

Computación Evolutiva

Manuel de la Herrán Gascón [1] [2]

Bilbao (España), 1998.

Introducción

La Inteligencia Artificial (IA) no sólo consiste en idear algoritmos y estructuras de datos para solucionar problemas. También trata acerca de la inteligencia humana, y por extensión, sobre la vida. Dentro de la IA, la Vida Artificial ofrece algunos mecanismos de resolución de problemas muy eficientes y originales. Además, toma muy en serio sus aspectos más filosóficos.

¿Qué tienen estas simulaciones de fantástico? Podemos sentirnos una especie de dios, observando y modificando a nuestro antojo un mundo poblado por seres virtuales [*Ray*]. También es interesante jugar con la idea de que nosotros mismos somos las ‘hormigas’, viviendo bajo los designios de ‘El Programador’ [*Adams, 1983*] [*Watson, 1987*] [*Gisbert, 1981*]. Pero no se trata sólo de eso. Se trata de que la Vida Artificial ofrece una nueva perspectiva sobre los problemas que afectan a cualquier grupo, como por ejemplo, la humanidad.

Vida y Evolución

La Vida Artificial consiste en observar la vida natural e imitarla en un ordenador [*Prata, 1993*] [*Herrán, Agosto 1997*]. La Computación Evolutiva [*Heitkötter, Beasley, January 2000*] interpreta la naturaleza como una inmensa máquina de resolver problemas y trata de encontrar el origen de dicha potencialidad para utilizarla en nuestros programas [*Herrán, Septiembre 1997*] [*Herrán, Octubre 1997*]. Efectivamente, en la naturaleza todos los seres vivos se enfrentan a problemas que deben resolver con éxito, como conseguir más luz del sol, o cazar una mosca. Un programador con espíritu práctico no envidia la capacidad de la naturaleza para resolver problemas: la imita [*Holland, Septiembre 1992*].

«En la naturaleza todos los seres vivos se enfrentan a problemas que deben resolver con éxito.»

El origen de esta capacidad está en la evolución, producida por la selección natural, que favorece la perpetuación de los individuos más adaptados a su entorno. Esto es lo que tenemos que simular. La principal diferencia conceptual entre la selección natural (o que se produce sin intervención del hombre), y la selección artificial que nosotros establecemos en nuestras granjas o en nuestros programas, es que la selección natural no es propiamente una selección [*Barnett, 1962*]. Nosotros podemos seleccionar para la reproducción los seres que más nos interesan, ya sean las vacas más lecheras o los agentes software que mejor resuelven un problema. En cambio, en la naturaleza no existe -en principio- una inteligencia exterior que determine la dirección de la evolución. Y sin embargo la evolución sí se produce. Si efectivamente, no existe algo o alguien que controle la evolución de la vida en nuestro planeta, entonces nosotros mismos seríamos el ejemplo de un principio básico y universal: que la vida, la inteligencia, la consciencia y quién sabe qué otros tipos de complejidad en el futuro, son sucesos inherentes, inevitables y espontáneos de

nuestro universo. A esto también se le ha llamado Darwinismo Universal [Dawkins, 1994], y supone que toda vida, en cualquier lugar, habría evolucionado por medios darwinianos.

El origen de la vida y el programa Tierra

El programa Tierra [Ray] es un ejemplo de cómo agentes software pueden evolucionar sin que sea necesaria una selección dirigida por una entidad externa. Tanto en este programa como en otros, existe una serie de componentes software de algún tipo capaces de reproducirse y sufrir mutaciones. En un Algoritmo Genético, los agentes deben resolver un problema particular, pero en Tierra los agentes hacen poco más que reproducirse. El espacio de memoria limitado produce una selección natural, ya que sólo las mejores entidades podrán dejar descendencia en él. La existencia de pequeñas mutaciones aleatorias basta para generar agentes con características muy complejas, capaces de invadir o cooperar con otros agentes. Después de estudiar una de estas simulaciones, parece lógico suponer que en nuestro planeta haya podido suceder algo parecido. Está bastante extendida la idea de que las primeras entidades replicantes surgieron al azar, a partir de la combinación de elementos, y que la selección hizo el resto. Sin embargo Thomas Ray decidió que su programa comenzara con un primer agente capaz de copiarse a sí mismo, sin pretender que esta autocopiamiento se produjera espontáneamente, como hizo Steen Rasmussen. Se plantea la cuestión de si la probabilidad de aparición de los componentes básicos de la vida es demasiado baja para un solo planeta. Fred Hoyle [Hoyle, 1983] sugiere la existencia de un bombardeo de material genético del exterior -ya sea con o sin entidad consciente detrás de ello-, lo que daría un mayor margen, al haber podido surgir este 'primer replicante' en cualquier otro mundo. Además, esto solucionaría algunos otros problemas, como los aparentes 'saltos de complejidad' evolutivos. Es sorprendente la aparición de órganos complejos como los ojos, que difícilmente han podido surgir de una evolución gradual si no proporcionan ninguna ventaja al individuo hasta que no se encuentran formados por completo. El lamarkismo, es decir, la transmisión genética de los caracteres adquiridos, está resucitando [Barnett, 1962]; y Máximo [Sandín, 1995] ofrece otra interesante explicación a todas estas cuestiones mediante infecciones de tipo vírico capaces de afectar rápidamente a gran parte de la población.

«La selección natural no es propiamente una selección.»

En mi opinión, la cuestión más apasionante de las difícilmente explicadas por el darwinismo es la de la aparición de sentimientos de placer y dolor en los seres vivos. Nosotros podemos asignar a cada agente una variable con un número llamada placer o dolor, pero no es necesario que la entidad tenga realmente esas sensaciones para que se comporte como si las tuviera. Tal vez podamos construir algún día robots que se comporten como seres humanos, pero ¿podremos hacer que sientan? No importa ahora la respuesta a esta pregunta. La cuestión es: aunque pudiésemos, ¿por qué hacerlo? ¿Por qué lo ha hecho la naturaleza con nosotros? Una cosa es la vida artificial como imitación de los procesos propios de la vida, y otra muy distinta es la recreación de su esencia.

Desde el punto de vista reduccionista, podemos pensar que la 'recreación de su esencia' no es posible precisamente porque dicha esencia no existe, y la vida no es más que sus procesos. Tal como comentaba cierto día mi amigo Vicent Castellar «¿Qué diferencia existe entre sumar uno más uno y simular que se suma uno más uno?» Ciertamente, es difícil de ver la diferencia. Si el universo fuese susceptible de ser descompuesto en unidades mínimas de espacio y tiempo, todo el universo podría considerarse como un gran sistema formal. ¿Qué diferencia habría entre el universo real y otro universo copia del primero? ¿Qué diferencia habría entre materia e información? ¿No sería lo mismo tener una unidad mínima de materia en cierta posición, que tener 'algo' que se comportase como si fuera una unidad mínima de materia, en la

misma posición? Lo mismo podemos aplicar a un cuerpo humano, cerebro incluido. ¿Qué diferencia habría entre dos cuerpos así? (Además de la obvia: que ambas copias a pesar de ser idénticas, o bien no ocupan el mismo lugar del espacio, o bien no se encuentran en el mismo tiempo) ¿Serían dos personas o una?

Sin embargo, sí existe una gran diferencia entre «me duele el estómago» y «simular que me duele el estómago». Aquí aparece un componente cuya simulación no puede considerarse equivalente. Hay algo que no se puede simular: el sentir. ¿Tiene el sentimiento un origen evolutivo? Un árbol no tiene capacidad de sentir sensaciones (vamos a suponer esto), y en cambio una rana sí. El placer, el dolor, etc. parecen muy buenos mecanismos de supervivencia, pero realmente no hacen falta si el ser vivo es capaz de comportarse 'como si' los tuviera. La rana busca la comida, como el árbol la luz; ambos realizan las acciones correctas gracias a siglos de evolución, aunque la rana sí siente hambre y el árbol no. ¿Por qué ha ocurrido esto con los animales? Los sentimientos probablemente sean mecanismos adicionales y potentes creados por y para los seres más complejos, que se deben enfrentar con problemas muy distintos, por ejemplo, por el hecho de ser móviles. Pero ¿realmente le resulta más fácil y económico a la naturaleza crear seres que realmente sienten, que seres que actúan como si sintieran?

En fin, las teorías sobre la vida y la evolución son difícilmente demostrables, y el debate es intenso. Podría parecer que estas cuestiones sólo atañen a los biólogos y filósofos, pero no es así. Recordemos que se trata de crear inteligencia en nuestros ordenadores imitando la forma en que lo hizo la naturaleza en nuestro planeta. Sin embargo, no podemos hacerlo sin más. Debemos captar al máximo la esencia de todo el proceso para evitar que nuestro ordenador tarde también millones de años.

¿Evolución egoísta, programas egoístas?

La teoría de la evolución puede llevarse directamente a su extremo lógico en forma de darwinismo social suponiendo que la lucha por la vida y la supervivencia del más apto son la única vía de todo progreso humano. También se puede tomar la postura de Pedro [*Kropotkin, 1970*] y defender el apoyo mutuo como otro factor de evolución. En los humanos se encuentran comportamientos muy distintos. Desde cierto punto de vista, se puede argumentar que cuando un individuo coopera, es porque, egoístamente, ha evaluado que esto le ayuda a conseguir sus objetivos, y por tanto sólo existe el egoísmo. Esta aparente paradoja no es más que una confusión de términos. Para tratar la cuestión adecuadamente es preciso no tener en cuenta motivaciones, sino sólo actos. Un comportamiento altruista será aquel que contribuya al bienestar ajeno a expensas del propio; uno egoísta sería exactamente lo contrario [*Dawkins, 1994*]. Voy a llamar cooperación a las diferentes formas de acuerdo que se encuentran oscilando en el límite entre altruismo y egoísmo.

«En la evolución se combinan egoísmo y altruismo.»

Antes de continuar quiero aclarar que no estoy discutiendo la ética de estos comportamientos, sino analizando por qué algunos o una combinación de ellos son seleccionados en la evolución. Para saber cuáles son los seleccionados, basta con mirar a nuestro alrededor, ya que nosotros y el resto de los seres vivos somos producto de ella. Aún así no debemos olvidar que el proceso evolutivo es algo dinámico. Hoy se selecciona una cosa, mañana otra. El hombre seguirá evolucionando, no sabemos hacia donde. Tal vez sea lo suficientemente diferente como para dar un giro radical al proceso, tal vez no. En mi opinión, y en contra de la de muchos otros, el autor que mejor trata este tema es Richard [*Dawkins, 1994*] [*Dawkins, Enero 1996*]. Para él somos máquinas de supervivencia construidas por nuestros genes para su

propia perpetuación. Venimos de los genes egoístas, moléculas autorreplicantes que en cierto momento ‘decidieron’ que la creación de una máquina como nosotros, con una capacidad de razonamiento flexible, era lo más adecuado para sus fines. Hay idealistas que rechazan impulsivamente la idea de una naturaleza basada en genes egoístas y hay quienes sólo ven en ella egoísmo y destrucción. Si bien es cierto que demasiadas veces la naturaleza no es madre, sino más bien cruel suegra, fuera de nuestra óptica, la naturaleza es simplemente indiferente. En cualquier caso, Dawkins habla de genes egoístas, no de individuos egoístas. Y lo que es más importante: con la teoría de Dawkins, la cooperación e incluso el altruismo (reales) entre individuos pueden ser explicados por el ‘egoísmo’ (metafórico) de los genes. Vamos a verlo. En todas las células de nuestro cuerpo existe la misma información genética: una larga cadena de ADN idéntica en el interior de cada célula. Existen muchas definiciones de gen. Para Dawkins, «el gen egoísta no es sólo una porción física de ADN: es todas las réplicas de una porción de ADN, distribuidas por el mundo». Es decir, el mismo alelo o valor en las mismas posiciones dentro de la cadena de ADN, es el mismo gen egoísta, ya se encuentre en uno o en distintos individuos. Un gen egoísta de los que habla Dawkins no sólo está simultáneamente en todas las células de nuestro cuerpo. Está simultáneamente en varios individuos. Esta abstracción de Dawkins sirve para expresar la siguiente idea: los genes tienen el objetivo de reproducirse a sí mismos, a costa de lo que sea, pero no a costa de otro segmento de ADN idéntico, ya que ambos son el mismo gen. Podemos decir que las células de nuestro cuerpo cooperan, e incluso que se comportan de forma altruista, ya que no intentan reproducirse a costa de sus vecinas, sino que producen un crecimiento ordenado, formando un cuerpo, tal como conviene a los genes egoístas. A nivel de individuo, cuando un individuo coopera con otro y ambos comparten genes (y dos miembros de la misma especie suelen compartir más del 90% de ellos), podemos decir que en realidad lo que ocurre es que los genes egoístas se están ayudando a sí mismos. Cuando dos individuos de la misma especie compiten, lo hacen por propagar su 10% diferencial. Es lógico que si dos individuos comparten el mismo nicho (por ejemplo, por ser de la misma especie), existirá una mayor competencia entre ellos. Pero en general, los individuos cooperarán con otros en función del número de genes en que coincidan. El apoyo mutuo entre los individuos es, efectivamente, un factor de evolución [*Kropotkin, 1970*], pero está basado en el egoísmo de los genes.

Genes y músicos de rock

Dawkins ha sido acusado de tratar los genes como unidad de selección [*Gould*] [*Sandín, 1995*]. Aunque la selección actúa sobre individuos, es evidente que la selección de individuos modifica el conjunto genético de la población, lo que indirecta y estadísticamente produce una selección de genes. ¡También podríamos criticar a casi todos los biólogos por tratar a los individuos como unidad de reproducción! Puede parecer asombroso, pero los animales no nos reproducimos, al menos, no directamente. No producimos copias de nosotros mismos: producimos copias de nuestros genes y estos, combinados tal vez con los de otro individuo, y afectados por el entorno, producirán un nuevo ser. La cuestión crucial es hasta qué punto la selección de individuos afecta a la selección de genes. Usando una metáfora, los integrantes de un grupo de rock (genes) forman un conjunto musical (cadena de ADN, genotipo) que se desarrolla componiendo canciones (fenotipo, ser vivo) que se escuchan en la radio (entorno) junto con otras canciones. La selección actúa sobre las canciones (fenotipo), pero es de suponer que la selección de canciones produce, en definitiva, una selección de músicos (genes).

Buenos, pero no tontos

Para Fernando Savater, aspectos éticos como el respeto hacia los demás son actitudes cuyo origen es en última instancia la búsqueda inteligente del beneficio propio [Savater, 1996]. Las simulaciones por ordenador parecen darle la razón. En juegos como El dilema del prisionero [Herrán, Agosto 1997] [Hofstadter, Agosto 1993], se observa que el altruismo es perjudicial para el que convive con individuos egoístas, pero el egoísmo necesita a quien explotar a largo plazo, por lo que ambas son estrategias destinadas a desaparecer. Son los pactos propios de la cooperación los que ofrecen los mejores resultados. La ética y las leyes son en realidad manifestaciones de estos pactos de cooperación que la evolución selecciona como útiles para nuestra propia supervivencia.

Computación Evolutiva

Espero haber presentado una visión aceptable del mecanismo de la evolución. Es el momento de pasar estas ideas a nuestros programas. ¿Por dónde empezar? ¿Qué tipos de Computación Evolutiva existen? Las nuevas ciencias siempre pasan por una fase inicial de desconcierto e inconsistencia hasta llegar a unas convenciones aceptadas por todos. Más o menos la clasificación aceptada en nuestros días es la del esquema siguiente.

- Inteligencia artificial.
 - Enfoque simbólico o ‘top-down’.
 - Enfoque subsimbólico o ‘botton-up’.
 - Redes neuronales artificiales.
 - Computación evolutiva o algoritmos evolutivos.
 - Solidificación o recocido simulado (simulated annealing).
 - Algoritmos genéticos.
 - Estrategias evolutivas.
 - Clasificadores genéticos.
 - Programación genética.

El enfoque subsimbólico de la IA se caracteriza por crear sistemas con capacidad de aprendizaje. Éste se puede obtener a nivel de individuo imitando el cerebro (Redes Neuronales); o a nivel de especie imitando la evolución, lo que se ha denominado Computación Evolutiva (CE); término relativamente nuevo que intenta agrupar un batiburrillo de paradigmas muy relacionados cuyas competencias no están aún muy definidas. Hasta hace poco era común hablar de Algoritmos Genéticos (AG) en general, en vez de identificar diferentes tipos de CE, ya que el resto de los algoritmos se pueden interpretar como variaciones o mejoras de los AG, más conocidos. En un AG los elementos de las cadenas (genes) son típicamente bits, y en ciertos casos, valores numéricos o strings. En la Programación Genética, los genes son instrucciones en un lenguaje de programación, y en las Estrategias Evolutivas, valores reales. Las Estrategias Evolutivas surgieron inicialmente para resolver problemas de optimización paramétrica; con el paso del tiempo fueron incorporando procedimientos propios de la computación evolutiva, con lo que han llegado a convertirse en una disciplina más. La agregación simulada (Simulated Annealing) se puede considerar una simplificación de los AG cuyo origen está en los procedimientos físicos de solidificación controlada. Los Clasificadores Genéticos solucionan problemas de reconocimiento de patrones mediante un entrenamiento basado en ejemplos, almacenando en las cadenas información que relacione los datos de entrada y la salida deseada. Finalmente, a mí me gusta considerar la Vida Artificial como un superconjunto de todas estas

técnicas [Heitkötter, Beasley, January 2000].

Opciones de un Algoritmo Evolutivo

Aunque existen otros esquemas [Herrán, Octubre 1997], por lo general, estos programas comienzan con una fase de inicialización de las entidades y su entorno, y seguidamente ejecutan repetidamente ciclos dentro de los cuales podemos distinguir tres etapas:

- **Evaluación:** se trata de asignar un valor de peso o fitness a cada individuo, en función de lo bien que resuelve el problema.
- **Selección:** ahora debemos clasificar a los agentes en cuatro tipos, según sobrevivan o no, y según se reproduzcan o no, en función de los pesos.
- **Reproducción:** se generan los nuevos individuos, produciéndose algunas mutaciones en los nacimientos.

En realidad las dos primeras fases se pueden fundir en una, ya que no es estrictamente necesario asignar a cada entidad un peso, sino simplemente saber cuáles son mejores que otras, pero asignar pesos suele ser lo más cómodo. También podemos combinar la segunda y la tercera generando las nuevas entidades únicamente en la zona de memoria donde se encuentran los agentes a eliminar, y mantener así la población constante. En cualquier caso, siempre existirá alguna forma de evaluación, selección y reproducción. Cada uno de estos procesos se puede realizar de muchas formas distintas, independientemente del problema que se esté resolviendo.

En la siguiente tabla se resumen algunas opciones de un algoritmo evolutivo. Podemos idear muchas más. En general se trata de distintas formas de producir, o bien una convergencia más rápida hacia una solución, o bien una exploración más a fondo del espacio de búsqueda. Ambas cosas son deseables y contradictorias, por lo que se ha de llegar a un compromiso. Este compromiso es lo que antes he llamado cooperación. Por supuesto que todo esto no son más que metáforas. Se requiere de una fuerza conservadora (explotación-egoísmo), que beneficie a los mejores agentes, es decir, a los que mejor resuelvan nuestro problema. Esto es evidente, pero no basta. También es necesaria una fuerza innovadora (exploración-altruismo), que permita la existencia de agentes muy distintos, aún cuando su peso sea menor. Así se puede obtener la variedad suficiente para evitar una población estancada en un máximo local, y permitir la resolución de problemas cambiantes o con varios máximos. Esto ocurre de forma espontánea en la naturaleza por ser algo inherente a cada entidad, que puede ser seleccionado, pero resulta más cómodo programarlo de forma externa, es decir, haremos que nuestro programa seleccione para la reproducción los agentes ‘buenos’, pero también unos cuantos de los ‘malos’ para mantener la variedad.

- **Opciones Generales.**
 - Número de entidades.
 - Número de elementos (genes, reglas) por cada agente.
- **Método de Evaluación:** Asignar un peso.
 - Desordenar las entidades antes de evaluarlas.
 - Diferentes formas de modificación de los pesos después de la evaluación. Por ejemplo, el peso de una entidad se puede calcular independientemente de las demás entidades, o se puede modificar posteriormente este valor, disminuyendo el peso si existe otra entidad muy parecida, analizando para ello un cierto subconjunto de la población vecina.

- **Método de Selección:** ¿Quién muere? ¿Quién se reproduce?
 - Con o sin reemplazamiento.
 - Método de la ruleta.
 - Método de los torneos.
 - Seleccionar el n% mejor y el m% peor.
- **Método de Reproducción:** Generar y mutar nuevos hijos.
 - Los padres pueden tomarse por parejas o en grupos más numerosos, elegidos al azar o en orden de pesos.
 - En el caso de detectar que los progenitores son muy parecidos, se puede realizar una acción especial, como incrementar la probabilidad de mutación.
 - Las entidades pueden comunicar a otras su conocimiento, ya sea a toda o a una parte de la población, directamente o a través de una pizarra, (una especie de tablón de anuncios).
 - Método de recombinación de genes: se puede tomar genes de uno u otro progenitor al azar, en un cierto orden, con uno, dos o más puntos de corte, etc.
 - Tasa de mutación variable.
 - Fijar una tasa de mutación diferente para cada individuo o incluso para cada gen.
 - Hacer que sea más probable que se produzca una mutación en un gen si en su vecino ya se ha producido [*Herrán, Septiembre 1997*].
 - Sustituir por mutaciones, genes sin utilidad, como reglas incorrectas o repetidas.
 - Tipos de mutaciones.

Los tipos de mutaciones requieren de una mayor explicación. Cuando los genes son bits, una mutación consiste en invertir un bit, pero ¿cómo mutar genes cuando cada gen puede tomar más de dos valores? Supongamos que tenemos un problema con tres variables de 8 bits cada una. Tendríamos cadenas como:

10001000-0101011111-11101110

Vamos a suponer que es probable que existan zonas de valores buenas y malas; es decir, que si el 7 es un buen valor y el 200 malo, será probable que también el 8 sea bueno y el 199 malo. Sería interesante que las mutaciones nos permitiesen movernos (cambiar de valor) lentamente y con exactitud si estamos en la zona buena, pero también rápida y bruscamente por si acaso nos encontráramos en la mala. Esto lo podemos conseguir precisamente cambiando un bit al azar, ya que nos movemos lentamente cambiando bits de la parte derecha, y rápidamente con los de la izquierda. En cambio, si la mutación consistiera en cambiar el valor de toda una variable por otro cualquiera, teniendo todos la misma probabilidad, podremos movernos rápidamente, pero no lentamente (al menos, con una probabilidad baja), con lo que perderíamos valores interesantes. Si las mutaciones consistieran en incrementar o restar uno de forma circular, podríamos movernos lentamente, pero no rápidamente, con lo que costaría mucho salir de la zona mala. En definitiva se trata de, dado un valor, definir la probabilidad de pasar a cada uno de los valores restantes.

Hemos supuesto que existen zonas de valores buenas y malas. Si esto no se cumpliera podríamos simplemente ‘incrementar’ uno de forma circular, para recorrer todos los valores cuanto antes. En general se observa que las mutaciones a nivel de bit son las más adecuadas, aunque no siempre ocurre así. En cualquier caso, cuando combinemos dos cadenas deberemos actuar a nivel de variable, y combinar variables completas. De lo contrario estaríamos generando multitud de mutaciones simultáneas, rompiendo por cualquier lugar valores que pueden ser valiosos.

Programas autoconfigurables

¿Qué opciones son las buenas? Tal vez sea posible determinar a simple vista si una elección es adecuada, pero en la mayoría de los casos será necesario un tedioso proceso de prueba, ya que la bondad de unas puede depender de la elección de otras. Si algo se aprende programando algoritmos evolutivos es que las ideas geniales no siempre lo son. Es muy corriente la existencia de parámetros que aunque disminuyen el número de ciclos necesarios para llegar al tipo de entidad que buscamos, en la práctica no son rentables dado el incremento de la duración de los ciclos.

Ejecutar todas las posibles combinaciones de opciones es inconcebible. ¿No será posible analizar varias en un único algoritmo, en diferentes momentos? ¿Y que tal analizar varias opciones simultáneamente, en fracciones separadas de la población? Es decir, ¿Podemos crear algoritmos evolutivos autoconfigurables? Para responder a estas preguntas es interesante agrupar todos los aspectos y opciones en cuatro categorías.

- **Tipo gen.** Los aspectos de tipo gen serán aquellos que, como los genes, son diferentes dentro de un mismo individuo.
- **Tipo individuo.** Los aspectos de tipo individuo son aquellos que se mantienen iguales en un mismo individuo, pero difieren de unos agentes a otros.
- **Tipo isla.** Los aspectos de tipo isla son aquellos que son iguales para un subconjunto de individuos; de más de uno, pero sin llegar al total.
- **Tipo población.** Los aspectos de tipo población son los que deben ser iguales para toda la población.

Las opciones o aspectos de tipo gen pueden probarse introduciéndolos como parte de cada uno de los genes, dentro de cada uno de los genes. Si se quiere, se puede pensar en una matriz de dos dimensiones, más que en una cadena de genes. Vamos a verlo en el caso de un programa que aprende a jugar al tres en raya [*Herrán, Octubre 1997*]. Cada uno de los genes es una regla que dice cómo jugar ante determinada situación. Antes de realizar un movimiento, el agente comprueba qué reglas puede usar y elige una de ellas. ¿Cómo elegirla? Podríamos pensar en asociar inicialmente a cada regla una prioridad arbitraria fija, y elegir la de mayor prioridad. Otra forma sería asignar a cada regla un peso que se incrementase cada vez que dicha regla fuera usada en una partida ganada, y aplicar la regla que en más victorias haya participado hasta el momento. Para llevar a la práctica cualquiera o las dos opciones combinadas, bastará con incluir la información como parte de cada gen. Si antes cada gen era una regla, ahora cada gen será una regla, más un valor de prioridad, más un peso. Otra opción de tipo gen sería asignar una probabilidad de mutación distinta para cada elemento de la cadena, de forma que los genes, después de ser heredados, tengan diferentes probabilidades de mutarse [*Mich, 1967*] [*Baeck, Fogel, 1997*] [*Fogel, 1995*]. Si en un determinado momento, cierto valor de un gen se muestra como muy útil, puede ser interesante que su probabilidad de mutación disminuya, y viceversa.

Los aspectos de tipo individuo pueden probarse introduciéndolos como si fueran nuevos genes, a continuación de los que ya existen. La tasa de mutación, aunque suele ser igual para toda la población, y hemos visto que podría ser incluso diferente para cada gen, podría también ser distinta para cada individuo, de forma que los hijos de ciertos agentes tengan mayor probabilidad de poseer mutaciones. Por ejemplo, en el tres en raya, una entidad estará compuesta por un conjunto de reglas más una probabilidad de mutación. Este valor se podrá heredar y mutar, y si es útil, se verá seleccionado, igual que una regla cualquiera. Antes hemos hablado de asignar una prioridad a cada regla. La prioridad asociada a cada regla es un aspecto de tipo gen, pero la decisión de asociar o no prioridades a las reglas es un aspecto de tipo individuo, que debe afectar obligatoriamente a todas sus reglas.

Probar e implementar aspectos de tipo isla va a ser más complicado. Por ejemplo, supongamos que queremos comprobar qué ocurriría si permitimos a nuestros bichos compartir su conocimiento, es decir, transmitirse información genética en forma de infección vírica, tal como relata Máximo Sandín [Sandín, 1995]. Podríamos crear varias islas o sub-grupos de agentes con distintas opciones, manteniéndolos separados por una barrera de algún tipo que no permita el contacto de unos con otros. Más tarde podríamos comparar el conocimiento obtenido en cada isla o incluso juntar de vez en cuando a algunos o a los mejores de cada grupo para la reproducción [Cases, Larrañaga]. Las opciones de tipo isla pueden ser evaluadas automáticamente sin modificar mucho el algoritmo, ya que es de esperar que los grupos con peores opciones se extingan. La idea general es que tanto para grupos como para individuos o genes, lo bueno sobrevive y lo malo perece. Para crear barreras, podemos definir tipos de especies distintas que sólo son capaces de reproducirse entre sí, poseyendo cada una de ellas, por ejemplo, distintos métodos de recombinación de genes, con uno, dos o más puntos de corte en el cruzamiento, o incluso concibiendo de forma distinta los puntos de comienzo y fin de las unidades mínimas que se recombinan (genes). Curiosamente, en este caso, el propio método de implementación de las barreras (tipos de reproducción incompatibles) es precisamente aquello que estamos probando. Para implementarlo podemos añadir un gen especial a cada agente de forma que sólo los agentes con el mismo valor en dicho gen sean capaces de recombinarse entre sí. Cada valor corresponderá con una opción diferente. Para probar más de una deberemos poner varios de estos genes, y obligar a una coincidencia en todos ellos.

Las opciones de tipo población sólo pueden probarse ejecutando algoritmos en paralelo. Estas opciones son las más generales, como el número total de agentes o el número de reglas que tiene cada individuo. Otro ejemplo de tipo población es el hecho de ajustar los pesos de los agentes, de forma que si ya existe otro agente parecido al que acaba de nacer, se disminuye el peso del nuevo agente. Esta opción no puede ser aplicada sólo a una parte de la población, ya que se vería en obvia desventaja. Evaluar opciones de tipo población es lo más complicado. Deberemos crear algoritmos independientes y compararlos. Por ejemplo, en el tres en raya, provocando una partida entre las mejores entidades de cada una de las simulaciones, es decir, creando otro plano superior de juego.

Algoritmos jerárquicos

Voy a cambiar de punto de vista para analizar las jerarquías dentro de los algoritmos evolutivos. En un algoritmo genético [Herrán, Septiembre 1997], la reproducción, selección, mutaciones, asignación de pesos, etc. se desarrollan a un sólo nivel o plano. Sin embargo, en el caso del tres en raya [Herrán, Octubre 1997], existen varios planos.

En el plano más elevado, juegan el ordenador contra el hombre. Este juego puede interpretarse como un algoritmo evolutivo reducido a su mínima expresión, con una población de dos individuos, donde el algoritmo termina al finalizar la partida. En este tipo de problemas tienen bastante éxito los paradigmas en los que el ordenador trata de aprender observando el juego humano, pero no es este el caso. Para aprender, el ordenador crea un segundo nivel de juego, donde muchas entidades jugarán por parejas. Aquella entidad que más partidas gane será la que represente al ordenador en el juego contra el hombre. Pero todavía se puede hablar de un tercer nivel, ya que dentro de un mismo agente, cada una de las reglas puede poseer también un peso, y aunque las reglas no se reproducen combinando sus elementos (podrían hacerlo), es posible sustituir algunas cada cierto tiempo.

Ya que la entidad de nivel superior ha creado un sub-mundo o plano donde juegan varias sub-entidades, gracias a las cuales es capaz de aprender por su cuenta antes de jugar contra el hombre, estas sub-entidades podrían a su vez crear otros planos que les proporcionasen conocimiento, hasta que llegue el momento de jugar 'de verdad'. Es decir, una entidad, al igual que jugar, o reproducirse, podría también tener una acción que podríamos llamar 'pensar' y que consistiera en experimentar partidas hipotéticas (hipotéticas para la entidad superior; a la inferior le parecerán las suyas tan reales como a cualquier otra), creando 'mundos virtuales', de la misma forma que las personas imaginamos o creamos hipótesis para prever los acontecimientos.

Éste sería además un esquema donde implementar el algoritmo minimax típico de los juegos, donde el agente evalúa posibles movimientos propios y del contrario a varios niveles de profundidad. Podemos combinar este tipo de jerarquías con las relativas a la elección de opciones de tipo población. Es decir, mientras buscamos la solución a nuestro problema inicial en los algoritmos o planos inferiores, podemos experimentar en un plano superior algunas opciones, las cuales pueden a su vez tener subopciones.

El problema más difícil

Una cadena de un algoritmo evolutivo tiene varios elementos (genes) que corresponden con variables del problema a resolver. Puede haber genes cuyos valores sean intrínsecamente buenos o malos, es decir, buenos o malos independientemente de los valores del resto de la cadena. Vamos a darles un nombre a este tipo de genes, por ejemplo, genes no-vinculados. Consecuentemente, un gen cuya bondad o maldad dependa de los valores de algunos de los otros genes será vinculado.

La propiedad de ser o no vinculado puede aplicarse también a un conjunto de elementos de la cadena, si tienen en cuenta todos ellos juntos, como un único gen, sin requerir que sean contiguos. Imaginemos un problema en el que todos los genes fueran no-vinculados: ¡No sería necesario un algoritmo evolutivo! Para encontrar los valores óptimos de las variables, bastaría con analizar independientemente todos los posibles valores de cada gen, manteniendo constante el resto de la cadena con cualquier valor. Imaginemos ahora que todos los segmentos son vinculados, menos dos, es decir, todos los genes son intrínsecamente buenos o malos, excepto dos, que dependen de otros. Sí, pero ¿de cuáles? ¿El uno del otro? ¿Ambos de todos los restantes? Si la bondad de cada uno de estos dos segmentos sólo dependiera de sí mismo y del valor del otro podrían ser tratados en conjunto como un único gen no-vinculado, y aplicar el mismo proceso. Si la bondad de los genes vinculados dependiera de todos los demás podremos utilizar el mismo método, teniendo la precaución de calcular primero los valores de los segmentos no-vinculados.

Imaginemos ahora el caso extremo de que todos los genes sean vinculados, y sus efectos siempre dependan del valor de todos los demás genes. Este no sería un problema irresoluble, pero sí el peor que nos podríamos encontrar, no sólo para las técnicas evolutivas, sino para cualquier otro método. Sería algo así como buscar un objeto con los ojos cerrados, y la mejor forma de tratarlo sería, lógicamente, una búsqueda secuencial. En cualquier caso, lo que más nos interesa es buscar segmentos y grupos de segmentos no-vinculados, aunque sean grandes, ya que si podemos dividir la cadena en varios segmentos no-vinculados podremos aplicar un algoritmo independiente para cada segmento, en el que el resto de la cadena sea siempre fijo, con un valor cualquiera, y obtener la solución mucho más rápido.

«En un problema complicado, los valores de los genes no son intrínsecamente buenos ni malos.»

La existencia de fuertes vinculaciones (también llamadas ‘interacciones’) entre los genes se ha denominado «el problema de la epístasis» o «el problema de la ausencia de bloques constructores». Los problemas con poca epístasis son triviales, y se pueden solucionar con un simple método de escalada. Para saber si un elemento (o un conjunto de elementos) es no-vinculado podríamos fijar valores al azar para el resto de la cadena, evaluar las cadenas que resulten de todas las posibles combinaciones de valores para el conjunto de elementos elegido y almacenar el orden de las cadenas ordenadas según su peso. Si repitiendo el proceso utilizando distintos valores en el resto de la cadena siempre obtenemos las cadenas en el mismo orden, entonces estaremos ante un segmento no-vinculado. Otra forma sería comprobar estadísticamente si en distintos ciclos del algoritmo, los mismos valores para un segmento producen posiciones relativas parecidas dentro de la lista de agentes ordenados por peso. Por ejemplo, si una variable puede tomar valores de 1 a 10, podemos almacenar, para cada valor, cuál es la posición media que ocupa la cadena que la contiene, dentro de la lista de cadenas ordenadas por pesos, en cada uno de los ciclos. Otra forma de encontrar las vinculaciones sería definir las al azar y actuar como si de hecho existiesen, comprobando si de esta forma se obtienen mejores resultados.

Por último, voy a presentar la que me parece la forma más fácil de todas. Para el algoritmo es mucho más sencillo encontrar los valores óptimos de los segmentos no-vinculados que los de los vinculados. Por tanto, después de unos cuantos ciclos, los genes de valores más homogéneos tendrán una mayor probabilidad de ser más no-vinculados que el resto. Los genes vinculados se comportan en realidad como un único gen, por lo que se ha de evitar su división. Se me ocurre que podríamos obtener alguna ventaja si obligamos a que exista una especie de cohesión entre los genes vinculados, es decir, entre los segmentos de valores más heterogéneos, de forma que se hereden y muten juntos, simultáneamente, con una mayor probabilidad.

Referencias bibliográficas

Sobre Vida Artificial

Ray, Thomas S. Jugué a ser Dios y creé la vida en mi computadora

En <http://www.hip.atr.co.jp/~ray/pubs/spanish/spanish.html>. Emocionante relato que describe una de las primeras simulaciones de vida por ordenador.

Prata, Stephen (1993) Vida artificial (Ed. Anaya)

Buen libro para iniciarse en el tema.

Swarm: multiagent simulation of complex systems

En <http://www.santafe.edu/projects/swarm/>. Plataforma de desarrollo de vida artificial para Unix Linux GNU.

J. J. Merelo Vida Artificial en español (Universidad de Granada)

En <http://kal-el.ugr.es/VidArt/VidaArti.html>.

Linux Alife

En <http://aisun0.ai.uga.edu/~jae/ai.html>. Vida Artificial y recursos para Linux.

Herrán, Manuel de la (Agosto 1997) "**Vamos a crear vida en un PC**" en *Sólo Programadores* (n. 36)

Forrest, Stephanie (Ed.) (1993) "**Emergent computation: self organizing, collective, and cooperative phenomena in natural and artificial computing networks**" en *Artificial Intelligence* (n. 60)

Shoham, Yoar (1993) "**Agent oriented programming**" en *Artificial Intelligence* (n. 60)

Fernandez Ostolaza, Julio y Moreno Bergareche, Alvaro (1992) **Vida artificial** (Ed. Eudema)
Epistemología de la Vida Artificial.

Sobre computación evolutiva y aprendizaje.

Holland, John H. (Septiembre 1992) "**Algoritmos genéticos**" en *Investigación y Ciencia*
Fantástico resumen de 'el padre' de los algoritmos genéticos.

Cases, Blanca y Larrañaga, Pedro **Matemáticas aplicadas: vida artificial y algoritmos genéticos**
(Universidad del País Vasco)
En <http://www.geocities.com/CapeCanaveral/9802/>. Muy interesante.

Sierra Molina, Guillermo y otros (1995) **Sistemas expertos en contabilidad y administración de empresas** (Ed. Ra-Ma)
Algoritmos Genéticos, Redes Neuronales y otras técnicas de resolución de problemas, explicado de una forma clara y sencilla.

Herrán, Manuel de la (Septiembre 1997) "**Algoritmos genéticos avanzados**" en *Sólo Programadores* (n. 37)

Drescher, Gary L (1991) **Made up minds** (MIT Press)
Fantástico libro sobre el aprendizaje automático.

Herrán, Manuel de la (Octubre 1997) "**Aprendizaje automático y vida artificial**" en *Sólo Programadores* (n. 38)

Holland, John H. (1975) **Adaptation in natural and artificial systems** (University of Michigan Press)

Hofstadter, Douglas R. (Agosto 1983) "**Temas Metamágicos**" en *Investigación y ciencia*

Herrán, Manuel de la (Noviembre 1996) "**Agentes autodidactas ¿Futuro o realidad?**" en *BASE* (n. 30)

Jáuregui, José Antonio (1990) **El ordenador cerebral** (Ed. Labor)
Un punto de vista muy original acerca del ser humano, la inteligencia y el darwinismo.

Heitkötter, Jörg and Beasley, David, (eds.) (January 2000) **The Hitch-Hiker's guide to evolutionary computation**

En <http://www.etsimo.uniovi.es/ftp/pub/EC/FAQ/www/top.htm>. FAQ de comp.ai.genetic con definiciones y clasificaciones de los términos más comunes en Computación Evolutiva.

U. Mich, Rosenberg (1967) **Ph.D. dissertation**

Baeck, Thomas y Fogel, David B. (eds.) (1997) **Handbook of evolutionary computation** (Oxford, NY)

Fogel, David B. (1995) **Evolutionary computation: toward a new philosophy of machine intelligence** (IEEE Press)

Sobre Darwinismo, Genética, y Evolución

Dawkins, Richard (1994) **El gen egoísta** (Ed. Salvat Ciencia)

Leakey, Richard E (1994) **Introducción en 'El Origen de las especies' de Charles Darwin** (Ediciones del Serbal - Reseña)

Exposición sencilla y muy acertada del darwinismo.

Dawkins, Richard Enero 1996 "¿Tiene sentido la viuera de sí misma?" en Investigación y Ciencia

Hoyle, Fred (1983) **El universo inteligente** (Ed. Grijalbo)

Una valiente crítica al darwinismo.

Kropotkin, Pedro (1970) **El apoyo mutuo** (Ed. Madre Terra)

Trata el darwinismo y la cooperación.

Castrodeza, Carlos (Septiembre 1997) "**La evolución sin Darwin: La biología ultramontana**" en Revista de Libros (n. 9) (Ed. fundación Caja Madrid)

Barnett y Otros (1962) **Un siglo después de Darwin 1. La Evolución** (Alianza Ed.)

Gould, Stephen Jay **El pulgar del panda** (Ed. Hermann Blume)

Sandín, Máximo (1995) **Lamarck y los mensajeros. La función de los virus en la evolución** (Ed. Istmo)

Sobre Redes Neuronales

Dewney, A. K. (1990) "**Perceptrones**", capítulo de Dewney, A. K. (1990) **Aventuras Informáticas** (Ed. Labor)

Los perceptrones corresponden con el nacimiento de las redes neuronales, y suponen una versión simplificada de éstas.

Sierra Molina, Guillermo y otros (1995) "**Redes neuronales**", capítulo de Sierra Molina, Guillermo y otros (1995) **Sistemas expertos en contabilidad y administración de empresas** (Ed. Ra-Ma)

Explica el funcionamiento más básico de redes neuronales.

Luna, Raúl (1994) "**Redes neuronales**" en Sólo Programadores

Obiol, Pablo (Julio 1997) "**Redes neuronales**" en Jumping

Ignacio Olmeda y S. Barba (1993) "**Aprendizaje y Generalización**" y "**Redes Borrosas y Evolutivas**" capítulos de Ignacio Olmeda y S. Barba (1993) **Redes neuronales artificiales: fundamentos y aplicaciones**

Olmeda, Ignacio y Pérez, Joaquín (1994) **Chaos, artificial neural networks and the predictability of stock market returns**

Isasi, Pedro; Olmeda, Ignacio; Fernández, Eugenio, y Fernández, Camino **Neural networks forecasts of intra-day futures and cash returns**

Otras

Adams, Douglas (1983) **Guía del autoestopista galáctico** (Ed. Anagrama)

En <http://www.cyberbiology.org/adams.html>. Ciencia-ficción, relato de humor. Presenta el planeta Tierra como un super-ordenador biológico.

Watson, Ian (1987) **Visitantes milagrosos** (Grupo Zeta)

Estados alterados de conciencia, fenómeno ovni, ecología, enseñanzas sufíes, mecánica cuántica y teorema de Gödel. Único.

Gisbert, Joan Manuel (1981) **El misterio de la isla de Tökland** (Ed. Espasa-Calpe)

Redfield, James (1997) **Las nueve revelaciones** (Ediciones B)

Evolución, Teilhard de Chardin y Gestalt.

Morgan, Marlo (1996) **Las voces del desierto (Muttant Message Down Under)** (Ediciones B)

Chalmers, David J. (Febrero 1996) "**El problema de la consciencia**" en Investigación y Ciencia

Savater, Fernando (1996) **Ética para amador** (Ed. Ariel)

Fecha de referencia: 15-10-2002

1: Publicado en marzo de 1998 en la revista **Sólo Programadores** (<http://www.towercom.es/>) n. 43, y revisado en abril de 1999 por el autor. Copyright 1998-2001. Este artículo completa la serie de los artículos aparecidos en los números 36, 37 y 38 de dicha revista.

2: Manuel de la Herrán (<http://www.manuherran.com/>) nació en Bilbao en 1971. Es ingeniero informático y ha trabajado en varias empresas desarrollando tecnología en Internet y dirigiendo sus departamentos técnicos. Ha escrito numerosos artículos sobre Evolución, Computación Evolutiva, Algoritmos Genéticos, Inteligencia Artificial, OLAP y Bases de Datos Multidimensionales, Bases de datos Oracle y Programación en Internet. Ha sido profesor de la Universidad de Deusto y es el creador de webs como Gaia, <http://www.redcientifica.com/gaia/> (finalista iBest 2000); REDcientífica <http://www.redcientifica.com/> (Premio Nacional Sociedad de la Información); REDhumana.com, <http://www.redhumana.com/>; iieh.com <http://www.iih.com/>; y MUNDO.ENk3.com, <http://mundo.enk3.com>.

Boletín CF+S > 21 -- El pasado es un país extraño > <http://habitat.aq.upm.es/boletin/n21/amher1.html>

Edita: Instituto Juan de Herrera. Av. Juan de Herrera 4. 28040 MADRID. ESPAÑA. ISSN: 1578-097X