Aplicación de Autómatas Celulares a Simulación Básica de Incendios Forestales

Marco J. Peredo, Ramiro Ramallo

Maestía en Ciencias de la Computación
Universidad Católica Boliviana San Pablo
Cochabamba, Bolivia
e-mail: peredom@ucbcba.edu.bo, rramallo@ucbcba.edu.bo

Resumen

El presente artículo explica brevemente el concepto de autómata celular, y expone resultados obtenidos con uno de estos autómatas, programado como una entidad de software sobre ambiente de procesamiento paralelo, en la simulación básica de un incendio forestal, sobre una matriz que representa un bosque de n por n árboles correspondientes a una misma especie. Se discute así mismo las posibilidades de crecimiento modular del programa, para adecuar la versión básica mostrada, a condiciones más complejas (bosques de gran extensión, condiciones de terreno cambiantes, especies diversas, etc.) de manera versátil y fácil de llevar a cabo en computadora.

Palabras Clave: autómatas celulares; paralelismo; simulación usando autómatas celulares; simulación de incendios forestales.

1. Introducción

Los resultados de la simulación de incendios forestales son de gran utilidad para la planificación y mantenimiento de grandes áreas boscosas, ya sean naturales o creadas por el hombre, puesto que permiten conocer de antemano dónde deben existir zonas de contención, además de contribuir a la toma de decisiones en caso de un incendio real.

En el caso de presentarse un incendio es muy importante tener el sistema listo para ingresar datos inherentes al incendio, tales como: el foco del mismo y las condiciones meteorológicas actuales. Con el fin de obtener resultados fiables, se deben realizar varias simulaciones cuyos valores deben ser conocidos en el menor tiempo posible. A partir de los resultados, se podrá realizar el trabajo de combatir el fuego determinando, por ejemplo, posiciones y direcciones óptimas para la construcción de barreras. Se minimiza de esta manera el área afectada por el fuego.

En general, las áreas forestales ocupan grandes extensiones de terreno y por ende la cantidad de especies forestales es elevada, pudiendo resultar demasiado grande como para realizar la simulación en una máquina secuencial, por tanto conviene mostrar que el uso de una máquina o arquitectura paralela contribuye a optimizar el tiempo de respuesta para la realización de la simulación.

2. Autómatas Celulares

Se debe pensar en un Autómata Celular (AC) como en un conjunto de objetos situados sobre una región geográfica o asociados a puntos de un espacio y susceptibles de adquirir ciertos estados según transcurre el tiempo, siempre en forma discreta o a saltos. Estos objetos cambian sus estados en función de sus propios estados previos y de los de aquellos otros entes o individuos situados en su vecindad. *El problema es entonces saber cuál será la evolución de la configuración del sistema según avance el tiempo.*

2.1. Descripción general de un AC

En pocas palabras, un AC es un modelo formal que está compuesto por un conjunto de entes elementales, cada uno de ellos susceptible de encontrarse en un cierto estado y de alterarlo de un instante al siguiente, asumiendo que el tiempo transcurre de forma discreta. La regla que gobierna la transición de estados en los entes es sensible a los estados de los demás elementos en su vecindad, siendo por tanto una *regla de transición local.* El aspecto que más caracteriza a los ACs es su capacidad para dotar al conjunto del sistema, visto como un todo, de una serie de propiedades emergentes inducidas por la propia dinámica local. En general, no es fácil obtener las propiedades globales de un sistema definido como el anterior, complejo por naturaleza, a no ser por vía de la simulación, partiendo de un estado inicial de la población de objetos y cambiando en cada instante los estados de todos ellos de forma síncrona.

La definición de un AC requiere fijar los siguientes puntos:

- **Conjunto de entes.** Se necesita saber cuántos objetos elementales van a formar la población del sistema. En principio no hay restricción a su número, pudiendo ser desde unos pocos hasta una infinidad. En ocasiones es importante situarlos sobre una región geográfica, identificándose entonces los entes con sus respectivas coordenadas geográficas.
- **Vecindades.** Para cada elemento del sistema es necesario establecer su vecindad, esto es, aquellos otros elementos que serán considerados como sus vecinos. En caso de asociar objetos con coordenadas de un sistema de referencia, el criterio suele ser construir la vecindad de un elemento dado con todos aquellos otros elementos que se encuentran a menos de una cierta distancia o radio, de forma que los más alejados no ejerzan influencia directa sobre él.

  Sea

$$V_i(t) = \{x_i(t), x_{i+1}(t), x)i - 1(t), \ldots\} \tag{1}$$

	N	
O	C	E
	S	

(a) La vecindad de Von Neumann consiste en la celda central más los cuatro vecinos más próximos.

NO	N	NE
O	C	E
SO	S	SE

(b) La vecindad de Moore consiste en 9 celdas: la celda central más los vecinos más próximos y los siguientes.

Tabla 1: Vecindades de Von Neumann y Moore.

el conjunto de celdas vecinas de la celda i, además de la propia celda considerada.

En un AC las N celdas se encuentra en un reticulado de dimensión d. Cuando el retículo es de tamaño finito siempre consideraremos condiciones de contorno periódicas, es decir que la celda en la posición O tiene como vecina a la celda de la posición N y viceversa.

En dos dimensiones hay dos tipos fundamentales de vecindad: la de Von Neumann y la de Moore (ver Tablas 1(a) y 1(b) respectivamente).

- **Conjunto de estados.** En cada instante, cada elemento deberá encontrarse en un cierto estado. El caso más sencillo corresponde a los elementos biestables, los cuales se pueden encontrar en sólo uno de dos estados posibles, 0 y 1, por ejemplo. Pero también el estado puede venir representado por un vector de componentes reales o por una cadena de un lenguaje formal.

- **Regla de transición local.** La regla de transición define la dinámica del sistema. Dado un elemento y un instante determinados, la regla devuelve el siguiente estado del elemento. Para ello necesita como argumentos los estados actuales, tanto del elemento considerado como de aquellos que conforman su vecindad. Las reglas de transición pueden ser deterministas o probabilistas. Además, no todos los autómatas celulares requieren que sus elementos obedezcan siempre la misma regla, si bien para el enfoque del presente artículo se considera una regla única para todas las celdas o células del autómata.

$$R = f(V_i(t)) \tag{2}$$

La regla local de evolución $x_i(t+1)$ nos permite obtener el valor cuando conocemos el valor de las celdas en la vecindad en el instante anterior.

2.2. Estructura de un *Autómata Celular*

Un *Autómata Celular* es una herramienta computacional que hace parte de la *Inteligencia Artificial* basada en *modelos biológicos*, el cual está básicamente compuesto por una estructura estática de datos y un conjunto finito de reglas que son aplicadas a cada nodo o elemento de la estructura. El interés que ha despertado esta técnica radica en la sencillez y en la simplicidad que caracteriza la construcción de los modelos (son

susceptibles de fácil programación); además, en la particularidad de los patrones de comportamiento presentados por el Autómata en tiempo de ejecución, permitiendo una observación directa y detallada de los mismos a medida que evolucionan en el tiempo.

Basados en el planteamiento que presenta Muñoz [2] acerca de la estructura de un *Autómata Celular*, se definen como sus componentes básicos:

- Un plano bidimensional o un espacio n-dimensional dividido en un número de subespacios homogéneos, conocidos como celdas. A todo esto se le denomina *Teselación Homogénea.*
- Cada celda puede estar en uno de un *conjunto finito* o *numerable* Σ de estados.
- Una *Configuración* C, que consiste en asignarle un estado a cada celda del autómata.
- Una *Vecindad* definida para cada celda, que consiste en un conjunto contiguo de celdas, indicando sus posiciones relativas respecto a la celda misma.
- Una *Regla de Evolución*, que define cómo debe cada celda cambiar de estado, dependiendo del estado inmediatamente anterior de su vecindad.
- Un *Reloj Virtual de Cómputo* conectado a cada celda del autómata, el cual generará "*tics*" o pulsos simultáneos a todas las celdas indicando que debe aplicarse la regla de evolución y de esta forma cada celda cambiará de estado.

Desde el punto de vista de Toffoli y Margolus [3], se define un *Autómata Celular* sólo si se tiene que todas las celdas:

- Tienen el mismo Conjunto *sum* de *Estados posibles.*
- Tienen la misma forma de Vecindad.
- Tienen la misma *Regla de Evolución.*

Según se indicó anteriormente, es este enfoque particular el que se usará para la simulación específica presentada.

2.2.1. Consideraciones adicionales

Un *Autómata Celular* puede ser construido definiendo alguna especificación para cada uno de sus componentes, es decir, de alguna forma se definirá su *teselación*, los posibles *estados*, las *vecindades* y la *regla de evolución.* Se permite mayor flexibilidad en la construcción del autómata considerando:

- El *autómata* puede ser de $1, 2, 3, \ldots, n$ dimensiones.
- La *teselación* puede ser finita o infinita, con condiciones de frontera abiertas o periódicas.

- El *conjunto de estados* Σ no necesita tener ninguna estructura algebraica adicional. La *vecindad* puede ser simétrica o no y puede incluir o no a la propia celda. La *regla de evolución* es una tabla o unas reglas.

3. Paralelización

3.1. Descripción del problema

El caso de estudio del presente trabajo consiste en realizar una simulación de un incendio forestal. Para lo cual se debe definir un Autómata Celular con reglas específicas de propagación del fuego según la velocidad y dirección del viento, constituyendo estas reglas la definición del vecindario de cada uno de los entes presentes durante la simulación.

Se ha aplicado el modelo a conjuntos de árboles hasta de un millón (1000×1000) de elementos, con distribuciones aleatorias sobre el área considerada. El modelo incluye también la densidad del "bosque" o conjunto de árboles como un parámetro más en la concepción del modelo. En el sencillo modelo presentado, los árboles forman un conjunto homogéneo, desde el punto de vista de su contribución a la propagación del fuego. Es decir, se "queman" todos ellos de la misma manera, si bien, con el objeto de enriquecer el modelo, es posible incluir características propias de cada individuo, programando atributos particulares de una celda o célula o sub-conjunto de celdas del autómata celular e introduciendo nuevas reglas de evolución en el tiempo. Esta posibilidad de crecimiento modular en la complejidad y precisión del modelo constituye una ventaja adicional en la adopción del mismo.

El origen del fuego se sitúa al azar dentro del conjunto de árboles. El programa hace entonces que el autómata celular evolucione en el tiempo para el conjunto completo estudiado. Se alcanza el estado final cuando ya no existen focos de incendio, es decir, cuando se ha quemado la totalidad de los árboles o cuando el último árbol en combustión ha quedado aislado de sus compañeros aún en pie y las condiciones del viento en ese momento no permiten que el fuego se propague hacia árboles más alejados. Los resultados se observan cualitativamente mediante un gráfico que representa con colores distintos la presencia de un árbol, de un foco de incendio, de un vacío, o de un árbol muerto. El programa cuantifica también los resultados, indicando, por ejemplo, el porcentaje de árboles que continúan en pie luego de la extinción del incendio.

Por la posibilidad de aplicación práctica como una ayuda en la determinación de uno de los factores a la hora de planificar distancias convenientes para la plantación de árboles en conjuntos similares al simulado, resulta particularmente interesante la determinación de densidades críticas por encima de las cuales existe alta probabilidad de un incendio devastador (que acabe con el 90 % de los árboles, o más) bajo condiciones aleatorias de viento y origen del fuego.

Se muestran los resultados obtenidos respecto precisamente a estas densidades críticas.

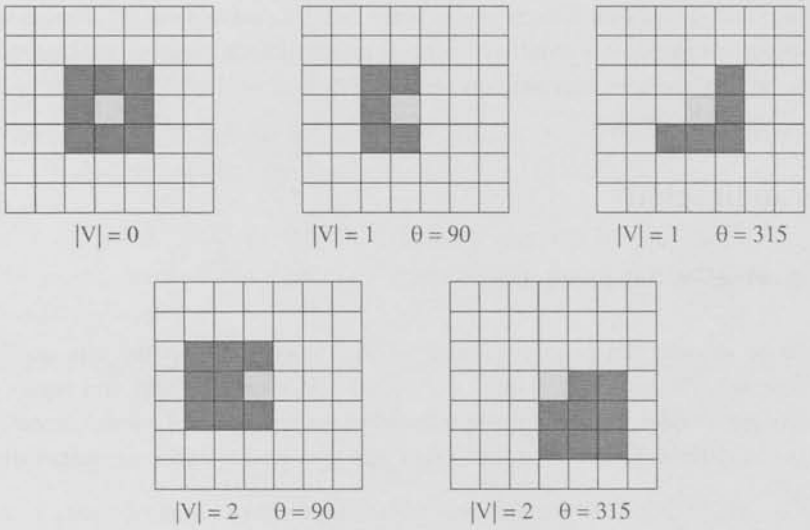


Figura 1: Vecinos (en negro) de una celda dada en función de las condiciones del viento.

3.2. Definición del modelo

El modelo de simulación para el presente trabajo consiste en un bosque representado por una matriz de DIM x DIM posiciones. El estado inicial está representado por la mencionada matriz. Cada elemento de la matriz puede estar vacío u ocupado por un árbol. Además, se debe conocer la posición específica del fuego en cada instante, así como las condiciones (velocidad, dirección) del viento.

La vecindad de cada uno de los elementos del bosque (célula) variará según las condiciones a las que se expone, en este caso el viento el cual posee una dirección y una magnitud de velocidad (intensidad). Para fines de simplicidad se restringe la velocidad a tres valores posibles, (0, 1 y 2) y las direcciones son ocho (Norte, Noreste, Este, Sudeste, Sur, Sudoeste, Oeste y Noroeste).

Cada célula puede tener hasta un máximo de ocho vecinos definidos según la Figura 1.

3.3. Paralelización

Con el fin de poder paralelizar el algoritmo de simulación, se debe en primer lugar estudiar la forma en la que debe estar dividida la matriz del bosque entre los procesadores que conforman la máquina paralela.

La distribución de la matriz es tal que cada uno de los procesadores almacena una porción equitativa de la misma.

Una vez distribuida la matriz debemos pensar en la vecindad de cada una de las células, y en especial de aquellas cuyos vecinos fueron trasladados por la distribución hacia otro procesador. La solución genérica que se plantea para este problema consiste

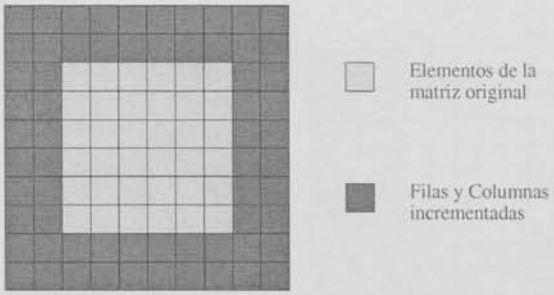


Figura 2: Confirguración de las matrices locales con incremento de cuatro filas y cuatro columnas.

en incrementar cada una de las matrices locales, en cuatro filas y cuatro columnas distribuidas como se muestra en la Figura 2.

El objetivo de incrementar las filas y columnas es el de recibir en ellas los elementos correspondientes a los vecinos que se encuentran almacenados en los procesadores vecinos.

El principio de funcionamiento del algoritmo paralelo se basa en que cada procesador debe informar a sus vecinos el estado de sus células que se encuentran en las filas y columnas externas, es decir en aquellos lugares que corresponden a la vecindad de las células de los procesadores vecinos. Además, deben enviar al procesador correspondiente un aviso de cambio de estado de las células que se encuentran en las secciones correspondientes a vecinos de otro procesador. Sólo se debe informar del cambio de estado de Árbol a Inicio de Fuego ya que los otros cambios de estado no requieren ser informados, cada procesador puede manejarlos independientemente.

La Figura 3 detalla la forma en que están distribuidas las células en los procesadores que conforman la máquina paralela. Por motivo de simplicidad y para facilitar el entendimiento se ilustra este mediante un ejemplo con cuatro procesadores.

4. Análisis teórico detallado

Puesto que tanto el procedimiento de llenado de la matriz como el procedimiento de exhibición gráfica de los resultados son ajenos a la ejecución del algoritmo de simulación propiamente dicho, para el presente análisis se considera la realización del autómata celular en sí. Esta realización se sintetiza a través del siguiente pseudo-código:

```
1: Iniciar_fuego;
2: Calcular_viento_inicial;
3: while fuego_posible do
4:   Calcular_nuevo_viento;
5:   for i ← 0 to DIM − 1 do
6:     for j ← 0 to DIM − 1 do
7:       Calcular_nuevo_estado;
```

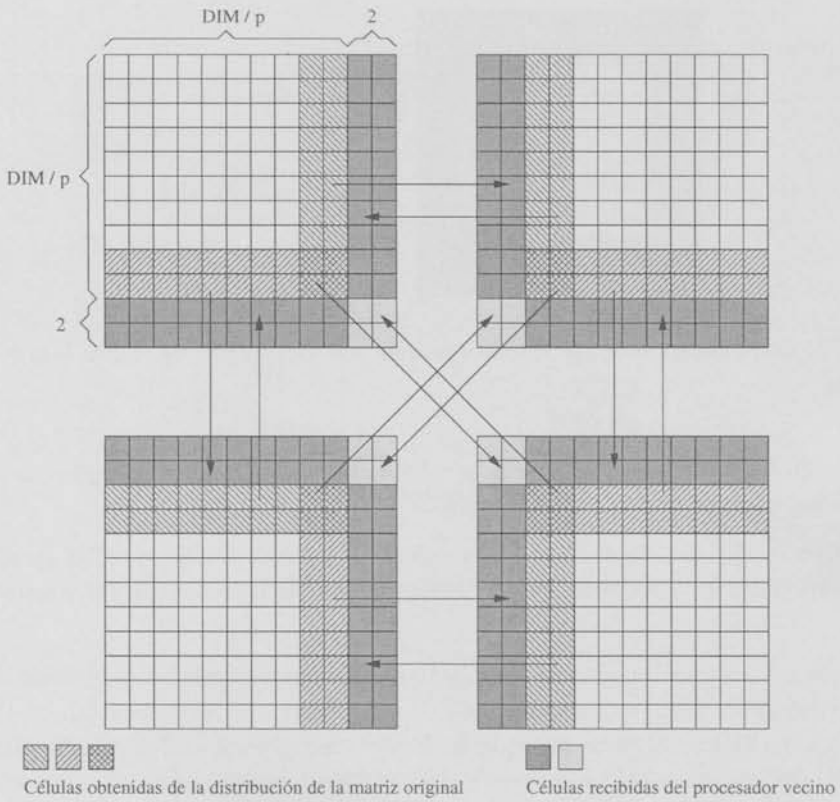


Figura 3: Las flechas indican que el contenido de las células achuradas debe copiarse en el destino de la flecha.

```
8:      end for
9:    end for
10:   for i ← 0 to DIM − 1 do
11:     for i ← 0 to DIM − 1 do
12:       Actualizar_matriz;
13:     end for
14:   end for
15: end while
```

Los procedimientos *Iniciar_fuego*, *Calcular_viento_inicial* y *Calcular_nuevo_viento* pueden considerarse ejecutables en una sola unidad de tiempo, ya que carecen de ciclos y, luego de evaluar una simple condición aleatoria, devuelven el resultado de una función matemática sencilla. El procedimiento *Calcular_nuevo_estado* en cambio, puede expresarse como:

```
1: if elemento[i][j] == arbol then
2:   Ver_vecinos;
3: end if
```

Donde *Ver_vecinos* está representado por:

```
1: for i_vecino ← i − 2 to i + 2 do
2:   for j_vecino ← j − 2 to j + 2 do
3:     if elemento[i_vecino][j_vecino] = vecino then
4:       Actualizarelemento[i][j]
5:     end if
6:   end for
7: end for
```

A pesar que potencialmente los dos ciclos anidados anteriores parecen involucrar 25 pasos (5 desde $i-2$ hasta $i+2$, por 5 desde $j-2$ hasta $j+2$), en realidad se los ha expresado así simplemente por facilidad, pero se evalúan, en el peor de los casos, tan sólo 8 veces, que corresponde a la mayor cantidad de vecinos que un elemento cualquiera puede tener, considerando todas las posibilidades de velocidad y dirección de viento.

Aun conociendo el detalle algorítmico que acaba de ser expuesto en función de operaciones que pueden considerarse de costo unitario en tiempo de computación, no es posible saber a priori cuantas veces (m) se ejecutará realmente el ciclo "Mientras", ni cuántos vecinos "árbol" encontrará realmente un árbol incendiado en particular. Se analizan por lo tanto a continuación dos situaciones extremas posibles:

(a) El origen del incendio casualmente ha surgido en un área totalmente despoblada de vecinos "árbol" y, por lo tanto, el incendio se extingue al final de la primera iteración.

(b) Es posible que el incendio surja más bien en un área tal que cada nueva iteración precise considerar el máximo posible (8) de vecinos incendiados. Cada una de estas situaciones se presentará con una cierta probabilidad (p_0 ó p_8, que no podrán confundirse con el número de procesadores p carente de subíndice).

Por lo tanto, en la realización de este algoritmo, el número de pasos elementales de computación, o tiempo secuencial (T_{seq}), entendiendo que cada paso elemental toma simplemente una unidad de tiempo, puede expresarse como:

$$\begin{aligned} T_{seq} &= 1+1+m(1+p_0\cdot 1\cdot n^2+1\cdot n^2) && (3)\\ &= 2+m(1+(p_0+1)n^2) && (4)\end{aligned}$$

en el mejor caso, y:

$$\begin{aligned} T_{seq} &= 1+1+m(1+p_8\cdot 8\cdot n^2+1\cdot n^2) && (5)\\ &= 2+m(1+(8p_8+1)n^2) && (6)\end{aligned}$$

en el peor caso.

El caso medio, que puede tomarse como genérico, estará dado entonces por un promedio ponderado de ambas expresiones que, por tener la misma forma, y siendo las probabilidades simplemente constantes comprendidas entre 0 y 1, inclusive, tendrá un comportamiento asintótico de crecimiento similar a:

$$T_{seq} = \Theta(mn^2) \qquad (7)$$

Al no estar disponible para la práctica investigativa otro modelo de realización del autómata celular, se tomará la expresión anterior en el cálculo del *speed-up*, la eficiencia, el costo, y la función de isoeficiencia.

Para el caso paralelo, se deberá considerar un tiempo de activación inicial de cada canal de comunicación (*start-up*, ts) para cada uno de los p procesadores y cada una de las m iteraciones del ciclo exterior "Mientras". Se deberá considerar asimismo un tiempo de transmisión unitario por palabra o elemento a ser transmitido (t_w) proporcional al cubo (tres ciclos "Para" anidados) de la dimensión dividida entre el número de procesadores en cada dimensión ($\sqrt{p}$). El tiempo requerido para ejecutar el algoritmo en paralelo, T_{par}, crece entonces asintóticamente como:

$$T_{par} = \Theta\left(m\left(\frac{n}{\sqrt{p}}\right)^2 + mpt_s + m\left(\frac{n}{\sqrt{p}}\right)^3 t_w\right) \quad (8)$$

$$= \Theta\left(\frac{mn^2}{p} + mpt_s + \frac{mn^3}{p\sqrt{p}}t_w\right) \quad (9)$$

Por lo tanto, el *speed-up* crecerá asintóticamente como:

$$S = \Theta\left(\frac{n^2 p\sqrt{p}}{n^2\sqrt{p} + p^2\sqrt{p}t_s + n^3 t_w}\right) \quad (10)$$

De ello se deduce que la eficiencia crecerá como:

$$E = \Theta\left(\frac{n^2\sqrt{p}}{n^2\sqrt{p} + p^2\sqrt{p}t_s + n^3 t_w}\right) \quad (11)$$

$$= \Theta\left(\frac{n^2\sqrt{p}}{n^2\sqrt{p} + p^2\sqrt{p} + n^3}\right) \quad (12)$$

teniendo en cuenta que los tiempos de *start-up* y transmisión de una unidad de información son constantes.

Se aprecia que inclusive con el valor:

$$p = \Theta(n^2) \quad (13)$$

se podría expresar la eficiencia como:

$$E = \Theta\left(\frac{n^2\sqrt{n^2}}{n^2\sqrt{n^2}}\right) = \Theta(1) = \Theta(K) \quad (14)$$

siempre y cuando los tiempos de comunicación sean pequeños. Se ha mencionado la conveniencia de realizar el algoritmo sobre una arquitectura paralela, con el objetivo de reducir los tiempos de cálculo. Así, normalmente intervendrá más de una CPU o unidad inteligente en la solución del problema. Debido a esto, las ecuaciones mostradas incluyen el impacto que los tiempos de comunicación entre CPUs tienen sobre la eficiencia del sistema. Una hipótesis, cuyo cumplimiento se debe cuidar en la práctica, es precisamente la existencia de tiempos de comunicación tan pequeños como permita el *hardware* (equipo) particular que esté en uso.

	D.I									
S	**20**	**30**	**40**	**41**	**42**	**43**	**44**	**45**	**55**	**65**
1111	19.93	29.81	19.58	12.46	5.85	6.97	4.24	9.84	0.97	0.08
2222	20.00	30.00	40.00	36.85	36.56	35.48	7.09	6.21	0.82	0.06
3333	20.00	30.00	34.17	35.01	33.73	30.79	3.96	6.06	0.45	0.19
4444	19.99	29.67	20.11	16.00	19.60	8.19	9.28	8.27	0.26	0.12
1234	19.67	29.51	37.99	38.70	39.74	10.11	8.54	8.99	0.70	0.04

Tabla 2: Densidades porcentuales finales en función de la semilla (S) y la densidad porcentual inicial (D.I.).

En principio, sería posible asignar un procesador a cada uno de los elementos de la matriz de dimensión n, antes de perder la capacidad de lograr eficiencias siempre constantes (definición de isoeficiencia). Los autómatas celulares se caracterizan precisamente por la posibilidad de trabajar con un procesador por elemento o celda.

En cuanto al costo C, por definición éste es igual al tiempo de ejecución en paralelo multiplicado por el número de procesadores necesarios para lograr dicho tiempo. Es decir, crecerá asintóticamente como:

$$C = \Theta\left(\frac{mn^3}{p} + mpt_s + \frac{mn^3}{p\sqrt{p}}t_w\right) \tag{15}$$

Se aprecia que aun para autómatas celulares pequeños, y aun para tiempos de comunicación despreciables, el costo puede crecer de manera prohibitiva (proporcional al cubo del número de procesadores usados) si se pretende en todo momento aplicar la función de isoeficiencia.

5. Resultados obtenidos

Se ha ejecutado el programa de simulación de incendio forestal una gran cantidad de veces, bajo condiciones siempre variables, en busca de obtener resultados reveladores del comportamiento general que puede esperarse del sistema simulado.

Las Tablas 2, 3, 4 y 5 sintetizan el comportamiento observado, haciendo énfasis en distintos aspectos de interés, como ser: la estimación inicial del valor probable de densidad crítica, la validación de este valor sobre matrices grandes y en condiciones de viento variable, la influencia del número entero empleado para activar el generador de números aleatorios, es decir, la influencia del número que se denomina "semilla" del generador aleatorio y, por último, los tiempos obtenidos para uno y cuatro procesadores.

La Tabla 2 se ha obtenido para una pequeña matriz (100×100) de prueba, con viento de velocidad cero. La zona sombreada corresponde a valores de densidad porcentual inicial mayor o igual al 44 %, y se aprecia que independientemente del valor de la semilla, el bosque siempre queda para estos casos con una densidad total de árboles sobrevivientes menores al 10 %. Por lo tanto, se sospecha que la densidad crítica r_0 debe corresponder al 44 %.

S	43	44
1111	7.25	5.01
2222	18.95	4.47

Tabla 3: Resultados obtenidos para confirmar el valor de densidad crítica.

	D.I.				D.I.				D.I.
	$n = 100$				$n = 500$				$n = 1000$
S	70	80	90	95	70	80	90	95	80
1111	23.67	15.21	3.17	55.33	52.63	66.77	33.45	82.43	46.22
2222	13.67	17.24	0.01	16.32	20.03	66.40	60.34	63.82	-
3333	21.71	42.28	0.88	01.34	38.77	47.47	69.05	50.66	-
4321	33.61	09.28	1.60	80.80	53.76	36.98	36.14	48.37	-
4444	34.85	32.35	4.81	00.38	37.33	15.28	04.00	16.03	-

Tabla 4: Densidades finales obtenidas con viento variable para distintas densidades iniciales, semillas, y tamaños de matriz.

Para la Tabla 3, se ha repetido el experimento con la semilla más "dañina" (1111) y la semilla más "benéfica" (2222) observadas en la prueba inicial, pero empleando esta vez una matriz de 1000×1000 elementos. Los resultados son consistentes con el valor crítico antes obtenido.

Al hacer que el viento sea de velocidad y dirección variables, no ha sido posible establecer una densidad crítica estable: los resultados dependen fuertemente no sólo de la densidad porcentual inicial, sino también de la semilla, y sólo aleatoriamente se ha podido elegir semillas especialmente "dañinas" que han conseguido incendios devastadores incluso para matrices muy grandes. Los resultados correspondientes se sintetizan en la Tabla 4.

Se aprecia que para elaborar la tabla anterior, se han explorado incluso valores de densidad porcentual inicial mucho mayores al límite crítico anteriormente definido. A pesar de ello, los resultados continúan inestables: el 90 % de densidad inicial parece representar un nuevo valor crítico, pero de pronto la densidad porcentual final crece nuevamente con ciertas semillas, aun para densidades del 95 %.

Los tiempos mostrados en la tabla anterior han sido obtenidos con el utilitario *time* propio del sistema operativo, para condiciones de viento variable, y el mismo valor de la semilla: 1111.

6. Discusión de resultados

La Tabla 2 muestra los resultados de simulaciones con viento de velocidad nula, pero con distintas densidades iniciales y distintas condiciones (elegidas al azar) para el inicio del fuego. Las otras tablas, en cambio (Tablas 3, 4 y 5 inclusive), muestran resultados para condiciones de velocidad variable del viento. Según se aprecia, cuando la velocidad del viento es nula (Tabla 2) se tienen las mayores posibilidades de un

	D.I.	100	200	300	400	500	1000
$p = 1$	50	0.20 s	4.48 s	25.92 s	55.50 s	1m 39 s	14 m 37 s
	80	0.54 s	4.60 s	17.10 s	17.50 s	1 m 5 s	14 m 8 s
$p = 4$	50	0.08 s	1.80 s	12.35 s	31.71 s	1 m 24 s	19.44 s
	80	0.21 s	1.82 s	8.26 s	13.82 s	55.1 s	18 m 36 s

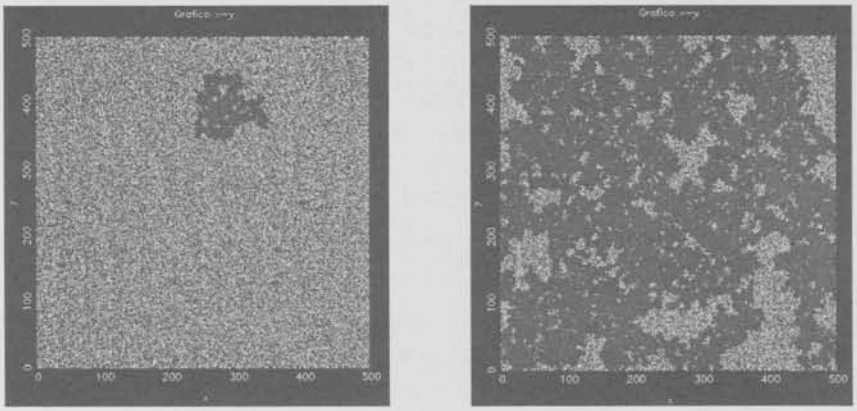
Tabla 5: Tiempos medidos en función de la dimensión de la matriz ($n = DIM$) y del número de procesadores empleados.

incendio devastador, independientemente del valor exacto de la semilla, ya que la posible quema masiva del bosque es función básicamente de la densidad inicial, sin el factor viento que, aleatoriamente, podría oponerse a un mayor avance del incendio. Distintos focos de incendio cambian la geometría del problema al determinar qué procesador inicia el fuego (o en qué posición dentro de la matriz se inicia éste, en caso de emplear un solo procesador), y cómo irá cambiando el viento a medida que evolucionan las iteraciones por toda la matriz pero, con viento nulo, no cambian el valor de la densidad crítica que se mantiene coherentemente en 44 %. Se ve que el valor de densidad crítica tampoco depende del tamaño del bosque: se exponen resultados similares para matrices comprendidas entre 10.000 y 1.000.000 de elementos, inclusive.

Cuando se deja que la velocidad y dirección del viento cambien al azar, sin embargo, no es posible ya determinar una densidad crítica: la densidad de árboles sobrevivientes depende fuertemente de la pareja densidad_inicial/semilla, y no solamente de la densidad inicial. Las simulaciones realizadas ilustran este hecho siempre para el rango comprendido entre 10.000 y 1.000.000 de celdas, y aun para 1.000.000 de árboles (matriz de 1.000×1.000 elementos, totalmente poblada).

Los resultados obtenidos exponen claramente la variabilidad en el ciclo "Mientras" externo: no es posible determinar a priori cuántas veces lo repetirá el algoritmo programado. Este hecho depende más bien de cuán devastador resulta el incendio (o la combinación *densidad_inicial/semilla*). Se encuentra incluso un caso en el cual la diferencia de ejecución para matrices de tamaños tan dispares como 300×300 y 400×400 es mínima (alrededor de 1 segundo). Esto se debe a que en la matriz más pequeña se quemó un porcentaje mucho mayor de árboles, haciendo que en la práctica el ciclo se repita casi tantas veces como en la matriz de 400×400 de población mucho mayor, pero también muchísimo más afortunada en cuanto a árboles sobrevivientes luego de las mismas condiciones iniciales para el generador de números aleatorios.

El alejamiento entre el valor ideal de *speed-up* previsto por la teoría, (donde p representa el número de procesadores empleados para resolver el problema) y los valores observados (incluyendo un caso extremo de *speed-up* menor a uno) se explica por la distribución poco eficiente de mensajes sobre la topología en estrella, a través de un "hub" no procesador en el centro de dicha estrella. Se aprecia que a pesar que el algoritmo desarrollado sólo intercambia mensajes cuando es indispensable hacerlo (bordes de matrices asignadas a procesadores vecinos, y solamente cuando un árbol incendiado contagia efectivamente a un árbol vecino explorado por otro procesador), al crecer la matriz los términos en t_s y t_w que figuran dentro de la expresión del *speed-up*, no son



(a) $\rho = 41\,\%$; el incendio queda circunscrito a un área pequeña.

(b) $\rho = 45\,\%$; el incendio es devastador.

Figura 4: Distribución aleatoria de 2.5×10^5 individuos. Simulación en ausencia de viento.

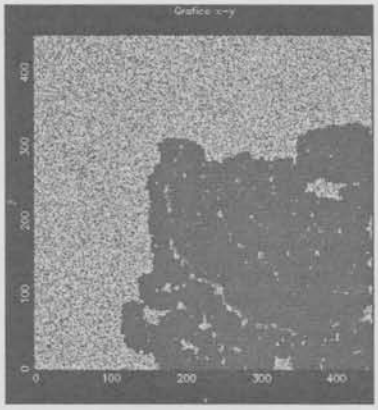
ya despreciables e influyen más bien negativamente sobre este parámetro.

7. Conclusiones

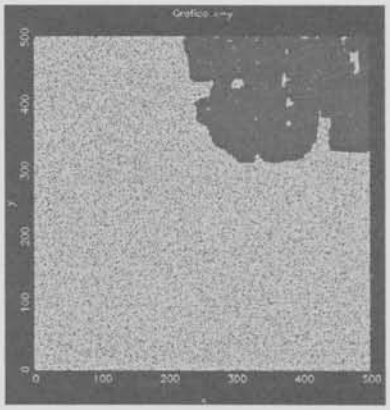
El modelo de simulación desarrollado muestra claramente que, en ausencia de viento, existe una densidad crítica (44 %) en la población de árboles dentro de un bosque, por encima de la cual, en caso de incendio, no llegará a sobrevivir ni el diez por ciento de los árboles originalmente existentes. Esto sucede con independencia del tamaño del bosque, de la distribución inicial de árboles dentro del bosque (geometría del bosque), e independientemente también de la posición geográfica del foco original de incendio.

Sin embargo, cuando el viento sopla de manera aleatoria (velocidad y dirección cambiantes), aunque esos cambios no sean frecuentes ni bruscos, no es posible determinar una densidad crítica constante: ésta depende fuertemente del tamaño y geometría del bosque y de manera muy especial depende también de la posición geográfica del foco inicial de fuego dentro del bosque, y de la manera en la cual va evolucionando el viento. Bosques muy densamente poblados pueden salvar sin intervención externa un altísimo porcentaje de su población inicial, si el azar define el surgimiento de fuego en una zona tal que la evolución posterior del viento empuje el fuego hacia áreas poco pobladas. Esta observación coincide con lo que sugiere la intuición.

La duración experimentalmente observada en las distintas instancias de ejecución de la simulación permite obtener asimismo otra conclusión intuitivamente correcta: si el incendio destruye una cantidad reducida de árboles, termina en un lapso igualmente reducido. Por el contrario, si el incendio alcanza grandes proporciones y termina quemando una población muy grande, demora un tiempo proporcionalmente mayor.



(a) El incendio puede ser devastador para ciertas densidades ($\rho = 50\,\%$) y distribuciones.



(b) Aún para densidades muy altas ($\rho = 85\,\%$), el incendio puede quedar contenido.

Figura 5: Distribución aleatoria de 2.5×10^5 individuos. Simulación con viento variable.

Estas observaciones y conclusiones coincidentes con lo que parece indicar la experiencia, sugieren la validez del modelo y su fácil expansión para incluir factores complejos en las ecuaciones y reglas de producción que lo definen. Por ejemplo, la consideración de especies distintas para poblar el bosque, desde el punto de vista de programación, simplemente implica incrementar el número de estados permitidos para cada célula en el autómata, así como el número de reglas de transición entre estados. Árboles con maderas especialmente resistentes al fuego, por ejemplo, pueden simularse a través de células para las cuales la transición entre los estados de ignición inicial y de totalmente quemado, dura más de un intervalo de tiempo en el proceso de simulación. De esta manera, el individuo permanece vivo más tiempo, pero su poder de contagiar fuego a los vecinos también perdura por un lapso mayor. Por el contrario, árboles con maderas muy combustibles o resinosas se pueden representar a través de células que evolucionan rápidamente entre los estados de ignición inicial y de quemado total: en este caso, es el individuo el que perece rápidamente, en tanto que su poder de contagiar fuego es menor ya que perdura sólo por un lapso breve.

Referencias

[1] A.K. Dewdney. Juegos de ordenador. *Investigación y Ciencia*, (154), 1989.

[2] J.D. Muñoz. Autómatas celulares y física digital. En *Memorias del 1er. Congreso Colombiano de Neurocomputación*, p 28, 1996. Academia Colombiana de Ciencias Exactas, Físicas y Naturales.

[3] T. Toffoli y N. Margolus. *Cellular Automata Machines: A New Environment for Modelling.* The M.I.T. Press., Cambridge, MA, 1987.