

## ALGORITMO GENÉTICO PARA EVALUACIÓN DE RATONES EN UN LABERINTO

Franyelit Suárez<sup>1</sup>, Luis Mendoza<sup>2</sup>, Alan Druet<sup>1</sup>, Melvin Gorozabel<sup>1</sup>, Juan Segura<sup>3</sup>  
Franyelit.suarez@puces.edu.ec, mluisjacinto@hotmail.com, axdr17@gmail.com, gorozabel13@gmail.com

<sup>1</sup>Pontificia Universidad Católica del Ecuador-Sede Esmeraldas- Ecuador

<sup>2</sup>Corporación Nacional de Telecomunicaciones, Ecuador

<sup>3</sup>Universidad Tecnológica Indoamérica, Ecuador

---

**Resumen:** Se presenta un trabajo de análisis del comportamiento de ratones en un laberinto. Se realiza una comparación del comportamiento virtual a través de un algoritmo genético y del comportamiento real presentado en un laberinto en cruz real, probado con 33 roedores Wistar machos expuestas a condiciones de luz controlada. El trabajo consistió en un robot guía para la salida del laberinto, se utilizó una red neuronal backpropagation donde la entrada es la ubicación del ratón. Al realizar una simulación inteligente se espera evaluar habilidades neurobiológicas con medidas de tiempo real y determinación estimada de soluciones factibles. Este trabajo permitió comprobar los resultados obtenidos previamente en Matlab y optimizar la solución con nuevas alternativas de salida del laberinto, empleando además un nuevo método de seguimiento de caminos.

---

**Palabras Clave:** Laberinto de n canales, simulación inteligente, red neuronal Backpropagation.

---

**Abstract:** A paper analyzing the behavior of mice in a labyrinth is presented. A comparison of the virtual behavior is made through a genetic algorithm and the real behavior presented in a real cross labyrinth, tested with 33 male Wistar rodents exposed to controlled light conditions. The work consisted of a robot guide for the exit of the labyrinth, we used a neural network backpropagation where the input is the location of the mouse. By performing an intelligent simulation, it is expected to evaluate neurobiological skills with real-time measurements and estimated feasible solutions. This work allowed to verify the results previously obtained in Matlab and to optimize the solution with new alternatives of exit of the labyrinth, also employing a new method of track tracking.

---

**Key words:** Labyrinth of n channels, intelligent simulation, neural network Backpropagation.

## I. INTRODUCCIÓN

Thorndike [1] observó la conducta de animales y más adelante extendió sus estudios con personas. En sus investigaciones implantó el uso de métodos usados en las ciencias exactas para las situaciones educativas y formativas, al hacer énfasis en el tratamiento cuantitativo exacto de la información. Johcich [1,2] afirmaba que cualquier cosa que exista, debe existir en determinada cantidad y por lo tanto es medible. La teoría del conexionismo, se refiere a que aprender consiste en establecer conexiones entre estímulos y respuestas. Por otro lado la ley de efecto [1] consiste en la generación de respuestas contiguas serán asociadas al estímulo y tendrán mayor probabilidad de ocurrencia con la aparición del mismo estímulo; así cuando una conexión entre un estímulo y respuesta es recompensado, el vínculo o conexión se refuerza y cuando es castigado la conexión se debilita, considerándose estas dos como retroalimentación positiva y negativa respectivamente [3],[4].

Párraga et al [5] implementó una hormiga artificial con un agente computacional cuyo objetivo consistía en seguir un rastro de feromonas en un ambiente matricial. La hormiga consta de un sensor que le permite detectar si existe presencia de feromonas en la celda frente a ella, además contiene otro sensor para determinar si la celda ha sido visitada. Dependiendo de las condiciones de la celda, la hormiga tomará la decisión de avanzar una celda hacia delante, girar a la derecha o marcar la celda como ya visitada.

Jiménez y Cruz [6] desarrollaron un sistema robótico para fines didácticos, que realice la función de recorrido en un laberinto T, que fue capaz de ubicarse en todo el trayecto logrando encontrar la salida de manera eficiente.

El comportamiento exploratorio de ratas en Laberinto en Cruz Elevado (LCE) es utilizado en el estudio de trastornos de ansiedad generalizada. Salgado et al [7] evaluó la exploración de una rata en ambiente provista de dos zonas diferentes: una potencialmente aversiva (brazos abiertos) y otra segura (brazos cerrados). El desempeño de la rata es el resultado de una ponderación entre la motivación de explorar y la ansiedad que experimenta en algunos fragmentos del laberinto. De forma natural los roedores eligen ubicarse en rincones, y lugares oscuros. Esto explica por qué la rata prefiere los caminos cerrados que los abiertos.

En 1932 Tolman [8], en pleno auge del conductismo, realizó experimentos para evaluar el aprendizaje de laberintos por ratas y en ellos percibió la presencia de mapas cognitivos, esto era, representaciones internas del laberinto no asociadas con respuestas motoras aprendidas.

En este trabajo se desarrolló un sistema inteligente que es capaz de simular el comportamiento de un roedor en un laberinto en T cruzado. El sistema consta de un robot guía

que conduce al roedor por los brazos del laberinto. Las redes de comportamiento son un mecanismo de selección de acción no jerárquico, descentralizado y dinámicamente reconfigurable, compuesto por un conjunto de nodos los cuales representan entidades en un nivel más bajo que el del sistema y que corresponden con las acciones (conductas) que un agente puede exhibir. Los nodos, a los que llamaremos módulos de habilidad, están conectados por ligas definidas como: predecesoras, sucesoras y conflictivas. Estas conexiones establecen una dinámica de activación/inhibición entre los módulos. Cada módulo de habilidad tiene asociado un nivel de activación, el cual es un número real y un conjunto de condiciones que deben observarse verdaderas para que el módulo sea considerado ejecutable. Los módulos ejecutables que tienen un nivel de activación superior a cierto umbral son activados, de manera que solamente un módulo puede activarse a un mismo tiempo. Una vez que un módulo ha sido activado su nivel de activación se vuelve cero. Un módulo ejecutable propaga una cantidad de activación a sus sucesores, favoreciendo su ejecución y estableciendo de esta forma, secuencias de acciones. Un módulo que no es ejecutable propaga una cantidad de activación a sus predecesores, tratando de que sus condiciones que son momentáneamente falsas se vuelvan verdaderas al momento de activarse estos. Todos los módulos de habilidad influyen en el decremento del nivel de activación de los módulos con quienes presentan conflictos [9].

Desarrollo

### Algoritmo Wall Following

Este algoritmo se basa en la regla de que siguiendo siempre por la derecha o por la izquierda se encontrará invariablemente la salida de un laberinto [10]. El algoritmo se aplica de la siguiente manera: Basados en el seguidor de paredes por la izquierda se aplican los pasos en la secuencia que se enlista enseguida: 1) Revisa la presencia de la pared izquierda. 2) Si la pared izquierda está presente se hace la bandera  $flagl=1$ , si no esta presente, entonces se hace que el valor de la bandera tenga valor 0, esto es  $flagl=0$ . 3) Si  $flagl=1$  entonces se pasa al paso 4, caso contrario,  $flagl=0$  hacer un giro izquierdo de 90 grados. 4) Comprobar la presencia de la pared frontal. 5) Si la pared frontal está presente se hace que la bandera  $flagf=1$ , si no,  $flagf=0$ .

### Algoritmo Backtracker

Backtracking (o búsqueda atrás) es una técnica de programación para hacer búsqueda sistemática a través de todas las configuraciones posibles dentro de un espacio de búsqueda. Para lograr esto, los algoritmos de tipo backtracking construyen posibles soluciones candidatas de manera sistemática. En general, dado una solución

candidatas: 1) Verifican si  $s$  es solución. Si lo es, hacen algo con ella dependiendo del problema. 2) Construyen todas las posibles extensiones de  $s$ , e invocan recursivamente al algoritmo con todas ellas. A veces los algoritmos de tipo backtracking se usan para encontrar una solución, pero otras veces interesa que las revisen todas para encontrar la mejor de ellas [11].

### Búsqueda heurística

En Inteligencia Artificial los algoritmos de búsqueda son herramientas de gran utilidad para diferentes situaciones, ya que ofrecen una amplia gama de opciones para la solución de problemas diversos. Estos se conocen como algoritmos de búsquedas heurísticas y no heurísticas. Un algoritmo de búsqueda deberá buscar potencialmente entre  $b^d$  nodos antes de encontrar la solución. Siendo  $b$  las opciones en cada nodo y  $d$  corresponde al nodo objetivo, de tal manera que se reduce el factor de ramificación de  $b$ . Por tanto es posible encontrar una heurística  $h_2(n)$  es menor que alguna otra  $h_1(n)$ , así pues si el grado de dominio de  $h_2(n)$  es mayor que  $h_1(n)$ , puede asegurarse que  $h_2(n) \geq h_1(n)$ .

En Inteligencia Artificial existen muchos algoritmos heurísticos que por su propia naturaleza consta de reglas heurísticas para el procesamiento del lenguaje. Cualquiera de estas reglas pudiera generar errores de clasificación, pero en forma conjunta generan robustez en el sistema. Esto conduce a la credibilidad en el reconocimiento de patrones.

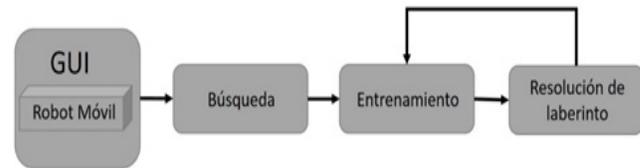
Recientemente se han realizado adaptaciones al término de heurística, definiéndola como una alternativa para solución de problemas, ya que está constituida como una técnica o procedimiento práctico. Por otro lado, Lakatos [12-19] define la heurística como una herramienta conformada por reglas, que sugieren cómo proceder a la hora de generar hipótesis, permitiendo la resolución de problemas haciendo uso de la creatividad y el pensamiento divergente.

### Métodos Computacionales

Para simular el comportamiento de un roedor en el laberinto cruzado, pueden utilizarse diferentes métodos inteligentes computacionales, entre los que se mencionan el algoritmo genético, las redes neuronales y la búsqueda heurística.

Un método muy común para realizar esta simulación consiste en colocar un robot en el laberinto o un guía, que motive al roedor a seguirlo, para realizar el recorrido de forma eficiente. Para realizar la búsqueda se realizó una red de backpropagation con algoritmos de búsqueda, asociados para que el roedor simulado siga al robot digital hasta la salida del laberinto [15-17]. La figura 1 muestra

la estructura del sistema desarrollado, donde es posible visualizar el comportamiento de la red y la búsqueda para encontrar el objetivo.



**Figura 1. Diagrama del sistema desarrollado.**

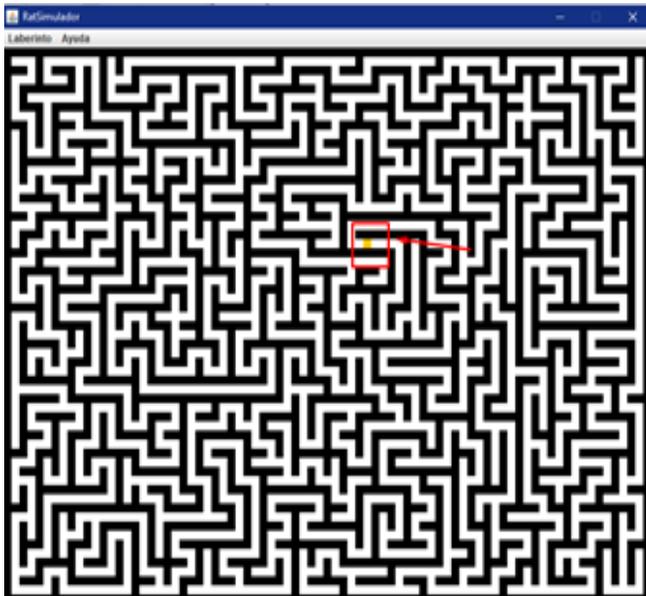
Interfaz de usuario: realiza la interacción entre los datos recolectados y el sistema, implicando en el proceso al usuario final.

Recolección de datos: se toman datos del entorno, según la ubicación del ratón y su posición dentro del laberinto. Los datos son cargados para luego ser asumidos en la simulación.

Búsqueda: consta del algoritmo heurístico de búsqueda, realizando a la vez el gráfico del recorrido en 2D, ubicando al objeto y al ratón en el entorno del laberinto. Determinando la ubicación y el tiempo de recorrido.

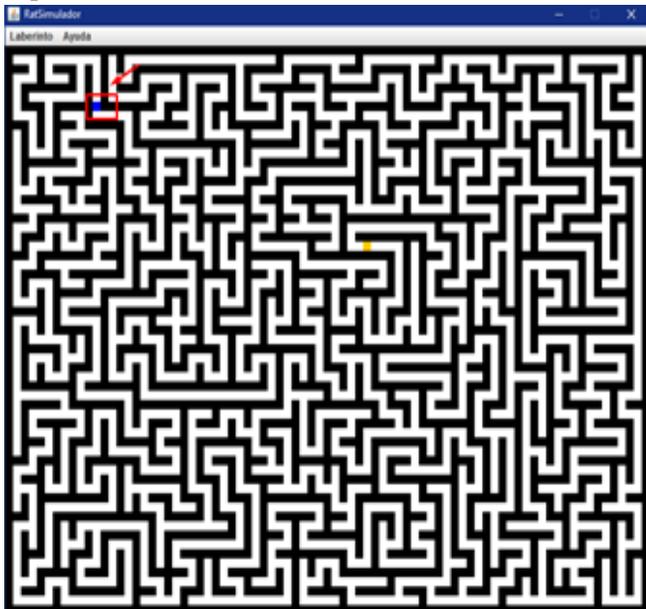
Entrenamiento: Se encuentran aquí los parámetros de entrada y de aprendizaje de la red, incluyendo el radio de la vecindad y el número de iteraciones. En el proceso se almacenan los pesos neuronales y se visualizan las sinapsis de la red, lo que servirá de información para el desplazamiento del ratón. La red realizará el entrenamiento cada vez que no encuentre la respuesta apropiada.

Desde la década de los ochenta se han desarrollado una serie de trabajos de búsqueda y solución de laberintos, donde se ha conseguido realizar simulaciones estables eficientes con diferentes tipos de roedores y con alguna aplicación de químicos en el sistema sanguíneo, observando que los ratones buscan la salida prefiriendo los espacios cerrados. En el caso de la simulación, los ratones siguen al guía de forma sistemática, si el camino parece cerrado el ratón buscará siempre una salida hacia la derecha o izquierda, teniendo preferencia en el que haya arrojado mejores resultados. La red neuronal recibe como entrada la ubicación del objetivo que puede visualizarse en la figura como el punto amarillo que se observa en la figura 2.



**Figura 2.** Laberinto con el punto guía, que conforma la entrada de la red neuronal.

La entrada también recibe la ubicación donde se encuentra el ratón que se puede apreciar como un punto de color azul en la figura 3, con ayuda de la neurona artificial el objeto de color azul recorre el laberinto simulando el comportamiento de la rata hasta llegar a su objetivo que es el punto amarillo.



**Figura 3.** Laberinto con el punto azul, entrada de la red neuronal, junto con el punto amarillo objetivo.

El objetivo se desplaza por el laberinto en busca del mejor camino posible, mientras el punto azul consiste en un elemento que realiza una búsqueda del punto amarillo. El sistema puede discriminar los caminos no aptos,

permitiendo la optimización del recorrido. Una vez que el punto azul encuentra al punto amarillo, el sistema es capaz de arrojar un mensaje de aviso del encuentro entre ambos puntos, y ofreciendo los parámetros de la ubicación del encuentro, así como el tiempo que tardó en encontrarse.

Una vez que el ratón encuentre al objetivo y sea conducido a la salida del laberinto, éste podrá reiniciarse para realizar una nueva búsqueda, de manera de más óptima.

Las redes neuronales artificiales han sido implementadas en una variedad de soluciones de laberinto, Callejas et al [13-18] realizó el diseño de un sistema de control mediante la utilización de Redes Neuronales Artificiales (RNA) con aprendizaje no supervisado tipo SOM (Self-Organizing Maps) en una Arreglo de Compuertas Reprogramable (FPGA) Spartan 6, con el propósito de determinar los efectos que tiene dicho sistema en el comportamiento de un robot móvil en la resolución de trayectorias tipo laberinto. Por otro lado, [14-15] señala que es posible realizar el recorrido de navegación por un laberinto empleando técnicas diferentes como las redes neuronales artificiales, el algoritmo genético y la lógica difusa, para lo cual fue posible observar que las redes neuronales de base radial resultaron las más apropiadas para el recorrido, ya que constan de una estructura con respuesta localizada [20].

En este trabajo se implementó una red del tipo supervisado, que consta de tres capas de neuronas con una función de transferencia lineal para encontrar la salida deseada. La función de salida empleada fue la función identidad con el fin de que la salida de la neurona de procesamiento sea idéntica a su nivel de activación.

Con (1) fue posible relacionar las variables asociadas a la red, siempre que se establezcan los parámetros respectivos del laberinto como objeto y ratón.

$$a = F_k(W_{ij} P_i + b_k) \quad (1)$$

Además se utiliza un algoritmo para la búsqueda del objetivo que es el backtracking, el cual consiste en ir proyectando soluciones de caminos posibles dentro de un espacio determinado que es en este caso el laberinto, una vez encuentra la solución o camino este lo almacena y lo usa nuevamente de ser necesario.

#### IV. CONCLUSIONES

La red neuronal tipo Backpropagation permite encontrar la solución del laberinto de forma óptima, siempre buscando la mejor forma posible de obtener un resultado eficiente. Esta red a la vez permite establecer una interacción con el usuario, que le permite verificar el avance del aprendizaje de la red y a la vez del ratón simulado en el mismo.

El sistema permite que el ratón encuentre al objetivo en

el menor tiempo posible, pudiendo llevar un registro de la ubicación de los mismos, logrando efectuar el recorrido completo después del entrenamiento de la red neuronal, reduciendo el tiempo aprendizaje un 12%.

Los pesos sinápticos presentan variaciones notables con el ajuste del aprendizaje del ratón, lo cual demuestra la evolución del entrenamiento y la toma de decisiones del ratón.

La herramienta GUI utilizada proporciona a los usuarios el estudio del comportamiento de la red de backpropagation, modificando los parámetros del laberinto y obteniendo a su vez la salida del mismo. Además esta interfaz permite verificar el comportamiento del ratón y del objetivo dentro del laberinto, visualizando el recorrido, la ubicación y el tiempo de búsqueda. Este proceso facilita y ofrece una herramienta didáctica de enseñanza de las redes neuronales artificiales.

## V. REFERENCIAS

- [1]Thorndike, E. (1898) Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, Vol 2(4), i-109
- [2]Jonassen, D. (1991) Objectivism versus constructivism: do we need a new philosophical paradigm? *Educational Technology Research and Development*, 39 (3), 5-14.
- [3]Mergel, B. (1998). Instructional design & Learning theory. *Rev. Learning Theories of Instructional Design*.
- [4]Watson, J. B. (1913). Psychology as the behaviorist views it. *Psychological Review*, 20(2), 158-177.
- [5]Párraga, J., Anzules, M., Ramírez, C., Santander, L. (2013) Implementación en C# de una Red Neuronal Artificial en el control locomotor de una hormiga robótica. *Revista Tecnológica ESPOL – RTE*, Vol. 26, N. 2, 16-27
- [6]Jiménez, J., Garrido, D. (2013). UAMIBOT: Un Robot Móvil Didáctico Multifunciones. *Rev. Avances de Ingeniería Electrónica*
- [7]Salgado, D., Salinas, E., Salinas, E. (2012) Determinación del efecto sedante de mezclas de extractos a base de vetiveria zizanioides, pasiflora edulis, melissa officinalis, y tilia cordata en ratones wistar en el periodo comprendido de abril-noviembre del 2012. *Repositorio Universidad*
- [8]Hederich, C., Camargo, A. (1998). Psicología cognitiva en la idea del Procesamiento de la información. *Revista de la Facultad de Artes Y Humanidades*
- [9]Hernández, A., Morales, M. (2014) Un modelo de conductas animales basados en redes de comportamiento.
- [10]Romero, J. (2007). Inteligencia artificial y computación avanzada. *Repositorio de la Universidad de Zaragoza*.
- [11]Felip, C. (2003) An Introduction to Satisfiability Algorithms. *Revista Iberoamericana de Inteligencia Artificial*. No.20 (2003)
- [12]Worrall, J. (2001). Programas de investigación y heurística positiva. Avance respecto de Lakatos. *Rev.La filosofía de Imre Lakatos*. pp. 247-268
- [13]Callejas, I., Piñeros, J., Rocha, J., Hernández, F., Delgado, F. (2013) Implementación de una red neuronal artificial tipo SOM en una FPGA para la resolución de trayectorias tipo laberinto," 2013 II International Congress of Engineering Mechatronics and Automation (CIIMA), Bogota, 2013, pp. 1-6.
- [14]González, H., Mejía, C. (2007). Estudio comparativo de tres técnicas de navegación para robot móviles. *UIS Ingenierías*, Vol. 06 N°01, pp.77-84
- [15]González, A., Vergara, R., Cano, M., Ramírez, J., Zavala, A. (2017) Algoritmos para la solución de laberintos utilizando kinect. *Rev. Jóvenes en la Ciencia*. Vol.3. Numero 2.
- [16]Jiménez, A., Petearson, J., Tarazona, G., Bolaños, S. (2017) Modelo para la simulación de sistemas de multi-agentes robóticos en python. *Rev. Redes de Ingeniería*. Volúmen Especial.
- [17]Fuentes, S., Ramos, P. (2017) Análisis de los nuevos paradigmas de computación. Trabajo de Grado. Universidad de Alcalá. España
- [18]Euresty, C., Gutiérrez, F. (2017) Algoritmo para desarrollar un autómata que reconozca ambientes desconocidos. *Rev.Pistas Educativas*, Vol.36. N°114
- [19]Euresty, C., Gutiérrez, F., Ortega, C. (2015) Técnica para reconocer ambientes desconocidos. *Rev.Pistas Educativas*, Vol.35. N°111
- [20]Molina, J., Torres, C., Restrepo, C. Técnicas de inteligencia artificial para la solución de laberintos de estructura desconocida. *Rev. Scientia et Technica Año XIV, No 39,*