

# ESS-IM applied to Forest Fire Spread Prediction: Parameters Tuning for a Heterogeneous Configuration

Miguel Méndez-Garabetti\*<sup>†</sup>, Germán Bianchini\*, Paola Caymes-Scutari\*<sup>†</sup>, María Laura Tardivo\*<sup>†‡</sup>

\*Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD),  
Departamento de Ingeniería en Sistemas de Información,  
Facultad Regional Mendoza - Universidad Tecnológica Nacional.  
(M5502AJE) Mendoza, Argentina.

<sup>†</sup>Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

<sup>‡</sup>Departamento de Computación, Universidad Nacional de Río Cuarto. (X5804BYA) Río Cuarto, Córdoba, Argentina.

**Resumen**—Forest fires are a critical natural hazard in many regions of the World. For this reason, the prediction of this kind of phenomenon is considered a very important task that involves a high degree of complexity and precision. The ability to predict the forest fire behaviour constitutes an important tool for managers, helping to improve the effectiveness of fire prevention, detection and firefighting resources allocation. For this reason, prediction methods should be configured to operate as efficiently as possible. In this paper, a calibration study of Evolutionary-Statistical System with Island Model's evolutionary parameters is presented (ESS-IM). ESS-IM is a general-parallel uncertainty reduction method applied to the forest fires spread prediction.

**Index Terms**—forest fire spread prediction, parallel evolutionary algorithms, parameters tuning, high performance computing.

## I. INTRODUCCIÓN

Los incendios forestales generan cuantiosas pérdidas alrededor del mundo, sólo en Chile la superficie afectada a causa de este fenómeno supera las 60 mil hectáreas anuales [1]. En la Fig. 1 se puede observar una fotografía de un incendio ocurrido en febrero de 2016 en la localidad de Chile Chico, Chile, donde más de 300 hectáreas fueron consumidas por las llamas [2]. Para luchar contra este tipo de riesgos, es importante utilizar de la mejor manera posible los escasos recursos disponibles. Para esto es necesario disponer de predicciones precisas de la evolución del fenómeno para tomar decisiones eficientes.

La propagación de un incendio forestal depende de diversas variables o factores: algunos con valores estáticos (i.e., la pendiente del terreno, el tipo de vegetación, la cantidad de combustible inicial), y otros en cambio, con valores dinámicos, como la velocidad y dirección del viento o la humedad en la vegetación y el ambiente. Para algunos de estos parámetros sus valores son conocidos, pero en otros, sus valores son estimados a partir de mediciones indirectas. Tales estimaciones implican cierto grado de incertidumbre en la entrada del modelo, lo que provoca predicciones distantes de la realidad. Dicha falta de precisión o existencia de incertidumbre afecta de forma directa la calidad de las predicciones generadas. Por esta razón, es

importante tratar dicha incertidumbre de modo que impacte de la menor manera posible sobre la salida del modelo.



Figura 1: Incendio forestal ocurrido en Chile Chico, Chile, febrero de 2016.

En este tipo de aplicaciones, además de priorizar la calidad de predicción, es muy importante tener en consideración el tiempo de respuesta del método, ya que es deseable obtener toda predicción antes que el incendio real haya finalizado. Para cumplir con este objetivo, tales métodos suelen ser implementados bajo arquitecturas paralelas de alto rendimiento [3]. Tal es el caso del *Sistema Estadístico Evolutivo con Modelo de Islas* (Evolutionary Statistical System with Island Model, ESS-IM) [4][5], el cual es un método de reducción de incertidumbre que ha sido aplicado satisfactoriamente en el pronóstico del comportamiento de incendios forestales. ESS-IM pertenece a la clasificación de Métodos Guiados por Datos con Múltiples Soluciones Solapadas (Data-Driven Methods with Multiple Overlapping Solutions, DDM-MOS). Estos métodos tienen la capacidad de efectuar las predicciones utilizando múltiples configuraciones de parámetros de entrada, los cuales previamente deben ser sometidos a un proceso de optimización (e.g., Algoritmos Genéticos, Taboo Search, Simulated Annealing).

Los DDM-MOS poseen grandes similitudes con los Métodos de Asimilación de Datos (Data Assimilation Methods, DAM), puntualmente, debido a que los últimos se caracterizan por incorporar datos a un modelo en funcionamiento [6]. Desde este punto de vista, podemos decir que ESS-IM es un método híbrido, ya que introduce conceptos de ambas clasificaciones. Por un lado, ESS-IM no realiza la búsqueda de un único conjunto de datos, ya que tiene como objetivo encontrar un grupo de valores que mejoren el resultado final del proceso, y por el otro, permite incorporar datos en tiempo de ejecución.

ESS-IM realiza la optimización de parámetros mediante un proceso basado en Algoritmos Evolutivos Paralelos (Parallel Evolutionary Algorithms, PEA), sobre un modelo multipoblacional basado en islas. En [4] y [5], ESS-IM incrementa la calidad de predicción respecto a metodologías previas (i.e., S<sup>2</sup>F<sup>2</sup>M [7][8] y ESS [9][10]), utilizando una configuración clásica de parámetros evolutivos. Una configuración clásica implica que todas las islas utilizan la misma configuración de parámetros (i.e., una configuración homogénea de parámetros evolutivos). En [11], mediante un análisis de calibración estática sobre un grupo de parámetros evolutivos, ESS-IM logró mejorar la calidad de predicción en ciertos instantes de tiempo, para un mismo incendio. Es decir, dicha calibración no ha permitido determinar una configuración de parámetros óptima y única, que posibilite obtener el mejor registro de calidad para la totalidad de los instantes de predicción de un mismo experimento.

Diferentes estudios han demostrado, que la combinación de varios hilos de búsqueda que implementan diferentes configuraciones de parámetros, incrementa la robustez del mecanismo de búsqueda global [12][13][14]. Por este motivo, se decide implementar el método utilizando una configuración heterogénea de parámetros evolutivos –a nivel de islas–, lo que podría permitir, que los resultados de calidad de predicción sean mejorados. Debido a esto, en este trabajo se presenta un estudio de calibración, complementario al realizado en [11], donde se considera el resto de los parámetros evolutivos de ESS-IM. De esta forma se intenta determinar aquellas configuraciones de parámetros que permiten obtener los mejores resultados de calidad de predicción, con el objetivo de que éstas sirvan como base para la implementación de ESS-IM, con una configuración heterogénea de parámetros evolutivos, i.e., donde cada isla opere con una configuración diferente de parámetros.

En la siguiente sección se describe el funcionamiento de los PEAs, junto con diferentes estrategias de paralelización. A continuación, en la Sección III se comentan los parámetros involucrados en el funcionamiento interno de los PEAs y la técnica de sintonización utilizada. Luego se proporciona una descripción detallada de la arquitectura y funcionamiento de ESS-IM. Seguidamente se brindan los detalles de la metodología utilizada para diseñar y evaluar los experimentos realizados, junto a los resultados obtenidos. Finalmente, se discuten las conclusiones y las líneas futuras de trabajo.

---

#### Algoritmo 1 Esquema general del algoritmo evolutivo paralelo

---

```

1: inicio
2:  $g=0$  /*número de generación*/
3: generar  $p$  poblaciones con  $n$  individuos
4: calcular la función de evaluación de cada individuo
5: while no se cumpla condición de fin do
6:   /* Producir nueva generación */
7:   seleccionar progenitores
8:   recombinar pares de progenitores
9:   mutar los descendientes con cierta probabilidad
10:  calcular la función de evaluación de cada individuo
11:  reemplazar los descendientes en la nueva generación
12:  if  $g \% \text{frecuencia} == 0$  then
13:    /* realizar migración de individuos */
14:    enviar  $m$  mejores individuos a población vecina
15:    recibir  $m$  individuos de población vecina
16:    reemplazar  $m$  individuos en población actual
17:  end if
18:  evaluar cada individuo con la función objetivo
19:   $g++$ 
20: end while
21: fin

```

---

## II. ALGORITMOS EVOLUTIVOS PARALELOS (PEAS)

La metaheurística conocida como Algoritmos Evolutivos (Evolutionary Algorithms, EAs) es un método de optimización, eficaz y eficiente, que puede ser aplicado en la resolución de problemas de alta complejidad. Los EAs obtienen su fuente de inspiración del proceso de evolución ocurrido en la naturaleza [15]. En términos generales, el mecanismo consiste en lograr la adaptación de una población de individuos en su ambiente circundante, comportándose como un sistema natural. Donde la aplicación del concepto de supervivencia, permita eliminar los rasgos perjudiciales y garantizar la permanencia de las mejores características.

De forma general, un EA produce un conjunto de posibles soluciones, que son sometidas a un proceso de iteraciones, llamadas generaciones evolutivas. En cada generación, se aplican ciertos operadores (i.e., *selección*, *cruciamiento*, *mutación* y *reemplazo*) que modelan el proceso de evolución natural. Cada individuo posee un valor que determina la calidad de la solución que dicho individuo representa, éste se conoce como *aptitud* o *fitness*, y se calcula utilizando la función matemática que describe el problema. El valor de aptitud en los EAs juega un rol fundamental, ya que permite caracterizar a los individuos según sus capacidades. De esta manera, aquellos individuos con mejor aptitud, deben tener mayor probabilidad de sobrevivir, de reproducirse, de no ser reemplazado y de esta forma, dejar sus rasgos hereditarios a las generaciones futuras [16].

Cuando los EAs son utilizados para resolver problemas complejos y/o de grandes dimensiones, usualmente éstos son paralelizados, dividiendo la carga de trabajo en  $N$  unidades de cómputo. Un esquema general de un Algoritmo Evolutivo

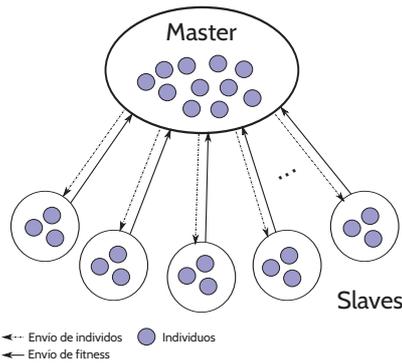


Figura 2: Esquema de paralelización master-slave o master-worker.

Paralelo se presenta en el Algoritmo 1. Tal como se puede observar, el proceso comienza con la generación de  $p$  poblaciones de  $n$  individuos, las cuales suelen ser creadas mediante un proceso estocástico. Una vez que las poblaciones han sido generadas, se evalúa la aptitud inicial de los individuos, aplicando la función de fitness a cada individuo. Luego, comienza el ciclo evolutivo, el cual itera mientras no se cumpla la condición de finalización, e.g., hasta que se alcance un cierto umbral de aptitud o una máxima cantidad de iteraciones.

Posteriormente se realiza el proceso de selección de progenitores, el cual debe brindar mayor probabilidad de ser elegidos, a aquellos individuos con mejores características. Los progenitores son sometidos a diferentes operadores evolutivos (i.e., cruzamiento y mutación), creando un nuevo conjunto de individuos, conocido como descendencia. Los individuos que conforman la descendencia deben ser incorporados en la población de la siguiente generación, mediante un procedimiento de reemplazo. Éste, al igual que el de selección, debe garantizar beneficiar a los mejores individuos. A continuación, se realiza el proceso de migración, el cual se efectúa cada cierta cantidad de generaciones o iteraciones (frecuencia de migración), donde  $m$  individuos son seleccionados y enviados por cada isla a sus vecinos. En la migración, la selección y el reemplazo, suelen ser elitistas, aunque también son válidas otras estrategias. A medida que la población evoluciona, los individuos deben aproximarse a una solución óptima para el problema tratado.

Según la cantidad de poblaciones involucradas, el tratamiento de los operadores, la relación cómputo/comunicación y demás factores; los EAs se pueden paralelizar siguiendo diferentes estrategias. Según [17] la clasificación de los PEAs, se puede resumir en cuatro grupos: 1) paralelismo automático a nivel del compilador, 2) master-worker con única población y evaluación en paralelo, 3) paralelismo de grano grueso, distribuido o basado en islas (múltiples poblaciones) y 4) de grano fino o masivamente paralelo.

En [18] se expone una taxonomía donde los modelos distribuidos de EAs se dividen en dos grandes grupos: *modelo de distribución por poblaciones* y *modelo de distribución por dimensiones*. La primera clasificación, consiste en distribuir

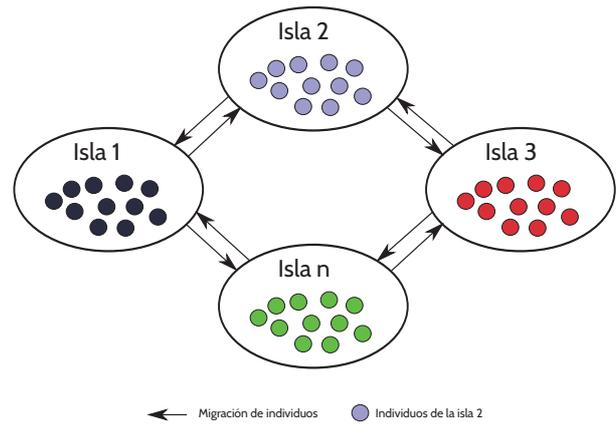


Figura 3: Esquema de paralelización basado en islas.

individuos de una población o subpoblación en diferentes unidades de cómputo, y la segunda, distribuye particiones de la dimensión del problema, en problemas más pequeños. A su vez, la clasificación de distribución basada en poblaciones, puede ser dividida en: modelo master-slave, modelo de islas, modelo celular o modelo jerárquico. Ambas clasificaciones, coinciden en gran parte y mantienen diferencias menores en ciertos aspectos. Las estrategias de paralelización que han sido utilizadas en este trabajo, corresponden con las tres clasificaciones siguientes: 1) modelo master-worker o master-slave, 2) modelo de islas y 3) modelo jerárquico.

El *modelo master-slave* (ver Fig. 2), se basa en el paradigma master-worker [16], donde una única población de individuos es administrada por un proceso central, usualmente denominado *master*. El master lleva a cabo gran parte de las funciones del PEA (i.e., selección, cruzamiento, mutación y reemplazo). En este esquema, sólo la evaluación de la aptitud de los individuos, se realiza de forma paralela por los procesos *slaves* o *workers*.

El *modelo de islas* (ver Fig. 3), utiliza múltiples poblaciones o islas, las cuales evolucionan de forma independiente. Cada población puede ser administrada por un worker o conjuntos de workers, dependiendo del nivel de paralelización con el cual el algoritmo sea implementado. Cada cierto número de generaciones, se produce un intercambio de individuos entre las diferentes islas o poblaciones, este proceso permite incrementar el grado de diversidad y minimizar la probabilidad de convergencia prematura o estancamiento en óptimos locales.

El *modelo jerárquico*, consiste en sumar las bondades de dos o más modelos diferentes en beneficio de la escalabilidad y/o la capacidad de búsqueda del método. Algunos de estos esquemas son: modelo de islas con master-slave, modelo de islas con modelo de islas y modelo de islas con doble jerarquía master-slave.

En la Fig. 4, se puede observar un esquema del modelo jerárquico, basado en islas con master-slave. Observe, que el primer nivel de paralelismo corresponde a un modelo de islas con migración, y en cada isla se aplica un modelo master-slave. En la Fig. 5 se observa un esquema de paralelismo basado en

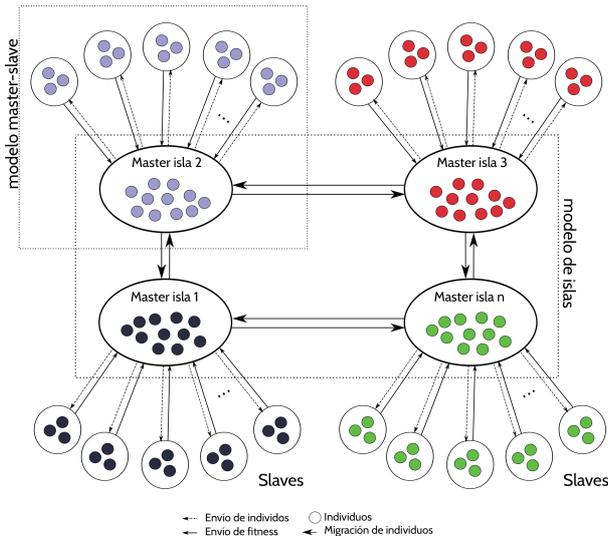


Figura 4: Esquema de paralelización jerárquico basado en islas con master-slave.

TABLA I: PARÁMETROS EVOLUTIVOS Y RANGOS UTILIZADOS.

Parámetro	Fijo / dinámico	Valor / rango
Umbral de fitness	dinámico	0,55 a 0,85
Cantidad máxima de iteraciones	dinámico	100 a 500
Cantidad de islas	fijo	4
Cantidad de individuos por isla	dinámico	10 a 200
Cantidad de individuos a migrar	dinámico	5 a 20
Criterio de migración	fijo	elitista
Frecuencia de migración	dinámico	1 a 10
Probabilidad de cruzamiento	dinámico	10 al 60 %
Probabilidad de mutación	dinámico	0,1 a 2,5 %
Topología de comunicación	fijo	anillo

islas, tanto a nivel de jerarquía superior como inferior. Dentro de cada isla la paralelización también utiliza un esquema basado en islas. Esta estrategia de paralelización incluye, un proceso de migración adicional dentro de cada isla de jerarquía mayor. Finalmente en la Fig. 6, se observa un esquema basado en islas con doble jerarquía master-slave. En el nivel superior, indicado como *modelo master-slave 1*, un proceso *master global* coordina el funcionamiento de la totalidad de las islas del PEA. El master de cada isla cumple la función de slave o worker respecto del master global. Posteriormente, en el segundo nivel de jerarquía master-slave (i.e., *modelo master-slave 2*), cada master de isla controla a un conjunto de workers dentro de cada isla, éstos son los encargados de evaluar la función de aptitud. Es importante remarcar que los operadores de cruzamiento, mutación y reemplazo, son aplicados entre individuos de la misma isla.

Evidentemente cada estrategia de paralelización ofrece diferentes ventajas en función de la implementación y el problema a abordar. En este caso, el método de reducción de incertidumbre propuesto, hace uso de un modelo de islas con doble jerarquía master-slave, el cual es desarrollado en la Sección IV.

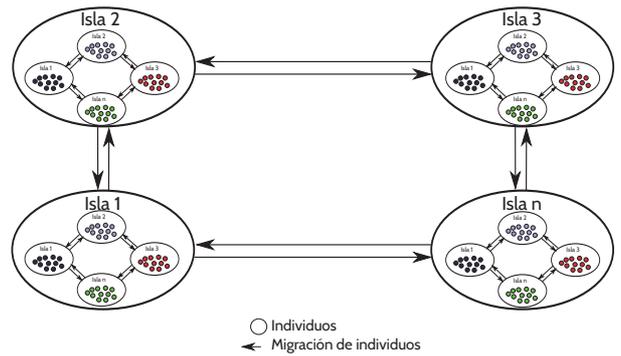


Figura 5: Esquema de paralelización jerárquico basado en islas con modelo de islas.

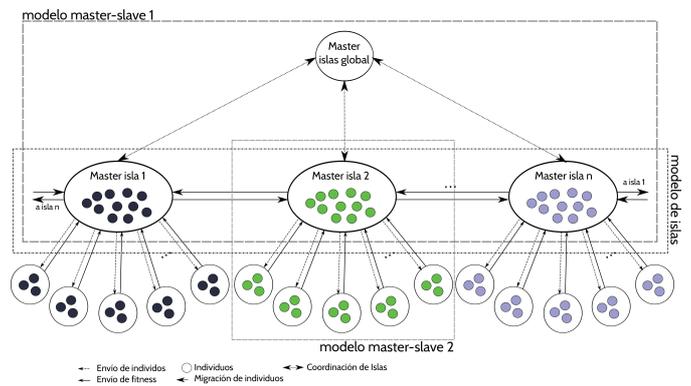


Figura 6: Esquema de paralelización jerárquico basado en islas con doble modelo master-slave.

### III. PARÁMETROS EVOLUTIVOS Y SINTONIZACIÓN

Los PEAs pueden obtener diferentes resultados, en función de la configuración de parámetros evolutivos utilizada [19]. Cada parámetro condiciona el comportamiento interno del PEA, haciendo que éste opere de mejor o peor manera. Esto ocurre tanto en términos de la calidad de las soluciones encontradas, como así también en relación a la cantidad de tiempo y recursos utilizados. Los parámetros evolutivos más significativos de todo PEA son los siguientes:

#### a) Umbral de fitness:

Determina el valor de fitness mínimo que debe ser obtenido por al menos un individuo. Un valor bajo implica menor tiempo para encontrar una la solución, pero con mayor probabilidad de ser un valor distante del óptimo. En cambio un valor elevado, dificultará la convergencia a una solución, incrementando el tiempo de procesamiento.

#### b) Cantidad máxima de iteraciones:

Determina la cantidad máxima de generaciones evolutivas que se llevarán a cabo, suele ser utilizada como condición de finalización junto con el umbral de fitness. Pocas iteraciones harán que el algoritmo finalice más rápido, con un pobre nivel de exploración del

espacio de búsqueda. Por el contrario, mayor cantidad de iteraciones puede generar pérdidas considerables de tiempo.

c) *Cantidad de islas:*

Establece la cantidad total de poblaciones o subpoblaciones que participarán del PEA. Mientras más poblaciones sean utilizadas, mejor se explorará el espacio de búsqueda incrementando la cantidad de recursos necesarios, como así también el tiempo de procesamiento.

d) *Cantidad de individuos por isla:* Determina la cantidad total de individuos que cada población mantendrá durante toda la ejecución del algoritmo (generalmente suele ser fija). Éste es un parámetro importante que afecta el rendimiento y la eficiencia del PEA, ya que si las poblaciones son muy pequeñas existirá poca diversidad, haciendo que no se puedan alcanzar soluciones de calidad. En cambio, a mayor tamaño aumentará la diversidad, pero será necesario realizar mayor cantidad de evaluaciones.

e) *Cantidad de individuos a migrar:* Especifica cuántos individuos deben ser copiados hacia las islas vecinas al producirse una migración.

f) *Criterio migración:* Este parámetro indica la forma en que serán seleccionados los individuos que migrarán (e.g., los mejores, aleatorios, mixto, etc.). También indica qué va a ocurrir con los nuevos individuos que se incorporen de otras poblaciones.

g) *Frecuencia de migración:* Este parámetro establece cuándo ocurre la migración, ésta puede realizarse de manera periódica, de manera determinista o a partir de un evento determinado. Una configuración habitual de este parámetro es en función del número de iteraciones. Es importante tener en cuenta que la migración introduce una sobrecarga de comunicación, por lo que éste parámetro se debe determinar cuidadosamente para que la comunicación no domine a la computación.

h) *Probabilidad de cruzamiento:* Este parámetro determina con que probabilidad se realizará el cruzamiento entre dos individuos.

i) *Probabilidad de mutación:* Este parámetro determina con que probabilidad se realiza la mutación sobre un individuo. Tiene el objetivo de realizar exploración aleatoria de nuevas porciones del espacio de búsqueda.

j) *Topología de comunicación:* Especifica la relación de comunicación entre las diferentes islas para el proceso de migración, una de las estrategias más comunes es la topología de anillo.

Para obtener una configuración óptima de éstos parámetros, es necesario someter el algoritmo a un proceso de optimización, análisis y ajuste [20]. Dicho proceso de optimización se conoce como *sintonización, calibración o análisis de parámetros*.

Debido a que las metaheurísticas son aplicadas a diferentes tipos de problemas, con características disímiles, es difícil definir una configuración genérica de parámetros para cada metaheurística. Según [20], dos son las técnicas más difundidas para realizar un análisis de calibración de parámetros:

a) *Sintonización Off-line:* Esta técnica consiste en definir diferentes configuraciones de parámetros para ser ejecutadas de forma independiente. Los valores de cada parámetro son establecidos antes de ejecutar el algoritmo y se mantienen estáticos durante toda la ejecución. Cuando todas las experimentaciones fueron llevadas a cabo, se analizan los resultados obtenidos y se identifica la configuración de parámetros que permite obtener mejores resultados. Cuando el estudio incluye una reducida cantidad de experimentos o cuando se trata de estudios de sondeo, los experimentos pueden ser realizados mediante un diseño por Hipercubo Latino (Latin Hypercube Design, LHD) [21][22], diseño fraccional [23], entre otros. Sin embargo, existen técnicas de sintonización estática que consisten en plantear la búsqueda de parámetros como un problema de optimización, esto se conoce como metaoptimización, algunos ejemplos son irace [24], ParamILS [25] y SMAC [26].

b) *Sintonización On-line:* Aquí, los valores de cada parámetro son determinados de manera dinámica en tiempo de ejecución. Cada técnica posee diferentes niveles de complejidad, ventajas y desventajas que deben ser analizadas en detalle junto con el tipo de problema a tratar.

Cabe destacar que debido al tipo de fenómeno tratado, la complejidad de la arquitectura interna del método, el grado de utilización de recursos necesario y la cantidad de parámetros involucrados, el análisis de calibración de parámetros realizado en el presente trabajo es de tipo estático con propósito de sondeo basado en un LHD.

#### IV. ESS-IM APLICADO EN LA PREDICCIÓN DEL COMPORTAMIENTO DE INCENDIOS FORESTALES

ESS-IM es un método paralelo de reducción de incertidumbre que ha sido diseñado para ser aplicado en sistemas de predicción que involucren simulación. Dicho método ha sido validado en la predicción del comportamiento de incendios forestales. Como todo método o sistema de predicción, en ESS-IM se persiguen dos claros objetivos: mejorar la calidad de las predicciones realizadas, e incrementar la eficiencia computacional, permitiendo así brindar mejores resultados en el menor tiempo posible.

Tal como se ha mencionado anteriormente, ESS-IM utiliza como técnica de optimización PEAs con un esquema de paralelización, basado en islas con doble jerarquía master-worker.

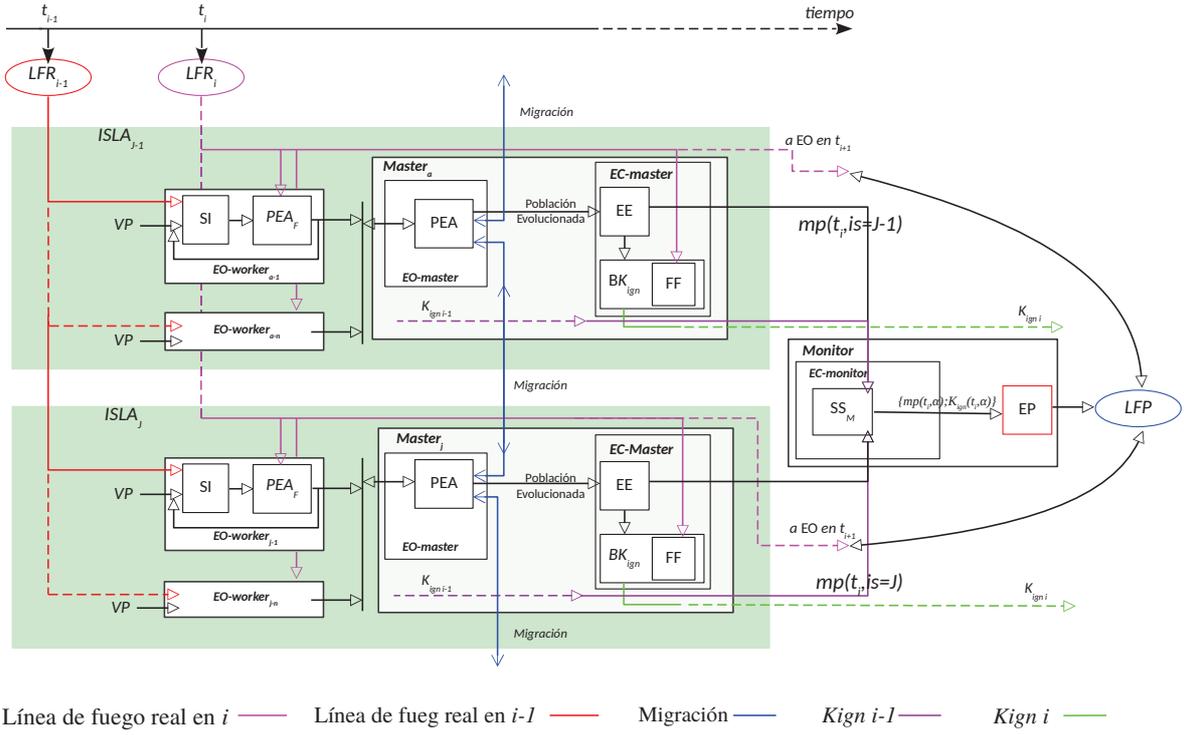


Figura 7: Evolutionary Statistical System with Island Model: **SI**: simulador de incendios; **PEA**: algoritmo evolutivo paralelo; **PEA<sub>F</sub>**: algoritmo evolutivo paralelo (evaluación de fitness); **EO**: etapa de optimización; **EE**: etapa estadística; **BK<sub>ign</sub>**: búsqueda de  $K_{ign}$ ; **K<sub>ign</sub>**: valor clave de ignición; **FF**: función de fitness; **EC**: etapa de calibración; **EP**: etapa de predicción; **LFP**: línea de fuego predicha; **LFR<sub>x</sub>**: línea de fuego real en tiempo  $x$ ; **VP**: vector de parámetros de entrada; **EE<sub>M</sub>**: etapa estadística en proceso monitor;  $mp$ : mapa de probabilidad.

En la Fig. 7 se presenta un esquema general del funcionamiento de ESS-IM, donde el proceso *monitor* (equivalente a *Master* islas global de Fig. 6), inicia el proceso predictivo, enviando a las *masters* de cada isla los datos del incendio a tratar (i.e., mapa con la línea de fuego inicial, rangos de parámetros conocidos, intervalos de tiempo a evaluar, etc.). Cuando todos los *masters* cuentan con la información requerida, la etapa de optimización del master (*EO<sub>master</sub>*) da inicio al PEA, el cual se encuentra dividido en dos sub-etapas: *PEA* y *PEA<sub>F</sub>*. La primera se encarga de inicializar las poblaciones de cada isla, realizar las modificaciones en los individuos de cada población mediante los diferentes operadores (i.e., selección, cruzamiento, mutación, reemplazo y migración), enviar los individuos a los workers (para que sean simulados y evaluados) y avanzar hacia la siguiente generación. La segunda sub-etapa corresponde a la porción del PEA que se lleva a cabo en los workers, donde se calcula el valor de aptitud de cada individuo mediante la utilización del simulador de incendios (SI). El simulador de propagación de incendios forestales utilizado está basado en el modelo desarrollado por Rothermel [27], e implementado para ambientes computacionales mediante la librería fireLib [28]. La aptitud de cada individuo es calculada mediante la utilización de la línea de fuego real del incendio (LFR) en  $t_{i-1}$  (i.e.,  $LFR_{i-1}$ ), junto con los valores de cada

parámetro de entrada contenidos en cada individuo (i.e., Vector de Parámetros VP). Es importante remarcar que todas estas funciones son llevadas a cabo por la etapa de optimización que opera en los procesos *workers* (*EO<sub>worker</sub>*).

Todos los individuos evaluados son enviados a la *EO<sub>master</sub>*, etapa encargada de almacenar y calcular los resultados parciales, realizar la migración de individuos entre las diferentes islas (en función de la topología de comunicación previamente establecida), evolucionar cada población e iterar hasta que se cumpla la condición de finalización. Posteriormente, las poblaciones evolucionadas son introducidas en la etapa de calibración del proceso *master* (*EC<sub>master</sub>*), donde se genera un mapa de probabilidad ( $m_p$ ) que considera la totalidad de los mapas simulados a partir de cada individuo. El mapa de probabilidad calculado en  $t_i$  por la isla  $is$  ( $is = 1, 2, \dots, j$ ), i.e.,  $m_p(t_i, is)$  es utilizado para calcular el valor clave de ignición ( $K_{ign}$ , por sus siglas en inglés). Este valor representa numéricamente el patrón de comportamiento del incendio y se obtiene en la etapa búsqueda de  $K_{ign}$  (*BK<sub>ign</sub>*) [4]. Es importante tener en cuenta que el valor clave de ignición generado en el instante  $t_i$ , es utilizado para efectuar la predicción de la línea de fuego en  $t_{i+1}$  y calculado en  $t_0$ , por este motivo ESS-IM no puede efectuar predicciones en el instante  $t_0$ .

Además, es importante remarcar que en todo instante de

TABLA II: LISTA DE EXPERIMENTOS SEGÚN LHD.

Experimento	Umbral	Cdad. iteraciones	Cdad. individuos	Cdad. ind. a migrar	Frec. de migración	Prob. cruce (%)	Prob. mutación (%)
0	0.70	200	200	5	5	60	0.5
1	0.78	373	68	3	2	44	0.7
2	0.75	419	194	17	4	37	1.7
3	0.74	198	175	3	7	19	1.1
4	0.71	302	41	11	10	42	0.3
5	0.81	172	25	9	3	22	2.0
6	0.56	359	125	4	9	27	2.1
7	0.68	129	145	20	7	24	2.2
8	0.69	117	91	5	7	58	1.7
9	0.59	429	187	7	5	30	0.4
10	0.57	144	51	6	4	28	0.6
11	0.65	249	97	19	8	10	0.9
12	0.73	454	39	9	12	40	0.35
13	0.84	465	110	12	3	13	1.5
14	0.62	263	15	18	6	53	1.4
15	0.64	328	62	16	3	17	1.3
16	0.80	293	123	13	9	33	0.5
17	0.67	486	162	15	6	56	1.1
18	0.82	234	139	9	5	48	2.4
19	0.77	398	37	8	9	39	1.8
20	0.161	304	84	14	1	45	2.4

tiempo  $i$ , en cada isla  $is$ , se genera un valor clave de ignición  $K_{ign}(t_i, is)$  y un mapa de probabilidad  $m_p(t_i, is)$ , los cuales son enviados a la etapa de calibración del proceso *monitor* ( $EC_{Monitor}$ ). Aquí, la etapa estadística ( $EE_M$ ) generará en  $t_i$  un par  $\{m_p(t_i, \alpha); K_{ign}(t_i, \alpha)\}$ , donde  $\alpha$ , indica el número de isla cuyo par de valores ha obtenido el mejor valor de fitness. Éstos valores finalmente son ingresados en la etapa de predicción (EP), la cual realizará el cálculo de predicción (i.e., línea de fuego predicha) para el próximo instante de tiempo  $i + 1$ , o sea  $LF R_{i+1}$ . Al utilizar los valores de la isla que mejor resultados obtiene, se prioriza la calidad de predicción del método sobre el rendimiento. Otras formas de realizar la predicción consiste en: a) calcular un valor clave de ignición en función de un mapa estadístico generado a partir de los mapas de probabilidad resultantes de todas las islas, o b) utilizar el  $K_{ign}$  de la isla que finalice en menor tiempo, tratando de obtener el mejor rendimiento posible.

## V. EXPERIMENTACIÓN Y RESULTADOS

En esta sección se describe cómo fueron diseñados, desarrollados y lanzados los experimentos, y cómo se analizaron los resultados obtenidos.

### V-A. Diseño de experimentos

En primer lugar, es necesario determinar cuáles parámetros serán sometidos al análisis de calibración y que rangos de valores serán considerados para cada uno de ellos. Para ello se tomaron aquellos parámetros que no fueron considerados en [11]. En la Tabla I se listan los parámetros incluidos en el estudio de calibración, junto con sus rangos asociados. Los parámetros con valores fijos no son calibrados, pero los valores indicados son los utilizados en todos los experimentos. Es importante tener en cuenta que los rangos utilizados para cada parámetro han sido seleccionados en función de la literatura y trabajos previos [11][29].

Una vez definidos los parámetros a calibrar y sus respectivos rangos, es necesario determinar cuántos experimentos

se realizarán y que configuración de parámetros tendrá cada uno de ellos. Para ello, tal como se comentó en la Sección III, se desarrolló un diseño de experimento por Hipercono Latino, mediante el software estadístico **R** [30]. En la Tabla II se observa cada uno de los experimentos del LHD con sus valores de parámetros, la fila 0 corresponde a la configuración de parámetros utilizada en trabajos previos (configuración clásica). El resto de las filas (del 1 al 20) corresponden a los experimentos del LHD.

### V-B. Casos de estudio

Los casos de estudio utilizados para evaluar los experimentos del LHD corresponden a dos incendios controlados (denominados Caso A y Caso B), ambos realizados en Portugal bajo el proyecto Spread [31]. En las Figs. 8a y 9a se puede observar una fotografía de cada incendio y en las Figs. 8b y 9b una representación computacional de la progresión total de ambos incendios, en ambos el fuego ha sido iniciado de forma lineal sobre la parte inferior del terreno. La Tabla III brinda información adicional sobre cada terreno y su correspondiente incendio, puntualmente podemos ver que el incendio del Caso A es comienza en el minuto 2,5 con incrementos de igual valor dando un total de 3 pasos de predicción: desde el minuto 5 al 7,5, del 7,5 al 10 y del 10 al 12,5. Y el incendio del Caso B da inicio en el minuto 2 hasta el minuto 14 con incrementos de 2 minutos, dando un total de 5 pasos de predicción (i.e., minutos 4 a 6, 6 a 8, 8 a 10, 10 a 12 y 12 a 14). Es importante remarcar que durante en primer intervalo de tiempo, entre  $t_0$  y  $t_1$ , ESS-IM calcula el primer valor  $K_{ign}$ , por lo que no puede generar predicción en dicho momento, para cada incendio estos intervalos corresponden a los siguientes: Caso A, desde minuto 2,5 al minuto 5, y Caso B, desde el minuto 2 al minuto 4.

### V-C. Función de Fitness

La calidad de predicción ha sido evaluada mediante una función de aptitud basada en el índice Jaccard [32] (compa-

TABLA III: DESCRIPCIÓN DE LOS CASOS DE ESTUDIO: DIMENSIÓN, PENDIENTE Y TIEMPO DE INICIO, FIN E INCREMENTO.

Caso	Ancho (m)	Alto (m)	Pendiente (grados)	Tiempo inicial (min)	Incremento (min)	Tiempo final (min)
A	89	91	21	2.5	2.5	12.5
B	89	109	21	2	2	14

ración del mapa predicho y el mapa real). Éste cálculo se realiza mediante una función (Ecuación 1) que considera una representación basada en celdas.

$$Fitness = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

En la Ecuación 1,  $A$  representa el conjunto de celdas en el mapa real sin el subconjunto de celdas quemadas antes de iniciar las simulaciones, y  $B$  el conjunto de celdas en el mapa simulado sin el subconjunto de celdas quemadas antes de comenzar la simulación. La corrección en el conjunto de celdas mediante la eliminación del subconjunto de celdas quemadas se utiliza para evitar resultados sesgados. Todo valor de aptitud se encontrará en el rango  $[0-1]$ , donde un valor igual a 1 representa una predicción perfecta, y un valor igual a 0, indica una predicción completamente errónea [4].

#### V-D. Resultados

La ejecución de los experimentos se realizó evaluando tanto el tiempo de ejecución promedio de cada configuración de parámetros, como así también la calidad de predicción para cada paso de predicción ( $t_i$ ). Por ejemplo, para cada uno de los 20 experimentos, en el Caso A, se obtuvieron tres valores de predicción: minutos 7,5, 10 y 12,5, y para el Caso B, cinco valores: minutos 6, 8, 10, 12 y 14. Debido a que ESS-IM posee un comportamiento estocástico, se fijaron 30 semillas para ser utilizadas en cada experimento. Por lo tanto, por cada configuración de parámetros del LHD se realizaron 30 ejecuciones, una con cada semilla, y se tomó como resultado final (para cada paso de predicción) el promedio de éstas. Es importante tener en cuenta que el conjunto de 30 semillas fue el mismo que se utilizó en cada experimento; los cuales fueron ejecutados en un clúster Linux, bajo un entorno MPI [33] y Gigabit Ethernet.

Los resultados obtenidos tras el análisis de calibración han sido comparados con los de trabajos previos, donde se utilizaba la configuración clásica de parámetros sobre los mismos casos de estudio. En las Tablas IV y V, se pueden observar los resultados de calidad de predicción de cada experimento para cada paso de predicción, junto con el tiempo consumido por cada configuración de parámetros (en negrita se han resaltado los mejores valores de cada paso de predicción y el menor tiempo). Con el objetivo de cuantificar de forma global cada configuración de parámetros, se ha incluido como métrica adicional, un valor promedio de calidad de predicción para cada experimento. De esta manera podemos determinar aquellas configuraciones de parámetros que obtienen mejor rendimiento en términos de calidad de predicción.

Para comparar de forma clara los resultados obtenidos, en las Figs. 11 y 12 se han graficado las curvas de calidad de

predicción para cada caso de estudio. Con el objetivo de simplificar la interpretación de dichos resultados, sólo se han graficado cinco resultados de interés por cada incendio, estos son:

- La línea routulada como *Clásica* representa los valores de calidad de predicción obtenidos utilizando la configuración clásica de parámetros (i.e., fila 0 de la Tabla II).
- Los tres mejores resultados en función de la métrica promedio:
  - Caso A: experimentos: 3, 4 y 5.
  - Caso B: experimentos: 9, 18 y 20.
- La configuración de parámetros óptima seleccionando manualmente los mejores resultados de cada experimento y cada paso de predicción. La línea bajo el nombre de *Exp.21*, representa los mejores resultados de calidad de predicción obtenidos tras comparar los 20 experimentos del LHD y la configuración clásica. En este caso los valores de calidad de predicción no corresponden a una única configuración de parámetros, es decir, los valores corresponden a los resaltados en negrita en las Tablas IV y V.

En función los resultados obtenidos podemos observar que no es posible determinar una única configuración de parámetros óptima, debido que los mejores valores encontrados (para cada paso de predicción) no son generados por una misma configuración de parámetros. Por lo tanto, una determinada configuración puede brindar en  $t_i$ , un valor de alta calidad de predicción, y en  $t_{i+1}$  lo contrario, o viceversa. Además, es importante observar que si bien se logra incrementar la calidad en diferentes pasos de predicción en ambos casos de estudio, los mejores valores (sin considerar el resultado óptimo) son obtenidos por la configuración clásica. Sin embargo nos permite evidenciar dos aspectos importantes, por un lado, la importancia de efectuar un estudio completo de sintonización, utilizando todos los parámetros a la vez y usando una técnica de metaoptimización, y por el otro, que ESS-IM puede realizar predicciones de calidad de predicción superior si se aborda una implementación heterogénea, en términos de parámetros evolutivos.

La implementación actual de ESS-IM utiliza una configuración de parámetros evolutivos de forma global, i.e., todas las islas evolucionan siguiendo el mismo criterio de búsqueda. Si bien cada isla inicializa su población utilizando una semilla diferente (lo que permite recorrer el espacio de búsqueda de forma más eficiente), todas las islas evolucionan siguiendo la misma configuración de parámetros. Tal como se menciona en [34] y en función de los resultados obtenidos, se observa que

TABLA IV: CASO A: RESULTADOS DE CALIDAD DE PREDICCIÓN Y TIEMPO DE EJECUCIÓN.

Experimento	Paso1	Paso2	Paso3	Promedio	Tiempo (seg.)
0	0,845428	<b>0,843672</b>	0,7690385	0,8193795	2988,665519
1	0,106004	0,108577	0,772652	0,3290776667	2612,619676
2	0,150586	0,80131	0,610826	0,5209073333	2244,432355
3	0,286221	0,813278	0,80639	0,6352963333	1429,851508
4	<b>0,862664</b>	0,779093	0,800682	0,8141463333	1206,655527
5	0,672522	0,576964	0,801539	0,683675	1713,710328
6	0,200071	0,793792	0,682604	0,5588223333	1382,225014
7	0,25554	0,801032	0,781745	0,6127723333	1253,853691
8	0,090953	0,81204	0,730489	0,544494	2212,788611
9	0,136276	0,8064392	0,81601	0,5862417333	2008,577016
10	0,239204	0,642478	0,793625	0,5584356667	1696,546906
11	0,0740595	0,732347	0,5	0,4354688333	1205,362326
12	0,0742444	0,79956	<b>0,82475</b>	0,5661848	1233,445413
13	0,245013	0,733593	0,763948	0,5808513333	4100,662664
14	0,158352	0,783797	0,7801559	0,5741016333	<b>870,643627</b>
15	0,0740207	0,640212	0,79864	0,5042909	1105,949597
16	0,0740498	0,7904	0,808626	0,5576919333	2261,60852
17	0,115447	0,715506	0,553548	0,4615003333	2756,909569
18	0,228006	0,72432	0,80432	0,5855486667	4860,861947
19	0,0979542	0,642802	0,806483	0,5157464	2441,529033
20	0,161613	0,628205	0,81578	0,5351993333	1830,742101

TABLA V: CASO B: RESULTADOS DE CALIDAD DE PREDICCIÓN Y TIEMPO DE EJECUCIÓN.

Experimento	Paso1	Paso2	Paso3	Paso4	Paso5	Promedio	Tiempo (seg.)
0	<b>0,7837408</b>	0,73424225	<b>0,785167</b>	0,7625736	0,7627824	<b>0,76570121</b>	3192,044672
1	0,1170257	0,876032	0,7187505	0,589248	0,666653	0,59354184	1498,11837
2	0,2778965	0,6598605	0,6486855	0,815	0,763179	0,6329243	2449,91527
3	0,239341	0,834291	0,721708	0,6763865	0,729711	0,6402875	1664,0920345
4	0,08269195	0,8067395	0,7594295	0,7647505	0,7076375	0,62424979	1601,167767
5	0,4682175	0,3455755	0,16225	0,4615845	0,790429	0,4456113	2794,542419
6	0,08166265	0,7454375	0,726323	0,759956	0,704482	0,60357223	2350,898576
7	0,3033375	0,723262	0,75049	0,7276985	0,6625795	0,6334735	1566,0197
8	0,11720365	0,6676135	0,631893	0,8344765	0,693067	0,58885073	1992,5737195
9	0,338391	0,7896695	0,772685	0,809299	<b>0,7965525</b>	0,7013194	2219,045241
10	0,12674535	0,7686335	0,665308	0,4499535	0,649841	0,53209627	1780,422953
11	0,0613998	<b>0,8829155</b>	0,6534615	<b>0,838659</b>	0,7605645	0,63940006	1680,966823
12	0,13265145	0,737667	0,718484	0,6731375	0,737675	0,59992299	1336,601607
13	0,1891811	0,14659734	0,759166	0,7312195	0,665361	0,498304988	6476,568882
14	0,27022865	0,7780405	0,7059405	0,616361	0,7913965	0,63239343	<b>543,8564625</b>
15	0,1213762	0,719276	0,611154	0,6411255	0,7299095	0,56456824	2054,585647
16	0,0388287	0,6261275	0,7768405	0,76491	0,599447	0,56123074	3149,3421195
17	0,10974265	0,8154585	0,7445985	0,7355995	0,780165	0,63711283	2620,3725005
18	0,2101255	0,862881	0,6853575	0,680936	0,772992	0,6424584	5334,4085265
19	0,0856205	0,662743	0,6850965	0,6971825	0,4949295	0,5251144	1797,251595
20	0,2460463	0,878967	0,642796	0,795094	0,6981135	0,65220336	1778,7067545

la robustez del método puede ser incrementada mediante el uso de una configuración heterogénea de parámetros evolutivos.

Es importante tener en cuenta, que para el Caso A, los resultados de calidad de predicción previos a la calibración se encontraban por encima del 0,7690385, lo que deja un margen de mejora bastante reducido, equivalente a 0,23096 (ya que difícilmente se pueda alcanzar un valor de fitness próximo a 1). Por este motivo y debido a que se está tratando de predecir un fenómeno natural de alta complejidad, se puede decir que los valores obtenidos son considerablemente importantes. En el Caso B, se produce una situación similar, el margen de mejora es igual a 0,26575775. Sin embargo, en este caso los incrementos de calidad de predicción son un tanto superiores, principalmente en el minuto 8 y 12.

Si observamos el comportamiento de los resultados en función del tiempo de ejecución, podemos observar que las configuraciones de parámetros con valores elevados, consu-

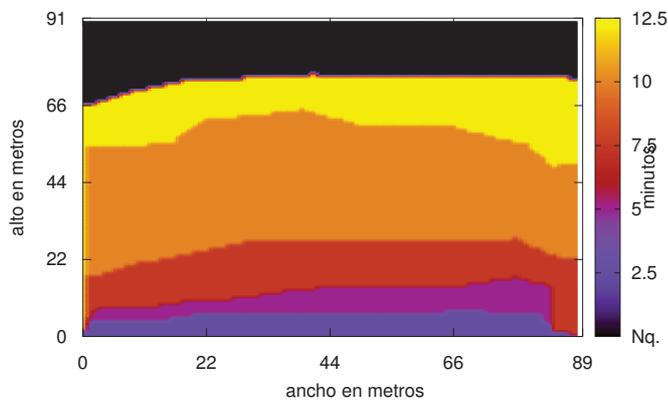
men mayor cantidad de tiempo. Por ejemplo, el experimento que utiliza mayor cantidad de tiempo (para ambos casos de estudio) es el número 13, el cual posee el mayor umbral de fitness de todo el diseño, igual a 0,84, y una alta cantidad de iteraciones equivalente a 465. Estos valores obligan a la metaheurística a iterar hasta que se alcance dicho valor de umbral o al llegar a la cantidad de 465 iteraciones evolutivas. En contraste, la configuración que menor tiempo consume es la número 14, la cual tienen valores intermedios de umbral de fitness y cantidad de iteraciones, pero una reducida cantidad de individuos, igual a 15. Finalmente también es interesante observar el tiempo consumido por las dos mejores configuraciones. Por ejemplo para el Caso A, la mejor configuración (i.e., experimento 0) emplea un total de 2988,66 segundos, con un valor promedio igual a 0,8193795, y la segunda mejor configuración (i.e., experimento 4) emplea 1206,65 segundos, con un valor promedio igual a 0,8141463333. En el Caso B,



(a) Incendio real.

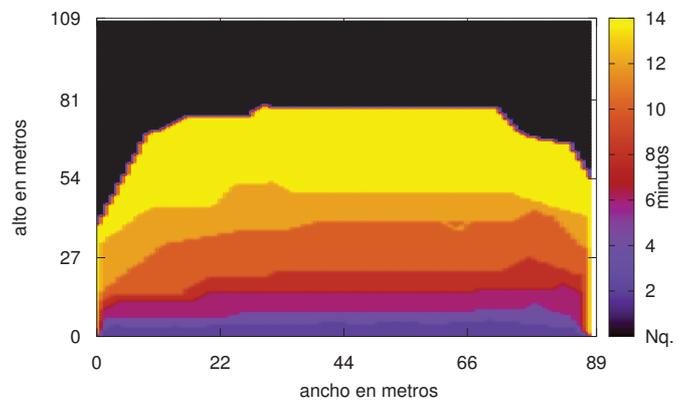


(a) Incendio real.



(b) Representación de la propagación del incendio.

Figura 8: Información sobre el caso de prueba: a) fotografía del incendio controlado en Gestosa, b) representación computacional del incendio utilizado por ESS-IM.



(b) Representación de la propagación del incendio.

Figura 9: Información sobre el caso de prueba: a) fotografía del incendio controlado en Gestosa, b) representación computacional del incendio utilizado por ESS-IM.

la mejor configuración (i.e., experimento 0) utiliza 3192,04 seg., con un promedio igual a 0,76570121, y la siguiente (i.e., experimento 9) consume 2219,04 seg. con un promedio de calidad de predicción equivalente a 0,7013194. En el Caso A, la reducción de tiempo es superior al 50% con una diferencia mínima en la calidad promedio, en cambio en el Caso B, la reducción no es tan significativa, y la diferencia en términos de calidad de predicción es bastante más notoria, lo cual puede justificar el mayor uso de recursos obteniendo una mejor calidad de predicción.

## VI. CONCLUSIONES

ESS-IM es un método general de reducción de incertidumbre desarrollado para operar en un ambiente paralelo/distribuido, el cual ha sido aplicado a la predicción del comportamiento de incendios forestales. ESS-IM centra su funcionamiento en la utilización de Algoritmos Evolutivos Paralelos

como herramienta de optimización, bajo un esquema de paralelización basado en islas. Los PEA poseen ciertos parámetros de control que condicionan su funcionamiento, por lo que sus valores deben ser cuidadosamente seleccionados. En el presente trabajo se ha presentado un análisis de calibración estática de los parámetros evolutivos de ESS-IM, para ser utilizados en una implementación que opere con una configuración heterogénea de parámetros. El estudio desarrollado evalúa el método con diferentes configuraciones de parámetros sobre dos casos reales de incendios controlados. Mediante los resultados obtenidos se ha podido determinar que la calibración de parámetros permite incrementar la calidad de predicción del método en más del 60% de los pasos de predicción, destacando la importancia de todo análisis de sintonización. Además, que no existe una configuración de parámetros óptima, ya que los mejores resultados de predicción a lo largo de distintos instantes de tiempo, para un mismo incendio, se

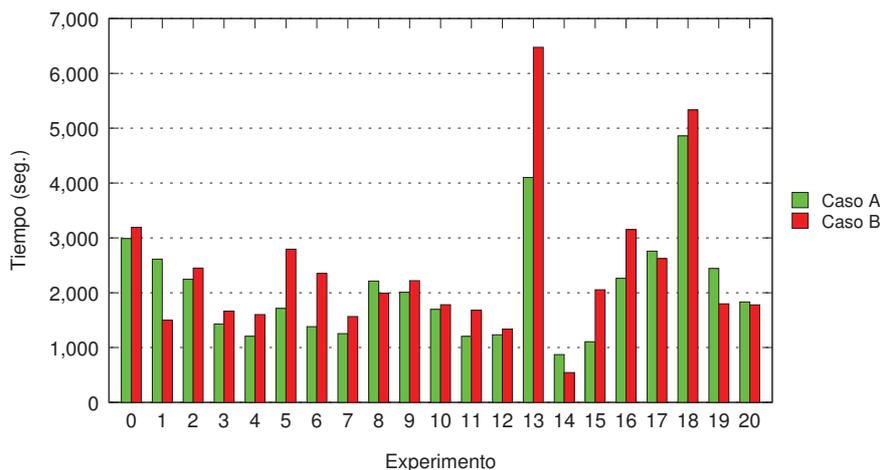


Figura 10: Comparación de tiempo de ejecución para cada experimento y caso de estudio.

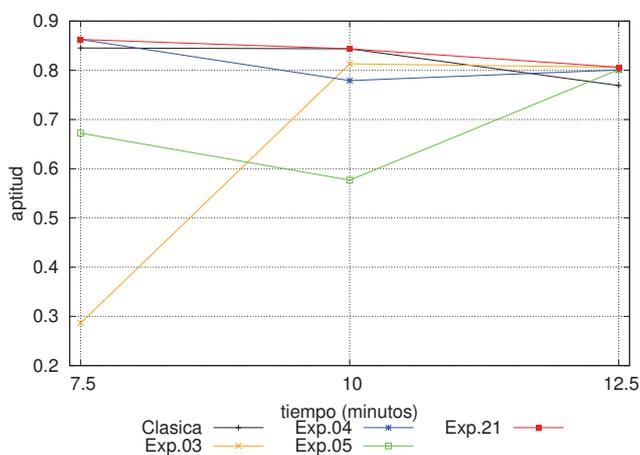


Figura 11: Caso A: Comparación de calidad de predicción utilizando configuración clásica de parámetros, las tres mejores configuraciones y la ideal.

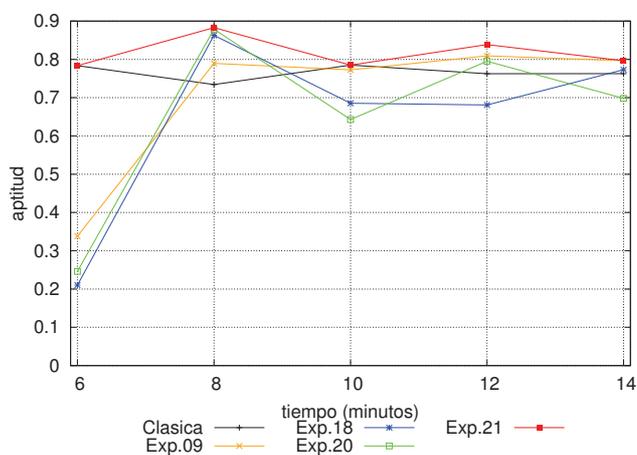


Figura 12: Caso B: Comparación de calidad de predicción utilizando configuración clásica de parámetros, las tres mejores configuraciones y la ideal.

obtienen mediante diferentes configuraciones de parámetros. Este comportamiento justifica la implementación de ESS-IM con una configuración heterogénea de parámetros evolutivos. Como trabajo futuro se realizará un análisis de sintonización estático basado en técnicas de metaoptimización incluyendo la totalidad de los parámetros del método. Posteriormente se evaluará la implementación de ESS-IM junto a una técnica de sintonización dinámica de parámetros, donde se evalúen los efectos de las variaciones de cada parámetro y las interdependencias entre ellos de forma automática. También se considera aconsejable acompañar los futuros trabajos de calibración con un análisis de rendimiento y utilización de recursos, ya que los parámetros evolutivos impactan directamente sobre dicha característica.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado por UTN bajo el proyecto EIUTIME0003939TC y EIUTNME0003952, y por el Ministerio de Economía y Competitividad de España bajo el proyecto TIN2014-53234-C2-1-R. El primer autor agradece a CONICET por la beca doctoral otorgada.

#### REFERENCIAS

- [1] Corporación Chilena de la Madera, El número y la magnitud de los incendios forestales constituyen una preocupación relevante para CORMA. <http://www.corma.cl/lineas-de-accion/incendios-forestales>, accedido el 25 de mayo de 2016.
- [2] Diario El Mercurio, Declaran alerta roja en Chile Chico por incendio forestal que ha consumido más de 300 hectáreas. <http://www.emol.com/noticias/Nacional/2016/02/10/787799/Declaran-alerta-roja-en-Chile-Chico-por-incendio-forestal-que-ha-consumido-mas-de-300-hectareas.html>, accedido el 25 de mayo de 2016.
- [3] R. Buyya, *High Performance Cluster Computing: Architectures and Systems*. Prentice Hall, 1999.

- [4] M. Méndez-Garabetti, G. Bianchini, P. Caymes-Scutari, M. L. Tardivo, "Increase in the quality of the prediction of a computational wildfire behavior method through the improvement of the internal metaheuristic," *Fire Safety Journal*, Volume 82, May 2016, Pages 49-62, ISSN 0379-7112, <http://dx.doi.org/10.1016/j.firesaf.2016.03.002>.
- [5] M. Méndez-Garabetti, G. Bianchini, M. L. Tardivo, P. Caymes-Scutari, "Comparative Analysis of Performance and Quality of Prediction Between ESS and ESS-IM," *Electronic Notes in Theoretical Computer Science*, Volume 314, 15 June 2015, Pages 45-60, ISSN 1571-0661, <http://dx.doi.org/10.1016/j.entcs.2015.05.004>.
- [6] F. Outtier C. Ourtier. Data assimilation concepts and methods, Training Course Notes, ECMWF, 1999.
- [7] G. Bianchini, M. Denham, A. Cortés, T. Margalef y E. Luque: "Wildland Fire Growth Prediction Method Based on Multiple Overlapping Solution," *Journal of Computational Science*, vol. 1(4), 2010, pp. 229–237.
- [8] G. Bianchini, M. Denham, A. Cortés, T. Margalef y E. Luque, "Improving forest-fire prediction by applying a statistical approach," en *Forest Ecology and Management*, vol. 234, s. 1, 2006, pp. 210.
- [9] G. Bianchini, M. Méndez-Garabetti y P. Caymes Scutari, "Evolutionary-Statistical System for Uncertainty Reduction Problems in Wildfires," en XVIII CACIC, XII Workshop de Procesamiento Distribuido y Paralelo (WDPDP). Bahía Blanca, Argentina, 2012, pp. 230–238.
- [10] G. Bianchini y P. Caymes Scutari, "Uncertainty Reduction Method Based on Statistics and Parallel Evolutionary Algorithms," en *Proceedings of High-Performance Computing Symposium - 40 JAIIO*. HPC 2011, Córdoba, Argentina, 2011, pp. 1–4.
- [11] M. Méndez Garabetti, G. Bianchini, P. Caymes Scutari, M. L. Tardivo, "Sistema Estadístico Evolutivo con Modelo de Islas: Calibración de parámetros evolutivos," 13 y 14 de Agosto de 2015, 6th International Symposium on Innovation and Technology, ISSN 2305-9958, ISBN 978-612-45917-9-2, IITEC 2015, pp. 77-81, IITEC (International Institute of Innovation and Technology), Lima (Perú)
- [12] E. Alba, F. Luna, A.J. Nebro, and J.M. Troya. Parallel heterogeneous genetic algorithms for continuous optimization. *Parallel Computing*, 30(5-6):699–719, 2004.
- [13] E. Alba, F. Luna, and A.J. Nebro. Advances in parallel heterogeneous genetic algorithms for continuous optimization. *International Journal of Applied Mathematics and Computer Science*, 14(3):101–117, 2004.
- [14] F. Herrera and M. Lozano. Gradual distributed real-coded genetic algorithms. *IEEE Transaction on Evolutionary Computation*, 4(1):43–63, 2000.
- [15] C. Darwin, *On the Origin of Species by Means of Natural Selection*. London, Murray, 1859.
- [16] T. Mattson, B. Sanders y B. Massingill, *Patterns for Parallel Programming*. Addison-Wesley, 2005.
- [17] E. Alba and J. M. Troya. 1999. "A survey of parallel distributed genetic algorithms," *Complexity* 4 (4) (March 1999), 31-52. DOI=[http://dx.doi.org/10.1002/\(SICI\)1099-0526\(199903/04\)4:4:31::AID-CPLX5<3.3.CO:2-W](http://dx.doi.org/10.1002/(SICI)1099-0526(199903/04)4:4:31::AID-CPLX5<3.3.CO:2-W)
- [18] Y. Gong, W. Chen, Z. Zhan, J. Zhang, Y. L., Q. Z., J. L., "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Applied Soft Computing*, Volume 34, September 2015, Pages 286-300, ISSN 1568-4946, <http://dx.doi.org/10.1016/j.asoc.2015.04.061>.
- [19] E. Eiben, Z. Michalewicz, M. Schoenauer, J. Smith. Parameter Control in Evolutionary Algorithms. Fernando G. Lobo and Cláudio F. Lima and Zbigniew Michalewicz. Parameter Setting in Evolutionary Algorithms, 54 (54), Springer Verlag, pp.19-46, 2007, *Studies in Computational Intelligence*, 978-3-540-69431-1.
- [20] E. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley and Sons, 2009.
- [21] M. D. McKay, R. J. Beckman y W.J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, 21(2), 1979, pp. 239–245
- [22] R. L. Iman y W.J. Conover, "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment," *Communications in Statistics, Part A. Theory and Methods*, Vol. 17, 1980, pp. 1749-1842.
- [23] D. Montgomery. *Design and Analysis of Experiments*. Wiley, 1984.
- [24] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari, "The irace package, iterated race for automatic algorithm configuration," Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
- [25] H. R. Lourenço, O.C. Martin, and T. Stützle, "Iterated Local Search: Framework and Applications," In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*. inbook, Boston, MA: Springer US. [http://doi.org/10.1007/978-1-4419-1665-5\\_12](http://doi.org/10.1007/978-1-4419-1665-5_12), 2010, pp. 363-397.
- [26] F. Hutter, H.H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," In *Learning and Intelligent Optimization*, 5th International Conference, LION 5, Lecture Notes in Computer Science. Springer, Heidelberg, Germany, 2011.
- [27] R. C. Rothermel, "A mathematical model for predicting fire spread in wildland fuels," Res. Pap. INT-115, US Dept. of Agric., Forest Service, Intermountain Forest and Range Experiment Station. (Ogden, UT.), 1972.
- [28] C. D. Bevins, 2004. *FireLib User Manual & Technical Reference*. <http://www.fire.org/>
- [29] M. Méndez Garabetti, G. Bianchini, M. L. Tardivo, P. Caymes Scutari, "Método de reducción de incertidumbre aplicado a la Predicción de incendios: análisis de sintonización Estática," Mayo de 2015. *Matemática Aplicada, Computacional e Industrial (MACI)*, Vol 5 2015. pp. 241-244. ISSN 2314-3282. ASAMACI.
- [30] R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>, 2015.
- [31] D. X. Viegas, "Project Spread - Forest Fire Spread Prevention and Mitigation," <http://www.algosystems.gr/spread/>, 2004.
- [32] R. Real, J.M. Vargas, *The Probabilistic Basis of Jaccard's Index of Similarity*. *Systematic Biology*, 45 Num 3 (1996) pp. 380–385.
- [33] W. Gropp, E. Lusk y A. Skjellum, *Using MPI - Portable Parallel Programming with the Message-Passing Interface*. The MIT Press, 1999.
- [34] E. Alba, F. Luna, A.J. Nebro, and J. M. Troya. Parallel heterogeneous genetic algorithms for continuous optimization," *Parallel Computing*, 30(5-6):699-719, 2004.