de realizar la simulación, se aprovechan diversos cálculos que el simulador ha realizado y se calculan los datos ideales del viento utilizando las ecuaciones 4.12 y 4.13 (parte (c) en la figura). De esta forma, el cálculo de los valores ideales del viento se realiza específicamente para cada individuo.

Luego, los valores de velocidad y de dirección del viento calculados por el Método Analítico son almacenados junto con cada individuo. Posteriormente, en el momento de la mutación, si se desea guiar la búsqueda, estos valores son asignados como valores del viento ((e) en la figura). En este caso, como los valores se han calculado para cada uno de los individuos, se espera que la guía sea más exacta, y no se utiliza un subrango de todo el rango válido, sino que se utilizan como valores absolutos durante la guía.

Pero, en este punto vale la pena considerar una característica importante que surge a partir del orden en que se ejecutan las operaciones del algoritmo genético: se evalúa la población (esto es, aquí se calculan los valores ideales del viento para cada uno de los individuos) y luego, se aplican los operadores genéticos, esto es: elitismo (si se utiliza), selección, cruzamiento y mutación.

Esta secuencia de operaciones, determina que si se guía la mutación, los

valores ideales de cada individuo, son asignados a los hijos creados mediante la operación de cruzamiento, y no a los padres (para los que realmente fueron calculados). Esto es porque la mutación se realiza después del cruzamiento. Entonces, no se puede garantizar que los valores que son buenos para los individuos elegidos como padres durante la selección, seguirán siendo buenos para los hijos.

Debido a esto, hemos implementado otra forma de guiar la búsqueda: en vez de guiar la operación de mutación, se guía la operación de elitismo (figura 4.29 (d)), de esta forma, forzamos a que los valores ideales de viento calculados para un individuo determinado, sean asignados a dicho individuo durante la operación de elitismo, y no a un hijo.

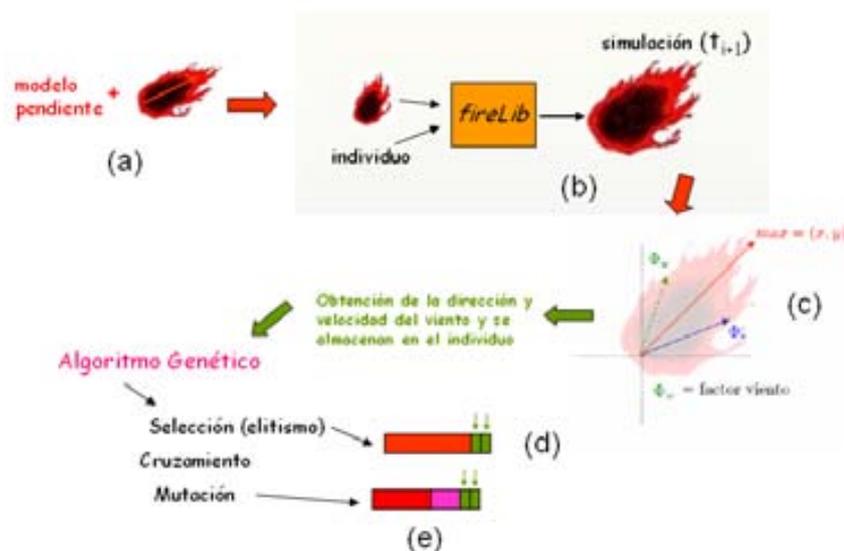


Figura 4.29: Operación del Método Analítico.

Ambas variaciones para el Método Analítico fueron implementadas y desarrolladas, y los resultados serán mostrados en el próximo capítulo.

Existen muchas configuraciones distintas, tanto para el algoritmo genético, como así también para la forma de realizar las guías. El próximo capítulo, intentará mostrar los resultados más significativos que hemos obtenido

utilizando estas técnicas para guiar y mejorar la calidad de las predicciones de incendios forestales.

Capítulo 5

Resultados Experimentales

5.1. Introducción

Durante los primeros capítulos de este trabajo, se han introducido brevemente las características del fuego, la problemática que significan actualmente los incendios forestales, luego, se trataron principalmente, los simuladores del comportamiento del fuego.

Además, se ha explicado la forma de trabajo que se ha adoptado, y las mejoras propuestas que son la principal contribución de este trabajo. Estas mejoras no trabajan ni modifican el modelo, ni el simulador, sino que se basan en la elección de los valores para los parámetros de entrada del simulador utilizado.

En este capítulo se detallarán las experimentaciones realizadas y los resultados obtenidos mediante dicha experimentación. A lo largo del trabajo se han evaluado distintos aspectos y configuraciones de la aplicación. Por esto, en cada prueba realizada se explicarán la configuración del algoritmo genético, los objetivos a evaluar, y se realizará un breve análisis de los resultados obtenidos.

5.2. Características de la Implementación

Para desarrollar nuestro sistema de predicción en dos etapas, se ha desarrollado una aplicación *master/worker*, donde todos los procesos *workers* ejecutan el mismo código sobre distintos datos.

Para desarrollar el código de los distintos procesos se ha utilizado el

lenguaje de programación C y para la creación y manejo de procesos, como así también para resolver las comunicaciones entre los mismos, se ha utilizado la librería MPI (*Message Passing Interface* [23] [34], utilizando específicamente en nuestro caso MPICH).

Arquitecturas Utilizadas

A lo largo de este trabajo, se utilizaron dos clusters, uno para desarrollar el código del sistema y realizar diversas experimentaciones, y el otro para evaluar exclusivamente el rendimiento de la aplicación. A continuación se enumeran las principales características del cluster utilizado para la evaluación del rendimiento de la aplicación desarrollada.

- Nodo front-end:
 - HP dc7100sff
 - Procesador Intel Pentium 4 @3.0GHz 1MB L2
 - 1 GB DDR
 - HD 80 GB
 - Tarjeta de red Broadcom NetXtreme
- Nodo File server:
 - HP dc7100sff
 - Procesador Intel Pentium 4 @3.0GHz 1MB L2
 - 1 GB DDR
 - 2xHD 200 GB
 - Tarjeta de red Broadcom NetXtreme
- 32 nodos de cómputo:
 - HP dc 7100sff
 - Procesador Intel Pentium 4 @3.0GHz 1MB L2
 - 1 GB DDR
 - HD 80 GB
 - Tarjeta de red Broadcom NetXtreme

Dicho cluster utiliza un sistema de colas para el manejo de procesos, por lo que mediante esta cola, cada aplicación accede a los recursos que necesita de forma exclusiva. De esta forma, es posible evaluar el rendimiento de la aplicación sin sufrir interrupciones en la ejecución debido a otros procesos

concurrentes.

A través de la experimentación evaluaremos principalmente dos aspectos: la mejora de la calidad de las simulaciones (disminución del error) como así también el tiempo de respuesta de la aplicación.

Tipos de Mapas Utilizados

Para la experimentación se han utilizado dos tipos de mapas: mapas de quemas reales y mapas sintéticos.

En nuestro caso, las quemas reales son quemas prescritas, quemas organizadas y controladas para las cuales existió algún tipo de monitoreo para seguir su progreso y poder utilizarlo, como en nuestro caso, post mórtem.

Los primeros casos, las quemas reales, son quemas pertenecientes al proyecto SPREAD [40] desarrollado en Gestosa (Portugal) durante los años 2002 al 2004. El terreno está ubicado sobre la ladera de la Serra de Lousã, con altitud entre 800 y 950 metros sobre el nivel del mar. El objetivo de dicho proyecto fue recojer datos experimentales para dar soporte al desarrollo de nuevos conceptos y modelos y para validar métodos ya existentes [10]. La figura 5.1 muestra el terreno dividido en distintos *plots* para las diferentes experimentaciones.

Algunas de las quemas realizadas en este proyecto han sido filmadas en vídeo y posteriormente procesadas, con el fin de extraer los frentes intermedios de la evolución del incendio a intervalos de tiempos regulares (aproximadamente 2 minutos) [10].



Figura 5.1: Vista del área utilizada en Gestosa por el proyecto SPREAD.

Por otro lado, los fuegos sintéticos son mapas de fuegos que se obtienen utilizando algún tipo de simulador de comportamiento del fuego. En

este caso, se realiza la simulación de un fuego inicial y se lo deja progresar. Luego, se analiza el mapa que se obtiene como salida del simulador y se lo utiliza como si fuera el mapa de un incendio real.

Las experimentaciones que se realizaron en este trabajo que utilizan mapas sintéticos, son progresos de fuego obtenidos utilizando el simulador *fireLib*. De esta forma, existen diversas ventajas frente a la utilización de cualquier otra herramienta para obtener este tipo de mapas: se cuenta con el conocimiento suficiente sobre este simulador, sus entradas y salidas, etc.

Antes de cada experimentación se darán las características del mapa utilizado, de la configuración del algoritmo genético, de las poblaciones utilizadas, de la cantidad de pasos de evaluación, etc., ya que estas características varían dependiendo del objetivo de cada una de las experiencias.

5.3. Predicción Clásica vs Predicción en Dos Etapas

La primer experimentación mostrará los resultados de utilizar la predicción clásica, y luego, se compararán estos resultados con los errores producidos al utilizar la predicción en dos etapas (sin aplicar métodos de sintonización de parámetros).

El objetivo de esta experimentación es evaluar las ganancias de utilizar una prebúsqueda de valores para los parámetros de entrada.

Para la predicción clásica se utilizan valores para los parámetros que más concuerdan con el conocimiento disponible del entorno del fuego. Estos son, entonces, valores que se obtienen de la observación del entorno, o datos que se tienen del lugar, sensores, etc., pero dichos valores no reciben ningún tipo de procesamiento.

Esta prueba se realizó utilizando un mapa obtenido a partir de una quema prescrita perteneciente al proyecto SPREAD (explicado en párrafos anteriores), un mapa con dimensiones de 60 columnas por 90 filas de un pie² cada celda. El terreno contaba con una inclinación de 6° y el avance del fuego fue dividido en intervalos de 2 minutos cada uno.

Entonces, se realizan las simulaciones para cada intervalo de tiempo en que se divide el progreso total del fuego. Los errores hallados para cada paso de simulación se muestran en la figura 5.2, donde el eje horizontal muestra cada intervalo de tiempo analizado, mientras que el eje vertical muestra el error obtenido para cada uno de los pasos de simulación.

La figura 5.2 nos permite ver que los errores producidos en los distintos pasos de simulación varían en el rango de 0,4 a 0,65. Para dar una idea de lo que esto significa, recordamos la fórmula del error empleada (explicada

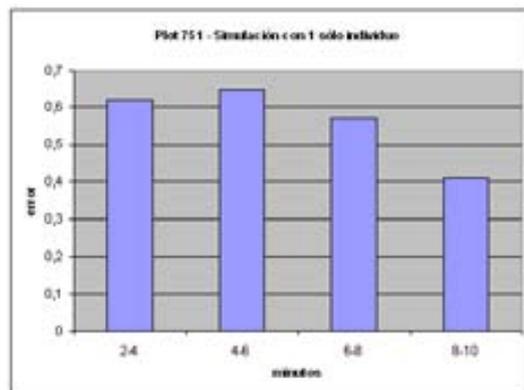


Figura 5.2: Errores obtenidos utilizando la simulación clásica.

en la sección 4.1.2):

$$Error = \frac{(\cup - Inicial) - (\cap - Inicial)}{Real - Inicio} \quad (5.1)$$

dimensión que, sólo para dar una idea de su magnitud, un error igual a 1 es que la simulación difiere del progreso real del fuego exactamente el tamaño del fuego real.

La aplicación del método en dos etapas se puede considerar como un pre-procesamiento de los parámetros de entrada. La búsqueda que realiza el algoritmo genético, buscando minimizar la función de error de las simulaciones permite ya por sí sólo encontrar buenas soluciones para el problema.

La figura 5.3 muestra esta característica, comparando los resultados anteriores (de la simulación clásica) y los resultados cuando se aplica el algoritmo genético sin guiar la búsqueda (sin aplicar los Métodos Computacional o Analítico).

En este caso (como en otros que mostraremos a continuación) se muestra el promedio de 3 poblaciones iniciales distintas para el método de dos etapas, para tener una representación más real del comportamiento de este esquema.

Como se puede ver en la figura, la reducción del error es muy importante cuando se aplica el método propuesto. Las ganancias observadas para cada paso son: 88,77% para el período del minuto 2 al 4, 88,20% para el período del minuto 4 al 6, 77,78% período 6-8 y finalmente, un 64,22% en el período del minuto 8 al 10. Estos porcentajes son las reducciones del error considerando a la simulación clásica como el 100%.

Esta primer experimentación muestra cómo el método en dos etapas

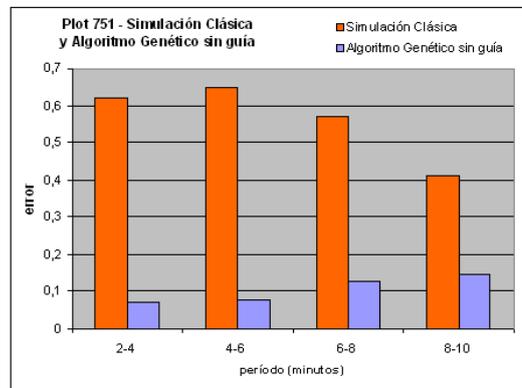


Figura 5.3: Errores obtenidos utilizando la simulación clásica y aplicando el método propuesto sin guiar la búsqueda.

propuesto logra mejorar los valores de los parámetros de entrada, logrando mejorar los resultados frente al método de predicción clásica.

Pero, como ya se ha mencionado, la contribución principal de este trabajo son los métodos para guiar la búsqueda del algoritmo genético, para mejorar la búsqueda realizada durante la etapa de ajuste. Los próximos casos de estudio comparan directamente la predicción en dos etapas cuando se sintoniza la búsqueda comparándolo con el método cuando ninguna sintonización es realizada.

5.4. Caso de estudio 1

El primer caso que se analizará corresponde a una de las experimentaciones evaluadas en [18]. La inclusión de esta prueba dará la posibilidad de comparar cómo fue “evolucionando” la aplicación a medida que se ha trabajado en ella, y cómo estos cambios han logrado mejorar los resultados obtenidos.

Esta primer experimentación tuvo como principal objetivo evaluar las ventajas de utilizar una búsqueda guiada frente a no guiar la búsqueda, es decir, se ejecuta el método de predicción en dos etapas, y, durante la etapa de ajuste se aplican los métodos de sintonización de parámetros.

Las principales características de esta primer experimentación son:

- Población: se utiliza una población inicial de 50 individuos donde los valores para los parámetros son aleatorios con distribución uniforme

en cada rango válido de variación. Esto intenta garantizar variabilidad en la población inicial.

- Cantidad de generaciones: a cada población se la evoluciona 5 generaciones. El número de evoluciones se ha fijado en 5 porque distintas experimentaciones permitieron observar que durante las primeras iteraciones el algoritmo genético converge hacia buenos individuos, después de la 5^{ta} evolución no se logra una reducción del error significativa que justifique seguir iterando. En [18] se muestra un análisis de dicha convergencia.
- Probabilidad de mutación: en este caso, se utiliza una probabilidad de 1% para realizar la operación de mutación. Así, la mutación ocurre con muy bajas probabilidades para evitar que el algoritmo genético se convierta en un búsqueda aleatoria (aunque más adelante en otra experimentación se aumentará esta probabilidad cuando se guíe dicha operación).
- Elitismo: se realizaron diversas pruebas las cuales permitieron notar la ventaja de utilizar elitismo durante la operación de selección. La inclusión de esta variante en la operación de selección, permite siempre utilizar los mejores individuos de la generación anterior, por lo que en cada iteración, los individuos hijos serán igual o mejores que sus padres. En cambio si no se utiliza, esto no queda garantizado. Debido a esta ventaja, en las pruebas utilizadas en este capítulo se incluye directamente esta variación durante la selección.

Dadas las características del algoritmo genético y la naturaleza aleatoria de algunas de sus características (población inicial y operación de mutación), se utilizan los resultados de 5 poblaciones distintas y en los gráficos se muestran los promedios de la utilización de estas 5 poblaciones. De esta forma, creemos caracterizar de mejor manera el desempeño del algoritmo genético y de los métodos de sintonización de parámetros.

El mapa utilizado para la primer experimentación corresponde a un mapa sintético, obtenido utilizando el simulador *fireLib*. El avance del fuego para dicho mapa se muestra en la figura 5.4.

Como se puede ver en la figura, el fuego dura 12 minutos, definiendo que los pasos evaluados durante la etapa de ajuste serán los períodos 2-4, 4-6, 6-8, y 8-10 (minutos), mientras que la predicción se hará para los períodos 4-6, 6-8 y 8-10. Vale la pena mencionar que al terminar cada paso de simulación, los individuos evolucionados de dicha evolución se utilizarán como entrada para el siguiente paso de simulación. Esto es, para el paso 2-4 se utiliza una población inicial con valores aleatorios, pero a partir del paso 4-6 y los sucesivos, la población inicial será la población evolucionada del

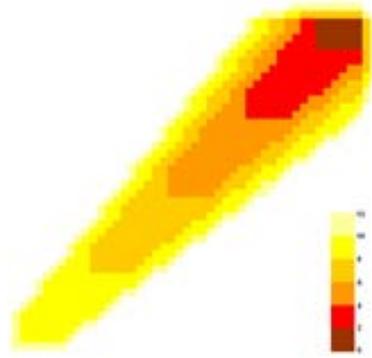


Figura 5.4: Avance del fuego en el mapa sintético utilizado para el caso de estudio 1.

paso anterior (aprovechando los valores encontrados).

Además, el mejor individuo encontrado en cada etapa de ajuste, es el que se utiliza como entrada para la etapa de predicción. Esta forma de trabajar asume la hipótesis de que las características del entorno se mantendrán parecidas en los períodos de tiempos involucrados en cada paso ajuste-predicción.

A continuación, se mostrarán los resultados para el método con las siguientes configuraciones: método sin guía, utilizando el Método Computacional donde se guía la operación de mutación y utilizando el Método Analítico, donde se guía tanto la operación de elitismo como así también guiando la operación de mutación.

Las figuras 5.5 y 5.6 muestran la comparación de los errores para cada uno de los métodos. Nuevamente, y formato que se mantiene para mostrar los errores de las distintas experimentaciones, en el eje horizontal se muestran los distintos pasos de simulación y el eje vertical muestra el error obtenido (observar que ambos gráficos difieren en la escala utilizada para estos errores). Cada una de las series representa una configuración distinta de la aplicación: el algoritmo genético cuando no se guía la búsqueda, cuando se guía utilizando el Método Computacional o cuando se lo guía utilizando el Método Analítico (sintonizando el elitismo o la mutación).

Se analizarán muy brevemente estos resultados ya que dicho análisis se ha realizado en el trabajo [18]. Entonces, observamos que:

1. Cualquier método utilizado para guiar la búsqueda (Computacional o Analítico) reduce el error cuando se lo compara con el método sin guiar la búsqueda.

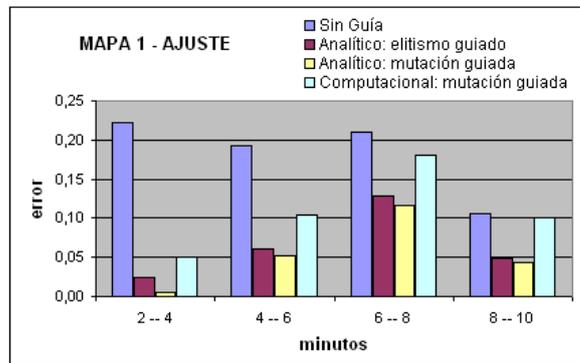


Figura 5.5: Resultados etapa de ajuste.

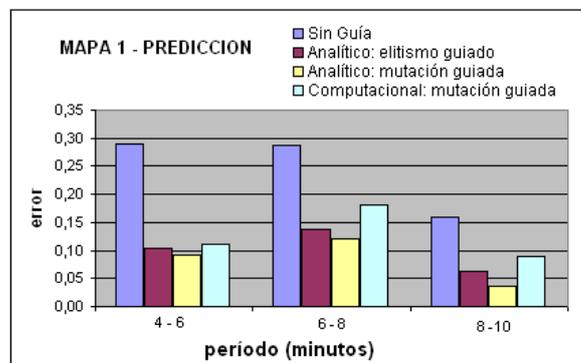


Figura 5.6: Resultados etapa de predicción.

2. Los errores alcanzados utilizando el Método Computacional o el Analítico no difieren en gran medida entre ellos.
3. Observando la característica expuesta en el punto anterior, el Método Analítico logra reducir aún más el error cuando se lo compara con el Método Computacional.

Entonces, la primer observación para este mapa muestra que se logra reducir el error tanto en la etapa de ajuste como así también en la etapa de predicción cuando algún tipo de guía es introducida durante la etapa de calibración, respecto a no guiarla en absoluto.

La segunda observación permite verificar el correcto funcionamiento del Método Computacional a través de la comparación con el Método Analítico,

ya que sus comportamientos siguen un patrón de ajuste/predicción similar.

La inclusión de esta prueba permite observar que el método funciona correctamente y que los métodos sirven para fuegos sintéticos. Para fuegos reales, es más difícil lograr este tipo de resultados, ya que nunca sabemos, si, por ejemplo, el método genético podrá llegar a encontrar un individuo que justamente propague igual (o, al menos, similar) a la propagación real.

La utilización de 50 individuos en esta prueba es debido a que se intenta evitar que la etapa de ajuste se convierta en un cuello de botella para la aplicación. De esta forma, hemos utilizado poblaciones pequeñas, ya que hay que tener en cuenta que dicha población se la evoluciona una determinada cantidad de veces y que cada evolución significa una simulación con cada individuo (incrementando así el tiempo de cómputo).

Entonces, estos resultados nos servirán como punto de partida para analizar los resultados de las demás experimentaciones, las cuales fueron obtenidas con la aplicación paralela desarrollada posteriormente.

5.5. Caso de estudio 2

El segundo caso de estudio corresponde a la utilización del algoritmo genético paralelo. Al distribuir el cómputo de las simulaciones entre los procesos *workers* se han utilizado poblaciones más grandes, de 500 individuos para realizar esta segunda experimentación.

El mapa utilizado en esta experimentación corresponde a otro mapa real, una quema prescrita del proyecto antes mencionado SPREAD. El mapa utilizado muestra un progreso del fuego como se puede observar en la figura 5.7. Este es el mapa utilizado en la sección 5.3 de este capítulo, donde se comparan los resultados utilizando la predicción clásica y la predicción en dos etapas.



Figura 5.7: Mapa del progreso del fuego del *plot* 751.

Se han evaluado distintas configuraciones para el algoritmo genético y los métodos de sintonización de parámetros. Se evaluarán las distintas configuraciones, mostrando el error para cada ejecución.

Las principales características de estas pruebas son:

- Se promedian los resultados utilizando 3 poblaciones distintas, a fin de describir de mejor forma el comportamiento de los métodos.
- Cada población inicial tiene 500 individuos, con valores aleatorios uniformemente distribuidos en cada rango de variación válido.
- Cada vez que se aplica el paso de ajuste, la población es evolucionada un total de 5 veces.
- Se han evaluado el Método Computacional como así también el Método Analítico, variando la asignación de valores en la mutación y en el elitismo.
- En los casos donde se guía la mutación o el elitismo, se utilizan distintas configuraciones, las cuales, serán explicadas en el momento de analizar los resultados.

El objetivo de las próximas figuras es mostrar la ganancia de la aplicación de los diferentes métodos para guiar la búsqueda, mostrando también distintas configuraciones para cada uno.

Los resultados mostrados en la figura 5.8 tienen la particularidad de utilizar una población aleatoria como entrada en el paso (minuto) 2 al 4 y luego, para el resto de los pasos se utiliza como entrada la población evolucionada del paso anterior. De esta forma, se intenta utilizar los valores hallados durante el paso de ajuste del paso previo, asumiendo que las características que han sido buenas para un paso, mantengan esta cualidad para un tiempo posterior.

Para cada uno de los períodos de tiempo analizados, la serie mostrada en azul (primera serie de cada grupo) es el método sin guiar la búsqueda, la cual sirve como referencia para analizar los métodos de sintonización propuestos. Las siguientes series muestran los resultados de la aplicación del Método Computacional, donde la diferencia reside en la probabilidad de mutación: 20% y 40% de probabilidad.

A su vez, para el Método Analítico cuando se guía el elitismo, no se hace bajo una probabilidad, pero se utilizan dos formas distintas de realizar la sintonización: una es guiando cada uno de los individuos copiados con el elitismo (esto sería, el 100% de los casos), indicado en la figura como "no gradual". Y la otra forma que se propone es guiar el elitismo bajo una probabilidad y esta probabilidad cambia de forma gradual acorde avanzan

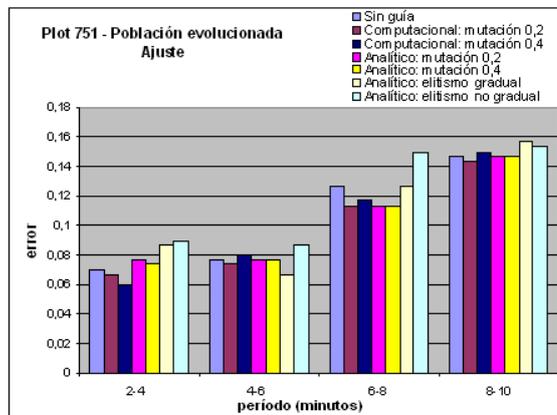


Figura 5.8: Errores obtenidos para cada método durante el paso de ajuste

las evoluciones. Se comienza con una probabilidad baja (para permitir mayor diversidad al comienzo) y se termina con una probabilidad más alta de sintonización (elitismo “gradual”).

Las figuras 5.9 y 5.10 tienen el mismo formato que las figuras anteriores, pero esta vez, varían las poblaciones de entrada para cada paso de simulación. En la figura 5.9 se utilizan poblaciones aleatorias para todos los pasos, a su vez, en la figura 5.10 se muestran los resultados utilizando una población con individuos evolucionados (50% de los individuos) y los restantes son individuos con valores aleatorios.

Considerando los resultados mostrados en las figuras 5.8, 5.9 y 5.10 se puede observar que:

- El Método Computacional logra, en la mayoría de los casos, reducir aún más el error cuando se lo compara con el método sin guiar la búsqueda.
- El Método Analítico, con la opción de guiar la mutación logra, en la mayoría de los casos, reducir el error al compararlo con el método sin guiar la búsqueda.
- El Método Analítico con elitismo guiado, la mayoría de los casos no logra mejorar al método sin guiar la búsqueda. En este caso, entonces, forzar la asignación de los valores de los mejores individuos no resulta del todo satisfactorio.
- La comparación de los resultados de las series correspondientes al Método Computacional y al Método Analítico muestra que son bas-

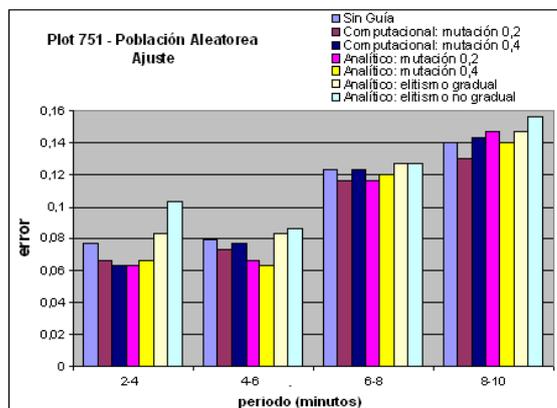


Figura 5.9: Errores obtenidos utilizando en cada paso poblaciones aleatorias.

tante similares, que en ningún caso se produce un comportamiento inesperado ni diferente entre ambos métodos. Esto muestra que el comportamiento del Método Computacional es correcto para estos casos de prueba.

- Por otro lado, las distintas probabilidades de mutación utilizadas, no muestran una tendencia precisa, por lo que para este mapa, no se puede concluir que una probabilidad es mejor que otra.

Entonces, es posible observar que el Método Computacional logra reducir los errores, que aunque sea en pequeña medida, estamos partiendo de la base de un rango de errores muy pequeños, por lo que sigue siendo significativa. Pero además de esto, los métodos para guiar la búsqueda han mostrado una convergencia más rápida hacia individuos con buena calidad en las simulaciones. Esto lo trataremos en la sección 5.5.1, donde se hará un análisis de dicha convergencia.

Vale la pena mencionar, que los resultados del Método Computacional son los que nos interesan en su totalidad, ya que al ser este método independiente del marco de trabajo y del simulador, es el que permite seguir mejorando al método, por lo que nuestro mayor esfuerzo reside en este método. El Método Analítico surge como forma de desarrollar un método que nos sirva para poder evaluar el comportamiento del Método Computacional, pero con el cual, no se prevee seguir trabajando.

Teniendo en cuenta todos estos resultados en su conjunto (estos son, los mostrados en las figuras 5.8, 5.9 y 5.10), es posible observar un compor-

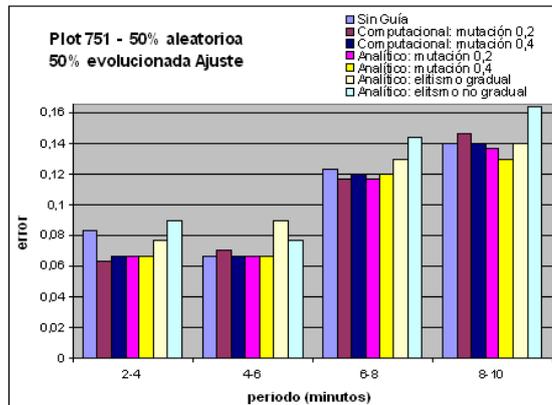


Figura 5.10: Errores obtenidos utilizando en cada paso poblaciones con 50 % de individuos aleatorios y 50 % de individuos evolucionados (excepto el período 2-4).

tamiento similar en la mayoría de los casos, que, a medida que se avanza en los pasos de simulación (minutos), se observa un pequeño aumento en los errores producidos por las simulaciones. Dicho comportamiento no fue observado en el caso de la predicción clásica (dichos casos son comparables, ya que se utilizó el mismo mapa para estas pruebas).

El análisis realizado a este comportamiento permite ver que:

1. El análisis del mapa (figura 5.7) a simple vista permite ver un progreso del fuego muy similar en los dos primeros períodos evaluados (2-4 y 4-6) mientras que después de estos períodos, el fuego avanza de forma más rápida sobre la superficie del terreno. Luego, el avance del fuego se detiene sutilmente, seguramente por algún cambio en las condiciones del ambiente o climáticas.
2. En la simulación clásica, es posible observar cómo los valores utilizados para realizar la simulación fueron favorables para los últimos períodos de tiempo y no así para los primeros dos períodos, donde el error de las simulaciones fueron mayores. Esto quiere decir que los valores elegidos para la simulación clásica, eran más adecuados para representar el entorno en los últimos períodos de simulación.
3. A su vez, al aplicar el paso de ajuste, guiando o sin guiar la búsqueda, se puede observar que el algoritmo genético encuentra mejores individuos para los primeros períodos evaluados, y no tanto para el nuevo comportamiento del fuego observado en los últimos pasos.

Lo anteriormente expuesto es un ejemplo de otros hallados a lo largo de este trabajo: el comportamiento del fuego es un fenómeno muy complejo de simular, y cada caso es particular, y dentro de casa caso, cada paso de simulación puede comprender distintos resultados.

5.5.1. Análisis de convergencia

El análisis de convergencia se hizo utilizando el error del mejor individuo de cada generación, para un único paso de simulación (período 2-4) y, en cada caso, promediando los errores de los mejores individuos de las 3 poblaciones utilizadas.

En la figura 5.11 se muestra la convergencia de los distintos métodos a medida que se avanza en las iteraciones del algoritmo genético. En el eje horizontal de dicha figura se muestran las evoluciones realizadas con la población, la evolución 0 es el error de la población inicial, donde no se han aplicado todavía los operadores genéticos (por lo tanto tampoco los métodos de sintonización de parámetros).

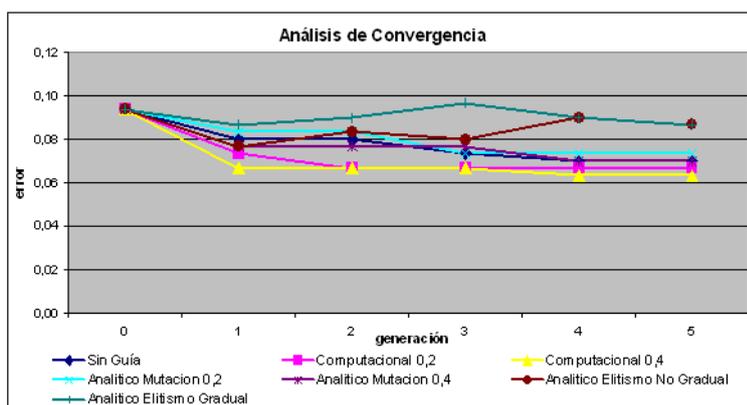


Figura 5.11: Convergencia de los distintos métodos (promedios de las 3 poblaciones).

Entonces, además de lograr reducir el error utilizando el Método Computacional, el método logra una convergencia más rápida a individuos con buena calidad, que logran reducir el error. Esta convergencia, hace que se puedan utilizar menos iteraciones (evoluciones) en el algoritmo genético y llegar a buenos resultados. De esta forma, se reduce también el tiempo total de ejecución.

En la gráfica, es posible observar cómo el algoritmo genético logra bue-

nos resultados desde la primera aplicación del Método Computacional. Con este método, se podría haber dejado de evolucionar en la generación 2, por ejemplo, ya que el error hallado no sería muy distinto al de la última generación utilizada. Esto no sucede con las demás configuraciones, en la mayoría, el error se sigue reduciendo sutilmente con más iteraciones.

Otra observación que se puede realizar es cuando se guía el elitismo: el error puede sufrir “subidas” y “bajadas” durante las evoluciones, esto es porque no se garantiza que el mejor individuo mantenga todas sus características originales. En cambio, cuando no se guía el elitismo, esto no sucede, siempre se mantiene el mejor de los individuos de generación en generación.

Entonces, considerando los gráficos de los errores y la convergencia observada durante las evoluciones lograda por los distintos métodos, se puede observar que se logra reducir el error en menos cantidad de evoluciones, logrando de esta forma reducir el tiempo total de ejecución de la aplicación.

5.6. Caso de estudio 3

El siguiente caso de estudio, se basa en el análisis de los tiempos y también del error logrado por los distintos métodos, utilizando el mapa sintético mostrado en la figura 5.12.

El objetivo principal de este caso de estudio es evaluar un mapa que tiene un aspecto muy distinto al de los casos anteriores, en este caso, el crecimiento del fuego es elíptico y de esta forma, se intenta descubrir cómo funcionan los métodos propuestos para estos casos. La figura 5.12 muestra el aspecto del progreso de dicho fuego sintético.

Las principales características de este caso de estudio son:

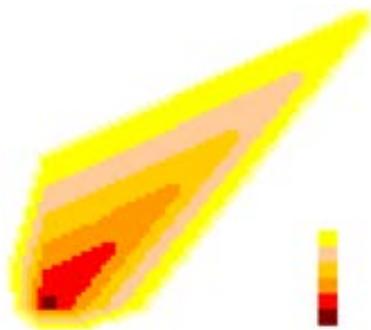


Figura 5.12: Progreso del fuego para el mapa utilizado en el tercer caso de estudio.

- Mapa sintético de 110 m², dividido en celdas de 1m².
- Promedio de 2 poblaciones de 400 individuos cada una.
- Se evoluciona 10 veces cada población.
- Se utilizan 1, 2 y 4 procesadores, para evaluar la escalabilidad del algoritmo.
- Se estudia y evalúan los errores encontrados para los distintos métodos.

Al evolucionar hasta 10 veces cada población, estamos aumentando el tiempo de procesamiento e intentando encontrar mejores resultados (minimizar el error). Se estudiará cómo influye esto en los tiempos y en la convergencia hacia buenas soluciones. A continuación se muestran los resultados obtenidos, primero se mostrará el error, la convergencia de los distintos métodos y luego los tiempos de ejecución.

En la figura 5.13 se muestran los errores encontrados para el método sin guiar la búsqueda y para el método guiando la búsqueda utilizando el Método Computacional (mutación guiada) y el Método Analítico cuando se guía la mutación y el elitismo.

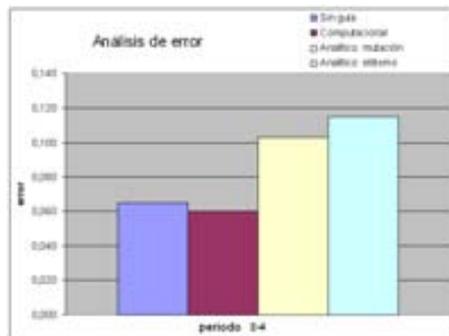


Figura 5.13: Errores producidos en el tercer caso de estudio.

Como se puede ver en la figura, los resultados del método sin guiar la búsqueda y del Método Computacional, son semejantes y aceptables (un error entre el 0.06 y el 0.12). A su vez, en este caso se incrementa el error para los dos casos del Método Analítico. Vale la pena mencionar que estos errores son los promedios de 6 errores distintos: utilizando 3 cantidades de *workers* distintos para cada una de las 2 poblaciones.

A su vez, la figura 5.14 muestra los errores (promedios) para cada una

de las generaciones (10 generaciones para este caso). En este caso, se puede apreciar nuevamente cómo el Método Computacional converge desde las primeras iteraciones hacia zonas con individuos con buena calidad en sus valores. De esta forma, este método logra acelerar la convergencia del método con gran rapidez, por lo que en este ejemplo también se puede concluir que con menos iteraciones se lograrían buenos resultados con una directa disminución de los tiempos de cómputo.

A continuación se muestran los tiempos donde se mantiene constante el

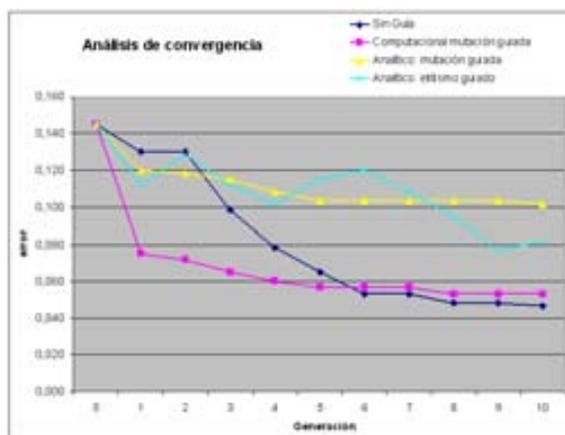


Figura 5.14: Convergencia de los distintos métodos utilizados para este caso de prueba.

tamaño de las poblaciones y se varía la cantidad de procesadores (procesos *workers*) utilizados (figura 5.15).

En la figura 5.15 se puede ver cómo al aumentar la cantidad de *workers* utilizados (1, 2 o 4, en el eje horizontal) el tiempo de procesamiento decrece de forma similar para cada uno de los métodos utilizados. Esta figura permite apreciar que todos los métodos logran una buena escalabilidad al utilizar más recursos de cómputo.

Aunque la cantidad de procesadores utilizados en esta prueba sea reducida teniendo en cuenta las configuraciones de los clusters actuales, es útil analizar el usuario final de una aplicación como la desarrollada. En nuestro caso, el usuario final pueden ser entidades de bomberos, de medio ambiente, gubernamentales, etc., las cuales no tienen por qué disponer de muchos nodos de cómputo a disposición para este tipo de utilidad en su infraestructura (de todas formas, la próxima prueba analiza la escalabilidad de la aplicación para clusters con más nodos).

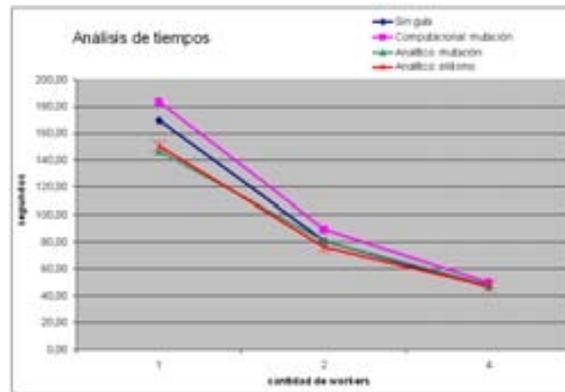


Figura 5.15: Análisis de tiempos variando la cantidad de *workers* utilizados.

Entonces, se observa que la escalabilidad de nuestra aplicación es buena para este tipo de entorno, donde la cantidad de recursos es limitado.

Además, al evaluar los tiempos de las distintas configuraciones para nuestra aplicación, se observa que estos tiempos son similares al utilizar el método sin guiar la búsqueda como cuando se la guía, con lo que podemos concluir nuevamente que no se penaliza el rendimiento de la aplicación al utilizar los métodos de sintonización de parámetros.

Además, el error encontrado se mantiene bajo, por lo que para este mapa elíptico, utilizar el Método Computacional mejora los resultados. En este caso particular, el Método Analítico no logró reducir el error al compararlo al método sin sintonización, aunque se mantiene de todas formas bajo.

5.7. Caso de estudio 4

Este caso de estudio tiene como objetivo principal analizar la escalabilidad de la aplicación paralela. Para esto, se utilizó un mapa real (el mismo mapa utilizado para la prueba 2, el *plot* 751). Esta vez, las principales características de las pruebas son:

- Se evalúa una sola población de 512 individuos (excepto que para la última se utilizaron poblaciones de 527 individuos).
- Dicha población se la evoluciona 5 veces cada vez.
- Se aumenta el tiempo de simulación, del minuto 2 al 10.

- Se utilizan 1, 2, 4, 8, 16 y 31 *workers* y se analiza el tiempo de cada ejecución.
- Se evalúa la aplicación cuando no se guía la búsqueda y cuando se aplica el Método Computacional.

El cluster utilizado dispone de 32 nodos, por lo que la última prueba se ha realizado con 31 *workers* (y no 32, como hubiese sido natural), para evitar que en un mismo nodo se ejecuten de forma concurrente el proceso *master* y uno de los *workers*, penalizando de esta forma el rendimiento general de dicha prueba.

La figura 5.16 muestra los tiempos de ejecución para el método sin guiar

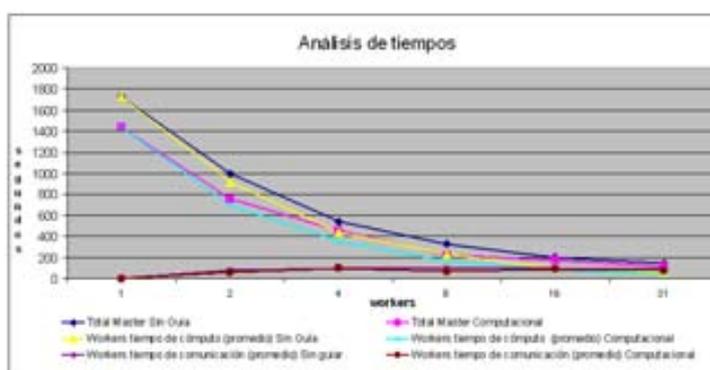


Figura 5.16: Análisis de los tiempos de los procesos *master* y *worker* cuando no se guía la búsqueda y cuando se utiliza el Método Computacional.

la búsqueda y para el Método Computacional. Se muestran los tiempos totales de ejecución para el proceso *master* y también el promedio de los tiempos de los procesos *workers* (cuando la cantidad de *workers* es mayor a uno). Además se muestran los tiempos promedios de comunicación en los procesos *workers*.

Entonces, los tiempos de procesamiento que se muestran en la figura permiten ver que la aplicación escala de forma correcta a medida que se incrementa el número de procesadores utilizados.

Por otro lado, los tiempos del método sin guiar y del Método Computacional son muy similares, por lo que se puede determinar que el Método Computacional no penaliza el rendimiento de la aplicación total.

Otro aspecto que se puede ver en la figura 5.16 son los tiempos de comunicación, el cual no superan (en promedio) los 100 segundos en ningún caso, de esta forma, se puede observar que en este caso las comunicaciones

no penalizan los tiempos totales de la aplicación. Por lo tanto, se puede concluir que no existen penalizaciones por el cómputo paralelo.

Estos casos de estudio intentan mostrar los resultados más representativos que hemos obtenido. Además, tratan de abarcar un rango amplio de distintas configuraciones y tipos de pruebas realizadas.

El siguiente capítulo abordará las conclusiones de todo el trabajo en su totalidad, teniendo en cuenta su desarrollo, las pruebas y los resultados mostrados en este capítulo.

Capítulo 6

Conclusiones y Líneas Abiertas

El objetivo principal de este trabajo es mejorar la predicción de la evolución de los incendios forestales. Para que una predicción sea realmente útil, es necesario que la calidad de la predicción sea alta (esto quiere decir, que la predicción refleje el crecimiento real del fuego), y que dicha predicción esté disponible en un período corto de tiempo.

Los resultados de una predicción son útiles para combatir el frente de fuego de forma eficiente durante un desastre, mejorar estrategias de reducción, o bien definir zonas de riesgos, definir planes de evacuación, etc.

En el transcurso de este trabajo se han tenido en cuenta estas características para mejorar métodos previamente propuestos para la predicción de la evolución del fuego en incendios forestales.

En las siguientes secciones se expondrán brevemente los métodos de predicción propuestos en esta línea de investigación, sus ventajas y puntos débiles, para de esta forma, enmarcar las principales contribuciones de este trabajo y las conclusiones a las que se han llegado.

6.1. Conclusiones y Observaciones

Como se ha ido mencionando a lo largo de este trabajo, la predicción clásica se basa en la utilización de un simulador, al cual se le aplica las entradas necesarias, y nos proporciona la evolución del fuego para un instante de tiempo posterior.

En este tipo de predicción, los valores de las entradas requeridas son los valores disponibles en el momento en que se inicia la simulación. Estos va-

lores pueden ser datos provenientes de algún dispositivo de monitorización, datos históricos, datos estimados, extrapolados, etc.

Una vez que se tienen todos los valores de entrada se ejecuta el simulador para obtener la predicción del avance del fuego.

Durante los primeros capítulos de este trabajo se hizo referencia a lo difícil que es disponer de estos valores correctos en el momento de un incendio. Esta incertidumbre en los parámetros de entrada resulta en predicciones incorrectas, por lo tanto, predicciones que no son útiles.

Los parámetros de entrada de una predicción clásica no tienen ningún tipo de procesamiento, se utilizan los valores “crudos” para obtener la predicción.

En trabajos anteriores ([1] y [10]) se propuso mejorar la predicción clásica, agregando un paso previo a la predicción, con el objetivo de mejorar la calidad de los valores de dichos parámetros, mejorando luego, la calidad de las predicciones.

Predicción en Dos Etapas

La predicción de la evolución del fuego en dos etapas tiene la ventaja de realizar un preprocesamiento de los valores de los parámetros de entrada, buscando los mejores valores, los que mejor describan las condiciones del entorno del fuego.

En esta predicción en dos etapas se agrega la etapa de *ajuste* o *calibrado*, durante la cual se realiza una búsqueda de los mejores parámetros de entrada para el avance del fuego que se desea predecir. Al agregar una etapa previa a la predicción, esta etapa requiere un paso de simulación, por lo que la predicción podrá realizarse para un instante de tiempo posterior.

Durante esta etapa se utiliza el mapa de avance del fuego para comparar las salidas del simulador (utilizando distintos escenarios, esto es, distintos valores para los parámetros de entrada) y cada mapa simulado se compara con el mapa del avance real del fuego. Esta información es utilizada para sintonizar y mejorar los valores para los parámetros de entrada.

La cantidad de parámetros y sus rangos de variación válidos resultan en un gran espacio de búsqueda. Esta búsqueda puede generar un cuello de botella para la predicción, por lo que se pone especial cuidado en realizarla de forma eficiente.

En [1] se muestra que los algoritmos genéticos son convenientes para esta búsqueda específica. En dicho trabajo, se comparan los resultados de utilizar tres tipos de búsquedas: algoritmos genéticos, *taboo search* y *simulated annealing*. Estas tres búsquedas fueron adaptadas para ser algoritmos evolutivos y de esta forma fueron utilizadas en el marco propuesto en dicho trabajo (*BBOF: Black Box Optimization Framework*).

Durante los primeros meses de esta investigación, se ha analizado dicho *framework*, y luego de dicho análisis y “puesta a punto”, se lo utilizó con mapas reales, obteniendo resultados que sirvieron para la comparativa presentada en los trabajos:

Germán Bianchini, Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque, “Improving ForestFire Prediction by applying a Statistical Approach”. Proceedings of V International Conference on Forest Fire Research. D.X. Viegas (Ed.), 2006.

Germán Bianchini, Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque, “Improving Forest Fire Prediction by applying a Statistical Approach”. Forest Ecology and Management, 234, Suplemento 1, pp. S210, ISSN: 0378-1127.

Una vez comparados los distintos métodos, se enfoca toda la atención al algoritmo genético ya que es el que mejores resultados logró (en los trabajos enumerados anteriormente, se utilizó esta búsqueda en dicho *framework*).

La predicción en dos etapas logra mejorar la predicción clásica, y el algoritmo genético utilizado durante la etapa de ajuste permite realizar la búsqueda de forma eficiente y correcta, y además, muestra buena adaptación tanto para mapas reales como para sintéticos.

En consecuencia, fue posible observar que el algoritmo genético realiza la búsqueda de forma correcta. Sin embargo, hasta entonces, la búsqueda se realizaba en todo el espacio de búsqueda sin utilizar ningún criterio o conocimiento sobre este espacio. Sin embargo, es posible utilizar conocimiento aportado por el avance real del fuego, inyectarlo en el algoritmo genético y de esta forma lograr guiar dicha búsqueda.

Se han propuesto distintos métodos para guiar la búsqueda, los cuales se describen a continuación. Estos métodos son la principal contribución de esta Tesis.

Predicción en Dos Etapas Guiada por los Datos

Utilizando la predicción en dos etapas, se proponen mejoras en la etapa de ajuste, principalmente con el objetivo de:

1. Mejorar la calidad de las predicciones: esto es, mejorando la calidad de las simulaciones en la etapa de ajuste, se mejoran las entradas para el paso de predicción.
2. Reducir el tiempo de respuesta de la aplicación: guiando la búsqueda, se logra converger hacia buenos resultados en menor cantidad de iteraciones, por lo que se logra disminuir el tiempo total de ejecución de la aplicación.

3. Guiando la búsqueda mediante el conocimiento aportado por el análisis del mapa real, se está inyectando datos reales, siendo este un primer paso en la obtención de una aplicación que se adapte de forma correcta a cambios en las condiciones medioambientales.

Para esto, se han desarrollado y analizado distintos métodos para guiar el algoritmo genético.

Para realizar la búsqueda de una buena combinación de valores de entrada se utiliza un algoritmo genético paralelo, el cual explota la capacidad de cómputo de sistemas paralelos/distribuidos. Teniendo en cuenta las principales características de estas arquitecturas y las características del dominio de la aplicación, se ha propuesto un algoritmo genético paralelo utilizando el paradigma *master/worker*. Esta aplicación paralela es escalable y además evita posibles penalizaciones propias del cómputo paralelo: desbalanceo de carga, demoras por comunicaciones, etc.

Para evitar el desbalanceo de carga se estudió la aplicación para detectar la distribución de cómputo de cada tarea realizada en toda la aplicación entera.

Estos estudios permitieron determinar que el mayor cómputo de la aplicación se realiza en el momento de las múltiples simulaciones (una por cada individuo de la población para cada una de las generaciones), lo cual derivó en repartir la carga de trabajo dividiendo en grupos los individuos de cada población y distribuyéndolos entre los procesos *workers*.

Por otro lado, se incrementa la granularidad de la aplicación y se evitan comunicaciones enviando grupos de individuos (*chunks*) en cada mensaje. Por el momento se propuso una sólo técnica de distribución de *chunks*, una solución simple y directa, quedando como línea abierta la definición y utilización de más criterios de distribución.

Métodos de Análisis de Mapas

Los métodos para guiar la búsqueda propuestos en este trabajo se basan en información que se obtiene a partir del análisis del mapa de la evolución real del fuego en el instante t_{i+1} . El análisis del mapa derivó en realizar diversas tareas:

1. Descubrimiento de la línea inicial del fuego, evaluando cada celda quemada, su tiempo de ignición y el estado de sus vecinos.
2. Descubrimiento de la línea final del fuego que, de forma similar a la línea inicial, evaluando para cada celda quemada, su tiempo de ignición y estado de los vecinos.

Una vez que se tienen estas líneas de fuego definidas, se procesan mediante una adaptación del algoritmo de Hausdorff realizándose las dos tareas que siguen:

1. Descubrimiento del formato del mapa. Esto es, discriminar entre mapas de avance elíptico o lineal. Una vez conocido el formato,
2. cálculo de los valores de velocidad y dirección de propagación más descriptivos para dicho fuego (los cálculos son distintos si se trata de un mapa lineal o un mapa elíptico).

Una vez que se conoce el formato del mapa se obtienen las principales características de la propagación: velocidad de máxima propagación, dirección en la que se produce y la distancia recorrida.

El método para analizar el mapa derivó en trabajos extras que no se planearon desde un principio. Estos análisis de los mapas tratan de responder de forma correcta a cualquier tipo de mapa, independientemente de su formato. Los resultados obtenidos son satisfactorios, consiguiendo describir los mapas probados acorde con la realidad.

Métodos para Guiar la Búsqueda

Se han propuesto dos métodos para guiar la búsqueda: el Método Computacional y el Método Analítico. El segundo surge para verificar la correctitud del funcionamiento del primero.

Ambos métodos se han evaluado con mapas sintéticos y también con mapas reales. Las pruebas se han realizado con diversos mapas, ya que los resultados cambian considerablemente dependiendo del mapa. Esto es atribuible a la capacidad o incapacidad del simulador utilizado de lograr un mapa con un formato similar a la propagación real del fuego.

Los primeros resultados fueron obtenidos utilizando el Método Analítico. Este método, fue primeramente evaluado con mapas sintéticos y reales. Resultados representativos se pueden encontrar en los trabajos:

Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque.
“Algoritmos Genéticos Guiados para Predicción de Incendios Forestales.” CEDI 2007, II Congreso Español de Informática, XVIII Jornadas de Paralelismo. Actas de las XVIII Jornadas de Paralelismo, Volumen 1. ISBN 978-84-9732-593-6. Páginas 733-741, Zaragoza, España, 2007.

Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque.
“Applying a Dynamic Data Driven Genetic Algorithm to Improve Forest Fire Spread Prediction.” International Conference

on Computational Science (ICCS) 2008. Computational Science - ICCS 2008. Lecture Notes in Computer Science, Volumen 5102, ISBN 978-3-540-69388-8. Páginas 36-45. Cracovia, Polonia (2008).

En dichos trabajos se muestra que utilizar el Método Analítico para guiar la búsqueda logra reducir el error cuando se compara con el método sin guiar la búsqueda. Este método, además de reducir el error mejora la convergencia a zonas del espacio de búsqueda con individuos de buena calidad.

Posteriormente, se desarrolla el Método Computacional, que como principal ventaja tiene la independencia con el marco de trabajo así también como con el simulador utilizado. Esto permite que tanto el marco de trabajo como así también el propio Método Computacional puedan modificarse, ampliarse, mejorarse, sin interferirse en dichas evoluciones.

Al desarrollarse el Método Computacional es posible la comparación entre los dos métodos. La primer versión del algoritmo fue un algoritmo secuencial, y las primeras pruebas que se realizaron con ambos métodos fueron con mapas sintéticos. Los resultados mostraron principalmente una buena reducción de error en casi todos los casos evaluados. Estos resultados fueron publicados en:

Mónica Denham, Ana Cortés, Tomás Margalef, Emilio Luque. "Algoritmos Genéticos Guiados para Predicción de Incendios Forestales." XIII Congreso Argentino de Ciencias de la Computación 2007. Proceedings - CD Rom. ISBN 978-950-656-109-3. Páginas 1620-1631. Corrientes y Resistencia, Argentina (2007).

Como punto de partida los métodos propuestos para guiar la búsqueda han mostrado lograr el objetivo específico de reducir el error. Haciendo un análisis más exhaustivo de sus comportamientos, se logró observar que cuando se guía la búsqueda, no se depende tanto de la suerte de los valores aleatorios del algoritmo genético como así también de la población inicial, sino que la guía logra dirigir la evolución de la búsqueda hacia zonas con individuos con buenos valores de forma rápida. Un pequeño análisis de esto está publicado en el artículo recientemente expuesto.

Posteriormente, se obtiene la versión paralela de dicho algoritmo, lo que permite más evaluaciones. En el trabajo:

Mónica Denham, Ana Cortés, Tomás Margalef. "Computational Steering Strategy to Calibrate Input Variables in a Dynamic Data Driven Genetic Algorithm for Forest Fire Spread Prediction." International Conference on Computational Science (ICCS) 2009. Computational Science - ICCS 2009. Lecture

Notes in Computer Science, Volumen 5545, ISBN 978-3-642-01972-2. Páginas 479-488. Baton Rouge, Lousiana, USA (2009)

se evalúan más mapas comparando ambos métodos de sintonización de parámetros, utilizando mapas sintéticos y reales. Además, en este trabajo se agrega un estudio de la escalabilidad de la aplicación paralela y los tiempos obtenidos. El análisis de estos tiempos permitió valorar que la escalabilidad de la aplicación es correcta, lográndose una reducción de los tiempos al incrementar la cantidad de nodos utilizados.

Además de la comparación entre los métodos de guiar la búsqueda frente a no guiarlos, el estudio de los errores en las ejecuciones utilizando el algoritmo paralelo (que permite aumentar la cantidad de individuos utilizados en cada población) permitió observar una importante reducción en los errores. Esto permite concluir que utilizar más individuos durante las evoluciones logra mejorar los valores de los mismos.

Lo anteriormente expuesto, se debe a una característica común de los algoritmos genéticos ([36]): en los algoritmos genéticos es usual que la búsqueda converja rápidamente a algún mínimo o máximo local, y que este mínimo (en nuestro caso) tenga mucha descendencia, produciendo individuos parecidos, generando poblaciones con poca diversidad de características, condicionando así el resultado de la búsqueda. Entonces, algunas formas de evitar esta característica son: utilización de poblaciones más grandes, con mayor cantidad de individuos, introducir algún tipo de ranking o peso en la operación de selección, etc.

Por otro lado, el desarrollo de esta aplicación generó distintas evaluaciones de aspectos exclusivos de su desarrollo paralelo. Un aspecto importante a evaluar fue el balance de carga. Uno de los análisis realizados tuvo como objetivo evaluar el tiempo de cómputo de las distintas tareas que componen esta aplicación paralela.

Utilizando la herramienta *MPE_Open_Graphics*, se evaluó el tiempo de procesamiento de cada tarea para poder determinar de forma correcta cómo se distribuiría la carga de trabajo en la aplicación. Dichos resultados fueron expuestos en el trabajo:

Mónica Denham, Ana Cortés, Tomás Margalef. "Parallel Dynamic Data Driven Genetic Algorithm for Forest Fire Prediction." EuroPVM/MPI 2009. Aceptado, aún no publicado. Espoo, Finlandia, 2009.

Dichos resultados, permitieron entender el comportamiento de esta aplicación y, a partir de dicho estudio, definir una forma simple de distribución del trabajo. Esta es una primera aproximación, quedando como línea abierta la continuación en este punto.

6.2. Líneas Abiertas

La realización de este trabajo permitió proponer y desarrollar métodos para mejorar las predicciones y para disminuir el tiempo de respuesta de las mismas. Pero durante el desarrollo de este trabajo, muchos problemas se intentaron solucionar y muchos otros no pudieron ser incluidos dentro del marco de dicha Tesis. Los próximos párrafos describirán algunos de los mismos los cuales definen líneas de trabajo futuro.

Una tarea a realizar es cambiar el motor de simulación, cambiando el simulador *fireLib* por algún otro que, posiblemente, no sea un autómata celular sino que simule el crecimiento elíptico del fuego. Los cambios que esto implica en la aplicación desarrollada, están relacionados con los formatos de las entradas y salidas. Una opción en estudio es el simulador FARSITE, el cual trabaja con ficheros vectoriales y *raster*. Al realizar la aplicación en módulos, los cuales resuelven distintas partes de la aplicación como “cajas negras” este cambio no afectaría al módulo que implementa el algoritmo genético, ni al Método Computacional, etc.

Otro cambio en el cual se está trabajando es mejorar la forma de implementar la base de conocimiento. Para esto, se utilizará un motor de base de datos y técnicas de *Data Mining*, las cuales implican búsquedas más inteligentes y almacenamiento de los datos más eficiente [50].

En esta Tesis, la forma de distribuir los *chunks* entre los procesos workers es estática, teniendo por el momento un sólo criterio para elegir el tamaño del *chunk*. En un futuro cercano se estudiarán distintos tamaños de *chunks* para determinar y desarrollar más criterios para la elección del tamaño de cada *chunk* y la forma de distribuirlos.

Otro aspecto que se tiene en cuenta para futuros trabajos es evaluar cómo afectaría la utilización de mapas de magnitudes más grandes, lo cual puede llegar a afectar el balance de carga debido al trabajo que realiza cada uno de los *workers*.

La realización de este trabajo permitió valorar el trabajo existente en este tema, pero también se pudo valorar todo lo que queda por hacer, lo cual tiene una magnitud aún mayor. La predicción del comportamiento del fuego fue el problema para el cual nuestra aplicación fue desarrollada, siendo esta una propuesta que se espera, contribuya de alguna forma a seguir mejorando las herramientas y métodos actuales.

Apéndice

Apéndice A

Fórmulas Trigonométricas del Modelo del Fuego

A.1. Obtención de la Dirección Donde Ocurre la Máxima Velocidad de Propagación e Intensidad del Fuego

Durante la explicación del modelo, su implementación y los métodos para guiar la búsqueda, se han utilizado vectores para calcular la dirección de máxima propagación y la intensidad en esta dirección. En este apartado se completan los pasos seguidos para llegar a las ecuaciones utilizadas en el Método Analítico explicado en el capítulo 4.

Las características del viento y de la pendiente se representan a través de vectores, donde la dirección del vector del viento corresponde a la dirección hacia donde corre el viento y su módulo o longitud están en función de su velocidad e intensidad. A su vez, la dirección del vector de la pendiente indica la orientación de la misma, esto quiere decir, hacia dónde sube la pendiente. A su vez, el módulo del vector, está en función de los grados de inclinación del terreno y de las características del terreno (pueden actuar como aceleradores o atenuantes).

Entonces, dados estos dos vectores, en nuestro caso el vector Φ_s y Φ_w cuyos ángulos son β y α respectivamente (figura A.1 (a)).

Dadas las propiedades trigonométricas de los triángulos rectángulos, se pueden obtener las dimensiones de sus lados a partir de la medida de la hipotenusa y de uno de sus ángulos.

Entonces, por cada vector, formamos dos triángulos rectángulos para

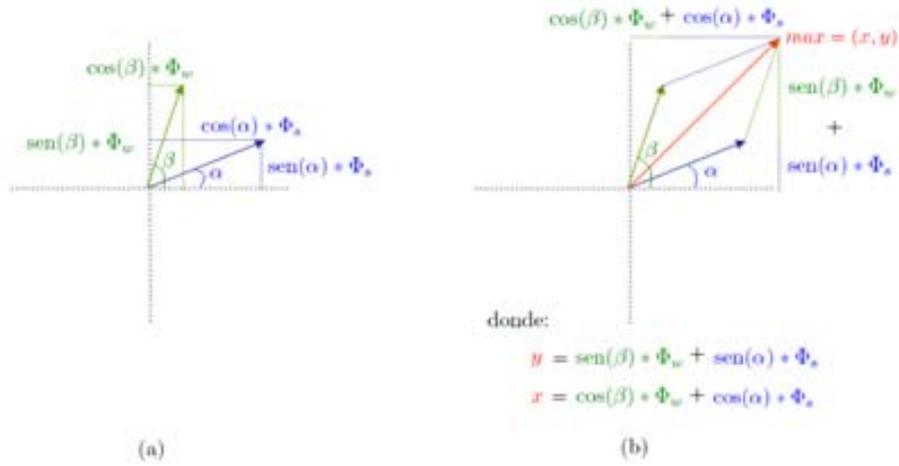


Figura A.1: Obtención de las características de la propagación del fuego.

hallar las dimensiones necesarias tal como muestra la figura A.1 (a). En verde, figuran los lados del triángulo formados por el vector del viento y en azul el formado por el vector de la pendiente.

Llamemos $x_pendiente$ e $y_pendiente$ a la base y a la altura respectivamente que tiene el vector de la pendiente. En este caso se conocen los valores de α y de Φ_s , entonces, partiendo de que:

$$\text{cos}(\alpha) = \frac{x_pendiente}{\Phi_s} \quad (\text{A.1})$$

entonces,

$$x_pendiente = \text{cos}(\alpha) * \Phi_s \quad (\text{A.2})$$

de la misma forma, como se conocen los valores de β y de Φ_w :

$$\text{sen}(\alpha) = \frac{y_pendiente}{\Phi_s} \quad (\text{A.3})$$

entonces,

$$y_pendiente = \text{sen}(\alpha) * \Phi_s \quad (\text{A.4})$$

De la misma forma, llamemos x_viento e y_viento a la base y a la altura respectivamente que tiene el vector del viento. En este caso es:

$$\text{cos}(\beta) = \frac{x_viento}{\Phi_w} \quad (\text{A.5})$$

entonces,

$$x_{viento} = \cos(\beta) * \Phi_w \quad (\text{A.6})$$

de la misma forma,

$$\text{sen}(\beta) = \frac{y_{viento}}{\Phi_w} \quad (\text{A.7})$$

luego,

$$y_{viento} = \text{sen}(\beta) * \Phi_w \quad (\text{A.8})$$

De esta forma, si el vector de propagación del viento se forma a partir de la suma de estas dimensiones, entonces:

$$x = x_{pendiente} + x_{viento} \quad (\text{A.9})$$

y

$$y = y_{pendiente} + y_{viento} \quad (\text{A.10})$$

Reemplazando en estas ecuaciones (ecuaciones A.9 y A.10) por los valores hallados en las ecuaciones anteriores (A.2, A.4, A.6 y A.8), quedan formadas las ecuaciones que se ilustran en la figura A.1 (b);

$$x = \cos(\alpha) * \Phi_s + \cos(\beta) * \Phi_w \quad (\text{A.11})$$

y

$$y = \text{sen}(\alpha) * \Phi_s + \text{sen}(\beta) * \Phi_w \quad (\text{A.12})$$

Apéndice B

Obtención del Ángulo y Velocidad del Viento

A partir de las fórmulas explicadas en el Apéndice A, se hallarán los valores para la velocidad y para la dirección del viento. Para una mayor claridad en el desarrollo, las ecuaciones del apéndice anterior se reescriben a continuación:

$$x = \cos(\alpha) * \Phi_s + \cos(\beta) * \Phi_w \quad (\text{B.1})$$

y

$$y = \text{sen}(\alpha) * \Phi_s + \text{sen}(\beta) * \Phi_w \quad (\text{B.2})$$

Los datos del viento que queremos hallar son la dirección (β) y su intensidad (Φ_w).

Mediante el estudio del mapa se analiza la propagación del fuego, por lo que se calculan y se obtienen los valores de x y de y en estas últimas ecuaciones. Además, se conocen los datos de la pendiente (α , su orientación y Φ_s , su intensidad). Conociendo todos estos datos, y teniendo como incógnitas los datos del viento, estos mismos se despejan partiendo de la ecuación B.1:

$$\Phi_w = \frac{x - \cos(\alpha) * \Phi_s}{\cos(\beta)} \quad (\text{B.3})$$

luego, a partir de la ecuación B.2 se despeja $\text{sen}(\beta)$:

$$\text{sen}(\beta) = \frac{y - \text{sen}(\alpha) * \Phi_s}{\Phi_w} \quad (\text{B.4})$$

ahora, utilizando Φ_w en la ecuación anterior:

$$\sin(\beta) = \frac{(y - \sin(\alpha) * \Phi_s) * \cos(\beta)}{x - \cos(\alpha) * \Phi_s} \quad (\text{B.5})$$

entonces,

$$\frac{\sin(\beta)}{\cos(\beta)} = \frac{y - \sin(\alpha) * \Phi_s}{x - \cos(\alpha) * \Phi_s} \quad (\text{B.6})$$

$$\tan(\beta) = \frac{y - \sin(\alpha) * \Phi_s}{x - \cos(\alpha) * \Phi_s} \quad (\text{B.7})$$

$$\beta = \arctan\left(\frac{y - \sin(\alpha) * \Phi_s}{x - \cos(\alpha) * \Phi_s}\right) \quad (\text{B.8})$$

Bibliografía

- [1] Abdalhaq B., “*A Methodology to Enhance the Prediction of Forest Fire Propagation.*” Tesis Doctoral. Universitat Autònoma de Barcelona (España). Junio 2004.
- [2] Albini F., “*Estimating wildfire behavior and effects.*” USDA Forest Service. General Technical Report INT-30. 1976.
- [3] Albright D., Meisner B., “*Classification of fire simulation systems.*” Fire Management Notes, 59 (2), p 5-12. Disponible en http://www.fs.fed.us/fire/fmt/volumes_authors.html Accedido en noviembre de 2008.
- [4] Allen, G., “*Building a Dynamic Data Driven Applications System for Hurricane Forecasting.*” International Conference of Computational Science 2007, Part I, LNCS 4487, pp. 1034-1041.
- [5] Anderson, H. “*Aids to Determining Fuel Models For Estimating Fire Behaviour.*” Intermountain Forest and Range Experiment Station Ogden, UT 84401. General Technical Report INT.122, 1982.
- [6] Andrews P., “*BEHAVE: Fire Behavior prediction and modeling systems - Burn subsystem.*” part 1. General Technical Report INT-194. Ogden, UT, US Department of Agriculture, Forest Service, Intermountain Research Station. pp. 130. 1986.
- [7] Andrews, P. “*BehavePlus fire modeling system: past, present, and future.*”. Proceedings of 7th Symposium on Fire and Forest Meteorological Society, 23-25 Octubre de 2007, Bar Harbor, Maine, Boston. <http://ams.confex.com/ams/pdfpapers/126669.pdf>. Accedido en octubre del 2008.
- [8] Andrews, P., Bevins C. “*Update and Expansion of the BEHAVE Fire Behavior Prediction System.*” Julio 1998. Disponible en <http://www->

- laep.ced.berkeley.edu/ itr/literature/behave/behave_single.html. Accedido en noviembre de 2008.
- [9] Bevins C., “*FireLib User Manual & Technical Reference*”. Disponible en: <http://www.fire.org/downloads/fireLib/1.0.4/firelib.pdf>. Accedido en enero 2006.
- [10] Bianchini G., “*Wildland Fire Prediction based on Statistical Analysis of Multiple Solutions*”. Tesis Doctoral. Universidad Autónoma de Barcelona. España. Julio de 2006.
- [11] Bonsor K., “*How Wildfires Work*”. How Stuff Works. Disponible en: <http://www.howstuffworks.com/wildfire.htm>. Accedido en abril 2007.
- [12] Caballero D. “*Taxicab Geometry: some problems and solutions for square grid-based fire spread simulation*”. V International Conference on Forest Fire Research. D.X. Viegas (Ed.) 2006
- [13] Cheney N., Gould J., Catchpole W., “*The Influence of Fuel, Weather and Fire Shape Variables on Fire-Spread in Grasslands*”. International Journal of Wildland Fire 3, pp. 31-44, 1993.
- [14] Cheney N. and Gould J. “*Fire growth and acceleration (letter to the editor.)*”. International Journal of Wildland Fire 7, pp. 1-5, 1997.
- [15] Coen J., Beezley J., Bennethum L., Douglas C., Kim M., Kremens R., Mandel J., Qin G. and Vodacek A. “*Wildland Fire Dynamic Data-Driven Application System.*” Center for Computational Mathematics Reports, University of Colorado at Denver and Health Sciences Center. November 2006. UCDHSC/CCM Report number 238.
- [16] Darema, F. “*Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements.*” ICCS 2004, LNCS 3038, Springer Berlin / Heidelberg, pp. 662-669.
- [17] Douglas C., “*Dynamic Data Driven Application Systems homepage.*” <http://www.dddas.org>. Accedido en noviembre de 2007, octubre 2008.
- [18] Denham M., “*Predicción de Incendios Forestales Basada en Algoritmos Evolutivos Guiados por los Datos*”. Master Tesis. Universitat Autònoma de Barcelona, España, julio 2007.
- [19] Denham M., Cortés A., Margalef T., Luque E., “*Algoritmos Genéticos Guiados Para Predicción de Incendios Forestales*”. Actas del XVIII Jornadas de Paralelismo JP'07. Zaragoza, 2007. Vol II, pp. 733-741 .

- [20] http://ies.jrc.ec.europa.eu/fileadmin/Documentation/Press%20Releases/2007-08-02_Forest_fires_2007.-ES.pdf
- [21] Finney M. “*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.
- [22] FIRE.ORG - Public Domain Software for the Wildland Fire Community. <http://www.fire.org>. Accedido en mayo de 2007.
- [23] Foster I., “*Designing and Building Parallel Programs.*” (Online) Addison-Wesley ISBN 0201575949, chapter 8 Message Passing Interface. Disponible en <http://www-unix.mcs.anl.gov/dbpp/text/node94.html> (1995)
- [24] Galeano E. “*Las venas abiertas de América Latina*” 1971.
- [25] Hu X., Ntaimo L. “*Dynamic Multi-resolution Cellular Space Modeling for Forest Fire Simulation*” DEVS Integrative M&S Symposium (DEVS’06) (DEVS 2006) Huntsville, Alabama, USA, April 2 - 6, 2006
- [26] Johnston P., Milne G., Kelso J., “*A heat transfer simulation model for wildfire spread*” V International Conference on Forest Fire Research, D. X. Viegas (Ed.), 2006.
- [27] Johnston P., Kelso J. and Milne G. “*Efficient simulation of wildfire spread on an irregular grid*” International Journal of Wildland Fire 2008. 17, pp. 614-627,
- [28] Koza J. “*Genetic Programming. On the programming of computers by means of natural selection.*” Massachusetts Institute of Technology. Cambridge, Massachusetts 02142. The MIT Press, 1992.
- [29] Lopes A., Cruz M. y Viegas D. “*FireStation - An integrated software system for the numerical simulation of fire spread on complex topography.*” Environmental Modelling and Software 17(3): 269-285 (2002)
- [30] Mandel J., Beezley J., Bennethm L., Chakraborty S., Coen J., Douglas C., Vodacek A. and Wang Z. “*Towards a Real-Time Data Driven Wildland Fire Model*”. <http://arxiv.org/abs/0801.3875v>. Accedido en junio de 2008.
- [31] Mandel J., Beezley J., Bennethm L., Chakraborty S., Coen J., Douglas C., Hatcher J., Kim M. and Vodacek A. “*A Dynamic Data Driven Wildland Fire Model*”. International Conference of Computational Science 2007, LNCS 4487, Springer Berlin / Heidelberg, pp. 1042-1049.

- [32] <http://rapidfire.sci.gsfc.nasa.gov/firemaps/> Accedido en octubre de 2008.
- [33] MPE_Open_Graphics
http://www-unix.mcs.anl.gov/mpi/www/www4/MPE_Open_graphics.html.
 Accedido en Septiembre de 2008.
- [34] “MPI: The Message Passing Interface Standard.” <http://www-unix.mcs.anl.gov/mpi/>
- [35] Matthews Stuart. “Moisture content models - a review”. Notes for 5th Course in Fire Behaviour Modelling, Figueria da Foz, 25/26 Noviembre 2006.
- [36] Michalewicz, Z. “Genetic algorithms + data structures = Evolution Programs.” ISBN 9783540606765, Springer 1996]
- [37] Reinhardt E., Keane R., Brown J., “First Order Fire Effects Model: FOFEM 4.0, User’s Guide.” General Technical Report INT-GTR-344. 1997.
- [38] Rothermel R., “A mathematical model for predicting fire spread in wildland fuels”. USDA FS, Ogden TU, Res. Pap. INT-115, 1972.
- [39] Rothermel R. C. “How to Predict the Spread and Intensity of Forest and Range Fires”. Gen. Tech. Rep. INT-143. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, 1983. Pág. 1..5.
- [40] SPREAD. Disponible en <http://www.algosystems.gr/spread/>.
- [41] Trunfio G.A., “Predicting Wildfire Spreading Through a Hexagonal Cellular Automata Model”. Cellular Automata for Research and Industry, University of Amsterdam, The Netherlands, pp. 385-394. LNCS 3305, Springer-Verlag, Berlin, 2004.
- [42] Viegas D., “On the existence of a steady state regime for slope and wind driven fires”. International Journal of Wildland Fire, 2004, 13, pp. 101-117.
- [43] Viegas D., “Parametric study of an eruptive fire behaviour model”. International Journal of Wildland Fire, 2006, 15, pp. 169-177.
- [44] http://es.wikipedia.org/wiki/Incendios_forestales_en_California_de_octubre_de.2007. Accedido en julio de 2008.

- [45] Top 500 Supercomputer Sites.
<http://www.top500.org>. Accedido en octubre de 2008.
- [46] Wikipedia, The Free Encyclopedia. Wildfire.
<http://en.wikipedia.org/wiki/Wildfire>. Accedido en julio de 2008.
- [47] Wikipedia, la enciclopedia libre. Algoritmo Genético.
http://es.wikipedia.org/wiki/Algoritmo_genetico. Accedido en diciembre de 2008.
- [48] Wikipedia, The Free Encyclopedia. Distancia de Hausdorff.
http://es.wikipedia.org/wiki/Distancia_de_Hausdorff. Accedido en febrero de 2009.
- [49] <http://www.monografias.com/trabajos15/transf-calor/transf-calor.shtml> Accedido en octubre de 2008.
- [50] Wendt K., “*Efficient knowledge retrieval to calibrate input variables in forest fire prediction*”. MsC Thesis. Universitat Autònoma de Barcelona (España). Julio 2008.
- [51] Yang J., Chen H., Kim B., Hariri S., Parashar M. “*Autonomic Runtime System for Large Scale Parallel and Distributed Applications*”. Lecture Notes in Computer Science, Springer Berlin/Heidelberg, ISSN 0302-9743 (Print) 1611-3349 (Online), pp 297-311, 2005.
- [52] Zárata López L. “*Estudio de las Características Físicas y Geométricas de la llama en los Incendios Forestales*” Tesis Doctoral. Universidad Técnica de Catalunya. España. Febrero 2004. Disponible en <http://www.tesisenred.net/TDX-0503104-095443>. Accedido en Octubre de 2008.

Índice alfabético

- Algoritmos Genéticos, 62, 72, 80
 - Cruzamiento, 67
 - Elitismo, 65
 - Función de Error, 70
 - Mutación, 68
 - Selección, 65
- Análisis de la Línea de fuego, 81
- Clusters de Ordenadores, 108
- DDDAS, 12, 14, 80
 - Método Analítico, 97, 103, 131
 - Método Computacional, 92, 103, 131
- Función de *fitness*, 62
- Hausdorff, 83
- Incendios Forestales, 1–3, 5
 - Cuadrado del fuego, 8
 - Predicción de Incendios Forestales, 51
 - Triángulo del fuego, 7
- Métodos de predicción, 14
 - Predicción Clásica, 14
 - Predicción en dos etapas, 14
 - Predicción en dos etapas guiada por los datos, 14, 61
- Modelos de Combustible, 20
- Modelos de Comportamiento del Fuego, 17, 21, 22, 50
 - Modelos Estadísticos, 23, 50
 - Modelos Físico-Estadísticos, 23, 50
 - Modelos Físicos, 22, 50
 - Modelos Probabilísticos, 23, 50
- MPI_Open_Graphics, 73
- ROS: Rate of Spread, 7, 22, 23, 26
- Rothermel, 23, 87
- Simulador de Incendios Forestales, 36, 47, 50
 - BehavePlus, 36
 - Clarke Cellular Automaton System, 38, 50
 - DYNAFIRE, 38, 50
 - EMBYR, 39, 50
 - FARSITE, 41, 50
 - fireLib, 44, 88
 - FIREMAP, 39, 50
 - FireStation, 42, 50
 - FOFEM, 42
 - NEXUS, 40
 - Parámetros de Entrada, 62
 - WILDFIRE, 40, 50
- SPREAD, 109
- Técnicas de Simulación, 21, 26, 50
 - Autómatas Celulares, 29, 31, 33, 34, 50
 - Bond Percolation, 26, 50
 - Propagación Elíptica, 27, 50