

Evolución de autómatas celulares utilizando algoritmos genéticos

Juan Ignacio Vázquez
ivazquez@eside.deusto.es
Facultad de Ingeniería - ESIDE
Universidad de Deusto
48007 Bilbao, Vizcaya
España

Javier Oliver
oliver@eside.deusto.es
Facultad de Ingeniería - ESIDE
Universidad de Deusto
48007 Bilbao, Vizcaya
España

Resumen

Las capacidades computacionales de los autómatas celulares son elevadas, sin embargo, su naturaleza paralela dificulta su programación mediante métodos convencionales. Entre las alternativas para esta programación se encuentra la aplicación de algoritmos genéticos con el fin de obtener el autómata celular óptimo que lleva a cabo una tarea concreta. En este trabajo se analizan los autómatas celulares, los algoritmos genéticos y el estado actual de la investigación en la aplicación de ambos modelos para la resolución de problemas.

1 Autómatas celulares

Los *autómatas celulares (AC)* constituyen un modelo matemático de interacción extremadamente simple, discreto en el espacio y en el tiempo, que emula el comportamiento de muchos sistemas naturales [Chau et al., 1999]. Además, otros sistemas dinámicos de extrema complejidad pueden ser modelados mediante autómatas celulares como el caso de corrientes circulatorias de tráfico o sistemas que exhiben autoorganización.

Muchos sistemas biológicos naturales, como el cerebro, los sistemas inmunológicos o las colonias de insectos están formados por la asociación de innumerables elementos relativamente homogéneos y simples que operan en paralelo, sin ningún tipo de control central y con un mecanismo de comunicación limitado [Crutchfield et al., 1998] que, sin embargo, son capaces de exhibir una conducta extremadamente compleja que parece emerger de las interrelaciones entre los mismos.

Los autómatas celulares comparten muchas de estas cualidades con los sistemas biológicos anteriormente descritos: un gran número de componentes homogéneos, extendidos en el espacio, sin control central y con un mecanismo limitado de comunicación entre los elementos.

Además, desde el punto de vista de la computación, los autómatas celulares pueden ser considerados como un tipo especial de máquinas de Turing, sin memoria interna [Chau et al., 1999], siendo algunos modelos capaces de llevar a cabo cualquier tipo de cálculo, es decir, computación universal [Berlekamp et al., 1982].

Un autómata celular está formado por elementos simples o células, que cambian de estado como consecuencia de un conjunto de reglas preestablecidas, a intervalos de tiempo discretos. A cada ciclo de tiempo, el estado de una célula depende del estado de las células que la rodean y cambia, por lo tanto, de la manera prescrita mediante la tabla de reglas que rige el comportamiento del autómata, conformando, así, su diagrama de transición de estados.

Un autómata celular bidimensional estaría constituido por un tablero o una rejilla, siendo cada una de las casillas una célula que puede adoptar uno de entre k estados, dependiendo de la configuración del entorno en el que se encuentra, es decir, de su estado actual y del estado de cierto número v de casillas o células vecinas, lo que se denomina vecindad del autómata celular.

Un autómata unidimensional, una secuencia lineal de casillas, con un radio de vecindad $r = 1$ estaría compuesto de células que dependen en cada momento del estado de sus dos células vecinas. En el mismo autómata unidimensional pero con un radio $r = 2$ el estado de cada célula dependería del estado de sus cuatro células vecinas, dos en un sentido y dos en otro.

En todo autómata celular existe una tabla de reglas que define el comportamiento del mismo a lo largo del tiempo, dependiendo de las dimensiones, el radio de vecindad y el número de estados.

Para un autómata celular unidimensional, con $r = 1$ y número de estados $k = 2$, un ejemplo de tabla de reglas posible sería el indicado en la Figura 1:

000	001	010	011	100	101	110	111
0	1	1	1	0	0	1	0

Figura 1. Ejemplo de tabla de reglas de un autómata unidimensional con $k = 2$ y $r = 1$.

La célula central cambia de estado dependiendo de las situadas a su derecha e izquierda. Analizando esta tabla de reglas es posible derivar que una célula cambia de estado bajo las siguientes condiciones:

- Cuando una célula se encuentra inactiva, se activa en el siguiente ciclo de tiempo si la de su derecha está activa y la de su izquierda no lo está

- b) Cuando una célula se encuentra activa, se desactiva en el siguiente ciclo de tiempo si ambas células adyacentes se encuentran activas.
- c) En cualquier otra condición una célula conserva su estado.

Se muestra en la Figura 2 las dos configuraciones que provocan un cambio en el estado de una célula:

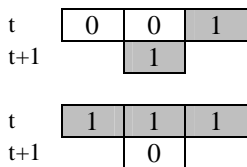


Figura 2. Las dos reglas que provocan un cambio en el estado de una célula para la tabla de reglas anterior.

Se representan a continuación las transiciones de un autómata celular en dos pasos de tiempo, aplicando esta tabla de reglas.

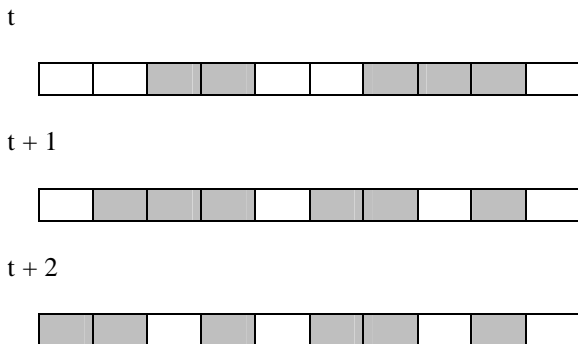


Figura 3. Aplicación de la tabla de reglas del autómata a una configuración inicial dada con 5 células activadas.

La primera de las células no tiene ninguna a su izquierda sobre la que interrogar el estado, al igual que ocurre simétricamente con la última por la de recha. Por ello, un autómata celular se simula con condiciones de límite periódicas es decir, el estado de cada célula limítrofe depende de las que se encuentran en el extremo opuesto, actuando como si de un anillo o toroide se tratase.

De esta forma es posible observar la evolución de un autómata celular a lo largo del tiempo, partiendo de una configuración inicial, y percibir cómo el sistema se transforma a cada ciclo de tiempo, adoptando distintas apariencias.

Un autómata celular se define a través de la tabla de reglas que describen su comportamiento. Un autómata unidimensional, con dos estados y radio de vecindad $r = 1$, se define a través de 8 reglas, como las mostradas en la Figura 1. Cada regla queda definida por el estado final de la célula afectada por la regla, en el ejemplo, 0110010. Sería posible establecer otras reglas distintas que definen las transiciones de estado de las células, como por ejemplo, 10000001:

000	001	010	011	100	101	110	111
1	0	0	0	0	0	0	1

Figura 4. Otro ejemplo de una tabla de reglas de un autómata unidimensional con $k = 2$ y $r = 1$.

De este modo, no resulta difícil determinar que el número de autómatas celulares unidimensionales $(k, r) = (2, 1)$ es 256, correspondientes a las 256 posibles combinaciones que se pueden realizar con las reglas, representadas en un alfabeto de dos símbolos, 0 y 1, en una cadena de 8 posiciones, $2^8 = 256$. Estos 256 autómatas celulares unidimensionales se han usado ampliamente en experimentaciones para demostrar las propiedades de este modelo matemático, y se han dado en denominar autómatas celulares elementales o ACE (*elementary cellular automata - ECA*).

Uno de los tipos de autómatas celulares más conocidos es *Life* o *Juego de la Vida*, diseñado por el matemático John Conway en 1960. Se trata de un autómata celular bidimensional en el que cada célula puede adoptar uno de dos estados, a saber: vivo o muerto. En cada momento el estado de una célula viene determinado por el estado de las 8 células adyacentes, puesto que el radio $r = 1$. Una célula viva muere si existen menos de dos células vecinas vivas o más de tres. Una célula vuelve a la vida si hay exactamente tres células vecinas vivas.

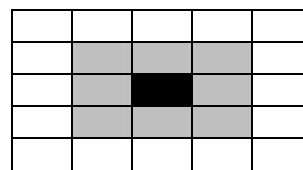


Figura 5. Un autómata celular bidimensional con $r = 1$. El estado de una célula se encuentra determinado por las inmediatamente adyacentes, que se representan sombreadas.

De la misma manera es posible diseñar autómatas celulares tridimensionales, en los que el estado de cada célula, con $r = 1$, estaría entonces determinado por las 26 células adyacentes.

La complejidad de un autómata celular, y por tanto la complejidad de su comportamiento, depende del número de reglas que lo gobiernen. Ese número de reglas N depende de tres factores distintos: el número D de dimensiones del autómata, el radio r de vecindad y el número k de estados que puede adoptar cada célula, según la siguiente fórmula:

Y el número NA posible de autómatas distintos sería:

$$NA \cong k^{(2r+1)k^{(2r+1)D}}$$

Para $(D, k, r) = (1, 2, 1)$, es decir, para el autómata unidimensional de las figuras anteriores, el número de reglas es 8, tal y como se había calculado previamente, siendo $2^8 = 256$ el número de autómatas distintos que pertenecen a este grupo: los ACE.

El número de reglas y por tanto la complejidad del sistema se puede elevar aumentando los tres factores:

- a) El número D de dimensiones del autómata, con lo que aumenta el número de células de las que depende cada célula del modelo, incrementándose de este modo el número de combinaciones posibles.
- b) El número k de estados de cada célula del autómata, con lo que se incrementa a su vez el número de reglas posibles, al aumentar el alfabeto de símbolos que definen las combinaciones.
- c) El radio r de vecindad, con lo que aumenta la dependencia de una célula con otras.

Para los autómatas bidimensionales binarios de radio 1, $(D, k, r) = (2, 2, 1)$, como *Life*, el número de reglas necesarias es 512, y, por lo tanto, existen 2^{512} tablas de reglas distintas, autómatas distintos.

Observando el comportamiento de los autómatas celulares es posible percibir que las configuraciones que adoptan dinámicamente varían entre un amplio espectro de conductas: algunos se estabilizan y permanecen en una configuración fija, otros exhiben un comportamiento periódico, y, por último, otros adoptan configuraciones aparentemente imprevisibles.

Wolfram [Wolfram, 1984a] propuso una clasificación mediante la cual se pudiera identificar la conducta de cualquier AC, como perteneciente a una de las siguientes categorías:

Clase 1 o comportamiento fijo: todas las configuraciones iniciales convergen tras un período transitorio a la misma configuración final. Por ejemplo, un autómata $(D, k, r) = (1, 2, 1)$ en el que la aplicación de sus reglas fuese capaz de transformar una configuración inicial cualquiera en una configuración con todos unos.

Clase 2 o comportamiento periódico: todas las configuraciones iniciales convergen tras un período transitorio a la misma configuración o a algún ciclo periódico de configuraciones, que depende de la inicial. Por ejemplo, un autómata que a cada paso de tiempo mueva las celdas con estados activos hacia la derecha, en un ciclo periódico.

Clase 3 o comportamiento caótico: algunas configuraciones iniciales convergen hacia una conducta caótica, es decir, impredecible en el espacio y en el tiempo.

Clase 4 o comportamiento complejo: algunas configuraciones iniciales convergen en estructuras

complejas, que se conservan en el tiempo. Esta clase de autómatas no se encuentra definida de forma exhaustiva, desgraciadamente, pero aquí se situarían todos aquellos autómatas capaces de llevar a cabo algún tipo de computación compleja.

De forma experimental Wolfram clasificó todas las tablas de reglas de autómatas pertenecientes a distintas familias de pares (k, r) en las cuatro clases expuestas, con los siguientes resultados porcentuales [Wolfram, 1984b]:

Clase	$k=2$	$k=2$	$k=2$	$k=3$
	$r=1$	$r=2$	$r=3$	$r=1$
1	0.50	0.25	0.09	0.12
2	0.25	0.16	0.11	0.19
3	0.25	0.53	0.73	0.60
4	0	0.06	0.06	0.07

Como es posible observar, la mitad de los autómatas sencillos de la familia $(k, r) = (2, 1)$ tiende a estabilizar sus configuraciones (clase 1) tras un número de pasos determinado. A medida que la tabla de reglas crece, al crecer el factor k o r , la mayor parte de los autómatas de la familia tiende a mostrar una conducta caótica (clase 3).

Langton [Langton, 1991] definió un parámetro λ , que representa la fracción de estados distintos a uno cualquiera tomado como referencia, en la salida de la tabla de reglas. En un autómata binario, el estado 0 podría ser tomado como referencia, y λ sería la fracción de 1 presentes en la salida de la tabla de reglas. A medida que λ progresa de 0 a $[1 - 1/k]$ se pasa de tener la tabla de reglas más homogénea a la más heterogénea, y lo que resulta más interesante, la conducta de los AC varía de la misma forma a medida que lo hace λ , pasando por:

$$\text{Clase 1} \Rightarrow \text{Clase 2} \Rightarrow \text{Clase 4} \Rightarrow \text{Clase 3}$$

O lo que es lo mismo:

$$\text{Fijo} \Rightarrow \text{Periódico} \Rightarrow \text{Complejo} \Rightarrow \text{Caótico}$$

Se han determinado transiciones de fase en valores críticos de λ , que provocan que los autómatas celulares de una determinada familia, alteren su conducta y pasen de una clase a otra.

Además de los autómatas uniformes, en los que todas las células comparten las mismas reglas, existen otro tipo de autómatas celulares aún más complejos, los no-uniformes, en los que distintas células pueden estar regidas por diferentes tablas de reglas [Sipper, 1994].

De este modo, se pretende demostrar la extrema complejidad de comportamiento que un autómata celular es capaz de exhibir, incluso los autómatas celulares relativamente sencillos, pueden ser capaces de emular una máquina de Turing capaz de llevar a cabo computación universal, como ha sido demostrado [Berlekamp et al., 1982; Lindgren et al., 1990; Smith, 1971].

Los autómatas celulares son modelos matemáticos masivamente paralelos. Todas las células del autómata evalúan su estado en paralelo, dependiendo de las células de la vecindad, para cambiar la configuración del autómata a cada paso de tiempo. Es por su propia naturaleza por lo que no existe un control central que dirija al autómata de una determinada manera, sino que es precisamente el paralelismo lo que multiplica la complejidad del modelo.

Sin embargo, el limitado mecanismo de comunicación entre las células, dependiente del radio de vecindad, consigue crear en ciertos ejemplos la sensación de autoorganización en la dinámica de las células, lo que ha servido para estudiar comportamientos complejos observados en el mundo real, a través de su modelización por medio de autómatas celulares.

En *Life*, se ha podido simular una complejidad fascinante a partir de las sencillas reglas de las que dispone [Creutz, 1996]. Durante una simulación pueden emerger *planeadores*, o combinaciones de células que se desplazan lentamente por el tablero manteniendo la misma configuración a intervalos periódicos, como si fuera un pequeño organismo. Gracias a los planeadores, es posible transmitir información a través de grandes distancias, e incluso simular el comportamiento de un computador [Berlekamp et al., 1982]. De la misma forma se han podido crear *disparadores*, que emiten planeadores a intervalos regulares.

Con modelos de autómatas celulares ligeramente más complejos, se han podido crear configuraciones capaces de autoreproducirse, es decir, configuraciones que después de un número determinado de pasos de tiempo son capaces de crear en espacio una réplica de si mismas, y de este modo generar toda una población [Levy, 1993; Morita, 1997].

Los autómatas celulares también pueden mostrar configuraciones fractales que exhiben autosimilitud, tal y como lo hacen algunos sistemas naturales.

2 Algoritmos genéticos

Los algoritmos genéticos son métodos de búsqueda en el espacio de soluciones de un problema [Crutchfield et al., 1998] basados en el mecanismo de la selección natural o lucha por la supervivencia [Darwin, 1985].

Mediante la utilización de algoritmos genéticos las soluciones al problema se codifican como una secuencia de símbolos, como un programa genético, generando una población inicial de estas soluciones. Generación tras generación, las soluciones más aptas, o que más se acercan al resultado final, sobreviven y se reproducen, haciendo cruzar entre sí sus cadenas de símbolos, introduciendo nuevos individuos en la población. Tras un número determinado de generaciones, se pretende obtener una población de soluciones con al menos un individuo capaz de llegar al resultado deseado para el problema.

En el mundo natural, se dice que un proceso está dirigido por la evolución cuando se satisfacen las siguientes cuatro condiciones [Koza, 1992]:

1. Una entidad debe tener la capacidad de reproducirse.
2. Debe existir una población de tales entidades.
3. Debe existir algún tipo de variedad entre las entidades de dicha población.
4. Debe existir alguna diferencia en la capacidad de sobrevivir en el entorno asociada a dicha variedad.

En la naturaleza, la variedad se manifiesta en las diferencias entre cromosomas en una población determinada. Esta variación en el *genotipo* se hace patente en la estructura y comportamiento del organismo, es decir, en la expresión externa del genotipo llamada *fenotipo*, lo que influye en la capacidad de la entidad para adaptarse al medio.

Los individuos que mejor se adaptan tienen más probabilidades de sobrevivir y reproducirse, y, por lo tanto, de legar a sus descendientes esas capacidades o variaciones en su genotipo que le han proporcionado un mayor grado de adaptación. A lo largo de sucesivas generaciones, la estructura y comportamiento de los individuos de la población hace que éstos se adapten perfectamente al entorno.

De este modo cualquier problema de adaptación puede observarse bajo el prisma genético. John Holland fue el primero en definir un marco para la experimentación con procesos evolutivos en sistemas artificiales [Holland, 15]. Tal marco es lo que se ha dado en denominar el *algoritmo genético*.

El algoritmo genético es un algoritmo matemático, altamente paralelo, que transforma un conjunto de objetos (población) representados matemáticamente cada uno de ellos mediante una cadena de símbolos (genotipo) asociada a una determinada medida de adaptación al medio, en otra población distinta (siguiente generación) mediante la aplicación del principio de supervivencia del más fuerte en términos de adaptación.

Se supone una población inicial de individuos, representados por una cadena de símbolos:

Individuo	Cadena
I ₁	011
I ₂	001
I ₃	110
I ₄	010

Dicha cadena podría interpretarse como la codificación de un programa en base a una serie de instrucciones, representadas por los símbolos.

Esta población inicial de individuos se sitúa en el entorno apropiado en el cuál se desea obtener el individuo con el

mejor comportamiento posible. Tras experimentar se obtienen, gracias a una función capaz de medir el grado de adaptación de cada individuo, las siguientes medidas:

Individuo	Cadena	Adaptación
I ₁	011	0.25
I ₂	001	0.08
I ₃	110	0.50
I ₄	010	0.17

Para la siguiente generación se escogen los individuos que mejor grado de adaptación han obtenido. En este punto entran en juego los que se han dado en llamar operadores genéticos, que son básicamente tres:

Reproducción: se escogen para reproducción aquellos individuos que han demostrado un mejor grado de adaptación al medio, generando una descendencia proporcional con dicho valor.

Mutación: un símbolo de la cadena puede alterarse para introducir una variación en el genotipo de un individuo y, por lo tanto, en la población. Por ejemplo, el individuo I₄ podría sufrir una mutación que alterase su cadena genética en la posición 2, creando un individuo en la población con configuración 000, que previamente no existía. La mutación crea nuevos caminos de búsqueda en el espacio de soluciones.

Cruzamiento o recombinación sexual: permite que nuevos individuos sean creados a partir de los ya existentes, de tal modo que se abren nuevos caminos por el espacio de soluciones del problema, en base a caminos ya existentes. El mecanismo consiste en cruzar las cadenas de símbolos que definen dos individuos en un punto determinado. Por ejemplo, efectuando la operación de cruzamiento sobre los individuos I₁ e I₃, en la posición 2, se divide la cadena de I₁ en 01|1, y la de I₃ en 11|0. Cruzándolas se obtendrían dos individuos: uno con cadena 010 y otro con 111. Para el cruzamiento se escoge a los individuos de forma proporcional a su grado de adaptación al medio.

El cruzamiento es uno de los pilares del algoritmo genético que permiten explicar matemáticamente la validez del mismo como método de búsqueda inteligente. Los mejores individuos se escogen con probabilidad proporcional a su grado de adaptación y se cruzan entre sí. De este modo, los patrones de su cadena genética que les proporcionan dicha adaptación son recombinados para crear nuevos individuos que poseen las capacidades de los progenitores.

Si en el ejemplo se supone que la presencia de un 1 en cualquier posición de la cadena genética aporta mayor adaptación a un individuo, gracias al cruzamiento es posible obtener el individuo óptimo, con cadena 111, a partir I₁ e I₃.

El cruzamiento permite que los *esquemas* genéticos que aportan adaptación a los individuos avancen generación

tras generación, y se recombinen para componer entidades mejor adaptadas.

Estos esquemas están formados por las subcadenas obtenidas en la división previa al proceso de cruzamiento, como 01*, **1, 11* ó **0, a partir de las cuales se lleva a cabo la recombinación. Estos esquemas pueden contener las cualidades que hacen a los individuos ajustarse al medio, de hecho, en el dominio biológico se denominan genes.

La posibilidad de rotura de un esquema depende de la distancia entre los símbolos específicos más alejados, y de la longitud de la cadena de los individuos. Por ejemplo, un esquema como **11111** tiene más posibilidades de romperse en una recombinación que el esquema *****111**, pues en el último solo se produciría ruptura si los puntos de cruzamiento se sitúan entre las posiciones 6 y 7, ó 7 y 8. Esquemas como 111*****11 se romperían siempre que se efectuase un cruzamiento, fuese cual fuese el punto del mismo.

Los esquemas forman los bloques constituyentes de las cadenas que representan a los individuos.

Es posible considerar cada cadena de símbolos, que define a un individuo, como una *idea*, codificada mediante ese alfabeto en cuestión, para resolver un determinado problema. Las subcadenas contenidas dentro de la cadena que representa la idea, podrían ser identificadas como *nociones* importantes o relevantes para la tarea. Una población de individuos no es simplemente un conjunto aleatorio de ideas, sino que contiene una multitud de nociones que se combinan para formar las ideas que resuelven el problema [Goldberg, 16].

De este modo, los algoritmos genéticos exploran este conjunto de conocimiento a) reproduciendo las ideas que han demostrado un grado superior de adaptación, y b) cruzando las nociones pertenecientes a las ideas que demuestran mayor grado de adaptación. De este modo se consigue especular con nuevas ideas, construidas a partir de los bloques constituyentes o nociones de intentos pasados [Goldberg, 16], en un proceso cuyo resultado es la innovación de conocimiento.

El rendimiento de los algoritmos genéticos en procesos de optimización y de búsqueda de soluciones no es fruto del azar, sino de la combinación de nociones para formar ideas. Esas nociones se encuentran codificadas en forma de esquemas que, mediante operaciones como el cruzamiento, se recombinan formando nuevas ideas. En lugar de probar cualquier combinación posible de símbolos, aquellas soluciones parciales de intentos anteriores se conservan y mejoran para finalmente obtener el individuo óptimo [Goldberg, 16].

La población inicial de individuos se genera de forma aleatoria, pero a partir de este punto, el resto del proceso es una búsqueda probabilística por el espacio de soluciones del problema. Sin embargo los algoritmos genéticos pueden sufrir el problema de convergencia

prematura [Mühlenbein, 17], si los individuos de la población inicial no contiene n suficientes nociones para obtener la idea óptima. En tal caso se obtendrían óptimos locales, pero no la mejor solución al problema.

Algunos algoritmos intentar resolver este problema introduciendo una mayor probabilidad de mutaciones o violando la metáfora biológica, permitiendo sólo descendientes que varíen de forma considerable con respecto a sus progenitores.

Una solución es la utilización de algoritmos genéticos paralelos [Mühlenbein, 17], que introducen diversificación de una forma más natural. En lugar de disponer de una única población, se crean varias subpoblaciones aisladas, que evolucionan por separado. Cada cierto tiempo las subpoblaciones permiten que sus individuos emigren a otras subpoblaciones, introduciendo, de este modo, variedad en las cadenas genéticas de los demás conjuntos de individuos.

De esta forma se consigue un intercambio de nociones sobre cómo resolver un problema, entre poblaciones de individuos que desean alcanzar dicho objetivo.

3 Aplicación de algoritmos genéticos a la evolución de autómatas celulares

Anteriormente, se ha mostrado una clasificación de autómatas celulares basada en el comportamiento de los mismos, dicha clasificación podía recorrerse a través del parámetro λ de Langton:

Fijo \Rightarrow Periódico \Rightarrow Complejo \Rightarrow Caótico

A través de observaciones se ha podido determinar cómo la región de transición entre el orden y el caos da lugar a los comportamientos más complejos, de hecho la vida surge al borde del caos, y es la selección natural la que alcanza y sostiene este estado [Kauffman, 18].

De este modo se demuestra la relación directa entre complejidad y selección natural, siendo esta última la que consigue mantener dicha complejidad de los organismos en los valores adecuados. Si la complejidad se reduce, el organismo no es capaz de desarrollar funciones avanzadas, si el organismo exhibe un comportamiento caótico tampoco es capaz de llevar a cabo dichas funciones y, en cualquier caso, no sobreviviría a lo largo de las generaciones, legando su información genética.

Al igual que en los sistemas naturales, existe una relación directa entre la complejidad de los organismos artificiales y la selección artificial basada en los algoritmos genéticos, siendo éstos capaces de dirigir la evolución a conveniencia, para generar individuos aptos para la resolución de un problema, como se ha descrito con anterioridad.

Los autómatas celulares se han descrito como modelos matemáticos masivamente paralelos, extremadamente

simples en su concepción, pero capaces de llevar a cabo en algunos casos conductas sumamente complejas y no predecibles a priori, a menos que se efectúe una simulación, debido al enorme número de interacciones entre las células.

Es, por esta circunstancia, prácticamente imposible diseñar a mano un conjunto de reglas para que un autómata celular realice una tarea compleja, es decir, programar la tabla de reglas del autómata. La mayoría de los descubrimientos relativos a los comportamientos de configuraciones determinadas de autómatas celulares, regidos por una tabla de reglas determinada, es fruto casual de la observación de la simulación, como en el caso de los planeadores, pero en ningún caso concebida a priori por el diseñador.

La pregunta que surge a continuación es: ¿sería posible hacer evolucionar una población de autómatas celulares para obtener uno o varios individuos capaces de llevar a cabo una tarea compleja?. Es decir, ¿sería posible aplicar algoritmos genéticos para obtener una población de autómatas celulares que lleven a cabo una determinada función?

Los AC están definidos por su tabla de reglas. Esta tabla de reglas constituiría la cadena genética, substrato del trabajo de los algoritmos genéticos, que, a su vez, conservarían aquellas partes de la cadena, o nociones, útiles para llevar a cabo la tarea deseada, de generación en generación mediante la aplicación de los operadores genéticos como se ha explicado anteriormente, obteniendo nuevos individuos en los que se sintetizan las nociones de generaciones pasadas que aportan una mayor adaptación a los organismos, en este caso, a los autómatas celulares.

Puesto que la programación de la tabla de reglas de un autómata celular que lleve a cabo una función determinada es una tarea demasiado ardua para el ser humano, los algoritmos genéticos constituyen una herramienta ideal que puede ser empleada con éxito para obtener dicha tabla, y por lo tanto, el autómata celular que lleve a cabo una función compleja.

Un sistema es un computador universal si dado un programa inicial, su evolución en el tiempo puede implementar cualquier algoritmo finito [Wolfram, 19]. Un autómata celular dotado de computación universal podría ejecutar en paralelo todos los programas posibles.

La idea del autómata celular como un sistema de computación se resume de la siguiente forma: a partir de una configuración inicial, que representa los datos de entrada, se pueda efectuar cierto proceso, determinado por la tabla de reglas del AC, para obtener una configuración final, que representa el resultado [Wolfram, 19].

Packard ha sido uno de los primeros investigadores que ha utilizado algoritmos genéticos para generar autómatas celulares. Examinó la frecuencia de las reglas generadas como función del parámetro λ de Langton, e interpretó

los resultados como evidencias de dos hipótesis [Mitchell, 20]:

1. Los autómatas celulares capaces de llevar a cabo cálculos complejos, se sitúan cerca de valores λ críticos que determinan transiciones de fase entre regímenes ordenados y caóticos.
2. La evolución tiende a seleccionar autómatas celulares con valores de λ cercanos a dichos valores críticos.

Generalmente en la experimentación de la aplicación de algoritmos genéticos a los autómatas celulares se han utilizado AC con $(D, k, r) = (1, 2, 3)$ y el cálculo a implementar ha sido el *problema de clasificación de la mayoría*.

Este problema es un medio para estudiar cómo pueden llevarse a cabo cálculos complejos sobre grandes áreas mediante reglas que operan por interacciones a distancias relativamente pequeñas. La configuración inicial de un conjunto de células en un autómata biestado unidimensional conforma los datos de entrada para el cálculo. El objetivo es obtener una tabla de reglas que defina un autómata celular capaz de clasificar correctamente la configuración inicial, haciéndola tender, tras un número finito de pasos, a todo 1 si había una mayoría de 1 en la configuración inicial, o a todo 0, si había una mayoría de 0.

Siendo r_0 , el porcentaje de 1 en la configuración inicial,

Si $r_0 > 0.50$, la configuración final debe ser todo 1.

Si $r_0 < 0.50$, la configuración final debe ser todo 0.

Si $r_0 = 0.50$, el resultado es indeterminado.

Generalmente el tamaño de la rejilla unidimensional, número de células a clasificar, es de 149 para estos experimentos.

Es complejo diseñar a mano la tabla de reglas de un autómata celular capaz de llevar a cabo esta tarea. Gacs, Kurdyumov y Levin definieron un autómata celular, llamado GKL, basado en las siguientes condiciones:

1. Si $s_i(t) = 0$ entonces $s_i(t+1) = \text{mayoría} [s_i(t), s_{i-1}(t), s_{i+1}(t)]$
2. Si $s_i(t) = 1$ entonces $s_i(t+1) = \text{mayoría} [s_i(t), s_{i+1}(t), s_{i+3}(t)]$

Dónde $s_i(t)$ representa el estado de la célula i en el instante de tiempo t .

Es decir, para cada vecindad de siete células adyacentes, si el estado de la célula central es 0, su nuevo estado se decide por el voto mayoritario de ella misma, su vecina izquierda y la situada dos posiciones más allá también a la izquierda. Si el estado de la célula central es 1, el nuevo estado se decide de forma similar pero tomando las células vecinas por la derecha.

El autómata celular GKL clasifica correctamente el 81.6% de las configuraciones iniciales. Lawrence Davis y Rajharsi Das crearon dos autómatas celulares capaces de llevar a cabo la misma tarea con un 81.8% y un 82.178% de efectividad respectivamente [Andre, 21].

Packard intentó aplicar el algoritmo genético al problema de la clasificación de la mayoría para obtener un autómata celular que mejorase los resultados del GKL. En cada generación:

1. Se calcula el grado de adaptación de cada autómata celular.
2. Se clasifica a la población en base a su grado de adaptación.
3. Una parte de los autómatas celulares con peor grado de adaptación se elimina del sistema.
4. Éstos últimos son sustituidos por otros nuevos obtenidos aplicando las operaciones de cruzamiento y mutación a partir de los más aptos.

Para evitar la convergencia prematura se introdujo un sistema que reforzase la diversidad decrementando artificialmente el grado de adaptación de aquellos individuos con tablas de reglas demasiado parecidas a otras ya existentes.

Packard llegó a unos resultados que le permitieron aseverar que los autómatas celulares con mejor comportamiento eran aquellos con valores λ cercanos a las transiciones de fase de ordenado a caótico [Mitchell, 20]:

“Autómatas celulares con λ cercano a las zonas de transición al caos tienen la capacidad de comunicar información selectivamente con estructuras complejas en el espacio-tiempo y así permitir la computación.”

Sin embargo, Mitchell et al. [Mitchell, 20] reprodujeron los mismos experimentos sin obtener los valores de λ indicados por Packard, concluyendo que los resultados originales pudieron deberse a mecanismos adicionales utilizados en el algoritmo genético de Packard, y no documentados. De esta forma, determinaron que la relación entre λ y la capacidad computacional de un autómata celular no es directa, aunque sí lo es la relación entre λ y el comportamiento dinámico del autómata. Es decir, no existe relación entre los valores de λ y la capacidad de un autómata de llevar a cabo cualquier tipo de computación (computación universal), aunque sí hay relación directa entre λ y la capacidad de un autómata para llevar a cabo una tarea compleja determinada, situándose los más aptos cerca de valores concretos de λ que, sin embargo, no tienen porque estar próximos a zonas de transición al caos.

La técnica utilizada por Mitchell et al. para aplicar los algoritmos genéticos a la generación de autómatas celulares es la siguiente [Mitchell, 20]:

1. Se escoge al azar una población inicial de M cromosomas, que representan los autómatas celulares mediante su tabla de reglas.
2. El grado de adaptación de un autómata celular de la población se calcula escogiendo al azar I configuraciones iniciales (CI) sobre la rejilla e iterando el autómata celular sobre cada CI hasta llegar a un punto fijo o hasta consumir un máximo de T_{max} ciclos de tiempo.
3. Se determina si la configuración final alcanzada es correcta para la tarea deseada, en este caso, si es todo 0 cuando $\rho_0 < 0.50$ o todo 1 cuando $\rho_0 > 0.50$. El grado de adaptación es la fracción de CI correctamente clasificadas. Las configuraciones parcialmente correctas no son tomadas en cuenta.
4. Para cada generación:
 - a) se crea un nuevo conjunto de I CI
 - b) se calcula el grado de adaptación de cada autómata
 - c) se ordenan según dicho grado de adaptación
 - d) se escoge un número E de autómatas celulares situados entre la élite, con alto grado de adaptación y se copian a la siguiente generación
 - e) los elementos $M - E$ restantes se forman por cruzamiento entre los más aptos, y aplicando la operación de mutación con una determinada probabilidad

El resultado obtenido fue la generación de unos autómatas celulares que desarrollaron distintas estrategias [Wolfram, 1984a] para llevar a cabo la tarea, consiguiendo en el mejor de los casos un éxito del 77.5% de clasificaciones correctas que contrasta con el GKL con un 81.6%.

Posteriormente Andre et al. [Andre, 21] utilizando programación genética con ADF (*automatically defined functions*) han conseguido obtener un autómata celular capaz de clasificar correctamente el 82.326% de los casos. Paredis y Juillé y Pollack han experimentado con técnicas de aprendizaje por coevolución, obteniendo un autómata celular capaz de clasificar correctamente el 86.3% de los casos, aunque su efectividad decae rápidamente a medida que el número de celdas a clasificar aumenta [Crutchfield et al., 1998].

En algunos casos, mediante la aplicación de la tabla de reglas, que define a un autómata celular, a una configuración inicial, ésta tiende a lo largo del tiempo hacia otra u otras que actúan como *atractores* [Wolfram, 1984a]. Es decir, la configuración inicial se altera gradualmente aproximándose a otra configuración final que actúa como un atractor: una configuración que puede ser perturbada dentro de cierto rango, volviendo a su estado anterior al cabo de cierto tiempo. Por ejemplo, podría decirse que el autómata celular GKL tiene dos atractores: todo 0 y todo 1. Si en una de estas configuraciones se varía el estado de alguna célula, la ejecución del autómata hace que se vuelva rápidamente a la configuración del atractor.

La mayoría de los CA tienen un comportamiento irreversible: dado una configuración concreta es imposible predecir la configuración o configuraciones anteriores. La tendencia al desorden y el incremento de la entropía son características universales de los sistemas intrínsecamente reversibles e identificadas en la segunda ley de la termodinámica [Wolfram, 1984a]. La posesión de los CA de cierto grado de irreversibilidad, actúa contra la tendencia al desorden y por ello son capaces de exhibir cierto nivel de autoorganización, directamente relacionado con la aparición de atractores, que definen esa estructura organizativa.

\uparrow Irreversibilidad \Rightarrow \uparrow Autoorganización

La existencia de dos atractores en GKL le proporciona esa característica para autoorganizarse hacia todo 0 o todo 1.

4 Conclusiones

La metodología de programación convencional es, naturalmente, de poca utilidad para un sistema de computación basado en un autómata celular. El desarrollo de una nueva metodología es difícil, pero un desafío importante, y los algoritmos genéticos pueden ser una herramienta fundamental en la búsqueda de la misma, puesto que el principal cuello de botella de la aplicación de autómatas celulares a la computación en paralelo es su programación [Mitchell, 20].

Mitchell et al. argumentan que el enfoque de la utilización de autómatas celulares para la computación debe estar dirigido hacia la búsqueda de AC que lleven a cabo tareas concretas, y no un autómata celular universal, como *Life*, que, aunque capaz de ejecutar cualquier tarea, lo hace de forma ineficiente [Mitchell, 20].

Por todo ello, la atención actual se dirige hacia el desarrollo de autómatas celulares que sean capaces de llevar a cabo tareas complejas como el procesamiento de imágenes [Wolfram, 19] o el análisis de turbulencias [Mitchell, 20], siendo los algoritmos genéticos un método ideal para la obtención de dichos autómatas celulares, a pesar de que su aplicación en este sentido aún está dando los primeros frutos.

5 Referencias

- [Chau et al., 1999] Chau, H.F. et al. "Classifying Rational Densities Using Two One-Dimensional Cellular Automata". 1999. <http://xxx.unizar.es/ps/adap-org/9709003>. 25 de Junio de 1999.
- [Crutchfield et al., 1998] Crutchfield, J. P., Mitchell, M., Das, R. "The Evolutionary Design of Collective Computation in Cellular Automata", 1998. <http://www.santafe.edu/projects/evca/evabstracts.html>. 25 de Junio de 1999.
- [Berlekamp et al., 1982] Berlekamp, E., Conway J.H., y Guy R. *Winning Ways For Your Mathematical Plays, volumen 2*. Academic Press, 1982.

- [Wolfram, 1984a] Wolfram, S. “Cellular Automata”. 1984.
<http://www.stephenwolfram.com/articles/indices/ca.html>, 25 de Junio de 1999.
- [Wolfram, 1984b] Wolfram, S. “Universality and Complexity in Cellular Automata”, 1984.
<http://www.stephenwolfram.com/articles/indices/ca.html>, 25 de Junio de 1999.
- [Langton, 1991] Langton, C.G. “Computation at the edge of chaos: Phase transitions and emergent computation”. *Emergent Computation*. MIT Press, 1991.
- [Sipper, 1994] Sipper, M. “Non-Uniform Cellular Automata: Evolution in Rule Space and Formation of Complex Structures”. *Artificial Life IV*. MIT Press, 1994.
- [Lindgren et al., 1990] Lindgren, K. y Nordahl M.G. *Universal computation in a simple one-dimensional cellular automaton*. *Complex systems*, 4:299-318,1990
- [Smith, 1971] Smith, A.R. *Simple computation-universal cellular spaces*. *Journal of the ACM*, 18:339, 1971.
- [Creutz, 1996] Creutz, M. “Cellular automata and self-organized criticality”, 1996.
<http://penguin.phy.bnl.gov/www/papers/hep-lat-9611017.ps.Z>. 25 de Junio de 1999.
- [Levy, 1993] Levy, S. *Artificial Life*. Penguin Books, 1993.
- [Morita, 1997] Morita, K. e Imai, K. “A Simple Self-Reproducing Cellular Automaton with Shape-Encoding Mechanism”. *Artificial life V*. MIT Press, 1997.
- [Darwin, 1985] Darwin. R. *El origen de las especies*. Editorial EDAF, 1985.
- [Koza, 1992] Koza, J. *Genetic Programming: On the programming of computers by means of natural selection*. MIT Press, 1992.
- [Holland, 15] Holland, J. *Adaptation in natural and artificial systems*. MIT Press, 1992.
- [Goldberg, 16] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [Mühlenbein, 17] Mühlenbein, H. “Evolution in Time and Space – The Parallel Genetic Algorithm”. *Foundations of Genetic Algorithms*. Morgan Kaufman, 1991.
- [Kauffman, 18] Kauffman, S. *The Origins of Order*. Oxford University Press, 1993.
- [Wolfram, 19] Wolfram, S. “Cellular Automaton on Supercomputing”, 1988.
<http://www.stephenwolfram.com/articles/indices/ca.html>, 25 de Junio de 1999.
- [Mitchell, 20] Mitchell, M. et al. *Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations*, 1993.
<http://www.santafe.edu/projects/evca/evabstracts.html>. 25 de Junio de 1999.
- [Andre, 21] Andre, D. et al. “Evolution in Intricate Long-Distance Communication Signals in Cellular Automata Using Genetic Programming”. *Artificial life V*. MIT Press, 1997.