

UNIVERSIDAD: Universidad Nacional de La Plata.

NUCLEO DISCIPLINARIO/COMITÉ ACADÉMICO/OTROS TEMAS: Ingeniería Mecánica y de la Producción.

TÍTULO DEL TRABAJO: **CONTROL DE ROBOTS AUTÓNOMOS A PARTIR DE LA EVOLUCIÓN CONTINUA DE MÓDULOS NEURONALES**

AUTOR(ES): Hernán Vinuesa, Germán Osella Massa

DIRECTOR: Lic. Laura Lanzarini

CORREOS ELECTRÓNICOS DE LOS AUTORES: {hvinuesa, gosella}@lidi.info.unlp.edu.ar

PALABRAS CLAVES: Robótica Evolutiva, Control, Procesos Adaptativos

1. INTRODUCCIÓN

La Robótica Evolutiva (RE) es el área de la robótica autónoma en la que se desarrollan controladores para robots utilizando algoritmos evolutivos generalmente aplicados a redes neuronales. Este tipo de estructura es elegida básicamente por dos motivos: en primer lugar porque satisface la necesidad que la RE tiene por comprender la manera en que el ser humano aprende a lo largo de su vida y en segundo lugar porque posee la característica de ser fácilmente representable dentro de un algoritmo evolutivo [Wall06].

Aunque el desarrollo de controladores utilizando algoritmos evolutivos se ve ampliamente beneficiado por su capacidad de adaptación al entorno, en su gran mayoría su aplicación se concentra en la fase de generación del controlador no permitiendo realizar adaptaciones con posterioridad. Esto perjudica la aplicación del controlador en el mundo real ya que en general se trabaja con entornos dinámicos, a los que el robot debe adaptarse a fin de poder realizar su tarea correctamente en tiempo y forma.

Este artículo propone extender la evolución del controlador a lo largo de su vida útil combinando un método basado en evolución de módulos neuronales con un algoritmo evolutivo específico. El primer método es utilizado para producir los controladores mientras que el segundo ajusta al controlador durante su funcionamiento. Como resultado se obtiene un controlador basado en una red neuronal que posee una arquitectura mínima y capacidad de adaptación en la fase de ejecución.

Esta tarea se divide en dos etapas. La primera permite obtener un conjunto de controladores iniciales a partir de la información del entorno actual y la segunda aplica un algoritmo evolutivo específico sobre un subconjunto reducido de estos controladores, permitiendo de esta forma que el robot se adapte a los cambios.

Los controladores producidos por la primera etapa consisten de una única estructura obtenida de la combinación de módulos neuronales simples e independientes, que han sido evolucionados previamente. Esto busca reducir el tiempo de entrenamiento necesario para obtener un buen desempeño.

Si el ambiente en el que se desempeñará el robot no sufriera modificaciones, el mejor de este conjunto de controladores sería suficiente para resolver el problema ya que su interacción con el entorno estará contemplada en la información suministrada durante el entrenamiento. Sin embargo, en el mundo real, se producen modificaciones, por ejemplo los obstáculos cambian de lugar o el objetivo a alcanzar modifica sus características externas. Esto lleva a realizar ajustes sobre el controlador en la etapa de uso. Ese es el objetivo de la

segunda etapa, ajustar la interacción del robot con el medio. Para ello se utilizará una pequeña población formada por los tres mejores controladores obtenidos en la primera etapa y se aplicará sobre ellos un segundo algoritmo evolutivo diseñado para ejecutarse sobre el robot.

2. OBJETIVOS

El objetivo principal del presente trabajo consiste en obtener un controlador basado en una red neuronal, con una arquitectura mínima y capacidad de adaptación en la fase de ejecución. También se busca extender la evolución del controlador a lo largo de su vida útil combinando un método basado en evolución de módulos neuronales con un algoritmo evolutivo específico. El primer método es utilizado para producir los controladores mientras que el segundo ajusta al controlador durante su funcionamiento.

3. METODOLOGÍA

Algoritmo Evolutivo de la Primera Etapa

Para resolver problemas complejos se han utilizados diferentes enfoques basados en la descomposición del problema original en partes más sencillas. Si bien los métodos existentes varían en la forma de adquirir el conocimiento, la mayoría de ellos adopta una estructura basada en la evolución y combinación de diferentes módulos.

En esta dirección se han desarrollado distintas soluciones que combinan técnicas de Evolución Incremental con Redes Neuronales Evolutivas con el objetivo de proveer un mecanismo adaptativo que minimice el conocimiento previo necesario para obtener un buen desempeño dando lugar a controladores formados por varias redes [Bru01]. Otro aspecto a tener en cuenta es la forma de determinar cual es la red neuronal que debe ejecutarse en cada instante de tiempo [Yao96][Yao99]; en esta línea existen diferentes alternativas que van desde el uso de un árbol de decisión diseñado ad-hoc [Oli05] hasta mecanismos que organizan en forma automática la estructura [Cor04].

Si cada una de estas tareas simples es resuelta con éxito en forma independiente, debería ser posible combinar dichas soluciones para resolver la tarea compleja. Sobre esta premisa es que se utilizó en este artículo una extensión al método NEAT [Sta01][Sta02], incorporando el concepto de composición de módulos [Ose06].

La propuesta asume que se dispone de un conjunto de redes neuronales donde cada una de ellas, denominada módulo, es capaz de resolver una de las tareas simples. El

objetivo de esta etapa será conseguir una **red neuronal unificada**, constituida por la combinación de todos estos módulos, que sea capaz de resolver la tarea compleja.

Para más información sobre el algoritmo evolutivo utilizado en esta etapa vea [Ose06].

Algoritmo Evolutivo de la Segunda Etapa

Esta fase de la evolución se desarrolla sobre el robot físico y puede tomarse como una etapa infinita o de larga duración ya que, en teoría, nunca debería terminar. Dentro del robot se mantiene una pequeña población de tres individuos que provienen de la primera etapa. La elección de los individuos que componen esta población inicial puede realizarse de diferentes maneras. Una estrategia de selección puede ser tomar los tres individuos con mejor desempeño de la población como candidatos. Otra alternativa sería elegir el mejor individuo de cada una de las tres mejores especies. En este trabajo se optó por la primera opción. Los individuos seleccionados contendrán el material genético que permitirá adaptar el comportamiento del robot ante los cambios del entorno.

Dicho proceso de adaptación consiste en realizar pequeñas modificaciones sobre los controladores, causando cambios en su comportamiento. Para ello, en cada generación se evalúan los tres controladores. A partir de los dos individuos con el mejor desempeño se obtendrá un nuevo controlador que reemplazará al peor individuo de la población original. De esta manera se asegura que nunca se pierda al mejor individuo.

El nuevo controlador se obtiene de aplicar al mejor individuo un operador de recombinación lineal extendida en la dirección del segundo mejor controlador. Este operador ha sido seleccionado en base a las mediciones realizadas en [Vin07]. Sean P1 y P2 los controladores seleccionados como padres y H el nuevo controlador que se obtiene a partir de ellos utilizando recombinación lineal extendida.

$$H_i = P_{li} + \frac{s_i \cdot r_i \cdot a \cdot (P_{2i} - P_{1i})}{\|P_1 - P_2\|}$$

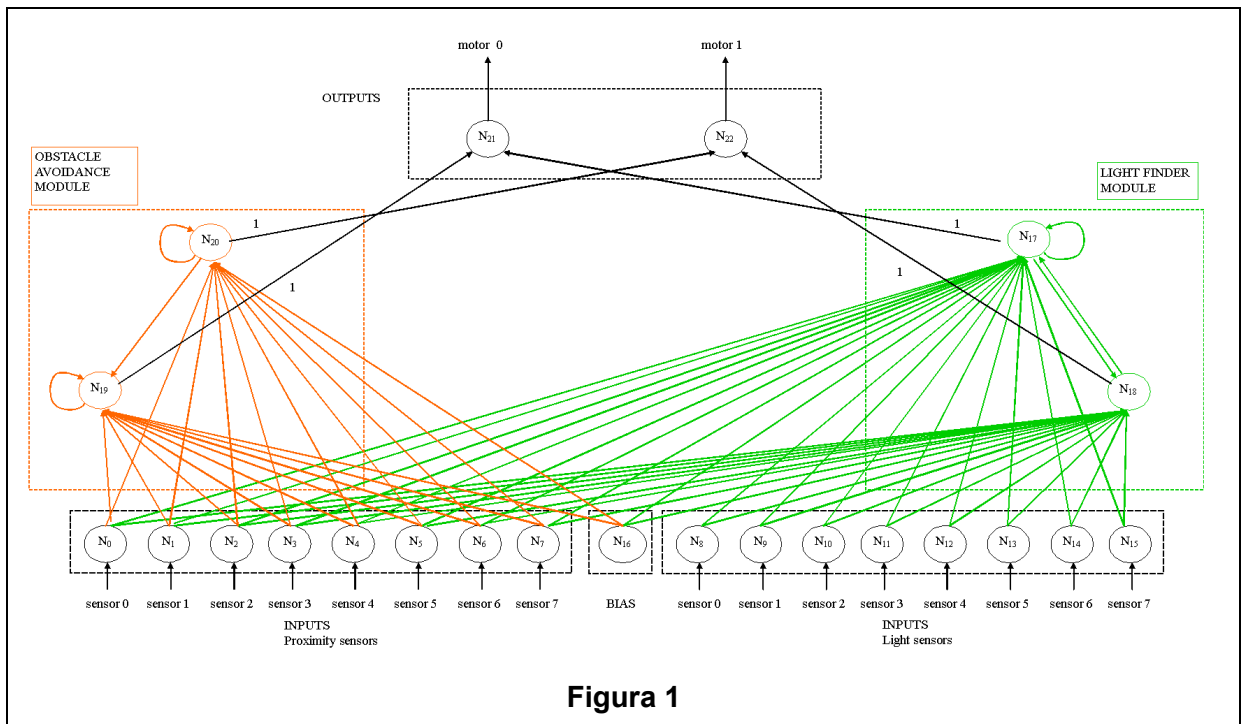
Dónde el subíndice $i \in [1, N]$ hace referencia a cada uno de los pesos de una red neuronal; a determina el tamaño del paso a utilizar y su valor se calcula como 2^{-ku} . k especifica la precisión de ese paso, tomando un valor entre 4 y 20 como se sugiere en [Poh05]. $u \in [0,1]$ es un número aleatorio distribuido uniformemente. $r_i \in [-0.5, 0.5]$ es un número aleatorio distribuido uniformemente que representa la máxima variación que puede ocurrir en el i -ésimo peso entre padre e hijo. s_i puede valer -1 ó +1, aleatoriamente. s es el sentido del paso.

Dentro del área determinada a partir de los padres seleccionados los hijos no son generados de manera uniforme: La probabilidad de crear hijos cerca de alguno de sus padres es alta y solamente con baja probabilidad los hijos son creados lejos de ellos. Para más detalles de este operador vea [Poh05].

4. RESULTADOS

La estrategia propuesta en este artículo ha sido utilizada para desarrollar un controlador basado en redes neuronales evolutivas que permita dotar a un robot Khepera II de la habilidad de esquivar obstáculos y alcanzar objetivos. Dicho controlador debe tener la capacidad de adaptarse a modificaciones del entorno una vez que se encuentre en su etapa de uso.

Dado que se han utilizado únicamente los sensores de luz y proximidad del robot, los objetivos buscados estarán representados por luces ubicadas en cualquier posición dentro de su entorno de navegación. Se espera lograr que el robot se desplace libremente, sin colisionar con los obstáculos hasta acercarse a una zona iluminada donde deberá intentar permanecer.



La población inicial de controladores fue generada de acuerdo a lo descrito para el algoritmo evolutivo de la primera etapa. Primero se combinaron dos módulos obtenidos de manera independiente: uno que permite esquivar obstáculos y otro que permite alcanzar la luz más próxima. Cada uno de estos módulos se encuentra representado por una red

neuronal recurrente obtenida a través del método NEAT convencional.

La figura 1 corresponde a la arquitectura inicial. En ella puede observarse que los módulos comparten la información suministrada por los sensores de proximidad.

La red neuronal unificada combina la salida de cada uno de los módulos a través de dos neuronas nuevas que serán las encargadas de comandar realmente los motores del robot. Por lo tanto, queda formada una red con 16 entradas, 8 para cada uno de los sensores de proximidad del robot, 8 para cada uno de los sensores de luz y dos neuronas de salida, una para cada uno de los motores. La población inicial está formada por redes cuyos genomas se obtienen de realizar pequeñas modificaciones a esta estructura inicial. Esta población se evoluciona según el método NEAT con las restricciones impuestas a los operadores genéticos.

En cada caso, el fitness de cada elemento estará dado por su capacidad para combinar los módulos originales. La expresión utilizada en este caso es la siguiente:

$$fitness = coef_obs * Fit_{Obs} + coef_lig * Fit_{Lig}$$

donde Fit_{Obs} y Fit_{Lig} corresponden a los fitness del individuo respecto de la evasión de obstáculos y el alcance de luz respectivamente. $coef_obs$ y $coef_lig$ son constantes que permiten escalar las respuestas de cada módulo a fin de que intervengan adecuadamente en la ponderación del fitness de la red unificada. En esta solución específica se utilizaron los valores 1 y 1.3 respectivamente con el fin realizar un mayor énfasis en el alcance de la luz.

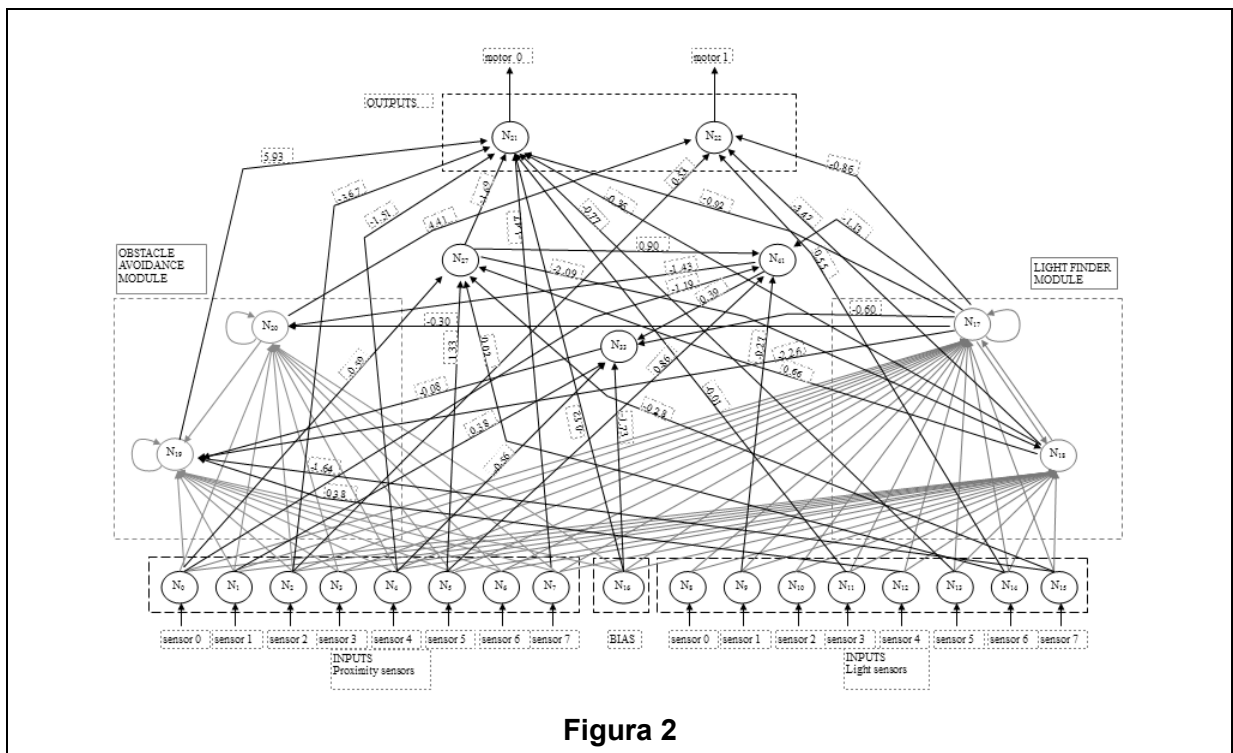


Figura 2

La figura 2 representa la red unificada con mejor desempeño obtenida como resultado.

Los tres mejores individuos obtenidos como resultado de la etapa evolutiva inicial fueron seleccionados para constituir la primera población de controladores encargada de comandar al robot Khepera II y sobre quienes se aplicará el algoritmo evolutivo de la segunda etapa en forma continua.

El tamaño extremadamente reducido de esta población busca minimizar el tiempo de cómputo necesario para evaluar la aptitud de cada uno de ellos al desempeñar la tarea.

Cada uno de los tres individuos es evaluado sobre el robot real en forma independiente de los otros dos. Una vez concluida la evaluación de la población, se ordena a los individuos de acuerdo a su valor de fitness y se seleccionan los dos mejores para realizar la operación de la recombinación lineal extendida. Como resultado de esta operación se obtiene un nuevo controlador basado en el mejor padre y en dirección del segundo mejor padre. Este nuevo individuo será el reemplazante del peor individuo de la población.

La frecuencia en que este proceso deberá realizarse dependerá de la tasa de cambio que tenga el entorno en el que se mueve el robot, por lo tanto, puede no ser necesario en todas las generaciones. Igualmente es recomendable aplicarlo en las primeras generaciones de la etapa del uso del controlador para permitir a los individuos adaptar el comportamiento aprendido en el simulador al mundo real.

También se realizaron distintas pruebas para analizar la capacidad de adaptación del robot ante cambios del ambiente. La figura 3 muestra una comparación generación a generación entre los fitness promedio de dos robots diferentes: A uno se le aplica el algoritmo evolutivo de la segunda epata (línea sólida) y al otro no (línea punteada). En la generación 60 se introduce un cambio en el entorno que se mantiene hasta la generación 225.

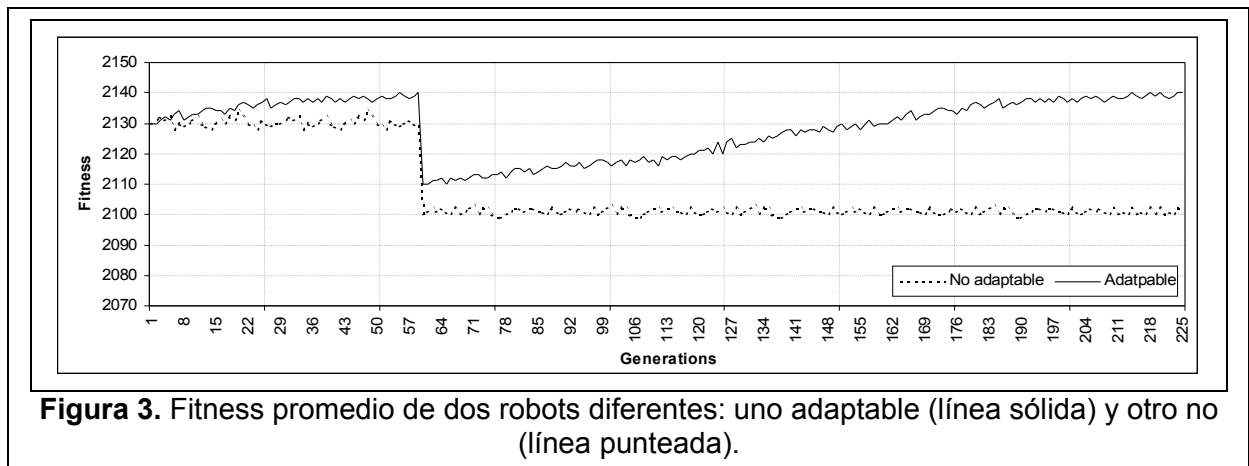
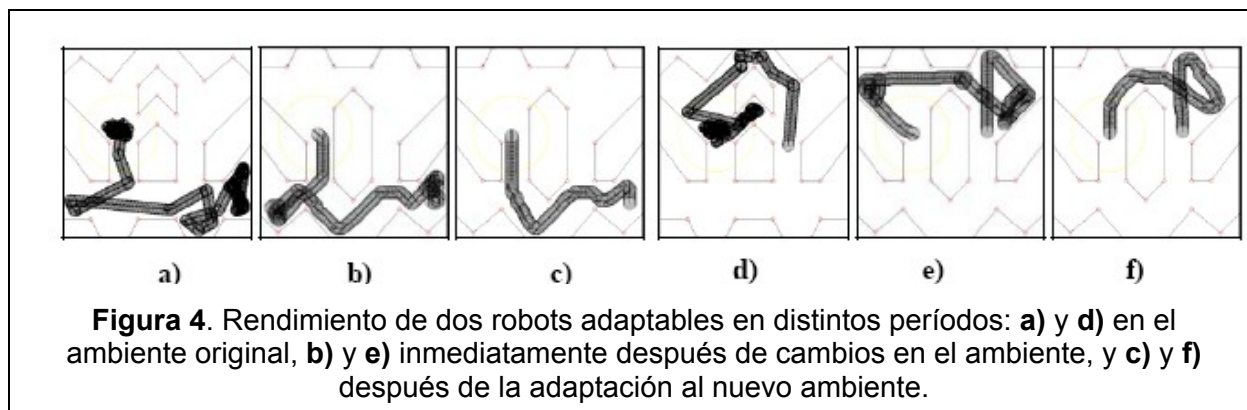


Figura 3. Fitness promedio de dos robots diferentes: uno adaptable (línea sólida) y otro no (línea punteada).

En las primeras 60 generaciones, se puede observar una mejora en el fitness del robot que realiza una adaptación al entorno comparado con el que no la hace. Las pequeñas variaciones en el fitness se deben a diferencias en las condiciones con las que se evaluó al controlador. Esto justifica la variación en el fitness del robot que no realiza ajustes.



En la generación 60, las condiciones en el entorno cambian y el fitness de ambos controladores cae inmediatamente. Esto se debe a que ninguno de los controladores fue producido para las condiciones del nuevo entorno. Sin embargo, a medida que el algoritmo evolutivo de la segunda etapa se ejecuta sobre el robot adaptable, puede observarse una mejora en el fitness del robot. Después de 200 generaciones, el fitness de ese robot termina siendo tan bueno como era antes de los cambios del entorno. Por otro lado, el robot no adaptable no mejoró después del cambio, como era lógico.

5. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

La obtención de controladores utilizando herramientas adaptables al entorno resulta, en general, una tarea computacional costosa. Por lo tanto, suele separarse abruptamente la etapa de generación de las soluciones de la etapa de su uso. En esta dirección, este artículo utiliza una estrategia que reduce el tiempo de entrenamiento inicial, aprovechando el conocimiento almacenado en módulos con comportamientos simples previamente generados y permite al robot seguir adaptando su respuesta a futuros cambios del entorno. En otras palabras, se evoluciona un controlador a lo largo de su vida útil combinando un método basado en evolución de módulos neuronales con un algoritmo evolutiva específico.

Los resultados obtenidos en la resolución del problema de evasión de obstáculos y alcance de objetivos utilizando un robot Khepera II han demostrado que la estrategia propuesta permite adaptar el comportamiento del robot a los cambios del entorno.

REFERENCIAS

- [Bru01] Bruce, J. 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), pp. 251-257. 2001.
- [Cor04] Corbalán L., Osella Massa G., Lanzarini L., De Giusti A. ANELAR. Arreglos Neuronales Evolutivos de Longitud Adaptable Reducida. X Congreso Argentino de Ciencias de la Computación.CACIC 2004. Universidad Nacional de La Matanza. Bs.As. Argentina. ISBN 987-9495-58-6. October 2004.
- [Gol87] Goldberg, D.E., Richardson, J. Genetic algorithms with sharing for multimodal function optimization. pp. 148-154. 1987.
- [Oli05] Olivera J. Lanzarini, L. Cyclic Evolution. A new strategy for improving controllers obtained by layered evolution. VI Workshop de Agentes y Sistemas Inteligentes 2005. Concordia, Entre Ríos. Argentina. ISBN: 950-698-166-3. October 2005.
- [Ose06] Osella Massa G., Vinuesa H., Lanzarini L. Modular Creation of Neuronal Networks for Autonomous Robot Control. 3rd IEEE Latin American Robotics Symposium. LARS 2006. Chile. October 2006.
- [Poh05] Pohlheim H. Evolutionary Algorithms: Overview, Methods and Operators. GEATbx version 3.7. Genetic and Evolutionary Algorithm Toolbox for use with Matlab. pp: 28-29. November 2005.
- [Rad93] Radcliffe, N.J.: Genetic set recombination and its application to neural network topology optimization. Neural computing and applications 1. pp. 67-90. 1993.
- [Sta01] Stanley, K.O., Miikkulainen, R. Competitive coevolution through evolutionary complexification. Journal of Artificial Intelligence Research 21. 2001.
- [Sta02] Stanley, K.O., Miikkulainen, R. Evolving neural networks through augmenting topologies. Evolutionary Computation 10. pp. 99-127. 2002.
- [Vin07] Vinuesa, H. Lanzarini, L. Neural Networks Elitist Evolution. 29th Internacional Conference Information Technology Interfaces (ITI 2007). Dubrovnik. Croatia. 2007.
- [Wal06] Walker J. Garrett, S. Wilson, M. The Balance Between Initial Training and Lifelong Adaptation in Evolving Robot Controllers. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B: CYBERNETICS, VOL. 36, NO. 2. APRIL 2006.

[Yao96] Yao, X. Liu, Y. Ensemble Structure of Evolutionary Artificial Neural networks. Computational intelligence Group, School of Computer Science University College. Australian Defence Force Academy, Canberra, ACT, Australia 2600. 1996.

[Yao99] Yao, X. Evolving Artificial Neural networks. School of Computer Science The University of Birmingham Edgbaston, Birmingham B15 2TT. Proceedings of the IEEE. Vol.87, No.9, pp.1423-1447.September 1999.