

Arreglos neuronales evolutivos. Un nuevo mecanismo para implementar evolución incremental.

Lic. Leonardo Corbalán, Lic. Laura Lanzarini
Universidad Nacional de La Plata, Facultad de Informática,
La Plata, Argentina, 1900
{corbalan,laural}@lidi.info.unlp.edu.ar

Abstract

Incremental evolution has proved to be an extremely useful mechanism in complex actions sequence learning. Its performance is based on the decomposition of the original problem into increasingly complex stages whose learning is carried out sequentially, starting from the simplest stage and thus increasing its generality and difficulty.

The present work proposes neural array applications as a novel mechanism for implementing incremental evolution. Each array is composed by several neural nets obtained by means of an evolutive process allowing them to acquire various degrees of specialization. Neural nets constituting the same array are organized so that, in each assessment, there is only one in charge of its response.

The proposed strategy is applied to problems presented by the elusion of hindrances and the fulfillment of goals as a means to show this proposal's capability for solving complex problems. The measurements carried out show the advantage of evolving neural arrays with respect to Moriarty's SANE method.

Finally, conclusions are presented as well as some future lines of work.

Key Words: Evolving Neural Nets, Learning, Incremental Evolution, Genetic Algorithms.

Resumen

La evolución incremental ha demostrado ser un mecanismo sumamente útil en el aprendizaje de secuencias de acciones complejas. Su funcionamiento se basa en la descomposición del problema original en etapas de complejidad creciente cuyo aprendizaje se realiza en forma secuencial comenzando por la etapa más simple e incrementando su generalidad y dificultad.

El presente trabajo propone la aplicación de arreglos neuronales como un nuevo mecanismo para implementar evolución incremental. Cada uno de estos arreglos está formado por varias redes neuronales, obtenidas a través de un proceso evolutivo, que les permite adquirir diferentes grados de especialización. Las redes neuronales que constituyen un mismo arreglo se organizan de modo que, en cada evaluación, sólo una se haga cargo de la respuesta del mismo.

La estrategia propuesta se aplica a los problemas de evasión de obstáculos y alcance de objetivos como forma de demostrar la capacidad de esta propuesta para la resolución de problemas complejos. Las mediciones realizadas muestran la superioridad de los arreglos neuronales evolutivos con respecto al método SANE propuesto por Moriarty.

Finalmente se presentan las conclusiones y se plantean algunas líneas de trabajo futuras.

Palabras claves: Redes Neuronales Evolutivas, Aprendizaje, Evolución incremental, Algoritmos Genéticos.

1. Introducción

Las Redes Neuronales Evolutivas (RNE) son un caso particular de las redes neuronales artificiales (RNA) en las cuales la adaptación no sólo se debe al aprendizaje sino también a la evolución [14]. La evolución se ha utilizado en diversas formas para conseguir los pesos de conexión, el diseño de la arquitectura, el valor de los parámetros iniciales, las reglas de aprendizaje, etc. [15]. La práctica habitual es tomar el mejor individuo de la última generación desechando todas aquellas redes neuronales que posean un valor de aptitud inferior.

No obstante, desde el punto de vista biológico, la actividad cerebral no parece surgir de una única red neuronal. Estudios realizados en humanos por Jackson y Penfield, alrededor de 1950, revelaron la existencia de una notable localización funcional en la corteza cerebral. Más tarde, se reconocieron otros centros neuronales en el interior del cerebro relacionados con las emociones y la actividad intelectual [12]. Así, es posible que el cerebro esté compuesto por muchos sistemas diferentes paralelos y distribuidos, que llevan a cabo funciones bien definidas.

Freeman y Skapura [3] argumentaron la necesidad de aprender a combinar RNAs pequeñas, poniéndolas bajo el control de otras redes para resolver el problema del escalado. Xin Yao y Yong Liu [13] estudiaron los beneficios de utilizar la población completa de redes neuronales obtenidas en la última generación, producto de un proceso evolucionario, en lugar de únicamente la de mejor fitness. Más recientemente, Bruce y Miikkulainen [1], trabajando sobre el problema de reconocimiento de caracteres escritos a mano alzada, demostraron que todas las redes neuronales de una población, combinado con una técnica efectiva de especialización, pueden responder mejor de manera colectiva que cualquiera de ellas individualmente.

Finalmente, Gómez y Miikkulainen han demostrado que la evolución incremental puede aplicarse satisfactoriamente en la resolución de problemas complejos [5].

2. Objetivo

Relacionando las investigaciones mencionadas previamente, este artículo propone una nueva metodología para resolver problemas complejos basada en arreglos neuronales evolutivos. Durante la evolución, cada individuo de la población está formado por más de una red neuronal e intenta resolver el problema con la participación colectiva de sus miembros y alguna división funcional de la tarea. El aprendizaje se basa en una estrategia incremental que evalúa la población en varias etapas, sobre diferentes tareas de complejidad creciente, hasta alcanzar el comportamiento final deseado. En la sección 3 se describe detalladamente el método propuesto.

La evolución incremental es aplicable a problemas que puedan naturalmente descomponerse en una secuencia de tareas de complejidad creciente [5].

Problemas como la evasión de obstáculos y el alcance de objetivos representan un verdadero desafío para las soluciones evolutivas ya que el comportamiento a aprender varía según las características del entorno. Por tal motivo, ha sido seleccionado para medir los resultados de esta propuesta.

En la sección 4 se explica el problema a resolver junto con la descripción del agente y la manera de calcular su fitness. En la sección 5 se analiza la aplicación de los arreglos neuronales evolutivos en la resolución de este problema. En este punto y con el objeto de explicar la representación utilizada, se describe brevemente SANE y las extensiones necesarias para ser utilizado en este caso. Finalmente, las secciones 6 y 7 muestran los resultados obtenidos y las conclusiones y líneas de trabajo futuras, respectivamente.

3. Arreglos Neuronales Evolutivos (ANE)

ANE simula la comprobada localización funcional del cerebro donde se aprecian distintos circuitos neuronales relacionados con funciones cognitivas, sensoriales y motoras concretas. Para ello se evoluciona una población de arreglos de redes neuronales artificiales. Las redes integrantes de un arreglo aprenden a especializarse en distintos aspectos de un problema y a delegar en sus compañeras cuestiones ajenas a su incumbencia. Así, de su accionar coordinado dentro del arreglo, surge la resolución de un problema complejo en forma más eficiente.

3.1 Organización interna de los Arreglos neuronales

Un arreglo neuronal es una n -upla de redes neuronales de la forma $G=(rn_1, rn_2, \dots, rn_n)$. Se define la noción de arreglo vacío como aquel que no contiene elemento y se denota $()$. También se define el operador *ins* que inserta una red neuronal como primer elemento de un arreglo neuronal y se denota con el signo “:”

$$rn : () = (rn) \qquad rn : (rn_1, rn_2, \dots, rn_k) = (rn, rn_1, rn_2, \dots, rn_k)$$

Un arreglo neuronal acepta una entrada de datos, se evalúa y produce la salida correspondiente. Las redes

neuronales que lo constituyen se organizan para que en cada evaluación sólo una se haga cargo de la respuesta. Para ello, se evalúan secuencialmente de acuerdo a su posición en el arreglo, de izquierda a derecha, con la misma entrada de datos, hasta hallar aquella que producirá la salida. La razón de este orden de evaluación se expone más adelante pues surge de los detalles de la estrategia de evolución incremental implementada.

Las redes del arreglo cuentan con una neurona de salida extra a la cantidad definida por el problema a resolver que le permite indicar si dará la respuesta en nombre del arreglo o no. Esta neurona produce, para cada entrada de datos, un número real del intervalo [0..1]. Una estrategia sencilla para la selección de la red que producirá la respuesta a una entrada de datos dada, aunque pueden implementarse otras, es evaluar las mismas en el orden prefijado hasta hallar una red cuya salida extra sea mayor que 0.5. La última red de la secuencia de evaluación puede prescindir de la neurona extra de salida ya que si es elegida deberá brindar la respuesta de todos modos.

3.2 Aprendizaje incremental en los arreglos neuronales

La evolución incremental es un mecanismo muy útil para el aprendizaje de comportamientos complejos. Su funcionamiento se basa en la descomposición del problema original en subproblemas de la siguiente forma: Sea O el objetivo final que se desea alcanzar, debe definirse previamente una serie de subobjetivos más simples de complejidad creciente O_1, O_2, \dots, O_n de forma que $O_1, O_2, \dots, O_n = O$. Es decir que, el proceso evolutivo se encuentra dividido en tantas etapas como subobjetivos se hayan definido (Figura 1).

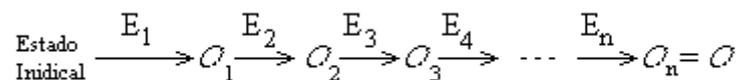


Figura 1

La evolución incremental sobre una población de arreglos neuronales utiliza un mecanismo que inserta en los arreglos una nueva red neuronal al mismo tiempo que la evolución avanza hacia el próximo subobjetivo.

En la etapa E_i , la población conformada por arreglos de exactamente i redes neuronales, evoluciona hacia el subobjetivo O_i . De esta forma, la evolución comienza con una población de arreglos de dimensión 1. Finalizada una etapa, se guarda el mejor arreglo conseguido y comienza la siguiente etapa hacia el próximo subobjetivo con la inicialización de una nueva población donde cada cromosoma codifica una red neuronal. Los fenotipos a evaluar (arreglos neuronales) se crean decodificando la red neuronal del cromosoma e insertándola como primer elemento del arreglo salvado en la etapa previa. En cualquiera de las etapas, el fenotipo asociado a cada cromosoma de la población no es una red neuronal sino un arreglo de ellas codificado parcialmente.

Durante la evaluación necesaria para la asignación del fitness, los datos de entrada se presentan a la primera red de cada arreglo, que es la única integrante sometida a evolución en esa etapa. Esta decide, por medio de la neurona extra de salida, responder por sí misma o delegar la tarea a la conformación anterior del arreglo, pasando el control a la segunda red neuronal. Así, las redes del arreglo se evalúan de izquierda a derecha pues este ordenamiento representa el orden inverso de inserción de las mismas en cada una de las etapas.

Obsérvese que de esta manera, en la etapa E_i del proceso evolutivo, la red sometida a evolución que se inserta en un arreglo capaz de alcanzar el subobjetivo O_{i-1} , debe aprender a delegar la funcionalidad necesaria para alcanzarlo (utilizando la neurona de salida extra) y a resolver por sí misma las circunstancias para superarlo hasta alcanzar O_i .

3.3 Algoritmo evolutivo utilizado

En un problema O compuesto por una serie de N subobjetivos más simples de complejidad creciente de la forma: $O_1, O_2, \dots, O_n = O$ el algoritmo utilizado para evolucionar arreglos neuronales es el siguiente:

$A = ()$ //arreglo vacío

Para cada etapa E_i , con $i=1, \dots, n$

generar aleatoriamente una población P de redes neuronales $P = \{ rn_1, rn_2, \dots, rn_k \}$

construir la población de arreglos neuronales $P_A = \{ rn_1 : A, rn_2 : A, \dots, rn_k : A \}$

evaluar cada arreglo neuronal de P_A asignando el fitness a los individuos de P (*)

Mientras no se alcance el objetivo O_i

$P = \text{próxima_generación}(P)$ (**)

construir la población de arreglos neuronales $P_A = \{ rn_1 : A, rn_2 : A, \dots, rn_k : A \}$

evaluar cada arreglo neuronal de P_A asignando el fitness a los individuos de P

Fin Mientras

$A = \text{mejor arreglo neuronal obtenido durante la etapa } E_i$

Fin Para

donde k y n son parámetros.

(*) El valor de fitness asignado a cada red neuronal rn_i de P es aquel obtenido por el arreglo del cual constituye el primer elemento.

(**) ANE no asume ningún algoritmo evolucionario específico para conseguir la próxima generación en la población P de redes neuronales. Puede utilizarse un algoritmo genético simple como el presentado por Goldberg [4] u otros más sofisticados. Suele ser conveniente utilizar estrategias neuroevolutivas, especialmente diseñadas para aplicarse a redes neuronales, como SANE ([8], [9], [10]) o ESP ([5], [6], [7], [11]) que han demostrado superar a los algoritmos tradicionales. La codificación genética adoptada, los operadores genéticos utilizados y los mecanismos de selección, reproducción y reemplazo, dependerán del algoritmo elegido. Tampoco se asume ninguna restricción sobre los parámetros de la red sometidos a evolución (pesos de conexión, arquitectura, función de transferencia, etc.). Puede hallarse un conjunto de variantes en [15] donde se citan varias investigaciones en neuroevolución incluyendo hibridaciones con algoritmos de aprendizaje tradicionales.

4. Evasión de obstáculos y alcance de objetivos

Se pretende conseguir comportamiento inteligente en un agente que se mueve libremente en dos dimensiones dentro de los confines de un entorno virtual, de forma rectangular, cuyos límites no puede sobrepasar. Objetos como obstáculos, que deberá aprender a evitar, y objetivos móviles o estáticos, a los que deberá alcanzar, interactúan con él en este ambiente virtual.

Un arreglo neuronal controla los movimientos del agente en un intervalo temporal simulado por la sucesión de n instantes discretos de tiempo (pasos de simulación). En cada instante el arreglo es estimulado por un conjunto de señales de entrada. La salida está conformada por un par ordenado $(\Delta x, \Delta y)$ que determina el desplazamiento que realiza el agente sobre la superficie. No obstante, los obstáculos presentes y los límites del entorno pueden impedir que el movimiento se lleve a cabo.

4.1 Agente

Cada agente posee 13 sensores y un reloj interno que trabajan de la siguiente forma:

- 8 sensores, distribuidos uniformemente a su alrededor, capaces de detectar objetivos sin importar la distancia a la cual se encuentren.
- 4 sensores que le permiten detectar obstáculos, sólo a corta distancia, en cuatro posibles direcciones (N, S, E y O).
- 1 sensor que indica si hubo colisión en cualquier punto de la circunferencia del agente .
- un reloj interno que posibilita algún cambio de conducta en función del tiempo transcurrido aún cuando los estímulos externos no varíen.

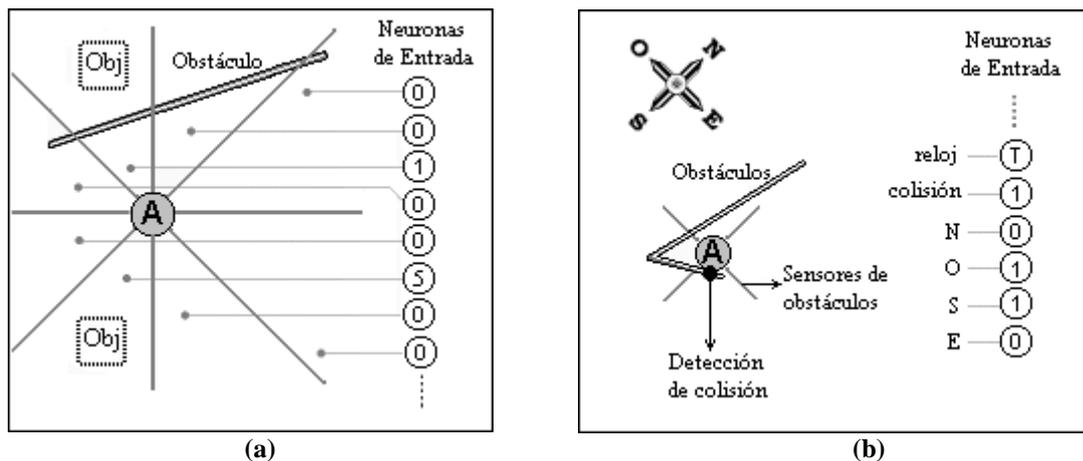


Figura 2

Los sensores de objetivos proporcionan tres valores posibles (0; 1 ó 5) indicando respectivamente ausencia, presencia detrás de algún obstáculo, o visualización directa de un objetivo. La figura 2.a) ejemplifica esta situación. De los 8 sectores donde analiza la presencia de objetivos, sólo dos de ellos tienen información. El valor más alto, 5, corresponde al objetivo visto directamente, mientras que el que se encuentra detrás del obstáculo es sentido con valor 1.

Los detectores de obstáculos pueden considerarse como prolongaciones sensibles al tacto orientadas en los cuatro

puntos cardinales y al igual que el sensor de colisión comunican dos valores 1 ó 0 (presencia o no de estímulo). La figura 2.b) indica los valores sensados para un agente que ha detectado obstáculos en las direcciones Sur y Oeste; por tal motivo, el sensor de colisión toma el valor 1 para indicar lo ocurrido.

El valor de reloj pertenece al intervalo real $(0,1]$ y constituye una apreciación del tiempo de simulación, así para el paso corriente p de una simulación de n pasos, $T = p/n$.

4.2 Asignación de fitness

El problema de evasión de obstáculos y alcance de objetivos pertenece a una clase genérica de problemas denominada *Sequential decision tasks*. Su característica principal es la dificultad para asignar con precisión la bondad de una acción tomada, siendo necesaria una secuencia de decisiones antes de poder medir cuál ha sido el efecto de cualquiera de ellas. Juegos como el ajedrez y damas, entre otros, caen dentro de esta categoría. Como ejemplos del mundo real pueden mencionarse: el enrutado de información en los routers de la Internet, el control de flujo químico en un reactor químico, el control de tráfico aéreo, etc. En todos estos casos, el efecto de una decisión simple se evidencia transcurrido un lapso de tiempo, y aún así, frecuentemente es difícil establecer cuáles decisiones fueron las responsables, y en qué medida, de lo acontecido [9].

Se ha enmarcado a la evasión de obstáculos y alcance de objetivos dentro de este tipo de problemas y por lo tanto, no se hace ninguna valoración de aptitud hasta que no haya concluido la simulación utilizada para evaluar cada individuo de la población.

Una vez finalizada la simulación, se calcula el fitness del arreglo neuronal que controla al agente de la siguiente manera: Sea $f(a)$ el valor de aptitud asignado al agente a . Si el objetivo no fue alcanzado, $f(a)$ tomará un valor del intervalo $[0,100)$ calculado proporcionalmente al camino recorrido hacia el objetivo. Por el contrario, si el objetivo es alcanzado, $f(a)$ pertenecerá al intervalo $[100,150)$. Si p es el número de pasos utilizados para alcanzar el objetivo, y P el número total de pasos de la simulación entonces el fitness se calcula como $f(a)=100+50(P-p)/P$. De esta forma se obtiene un valor entre 100 y 150 proporcional a la velocidad con que se ha alcanzado el objetivo.

5. ANE aplicado a evasión de obstáculos y alcance de objetivos

El método ANE ha sido testeado sobre la evasión de obstáculos y alcance de objetivos estáticos o móviles (captura de presa) que representan problemas de *control de procesos* difíciles de evolucionar. Además, plantean los inconvenientes propios de los problemas del tipo *Sequential decision tasks*, por lo que resulta interesante resolverlos.

5.1 Detalles de implementación

Se ha implementado ANE dividiendo el proceso evolutivo en dos etapas. Esta elección llevará a trabajar con arreglos formados por dos redes neuronales (figura 3) y, como puede observarse en la sección 6, ha sido suficiente para demostrar que el método supera en varios casos a otras estrategias neuroevolutivas.

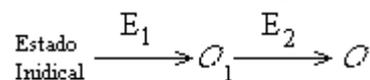


Figura 3

Cada etapa posee longitud fija expresada en cantidad de generaciones. El objetivo final O es controlar a los agentes para que, partiendo desde un origen determinado, alcancen su destino en la menor cantidad de pasos de simulación posible. Se considera como destino algún lugar geográfico fijo o algún blanco en movimiento.

Reduciendo la complejidad de la travesía que deben realizar los agentes hacia su destino se obtiene el subobjetivo O_1 . Su función es controlarlos para que, partiendo desde algún lugar intermedio del camino entre origen y destino, alcancen este último en el menor tiempo posible, o al menos se aproximen lo suficiente una vez finalizada la cantidad de generaciones permitidas para la etapa E_1 (figura 4).

Así se espera que en la segunda etapa los arreglos de la forma (m_1, m_2) sean capaces de utilizar m_1 para conducir al agente a las cercanías del punto de partida de la etapa anterior, desde donde podría tomar el control m_2 especializada en el recorrido desde allí hasta el destino final. Cabe aclarar que los agentes no poseen información alguna sobre el sitio desde donde partieron en la etapa 1. La organización en la delegación de control queda determinada en función del valor de la tercera neurona de salida de m_1 que depende únicamente de los valores de entrada que el agente sensa en cada paso de simulación.

En la presente implementación del método ANE, se ha impuesto la siguiente restricción temporal al cambio de control: una vez que m_1 delega la tarea a m_2 , lo hace hasta que haya alguna colisión o finalice la simulación. Si ocurre una colisión, m_1 vuelve a tomar intervención en la decisión de control optando por responder por sí misma o

volviendo a transferir la responsabilidad a m_2 .

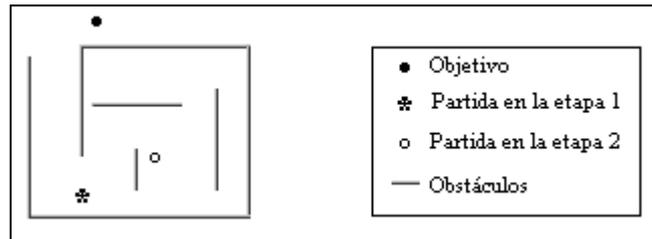


Figura 4

Esta restricción responde a la necesidad de frenar el cambio repetitivo de control que suele producir movimientos cíclicos indeseables en los agentes y está relacionada con la implementación particular que se hizo sobre los problemas de evasión de obstáculos y alcance de objetivo.

5.2 Arquitectura de las redes neuronales artificiales

Para la conformación de los arreglos se utilizan redes recurrentes con esquema de conexión libre, con término de tendencia y evolución de función de transferencia pudiendo cada nodo poseer una de entre cuatro sigmoides distintas: $f_1(x)=1/(1+\exp(-0.5x))$, $f_2(x)=1/(1+\exp(-x))$, $f_3(x)=1/(1+\exp(-1.5x))$, $f_4(x)=1/(1+\exp(-2x))$ (figura 5). Este tipo de redes ha mostrado un mejor rendimiento que las feedforward al ser aplicadas a los problemas de evasión de obstáculos y alcance de objetivos, permitiendo obtener mejores individuos en menos generaciones [2].

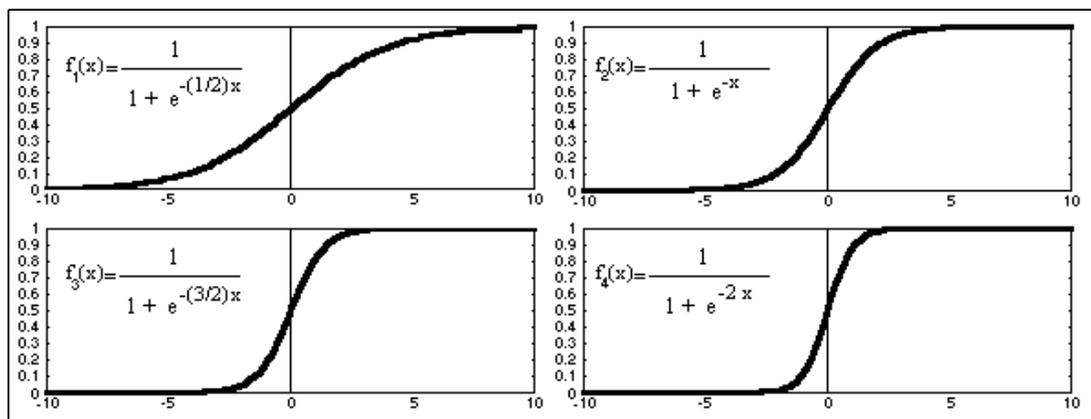


Figura 5

En cuanto a la representación seleccionada, se utilizó una variante de SANE implementada especialmente para la resolución de este problema. Como introducción para analizar las extensiones propuestas, se presenta a continuación una muy breve descripción del método SANE; puede consultarse [9] para una explicación más detallada.

5.2.1. SANE

En las soluciones convencionales que evolucionan Redes Neuronales, cada individuo de la población representa una red neuronal completa. Esto no ocurre en SANE donde cada individuo es sólo una neurona de la red y deben ser combinados para expresar la solución buscada. Por esto, SANE utiliza dos poblaciones, una población de neuronas y otra de "blueprints" que representan las redes neuronales a analizar.

Dentro de la población de neuronas, cada individuo representa un nodo oculto de una RNA feedforward de una sola capa oculta. La red neuronal completa se crea por la combinación de varias neuronas elegidas de esta población. En el cromosoma cada neurona se codifica como un conjunto de conexiones de la forma (etiqueta, peso) (Figura 6). El valor del peso de la conexión queda especificado por el campo *peso* (16 bits) interpretado como número real. La neurona con que se conecta queda determinada por el campo *etiqueta* (8 bits) de la siguiente manera: si el valor decimal de la etiqueta es D , n_E es la cantidad de neuronas de entrada y n_S es la cantidad de neuronas de salida:

Si $D > 127 \rightarrow$ la conexión es hacia la neurona de salida ($D \bmod n_S$)

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada ($D \bmod n_E$)

El alfabeto utilizado en la codificación es el binario.

Por otro lado, el conocimiento de las buenas combinaciones de neuronas es mantenido y explotado en la población de blueprints a partir de los cuales se construyen los fenotipos. Los cromosomas de esta población están constituidos por un arreglo de longitud L (donde L es la cantidad de neuronas ocultas que tendrá la red resultante) de punteros a

cromosomas de neuronas en la otra población. Se utilizan números reales para la codificación genética (Figura 6).

El algoritmo evolutivo trabaja construyendo las redes neuronales a partir de cada cromosoma blueprint. El fitness de cada red, obtenido por el desempeño del arreglo neuronal en el que participa, es asignado al cromosoma blueprint. El fitness de los individuos de la población de neuronas se calcula como la suma de los cinco mejores fitness obtenidos en las redes en las que hayan participado. Se obtiene la próxima generación en la población de neuronas y luego la próxima generación en la población de blueprints.

SANE utiliza una estrategia de selección y reemplazo fuertemente elitista en ambas poblaciones. La mitad mejor ranqueada de la población pasa a la próxima generación. El mejor cuarto es seleccionado para reproducirse completando con su descendencia la mitad restante (Figura 7).

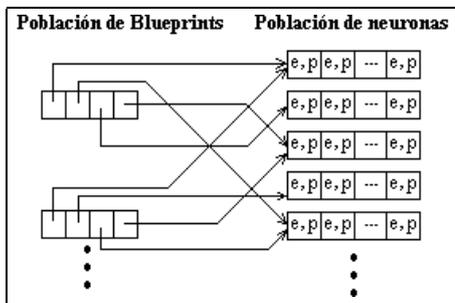


Figura 6

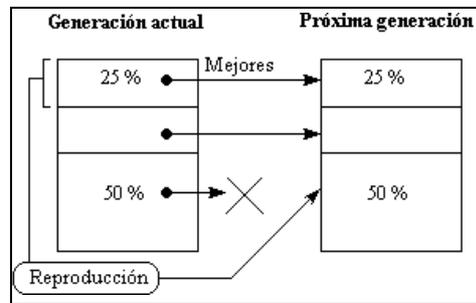


Figura 7

En cuanto a la reproducción, se utiliza crossover de un punto para conseguir el primer descendiente mientras que el segundo se obtiene por copia de uno de los parentales. En la población de neuronas se aplica mutación binaria con probabilidad 0.001 por bit. En la población de blueprints se aplican dos tipos de mutaciones: i) cambio de puntero a otra unidad de la población de neuronas elegida al azar con probabilidad 0.01 y ii) cambio de puntero a un descendiente de la unidad apuntada con probabilidad 0.5.

5.2.2. Extensiones realizadas a SANE:

Cambiando la codificación genética en la población de neuronas se extiende SANE para soportar redes con término de tendencia y evolución de función de transferencia. Se antepone a la secuencia de conexiones que definen una neurona los campos correspondientes al tipo de sigmoide TS y el peso de la conexión de tendencia p_0 (Figura 8).

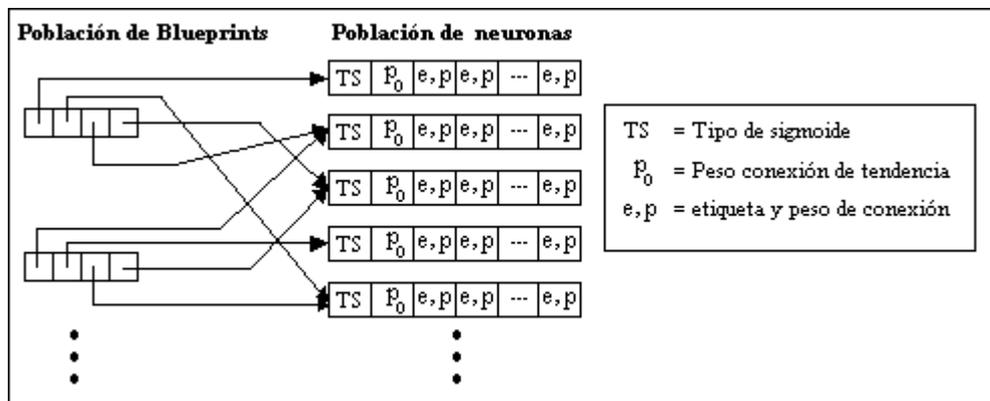


Figura 8

Para soportar redes neuronales recurrentes con esquema libre de conexión se codifican en los blueprints tanto las neuronas ocultas como las de salida. (Figura 9).

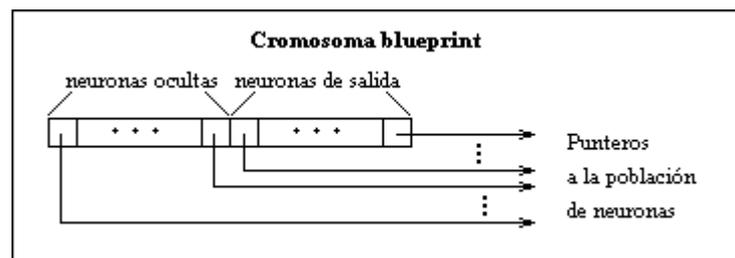


Figura 9

Además, es necesario modificar la interpretación de la etiqueta del par (e, p) en el momento de la construcción del

fenotipo. Numerando todas las neuronas a partir de cero, colocando primero las de entrada, luego las ocultas y por último las neuronas de salida, el campo e se interpreta de la siguiente manera:

Sean n_E , n_O y n_S el número de neuronas de entrada, ocultas y de salida respectivamente, y sea D el valor decimal del campo e :

Si $D > 127 \rightarrow$ la conexión es hacia la neurona numerada $(D \bmod (n_O + n_S)) + n_E$

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada $(D \bmod n_E)$

La figura 10 muestra el genotipo correspondiente a una RNA recurrente con 5 neuronas de entrada, 2 ocultas y 2 de salida. Por simplicidad se asume sólo dos conexiones por neurona, sin término de tendencia ni evolución de función de transferencia. Cada blueprint se forma por cuatro punteros a individuos de la población de unidades.

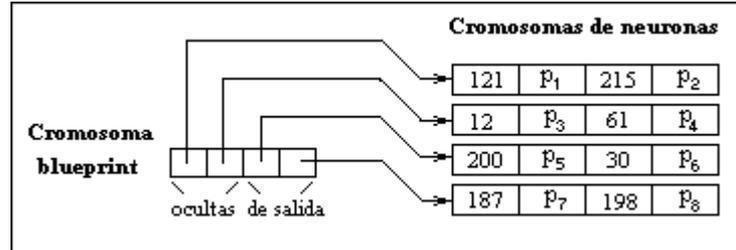


Figura 10

La decodificación del fenotipo puede verse en la figura 11 y requiere los cálculos que se detallan a continuación:

Cálculo	Conexión
$(121 \bmod 5) = 1$	Conexión desde la neurona 1
$(215 \bmod 4) + 5 = 8$	Conexión hacia la neurona 8
$(12 \bmod 5) = 2$	Conexión desde la neurona 2
$(61 \bmod 5) = 1$	Conexión desde la neurona 1
$(200 \bmod 4) + 5 = 5$	Conexión hacia la neurona 5
$(30 \bmod 5) = 0$	Conexión desde la neurona 0
$(187 \bmod 4) + 5 = 8$	Conexión hacia la neurona 8
$(198 \bmod 4) + 5 = 7$	Conexión hacia la neurona 7

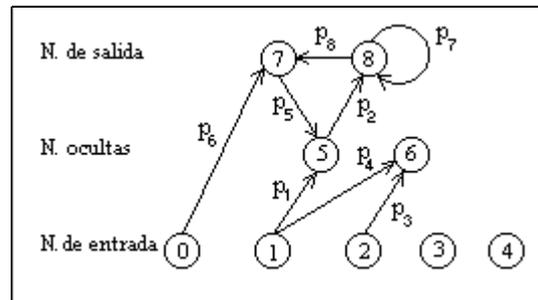


Figura 11

En este ejemplo se ve claramente la importancia de la cantidad de conexiones codificadas por neurona. Para simplificar el gráfico se ha elegido sólo 2, pero tal escasez de conexiones, si bien arroja como resultados redes simples (característica deseable), aumenta el riesgo de dejar neuronas de entrada desconectadas.

6. Resultados obtenidos

El rendimiento de ANE fue medido y comparado con SANE extendido, sobre diversos escenarios de distintas complejidades (figura 12).

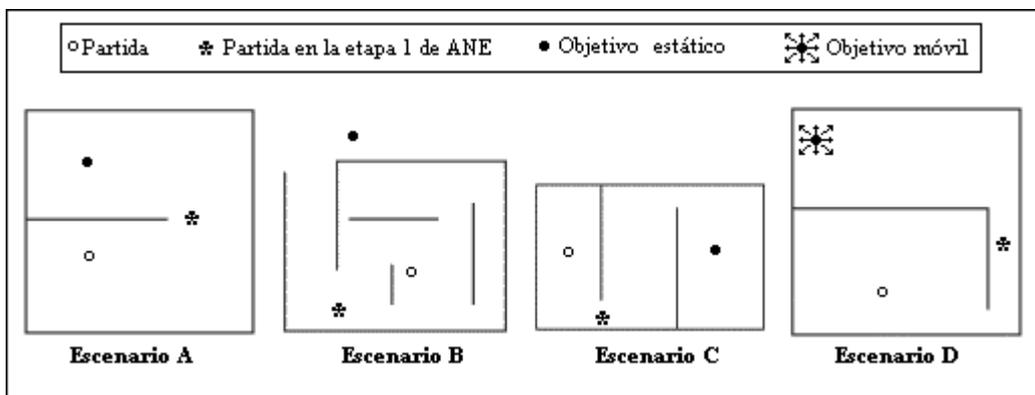


Figura 12

De acuerdo con los autores de SANE ([8], [9], [10]), este método ha demostrado las ventajas de la coevolución cooperativa en la búsqueda de soluciones a problemas de control, evidenciándose superior a las estrategias tradicionales que codifican, en un solo cromosoma, una red neuronal completa. Por ello se ha elegido como

referente comparativo.

El escenario C resulta especialmente difícil de sortear para los agentes debido a su limitada “visión” de los obstáculos. Obsérvese que existen posiciones donde la información sensada desde el exterior es la misma y sin embargo el movimiento que deben realizar es totalmente distinto (figura 13)

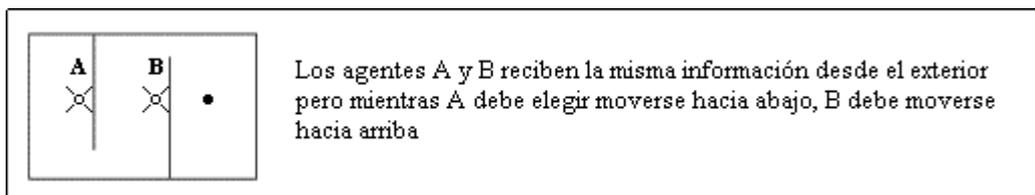


Figura 13

Se han realizado 30 evoluciones sobre cada escenario para cada método neuroevolutivo. De esta forma, es posible obtener un promedio aceptable del mejor fitness conseguido, en cada generación de la evolución, para cada método.

La cantidad de evoluciones exitosas, de entre las 30 realizadas, donde se alcanza efectivamente el objetivo, también es un dato importante que caracteriza el rendimiento de los métodos testeados. Con este último, es posible estimar la probabilidad de éxito al correr una evolución utilizando SANE o ANE.

En todas las pruebas realizadas se evolucionaron redes recurrentes con 14 neuronas de entrada, 2 de salida, 6 neuronas ocultas, conexión de tendencia y evolución de función de transferencia. Se utilizaron poblaciones de 80 agentes (80 blueprints) y 640 neuronas. Se codificaron el tipo de sigmoide (2 bits), el peso de la conexión de tendencia (16 bits) y 18 conexiones por neurona. Cada conexión se codificó con 8 bits para la etiqueta y 16 bits para el peso. Las evoluciones se prolongaron por 75 generaciones (incluye ambas etapas en el método ANE) salvo en el escenario A donde fueron suficiente 60 generaciones.

Para la primera etapa del método ANE se utilizaron tan sólo 3 generaciones en el escenario A, 20 generaciones en el escenario B, 30 generaciones en el escenario C, y 20 generaciones en el escenario D. Estas diferencias responden a las distintas complejidades que representa cada escenario pero no se descarta la utilización de otros valores. En la comparación con el método SANE debe prestarse atención a los datos recogidos a partir del comienzo de la segunda etapa de ANE, ya que las primeras generaciones los datos obtenidos están en función del subobjetivo y no del objetivo global.

Analizando los resultados obtenidos, ANE ha mostrado ser superior a SANE en todos los escenarios testeados. Sólo en el escenario A, de baja complejidad, puede decirse que SANE alcanza el rendimiento de ANE en las últimas generaciones de la evolución. A medida que aumenta la complejidad del problema a resolver (escenarios B, C y D), crece la brecha de rendimiento que separa ANE de SANE. Este hecho muestra la importancia de aplicar ANE, basado en una estrategia de neuroevolución incremental, para la resolución de tareas complejas (ver figura 14).

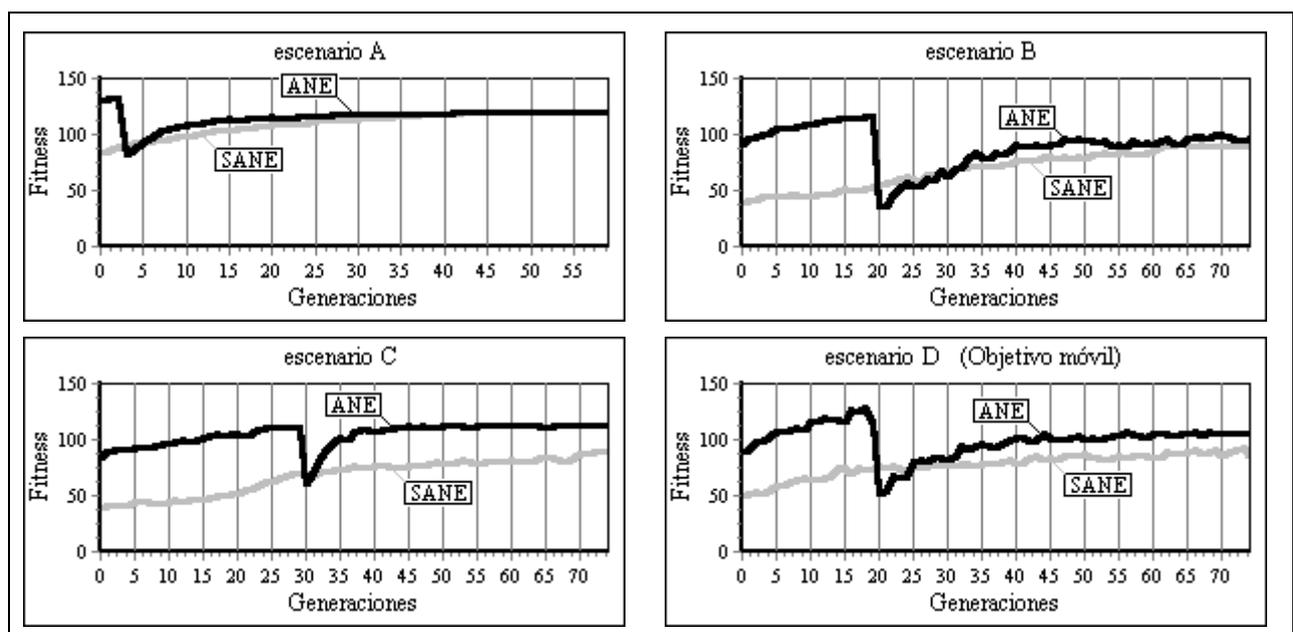


Figura 14: Fitness alcanzado por los métodos ANE y SANE.

Los datos recolectados acerca de la cantidad de evoluciones exitosas (en la que el objetivo es efectivamente alcanzado) sobre las 30 realizadas, también favorecen ampliamente al método ANE mostrando una relación similar a la del fitness promedio (ver figura 15). Esto implica que existe mayor probabilidad de alcanzar el objetivo por medio de una evolución de arreglos neuronales que por medio de SANE.

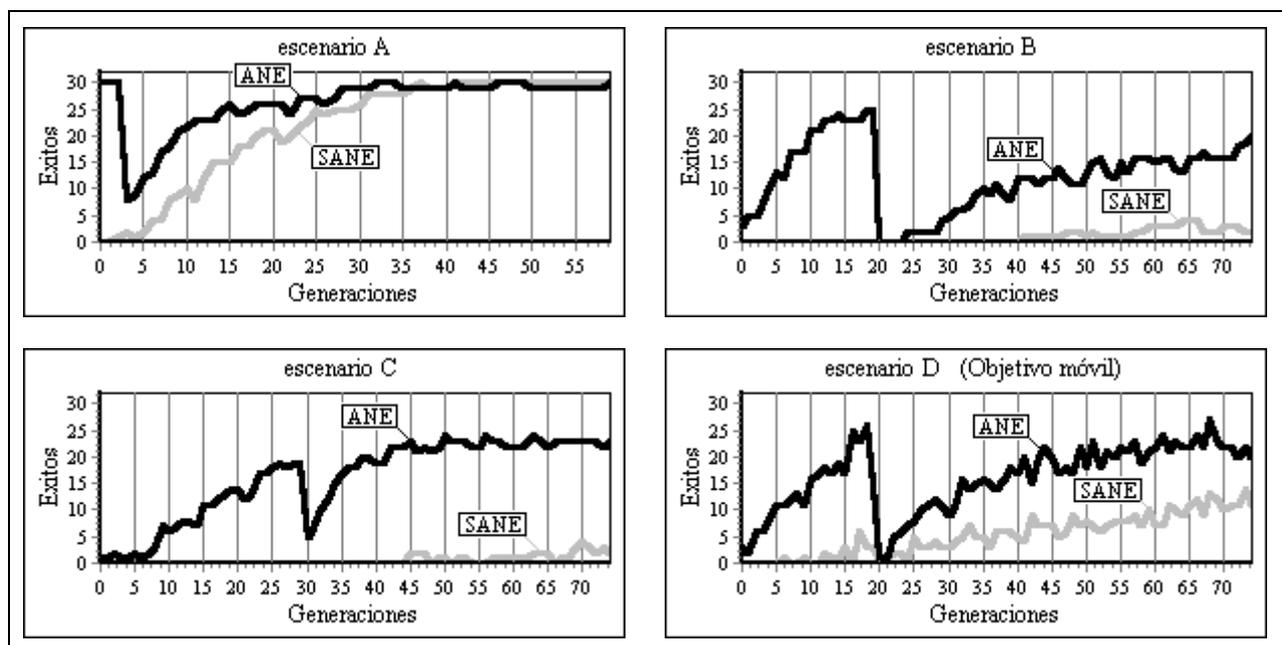


Figura 15: Cantidad de evoluciones exitosas de los métodos ANE y SANE.

7. Conclusiones y líneas de trabajo futuras

Se ha presentado un nuevo método para implementar evolución incremental basado en arreglos neuronales evolutivos (ANE).

Los resultados obtenidos aplicando ANE en la resolución de los problemas de alcance de objetivos y evasión de obstáculos han sido superiores a los logrados utilizando SANE extendido. Esto último indica claramente la utilidad de este nuevo método ya que la representación aplicada en ANE se basa en una extensión del método de Moriarty.

Queda pendiente la comparación de los ANE con otras soluciones existentes basadas en evolución incremental como por ejemplo [5]. Esto se debe a que el problema resuelto por Gómez no brinda la posibilidad de alcanzar objetivos conjuntamente con la evasión de obstáculos, y por lo tanto es necesario realizar algunas modificaciones a fin de efectuar las mediciones correspondientes.

Durante la aplicación de ANE a los problemas de alcance de objetivos y captura de presa se ha observado, en reiteradas oportunidades, que poseen capacidad de corregir los efectos de una etapa previa deficiente. La figura 16 ejemplifica una evolución en la cual el primer subobjetivo no ha sido alcanzado. Recuérdese que sobre el escenario C se realizaron 30 generaciones para la etapa 1. En esta dirección, se propone estudiar este comportamiento de una manera más general.

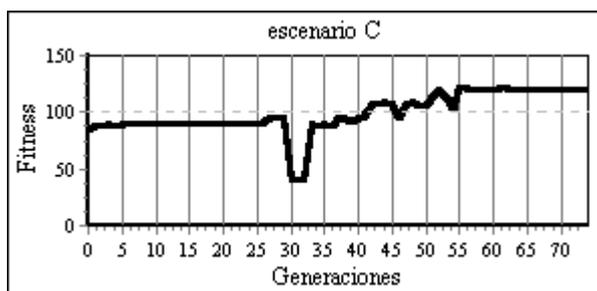


Figura 16

Por último, una de las grandes deficiencias de la evolución incremental es la necesidad de establecer *a priori* los subobjetivos necesarios para resolver el problema. Por este motivo, resulta de fundamental interés analizar la

posibilidad de segmentar estas etapas de la manera más automática posible.

Toda la documentación de este trabajo junto con el ambiente desarrollado ad-hoc para realizar la simulación y medidas aquí mencionadas, se encuentran disponibles en el LIDI (Laboratorio de Investigación y Desarrollo en Informática).

Referencias

- [1] Bruce, J. and Miikkulainen, R. Evolving Populations Of Expert Neural Networks. Department of Computer Sciences, The University of Texas at Austin. *Proceedings of the Genetic and Evolutionary Computation Conference*. (GECCO-2001, San Francisco, CA), (2001), pp. 251--257.
- [2] Corbalán, L., Pisano, M., Osella Massa, G. y Lanzarini L. Criaturas Virtuales especificadas a través de Redes Neuronales Evolutivas. *VII Congreso Argentino de Ciencias de la Computación*. Argentina, Vol. 2, (Octubre 2001), pp. 1105-1115.
- [3] Freeman, J. A. & Skapura, D. M. *Redes neuronales Algoritmos, aplicaciones y técnicas de programación*. Addison-Wesley, 1991. Versión en español de: Rafael García -Bermejo Giner. Addison-Wesley Iberoamericana 1993.
- [4] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989. pág. 10
- [5] Gomez, F. and Miikkulainen, R. Incremental Evolution Of Complex General Behavior Department of Computer Sciences, The University of Texas at Austin. *Adaptive Behavior*. Vol 5, (1997), pp.317-342.
- [6] Gomez, F. and Miikkulainen, R. Solving non-markovian Control Tasks with Neuroevolution Department of Computer Sciences, The University of Texas at Austin. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99, Stockholm, Sweden)*, San Francisco, CA: Kaufmann. (1999), pp. 1356-1361.
- [7] Han Yong, Ch. and Miikkulainen, R. Cooperative Coevolution of Multi-Agent Systmes *Department of Computer Sciences, The University of Texas at Austin. Technical Report AI-01-287* (Februaury 2001).
- [8] Moriarty, D. E. & Miikkulainen, R. Efficient Reinforcement Learning through Symbiotic Evolution. Department of Computer Sciences, The University of Texas at Austin. Austin, TX 78712. *Machine Learning*, Vol. 22, (1996), pp.11-33.
- [9] Moriarty, D. E. Symbiotic Evolution of Neural Networks in Sequential Decision Tasks. *Department of Computer Sciences, The University of Texas at Austin Ph.D. Dissertation. Technical Report AI97-257*, January 1997.
- [10] Moriarty, D. E. & Miikkulainen, R. Forming Neural Networks Through Efficient and Adaptive Coevolution. Information Sciences Institute, University of Southern California. Department of Computer Sciences, The University of Texas at Austin. *IEEE Transactions on Evolutionary Computation*. Vol.5, (1997), pp.373-399.
- [11] Pérez Bergquist, A. S. Applying ESP and Region Specialists to Neuro-Evolution For Go *Department of Computer Sciences, The University of Texas at Austin. Honors Thesis, Technical Report CSTR01-24* (May 2001).
- [12] Tapia, R. *Las Células de la Mente*. Impresora y Encuadernadora Progreso, S. A. de C. V. (IEPSA), Calz. de San Lorenzo, 244; 09830 México, D.F. Diciembre de 1996.
- [13] Yao, X. and Liu, Y. Ensemble Structure of Evolutionary Artificial Neural networks. *Computational intelligence Group, School of Computer Scienie University College*. Australian Defence Force Academy, Canberra, ACT, Australia 2600. 1996.
- [14] Yao, X. and Liu, Y. A New Evolutionary System for Evolving Artificial Neural Networks. *IEEE Transactions on Neural Networks*, Vol 8, nro. 3, pp 694-713, 1997
- [15] Yao, X. Evolving Artificial Neural networks. School of Computer Scienie The University of Birmingham Edgbaston, Birmingham B15 2TT. *Proceedings of the IEEE*. Vol.87, No.9, (September 1999), pp.1423-1447.