

# Red Neuronal Pulsante Adaptada al Problema del Camino más Corto

## Spiking Neural Network Adapted to the Shortest Path Problem

Manuel Mejía-Lavalle, José Jesús Paredes-Cano , Dante Mújica-Vargas , Andrea Magadán-Salazar 

Tecnológico Nacional de México / Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)  
Interior Internado Palmira SN, 62490 Palmira, Cuernavaca, Morelos, México

Correo-e: mlavalle, joseparedes16c, dantemv, magadan { @cenidet.edu.mx }

### PALABRAS CLAVE:

Red Neuronal Pulsante, Optimización, Problema de Camino más Corto, Explicación de Conocimiento, Diseño Paralelo.

### RESUMEN

La solución eficiente del problema del camino más corto tiene aplicaciones en áreas tan importantes y actuales como la robótica, las telecomunicaciones, la investigación de operaciones, la teoría de juegos, las redes de computadoras, internet, diseño industrial, fenómenos de transporte, diseño de circuitos electrónicos y otros, por lo que es un tema de gran interés en el área de optimización combinatoria. En el presente trabajo se describe una Red Neuronal Artificial Pulsante capaz de atacar eficientemente el problema del camino más corto entre dos nodos. Una vez que la Red Pulsante encuentra el nodo meta a costo mínimo, se realiza una extracción o Explicación de Conocimiento de esta Red para recuperar la trayectoria final. Debido al diseño en paralelo de la Red Neuronal aquí presentada, este enfoque de solución puede resultar altamente competitivo, según se observó en la etapa de experimentación a partir de los buenos resultados obtenidos, aún en casos con miles de nodos.

### KEYWORDS:

Spiking Neural Network, Optimization, Shortest Path Problem, Knowledge Explication, Parallel Design.

### ABSTRACT

The efficient solution of the shortest path problem has applications in such important and current areas as robotics, telecommunications, operations research, game theory, computer networks, internet, industrial design, transport phenomena, design of electronic circuits and others, so it is a subject of great interest in the area of combinatorial optimization. In the present work, we describe a Spiking Artificial Neural Network capable of efficiently attack the problem of the shortest path between two nodes. Once the Spiking Network finds the target node at minimum cost, an extraction or Knowledge Explication of this Network is performed to recover the final trajectory. Due to the parallel design of the Neural Network presented here, this solution approach can be highly competitive, as observed from the good results obtained in the experimental phase, even in cases with thousands of nodes.

**Recibido:** 28 de junio de 2018 **Aceptado:** 10 de octubre de 2018 **Publicado en línea:** 28 junio de 2019

## 1. INTRODUCCIÓN

El problema de encontrar el camino o la ruta más corta entre dos nodos ha sido ampliamente estudiado en el área de optimización combinatoria [1] debido a la importancia que tiene relacionado con dominios tan actuales como: la robótica, las telecomunicaciones, la inteligencia artificial y aprendizaje automático, la investigación de operaciones, la teoría de juegos, las redes de computadoras, internet, diseño industrial, telemática, fenómenos de transporte, diseño de circuitos electrónicos altamente integrados VLSI, e incluso se ha aplicado a situaciones de riesgo y de toma de decisiones [1].

Dado un conjunto de múltiples nodos conectados con un cierto costo, el problema de la ruta más corta consiste en encontrar una trayectoria que una a un nodo de inicio con un nodo meta de tal manera que la suma de los costos de ese camino sea la mínima posible. Una manera sencilla de describir el problema es hacer una analogía con un grupo de ciudades: cada ciudad es un nodo y la distancia entre dos ciudades es el costo. Sin embargo, este problema encuentra aplicaciones del mundo real en múltiples y diversos dominios. En teoría de grafos existen definiciones formales para este problema [2].

Computacionalmente el grafo de rutas se representa normalmente mediante una matriz de adyacencia de tamaño  $n \times n$ , donde  $n$  es la cantidad de nodos del problema: si el elemento  $i, j$  de la matriz de adyacencia vale cero indica que del nodo  $i$  al nodo  $j$  no existe un camino; por otro lado, si el elemento matricial tiene un valor positivo, éste indicará el costo en que se incurre por transitar del nodo  $i$  al nodo  $j$ . Aunque en apariencia el problema pareciera sencillo, en realidad se trata de un problema de alta complejidad que se hace evidente cuando tratamos de encontrar soluciones óptimas para redes con cientos o miles de nodos conectados. Esto se debe a que la cantidad de rutas posibles crece de manera exponencial en relación al número de nodos, por lo que es inviable analizar todas las combinaciones posibles en un tiempo razonable, aún usando las computadoras más veloces.

Existen diversos algoritmos que buscan resolver el problema sin tener que recorrer todas las rutas posibles. Pero, aún usando un algoritmo eficiente que no analiza todas las rutas posibles, para un problema con 9,000 nodos se necesitarían alrededor de 81 millones de operaciones para encontrar una solución óptima o cercana a la óptima.

Entre los algoritmos eficientes más conocidos está el de Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson, Gabow, Thorup, Pettie-Ramachandran o Viterbi [1]. Incluso del área de la Inteligencia Artificial han surgido propuestas como el *Branch & Bound* o el algoritmo  $A^*$  [2]. Buscando la eficiencia, no todos los algoritmos garantizan encontrar la solución óptima (la ruta de menor costo), sino que ofrecen soluciones cercanas a la óptima. Adicionalmente, todos estos algoritmos han sido concebidos para ser resueltos esencialmente de una manera secuencial [3].

En el presente trabajo se propone y presenta la aplicación del enfoque conexionista de las Redes Neuronales Artificiales (RNA) [4], en particular el de las RNA de tipo Pulsante adaptada para resolver el problema del camino más corto. Lo anterior debido a que las RNA tienen un diseño intrínsecamente paralelo, lo que las hace susceptibles de ser más eficientes que los métodos secuenciales tradicionales que no cuentan con este enfoque conexionista. Adicionalmente, las RNA de tipo Pulsante son consideradas la tercera generación de las RNA y son más cercanas en concepto a las redes neuronales biológicas, lo que promete un mayor acercamiento a los procesos cognitivos de los humanos [5]. Una vez que la Red Neuronal Pulsante encuentra el nodo meta a un costo mínimo, se realiza una extracción o Explicitación de Conocimiento [6], a partir de la información oculta en la Red, para recuperar la trayectoria final.

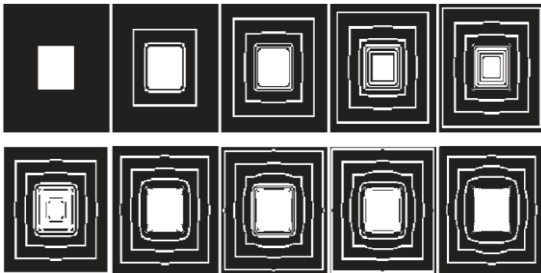
De acuerdo a la experimentación realizada utilizando casos con miles de nodos, los resultados son alentadores tanto en eficiencia como en la calidad de la solución. Con los casos experimentados se obtuvieron tiempos de respuesta en el orden de los segundos y todos los casos fueron resueltos con la solución óptima de mínimo costo global.

## 2. RED NEURONAL PROPUESTA

En esta sección se presenta la Red Neuronal Pulsante adaptada para resolver problemas que buscan la ruta de menor costo entre dos nodos. Para ello primeramente se introducen las Redes Neuronales Pulsantes, para luego describir la Red Neuronal propuesta y los resultados obtenidos.

### 2.1 Redes Neuronales Artificiales Pulsantes

Dentro de las Redes Neuronales Artificiales (RNA), la tercera generación corresponde a las denominadas Pulsantes (RNP) y son las que en este trabajo proponemos y aplicamos al problema de encontrar el camino más corto entre dos nodos. La principal diferencia de las RNP con respecto a las dos generaciones anteriores es que son un paradigma cuyas neuronas son activadas no sólo por valores numéricos, sino además por pulsos o señales externas que dependen del tiempo [5]. La RNP es implementada en un modelo de una neurona por cada elemento de procesamiento y su arquitectura tiene tres módulos: a) el módulo de dendritas, mediante el cual entra la información circundante a la RNP; b) el módulo de encadenamiento, que une la información de entrada con la de salida y c) el generador de pulsos, que tiene un umbral dinámico que se basa en el proceso interno de la misma RNP. Las RNP tienen la particularidad de que la actividad de cada neurona afecta en cierto grado a las neuronas vecinas, obteniéndose un efecto de auto-onda [5]. La Figura 1 muestra un ejemplo de auto-onda: a partir de una imagen de un cuadrado blanco, se muestran en secuencia nueve salidas de la RNP.



**Figura 1.** Salidas de RNP auto-onda para nueve épocas [5].

Dentro de las RNP existen diversas variantes, como la denominada *Pulse-Coupled Neural Network*, la *Intersecting Cortical Model* o la *Spiking Cortical Model* [5]. La RNP que aquí se propone es una variante diseñada especialmente para el problema del camino más corto.

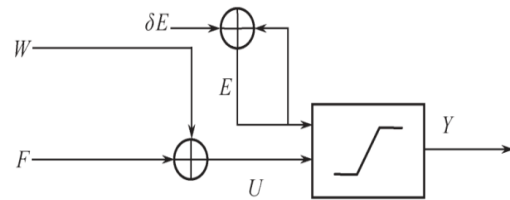
### 2.2 Red Neuronal Adaptada

La descripción del paradigma propuesto se presenta en dos partes: primero se aborda el proceso para, a partir de un nodo inicial, ir explorando los nodos que

conduzcan al nodo meta de manera óptima; posteriormente se presenta la forma en que se recupera el conjunto de nodos que determinan el camino más corto, mediante la extracción o Explicitación del Conocimiento [6] oculto en dicha Red.

#### Proceso para encontrar el nodo meta

La Red Neuronal Pulsante que aquí se propone es una extensión de la presentada en [7] denominada por sus autores como AWNN (*Auto-Wave Neural Network*). En la Figura 2 se muestra una neurona de este modelo.



**Figura 2.** Auto-Wave Neural Network, según [7].

En la Figura 2 se puede observar que cada neurona recibe como entrada información de la matriz de adyacencia  $W$  y además recibe un valor  $F$  del costo adicional en caso de continuar por cierta ruta: sumando  $W$  y  $F$  se obtiene el valor  $U$  que es el costo total incurrido por algún camino hasta un momento dado determinado por el umbral dinámico  $E$ . En cada época el parámetro  $E$  se incrementa por un valor constante  $\delta E$ : si el costo acumulado en  $U$  es igual o supera al umbral  $E$ , entonces la neurona se activa tomando la salida  $Y$  el valor de “1”, de otra manera la salida de la neurona será cero. Dado un problema con  $n$  nodos, la red neuronal tendrá entonces tantas neuronas como  $n \times n$ . El efecto de auto-onda se obtiene precisamente al ir realizando incrementos en el tiempo del umbral  $E$ . Las siguientes ecuaciones resumen el comportamiento de la neurona tipo AWNN que proponemos:

$$F_{actual} = F_{anterior} + W_{anterior, actual} \quad (1)$$

$$U_{actual, actual} = F_{actual} + W_{actual, actual} \quad (2)$$

Si  $U > E$  entonces  $Y = 0$ ; de otra manera  $Y = 1$

$$E = E + \delta E \quad (3)$$

Para hacer más eficiente el proceso nosotros definimos el valor inicial de  $E$  como el costo menor

del nodo de inicio a sus siguientes nodos. De igual manera, si los costos son enteros positivos, recomendamos emplear una  $\delta E$  igual a 1 (o bien normalizar todos los costos de  $W$  para que sean enteros positivos).

A continuación se ilustrará con un ejemplo pequeño la forma en que la Red Neuronal Pulsante funciona. Iniciamos con una matriz de adyacencia  $W$  con 5 nodos que definen un grafo de interconexión. La matriz de ejemplo  $W$  es:

$W$	A	B	C	D	E
A	0	2	3	7	2
B	2	0	3	0	0
C	3	3	0	6	0
D	7	0	6	0	4
E	2	0	0	4	0

Se desea encontrar el camino más corto del nodo A al nodo D. Analizando los costos de A a sus siguientes nodos, vemos que el menor costo es A-B y A-E con un valor de 2, por lo tanto con ese valor iniciamos el umbral  $E$ . En el inicio, la  $F_{actual}$  será igual al renglón de la matriz de adyacencia donde se encuentra el nodo de inicio A. Entonces:

$$F_{actual} = 0 \ 2 \ 3 \ 7 \ 2$$

A continuación se actualiza la matriz  $U$  como lo indica la ecuación 2, quedando  $U$  con la siguiente información:

$U$	A	B	C	D	E
A	<b>0</b>	<b>2</b>	<b>3</b>	<b>7</b>	<b>2</b>
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0

Seguidamente verificamos los elementos donde  $U = E$  y se actualiza la matriz  $Y$ , con lo que resulta una primera salida de la RNP:

$Y$	A	B	C	D	E
A	0	<b>1</b>	0	0	<b>1</b>
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0

Como los caminos A-B y A-E ya fueron seleccionados, se les asigna un “costo alto” para que ya no vuelvan a ser usados (para el ejemplo usamos el valor 10 como “costo alto”) y se incrementan los costos de los renglones de B y E de acuerdo a los costos de la matriz  $W$ . La matriz  $U$  entonces queda de la siguiente manera:

$U$	A	B	C	D	E
A	<b>10</b>	2	3	7	2
B	<b>10</b>	0	<b>2+3</b>	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	<b>10</b>	0	0	<b>2+4</b>	0

Luego se incrementa en 1 el umbral  $E$  y se repite el proceso. En este caso es el camino A-C el que cumple con tener un costo igual o menor al umbral  $E$ , por lo que la matriz  $Y$  se actualiza a:

$Y$	A	B	C	D	E
A	<b>1</b>	1	<b>1</b>	0	1
B	0	0	0	0	0

C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0

En consecuencia la matriz  $U$  se actualiza a:

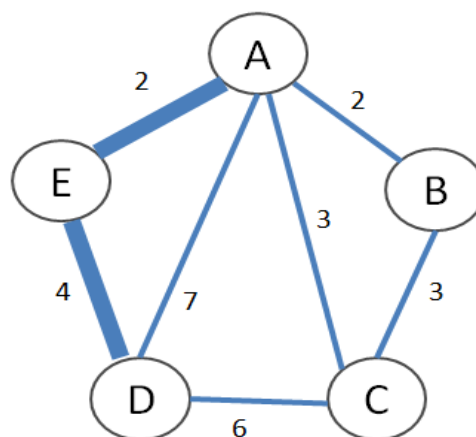
$U$	A	B	C	D	E
A	10	2	3	7	2
B	10	0	2+3	0	0
C	10	10	0	3+6	0
D	0	0	0	0	0
E	10	0	0	2+4	0

A continuación se incrementa el umbral y entonces  $E=4$ . En este momento ningún camino (no recorrido anteriormente) es igual o menor a 4, por lo que la salida  $Y$  no se modifica y por lo tanto la matriz  $U$  tampoco cambia. Hasta que  $E=6$  ocurre que el camino E-D ( $2+4$ ) en la matriz  $U$  satisface el umbral y se actualiza la salida:

$Y$	A	B	C	D	E
A	1	1	1	0	1
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	1	0

Puesto que en este paso se llegó al nodo meta (D), entonces el proceso termina. El costo del camino encontrado es igual al valor del umbral  $E$ , en este caso el costo es de 6. Se hace notar que cualquier otro camino que llegue a D tendría mayor costo, por lo que el camino encontrado es el óptimo. La Figura 3 muestra el grafo con los 5 nodos y los costos de uno a otro nodo; además, el camino óptimo A-E-D

encontrado por la RNP se resalta con líneas más gruesas.



**Figura 3.** Grafo que muestra la solución final de la RNP.

### *Recuperación de la ruta más corta por Explicitación*

La Explicitación de Conocimiento [6] en el contexto de las RNA consiste en extraer el conocimiento oculto en la misma Red Neuronal. Este sub-tema dentro de las RNA se debe a que, en el surgimiento de la tecnología, existía la crítica de que aunque los paradigmas neuronales daban buenos resultados, no era posible acceder al conocimiento que de manera automática el algoritmo había generado. Se decía entonces que las RNA eran algoritmos de caja negra. Sin embargo, con el tiempo, han surgido propuestas exitosas para precisamente transparentar el conocimiento oculto entre las neuronas, de tal manera que el usuario tenga acceso a, por ejemplo, las reglas de producción que se generan de manera automática como resultado de la actividad interna de la RNA. En ese sentido, nosotros logramos rescatar el camino óptimo o de menor costo, extrayendo el conocimiento que queda oculto en la capa de salida  $Y$  de la RNP propuesta.

La capa de salida  $Y$  es básicamente una matriz de tamaño  $n \times n$  con valores de ceros y unos. La ruta se recupera de manera inversa, saltando del nodo final al nodo inicial. En la matriz de salida  $Y$ , el nodo final se encontrará en la columna  $i$  con el mismo número que el nodo meta. En esa columna se buscará el valor "1" y ahí donde aparezca se tomará el valor del renglón  $j$ . Con ese valor de renglón se entrará nuevamente a la matriz  $Y$  pero ahora buscando en la columna con valor  $j$ , el valor "1" y repitiendo el

proceso hasta llegar al nodo inicial. Al final los valores  $j$  así recuperados definirán la ruta de costo mínimo.

Siguiendo con el ejemplo de la sección anterior y considerando que la salida  $Y$  final es:

<b>Y</b>	A (i1)	B (i2)	C (i3)	D (i4)	E (i5)
A (j1)	1	1	1	0	1
B (j2)	0	0	0	0	0
C (j3)	0	0	0	0	0
D (j4)	0	0	0	0	0
E (j5)	0	0	0	1	0

el nodo final D se encuentra en  $i=4$  y  $j=5$ . Usando el valor de  $j$  como referencia, vamos a la columna  $i=5$ , correspondiente al nodo E, y en donde el elemento marcado con "1" se encuentra en el renglón  $j=1$ . Usando el valor de  $j$  una vez más como referencia, vamos a la columna  $i=1$ , correspondiente al nodo A y en donde el elemento marcado con "1" se encuentra en el renglón  $j=1$ . En este momento se termina la explicitación pues se alcanzó el nodo inicial. De esta manera se recupera el camino de menor costo que en este caso es A-E-D.

### 3. EXPERIMENTACIÓN Y RESULTADOS

Los casos de prueba se generaron de manera aleatoria. En la Tabla I se muestran los resultados obtenidos aplicando la RNP propuesta. La Figura 4 muestra los mismos resultados pero de manera gráfica; observando esta Figura se puede apreciar que el comportamiento de la RNP es de complejidad aproximadamente lineal  $O(n)$ , es decir, el tiempo de procesamiento depende de manera lineal del número de nodos.

Abundando en el análisis de la complejidad computacional de los algoritmos que se han propuesto en la literatura especializada, encontramos que el algoritmo de Dijkstra, que es uno de los más conocidos y utilizados, tiene una complejidad cuadrática, es decir  $O(n^2)$ ; esto significa que el tiempo de procesamiento crece exponencialmente en

relación al número de nodos; así, por dar un ejemplo, para un problema con 9,000 nodos, el algoritmo de Dijkstra requerirá tanto tiempo de procesamiento como de 81,000,000  $t$ , donde  $t$  es el tiempo que requiere una computadora particular para ejecutar un ciclo de análisis. En el caso de la RNP propuesta, aún sin considerar el paralelismo, la complejidad aproximada resulta ser de tan sólo 9,000  $t$ . Esto contrasta de manera muy significativa, pues aun suponiendo que se cuenta con un computador muy potente, con una  $t$  de 1 nano segundo, el tiempo requerido por Dijkstra sería de 81 mili-segundos, mientras que con el algoritmo propuesto tan sólo se necesitarían 9 micro-segundos, es decir, resulta ser 9,000 veces más veloz el algoritmo propuesto.

Es importante mencionar que por ahora no se aprovechó el diseño paralelo de la RNP. Cuando tengamos completamente implementado el código en forma paralela, esperamos obtener tiempos de respuesta mucho menores. Por ejemplo, continuando con el caso con 9,000 nodos, el tiempo  $T$  esperado de respuesta podría bajar en alrededor de  $T/81,000,000$ , lo cual es aproximadamente 30 millones de veces más eficiente que el algoritmo de Dijkstra.

Se hace notar que en todos los casos con los que se experimentó, la RNP propuesta logró encontrar el camino óptimo.

**Tabla I.** Tiempos para diferente número de nodos

Número de Nodos	Tiempo de procesamiento de la RNP propuesta (mili - segundos)
5	0.003175
30	0.03458
50	0.9887
100	4.96295
200	3.81465
300	29.552
400	32.5496
500	62.1867
600	75.7672
700	88.5191
800	98.7029
900	216.031
1,000	288.279
1,300	301.116
1,500	562.768
1,800	591.703
2,000	717.14
2,400	1,042.9
2,600	1,346.87
3,000	1,807.27
3,500	1,956.36
4,000	2,957.94
4,500	3,980.56

5,000	4,019.82
5,500	7,091.83
6,000	7,380.67
7,000	7,904.95
8,000	9,221.16
9,000	10,962.2

#### 4. CONCLUSIONES Y TRABAJO FUTURO

Se ha propuesto y descrito una RNP capaz de resolver de manera satisfactoria el problema del camino más corto. De acuerdo a los resultados obtenidos nuestra propuesta compite con otros métodos existentes en la literatura especializada, pues tiene una complejidad aproximadamente lineal. Sin embargo, creemos que al aprovechar la característica paralela del paradigma propuesto, los tiempos de respuesta disminuirán considerablemente. Por lo tanto, como trabajo futuro tenemos contemplado realizar la implementación paralelizada usando miles de núcleos de la Unidad de Procesamiento de Gráficos (GPU) para acelerar el proceso, según se detalla en [8]. Adicionalmente se experimentará con casos del mundo real como los mencionados en [9] ó [10] (ver Figura 5). En particular nos interesa aplicar los resultados obtenidos en el área de planeación óptima de trayectorias de robots móviles (ver Figura 6) [11], en donde no solamente se desea minimizar el costo del recorrido, sino también lograr tiempos de respuesta en tiempo real, lo cual es de gran relevancia en situaciones del mundo real de la robótica, además de navegar evitando obstáculos en mapas 2D.

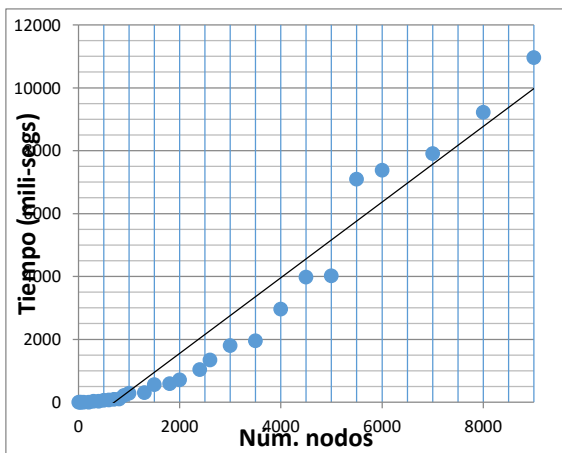


Figura 4. Tiempo de procesamiento (mili-segs) vs número de nodos.

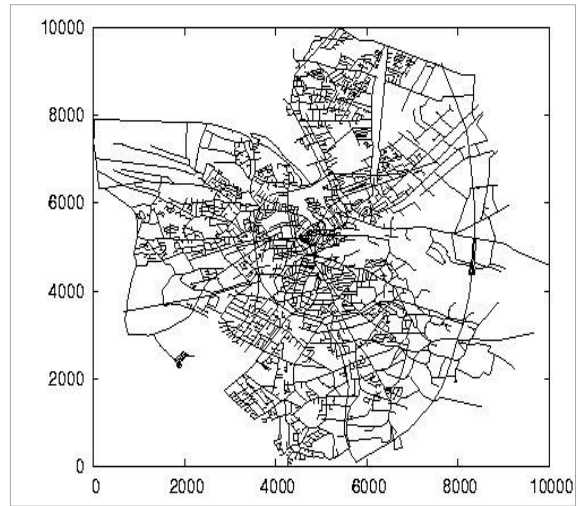


Figura 5. Caso real para trabajos futuros [10].

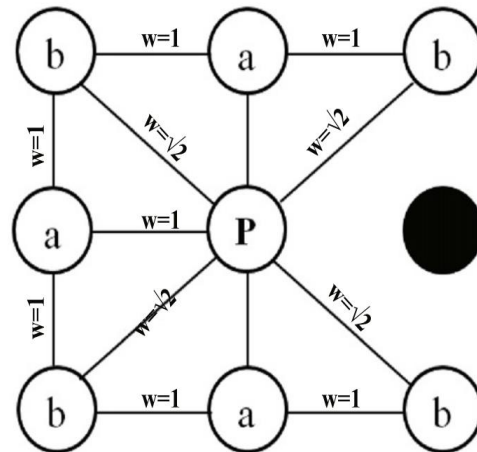


Figura 6. Esquema neuronal parcial para planear rutas en mapas 2D considerando obstáculos (neurona color negro) [11].

## 5. REFERENCIAS

- [1] Moustapha, D., Mark, K., *Advances in combinatorial optimization*, World Scientific, 2016.
- [2] Thulasiraman, K., Arumugam, S., et al., *Handbook of graph theory*, CRC Press, 2016.
- [3] Beck, A., *First-Order Methods in Optimization*, Vol 25 de *MOS-SIAM Series on optimization*, 2017.
- [4] Daniel, G., *Principles of artificial neural networks*, World Scientific, 2013.
- [5] Lindblad, T., Kinsler, J., *Image processing using pulse-coupled neural networks*, Springer, 1998.
- [6] Chuanli, Z., Jinzheng, R., Elicitation of decision maker preference by artificial neural networks, *IEEE Int. Conf. on Neural networks and brain*, 2005.
- [7] Ma, Y., Zhan, K., Wang, Z., *Applications of pulse-coupled neural networks*, Springer, 2011.
- [8] Kirk, D., *GPU Gems*, Addison-Wesley, 2004.
- [9] Computer science. Recuperado en abril 2018, de <https://cs.stackexchange.com/questions/1151/where-to-get-graphs-to-test-my-search-algorithms-against>.
- [10] Brinkhoff, T., Recuperado en abril 2018, de <https://www.cs.utah.edu/~lifeifei/SpatialDataset.htm>.
- [11] Syed, U.A., Kunwar, F., Iqbal, M., Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots, *Robotics and autonomous systems*, 62, 2014, 474-486.

## 6. AGRADECIMIENTOS

Los autores reconocen al Tecnológico Nacional de México el apoyo prestado para la realización del presente trabajo.



Acerca de autores



Manuel Mejía-Lavalle. Es Doctor en Ciencias Computacionales por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) y profesor - investigador en el Centro Nacional de Investigación y Desarrollo Tecnológico / TecNM.

Pertenece al IEEE habiendo sido presidente del Capítulo de Computación de la Sección Morelos, México en el período 2011-2012. Cuenta con Perfil Deseable PRODEP. Ha pertenecido también al Sistema Nacional de Investigadores, al Sistema Estatal de Investigadores (Morelos), a la Sociedad Mexicana de Inteligencia Artificial, y a la Academia Nacional de Investigación en Ciencias Computacionales. Investiga principalmente en el área de Inteligencia Artificial y de las Redes Neuronales Artificiales.



José de Jesús Paredes. Actualmente es estudiante de la Maestría en Ciencias Computacionales con especialidad en Inteligencia Artificial en el Centro Nacional de Investigación y Desarrollo

Tecnológico (CENIDET) perteneciente al Tecnológico Nacional de México (TecNM), desarrollando el tema de tesis "Redes neuronales pulsantes como alternativa para la optimización de trayectorias". Es Ingeniero en Telemática por la Universidad Politécnica del Estado de Guerrero. En 2015 realizó una estadía en el CENIDET con el tema "Comunicación mediante conexión en red de dos motores de inducción trifásicos. Sus áreas de investigación son las Redes Neuronales Artificiales, en especial las de Tercera Generación aplicadas a resolver problemas de optimización.



Dante Mújica Vargas. Obtuvo los grados de Dr. en Comunicaciones y Electrónica, así como el de Maestro en Ciencias en Ingeniería Electrónica en la Sección de Estudios de Posgrado e Investigación del Instituto Politécnico Nacional. Es miembro del Sistema Nacional de Investigadores (SNI) nivel I, miembro honorífico del Sistema Estatal de Investigadores (Morelos) y cuenta con el reconocimiento al Perfil Deseable. Ha publicado más de 10 artículos en revistas indizadas en el JCR, principalmente en Springer, Elsevier e IEEE. Su trabajo de investigación está enfocado en el Procesamiento Digital de Imágenes, Aprendizaje Profundo, Agrupamiento Difuso, Aceleración de algoritmos mediante GPU's, así como Robótica Móvil Probabilística.



Andrea Magadán-Salazar. Es Doctora en Tecnologías de la Información y Sistemas Informáticos por la Universidad Rey Juan Carlos del país

de España, obteniendo el título en el año 2016. Es Maestra en Ciencias, en Ciencias de la Computación, por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) perteneciente al Tecnológico Nacional de México (TecNM), en Cuernavaca, Morelos, México, titulada en el año de 1999. Licenciada en Informática por el Instituto Tecnológico de Zacatepec, Morelos, México, con título del año de 1995. Sus áreas de interés son principalmente el Reconocimiento de Patrones, el Aprendizaje de Máquina y la Visión Artificial, entre otras.