



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

**Redes Neuronales Pulsantes para la Planeación
Eficiente de Trayectorias en Mapas Discretos**

presentada por

**L. I. Oliver Eder Guillermo Espinosa
Meneses**

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis

Dr. Manuel Mejía Lavalle

Cuernavaca, Morelos, México. Junio de 2018.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., 09/enero/2020

OFICIO No. DCC/002/2020
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Lic. Oliver Eder Guillermo Espinosa Meneses**, con número de control M18CE005, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado "**Redes Neuronales Pulsantes para la Planeación Eficiente de Trayectorias en Mapas Discretos**" y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Manuel Mejía Layalle
Doctor en Ciencias Computacionales
8342472
Director de tesis

M.C. Gerardo Reyes Salgado
Maestro en Ciencias de la Computación
2493370
Revisor 1

Dr. José Ruiz Ascencio
Doctor en Ciencias
5009035
Revisor 2

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JGCS/Imz



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos, **29/enero/2020**

OFICIO No. SAC/095/2020
Asunto: Autorización de impresión de tesis

LIC. OLIVER EDER GUILLERMO ESPINOSA MENESES
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "*Redes Neuronales Pulsantes para la Planeación Eficiente de Trayectorias de Mapas Discretos*", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

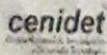
DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares.
Expediente.

GVGR/CHG



Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir_cenidet@tecnm.mx
www.tecnm.mx | www.cenidet.edu.mx



Dedicatoria

Quiero dedicar este logro a mi familia, especialmente a mi madre, por que hizo lo mejor por que fuera una persona de bien. Se lo dedico a mi padre porque siempre ha estado dispuesto a darme su mejor consejo. Se lo dedico a mi novia que me ha apoyado todo el tiempo y me ha impulsado desde el inicio a perseguir mis metas.

Le dedico de corazón a mis tres compañeros de armas y hermanos de la orden de los miércoles de Batman, por su amistad y porque siempre me esforcé por aprender de ellos.

Quiero dedicarles este logro a mis excelentes profesores, porque se han ganado todo mi admiración y respeto, y porque me han mostrado el camino que quiero seguir.

Agradecimientos

En primer lugar, Agradecen al Tecnológico Nacional de México y al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).

Un agradecimiento especial al Instituto Nacional de Electricidad y Energías Limpias por abrir sus puertas para la realización de mi estancia fue una grata experiencia y especialmente gracias al Dr. Miguel Pérez Ramírez por compartir su experiencia, conocimiento y tiempo conmigo.

Doy las gracias a los profesores y personal del CENIDET que contribuyeron a mi formación como Maestro en Ciencias, especialmente a mi director el Dr. Manuel Mejía Lavalle, por su tiempo, dedicación, consejos y enseñanzas. Así mismo agradezco a mis revisores el Dr. José Ruiz y el Dr. Gerardo Reyes por sus enseñanzas, tiempo, consejos y críticas. Así mismo, agradezco a la oficina de Servicios estudiantiles por su tiempo.

Finalmente agradezco a toda el área de ciencias de la computación, en especial al departamento de inteligencia artificial.

Resumen

En el presente documento de tesis contiene una investigación sobre distintas variantes de Red Neuronal Pulso Acoplada aplicadas en el campo de la Planeación de Trayectorias. Durante el desarrollo de la tesis, se estudiaron e implementaron cuatro variantes de Red Neuronal Pulso Acoplada diseñadas para resolver el problema de la Ruta más Corta, tres variantes fueron seleccionadas de la literatura y una variante más fue propuesta como producto de esta investigación. Adicionalmente, se implementaron dos variantes con un diseño específico para tratar con la Planeación de Trayectorias, un modelo fue encontrado en la literatura y uno más es presentado en esta investigación.

De acuerdo con los resultados de las pruebas realizadas con los algoritmos de Optimización de Trayectorias, la variante propuesta de Red Neuronal Pulso Acoplada es hasta 39.28% más eficiente que la variante más eficiente presentada en la literatura y hasta un 62.77% más eficiente que la variante *Auto Wave Neural Network*. Por otro lado, los resultados de los experimentos con los algoritmos de Planeación de Trayectorias indican que el modelo propuesto es hasta un 50% más eficiente en el número de iteraciones y cerca del 50% más rápido en tiempo de ejecución.

Esta tesis incluye un estudio del marco teórico sobre la Planeación de Trayectorias, Optimización de Trayectorias y Redes Neuronales Pulso Acopladas. Además, se presentan los trabajos que conforman el estado del arte. Asimismo, se presenta el estudio realizado sobre cada una de las variantes de Red Neuronal Pulso Acoplada implementadas en esta investigación. Por otra parte, se presentan y analizan los experimentos y resultados obtenidos de las búsquedas de distintas trayectorias. Finalmente se exponen las conclusiones a las que se llegó en esta investigación.

Abstract

In this thesis contains an investigation on different variants of Pulse Coupled Neural Networks applied in the Path Planning field. During the development of the thesis, four variants of Pulse Coupled Neuronal Network were studied and implemented to solve the problem of the Shortest Path Problem, three variants were selected from the literature and one more variant was proposed as a product of this research. In addition, two variants are implemented with a specific design to deal with Path Planning, a model found in the literature and one more is presented in this research.

According to the results of the tests carried out with the Path Optimization algorithms, the proposed variant of the Pulse Coupled Neural Network is up to 39.28% more efficient than the most efficient variant presented in the literature and up to 62.77% more efficient than the variant Auto Wave Neural Network. On the other hand, the results of the experiments with the Path Planning algorithms indicate that the proposed model is up to 50% more efficient in the number of iterations and about 50% faster at runtime.

This thesis includes a study of the theoretical framework on the Planning of Paths, Optimization of Paths and Connected Pulse Neural Networks. In addition, the works that make up the state of the art are presented. Likewise, we present the study carried out on each of the variants of Neural Network Pulsed Coupling implemented in this investigation. On the other hand, the experiments and results obtained from the searches of different trajectories are presented and analyzed. Finally, the conclusions reached in this investigation are presented.

Índice general

Capítulo 1 Introducción	15
1.1 Motivación	15
1.2 Planteamiento del problema.....	15
1.3 Complejidad del problema	16
1.4 Enfoque de solución.....	16
1.5 Objetivos	17
1.5.1 Objetivo General	17
1.5.2 Objetivos Específicos.....	17
1.6 Alcances y limitaciones	17
1.6.1 Alcances	17
1.6.2 Limitaciones.....	17
1.7 Justificación	18
1.8 Estructura de la tesis	18
Capítulo 2 Marco de referencia.....	19
2.1 Antecedentes	19
2.2 Estado del arte.....	22
2.3 Discusión del estado arte.....	32
Capítulo 3 Marco teórico	38
3.1 Robótica Móvil	38
3.2 Planeación de Trayectorias	39
3.3 Optimización de Trayectorias	41
3.4 Redes Neuronales Pulsantes.....	45
3.5 Discusión.....	49
Capítulo 4 Metodología de Solución	50
4.1 Optimización de Trayectorias	50
4.1.1 <i>Auto-Wave Neural Network</i>	50
4.1.2 <i>Auto-Wave Neural Network Explicitada</i>	51
4.1.3 <i>Self-adaptive Autowave Pulse-Coupled Neural Network</i>	53
4.1.4 <i>Self-Adaptive Autowave Modified Pulse-Coupled Neural Network</i>	55
4.1.5 Complejidad Algorítmica.....	57
4.2 Planeación de Trayectorias	59
4.2.1 Modified Pulse-Coupled Neural Network	59
4.2.2 <i>Bidirectional Search Pulse-Coupled Neural Network</i>	63

4.3 Detalles de implementación	69
4.3.1 Implementación del sistema de algoritmos que utilizan grafos	69
4.3.2 Implementación de sistema de algoritmos que utilizan mapas	71
4.4 Discusión.....	73
Capítulo 5. Experimentación y resultados	75
5.1 Experimentos de Optimización de Trayectorias (Grafos).....	75
5.1.1 Experimento 1	75
5.1.2 Experimento 2	77
5.2 Experimentos de Planeación de Trayectorias (Mapas)	78
5.2.2 Sintonización de parámetros del modelo MPCNN.....	80
5.3 Discusión.....	87
Capítulo 6. Conclusiones	88
6.1 Objetivos logrados	88
6.1.1 Objetivo general.....	88
6.1.2 Objetivos específicos	88
6.1.3 Alcances y limitaciones	89
6.2 Productos y actividades extra.....	89
6.2.1 Participación como ponente en Escuela de Inteligencia Artificial y Robótica 2018 (octubre - 2018)	89
6.2.2 Presentación de poster 1ra Jornada de Ciencia y Tecnología Aplicada (noviembre - 2018). 90	
6.2.3 Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2018).....	90
6.2.4 Presentación de poster 2da Jornada de Ciencia y Tecnología Aplicada (abril - 2018).....	90
6.2.5 Presentación para el 1er coloquio “Artificial Intelligence” (mayo - 2019)	90
6.2.6 Participación como ponente en el XI Congreso Mexicano de Inteligencia Artificial (junio - 2019)	90
6.2.7 Participación como ponente en la 1er Jornada de Tecnologías de la información 2019 (junio -2019).....	90
6.2.8 Participación como ponente en 3ra Jornada de Ciencia y Tecnología Aplicada (noviembre - 2019)	90
6.2.9 Participación como ponente en Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2019)	90
6.2.10 Publicación de Artículo en la revista “ <i>Research in Computing Science</i> ” (noviembre - 2019)	90
6.3 Aportaciones	91
6.4 Conclusiones	91
6.5 Trabajo futuro	92

Referencias.....	93
Anexos	97
Anexo E. Presentación para el 1er coloquio “Artificial Intelligence” (mayo - 2019)	101
Anexo F. Participación como ponente en el XI Congreso Mexicano de Inteligencia Artificial (junio -2019)	102

Lista de Figuras

Figura 1. Etapas del enfoque de solución.	17
Figura 2. Comparación de trayectorias en el mapa planta alta, humano vs. Algoritmos [Rojas, 2018].	20
Figura 3. Resultados del mejoramiento de imágenes luminosas [Aguilar, 2019].....	22
Figura 4. Grafo asimétrico generado de manera aleatoria utilizado en los experimentos [Qu, 2007]....	22
Figura 5. Trayectoria generada por la Red Neuronal [Rios, 2008]......	23
Figura 6. (a) arquitectura de Red sin barrera. (b) arquitectura con barrera en el lugar 7 [Qu, 2009].	24
Figura 7. Comparación de los tiempos de ejecución entre el modelo propuesto y el algoritmo Dijkstra [Wang, 2009].	24
Figura 8. Comparativa del tiempo de ejecución entre la variante DSPCNN y la PCNN [Sang, 2010].	25
Figura 9. Simulaciones realizadas en la investigación de [Kroumov, 2011]......	26
Figura 10. Estructura de la neurona para la multiplicación de matrices [Zhang, 2011].	26
Figura 11. Gráfico ponderado simétrico con diez nodos y veinte aristas [Li, 2013].	27
Figura 12. Comparativa de espacio de búsqueda entre la variante MPCNN y GAPCNN [Usman, 2014].	28
Figura 13. Ejemplos de trayectorias óptimas [Khouil, 2016].	29
Figura 14. Mapas utilizados para evaluar la efectividad de los algoritmos [Carballo, 2017]......	29
Figura 15. Tiempo de procesamiento vs número de nodos de la RNP [Paredes-Cano, 2018].	30
Figura 16. Mapa de pruebas de [Yao, 2019]......	31
Figura 17. Ejemplo de planeación de trayectoria dentro de un laberinto [Muralidharan, 2019].	31
Figura 18. Representación de espacio de trabajo skeleton [Barraquand, 1992].	40
Figura 19. Modelado de fuerzas en un campo potencial [González, 2005]......	40
Figura 20. (A) problema siete puentes, (b) grafo basado en el problema de los 7 puentes [Ma, 2011].	42
Figura 21. De derecha a izquierda se muestra un grafo dirigido y un grafo no dirigido [Thulasiraman, 2016].	43
Figura 22. Grafo ponderado que muestra una trayectoria [Li, 2013].	44
Figura 23. Estructura básica de la neurona PCNN [Sabashini, 2014].	46
Figura 24. Modelo matemático de la PCNN [Sabashini, 2014]......	48
Figura 25. Modelo de la neurona Auto-Wave Neural Network [Ma, 2011]......	50
Figura 26. Grafo de tres nodos resuelto por el AWNN [Ma, 2011].	51
Figura 27. Grafo del ejemplo que muestra la solución de la RNP [Paredes-Cano, 2018]......	52
Figura 28. Grafo de tres nodos resuelto por el AWNN Explicitada [Paredes-Cano, 2018].	53
Figura 29. Diagrama del modelo SAPCNN [Li, 2013]......	53
Figura 30. Grafo de tres nodos resuelto por el SAPCNN [Li, 2013]......	54
Figura 31. Grafo del ejemplo.	55
Figura 32. Diagrama de la neurona del modelo propuesto SAM-PCNN.....	56
Figura 33. Grafo de tres nodos resuelto por el SAPCNN [Espinosa, 2019].....	58
Figura 34. (a) Ejemplo de la topología de la Red sin obstáculos, (b) topología de la Red con un obstáculo [Qu, 2009]......	59
Figura 35. Topología de la Red con conexiones locales.....	60
Figura 36. Diagrama del modelo MPCNN [Qu, 2009]......	61
Figura 37. Ejemplo de una búsqueda dual [Sang, 2010].	63
Figura 38. La Red a genera la auto-onda desde el nodo 1 y la Red b desde el nodo 15.....	64
Figura 39. Ejemplo de auto-ondas encontrándose en más de un punto.	65
Figura 40. Diagrama del sistema de algoritmos que utilizan grafos.....	69
Figura 41. Diagrama de los módulos del sistema de algoritmos que utilizan grafos.....	70

Figura 42. Diagrama del sistema de algoritmos que utilizan mapas.....	71
Figura 43. Diagrama de los módulos del sistema de algoritmos que utilizan mapas.	72
Figura 44. Ejemplo de mapa cargado en el sistema.	72
Figura 45. Ventana donde se seleccionan los nodos.	73
Figura 46. Trayectoria obtenida en el ejemplo.	73
Figura 47. Búsquedas realizadas sobre el grafo asimétrico con 10 nodos y 20 aristas [Li, 2013].	75
Figura 48. Gráfica comparativa del número de iteraciones.	77
Figura 49. Comparativa de iteraciones del Experimento 2.	77
Figura 50. Mapas utilizados para los experimentos [Vanasek,2009].	78
Figura 51. Mapas con las trayectorias obtenidas.	79
Figura 52. Grafica con los resultados de la búsqueda en mapas de 200 x 200 nodos.	80
Figura 53. rangos para los parámetros del MPCNN [Qu, 2009].....	80
Figura 54. Tiempos de ejecución de cada una de las combinaciones de parámetros.	81
Figura 55. Comparativa entre los resultados de la combinación A y la combinación B.	82
Figura 56. Comparación de distancias obtenidas por los algoritmos.....	84
Figura 57. Comparación de los tiempos de ejecución de cada algoritmo.....	84
Figura 58. Las iteraciones en cada experimento.	85
Figura 59. Trayectorias obtenidas por las PCNN en el mapa 2.	86
Figura 60. Comparativa de los tiempos de ejecución con diferentes resoluciones del mapa 2.	87

Lista de Tablas

Tabla 1. Los resultados muestran todos los árboles más cortos de la fig. 4 [Li, 2013].	28
Tabla 2. Comparativa del estado del arte y trabajos relacionados.	34
Tabla 3. Matriz de salida del ejemplo [Paredes-Cano, 2018].	52
Tabla 4. Tabla comparativa de complejidades algorítmicas.	57
Tabla 5. Resultado de las iteraciones por algoritmo del Experimento 1.	76
Tabla 6. Comparativa de tiempos de procesamiento en segundos del Experimento 2.	78
Tabla 7. Resultados de la búsqueda en mapas de 200 x 200 nodos.	79
Tabla 8. Resultados con diferentes combinaciones de parámetros.	81
Tabla 9. Resultados con la combinación de parámetros C.	82
Tabla 10. Resultados de las búsquedas A y B.	83
Tabla 11. Comparativa entre MPCNN y BSPCNN con sólo un agente de búsqueda.	83
Tabla 12. Resultados con mapas de 250 mil píxeles.	85
Tabla 13. Comparativa con mapas de distintas magnitudes.	86
Tabla 14. Objetivos específicos de la tesis	88
Tabla 15. Alcances y limitaciones de la tesis.	89

Acrónimos

Sigla	Nombre
BSPCNN	<i>Bidirectional Search Pulse-Coupled Neural Network</i>
DSPCNN	<i>Dual Source Pulse Couplet Neural Network</i>
JNN	<i>Jacobian Neural Networks</i>
MPCNN	<i>Modified Pulse-Coupled Neural Network</i>
MOPCNN	<i>Multi-Output Pulse Coupled Neural Network</i>
PCNN	<i>Pulse-Coupled Neural Network (Redes Neuronales Pulso Acopladas)</i>
RNA	Redes Neuronales Artificiales
SP	<i>Shortest Path (Camino más Corto)</i>

Capítulo 1 Introducción

En este capítulo, se presenta la motivación del tema de tesis, así como, el problema, los objetivos, alcances y limitaciones que se plantearon para la realización del trabajo de investigación.

1.1 Motivación

La Planificación de Trayectorias de Robots Móviles es una rama importante en el campo de la Robótica Móvil y de la Inteligencia Artificial [Khouil, 2016] [Yao, 2019]. Hasta ahora, ha habido una gran cantidad de resultados de investigación en este campo [Wu, 2016]. Las Redes Neuronales Artificiales Pulso-Acopladas representan un nuevo enfoque para abordar este problema y podrían ser de utilidad al mejorar la eficiencia del proceso debido a su paralelismo intrínseco [Qu, 2009]. Sin embargo, lograr la automatización es un problema con características difíciles de resolver y aunque se han propuesto múltiples esquemas de solución, aún sigue siendo un tema abierto de investigación.

Buscando el camino de soluciones computarizadas eficientes y eficaces, las Redes Neuronales Artificiales (RNA) se han empleado exitosamente en diversos dominios, siendo uno de ellos el área Planificación [Qu, 2009] [Syed, 2014]. En particular las RNAs de tipo Pulsante (consideradas como la tercera generación de RNAs) [Ma, 2011] son una novedosa variante que promete ser superior a los anteriores paradigmas neuronales y ya están siendo aplicadas en diversos dominios fuera del CENIDET. Planteamiento del Problema. El tema de tesis nace por el interés de explorar y evaluar esta tecnología en problemas de planeación de trayectorias de robots móviles.

1.2 Planteamiento del problema

La Planificación de Trayectorias es un problema fundamental en el campo de los robots móviles. Se encuentra clasificado como un problema de Optimización Combinatoria [Kroumov, 2011]. El objetivo de la Planificación de Trayectorias es mover robots desde sus ubicaciones iniciales hasta la ubicación de la meta mediante sus propios actuadores y estrategias, y durante el proceso, los robots siempre deben ser capaces de calcular trayectorias sin colisiones evitar obstáculos para mantener la seguridad del robot. Además, busca un camino más corto o *Shortest Path* (SP), por sus siglas en inglés), el cual le permite minimizar el esfuerzo [Yang, 2016]. El objetivo de resolver el problema del camino más corto es encontrar el camino de menor costo entre una fuente dada y un destino en una Red determinada, es decir, encontrar la ruta más corta en un grafo de un vértice a otro. Aquí, el "más corto" significa que el costo total que se correlaciona con la ruta debe ser el menor [Ma, 2011].

Se han presentado diversos enfoques para resolver el problema de la Planeación de Trayectorias. Sin embargo, aunque estos métodos son principalmente adecuados para entornos estáticos y son computacionalmente costosos en entornos complejos, tienen problemas para resolver problemas de mínimos locales y en ocasiones no garantizan la ruta óptima. Se ha utilizado el enfoque de las RNA para resolver el problema del camino más corto, en particular de las *Pulse-Coupled Neural Network* (PCNN, por sus siglas en inglés). Las PCNN son Redes no supervisadas, forman una Red auto organizada que no requiere entrenamiento. Las PCNN son capaces de emular el comportamiento de las neuronas corticales como se observa en las cortezas visuales de los mamíferos [Eckhorn, 1990]. Existen variantes de PCNN que resuelven el problema SP para Planeación de Trayectorias con éxito, buscando un rendimiento cada vez más eficiente. Sin embargo, no existe una investigación donde se estudien y se comparen los comportamientos de diferentes variantes de PCNN. Para realizar una investigación comparativa es necesario analizar e implementar los mecanismos de cada variante de PCNN y realizar experimentos para contrastar los resultados obtenidos.

Dado lo anterior, se puede decir que las PCNN son un enfoque eficiente dentro del campo de la Planeación de Trayectorias. Sin embargo, de esta afirmación es posible plantear una hipótesis.

Hipótesis:

A partir del estudio de diferentes variantes de PCNN orientadas a la Planificación de Trayectorias, es posible proponer nuevas variantes con el objetivo de realizar con mayor eficiencia el cálculo del camino más corto.

1.3 Complejidad del problema

La complejidad del problema radica principalmente en el estudio y la implementación de las variantes de las PCNN presentadas en la literatura. Además de la experimentación con problemas presentados en la literatura, así como experimentación con bases de datos de uso común en la comunidad de optimización de trayectorias. Por último, la realización de un estudio comparativo con las técnicas tradicionales que puedan ser utilizadas en proyectos y desarrollos posteriores.

1.4 Enfoque de solución

Se pretende llevar a cabo el estudio e implementación de diversas variantes de PCNN para la Planeación y Optimización de Trayectorias. Tomando en cuenta las características de las PCNN y conociendo las mejores características de las variantes presentadas en la literatura, se propondrán nuevas variantes cuyos mecanismos sean más eficientes en términos de complejidad algorítmica y tiempo de ejecución. Se medirá el desempeño de cada una, realizando búsquedas en grafos y mapas utilizados en la literatura y en la comunidad científica. Se integrará el estudio comparativo con los resultados propios y los resultados reportados en la literatura de otras técnicas tradicionales. De acuerdo con la Figura 1, El proceso antes mencionado, consta de cuatro etapas; la preparación, la implementación, la experimentación y la evaluación.

En la preparación se analizarán los problemas típicos de planeación de trayectorias a través del estudio de los temas relacionados y del estado del arte, para posteriormente seleccionar los problemas a tratar. En la implementación se replicarán los algoritmos basados en variantes de PCNN para la planeación y optimización de trayectorias en software de desarrollo de uso libre. En la experimentación se aplicarán los algoritmos a base de datos de uso común en la comunidad de planeación de trayectorias y con casos de problemas reportados en la literatura. Finalmente, en la evaluación se analizarán y compararán los resultados de la literatura con las técnicas tradicionales con las técnicas basadas en las RNAPs. Con esta información se conformará el estudio comparativo que permita visualizar las ventajas y desventajas de las distintas técnicas de optimización de trayectorias.

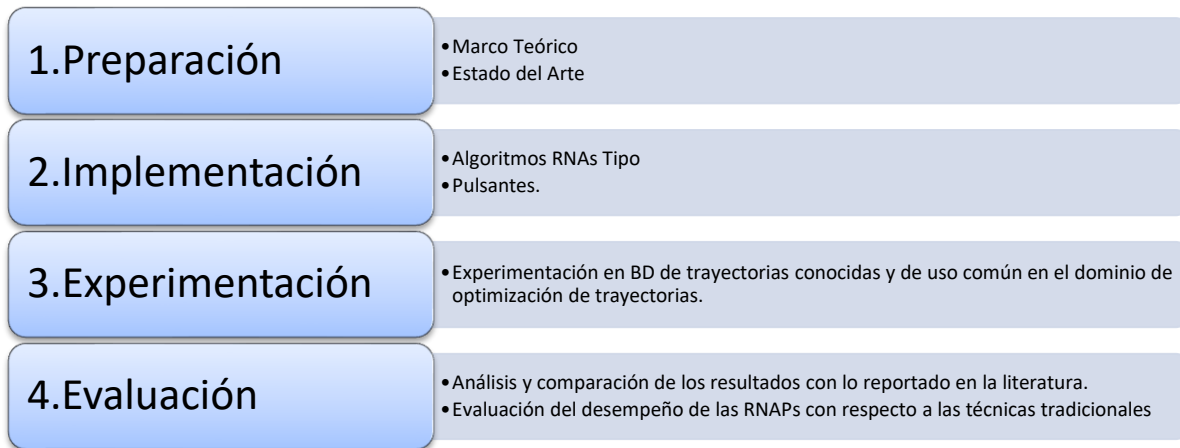


Figura 1. Etapas del enfoque de solución.

1.5 Objetivos

1.5.1 Objetivo General

Implementar, experimentar y evaluar la RNA tipo Pulso-Acoplada en el dominio de la planeación de trayectorias discretas para robots móviles.

1.5.2 Objetivos Específicos

- Estudiar los métodos convencionales empleados en el campo de la planeación de trayectorias para robots móviles.
- Estudiar el área de las RNAs tradicionales.
- Revisar y entender los problemas típicos de las técnicas tradicionales y de las RNAs en el dominio de la planeación de trayectorias.
- Implementar la RNA Pulsante aplicada al problema de la planificación de trayectorias.
- Experimentar y evaluar con bases de datos de uso común en la comunidad de Inteligencia artificial y de robótica.
- Comparar el algoritmo pulsante con respecto a otros algoritmos previos.
- Realizar un estudio comparativo con por lo menos 3 variantes de la RNA de tipo Pulso-Acoplada.

1.6 Alcances y limitaciones

1.6.1 Alcances

- Experimentar con casos de problemas planteados en la literatura.
- Realizar el estudio comparativo entre las RNAPs y las técnicas tradicionales
- basado en el tiempo de procesamiento y resultados presentados en la literatura de optimización de trayectorias.

1.6.2 Limitaciones

- Se realizarán comparaciones a partir de la propia implementación de los algoritmos, ya que en muchos casos no se reportan tiempos ni número de operaciones

- Para el desarrollo de los algoritmos y las simulaciones utilizará software de desarrollado de uso libre.
- No se busca mejorar resultados sino detectar claramente las ventajas y desventajas de la implementación de algoritmos de optimización de trayectorias con variantes de la PCNN.

1.7 Justificación

Las RNAs constituyen una técnica ampliamente explotada en los ámbitos de la clasificación, el seguimiento de señales, la aproximación de funciones, la predicción y el reconocimiento, las PCNN representan aún un nuevo objeto de estudio dentro del CENIDET en el área de optimización de trayectorias. Los parámetros utilizados por las PCNNs se encuentran en constante modificación, y han atraído la atención de los investigadores alrededor del mundo.

Esta investigación aporta información valiosa sobre el uso de las PCNN en el campo de la planeación de rutas. Se estudiará con detenimiento cada algoritmo con el propósito de obtener conocimiento útil tanto para el campo de las PCNN como para el de la planeación de trayectorias. Esta información será provechosa para el CENIDET y para investigadores del ramo, así como para futuras aplicaciones de trayectorias en la robótica móvil. Finalmente se espera que con este nuevo estudio comparativo se puedan contrastar los resultados de los modelos de PCNN con los resultados de las técnicas tradicionales reportadas en la literatura, poniendo especial atención en la gran cantidad de parámetros y en la influencia que tienen sus valores en los resultados generados.

1.8 Estructura de la tesis

Dado lo anterior, se presenta el documento de tesis, el cual se encuentra dividido en seis capítulos:

Capítulo 1. Presenta la motivación del tema de investigación, el planteamiento del problema y el enfoque de solución que se siguió durante el desarrollo del proyecto.

Capítulo 2. Muestra un panorama global y local de los trabajos relacionados a la Planeación y Optimización de Trayectorias.

Capítulo 3. Presenta los fundamentos teóricos necesarios para conocer a profundidad la Planeación y Optimización de Trayectorias las Redes Neurales Pulso Acopladas.

Capítulo 4. Describe la metodología que se siguió para realizar esta investigación, además describe las variantes de PCNN que se estudiaron e implementaron en esta investigación.

Capítulo 5. Resume las pruebas y los resultados obtenidos por cada una de las variantes de PCNN en la búsqueda de trayectorias en mapas y grafos utilizados por la comunidad científica.

Capítulo 6. Se presentan las conclusiones del presente trabajo de investigación, las aportaciones y el trabajo futuro que se podría realizar.

Capítulo 2 Marco de referencia

Se analizan investigaciones afines al tema, se muestran diversos trabajos de tesis realizados en el TecNM / CENIDET y una variedad de artículos científicos que hacen énfasis en la Planeación de Trayectorias y en las Redes Neuronales de Tercera Generación.

2.1 Antecedentes

En esta sección se presentan algunas investigaciones desarrolladas dentro del Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM / CENIDET). Dichas investigaciones están relacionadas con la planeación de trayectorias. También se presentan algunos proyectos donde se han aplicado PCNN's para resolver problemas en otras áreas de investigación.

Control embebido de un vehículo guiado automáticamente mediante Redes Neuronales Artificiales [Martínez, 2012].

En esta tesis se desarrolló un esquema de control basado en Redes Neuronales Artificiales de tipo *Recurrent High Order Neural Networks*, el cual fue aplicado en un vehículo guiado automáticamente para el seguimiento de trayectorias. Se implementó un esquema embebido que utiliza la técnica de Block Control y por otro un identificador Neuronal. Las señales digitales y análogas recibidas por el robot fueron procesadas por el sistema de control y posteriormente por el modelo de Red Neuronal Artificial, por lo cual no se utilizaron mapas para conocer el entorno a priori. Se logró que el robot fuera autónomo, no habiendo necesidad de una computadora para controlarlo. Se diseñó una ley de control basada en el identificador neuronal. Se realizaron simulaciones de seguimiento de trayectorias aplicando la ley de control diseñada y se presentaron los resultados.

Planeación de trayectorias y control por planitud diferencial de múltiples robots móviles [Escamilla, 2013]

El objetivo de esta tesis fue realizar la planeación de trayectorias utilizando la técnica de planitud de diferencia, la cual es una propiedad que posee cierta clase de sistemas dinámicos, la cual permite la transformación de ecuaciones diferenciales no lineales a un sistema algebraico de ecuaciones el cual puede ser relativamente fácil de resolver. Los controladores para las trayectorias se implementaron en un dispositivo de la marca Lego, que cuenta con las características adecuadas, en software y hardware, para la optimización de las trayectorias. Se realizaron simulaciones en LINGO, donde se utilizaron mapas 2D donde se conocía la posición del robot y de los obstáculos. Debido a que se necesitaba calcular curvas suaves en las trayectorias no se presentó un espacio discretizado en las simulaciones.

Evaluación de las técnicas SLAM disponibles en ROS [López, 2017]

Se realizó un estudio comparativo de las metodologías de localización, mapeo y navegación que ROS dispone, debido a que, en el estado del arte, no existe una guía que proporcione los criterios básicos para la elección de un conjunto de técnicas para incursionar en esta área. Para ello, se utilizaron los algoritmos Gmapping, HectorSLAM y CRSM SLAM, los cuales, permitieron realizar la localización del robot y el mapeo del entorno; mientras que, para la navegación, se utilizó el paquete *Frontier Exploration* y la teleoperación. Se realizó la construcción de mapas en 2D de entornos de interiores utilizando los algoritmos antes mencionados, los cuales se probaron en una plataforma robótica conformada por un robot con configuración diferencial, un sensor RGB-D y una computadora. Los mapas seleccionados tuvieron que pasar por un preprocesamiento: primero se recortaron los bordes excedentes, dejando un marco de aproximadamente 20 píxeles.

Evaluación del algoritmo Theta* para planeación de trayectorias [Rojas, 2018]

En este documento de tesis, se presenta la implementación de un algoritmo de planeación de trayectorias llamado Theta*, se realizó una comparativa con los algoritmos *Dijkstra* y A*. Se hicieron pruebas a nivel simulación, se implementaron con un robot de configuración diferencial, en cinco mapas diferentes, y se evaluaron veintisiete trayectorias con métricas de tiempo y distancia, donde el algoritmo Theta* generó y llegó a la meta en el 100% de los casos, a diferencia de los algoritmos Dijkstra que generó el 84% de las trayectorias y llegó a la meta el 58% y de A* que generó el 97% de las trayectorias y llegó a la meta en el 80% de los casos. En la Figura 2 se muestra la comparativa de las trayectorias trazadas por un humano con respecto de los algoritmos implementados en el trabajo de tesis.

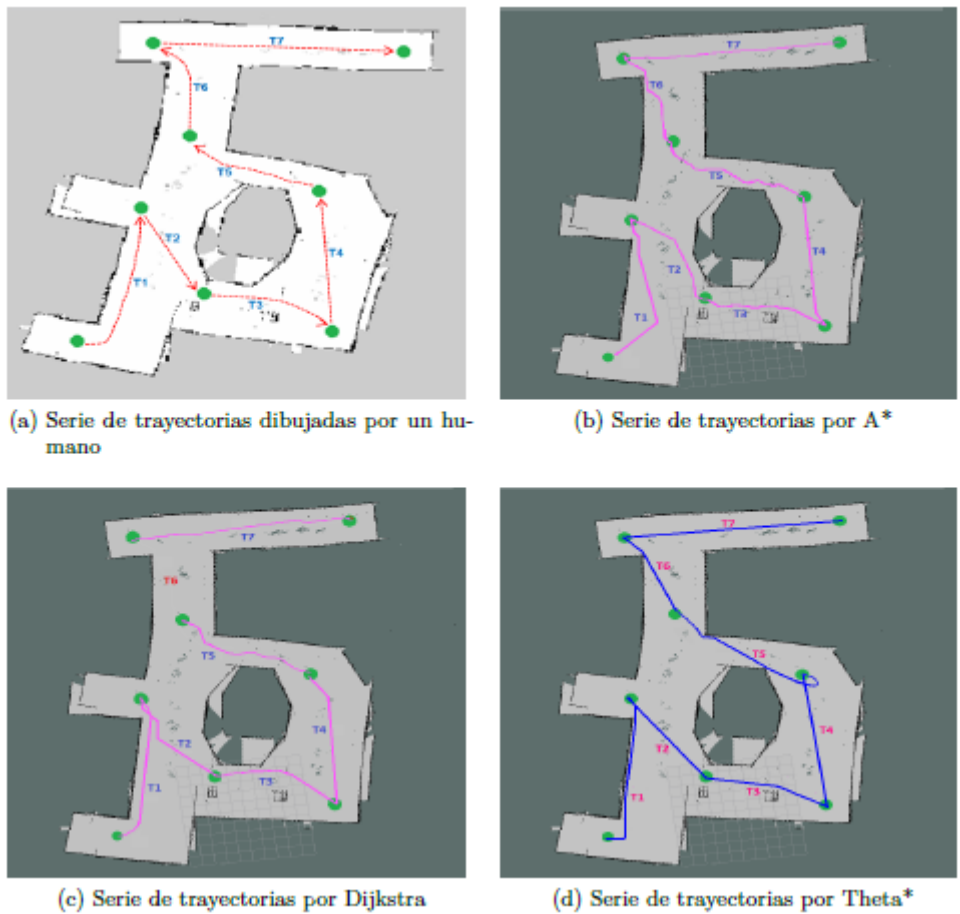


Figura 2. Comparación de trayectorias en el mapa planta alta, humano vs. Algoritmos [Rojas, 2018].

Ahora se presentan algunos trabajos, donde se han usado las variantes de PCNN aplicadas a dominios de Visión Artificial.

Implementación y Evaluación de Redes Neuronales Artificiales tipo “Pulse-Coupled Neural Networks” (PCNN) Aplicadas a Visión Artificial [Cárdenas, 2015]

La investigación consiste en un estudio comparativo sobre algoritmos de segmentación de imágenes, por medio de técnicas como PCNN y técnicas clásicas como Canny y Sobel. Se utilizaron metodologías como entropía, entropía cruzada, *PixToPix (Pixel to Pixel)* y *Pixel Correspondance Metric*, posteriormente se aplicó un mecanismo para evaluar el desempeño de las PCNN en la segmentación de imágenes. Los resultados finales mostraron que la PCNN y la ICM son comparables en el proceso de segmentación de los bordes con algoritmos tradicionales como Canny, mientras que la PCNN obtiene mejores resultados en la entropía.

Detección de Ruido Impulsivo o Gaussiano en imágenes Monocromáticas Mediante Redes Neuronales Artificiales Pulso-Acopladas [Ortiz, 2017]

Se presenta un esquema de comparación entre técnicas tradicionales y Redes Neuronales tipo Pulso Acopladas para llevar a cabo la detección y filtrado de Ruido impulsivo y Gaussiano en imágenes digitales. En la investigación se encontró que El tiempo de procesamiento con ICM es susceptible de mejora al indexar datos. Los filtros con ICM y SCM permiten el filtrado selectivo de ruido impulsivo sin afectar líneas y bordes. Adicionalmente se concluyó que la Red ICM y la Red SCM detectan más del 80% del ruido impulsivo. Finalmente fue posible determinar experimentalmente los valores de los parámetros de la Red ICM para detectar el 100% de los pixeles ruidosos ($f=0.7$, $g=0.8$, $h=20$).

Red Neural Pulsante para el Reconocimiento de Imágenes de Cerebros Sanos y No-Sanos [Córtez, 2018]

Se desarrolló un análisis en la clasificación de una neurona *Spiking Neural Networks* representada por el modelo *Izhikevich*, usada en el reconocimiento de imágenes cerebrales obtenidas de Imágenes de Resonancia Magnéticas, demostrando que la precisión de la clasificación del modelo propuesto puede alcanzar valores comparables a otras Redes Neuronales Artificiales como la *Multi-Layer Perceptron*. Por otra parte, se estimó mediante *Accuracy* que cuando existe el mismo número de imágenes tanto de cerebros sanos como de cerebros con anomalías, y las imágenes son nítidas, el aprendizaje llega a alcanzar hasta el 100%.

Mejoramiento de la Luminosidad de Imágenes Digitales del Cerebro Humano mediante Redes Neuronales Pulso-Acopladas [Aguilar, 2019]

Se propuso e implementan dos métodos basados en el Modelo de Intersección Cortical para mejorar la luminosidad en imágenes médicas del cerebro humano. El algoritmo ICM-LTM y una variante modificando algunos parámetros y procesos. En los experimentos se logró observar que ICM-LTM entregó en promedio un 10% de mejora con la métrica C y un 4% de mejora con $SSIM$; ésto en comparación con CLAHE, siendo ésta la técnica que entregó mejores resultados en promedio. Sin embargo, el aspecto visual parece favorecer a el algoritmo ICM-LTM2 ya que por la reducción de la auto-onda al modificar la Matriz de Pesos reduce los destellos de luz en los bordes, lo que podría ayudar a la hora de procesar las imágenes ya que este destello podría afectar en el procesamiento. La modificación realizada en ICM-LTM2 ayudó a obtener los resultados en menor tiempo hasta en un 25%

más rápido. La experimentación realizada muestra que los métodos propuestos son competitivos. En la Figura 3 se muestra la comparativa de los resultados de distintos algoritmos sobre imágenes luminosas.

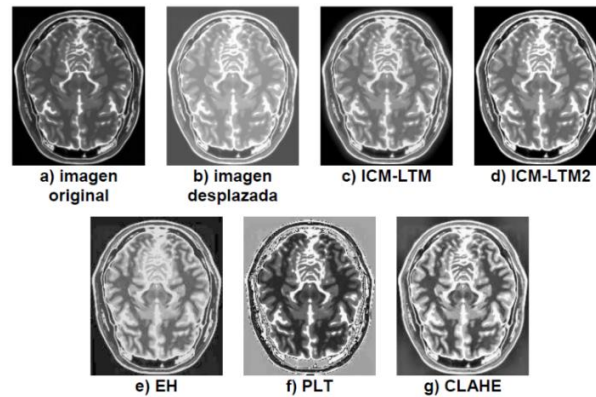


Figura 3. Resultados del mejoramiento de imágenes luminosas [Aguilar, 2019].

2.2 Estado del arte

En esta Sección se presentan los trabajos analizados en el estado del arte, los cuales tienen como objetivo en común la planificación de trayectorias, utilizando diferentes técnicas para obtener una trayectoria óptima. Esta revisión de artículos se hizo con la finalidad de encontrar material con el cual poder obtener información de referencia para desarrollar una comparación de resultados.

A new Algorithm for Finding the Shortest Paths Using PCNNs [Qu, 2007]

Se estudia el rendimiento de la auto-onda en PCNN teniendo como objetivo aplicarlo a problemas de optimización, como el problema del camino más corto. Se estudia un modelo de *Multi-Output Pulse Coupled Neural Network* (MOPCNN). Se realizan simulaciones para ilustrar el rendimiento del método propuesto. Algunas condiciones para la exploración de la auto-onda en MPCNN [Qu, 2007] se explotan mediante análisis matemático, lo que puede garantizar que las ondas exploren desde el nodo único "inicio" hasta múltiples nodos objetivo. Para verificar los resultados teóricos y la efectividad de los modelos de PCNN de tiempo discreto cuando se aplica para resolver el problema de los caminos más cortos, se han llevado a cabo varios experimentos sobre grafos simétricos. En la Figura 4 se muestra un grafo de 100 nodos asimétrico generado de manera aleatoria con el cual se realizaron los experimentos.

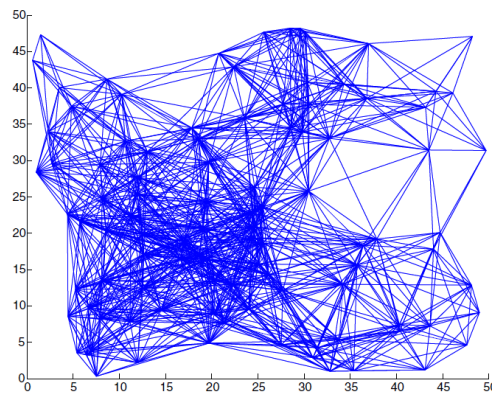


Figura 4. Grafo asimétrico generado de manera aleatoria utilizado en los experimentos [Qu, 2007].

Generación de Trayectorias para un Robot Móvil empleando Redes Neuronales [Rios, 2008]

Se propone una metodología de navegación de robots móviles en ambientes estructurados, utilizando Redes Neuronales Artificiales del tipo propagación hacia atrás, tanto para la clasificación de superficies como para la generación de trayectoria óptima. Para la generación de trayectorias se implementó una interfaz gráfica, cuyo objetivo es generar una base de datos con la cual se entrenó la Red Neuronal para que el robot móvil pudiera desplazarse desde un punto inicial a un punto final previamente establecidos. Para la generación de trayectorias no se hizo comparativa de resultados, pues la base de datos se alimentó con los sensores y esta metodología no ha sido probada con otro tipo de Red neural. En la Figura 5 se muestra una trayectoria generada por la Red Neuronal entrenada.

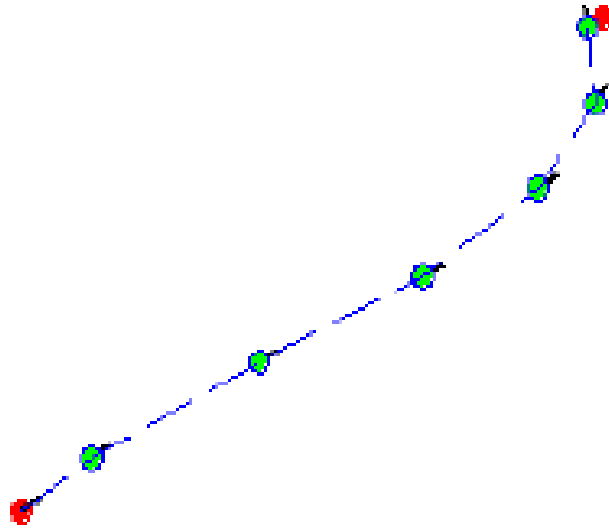


Figura 5. Trayectoria generada por la Red Neuronal [Rios, 2008].

Real-Time Robot Path Planning Based on a Modified Pulse-Coupled Neural Network Model [Qu, 2009]

Este artículo presenta la variante Modified Pulse Coupled Neural Network (MPCNN) para la planificación de rutas libres de colisiones en tiempo real de robots móviles en entornos no estacionarios. Como se muestra en la Figura 6, se propone una Red Neuronal para robots, organizada topológicamente con sólo conexiones laterales locales entre las neuronas. Se cuentan entre sus ventajas que funciona en entornos dinámicos y no requiere conocimiento previo del objetivo o del movimiento de los obstáculos. La neurona objetivo dispara primero, y luego el evento de disparo se extiende, a través de las conexiones laterales entre las neuronas, como la propagación de una onda. Los obstáculos no tienen conexión con sus vecinos. Cada neurona registra su padre, es decir, el vecino que lo hizo disparar. El camino óptimo en tiempo real es la secuencia de neuronas padres del robot al objetivo.

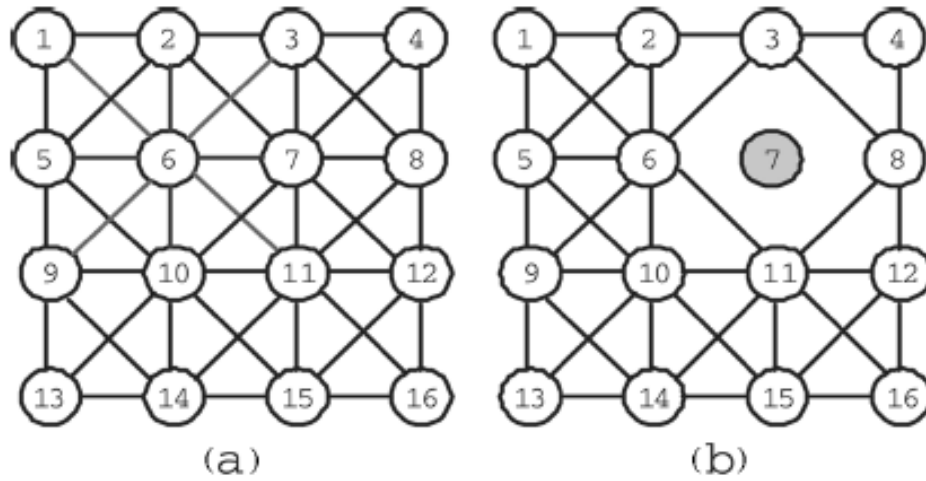


Figura 6. (a) arquitectura de Red sin barrera. (b) arquitectura con barrera en el lugar 7 [Qu, 2009].

A Modified Pulse Coupled Neural Network for Shortest-Path Problem [Wang, 2009]

Este artículo presenta un algoritmo biológico basado en Redes Neuronales Pulso Acopladas para encontrar el camino más corto en sistemas a gran escala. La neurona de inicio se dispara primero, y luego el evento de activación se extiende a través de las conexiones laterales entre las neuronas, como la propagación de una onda. Luego, la onda de espiga generada se propaga a una velocidad constante, de modo que el tiempo de viaje entre dos neuronas es proporcional a la longitud del camino entre ellas. La complejidad computacional del algoritmo sólo está relacionada con la longitud de la ruta más corta e independiente del número de rutas existentes en el gráfico. Los resultados de la simulación muestran que el método propuesto es más eficiente que el de *Dijkstra* en los sistemas a mayor escala. En la Figura 7 se muestra la comparativa de los tiempos de ejecución entre el modelo propuesto y el algoritmo Dijkstra.

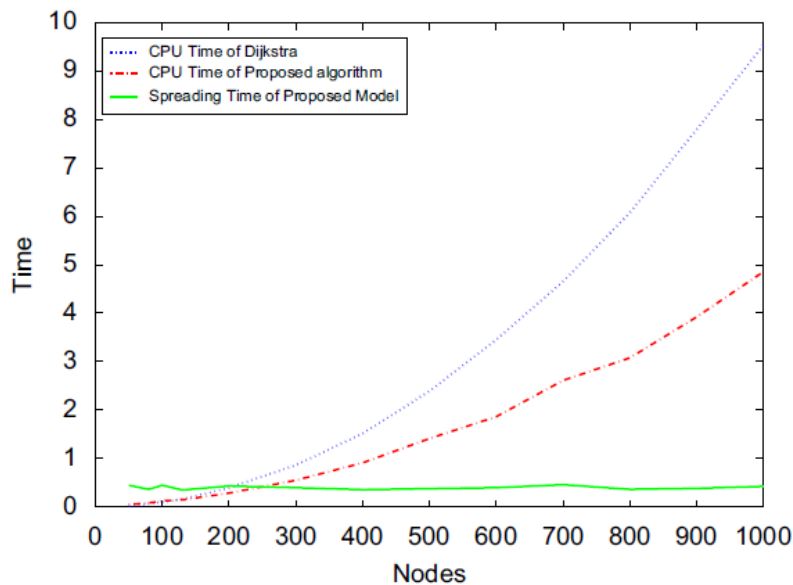


Figura 7. Comparación de los tiempos de ejecución entre el modelo propuesto y el algoritmo Dijkstra [Wang, 2009].

A Modified Pulse Coupled Neural Network for Shortest Path Computation [Sang, 2010]

En esta investigación se implementa un modelo *Dual Source Pulse Coupled Neural Network* (DSPCNN) para la búsqueda del camino más corto. La peculiaridad de este modelo radica en que puede mejorar la eficiencia computacional de los PCNN para los problemas del camino más corto. DSPCNN produce dos auto-ondas: una proviene de la neurona fuente y la otra proviene de la neurona meta. Una vez que las auto-ondas provenientes de las dos fuentes de disparo se encuentran, la DSPCNN se detiene, y luego se encuentra el camino más corto al dar marcha atrás a las dos auto-ondas. Los resultados experimentales muestran que el costo computacional sólo se relaciona con la mitad de la ruta más corta devuelta y, por lo tanto, la variante DSPCNN logra una mayor eficiencia que algunos modelos PCNN tradicionales para el cálculo de la ruta más corta. Los experimentos muestran que el método propuesto puede buscar la ruta más corta global e implicar un menor espacio de búsqueda, lo que puede ahorrar tiempo de ejecución. En la Figura 8 se muestra la comparativa del tiempo de ejecución entre la variante DSPCNN y la PCNN.

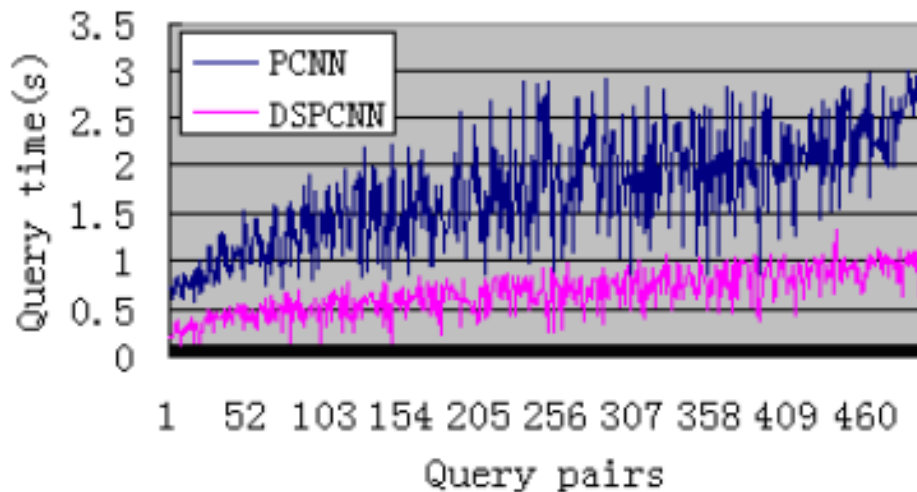


Figura 8. Comparativa del tiempo de ejecución entre la variante DSPCNN y la PCNN [Sang, 2010].

Neural Networks Based Path Planning and Navigation of Mobile Robots [Kroumov, 2011]

En este trabajo se desarrolla un algoritmo de redes neuronales para resolver el problema de planificación de ruta para robots con alguna limitante de movimiento en ambientes que se conocen a priori. El algoritmo también se puede aplicar a los robots de vuelo libre y tipo serpiente. En general, se trabajó en entornos conocidos bidimensionales, donde los obstáculos son polígonos fijos u óvalos, pero el algoritmo puede aplicarse para el caso tridimensional. Para mostrar la efectividad del algoritmo se presentan ejemplos de simulaciones. Se comparan los tiempos de ejecución de 12 simulaciones, descartando el mejor y el peor tiempo de ejecución. El tiempo de cálculo depende del número de bucles (100-1000) y el número de los vértices (20-120) un algoritmo rápido para resolver el problema de planificación de ruta para los robots de accionamiento diferencial (holonómicos). En la Figura 9 se muestran algunas simulaciones realizadas durante la investigación.

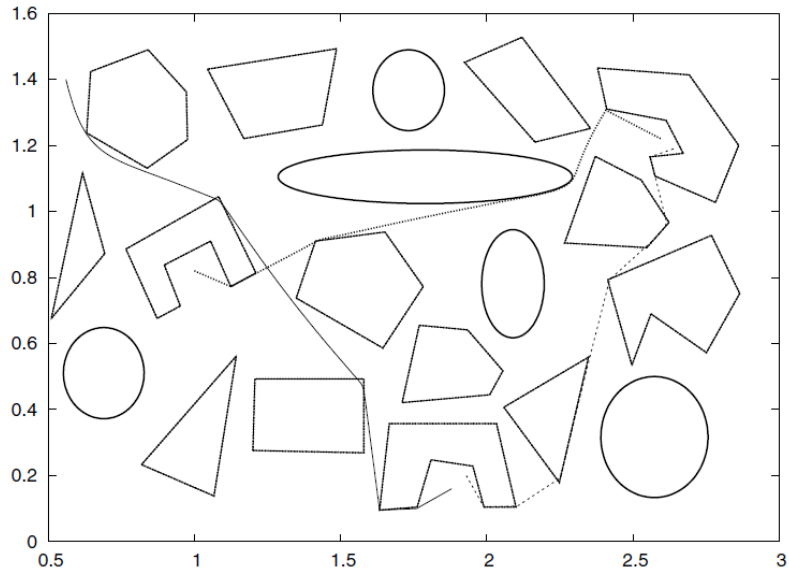


Figura 9. Simulaciones realizadas en la investigación de [Kroumov, 2011].

A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network [Zhang, 2011]

En este artículo se propone un nuevo algoritmo paralelo para resolver el problema del camino más corto usando la multiplicación de matriz, y luego se empleó una Red Neuronal Pulso Acoplada para simular el método de la multiplicación de matriz. La estrategia se puede hacer dentro de $O(\log_2 n)$, donde n representa el número de nodos. El método propuesto es determinista y garantizaría las soluciones globales. Los caminos más cortos de todos los pares se alcanzan ejecutando la Red sólo una vez. Para verificar los resultados teóricos y la efectividad del algoritmo propuesto, se llevaron a cabo varios experimentos utilizando grafos simétricos y asimétricos. En el artículo se muestran las diferencias entre el algoritmo propuesto y otros algoritmos basados en Redes Neuronales Pulso Acopladas, lo que indica que el método propuesto requiere más espacio que el método *Multi-Output Pulse Coupled Neural Network* (MOPCNN, por sus siglas en inglés) pero mucho menos espacio que el método PCNN y *Modified Pulse Coupled Neural Network* (MPCNN). Además, la complejidad temporal del algoritmo propuesto es la menor. En la Figura 10 se muestra la estructura de la neurona propuesta.

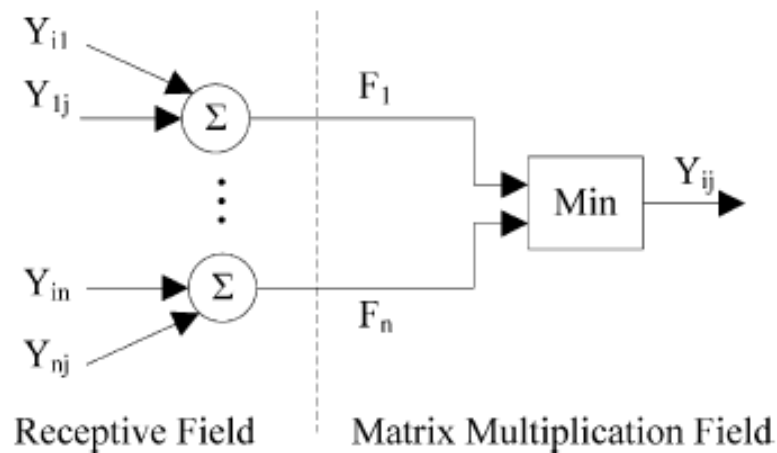


Figura 10. Estructura de la neurona para la multiplicación de matrices [Zhang, 2011].

Self-adaptive auto wave pulse-coupled neural network for shortest-path problem [Li, 2013]

En este artículo se presenta una variante de las Redes Neuronales Pulso Acopladas llamada *Self-Adaptive Auto Wave Pulse-Coupled Neural Network* (SAPCNN). Esta variante propaga la onda de forma dinámica según el estado de la Red. La onda automática generada por SAPCNN se propaga de forma adaptativa de acuerdo con el estado actual de la Red, lo que garantiza que se propague de manera más efectiva para encontrar las rutas más cortas. Esto representa una mejora, pues la mayoría de los métodos sugieren que la onda automática de los modelos PCNN debe mantener una velocidad constante para encontrar los caminos más cortos. Además, se realizan experimentos utilizando el modelo propuesto y comparándolo contra una variante de PCNN modificada y el algoritmo *Dijkstra*, aplicándolos en el problema de la búsqueda del camino más corto. Adicionalmente, se realizan pruebas donde se aplica esta variante de PCNN en el problema de la Planeación de Trayectorias.

Para comparar el rendimiento de cada algoritmo, se utilizó un grafo ponderado simétrico con diez nodos y veinte aristas. En la Figura 11 se muestra el grafo empleado para dicha comparación.

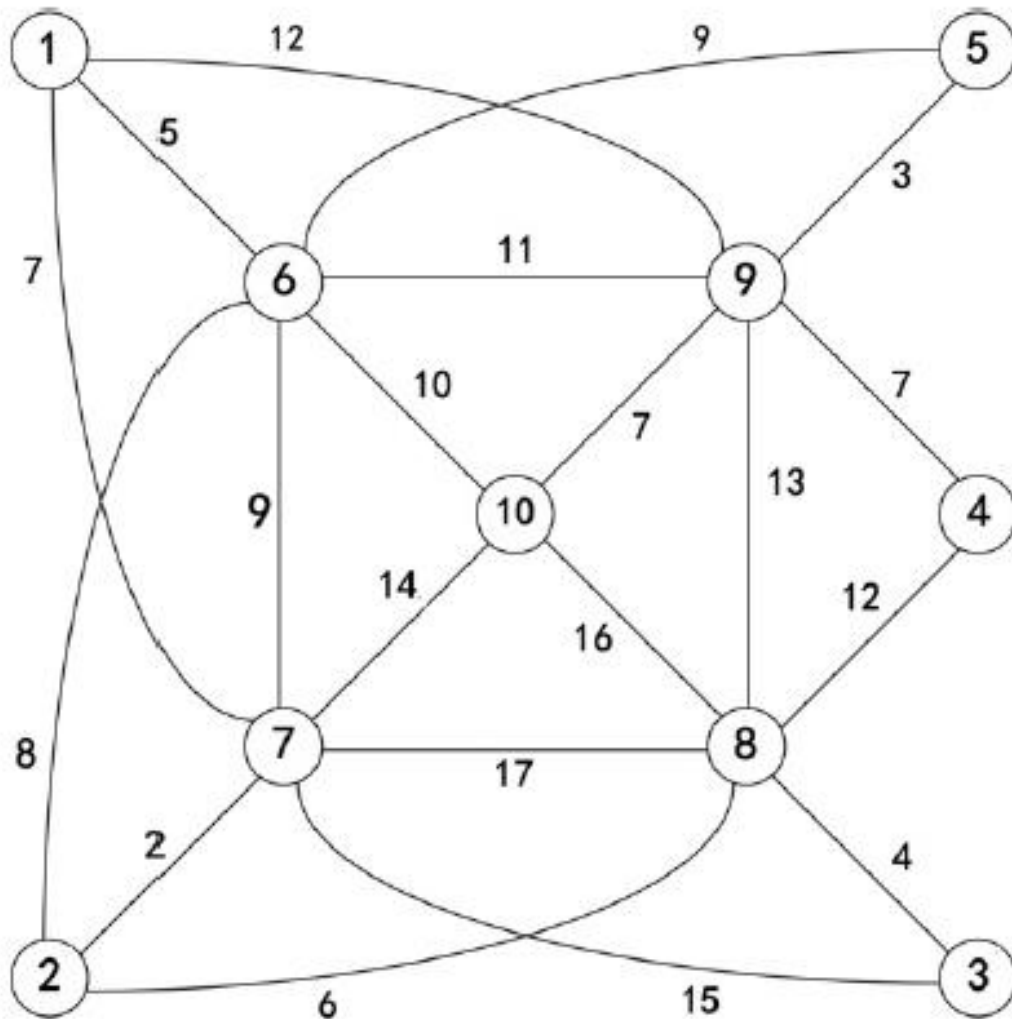


Figura 11. Gráfico ponderado simétrico con diez nodos y veinte aristas [Li, 2013].

En la Tabla 1, se presentan los resultados obtenidos, luego de realizar las búsquedas de todas las trayectorias.

Tabla 1. Los resultados muestran todos los árboles más cortos de la fig. 4 [Li, 2013].

Source node	Shortest path tree	Iteration number		Global solutions
		SAPCNN	MPCNN	
1	1→7→2→8→3, 1→9→4, 1→6→5, 1→6→10	11	23	Yes
2	2→7→1, 2→8→3, 2→8→4, 2→6→5, 2→6→9, 2→7→10	12	21	Yes
3	3→8→2→7→1, 3→8→4, 3→8→9→5, 3→8→2→6, 3→8→10	12	23	Yes
4	4→9→1, 4→8→3, 4→9→5, 4→9→6, 4→8→2→7, 4→9→10	12	29	Yes
5	5→6→1, 5→6→2, 5→9→8→3, 5→9→4, 5→6→7, 5→9→10	12	23	Yes
6	6→1, 6→2→8→3, 6→9→4, 6→5, 6→7, 6→10	10	21	Yes
7	7→2→8→3, 7→2→8→4, 7→6→5, 7→1→9, 7→10	13	25	Yes
8	8→2→7→1, 8→3, 8→4, 8→9→5, 8→2→6, 8→10	10	18	Yes
9	9→8→2, 9→8→3, 9→4, 9→5, 9→6, 9→1→7, 9→10	11	21	Yes
10	10→6→1, 10→7→2, 10→8→3, 10→9→4, 10→9→5	13	22	Yes

Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time Path Planning and an Obstacle Avoidance Scheme for Mobile Robots [Usman, 2014]

Se presenta un modelo *Guided Autowave Pulse Coupled Neural Network* (GAPCNN), el cual es un modelo de Red Neuronal Pulso Acoplada diseñado para la Planeación de Trayectorias para Robots Móviles. Dicho modelo es una mejora al modelo MPCNN de [Qu, 2009], la cual consiste en que la búsqueda no se hace de manera radial, sino que se busca en dirección hacia el objetivo. Para lograr que la búsqueda sea guiada, se utilizan los ángulos que existen entre el robot y su objetivo con respecto de su posición actual. La simulación y los resultados experimentales han demostrado que el GAPCNN tiene una mejora significativa en el tiempo con respecto al MPCNN al tiempo que conserva todas sus capacidades. Es importante tener en cuenta que los parámetros utilizados en el modelo GAPCNN son independientes del espacio de configuración y una propiedad de una neurona. Estos parámetros deben funcionar independientemente del entorno en el que se planifica la ruta. En la Figura 12 se presentan ejemplos de trayectorias generadas por el algoritmo presentado, en comparación con la generada por el MPCNN.

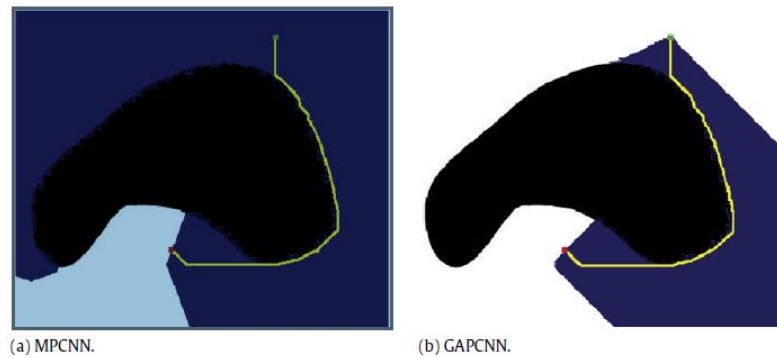


Figura 12. Comparativa de espacio de búsqueda entre la variante MPCNN y GAPCNN [Usman, 2014].

Planification of an Optimal Path for a Mobile Robot Using Neural Networks [Khouil, 2016]

El objetivo de este trabajo es calcular una ruta óptima para un móvil de 4 ruedas, en poco tiempo, siguiendo un algoritmo especializado que resuelve sistemas no lineales con múltiples restricciones. El modelo cinemático relacionado con el motor estudiado es un sistema no lineal, donde varias funciones objetivo no lineales deben optimizarse en una situación de conflicto. Para mayor eficiencia, se aprovechó el paralelismo de Redes Neuronales Artificiales, específicamente el enfoque de las *Jacobian Neural Networks* (JNN). Se generaron simulaciones en

Matlab donde se observaron cambios en las trayectorias generadas, de acuerdo con las condiciones iniciales y el número de pasos seleccionados. En la Figura 13 se presentan ejemplos de las trayectorias obtenidas en los experimentos.

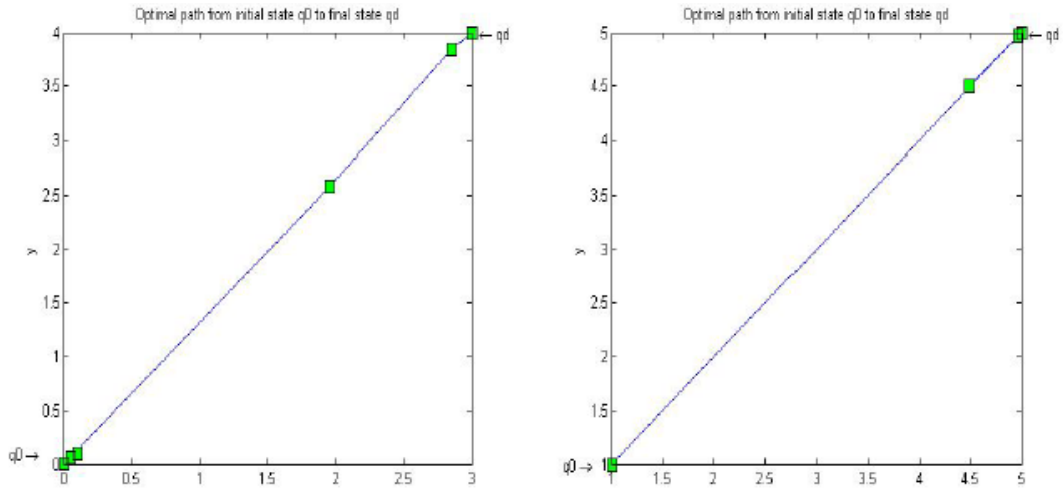


Figura 13. Ejemplos de trayectorias óptimas [Khouil, 2016].

Path planning for a mobile robot using genetic algorithm and artificial bee colony [Carballo, 2017]

Se presenta una comparación entre el algoritmo de Colonia de Abejas Artificiales y el Algoritmo Genético para problemas de Planificación de Trayectorias. Para evaluar los algoritmos, se usaron diez mapas en 2D, cada uno con diferente número de obstáculos y geometría. Los resultados se compararon de acuerdo con las siguientes mediciones: tiempo de ejecución y duración del camino. Los resultados de la simulación muestran que el algoritmo ABC es eficaz para encontrar el camino en menos tiempo. Los algoritmos mostraron resultados satisfactorios, de acuerdo con la variación de los parámetros y de acuerdo con cada mapa presentado. Finalmente, se observó una diferencia en el tiempo de ejecución de las métricas, mostrando un mejor rendimiento el algoritmo de colonia de abejas artificiales cercano al 80%. Las rutas se inician según el contexto estático y según las diferentes configuraciones de mapas, conectando el punto de partida con el objetivo. También se generó una prueba utilizando un robot móvil, recreando uno de los mapas de la Figura 14.

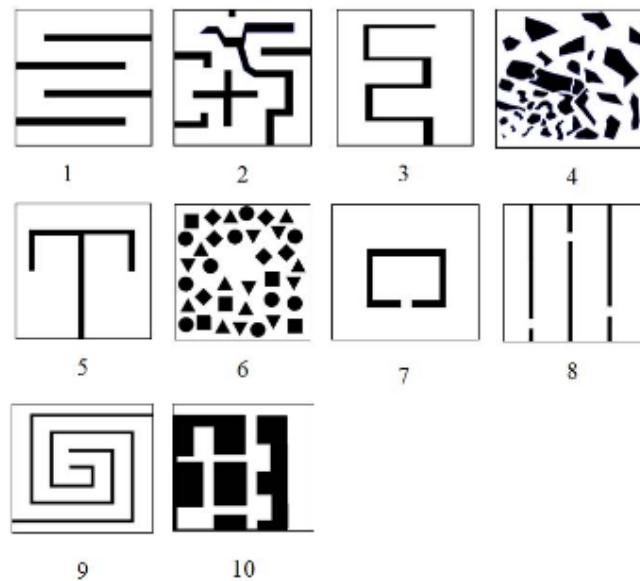


Figura 14. Mapas utilizados para evaluar la efectividad de los algoritmos [Carballo, 2017].

Red Neuronal Pulsante Explicitada para el problema del Camino más Corto [Paredes-Cano, 2018]

En este artículo realizado en TecNM / CENIDET, se presenta un trabajo donde se aplican las Redes Neuronales Artificiales de tipo Pulsante, para resolver el problema del camino más corto utilizando grafos ponderados. Además, se presenta una técnica para extraer el conocimiento generado por la Red Neuronal el cual es la trayectoria óptima encontrada. Se presenta una gráfica y una Tabla comparativa donde se muestran los resultados del desempeño de esta técnica con grafos de diferentes dimensiones, desde grafos pequeños de 5 nodos hasta grafos de 9,000 nodos. En la Figura 15 se muestra una gráfica con la relación entre el número de nodos y el tiempo de procesamiento.

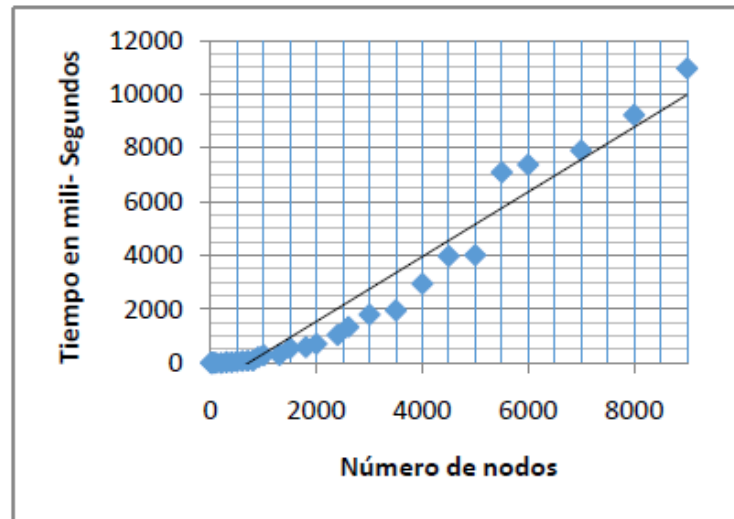


Figura 15. Tiempo de procesamiento vs número de nodos de la RNP [Paredes-Cano, 2018].

ReinforcedRimJump: Tangent-based Shortest-path Planning for Two-dimensional Maps [Yao, 2019]

Teniendo en cuenta que el camino más corto está compuesto por tangentes entre obstáculos, en esta investigación se propone un método llamado *ReinforcedRimJump* (RRJ). Dicho método, no se basa en el recorrido punto por punto, sino que obtiene el camino más corto en un entorno estático 2D al encontrar tangentes múltiples veces entre obstáculos. La primera mejora de este método es el preprocesamiento de tangentes, lo que hace que el método tenga un costo de tiempo menor que los métodos tradicionales. La segunda mejora de RRJ es la segmentación de bordes, que permite usar RRJ cuando el objetivo está en la depresión del obstáculo. RRJ introduce el concepto de "sub-edge", lo que permite que la planificación de ruta se basa en la tangente que no se ve limitada por la forma del obstáculo (el gráfico de visibilidad se limita a los obstáculos en forma de polígono) y el preprocesamiento de tangentes para disminuir el costo de búsqueda de tangentes en línea. Para verificar las ventajas teóricas de RRJ, se realizan algunos experimentos comparativos con varios mapas. Los resultados experimentales muestran que RRJ siempre puede encontrar el camino más corto en el menor tiempo. Además, el costo de tiempo de RRJ es insensible al tamaño del mapa en comparación con otros métodos. Los resultados experimentales presentados en este documento demuestran que RRJ cumple con las expectativas teóricas. En la Figura 16 se muestra uno de los mapas con los que se realizaron pruebas.



Figura 16. Mapa de pruebas de [Yao, 2019]

Path Planning for Minimizing the Expected Cost Until Success [Muralidharan, 2019]

Se considera el problema de la planificación de rutas en un grafo para minimizar el costo esperado hasta tener éxito. El problema es *NP-hard* y puede plantearse como un Proceso de Decisión de Markov de un problema del camino más corto estocástico. Se propuso un algoritmo de Planeación de Trayectorias basado en un *Framework* de Teoría de Juego que proporciona una solución subóptima al problema de forma asintótica. Además, se propusieron dos estrategias subóptimas no miopes que encuentran una buena solución de manera eficiente. Finalmente, a través de resultados numéricos, se mostró que los planificadores de trayectoria propuestos superan significativamente la heurística ingenua y codiciosa. Se consideraron dos escenarios en las simulaciones: el de un *rover* en Marte buscando un objeto para estudio científico, y el de un escenario de planificación de ruta realista para un robot que busca conectividad. Los resultados indicaron una reducción significativa en la distancia recorrida esperada (por ejemplo, una reducción del 35% para la planeación de trayectoria en el escenario de conectividad), al usar los enfoques propuestos. En la Figura 17 se muestra una simulación de la planeación de trayectorias con uno de los métodos propuestos.

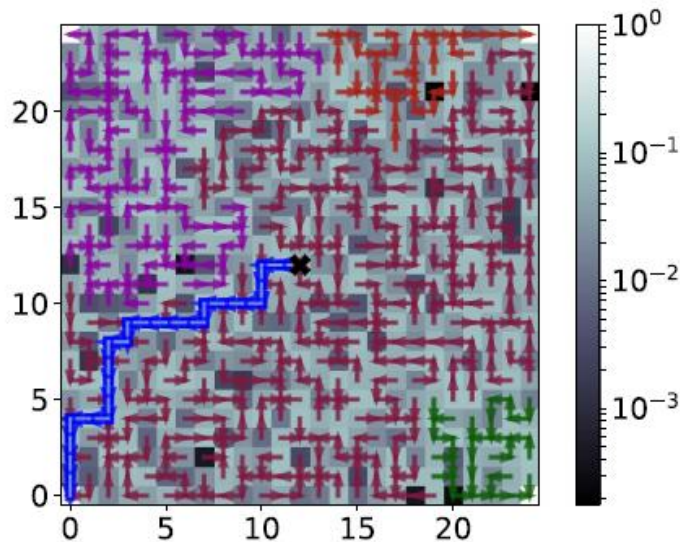


Figura 17. Ejemplo de planeación de trayectoria dentro de un laberinto [Muralidharan, 2019].

2.3 Discusión del estado arte

La Planeación de Trayectorias para Robots Móviles es un campo donde se han presentado diversas alternativas de solución en los últimos años. Entre estas alternativas, [Ríos, 2008] utiliza un robot equipado con un anillo de sensores ópticos para generar una base de datos y entrenar una Red Neuronal que clasifica superficies comunes dentro de un ambiente robotico. [Kroumov, 2011] Utiliza el enfoque de Campos Potenciales y presenta una Red Neuronal Artificial que puede calcular la mejor ruta de manera exitosa. En [Khouil, 2016] se utilizan la propiedad paralela de las *Jacobian Neural Networks* para calcular una ruta óptima para un móvil de 4 ruedas, en poco tiempo, resolviendo sistemas no lineales con múltiples restricciones.

Los trabajos de [Ríos, 2008] y [Kroumov, 2011] y [Khouil, 2016], son un claro ejemplo de que diferentes variantes Redes Neuronales Artificiales presentan resultados satisfactorios. Por ejemplo, la Red Neuronal del tipo de Propagación Hacia Atrás, la Red Ponderada, y la Red Jacobiana, pueden ser una herramienta útil dentro del campo de la Planeación de Trayectorias para Robots Móviles.

Las Redes Neuronales Pulso Acopladas son algoritmos que se han presentado como una alternativa efectiva en el campo de la Optimización y Planeación de Trayectorias. [Qu, 2007] presenta una variante de PCNN llamada *Multi-Output Pulse Coupled Neural Network* para resolver el problema del camino más corto. La variante MOPCNN, utiliza las propiedades de su auto-onda para calcular el camino más corto del nodo de inicio a múltiples nodos. [Qu, 2009] propone la variante *Modified Pulse Coupled Neural Network* para la Planeación de Trayectorias para Robots Móviles en espacios bidimensionales discretos. Esta variante utiliza la auto-onda para explorar el entorno a través de una malla de nodos los cuales sólo pueden conectarse con sus vecinos más cercanos.

Por otro lado, [Wang, 2009] presenta la variante MPCNN para resolver el Problema del Camino más Corto utilizando una auto-onda con una velocidad de propagación constante de modo que el tiempo de viaje entre dos neuronas es proporcional a la longitud del camino entre ellas. [Sang, 2010] presenta una variante *Dual Source Pulse Couplet Neural Network* para la búsqueda del Camino más Corto. DSPCNN produce dos auto-ondas: una proviene de la neurona de inicio y otro de la neurona meta. Los resultados experimentales muestran que el costo computacional sólo se relaciona con la mitad de la ruta obtenida. [Zhang, 2011] propone un algoritmo para resolver el problema del Camino más Corto que utiliza una PCNN para simular el método de la multiplicación de matrices.

En [Li, 2013] se presenta la variante *Self-Adaptive Auto Wave Pulse-Coupled Neural Network*, la cual cuenta con una velocidad de propagación de auto-onda dinámica. En otras variantes la velocidad es constante, sin embargo, en esta variante la velocidad es dinámica, lo que permite mayor eficiencia para calcular el camino más corto. [Usman, 2014] desarrolla la variante *Guided Autowave Pulse Coupled Neural Network*, esta variante se basa en el trabajo de [Qu, 2009], sin embargo, propone un mecanismo que le permite reducir el espacio de búsqueda dirigiendo la auto-onda en dirección al objetivo. Finalmente, [Paredes-Cano, 2018] presento la variante basada en Auto-Wave Neural Network de [Ma, 2011], proponiendo un mecanismo para recuperar la trayectoria calculada una vez terminada la búsqueda. Además, de una velocidad de auto-onda adaptativa en la primera iteración de la búsqueda.

Los trabajos presentados en [Qu, 2007], [Qu, 2009], [Wang, 2009], [Sang, 2010], [Zhang, 2011], [Li, 2013], [Usman, 2014] y [Paredes-Cano, 2018] son muestra de que el paradigma de las Redes Pulso Acopladas puede dar buenos resultados además de que servirán para llevar a cabo un estudio comparativo entre las Redes Neuronales y las técnicas de optimización de trayectorias tradicionales.

Por otro lado, se encontraron otros enfoques que se han aplicado con éxito en el campo de la Planeación de Trayectorias. [Carballo, 2017] se presenta una Colonia de Abejas Artificiales y la compara Oliver Eder Guillermo Espinosa Meneses

con un Algoritmo Genético utilizando un set de mapas bidimensionales. [Muralidharan, 2019] un algoritmo de Planeación de Trayectorias basado en un *Framework* de Teoría de Juego que proporciona una solución subóptima al problema de forma asintótica. [Yao, 2019] propone un método llamado *ReinforcedRimJump* que no se basa en el recorrido punto por punto, sino que obtiene el camino más corto en un entorno estático 2D al encontrar tangentes múltiples veces entre obstáculos.

En resumen:

- a) El tema es actual y de interés para la comunidad científica, ya que se encontraron 14 artículos entre trabajos relacionados y el estado del arte, siendo tres de ellos publicados en 2018 y 2019.
- b) Se revisaron trabajos relacionados con la Optimización y la Planeación de Trayectorias que emplearon diversas técnicas y algoritmos para calcular una trayectoria óptima, en particular las Redes Neuronales Pulso Acopladas.
- c) Dentro de los modelos de PCNN revisados, se encontraron modelos con una velocidad constante de propagación de la auto-onda, y modelos con una velocidad dinámica de propagación de auto-onda, los cuales obtienen la solución de manera más eficiente.
- d) No se encontró un trabajo donde se presente un modelo de PCNN con una auto-onda dinámica que pueda alcanzar por lo menos a una neurona en cada iteración, siendo esto último lo que se presenta en esta tesis.

A continuación, se muestra una tabla comparativa con las características más importantes de cada uno de los artículos revisados en el estado del arte y trabajos relacionados.

Tabla 2. Comparativa del estado del arte y trabajos relacionados.

Nombre	Objetivo	Técnicas empleadas	Pruebas y resultados	Observaciones
<i>A new algorithm for finding the shortest paths using PCNNs</i> [Qu, 2007]	Utilizar la característica de auto-onda de las PCNN para resolver el Problema del Camino más Corto de un nodo a múltiples nodos.	<i>Multi-Output Pulse Coupled Neural Network</i> (MOPCNN).	Se realizaron experimentos utilizando grafos de diferentes magnitudes.	El modelo permite encontrar el camino más corto de un nodo a múltiples nodos, pero no se hace una comparativa con otras variantes.
Generación de trayectorias para un robot móvil empleando redes neuronales [Ríos, 2008]	Desarrollar una metodología de navegación para robots móviles, tanto para la clasificación de superficies como para la generación de trayectoria óptima.	Redes Neuronales Artificiales del tipo propagación hacia atrás.	Para la generación de trayectorias se implementó una interfaz gráfica, cuyo objetivo es generar una base de datos con la cual se entrenará la Red Neuronal. De las superficies clasificadas se obtuvo una eficiencia del 97%.	Muestra el potencial de las redes neuronales del tipo propagación hacia atrás, aplicadas en la generación de trayectorias.
<i>Real-Time Robot Path Planning Based on a Modified Pulse-Coupled Neural Network Model</i> [Qu, 2009]	Presentar un modelo de <i>Modified Pulse Coupled Neural Network</i> (MPCNN) para la planificación de rutas libres de colisiones en tiempo real de robots móviles en entornos no estacionarios.	<i>Modified Pulse-Coupled Neural Network</i> (MPCNN).	Se generan simulaciones aplicando el modelo propuesto, comparándolo con el modelo <i>Dynamic Wave Expansión Neural Network</i> (DWENN).	El enfoque propuesto utiliza la característica de transmisión de impulsos paralelos de PCNN para encontrar rápidamente el camino más corto.
<i>A Modified Pulse Coupled Neural Network for Shortest-Path Problem</i> [Wang, 2009]	Utilizar una PCNN para encontrar el camino más corto en sistemas a gran escala.	<i>Modified Pulse Coupled Neural Network</i> (MPCNN).	Los resultados de la simulación muestran que el método propuesto es más eficiente que el de <i>Dijkstra</i> en los sistemas a mayor escala.	La variante presenta una velocidad de propagación constante de la auto-onda. El tiempo de viaje entre dos neuronas es proporcional a la longitud del camino entre ellas.

Tabla 2. Comparativa del estado del arte y trabajos relacionados (continua).

Nombre	Objetivo	Técnicas empleadas	Pruebas y resultados	Observaciones
<p><i>A Modified Pulse Coupled Neural Network for Shortest Path Computation</i> [Sang, 2010]</p>	<p>Utilizar una variante de PCNN con dos auto-ondas. Una viajara desde el nodo de inicio u otra desde el nodo final.</p>	<p><i>Dual Source Pulse Couplet Neural Network.</i></p>	<p>Los resultados experimentales muestran que el costo computacional sólo se relaciona con la mitad de la ruta más corta devuelta y, por lo tanto, la variante DSPCNN logra una mayor eficiencia que algunos modelos PCNN tradicionales para el cálculo de la ruta más corta.</p>	<p>El método propuesto puede buscar la ruta más corta global e implicar un menor espacio de búsqueda, lo que puede ahorrar tiempo de ejecución.</p>
<p><i>Neural Networks Based Path Planning and Navigation of Mobile Robots</i> [Kroumov, 2011]</p>	<p>Presentar un algoritmo rápido para resolver el problema de planificación de ruta para los robots holonómicos.</p>	<p>Redes Neuronales.</p>	<p>Para mostrar la efectividad de lo propuesto en este algoritmo de papel, se dan varios ejemplos de simulaciones.</p>	<p>El modelo presentado obtiene las rutas exitosamente, no se presentan los resultados de manera tabular.</p>
<p><i>A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network</i> [Zhang, 2011]</p>	<p>Propone un nuevo algoritmo paralelo para resolver el problema del camino más corto usando una Red Neuronal Pulso Acoplada para simular el método de multiplicación de matriz.</p>	<p>Red Neuronal Pulso Acoplada.</p>	<p>Se llevaron a cabo varios experimentos utilizando grafos simétricos y asimétricos.</p>	<p>Los caminos más cortos de todos los pares se alcanzan ejecutando la Red sólo una vez.</p>
<p><i>Self-adaptive auto wave pulse-coupled neural network for shortest-path problem</i> [Li, 2013]</p>	<p>Propone una variante de PCNN que pueda resolver el problema del camino más corto eficientemente. Utiliza una velocidad dinámica de la auto-onda.</p>	<p><i>Self-Adaptive Auto Wave Pulse-Coupled Neural Network (SAPCNN).</i></p>	<p>Se realizan experimentos utilizando el modelo propuesto y comparándolo con una variante de PCNN modificada y el algoritmo <i>Dijkstra</i>.</p>	<p>La variante tiene una velocidad de propagación de velocidad dinámica, que le permite calcular una trayectoria en menos iteraciones que otras variantes.</p>

Tabla 2. Comparativa del estado del arte y trabajos relacionados (continua).

Nombre	Objetivo	Técnicas empleadas	Pruebas y resultados	Observaciones
Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time Path Planning and an Obstacle Avoidance Scheme for Mobile Robots [Usman, 2014]	Proponer una variante de PCNN para la Planeación de Trayectorias con una auto-onda que se propague de manera guiada hacia el objetivo.	<i>Guided Autowave Pulse Coupled Neural Network (GAPCNN).</i>	Los resultados experimentales han demostrado que el GAPCNN tiene una mejora significativa en el tiempo con respecto al MPCNN al tiempo que conserva todas sus capacidades.	La variante reduce el espacio de búsqueda por lo que se tarda menos calcular una trayectoria. Sin embargo, utiliza muchas heurísticas en su mecanismo.
Planification of an Optimal Path for a Mobile Robot Using Neural Networks [Khouil, 2016]	Calcular una trayectoria óptima para un móvil de 4 ruedas, siguiendo un algoritmo que resuelve sistemas no lineales con múltiples restricciones.	<i>Jacobian Neural Networks (JNN).</i>	Se observaron cambios en las trayectorias generadas, de acuerdo con las condiciones iniciales y el número de pasos seleccionados.	Para mayor eficiencia, se aprovechó el paralelismo de Redes Neuronales Artificiales.
Path planning for a mobile robot using genetic algorithm and artificial bee colony [Carballo, 2017]	Se presenta una comparación entre el algoritmo de Colonia de Abejas Artificiales y el Algoritmo Genético para problemas de planificación de caminos.	Colonia de Abejas Artificiales y el Algoritmo Genético.	Se observó una gran diferencia en el tiempo de ejecución de las métricas, mostrando un mejor rendimiento el algoritmo de colonia de abejas artificiales.	Los algoritmos mostraron resultados satisfactorios, de acuerdo con la variación de los parámetros y de acuerdo con cada mapa presentado.
Red Neuronal Pulsante Explicitada para el problema del Camino más Corto [Paredes-Cano, 2018]	Presenta una variante del modelo AWNN, proponiendo una manera de extraer la ruta obtenida.	<i>Auto-Wave Neural Network Explicit.</i>	Se presenta una comparativa donde se muestran los resultados del desempeño con grafos de diferentes dimensiones, desde grafos pequeños de 5 nodos hasta grafos de 9000 nodos.	Se utiliza una auto-onda dinámica en la primera iteración.

Tabla 2. Comparativa del estado del arte y trabajos relacionados (continua).

Nombre	Objetivo	Técnicas empleadas	Pruebas y resultados	Observaciones
<p><i>ReinforcedRimJump: Tangent-based shortest-path planning for two-dimensional maps</i> [Yao, 2019]</p>	<p>propone un método llamado <i>ReinforcedRimJump</i> (RRJ) que no se basa en el recorrido punto por punto, sino que obtiene el camino más corto en un entorno estático 2D al encontrar tangentes múltiples veces entre obstáculos.</p>	<p><i>ReinforcedRimJump</i> (RRJ).</p>	<p>resultados experimentales muestran que RRJ siempre puede encontrar el camino más corto en el menor tiempo. Además, el costo de tiempo de RRJ es insensible al tamaño del mapa en comparación con otros métodos.</p>	<p>RRJ introduce el concepto de "<i>sub-edge</i>", lo que permite que la planificación de ruta basada en la tangente que no se ve limitada por la forma del obstáculo.</p>
<p><i>Path Planning for Minimizing the Expected Cost Until Success</i> [Muralidharan, 2019]</p>	<p>Se propuso un algoritmo de Planeación de Trayectorias basado en un <i>Framework</i> de Teoría de Juego que proporciona una solución subóptima al problema de forma asintótica.</p>	<p><i>Framework</i> de Teoría de Juego y dos estrategias subóptimas no miopes.</p>	<p>Los resultados indicaron una reducción significativa en la distancia recorrida esperada al usar los enfoques propuestos.</p>	<p>Se mostró que los planificadores de trayectoria propuestos superan significativamente la heurística ingenua y codiciosa.</p>

Capítulo 3 Marco teórico

En este Capítulo se detallan de los conocimientos generales necesarios para la mejor comprensión de la terminología de este proyecto. Se abordarán temas como Robótica Móvil, Planeación de Trayectorias, Optimización de Trayectoria y las Redes Neuronales Artificiales de tipo Pulsante. Adicionalmente se buscará situar adecuadamente el contexto en el que sitúa el proyecto.

3.1 Robótica Móvil

Un robot móvil se define como un sistema electromecánico capaz de desplazarse de forma autónoma de un punto a otro en un determinado espacio de trabajo (nave industrial, laboratorio, hogar, corredor, etc.). Se entiende como autonomía de un robot móvil, al dominio que tiene éste para determinar su curso de acción, mediante su propio proceso de razonamiento (basado en sensores que le permiten percibir el espacio de trabajo), en lugar de seguir una secuencia fija de instrucciones. [Ortigoza, 2012].

A principios de los años sesenta se introducen en la industria, de modo significativo, los robots manipuladores como un elemento más del proceso productivo. Esta proliferación, motivada por la amplia gama de posibilidades que ofrecía, suscitó el interés de los investigadores para lograr manipuladores más rápidos, precisos y fáciles de programar. La consecuencia directa de este avance originó un nuevo paso en la automatización industrial, que flexibilizó la producción con el nacimiento de la noción de célula de fabricación robotizada.

Frecuentemente fue necesario modificar el espacio de trabajo, de tal manera que todo quedara un área accesible para el manipulador, lo cual podría resultar imposible a medida que la célula sufría progresivas ampliaciones. Una solución a este problema se logra al desarrollar un vehículo móvil sobre rieles para proporcionar un transporte eficaz de los materiales entre las distintas zonas de la cadena de producción. De esta forma, aparecen en los años ochenta los primeros vehículos guiados automáticamente. Una mejora con respecto a su concepción inicial estriba en la sustitución de los rieles como referencia de guiado en la navegación por cables enterrados, reduciéndose, con ello, los costes de instalación.

La posibilidad de estructurar el entorno industrial permite la navegación de vehículos con una capacidad sensorial y de razonamiento mínimas. De este modo, la tarea se ordena en una secuencia de acciones en la que a su término el vehículo supone que ha alcanzado el objetivo para el que está programado. Ante cualquier cambio inesperado en el área de trabajo que afecte el desarrollo normal de la navegación, el sistema de navegación del vehículo se encontrará imposibilitado para ejecutar acciones alternativas que le permitan reanudar su labor. Sin embargo, por sus potenciales aplicaciones fuera del ámbito industrial, donde resulta costoso o imposible estructurar el entorno, se les dotó, en la búsqueda de un vehículo de propósito general apto para desenvolverse en cualquier clase de ambiente, de un mayor grado de inteligencia y percepción. Así en los años noventa surge el robot móvil [Lozano-Pérez, 1990].

Los robots móviles se pueden clasificar por el tipo de locomoción utilizado. En general, los tres medios de movimiento son: por ruedas, por patas y orugas. Cabe señalar que, aunque la locomoción por patas y orugas han sido ampliamente estudiadas, el mayor desarrollo se presenta en los robots móviles con ruedas. Dentro de los atributos más relevantes de los robots con ruedas, destacan su eficiencia en cuanto a energía en superficies lisas y firmes, a la vez que no causan desgaste en la superficie donde se mueven y requieren un número menor de partes las cuales son de menor complejidad, en comparación con los robots de patas y de orugas, lo que permite que su construcción sea más sencilla [Silva, 2007].

Sus numerosas aplicaciones como: exploración planetaria, minera marítima, reconocimiento de terreno, inspección, vigilancia, misiones de búsqueda, rescate de personas, limpieza de desechos peligrosos en ambientes diversos, asistencia médica, entretenimiento, entre otros. Con el objetivo de ser una

herramienta más eficaz, los robots móviles se complementan con otros robots (tal es el caso de los que incorporan un brazo manipulador), siendo utilizados como elementos de seguridad en la inspección y manejo de paquetes sospechosos, neutralización de dispositivos explosivos, manipulación de productos químicos peligrosos o materiales radioactivos, entre otras tareas [Ortigoza, 2012].

3.2 Planeación de Trayectorias

La Planeación de Trayectorias para los robots móviles es un problema fundamental en el campo del control de vehículos no tripulados. El propósito del planificador de trayectorias es calcular una ruta desde la posición de inicio del vehículo hasta el objetivo que se debe alcanzar. Es importante mencionar que es posible referirse a las trayectorias con sinónimos como caminos o rutas. La principal preocupación de la planificación de rutas es calcular rutas sin colisiones. Otro tema de igual importancia es calcular un camino realizable y, si es posible, que sea el camino más corto u óptimo, llevando el vehículo a la posición final.

Incluso para los problemas de planificación de movimiento en el espacio bidimensional, los algoritmos completos existentes que garantizan una solución a menudo requieren una gran cantidad de memoria. En algunos casos, pueden requerir un tiempo de cálculo prolongado. Por otro lado, los algoritmos heurísticos rápidos pueden no encontrar una solución, incluso si ésta existe [Kroumov, 2011].

Los planificadores de trayectorias pueden clasificarse en general como:

1. Completos.
2. Heurísticos.

Los planificadores de trayectorias completos pueden requerir tiempos de cálculo largos, pero pueden encontrar una solución si hay una o probar que no hay ninguna. Los planificadores de rutas heurísticos son rápidos, pero a menudo no logran encontrar una solución, incluso si ésta existe.

Los planificadores de trayectorias se pueden clasificar en cuatro categorías [Latombe, 1991].

1. *Skeleton*.
2. *Cell decomposition*.
3. *Subgoal graph*.
4. *Potential field*.

En el enfoque *skeleton*, el espacio libre está representado por una Red de caminos unidimensionales llamada esqueleto. La solución consiste en mover el robot a un punto en el esqueleto desde la configuración de inicio hasta el objetivo, y conectando los dos puntos a través de caminos en el esqueleto. El enfoque es intuitivo para problemas bidimensionales, pero se vuelve más difícil de implementar para mayores grados de libertad.

Algoritmos basados en el gráfico de visibilidad, en el diagrama de Voronoi y los basados en la proyección de los límites de obstáculos, son ejemplos del enfoque *skeleton*. En la Figura 18 se puede apreciar un ejemplo del diagrama de Voronoi. En el algoritmo de gráfico de visibilidad, la planeación de la ruta se logra al encontrar una ruta a través de un grafo, que conecta los vértices de las regiones prohibidas (obstáculos) y la ruta generada es casi óptima.

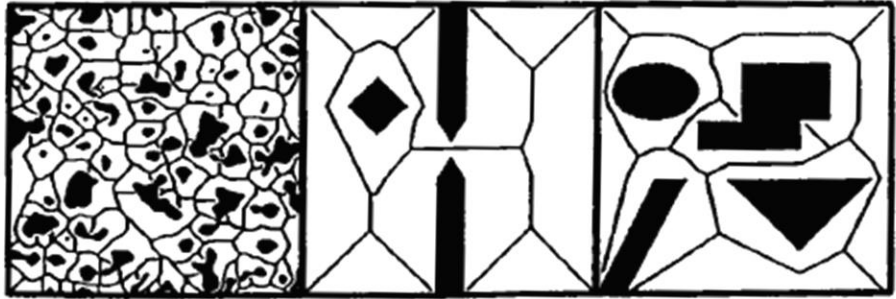


Figura 18. Representación de espacio de trabajo skeleton [Barraquand, 1992].

En el enfoque de *cell decomposition*, el espacio libre se representa como una unión de células. Una secuencia de células comprende una ruta de solución. Los obstáculos tienen forma poligonal, lo que da como resultado un árbol cuádruple con cuadrantes de hojas de tamaño mínimo a lo largo de los bordes del polígono, pero el árbol cuadrangular es grande, y el número de hojas en el árbol es proporcional al perímetro del polígono, ésto hace que se consuma memoria.

En el enfoque del gráfico de subgrupos, los objetivos secundarios representan las configuraciones clave que se espera sean útiles para encontrar caminos libres de colisiones. Un planificador global genera y mantiene un grafo de sub-objetivos, y se utiliza un planificador local simple para determinar la accesibilidad entre los objetivos secundarios. Este enfoque de planificación de dos niveles fue reportado por primera vez por Faverjon & Tournassoud (1987) y ha resultado ser uno de los métodos de planificación de ruta más efectivos [Kroumov, 2011].

El método *potential field* y sus aplicaciones para la planificación de rutas para robots móviles autónomos ha sido ampliamente estudiado. En la Figura 19 se ilustra el concepto básico de este tipo de métodos, los cuales consisten en llenar el espacio de trabajo con un campo de potencial artificial, en el que el robot se siente atraído por la posición del objetivo, al mismo tiempo que es rechazado por los obstáculos [Latombe, 1991].

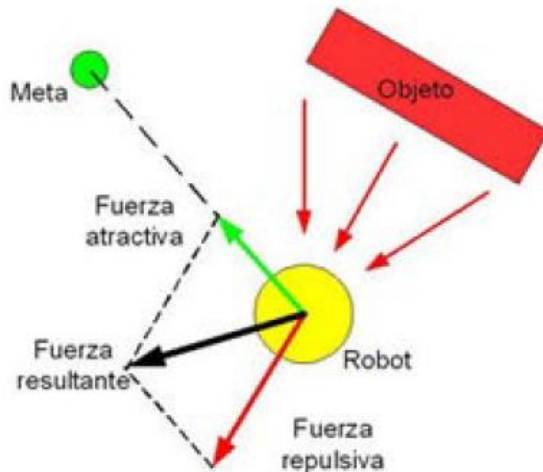


Figura 19. Modelado de fuerzas en un campo potencial [González, 2005].

Es bien sabido que con los campos potenciales es posible construir planificadores de movimiento bastante eficientes y relativamente fiables [Latombe, 1991]. Pero estos métodos suelen ser incompletos

y pueden no encontrar un camino libre, incluso si existe, porque pueden quedar atrapados en un mínimo local.

Es conveniente destacar que la mayoría de las investigaciones relacionadas con el uso de Redes Neuronales aplicadas en la planificación de trayectorias, hacen uso del enfoque *skeleton*. Los artículos que están relacionados con esta investigación utilizan grafos ponderados para la planeación de trayectorias. Por esta razón, se determinó que el enfoque más apropiado para este proyecto es el enfoque *skeleton* con el uso de grafos ponderados.

3.3 Optimización de Trayectorias

El dominio de la optimización combinatoria se compone de problemas donde el conjunto de posibles soluciones es discreto. Su objetivo consiste en encontrar la mejor solución posible o solución óptima, es decir, aquella que minimice o maximice una función objetivo.

Dentro de la optimización combinatoria se encuentran los problemas de optimización de trayectorias cuyo propósito es minimizar el costo generado por ir de un punto a otro. Existen técnicas exhaustivas para resolver dichos problemas como: búsqueda en profundidad y búsqueda en anchura [Mathivet, 2015], pero a medida que la complejidad del espacio de búsqueda aumenta, el costo computacional de dichos algoritmos puede aumentar de forma exponencial, convirtiéndolas en técnicas ineficaces.

Otra forma de afrontar este tipo de problemas consiste en buscar una solución sub-óptima, pero en un tiempo razonable. En algunos casos es posible encontrar incluso la solución óptima al problema. Sin embargo, cuando se trata de escenarios reales, aumenta la complejidad de las soluciones.

La teoría de grafos siempre se utiliza como la base matemática para la solución de este tipo de problemas [Ma, 2010]. Desde un punto de vista general, un grafo G consiste en un par (V, E) donde V es un conjunto finito no vacío cuyos elementos se llaman vértices y E es un conjunto de pares desordenados de elementos distintos de V [Thulasiraman, 2016]. Dada la definición anterior, cada vértice corresponde a un punto dentro de un espacio conocido, y está unido a otros puntos por medio de aristas, que en este caso representarían los caminos.

Un ejemplo clásico de optimización combinatoria es el problema de los siete puentes [Thulasiraman, 2016]. Se dice que hay un río llamado Pregel que divide la ciudad de Königsberg en cuatro masas de tierra A, B, C , y D , en la Figura 3 se muestra esta distribución. Siete puentes conectan las diversas partes de la ciudad y algunos ciudadanos curiosos que viven en esta ciudad usaron como una forma de recreación en la que intentaron tomar un viaje a través de los siete puentes y volver a la masa de tierra en que comenzaban sin tener que cruzar cualquier puente más de una vez. Todos los que intentaron terminaron con el fracaso, incluyendo al matemático suizo Leonhard Euler (1707-1783), el cual fue un genio famoso del siglo XVIII. Euler consideró las diferentes partes de la ciudad como un punto, y los puentes que conectaban dos partes como una línea y dedujo el grafo. El método utilizado por Euler cimentó la base para la topología y la optimización combinatoria. En la Figura 20 se muestra el grafo baso en el problema de los 7 puentes.

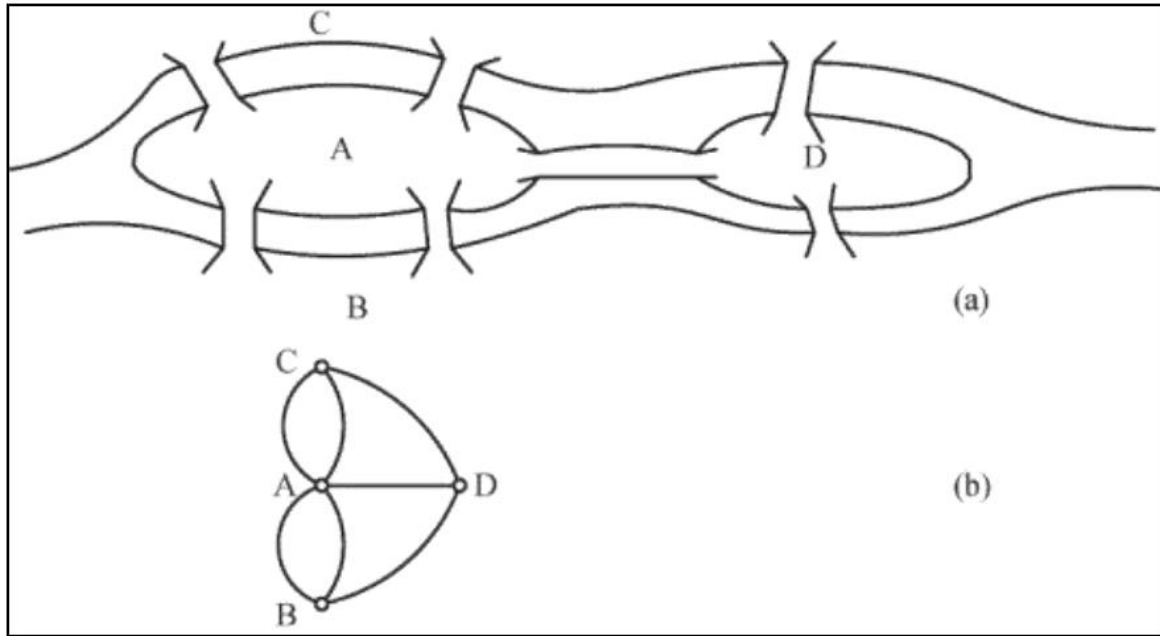


Figura 20. (A) problema siete puentes, (b) grafo basado en el problema de los 7 puentes [Ma, 2011].

Según Euler se puede definir un grafo como G para describir el problema de siete puentes en la teoría de grafos.

$$G = (V(G), E(G)) \quad (1)$$

$$V(G) = \{A, B, C, D\} \quad (2)$$

$$E(G) = \{AC, CA, AB, BA, AD, BD, CD\} \quad (3)$$

Los elementos de E se llaman bordes del gráfico G . Si $e = \{u, v\} \in E$, el borde e se dice que une u y v . Los vértices u y v se llaman los vértices finales del borde uv . Escribimos $e = uv$ y decimos que los vértices u y v son adyacentes. También decimos que el vértice u y el borde e son incidentes entre sí. Si dos bordes distintos e_1 y e_2 son incidentes con un vértice común, entonces se llaman bordes adyacentes. Un gráfico con n vértices y m bordes se denomina gráfico (n, m) . El número de vértices en G se llama el orden de G . El número de bordes de G se llama el tamaño de G [Thulasiraman, 2016].

La definición de un gráfico no permite más de un borde que une dos vértices. Tampoco permite que ningún borde se una a un vértice consigo mismo. Tal borde que une un vértice a sí mismo se denomina bucle automático o simplemente un bucle. Si se permiten más de un borde que une dos vértices, el objeto resultante se llama multigrafía. Los bordes que unen el mismo par de vértices se llaman aristas múltiples. Si también se permiten bucles, el objeto resultante se llama pseudo gráfico [Thulasiraman, 2016].

Existen otros ejemplos de problemas de optimización combinatoria relacionados con los grafos. Sin embargo, el objetivo de este trabajo es resolver el problema de la trayectoria más corta *shortest path (SP)*, el cual consiste en encontrar el camino más corto entre la posición inicial y el destino en una Red dada.

En la teoría de grafos un Sub-grafo de G es cualquier grafo $G' = (V, A')$, donde A' sea un subconjunto de A .

Dependiendo de si el orden de los vértices en las aristas importa o no, tenemos:

- Grafo dirigido. - el orden importa, el que el vértice i esté conectado con el vértice j no implica que el vértice j esté conectado con el vértice i .
- Grafo No dirigido. - el orden no importa.

En la Figura 21 se muestra el grafo dirigido del lado derecho y el grafo no dirigido del lado izquierdo.

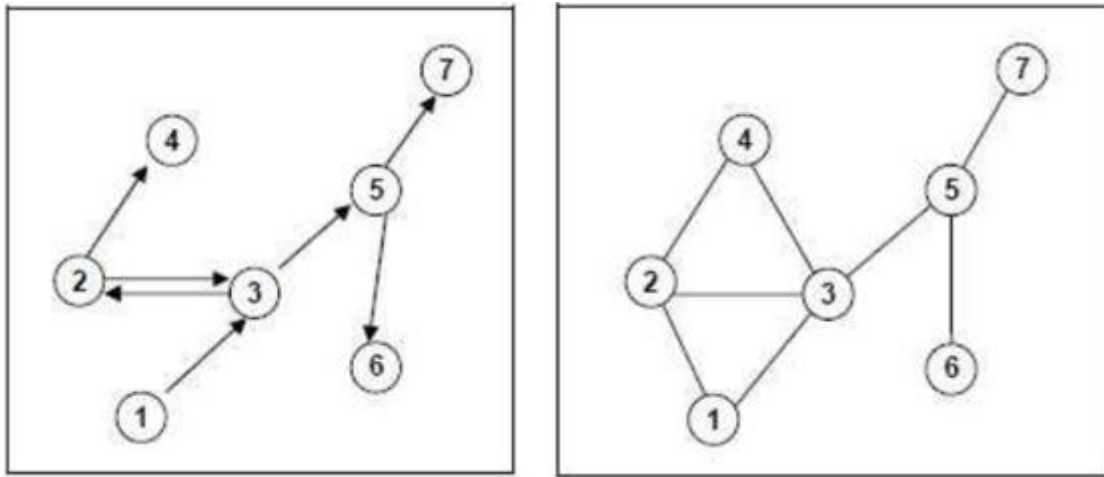


Figura 21. De derecha a izquierda se muestra un grafo dirigido y un grafo no dirigido [Thulasiraman, 2016].

En muchas aplicaciones de los grafos las aristas llevan asociada información adicional. En este caso hablaremos de grafos etiquetados. Si esa información es numérica y tiene significado del costo necesario para recorrer esa arista, entonces usaremos el nombre de grafo ponderado o Red.

Un grafo no dirigido se distingue entre grado interior de un vértice (número de aristas que llegan a él, es decir aristas donde el vértice aparece como segundo componente) y grado exterior (número de aristas que salen de él, aristas donde el vértice aparece como primer componente).

Un camino entre dos vértices i y j es cualquier secuencia de vértices que cumpla que $v_1 = i$, $v_p = j$ y exista una arista entre cada par de vértices contiguos. Un camino simple es aquel donde no hay vértices repetidos en la secuencia, salvo el primero y el último (que puedan ser iguales o distintos). Un ciclo es un camino simple donde el vértice inicial y el final son el mismo. Se dice que un grafo es acíclico cuando todos sus posibles caminos son simples (no existen ciclos). Un árbol es un grafo no dirigido y acíclico [Thulasiraman, 2016].

Se dice que un grafo está conectado si existe como mínimo un camino entre cualquier par de vértices distintos. Un grafo está completamente conectado si para cada vértice existen aristas que lo conecten con los $n - 1$ vértices restantes.

Se define longitud/costo de un camino como la suma de las longitudes de las aristas que recorre la trayectoria.

Para un grafo conectado, se define ruta óptima entre los vértices i y j como la trayectoria de la longitud mínima entre los vértices i y j . Las dos representaciones principales de grafos son las siguientes:

Matriz de adyacencia (MA): se utiliza una matriz de tamaño $n \times n$ donde las filas y columnas hacen referencia a los vértices para almacenar en cada casilla la longitud entre cada par de vértices.

Lista de adyacencia (LA): se utiliza un vector de tamaño n (un elemento por cada vértice) donde la lista almacena la referencia a una lista de los vértices adyacentes.

Tomando a G como un grafo ponderado, donde cada arista está asociada con un número que representa la longitud de dicha arista. La longitud de cada arista es vista como la distancia del vértice i al vértice j y se denota como $w(i, j)$.

Existen numerosos algoritmos para solucionar el problema del camino más corto. Desde las soluciones exhaustivas como la Búsqueda en anchura o la Búsqueda en Profundidad, hasta las soluciones con mayor eficiencia como el Branch & Bound [Morrison, 2016], el algoritmo A* [Thulasiraman, 2016], [Dijkstra, 1959] y el algoritmo Bellman-Ford [Cherkassky, 1996]. El problema del camino más corto tiene diversas aplicaciones en la robótica, las telecomunicaciones, el transporte, la teoría de juegos, las redes de computadoras, búsquedas web, entre otros. Si no hay una arista que vaya del vértice i al vértice j , entonces $w(i, j) = \infty$. La longitud de una trayectoria de G es la suma de las longitudes de las aristas de la trayectoria. La longitud mínima de el vértice s al vértice t es llamada la trayectoria más corta de s a t . La longitud de SP de s a t , si es que existe, es conocida como la distancia de s a t , y se denota como $d(s, t)$.

En la Figura 22, se muestra un grafo ponderado, donde cada número representa la distancia o el costo que existe al recorrer cada arista. En este caso se busca la trayectoria más corta del nodo uno al nodo diez. La línea roja representa la trayectoria más corta entre el nodo inicial uno y el nodo meta diez. La suma de los costos de las aristas en rojo es de 15, por lo tanto, la ruta más corta tiene una distancia de 15.

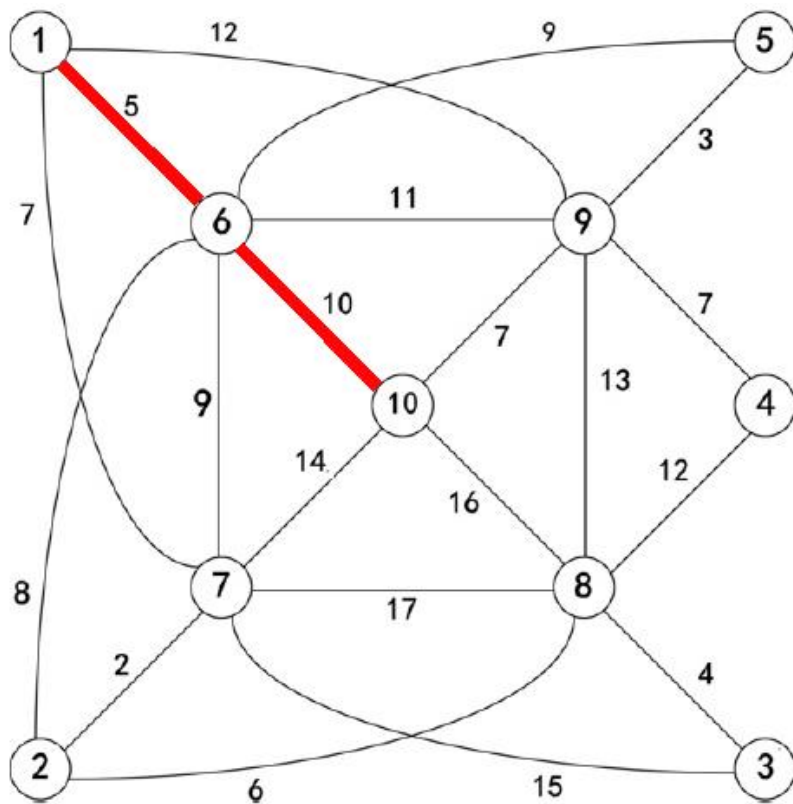


Figura 22. Grafo ponderado que muestra una trayectoria [Li, 2013].

Existen diversos tipos de problemas de optimización asociados con el cálculo de la trayectoria más corta. Por ejemplo: encontrar el flujo mínimo dentro de una Red de transporte, o calcular la mejor trayectoria para el envío de paquetes a través de una Red de telecomunicaciones o la planeación de trayectorias de robots móviles.

3.4 Redes Neuronales Pulsantes

La inteligencia Artificial ha tenido como objetivo, durante mucho tiempo, simular la inteligencia humana y obtener un sistema artificial con capacidad de reflexionar, de tomar decisiones y de aprendizaje. Los investigadores se han interesado rápidamente en el funcionamiento del cerebro para reproducirlo. De este modo es como aparecen las primeras neuronas artificiales por McCulloch y Pitts en 1943 [Mathivet, 2015].

Las Redes Neuronales Artificiales se definen como una estructura compuesta por estas unidades básicas las cuales están interconectadas mediante distintos pesos que la dan la capacidad de almacenar conocimiento experimental y hacerlo utilizable [Arista, 2012].

La Red Neuronal artificial se usa comúnmente en el área de diferentes clases de problemas de optimización. Las redes neuronales análogas tienen la facultad de procesar un gran número de variables simultáneamente, lo que hace posible encontrar soluciones para problemas complejos de optimización multi-objetivo en tiempo real [Khouil, 2016].

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 [Lindblad, 2013].

La corteza visual del mamífero formó la base de algunos modelos de Red. La corteza visual de gato y conejillo de indias ayudó a desarrollar algunos modelos digitales. La información de entrada es recibida por el ojo. Los receptores dentro del ojo (retina) no son sensibles a toda la información. La sensibilidad se basa en el color, el movimiento y la intensidad. El receptor después de recibir la información altera el comportamiento de los receptores circundantes con respecto a los contenidos y luego avanza hacia la corteza visual, donde la información recibida es analizada por el cerebro. El funcionamiento de la corteza visual debe estudiarse para desarrollar algoritmos. Ésto es más complicado que la programación de computadoras. Los investigadores comenzaron a trabajar a principios de la década de 1950 [Hodgkin, 1952]. Habían descrito los potenciales de membrana en términos de tasa de cambio de diferentes elementos químicos [Sabashini, 2014].

A principios de la década de 1960, [FitzHugh, 1961] desarrolló un modelo matemático basado en osciladores acoplados. Se describió a la neurona como dos osciladores acoplados que están conectados a las neuronas vecinas. Más tarde, en la década de 1990, en [Arndt, 1990] introdujo un modelo en la corteza visual de los gatos. En su modelo, cada neurona recibía información de su propio estímulo y también de las neuronas vecinas. Las salidas de otras neuronas también eran una entrada para la neurona madre. En 1992, Rybak, en [Rybak, 1992] se ideó un modelo basado en la corteza visual de los conejillos de indias. El modelo era similar a Eckhorn excepto en ecuaciones. Luego, en [Johnson, 1993] se implementó la Red Neuronal acoplada por pulsos y sugirieron un nuevo mecanismo con conectividad limitada para la transmisión de información [Sabashini, 2014].

Las Redes Neuronales Pulso Acopladas o *Pulse Coupled Neural Networks* (PCNN por sus siglas en inglés) son redes no supervisadas, forman una Red auto organizada que no requiere entrenamiento. La Red Neuronal se construyó simulando las actividades de las neuronas de la corteza visual del mamífero. La estructura básica del modelo PCNN se muestra en la Figura 23 [Sabashini, 2014].

En un principio las PCNN fueron empleadas principalmente en problemas que tienen que ver con el procesamiento de imágenes. En esos modelos la cantidad de neuronas en la Red es igual al número de

pixeles de entrada. Cada píxel de la imagen está conectado a una neurona única y cada neurona está conectada con las neuronas circundantes a través de un radio de campo de enlace.

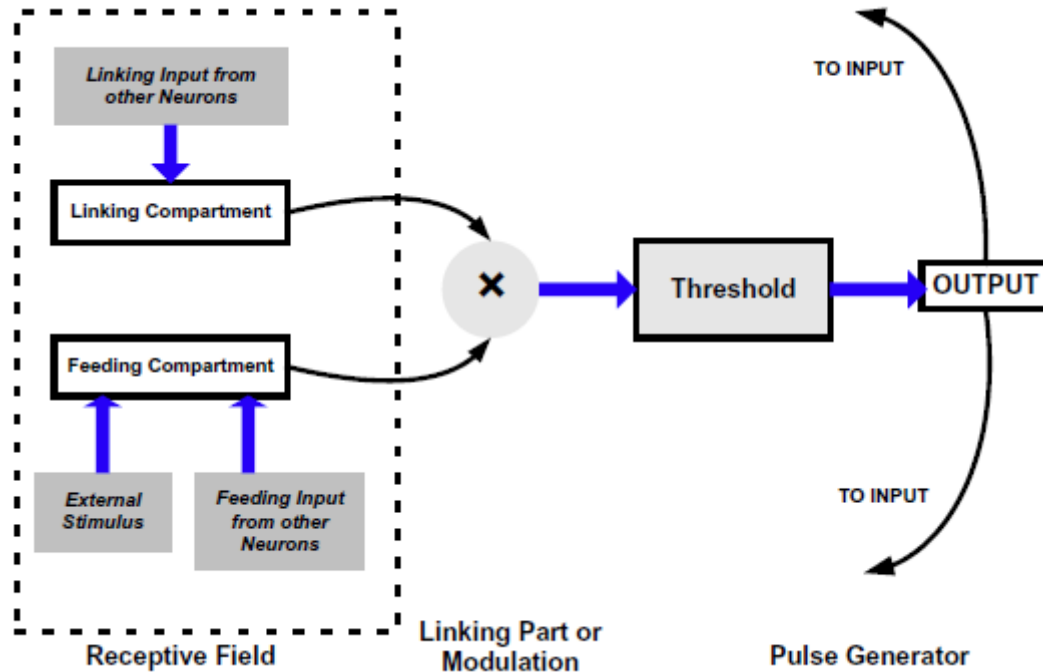


Figura 23, Estructura básica de la neurona PCNN [Sabashini, 2014].

La Red Neuronal acoplada por pulsos tiene tres compartimentos:

1. *Receptive field*, (campo receptivo) a la izquierda.
2. *Linking part or modulation*, (modulación) en el centro.
3. *Pulse generator*. (generador de pulsos) a la derecha.

El campo receptivo es la parte principal para recibir señales de entrada de las neuronas vecinas y de fuentes externas, y el campo tiene dos canales internos conocidos como Compartimiento de alimentación F y Compartimiento de enlace L . Las entradas de enlace tienen una constante de tiempo de respuesta característica más rápida en comparación con las conexiones de alimentación.

Las entradas de enlace sesgadas y multiplicadas se multiplican con la entrada de alimentación para producir la actividad interna total U que constituye la parte de enlace o modulación.

Finalmente, el generador de impulsos de la neurona consiste en un generador de función de paso y un generador de señal de umbral. Las neuronas en la Red tienen la capacidad de responder a los estímulos. Esta respuesta se conoce como disparar. Esta activación se habilita cuando la actividad interna de la neurona supera un cierto umbral. La salida de neuronas Y es 1. Ahora el umbral comienza a decaer hasta la siguiente actividad interna de la neurona. La salida de la neurona se retroalimenta iterativamente con un retraso de iteración única. La salida de la neurona se restablece a cero cuando el umbral es mayor que la actividad interna U . Las neuronas PCNN producen series temporales de salidas de pulso después de n número de iteraciones. La salida de impulsos contiene información sobre la imagen de entrada. La decisión sobre el contenido de la imagen se logra al examinar la salida de pulso de la Red (PCNN) [Sabashini, 2014].

La entrada de alimentación y la entrada de enlace se comunican con las neuronas vecinas a través de los pesos sinápticos M y W . El estímulo de entrada sólo se aplica al compartimento de alimentación. La neurona recibe el estímulo de entrada S , cuando se utiliza este modelo para manipulación de imágenes, cada entrada es la información de color de un píxel junto con el estímulo de los vecinos en ambos compartimentos. En la Figura 8 se presenta el modelo matemático de la PCNN y la salida de los compartimentos está determinada por las siguientes ecuaciones:

$$F_{ij}[n] = e^{\alpha_F \delta_n} F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \quad (4)$$

$$L_{ij}[n] = e^{\alpha_L \delta_n} L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \quad (5)$$

$F_{ij}[n]$: compartimento de alimentación primaria de la neurona (i, j) ;

$L_{ij}[n]$: salida del compartimento de enlace;

S_{ij} : estímulo de entrada de los píxeles de la imagen en la posición (i, j) ;

Y_{kl} : Salida de neuronas de una iteración previa $[n-1]$;

β : Coeficiente de enlace;

M_{ijkl} y W_{ijkl} : funciones constantes de peso gaussianas con distancia;

V_F y V_L : potenciales de tensión inherentes.

Los estados de éstos dos compartimentos se combinan para crear el estado interno de la neurona, y su actividad $U_{ij}[n]$ se calcula usando la siguiente ecuación:

$$U_{ij}[n] = F_{ij}[n][1 + \beta L_{ij}[n]] \quad (6)$$

El estado interno de la neurona se compara con un umbral, θ_{ij} a producir la salida pulsante, Y ;

$$Y = \begin{cases} 1, & \text{if } U_{ij}[n] > \theta_{ij}[n] \\ 0, & \text{de otra manera} \end{cases} \quad (7)$$

Cuando la neurona dispara ($Y > h$), el umbral aumenta significativamente su valor. Este valor del umbral luego decae hasta que la neurona dispara nuevamente. Por lo tanto, el umbral es dinámico. El proceso es descrito por,

$$\theta_{ij}[n] = e^{\alpha_\theta \delta_n} \theta_{ij}[n-1] + V_\theta Y_{ij}[n] \quad (8)$$

donde V_h es una gran constante mayor que el valor promedio de U_{ij} [Sabashini, 2014].

Para reducir la complejidad de la Red (PCNN) y aumentar la velocidad de cálculo, se han realizado ciertas modificaciones en la Red. Estas modificaciones sin duda han mejorado el rendimiento general de PCNN. Una versión simplificada del modelo PCNN es el modelo de corte transversal. Se desarrolló un sistema mínimo utilizando dos osciladores acoplados, función no lineal y un pequeño número de conexiones.

Las ecuaciones correspondientes de la Red del modelo de corte transversal se ilustran a partir de las siguientes ecuaciones:

$$F_{ij}[n-1] = f F_{ij}[n] + S_{ij} + W\{Y\}_{ij} \quad (9)$$

$$Y_{ij}[n+1] = \begin{cases} 1, & \text{if } F_{ij}[n+1] > \theta_{ij} \\ 0, & \text{de otra manera} \end{cases} \quad (10)$$

$$\theta_{ij}[n + 1] = g\theta_{ij} + hY_{ij}[n + 1] \quad (11)$$

S_{ij} es el estímulo, θ_{ij} es el umbral de la neurona, Y_{ij} es la salida y f , g , y h son escalares. Los parámetros se redujeron y las entradas de enlace se hicieron uniformes, lo que dio como resultado otro modelo conocido como modelo de vinculación de unidades. [Sabashini, 2014].

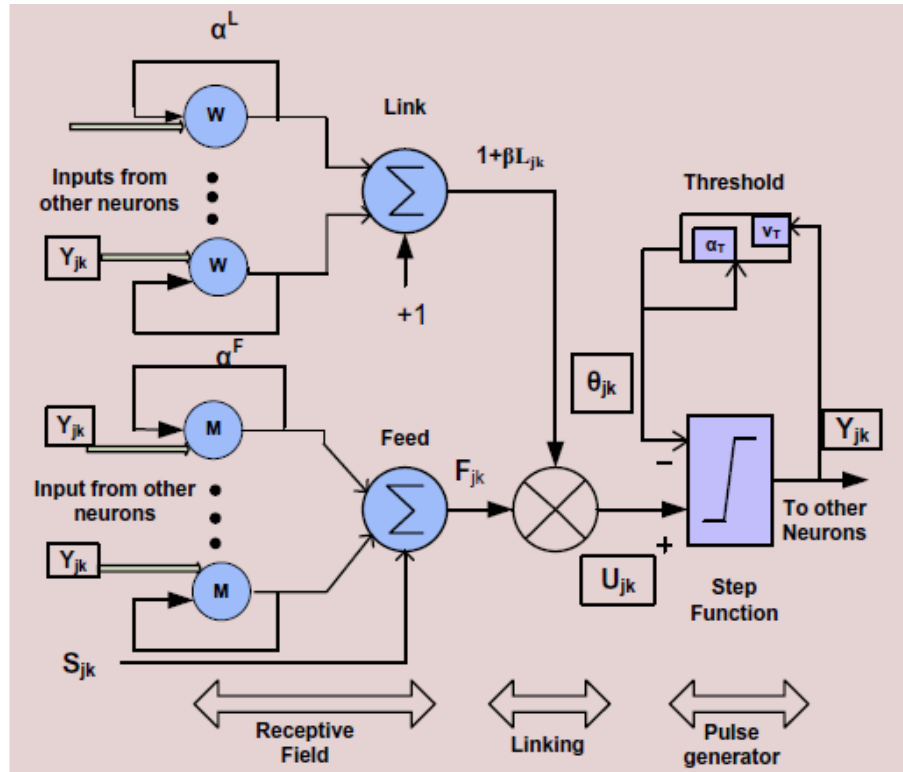


Figura 24. Modelo matemático de la PCNN [Sabashini, 2014].

Otros modelos modificados de PCNN que se pueden encontrar en la literatura son el *Intersection Cortical Model* (ICM), que es un caso especial donde no hay *linking* por lo que β es igual a cero, sólo existe el *feeding*. En el *Spiking Cortical Model* (SCM) las neuronas reciben la señal de entrada por medio de las sinapsis del *feeding*, y cada neurona está conectada con sus vecinas de modo tal que la señal de salida de cada neurona modula la actividad de sus vecinas por medio de las sinapsis del *linking*.

Ha aumentado el interés en el uso de PCNN para diversas aplicaciones, como segmentación [Jio, 2019], reconocimiento de objetos [Harris, 2015], el procesamiento de imágenes [Lei, 2019] [Yang, 2019] y reducir el efecto de ruido gaussiano [Ortiz, 2016]. Algunas investigaciones recientes muestran que la dinámica espacio temporal de las PCNN proporciona una buena capacidad computacional para resolver algunos problemas de optimización.

En 1999, Caulfield y Kinser [Caulfield, 1999] presentaron la idea de utilizar la onda automática en PCNNs para encontrar la solución al problema del laberinto. Su modelo puede encontrar la ruta más corta rápidamente y con el mínimo esfuerzo, donde la solución está relacionada con la longitud de la ruta más corta e independiente de la complejidad del gráfico de la ruta. Sin embargo, se necesitan muchas neuronas para encontrar la ruta más corta en grandes laberintos o gráficos, ya que una neurona corresponde a una unidad de longitud de ruta.

Posteriormente [Qu, 2007] presenta el modelo *Multi-Output Pulse-Coupled Neural Network* (MPCCN), el cual tiene un costo computacional menor. El número de iteraciones de la MPCNN es proporcional a la longitud del camino más corto. [Ma, 2011] publicaron el modelo *Auto-Wave Neural Network* (AWNN), el cual aprovecha la característica de auto-onda de las PCNN para resolver el problema de SP. [Paredes-Cano, 2018] presentaron una mejora del modelo AWNN además de una técnica para explicitar el conocimiento generado por la Red Neuronal. Los modelos mencionados anteriormente sugieren que la velocidad de la onda debe mantener una velocidad constante. Posteriormente se presentaría un modelo llamado *Self-adaptive Autowave Pulse-Coupled Neural Network* (SAPCNN) [Li, 2013]. Este modelo puede ajustar la velocidad de propagación de la onda automáticamente de acuerdo con el estado actual de la Red. Adicionalmente los resultados indican que el modelo logra calcular el camino más corto con menor cantidad de iteraciones.

3.5 Discusión

La Robótica Móvil es un campo con diversas aplicaciones como: la exploración planetaria, minera marítima, reconocimiento de terreno, vigilancia, misiones de búsqueda y rescate, neutralización de dispositivos explosivos, limpieza de desechos peligrosos, asistencia médica, entretenimiento, entre otros. [Ortigoza, 2012]. Uno de los problemas más importantes para los robots móviles y los vehículos no tripulados es la Planeación de Trayectorias. El propósito del planificador de trayectorias es calcular una ruta desde la posición de inicio del vehículo hasta el objetivo que se debe alcanzar. La principal preocupación de la planificación de rutas es calcular rutas sin colisiones. Otro tema de igual importancia es calcular un camino realizable y, si es posible, que sea el Camino más Corto u óptimo, llevando el vehículo a la posición final [Kroumov, 2011]. Para encontrar el camino más corto entre la posición del robot y un punto objetivo, es necesario introducir algoritmos de optimización combinatoria que puedan obtener la solución en tiempo real.

Dentro de la optimización combinatoria se encuentran los problemas de optimización de trayectorias cuyo propósito es minimizar el costo generado por ir de un punto a otro. Existen técnicas exhaustivas para resolver dichos problemas, pero a medida que la complejidad del espacio de búsqueda aumenta, el costo computacional de dichos algoritmos puede aumentar de forma exponencial, convirtiéndolas en técnicas ineficaces [Mathivet, 2015]. Adicionalmente, se han implementado algoritmos inteligentes para resolver el Problema del Camino más Corto, sin embargo, algunas alternativas tienen problemas de mínimos locales o no son capaces de encontrar la trayectoria óptima. Por otro lado, Las Redes Neuronales Pulsantes se presentan como una alternativa efectiva para resolver problemas de Optimización de Trayectorias y Planeación de trayectorias según se observa en [Caulfield, 1999], [Qu, 2007], [Ma, 2011], [Li, 2013] y [Paredes-Cano, 2018].

Capítulo 4 Metodología de Solución

Se realizó la implementación de los modelos seleccionados en dos tipos de problemas diferentes. Los primeros en el campo de Optimización de Trayectorias, resolviendo el problema del camino más corto (grafos). El segundo en el campo de la Planeación de Trayectorias para Robots Móviles (mapas).

4.1 Optimización de Trayectorias

Se realizó una búsqueda en el estado del arte con la finalidad de seleccionar y analizar diferentes modelos de PCNN, los cuales, sean útiles para el desarrollo de un estudio comparativo en el campo de la Optimización de Trayectorias

Se presentan los siguientes cuatro modelos de Redes Neuronales Pulso Acopladas orientados a la optimización de rutas.

- AWNN (*Auto-Wave Neural Network*) [Ma, 2011]
- AWNN Explicitada (*Auto-Wave Neural Network E*) [Paredes-Cano, 2018]
- SAPCNN (*Self-adaptive Autowave Pulse-Coupled Neural Network*) [Li, 2013]
- SAM-PCNN (*Self-adaptive Autowave Modified Pulse-Coupled Neural Network*) [Espinosa, 2019]

A continuación, se dará una breve descripción de los modelos que fueron implementados.

4.1.1 *Auto-Wave Neural Network*

En la construcción del estado del arte, se encontraron diversas variantes de PCNN diseñadas específicamente para resolver problemas de optimización. La primera variante que se seleccionó para esta investigación es el denominado por sus autores como AWNN (*Auto-Wave Neural Network*, por sus siglas en inglés) presentado en [Ma, 2011]. Dicho modelo está diseñado para resolver el problema de la ruta más corta. Utiliza como entrada la matriz de adyacencias de un grafo ponderado. En la Figura 25 se muestra una neurona de este modelo, el cual está compuesto por la capa de recepción, de enlace y de pulsos. La capa de enlace está formada por la combinación de W y F . La matriz U representa la capa de enlace mientras que la capa de pulsos está compuesta por el umbral E y la salida Y . En la Figura 25 se muestra el diagrama del modelo AWNN.

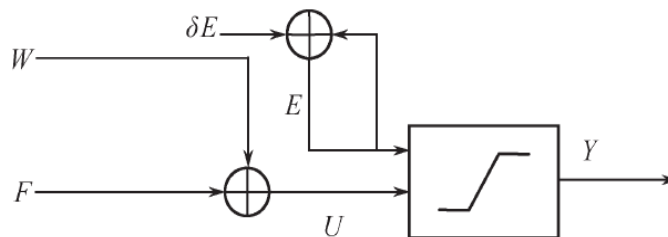


Figura 25. Modelo de la neurona *Auto-Wave Neural Network* [Ma, 2011].

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 δE : 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 AWNN.

$$F_i[n] = F_h[n] + W_{hi} \tag{12}$$

$$U_{ij}[n] = F_i[n] + W_{ij} \tag{13}$$

$$Y_{ij} = \begin{cases} 1 & E[n] \geq U_{ij}[n] \\ 0 & \text{de otra manera} \end{cases} \tag{14}$$

$$E[n] = E[n - 1] + \delta E \tag{15}$$

Se seleccionó el modelo AWNN por que cuenta con un mecanismo sencillo y fácil de comprender. Dicho modelo es un buen candidato para comenzar a entender la manera en que las Redes Neuronales Pulso Acopladas son empleadas en problemas de optimización de trayectorias.

Partiendo del grafo de tres nodos presentado en la Figura 26, se busca calcular la trayectoria entre el nodo A y el nodo C . En este ejemplo, el incremento de la actividad interna δE tiene un valor constante de 1, la actividad interna E incrementa en cada interacción como lo indica la Ecuación 15, posteriormente, se compara contra el umbral de U_{ij} de las neuronas de la Red. Cuando la actividad interna alcanza el umbral de alguna de las neuronas se produce un pulso como se indica en la Ecuación 14. En la Figura 26 se muestra cómo el modelo AWNN necesita de 16 iteraciones para alcanzar el nodo objetivo.

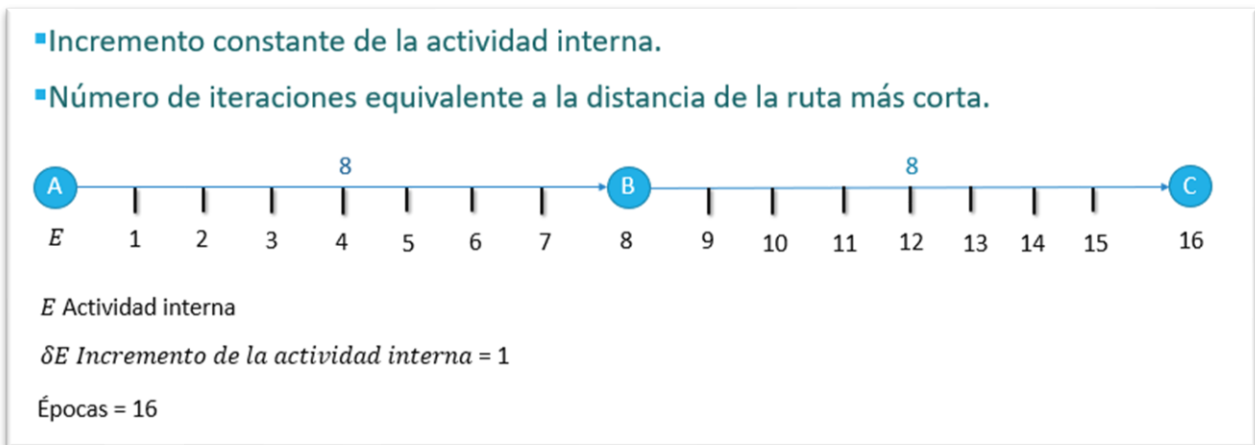


Figura 26. Grafo de tres nodos resuelto por el AWNN [Ma, 2011].

4.1.2 Auto-Wave Neural Network Explicitada

Este modelo de RNAP surge a partir de un artículo realizado en CENIDET [Paredes-Cano, 2018]. Es una extensión del modelo AWNN presentado en [Ma, 2011]. Para hacer más eficiente el proceso los autores proponen un valor inicial de E como el costo menor del nodo de inicio a sus siguientes nodos. Esta modificación permite que desde la primera interacción pulse por lo menos una neurona.

Adicionalmente se propone un método para recuperar la ruta calculada por la Red Neuronal. La Explicitación del Conocimiento es un sub-tema dentro de las RNA y 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, en [Paredes-Cano, 2018] se logra 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.

Para ilustrar este proceso de explicitación de conocimiento se presenta un ejemplo. En un grafo de cinco nodos se busca la trayectoria más corta del nodo A al nodo D . Una vez que la Red Neuronal ha calculado la trayectoria que se muestra la Figura 27, la matriz de salida Y es tal cómo se muestra en la Tabla 3.

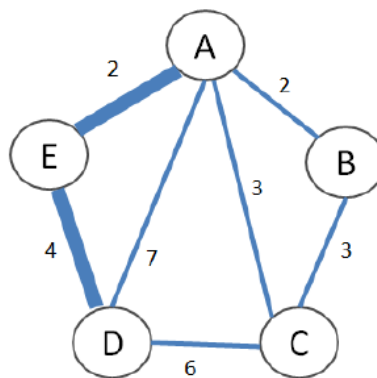


Figura 27. Grafo del ejemplo que muestra la solución de la RNP [Paredes-Cano, 2018].

Tabla 3. Matriz de salida del ejemplo [Paredes-Cano, 2018].

Y	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 $A - E - D$.

En el siguiente ejemplo se muestra un grafo de 3 nodos, donde se realiza la búsqueda del camino más corto entre el nodo A y el nodo C. En la Figura 28 se puede apreciar como δE tiene un comportamiento dinámico en la primera iteración, tomando el valor del menor de los umbrales de la Red, lo que le permite a E alcanzar al siguiente nodo. Posteriormente δE toma un valor constante de 1. En el ejemplo se puede ver que a la variante AWNN le toma 9 iteraciones para alcanzar al nodo objetivo.

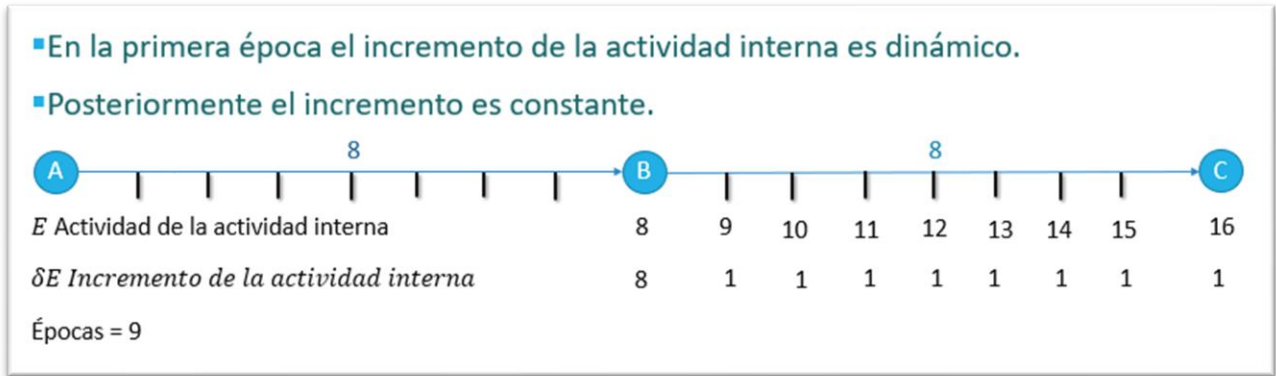


Figura 28. Grafo de tres nodos resuelto por el AWNN Explicitada [Paredes-Cano, 2018].

4.1.3 Self-adaptive Autowave Pulse-Coupled Neural Network

El modelo SAPCNN [Li, 2013] propaga la onda de forma dinámica según el estado de la Red. Esto representa una mejora, pues la mayoría de los métodos sugieren que la onda automática de los modelos PCNN debe mantener una velocidad constante para encontrar los caminos más cortos. En la Figura 29 se muestra el diagrama del modelo.

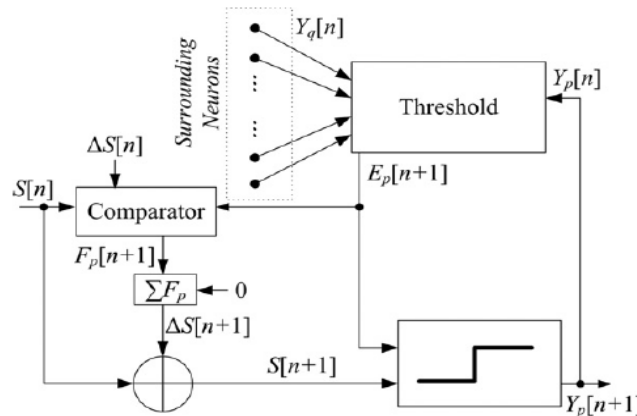


Figura 29. Diagrama del modelo SAPCNN [Li, 2013].

Las ecuaciones que emplea el modelo SAPCNN son:

$$E_p[n + 1] = \begin{cases} V_E & \text{if } Y_p[n] = 1 \\ \min(S[n] + W_{qp}[n], E_p[n]) & \text{if } Y_p[n] = 0, Y_q[n] = 1 \text{ y } p \in R_q \\ E_p & \text{de otra manera} \end{cases} \quad (16)$$

$$F_p[n + 1] = \begin{cases} N & \text{if } E_p[n + 1] < S[n] + \Delta S[n] \\ -1 & \text{if } E_p[n + 1] = S[n] + \Delta S[n] \\ 0 & \text{de otra manera} \end{cases} \quad (17)$$

$$\Delta S[n] \begin{cases} \gamma & \text{if } F_p > 0 \\ \Delta S[n] + \gamma & \text{if } F_p = 0 \\ \Delta S[n] & \text{if de otra manera} \end{cases} \quad (18)$$

$$S[n + 1] < S[n] + \Delta S[n] \quad (19)$$

$$Y_p[n + 1] = \begin{cases} 1 & \text{if } S[n + 1] \geq E_p[n + 1] \\ 0 & \text{de otra manera} \end{cases} \quad (20)$$

Este modelo está diseñado para resolver el problema del camino más corto, sin embargo, también puede ser usado para la planeación de trayectorias para robots móviles.

Una manera de ilustrar el funcionamiento del modelo SAPCNN es resolviendo el problema del camino más corto entre el nodo *A* y el nodo *C* del grafo mostrado en la Figura 30. La primera iteración $\Delta S[n]$ toma el valor de γ el cual se define con el valor constante de 1. Posteriormente, se compara la actividad interna más su incremento $S[n] + \Delta S[n]$ con respecto del umbral $E_p[n + 1]$ cómo se indica en la Ecuación 17. Si $S[n] + \Delta S[n]$ no es suficiente para superar al umbral $E_p[n + 1]$, entonces, se suma el valor de γ al incremento de la actividad interna $\Delta S[n] + \gamma$ como se muestra en la Ecuación 18. Cuando la actividad interna $S[n + 1]$ logra superar al umbral $E_p[n + 1]$ se genera un pulso en la neurona alcanzada como se indica en la Ecuación 20. Como se puede apreciar en la Figura 30, $\Delta S[n]$ incrementa su valor en cada iteración mientras $E_p[n + 1]$ no se vea superado por $S[n] + \Delta S[n]$. Al modelo SAPCNN le tomaría solamente 6 iteraciones para resolver la trayectoria. En la primera iteración $S[n] + \Delta S[n]$ es igual a 1 y se compara con respecto de $E_p[n + 1]$ que es igual a 8 en la neurona *B* y 16 en la neurona *C*. Al no superar a ningún umbral $\Delta S[n]$ incrementa en γ , lo que permite que $E_p[n + 1]$ termine con un valor de 2 en la primera iteración.

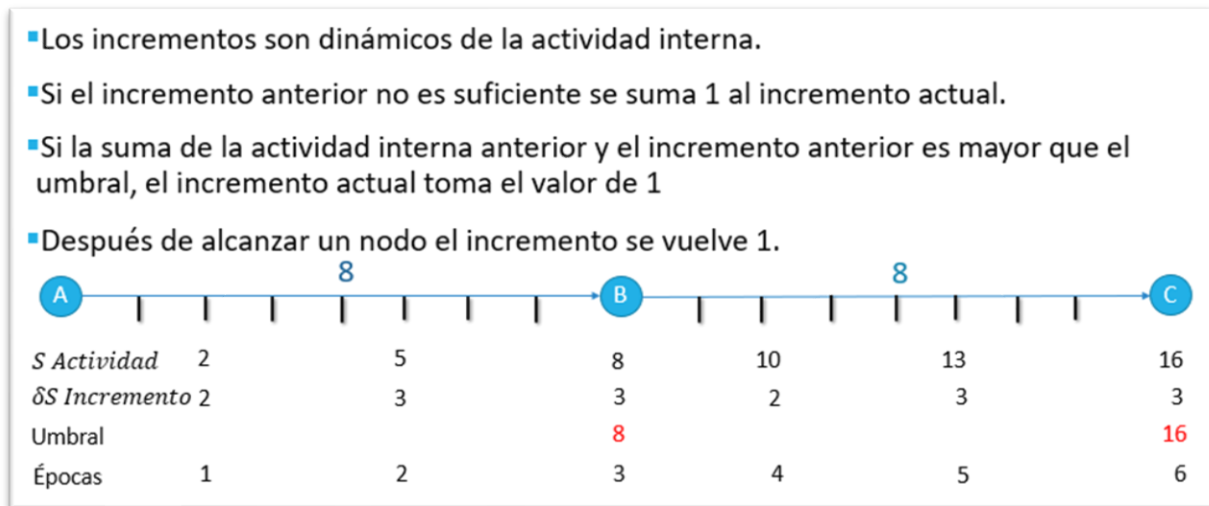


Figura 30. Grafo de tres nodos resuelto por el SAPCNN [Li, 2013].

Tomando en cuenta el grafo que se presenta en la Figura 31, se busca el camino más corto del nodo de inicio *A* al nodo meta *C*. El algoritmo SAPCNN [Li, 2013] tiene el siguiente comportamiento:

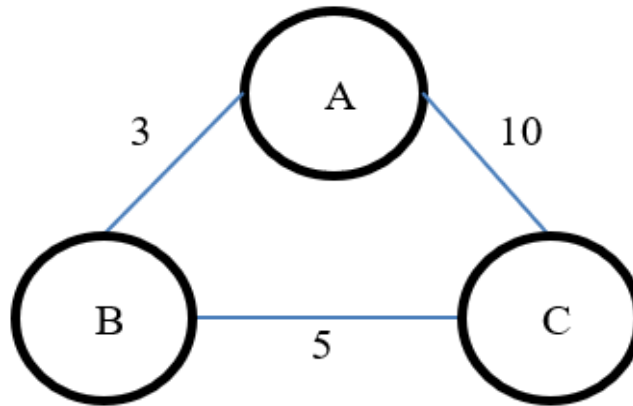


Figura 31. Grafo del ejemplo.

Al inicializar la Red en la iteración $n = 0$, se hace pulsar el nodo A

1. Cuando $n = 1$: El valor de la actividad interna calculado por las Ecuaciones 16,17 y 18 es $S[1] = 2$, y no le permite alcanzar el umbral de ninguna de las neuronas vecinas.
2. Cuando $n = 2$: La actividad interna toma el valor $S[2] = 3$, permitiéndole a la neurona B generar un pulso en su salida $Y_B[2] = 1$.
3. Cuando $n = 3$: $S[3] = 5$ y no alcanza al umbral de la neurona C.
4. Cuando $n = 4$: La actividad interna es $S[4] = 8$ y $Y_C[4] = 1$, dando por finalizada la búsqueda.

4.1.4 Self-Adaptive Autowave Modified Pulse-Coupled Neural Network

Además de los modelos de RNAP encontrados en el estado del arte, se presenta una nueva variante de Red Neuronal Pulsante, el cual está inspirado en los objetivos que persigue el SAPCNN [Li, 2013] y en la arquitectura del AWNN [Ma, 2010]. El SAM-PCNN es presentado como un modelo que busca calcular la ruta óptima en el menor número de iteraciones posibles. El modelo cuenta con un mecanismo que le permite a la actividad interna de la neurona incrementar de manera dinámica para alcanzar el umbral en el menor número de iteraciones.

El objetivo del modelo SAM-PCNN que aquí se propone es resolver el problema del camino más corto o *Shortest Path* (SP, por sus siglas en inglés) en un grafo ponderado no dirigido en la menor cantidad de iteraciones. Cuenta con una velocidad de onda dinámica que puede adaptarse dependiendo del estado de la Red. Sin embargo, a diferencia de la SAPCNN [Li, 2013], el modelo SAM-PCNN logra adaptar la velocidad de onda de manera exacta en cada iteración, logrando eliminar las iteraciones donde no hay cambios en la Red.

El diagrama del modelo propuesto se presenta en la Figura 32. Se puede apreciar que la neurona se alimenta de las salidas del resto de las neuronas y se retroalimenta de su propia salida para calcular el umbral. Posteriormente se calcula la velocidad dinámica de la auto-onda, la cual ayuda a la actividad interna de la Red a alcanzar el umbral de por lo menos una neurona en cada iteración.

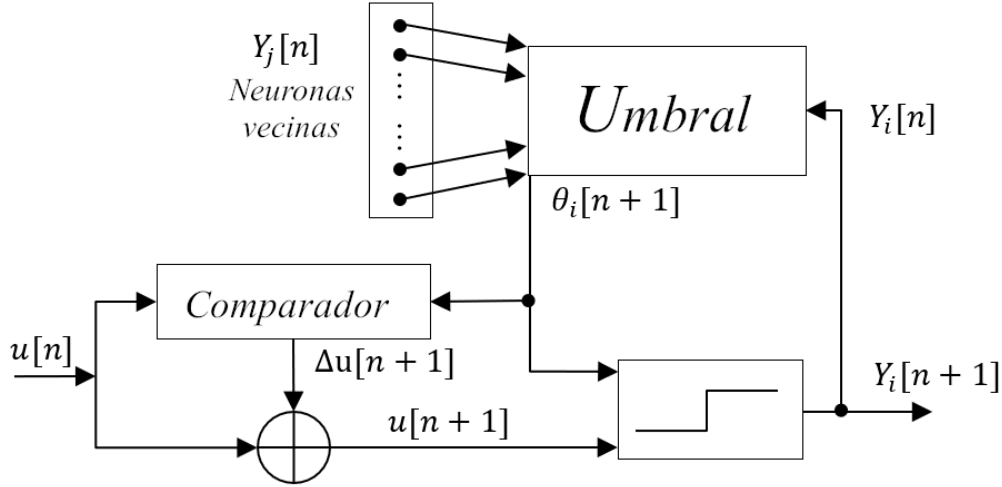


Figura 32. Diagrama de la neurona del modelo propuesto SAM-PCNN.

Para explicar con mayor claridad el modelo propuesto, es necesario definir las notaciones más relevantes del tema. Los índices i y j se refieren a diferentes neuronas. Para hacer referencia a los vecinos de la neurona i se utiliza R_i . W_{ij} es la distancia positiva diferente de cero que existe entre la neurona i y la neurona j sólo si $j \in R_i$, de lo contrario, W_{ij} es cero. N es el número total de neuronas.

La descripción matemática del modelo SAM-PCNN es la siguiente: cada una de las neuronas tiene una salida binaria Y_i la cual se calcula de acuerdo con la Ecuación 21.

$$Y_i[n+1] = \begin{cases} 1 & \text{if } u[n+1] > \theta_i[n+1] \\ 0 & \text{de otra manera} \end{cases} \quad (21)$$

Si $Y_i[n] = 1$, se dice que la neurona i ha pulsado en la iteración n , u es la actividad interna de la Red y θ_i es el umbral de la neurona, el cual se calcula con la Ecuación 22.

$$\theta_i[n+1] = \begin{cases} V_\theta & \text{si } Y_i[n] = 1 \\ \min(u[n] + W_{ji}[n], \theta_i[n]) & \text{si } Y_i[n] = 0, Y_j[n] = i \text{ y } i \in R_j \\ \theta_i[n] & \text{de otra manera} \end{cases} \quad (22)$$

$$V_\theta = NW_{max} \quad (23)$$

El umbral θ_i puede tomar tres valores diferentes dependiendo de las salidas de la Red. V_θ es un valor alto que el umbral toma cuando la neurona tiene una salida positiva. Se calcula con la Ecuación 8 multiplicando el número total de neuronas N por la máxima distancia de W . V_θ evita que la neurona pulse nuevamente. Si la neurona vecina j pulsa, se busca el valor mínimo entre $u[n] + W_{ji}[n]$ y el valor actual del umbral $\theta_i[n]$.

En la ecuación 24 se muestra que el incremento de u es Δu . Se calcula con la diferencia entre el valor mínimo de las x y u . Siendo x un elemento de los umbrales θ mayores que u . El cálculo de $u[n+1]$ se muestra en la Ecuación 25.

$$\Delta u[n+1] = \min(x) - u[n] \quad (\forall x \in \theta[n+1] | x > u[n]) \quad (24)$$

$$u[n+1] = u[n] + \Delta u[n+1] \quad (25)$$

En la Ecuación 26 se muestra el cálculo de W que representa la matriz de distancias. Cuando una neurona ha pulsado se modifican las distancias que llevan a esa neurona por el valor de V_θ .

$$W_{ji}[n + 1] = \begin{cases} V_{\theta} & \text{if } Y_i[n + 1] = 1 \text{ y } i \in R_j \\ W_{ji}[n] & \text{de otra manera} \end{cases} \quad (26)$$

Si una neurona i es alcanzada por la auto-onda proveniente de una neurona j , se guarda la neurona j como el padre de la neurona i utilizando la notación R_i^P . Cuando la onda de disparo alcanza la posición de la neurona meta, la neurona meta se disparará y se determinará una ruta desde la meta a la neurona inicio moviéndose sucesivamente hacia el padre de cada neurona. Éste será el camino más corto desde la neurona de inicio a la neurona meta.

4.1.5 Complejidad Algorítmica

Para un gráfico topológico G con nodos N y aristas E , la complejidad del algoritmo Dijkstra para encontrar el árbol de ruta más corta es $O(N^2)$, donde N es el número de nodos de un grafo G [Li,2013].

Para los modelos de PCNN como el modelo AWNN [Ma, 2011], el cual cuenta con una velocidad constante de propagación, la complejidad algorítmica es (LN) , donde L es equivalente a la distancia entre un nodo de inicio y el nodo más lejano.

En el caso de los modelos SAPCNN [Li, 2013] y SAM-PCNN, los cuales cuentan con una velocidad dinámica de propagación, la complejidad algorítmica es $O(N)$, debido a que para cada iteración es necesario calcular los umbrales, la velocidad de la auto-onda y la actividad interna. Tomando en cuenta que M es el número de iteraciones necesarias para encontrar la solución, la complejidad de ambos modelos es $O(MN)$.

Sin embargo, para el modelo propuesto SAM-PCNN, el número de iteraciones es proporcional al número de neuronas cuya distancia al nodo de inicio sea menor que la distancia del nodo objetivo. Es decir, si se desea calcular el camino más corto del nodo A al nodo C , pero el nodo B es más cercano al nodo A , el número de neuronas más cercanas que C es 1 que se puede representar como P . En este caso M es igual a $P + 1$, pues en la primera iteración pulsará la neurona B y en la siguiente iteración pulsará la neurona C . Por lo tanto, la complejidad algorítmica para el modelo propuesto SAM-PCNN puede reescribirse como $O((P + 1)N)$. En la Tabla 4 se muestra la comparativa de las complejidades algorítmicas.

Tabla 4. Tabla comparativa de complejidades algorítmicas.

Algoritmo	Complejidad
Dijkstra	$O(N^2)$: N es el número de nodos de un grafo
AWNN	$O(LN)$: L es equivalente a la distancia entre el nodo de inicio y el nodo más lejano
SAPCNN	$O(MN)$: M es el número de iteraciones necesarias para encontrar la solución
SAM-PCNN	$O((P + 1)N)$: P es el número de nodos cuya distancia al nodo de inicio sea menor que la distancia al nodo objetivo

En comparación con el modelo SAPCNN, el modelo SAM-PCNN es capaz de encontrar esta trayectoria entre el nodo A y el nodo C de la Figura 31 en menos iteraciones. En la iteración $n = 0$ los valores de la Red son: $\theta_s[0] = V_{\theta}$, $Y_s[1] = 1$, $u[0] = 0$. A partir de ahí, la Red SAM-PCNN se comporta de la siguiente manera:

1. Cuando $n = 1$:

Ecuación 22 cálculo de umbrales:

$$\theta_B[1] = 3, \theta_C[1] = 10$$

Ecuación 24 cálculo de velocidad de auto-onda:	$\Delta u[1] = 3$
Ecuación 25 cálculo de actividad interna:	$u[1] = 3$
Ecuación 21 se obtiene salida:	$Y_B[1] = 1$
Actualizar umbrales	$\theta_B[2] = V_\theta$
Guardar padres:	$R_B^P = s$
2. Cuando $n = 2$:	
Ecuación 22 cálculo de umbrales:	$\theta_C[2] = 8$
Ecuación 24 cálculo de velocidad de auto-onda:	$\Delta u[2] = 5$
Ecuación 25 cálculo de actividad interna:	$u[2] = 8$
Ecuación 21 se obtiene salida:	$Y_C[2] = 1$
Actualizar umbrales	$\theta_C[2] = V_\theta$
Guardar padres:	$R_C^P = B$

Se terminan las iteraciones de la Red ya que $Y_C = 1$. Se obtiene la ruta calculada recorriendo el vector de padres empezando por la neurona meta: $C \rightarrow R_C^P = B \rightarrow R_B^P = A$

En la Figura 33 se ilustra el funcionamiento del modelo SAM-PCNN buscando el camino más corto entre el nodo A y el nodo C en un grafo de tres nodos. En la primera iteración la actividad interna de la de la red es $u[0] = 0$. Se calcula el incremento de la actividad interna $\Delta u[1]$ con la Ecuación 24, buscando el valor mínimo del conjunto de umbrales que sean más altos que la actividad interna $u[0]$, dando como resultado $\Delta u[1] = 8$. Aplicando la Ecuación 25 la actividad interna $u[1] = 8$, alcanzaría al umbral de la neurona B generando un pulso como se indica en la Ecuación 21. En la Iteración 2, se calcularía el incremento de la actividad interna con la Ecuación 24, siendo $\Delta u[2] = 8$. Posteriormente se calcularía la actividad interna de la red con la Ecuación 25 dando como resultado $u[1] = 16$, lo que generaría un pulso en la neurona C de acuerdo con la Ecuación 21. Al final, SAM-PCNN tardaría dos iteraciones en resolver la trayectoria.

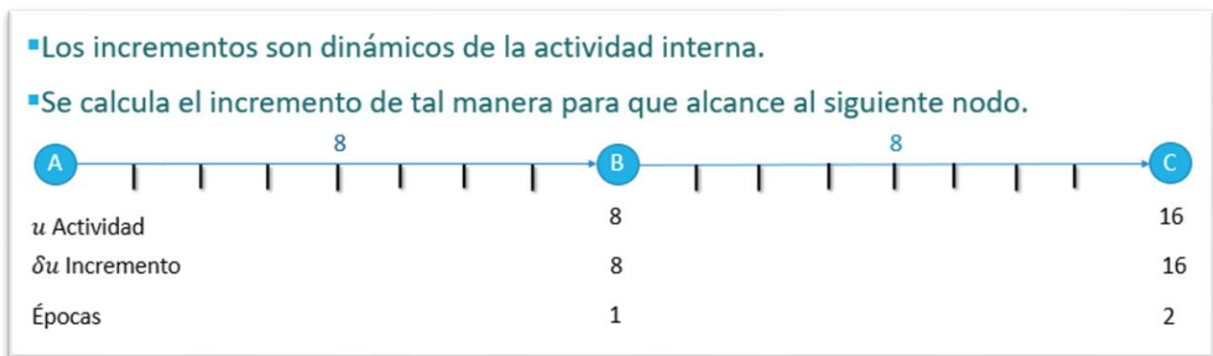


Figura 33. Grafo de tres nodos resuelto por el SAPCNN [Espinosa, 2019]

4.2 Planeación de Trayectorias

Se presentan los siguientes dos modelos de Redes Neuronales Pulso Acopladas orientados a la optimización de rutas utilizando mapas bidimensionales.

- MPCNN (Modified Pulse-Coupled Neural Network) [Hong, 2009]
- BSPCNN (Bidirectional Search Pulse-Coupled Neural Network) (Modelo propuesto).

A continuación, se dará una breve descripción de los modelos que fueron implementados.

4.2.1 Modified Pulse-Coupled Neural Network

El modelo *Modified Pulse-Coupled Neural Network* (MPCNN por sus siglas en inglés) [Qu, 2009] es la primera variante de RNAP diseñada específicamente para la Planeación de Trayectorias de Robots Móviles en tiempo real. Existe otro modelo llamado MPCNN (*Multi-Output Pulse Coupled Neural Network*) [Qu, 2007], sin embargo, es utilizado para problemas de Optimización de Trayectorias, no para Planeación de Trayectorias. El modelo MPCNN [Qu, 2009] está topológicamente organizado con sólo conexiones laterales locales entre las neuronas. En este modelo la Red Neuronal puede representar un espacio cartesiano que maneja distancias euclidianas entre las neuronas. En la Figura 34 se presenta la topología del modelo propuesto.

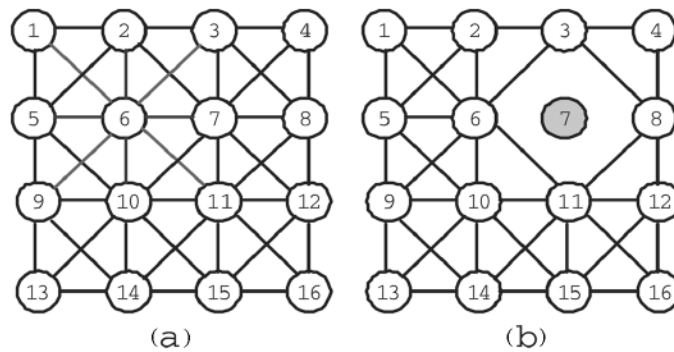


Figura 34. (a) Ejemplo de la topología de la Red sin obstáculos, (b) topología de la Red con un obstáculo [Qu, 2009].

La arquitectura de Red Neuronal del modelo MPCNN [Qu, 2009] es un mapa discreto topológicamente organizado y se puede expresar en un espacio de estados de dimensión finita. Cada neurona tiene sólo conexiones laterales locales con sus neuronas vecinas, que constituyen un subconjunto de la R_i , como se muestra en la Figura 35 (a). Tanto el robot como el objetivo se manejan como espacio libre. Las neuronas que representan un obstáculo no se comunican con las demás neuronas.

Los vecinos se indican mediante r^1, \dots, r^k . Cada neurona es, o bien un espacio libre, o una ubicación de barrera; los objetivos y el robot pueden ocupar cualquier espacio libre. La conexión ponderada w_{ij} de la neurona j a la neurona i es definida como la distancia euclidiana entre i y j dentro del espacio S . Si la neurona j no es vecina de la neurona i , se dice que la distancia $w_{ij} = \infty$. Si la neurona j representa un obstáculo dentro del espacio, se dice que la distancia es $w_{ij} = \infty$, como se muestra en la Figura 35(b).

Los vecinos de una neurona en particular están conectados de manera horizontal o vertical son miembros de un conjunto R^l (neuronas de tipo a , como se muestra en la Figura 35) si sus distancias euclidianas desde la neurona son 1. Del mismo modo, se dice que los vecinos están conectados de forma inclinada pertenecen al conjunto de vecinos R^r (neuronas de tipo b , como se muestra en la Figura 35) de la neurona si sus distancias euclidianas son $\sqrt{2}$. Las neuronas no necesitan estar separadas por una distancia unitaria (1 o $\sqrt{2}$ en dirección inclinada) una de la otra. Las neuronas se pueden colocar a

distancias mayores para entornos con menos obstáculos, siempre que se garantice el cumplimiento de las reglas de conexión recta e inclinada.

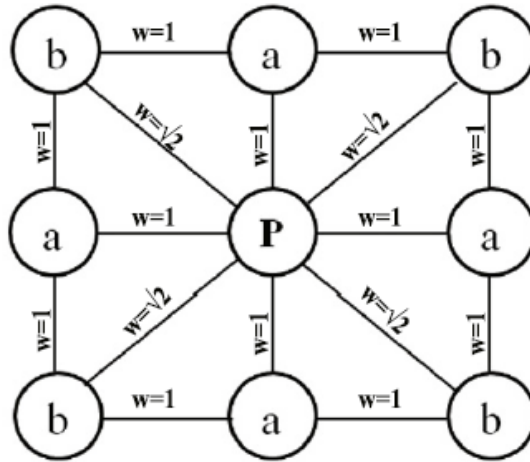


Figura 35. Topología de la Red con conexiones locales.

Una neurona i está conectada a cualquiera de sus vecinos inmediatos j a través de un peso w_{ij} cómo se indica en la ecuación 27:

$$w_{ij} = \begin{cases} 1 & \text{si } j \in R^l \text{ neuronas tipo } a \\ \sqrt{2} & \text{si } j \in R^r \text{ neuronas tipo } b \end{cases} \quad (27)$$

En la ecuación 28 se muestra la salida de una neurona i en el tiempo t , la cual está representada por $Y_i(t)$. Cada neurona se dispara sólo una vez durante la ejecución del algoritmo.

$$Y_i(t) = \begin{cases} 0 & \text{para } T - \varepsilon \leq t < T \\ 1 & \text{para } t = T \\ 0 & \text{para } T < t \leq T + \varepsilon \end{cases} \quad (28)$$

Una neurona i se dispara sólo si alguna neurona j en su vecindario N_i se dispara y esto hace que la neurona i sea una neurona hijo para activarse en algún momento posterior. Se dice que la neurona j es su padre temporal N_i^{P*} y su tiempo de activación se denota como $t^{N_i^{P*}}$. Si se dispara otra neurona en el mismo vecindario y si puede hacer que la neurona lo haga antes, se convertirá en el nuevo padre temporal de i . Cuando una neurona i dispara en algún momento, lo registra su padre temporal actual como su padre $t^{N_i^{P*}}$.

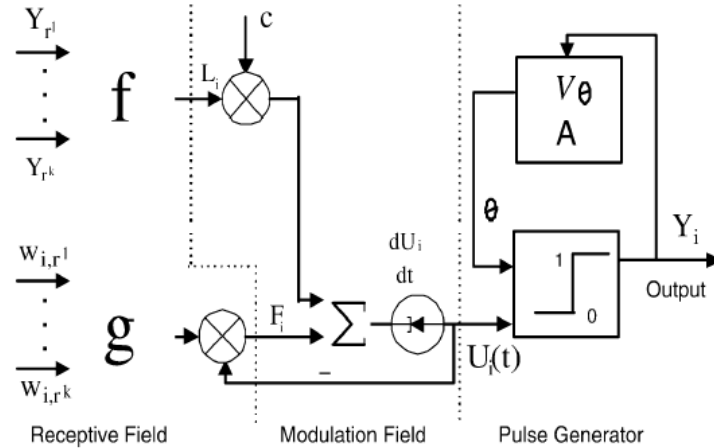


Figura 36. Diagrama del modelo MPCNN [Qu, 2009].

En el campo de recepción se tienen dos compartimientos f y g . El compartimiento g recibe la matriz de distancias W y f recibe los pulsos de la matriz de salida Y . En el campo de enlace, F se calcula con la multiplicación entre g y U , L adquiere el valor de f . Para calcular la actividad interna de la neurona U , se realiza la sumatoria entre F y L .

En el compartimiento generador de pulsos se realiza la comparación entre la actividad interna U y el umbral θ . Si $U \geq \theta$ entonces se genera un pulso en la matriz de salida Y . El umbral puede tener diferentes valores dependiendo de la salida Y . Si la neurona no ha pulsado $\theta = V$, en caso contrario $\theta = A$. V y A son constantes positivas.

Cada neurona $i, 1, 2, \dots, N$, tiene una salida Y_i .

$$Y_i(t) \begin{cases} 1, & \text{if } U_i(t) \geq \theta_i \\ 0, & \text{de otra manera} \end{cases} \quad (29)$$

Donde $U_i(t)$ y $\theta_i(t)$ son la actividad interna de la neurona y el umbral respectivamente, t es el tiempo, y N es el número de neuronas.

El mecanismo del umbral de cada neurona se representa de acuerdo con la ecuación 30.

$$\theta(t) = \begin{cases} A_{init}, & \text{cuando } t < t_{fire}^{R_i^F} \\ A_{ij}, & \text{cuando } t_{fire}^j \leq t < t_{fire}^i, j \in \xi(i, t) \\ V_\theta, & \text{cuando } t > t_{fire}^i \end{cases} \quad (30)$$

Para cada $i = 1, 2, \dots, N$, donde A_{init} y V_θ son constantes positivas. V_θ es un valor grande al que los autores le asignan el número total de neuronas en la Red. El valor de A_{ij} se calcula con la ecuación 31.

$$A_{ij} = \begin{cases} A^r, & \text{si } j \in R_i^r \\ A^l, & \text{si } j \in R_i^l \end{cases} \quad (31)$$

Donde A^r y A^l son constantes positivas.

El campo de enlace $L_i(t)$ y el campo de alimentación $F_i(t)$ de cada neurona se representan con las ecuaciones 32 y 33 respectivamente.

$$L_i(t) = f(Y_{r^1}, \dots, Y_{r^k}, t) = \begin{cases} 0, & \text{cuando } t < t_{fire}^{R_i^F} \\ 1, & \text{en otro caso} \end{cases} \quad (32)$$

$$F_i(t) = -g(w_{ir^1}, \dots, w_{ir^k}, t)U_i(t) \quad (33)$$

Donde $w_{ir^1}, \dots, w_{ir^k}$ representan las distancias entre la neurona i a sus k vecinos.

La actividad interna de cada neurona se calcula con la ecuación 34.

$$U_i(t) = \begin{cases} \frac{dU_j(t)}{dt} = F_i + CL_i = -g(w_{ir^1}, \dots, w_{ir^k}, t)U_i(t) + CL, & t \geq t_{fire}^{R_i^P} \\ U(t_{fire}^{R_i^P}) = 0 \end{cases} \quad (34)$$

Donde C es un constante positiva y la $g(\cdot)$ es una función de las salidas y los pesos ponderados de las neuronas vecinas y el tiempo t . La función g se define en la ecuación 35.

$$g_i(w_{ir^1}, \dots, w_{ir^k}, t) = \begin{cases} 0, & \text{cuando } t < t_{fire}^{R_i^F} \\ \mu(w_{ij}), & \text{cuando } t \geq t_{fire}^j, \text{ para cada } j \in \xi(i, t) \end{cases} \quad (35)$$

Donde $\mu(w_{ij})$ se define en la ecuación 36.

$$\mu(w_{ij}) = \frac{B}{w_{ij}} = \begin{cases} B, & \text{si } j \in R^l \\ \frac{B}{\sqrt{2}}, & \text{si } j \in R^r \end{cases} \quad (36)$$

B es una constante positiva cuyo valor es la distancia de una neurona i a una j en posición vertical u horizontal. Por otro lado, resolviendo la ecuación diferencial 35 se obtiene la ecuación 37.

$$U_i(t) = \frac{w_{iR_i^P}}{B} \left(1 - \exp \left[-\frac{B}{w_{iR_i^P}} (t - t_{fire}^{R_i^P}) \right] \right) \quad (37)$$

La definición matemática de la variante MPCNN puede ser expresada de la Ecuación 29 a la Ecuación 37. Sin embargo, es necesario establecer algunas condiciones iniciales para los parámetros de la Red Neuronal. Las condiciones para el modelo se definen en las ecuaciones 38, 39 y 40.

$$\text{Condición 1: } 0 < A_{ij} < \frac{w_{ij}}{B} C < V_\theta \quad (38)$$

$$\text{Condición 2: } 0 < A_{ik} < \frac{w_{ik}}{B} C < V_\theta \quad (39)$$

$$\text{Condición 3: } A_{ik} = \frac{w_{ik}}{w_{ij}} A_{ij} \quad (40)$$

En un caso estático, donde los obstáculos y los objetivos no se mueven, la MPCNN propuesta garantiza que el evento de disparo de las neuronas objetivo puede ser propagado a todo el espacio de estado a través de la conectividad local de las neuronas, mientras que el estado del obstáculo permanece sin disparar. Si la auto-onda alcanza la neurona i desde la neurona j , se guarda a la neurona j como el padre de la neurona i definiéndola como R_i^P . Cuando la auto-onda alcance a la neurona que representa el robot, la neurona dispara y se recupera la trayectoria generada desde la neurona del robot hasta la neurona del objetivo. Este camino representara el camino más corto entre el robot y su objetivo.

Para ofrecer mayor claridad sobre el funcionamiento de la variante MPCNN, en el Pseudo-Código 1 se presenta para mostrar los pasos de la Red propuesta cuando se utiliza para la planificación de la

Oliver Eder Guillermo Espinosa Meneses 62

trayectoria del robot. Las ideas innovadoras acerca de este algoritmo es que puede hacerse en paralelo, lo que resulta en computación eficiente de todo el sistema, especialmente en entornos dinámicos.

Pseudo-Código 1: MPCNN en un ambiente estático [Qu, 2009].

- 1) Se inicializa la Red: Las constantes A , B , C y V_θ se inicializan de acuerdo con las condiciones 1, 2 y 3 de las Ecuaciones 38, 39 y 40. $\theta_i(0) = A_{init}$, $U_i(0) = 0$ para todas las neuronas menos para la neurona de inicio. $U_{inicio}(0) = A_{init}$, lo que causa que se dispare la neurona de inicio.
- 2) Inicio de la Red: para cada neurona i en la Red:
 - a. Calcular $U_i(t)$ de acuerdo con la ecuación 37;
 - b. Calcular $Y_i(t)$ de acuerdo con la ecuación 29;
 - c. Si $Y_i(t) = 1$;
 - i. $\theta_i(t) = V_\theta$;
 - ii. Para $j \in R_i$, si $i \in \xi(i, t)$ entonces $\theta_i(t) = A_{ij}$;
 - iii. Se guardan el padre de las neuronas i en R_i^P
- 3) Si $Y_{robot} = 1$, se termina el proceso

Cuando la auto-onda se propague a la posición del robot, existe una secuencia de nodos padres a partir de la neurona objetivo y terminando en la neurona robot. A través de la secuencia de padres, es posible determinar el movimiento del robot.

4.2.2 Bidirectional Search Pulse-Coupled Neural Network

Además de trabajar con el modelo MPCNN [Qu, 2009] se propone una nueva variante de PCNN diseñada para resolver el problema de la Planeación de Trayectorias Discretas 2D. El modelo propuesto se inspira en el modelo presentado en [Sang, 2010], donde se propone una Red Neuronal Pulsante que es capaz de realizar una búsqueda dual, la cual proporciona mayor eficiencia computacional. El modelo *Dual Source Pulse-Coupled Neural Network* [Sang, 2010] (DSPCNN) produce dos auto-ondas, la primera que parte desde la neurona de inicio y la segunda que parte desde la neurona objetivo. Una vez que se encuentran las auto-ondas de las dos fuentes de disparo, se detiene el proceso de búsqueda y se extrae la ruta obtenida. En la Figura 37 se presenta la idea propuesta en el modelo DSPCNN [Sang, 2010].

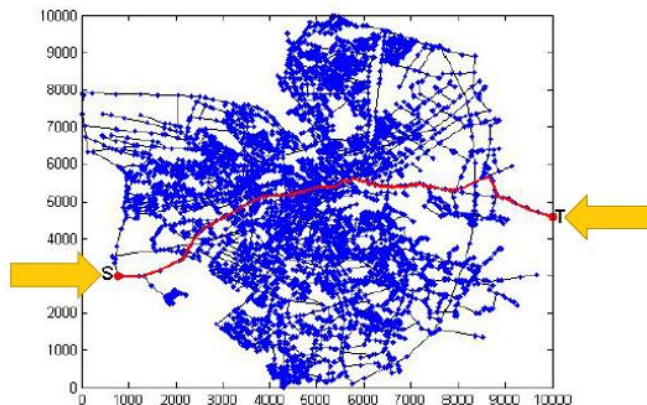


Figura 37. Ejemplo de una búsqueda dual [Sang, 2010].

La variante propuesta *Bidirectional Search Pulse-Coupled Neural Network* (BSPCNN, por sus siglas en inglés), tomó como base el mecanismo y la topología del modelo MPCNN [Qu, 2009]. La Red representa un mapa discreto topológicamente organizado y se puede expresar en un espacio de estados de dimensión finita. Se tomó como referencia la topología de la Red presentada en [Qu, 2009] y las relaciones entre las neuronas. Cada neurona tiene sólo conexiones laterales locales con sus neuronas vecinas. Las neuronas que representan un obstáculo no se comunican con las demás neuronas. Cada neurona es o bien un espacio libre o una ubicación de barrera. El robot y el objetivo se manejan como un espacio libre. Las neuronas que representan un obstáculo no se comunican con las demás neuronas. La conexión ponderada w_{ij} de la neurona j a la neurona i es definida como la distancia euclidiana entre i y j dentro del espacio S . Si la neurona j no es vecina de la neurona i , se dice que la distancia $w_{ij} = \infty$. Si la neurona j representa un obstáculo dentro del espacio, se dice que la distancia es $w_{ij} = \infty$.

A diferencia de [Qu, 2009], la variante BSPCNN se implementó con la capacidad de realizar una búsqueda bidireccional dentro de un espacio discreto. En lugar de tener una sola Red con dos auto-ondas como en el modelo de [Sang, 2010], se implementaron dos redes con una auto-onda cada una. En la Figura 38 se muestran dos redes representando el mismo espacio discreto. La Red a iniciará la propagación de la auto-onda del nodo 1, mientras la Red b comenzará la búsqueda del nodo 15 y su objetivo el nodo 1.

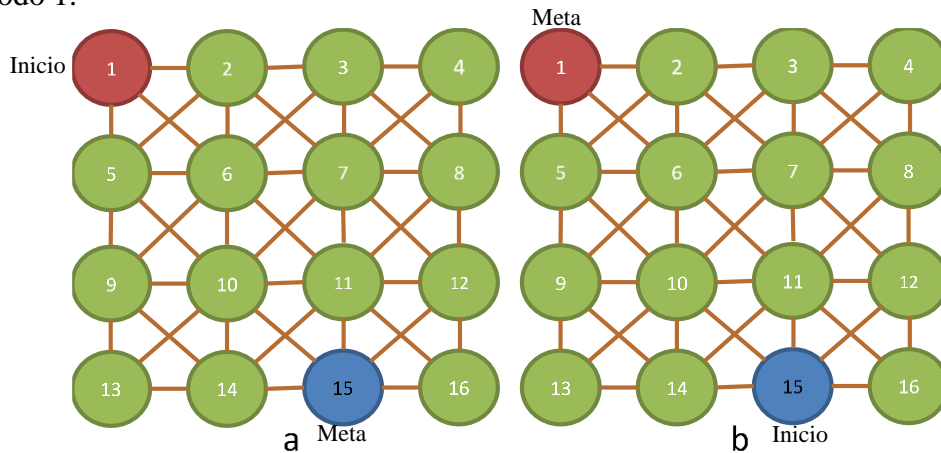


Figura 38. La Red a genera la auto-onda desde el nodo 1 y la Red b desde el nodo 15.

La definición matemática de la variante BSPCNN se asemeja a la presentada en [Qu, 2009], no obstante, presenta diferencias sustanciales las cuales se presentan a continuación:

BSPCNN tiene dos Redes superpuestas que representan el mismo espacio discreto. La Red a inicia la auto-onda desde la posición del robot y la Red b inicia la auto-onda desde la posición objetivo. La conexión ponderada w_{ij}^a de la neurona i^a a la neurona j^a es definida como la distancia euclidiana entre i^a y j^b dentro de la Red a . Para la Red b la distancia entre la neurona i^a y j^b es w_{ij}^b .

En la ecuación 42 se muestra la salida de una neurona i^a en el tiempo t , la cual está representada por $Y_i^a(t)$. Lo mismo sucede con las neuronas de la Red b , lo cual se muestra en la Ecuación 43. Cada neurona se dispara sólo una vez durante la ejecución del algoritmo.

$$Y_i^a(t) = \begin{cases} 1, & \text{si } U_i^a(t) \geq \theta_i^a \\ 0, & \text{de otra manera} \end{cases} \quad (42)$$

$$Y_i^b(t) = \begin{cases} 1, & \text{si } U_i^b(t) \geq \theta_i^b \\ 0, & \text{de otra manera} \end{cases} \quad (43)$$

Donde $U_i^a(t)$ y $U_i^b(t)$ son la actividad interna de las neuronas en la Red a y b respectivamente. Los umbrales son representados por θ_i^a y θ_i^b y t es el tiempo. Los umbrales son calculados como se indica en la ecuación 30 del modelo MPCNN. La actividad interna de las neuronas de cada Red se calcula de acuerdo con la ecuación 37 del modelo MPCNN.

Dentro de la Red a , si una neurona i^a es alcanzada por la auto-onda proveniente de una neurona j^a , se guarda la neurona j^a como el padre de la neurona i^a utilizando la notación $R_{i^a}^P$. Lo mismo sucede en la Red b , donde se guarda a la neurona j^b como el padre de la neurona i^b con la notación $R_{i^b}^P$, cuando la neurona i^b es alcanzada por la auto-onda proveniente de una neurona j^b . Cuando la onda de disparo de la Red a alcanza una posición alcanzada por la auto-onda de la Red b , la neurona donde las auto-ondas convergieron se disparará y se determina una ruta desde dicha neurona a la neurona inicio moviéndose sucesivamente hacia el padre de cada neurona en la Red a . Del mismo modo, se determina la ruta desde la neurona de convergencia a la neurona objetivo en la Red b . Posteriormente se combina la lista de nodos padre de ambas Redes. Éste será el camino más corto desde la neurona de inicio a la neurona meta.

Como en el modelo DSPCNN [Sang, 2010], la condición de paro consiste en verificar en cada t donde se genera un pulso en alguna neurona si las auto-ondas encuentran el mismo nodo. No obstante, debido a la topología de Red del modelo BSPCNN, es frecuente que las auto-ondas se encuentren en más de un punto en determinado t , por lo cual, es necesario guardar todos los nodos en donde las auto-ondas se encuentran para extraer todas las rutas encontradas y determinar cuál es la menor de todas. En la Figura 39 se muestra un ejemplo de cómo las auto-ondas se pueden encontrar en más de un punto, en este caso en los puntos azules.

Para tener mayor claridad del funcionamiento del modelo propuesto, se presenta el siguiente pseudocódigo para mostrar los pasos de la Red propuesta cuando se utiliza para resolver la Planeación de Trayectorias. Aquí la neurona de inicio se representa con una s , la neurona meta se representa con una g . Uno de los aspectos importantes de este algoritmo y de las PCNN en general, es que se puede hacer en paralelo, lo que resulta en un cálculo más eficiente, especialmente con búsquedas de gran magnitud.

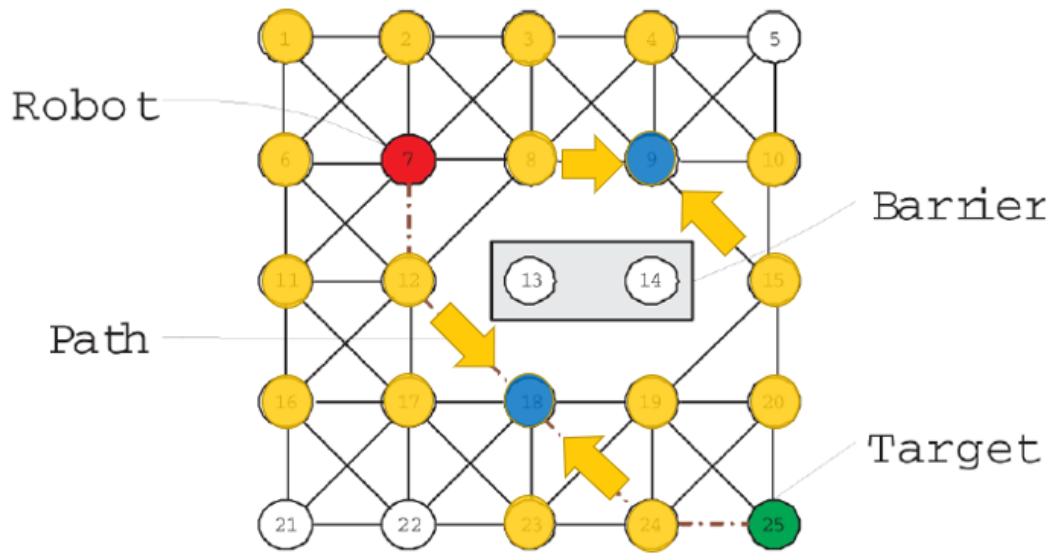


Figura 39. Ejemplo de auto-ondas encontrándose en más de un punto.

Pseudo-Código 2: BSPCNN en un ambiente estático.

- 1) Se inicializa la Red: Las constantes A , B , C y V_θ se inicializan de acuerdo con las condiciones iniciales 1, 2 y 3 de las Ecuaciones 38, 39 y 40, presentadas en el modelo MPCNN de [Qu, 2009]. $\theta_i^a(0) = A_{init}$, $\theta_i^b(0) = A_{init}$, $U_i^a(t) = 0$ y $U_i^b(t) = 0$ para todas las neuronas de las redes a y b menos para la neurona de inicio. $U_{inicio}^a(t) = A_{init}$ y la neurona objetivo $U_{robot}^b(t) = A_{init}$, lo que causa que se disparen las neuronas donde se inician las búsquedas.
 - 2) Inicio de la Red: para cada neurona i en la Red:
 - a. Calcular $U_i^a(t)$ de acuerdo con la ecuación 37;
 - b. Calcular $U_i^b(t)$ de acuerdo con la ecuación 37;
 - c. Calcular $Y_i^a(t)$ de acuerdo con la ecuación 42;
 - d. Calcular $Y_i^b(t)$ de acuerdo con la ecuación 43;
 - e. Si $Y_i^a(t) = 1$;
 - i. $\theta_i^a(t) = V_\theta$;
 - ii. Para $j^a \in R_{i^a}$, si $i^a \in \xi(i^a, t)$ entonces $\theta_i^a(t) = A_{ij}$;
 - iii. Se guardan el padre de las neuronas i^a en $R_{i^a}^P$
 - f. Si $Y_i^b(t) = 1$;
 - i. $\theta_i^b(t) = V_\theta$;
 - ii. Para $j^b \in R_{i^b}$, si $i^b \in \xi(i^b, t)$ entonces $\theta_i^b(t) = A_{ij}$;
 - iii. Se guardan el padre de las neuronas i^b en $R_{i^b}^P$
 - 3) Si $Y_i^a(t) = 1 \wedge Y_i^b(t) = 1$, se termina el algoritmo.
-

Para ilustrar el funcionamiento del modelo BSPCNN, se toma como ejemplo el grafo presentado en la Figura 39, donde se busca la trayectoria del nodo 25 al nodo 7. Se inicializan las constantes del modelo como se indicia a continuación: n es igual al número de neuronas en la Red, $A^r = \sqrt{2}C(1 - e^{-1}) = 8.93953467$, $A^l = C(1 - e^{-1}) = 6.32120559$, $B = 1$, $C = 10$, $V_\theta = n$.

Cuando $t = 0$:

Se genera un pulso en las neuronas donde se inicia la búsqueda.

Pulso en la neurona 7 de la Red a $Y_7^a = 1$.

Pulso en la neurona 25 de la Red b $Y_{25}^b = 1$.

Utilizando la ecuación 38, se calcula $U_i(t)$ donde t es el tiempo actual y $t_{Fire}^{R_i^P}$ es el tiempo en que disparo el padre.

$$U_i^a(t) = \frac{1*10}{1} (1 - \exp\left[-\frac{1}{1}(0 - 0)\right]) = 0 \quad i = 2, 6, 8, 12$$

$$U_i^a(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(0 - 0)\right]) = 0 \quad i = 1, 3, 11$$

$$U_i^b(t) = \frac{1 * 10}{1} (1 - \exp\left[-\frac{1}{1}(0 - 0)\right]) = 0 \quad i = 20, 24$$

$$U_{19}^b(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(0 - 0)\right]) = 0$$

Se guardan los padres de cada neurona.

$$R_1^{Pa} = 7, R_2^{Pa} = 7, R_3^{Pa} = 7, R_6^{Pa} = 7, R_8^{Pa} = 7, R_{11}^{Pa} = 7, R_{12}^{Pa} = 7$$

$$R_{19}^{Pb} = 25, R_{20}^{Pb} = 25, R_{24}^{Pb} = 25$$

Cuando $t = 1$:

Se calcula la actividad interna con la Ecuación 38.

$$U_i^a(t) = \frac{1 * 10}{1} (1 - \exp\left[-\frac{1}{1}(1 - 0)\right]) = 6.3212 \quad i = 2, 6, 8, 12$$

$$U_i^a(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(1 - 0)\right]) = 7.1690 \quad i = 1, 3, 11$$

$$U_i^b(t) = \frac{1 * 10}{1} (1 - \exp\left[-\frac{1}{1}(1 - 0)\right]) = 6.3212 \quad i = 20, 24$$

$$U_{19}^b(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(1 - 0)\right]) = 7.1690$$

Se calcula la función de disparo con las Ecuaciones 42 y 43.

$$Y_i^a = 1 \quad i = 2, 6, 8, 12$$

$$Y_i^b = 1 \quad i = 20, 24$$

Se guardan los padres de cada neurona.

$$R_9^{Pa} = 8, R_4^{Pa} = 8, R_{16}^{Pa} = 12, R_{17}^{Pa} = 12, R_{18}^{Pa} = 12$$

$$R_{15}^{Pb} = 20, R_{18}^{Pb} = 24, R_{23}^{Pb} = 24$$

Cuando $t = \sqrt{2}$:

Se calcula la actividad interna con la Ecuación 38.

$$U_i^a(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(\sqrt{2} - 0)\right]) = 8.9395 \quad i = 1, 3, 11$$

$$U_{19}^b(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(\sqrt{2} - 0)\right]) = 8.9395$$

Se calcula la función de disparo con las Ecuaciones 42 y 43.

$$Y_i^a = 1 \quad i = 1, 3, 11$$

$$Y_i^b = 1 \quad i = 19$$

Se guardan los padres de cada neurona, sin embargo, en esta iteración no se registra ningún padre.

Cuando $t = 2$:

Se calcula la actividad interna con la Ecuación 38.

$$U_i^a(t) = \frac{1*10}{1} (1 - \exp\left[-\frac{1}{1}(2 - 1)\right]) = 6.3212 \quad i = 17,9$$

$$U_i^a(t) = \frac{\sqrt{2}*10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(2 - 1)\right]) = 7.1690 \quad i = 16, 18, 4$$

$$U_i^b(t) = \frac{1*10}{1} (1 - \exp\left[-\frac{1}{1}(2 - 1)\right]) = 6.3212 \quad i = 15,23$$

$$U_{18}^b(t) = \frac{\sqrt{2} * 10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}(2 - 1)\right]) = 7.1690$$

Se calcula la función de disparo con las Ecuaciones 42 y 43.

$$Y_i^a = 1 \quad i = 17, 9$$

$$Y_i^b = 1 \quad i = 15, 23$$

Se guardan los padres de cada neurona.

$$R_{21}^{Pa} = 17, R_{22}^{Pa} = 17, R_{23}^{Pa} = 17, R_5^{Pa} = 9, R_{10}^{Pa} = 9, R_{15}^{Pa} = 9$$

$$R_9^{Pb} = 15, R_{10}^{Pb} = 15, R_{22}^{Pb} = 23, R_{22}^{Pb} = 17$$

Cuando $t = 1 + \sqrt{2}$:

Se calcula la actividad interna con la Ecuación 38.

$$U_i^a(t) = \frac{\sqrt{2}*10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}((1 + \sqrt{2}) - 1)\right]) = 8.9395 \quad i = 16, 18, 4$$

$$U_i^b(t) = \frac{\sqrt{2}*10}{1} (1 - \exp\left[-\frac{1}{\sqrt{2}}((1 + \sqrt{2}) - 1)\right]) = 8.9395 \quad i = 18$$

Se calcula la función de disparo con las Ecuaciones 42 y 43.

$$Y_i^a = 1 \quad i = 16, 18, 4$$

$$Y_i^b = 1 \quad i = 18$$

Se guardan los padres de cada neurona.

$$R_{19}^{Pa} = 18, R_{24}^{Pa} = 18$$

$$R_{12}^{Pb} = 18$$

Se cumple la condición de paro $Y_{18}^a(t) = 1 \wedge Y_{18}^b(t) = 1$, se termina el algoritmo.

4.3 Modificación de Matriz de Adyacencia

Los modelos de PCNN implementados en esta tesis utilizan la Matriz de Adyacencias de los grafos para realizar la búsqueda de una determinada trayectoria entre un punto y otro. En un Grafo con n nodos se genera una Matriz de Adyacencias de magnitud $n \times n$, es decir, en un grafo de 200 nodos se generaría una matriz de 200×200 posiciones. Sin embargo, cuando se trabajó con mapas bidimensionales, los grafos llegan a tener por lo menos unos 40,000 nodos, lo que generaría una Matriz de Adyacencias de $40,000 \times 40,000$ posiciones. No obstante, En la Figura 35 se muestra que la topología de los grafos

utilizados por los algoritmos de Planeación de Trayectorias está formada únicamente por conexiones locales, por lo que cada nodo está conectado con un máximo de 8 nodos vecinos.

Para reducir el espacio de búsqueda utilizado por los algoritmos de PCNN para Planeación de Trayectorias que utilizan mapas bidimensionales, se utilizó una estructura diferente para almacenar la información de la Matriz de Adyacencias. La estructura consiste en dos matrices superpuestas de $n \times 8$, una de las matrices contiene los vecinos de cada neurona y la otra contiene las distancias hacia dichos vecinos. Para un mapa con una dimensión de 200×200 nodos, se generarían dos matrices de $40,000 \times 8$ valores, lo que equivaldría a 640,000 posiciones, en lugar de 1,600,000,000 de posiciones de memoria que requeriría la Matriz de Adyacencias clásica.

4.3 Detalles de implementación

A continuación, se describen los detalles de la implementación de los sistemas y algoritmos utilizados para realizar los experimentos de esta investigación. Se utilizaron las cualidades del lenguaje de programación compilado C++ para el manejo de memoria dinámica.

4.3.1 Implementación del sistema de algoritmos que utilizan grafos

Las entradas para todos los modelos de PCNN que se implementaron para optimizar trayectorias utilizando grafos, son las matrices de adyacencias de los grafos que representan el espacio de búsqueda. Las matrices de adyacencias con las que se trabajó contenían la información de las distancias de grafos con cientos de nodos. Al tener grafos con dimensiones tan grandes fue necesario guardar las matrices en archivos de texto. Por tal motivo fue necesario desarrollar un programa que fuera capaz de proporcionar toda la información que cada modelo necesita. En la Figura 40 se presenta el diagrama de los módulos y funciones que componen el software desarrollado.

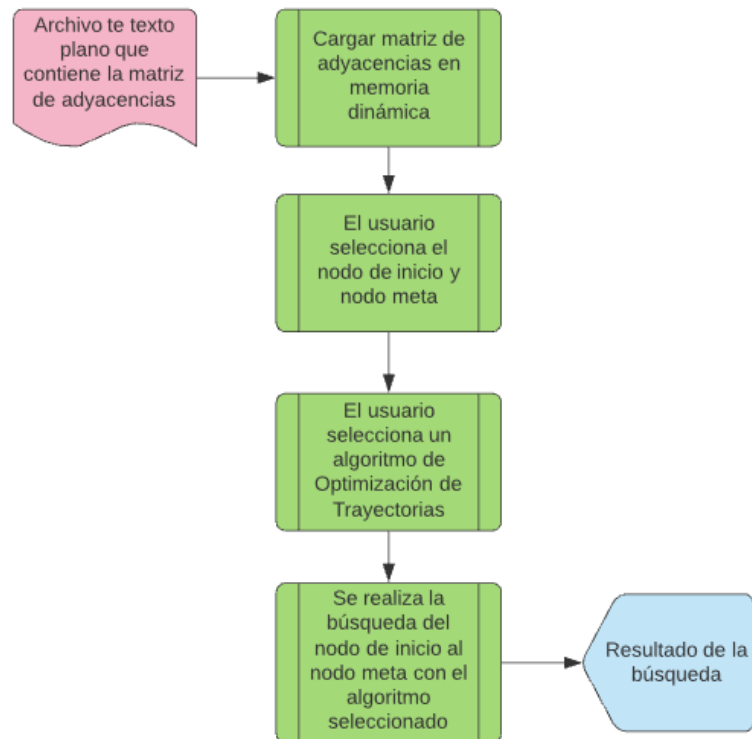


Figura 40. Diagrama del sistema de algoritmos que utilizan grafos.

La función principal del sistema contiene a la función *menú_principal*, dicha función muestra un menú donde se puede elegir a los principales módulos del sistema: Cargar grafo (*cargar_grafo*), Seleccionar nodo (*eleccionar_nodos*), Seleccionar algoritmo (*seleccionar_algoritmo*) e Iniciar búsqueda (*run*). En el módulo de Cargar grafo (*cargar_grafo*) se muestra una lista de los archivos *txt* contenidos en una carpeta específica utilizando la función *lista_archivos*. Una vez que se selecciona el archivo, se carga su contenido en una matriz dinámica utilizando la función *carga_matriz*.

En el módulo de Seleccionar nodos (*eleccionar_nodos*), se desplegará un vector con los nodos del grafo, el usuario seleccionará el nodo de inicio y el nodo objetivo de la búsqueda. En Seleccionar algoritmo (*seleccionar_algoritmo*) aparecen listados todos los modelos implementados: AWNN [Ma, 2011], AWNN E [Paredes-Cano, 2018], SAPCNN [Li, 2013], SAM-PCNN [Espinosa, 2019]. El usuario seleccionará un algoritmo y una vez seleccionado un algoritmo se podrá ingresar a la función *run para* correr el algoritmo seleccionado en (*seleccionar_algoritmo*) para realizar la búsqueda del nodo de inicio al nodo meta utilizando como entrada la matriz de adyacencias. Al término de la búsqueda se muestran los resultados generados. En la Figura 41 se muestra el diagrama de los módulos que componen al sistema.

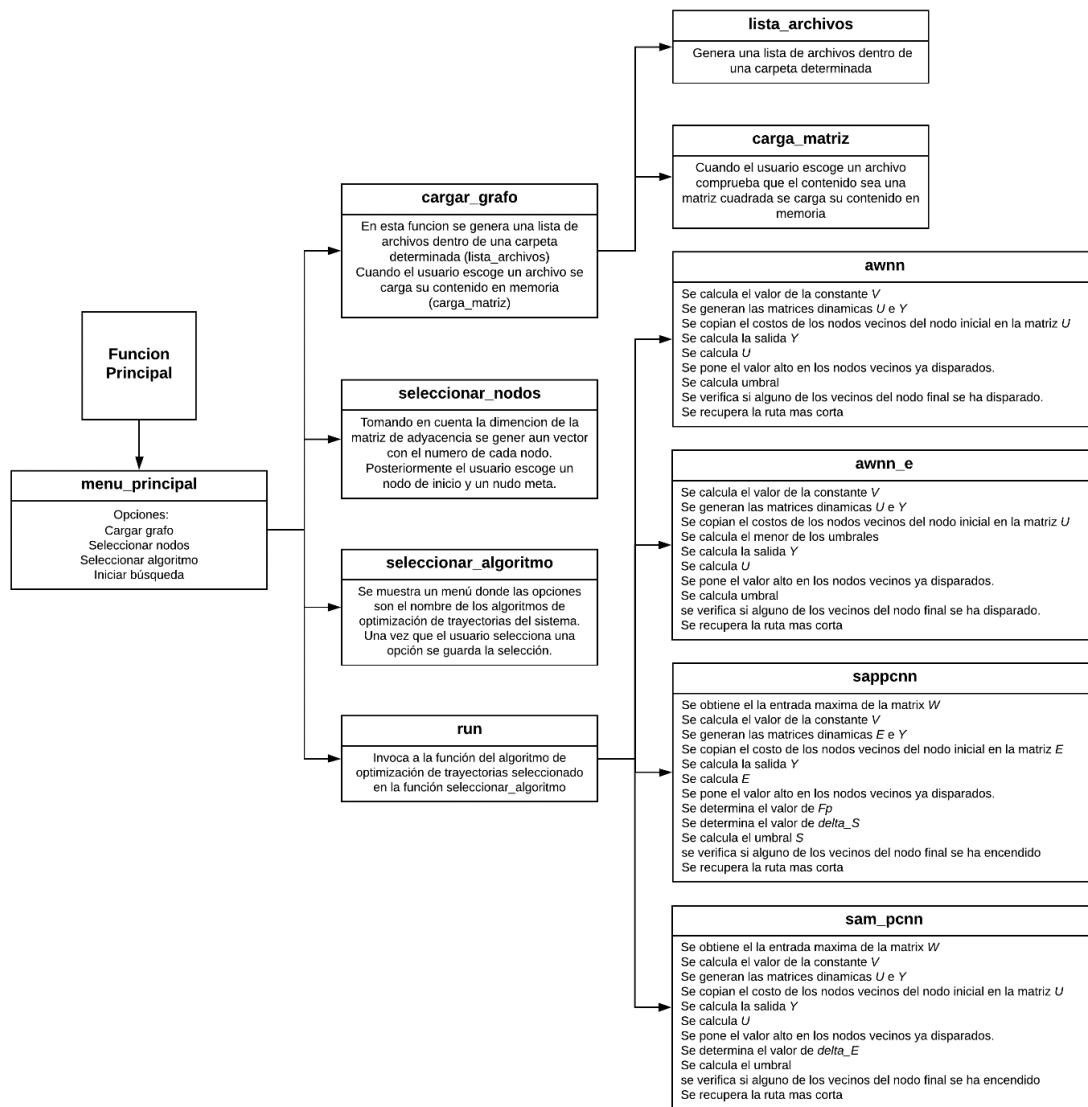


Figura 41. Diagrama de los módulos del sistema de algoritmos que utilizan grafos.

4.3.2 Implementación de sistema de algoritmos que utilizan mapas

Se implementó un sistema para experimentar con los algoritmos de PCNN que realizan Planeación de Trayectorias utilizando mapas bidimensionales. El sistema se realizó en C++, implementando una interfaz gráfica usuario, utilizando el Entorno de Desarrollo *QtCreator* y la biblioteca *OpenCV* para el tratamiento de imágenes digitales. En la Figura 41 se muestra el esquema del funcionamiento del sistema y en la Figura 42 se presenta un diagrama con los módulos que conforman el sistema.

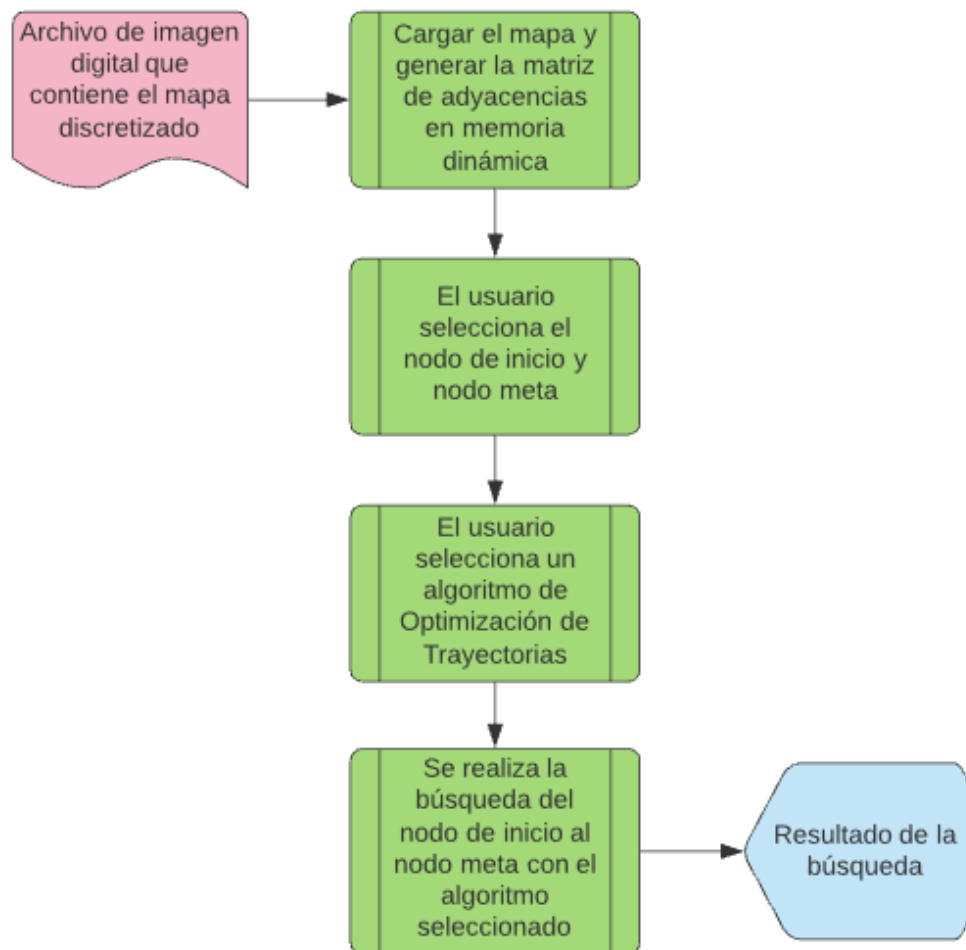


Figura 42. Diagrama del sistema de algoritmos que utilizan mapas.

El diagrama del sistema de los algoritmos que realizan Planeación de Trayectorias con mapas es similar al de los algoritmos que optimizan trayectorias utilizando grafos. En el módulo de Abrir Mapa se selecciona el archivo de una imagen digital que contenga un mapa binario donde los píxeles en blanco representen espacio libre y los píxeles negros representen obstáculos, una vez seleccionado se cargara la imagen en el sistema. En la Figura 44 se muestra un mapa cargado en el sistema.

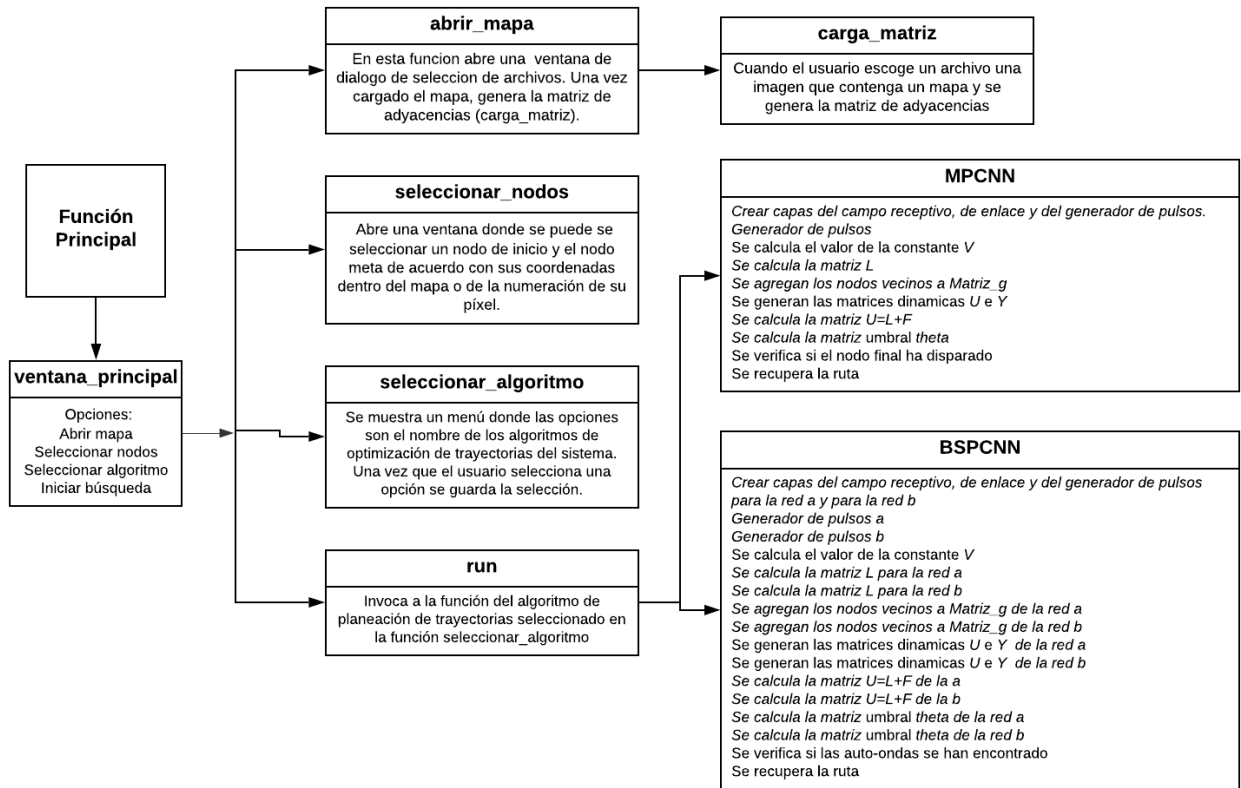


Figura 43. Diagrama de los módulos del sistema de algoritmos que utilizan mapas.

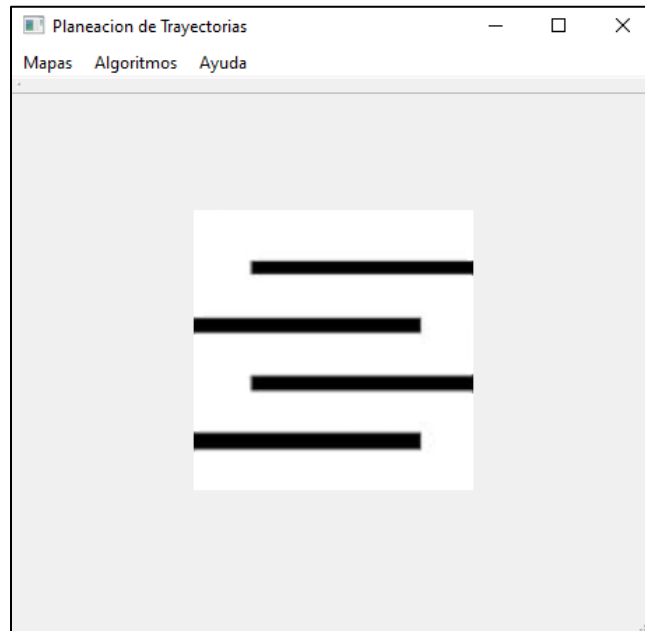


Figura 44. Ejemplo de mapa cargado en el sistema.

En el módulo Seleccionar nodos ubicado en el menú Mapas, abre una ventana donde se puede seleccionar un nodo de inicio y el nodo meta de acuerdo con sus coordenadas dentro del mapa o de la numeración de su píxel. En la Figura 45 se muestra la ventana donde se han escogido los nodos.

el objetivo de encontrar la solución reduciendo número de iteraciones necesarias para encontrar una solución.

Como resultado del estudio y análisis de los modelos de PCNN de Optimización de Trayectorias sobre grafos presentados en la literatura, se planteó una nueva variante de Red Neuronal Pulso Acoplada. La variante SAM-PCNN cuenta con una auto-onda que se propaga de manera dinámica, lo cual le permite alcanzar por lo menos a una neurona en cada iteración y así eliminar las iteraciones donde no ocurre ningún cambio en la Red. El mecanismo de SAM-PCNN le da ventaja en búsquedas de grafos donde las distancias entre cada nodo son considerables, sin embargo, cuando se trata de grafos donde las distancias son muy cortas puede llegar a requerir más tiempo de procesamiento que otros algoritmos como el AWNN.

Por otro lado, en el ámbito de la Planeación de Trayectorias sobre mapas, se estudió al modelo MPCNN propuesto en [Qu, 2009]. Dicho modelo presenta un mecanismo que le permite buscar el Camino más Corto dentro de un espacio discretizado en dos dimensiones. El mecanismo de la auto-onda le permite avanzar de forma constante a través del tiempo, siendo el tiempo equivalente a la distancia recorrida. Finalmente, se presenta una nueva variante llamada *Bidirectional Search Pulse-Coupled Neural Network* (BSPCNN). La variante BSPCNN se basa en el modelo MPCNN [Qu, 2012], introduciendo una segunda Red que representa el mismo espacio discreto. Cada Red cuenta con una auto-onda que le permite realizar una búsqueda en dos direcciones. El mecanismo de búsqueda bidireccional le proporciona a la BSPCNN la habilidad de encontrar una Trayectoria en la mitad de las iteraciones realizadas por el modelo MPCNN.

Finalmente se realizó un trabajo exhaustivo en la implementación de cada uno de los algoritmos de PCNN. Todos los algoritmos fueron implementados en C++. Las variantes de PCNN para Optimización de Trayectorias reciben una matriz numérica de adyacencias, mientras que para las variantes para Planeación de Trayectorias fue necesario utilizar la librería OpenCV para procesar imágenes de mapas binarios.

En el siguiente capítulo se presentan los experimentos realizados con los modelos de PCNN de Optimización de trayectorias sobre grafos y de Planeación de Trayectorias mapas bidimensionales presentados en la literatura.

Capítulo 5. Experimentación y resultados

Se realizó la implementación de los modelos seleccionados en dos tipos de problemas diferentes. Los primeros en el campo de Optimización de Trayectorias, resolviendo el problema del camino más corto (grafos). El segundo en el campo de la Planeación de Trayectorias para Robots Móviles (mapas). La implementación de los algoritmos se hizo en C++ de manera secuencial, y se compiló y ejecutó en un equipo de cómputo con un microprocesador i5-7200U de séptima generación @ 2.50 GHz 2.70 GHz con 8 GB de memoria RAM. Para el registro del tiempo de procesamiento se presenta un tiempo promediado de 100 repeticiones.

5.1 Experimentos de Optimización de Trayectorias (Grafos)

En esta Sección se presentan experimentos que permiten contrastar la eficiencia de las variantes de PCNN para optimización de trayectorias. Para realizar los experimentos se estudiaron cuidadosamente y se implementaron los modelos de PCNN propuestos por [Ma, 2011], [Paredes-Cano, 2019] y [Li, 2013].

5.1.1 Experimento 1

Para el primer experimento se utilizó un grafo asimétrico con diez nodos y veinte aristas presentado en [Li, 2013] el cual se muestra en la Figura 47. Se buscaron 10 caminos diferentes dentro del grafo. Las búsquedas se originaron desde todos los nodos al nodo 10 y del nodo 10 al nodo 1. En la Tabla 5 se presentan los resultados de cada uno de los algoritmos al realizar estas búsquedas.

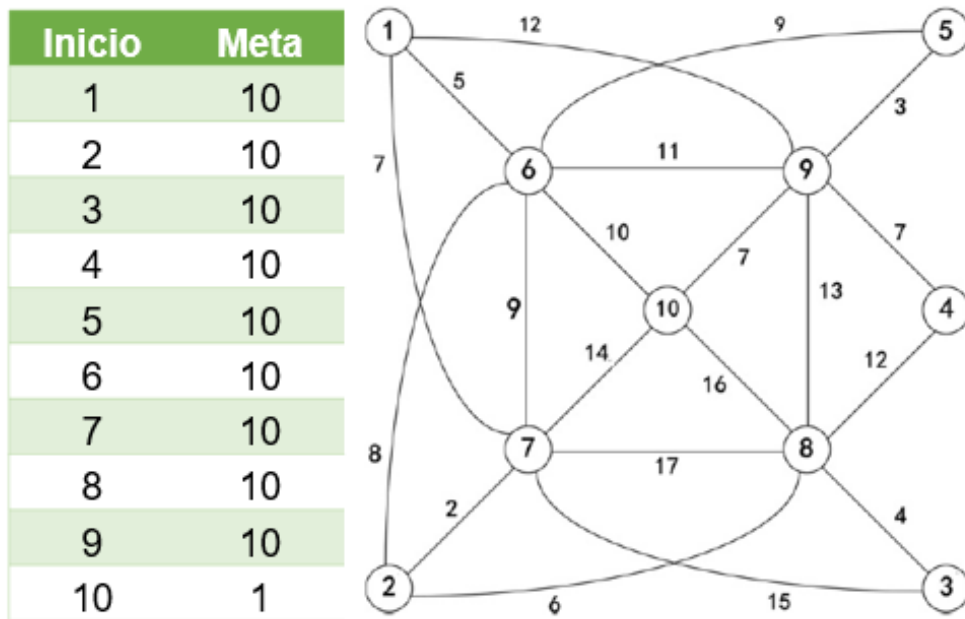


Figura 47. Búsquedas realizadas sobre el grafo asimétrico con 10 nodos y 20 aristas [Li, 2013].

Tabla 5. Resultado de las iteraciones por algoritmo del Experimento 1.

Experi- mento	Ruta	Costo	Iteraciones			
			AWNN [1]	AWNNE [14]	SAPCNN [15]	SAM-PCNN
1	1-6-10	15	15	11	9	6
2	2-7-10	16	16	16	10	6
3	3-8-10	20	20	17	13	8
4	4-9-10	14	14	8	8	4
5	5-9-10	10	10	8	5	3
6	6-10	10	10	6	6	4
7	7-10	14	14	13	9	6
8	8-10	16	16	13	11	8
9	9-10	7	7	5	4	2
10	10-6-1	15	15	9	9	4
Total			137	106	84	51
Diferencia con respecto a SAM-PCNN			+86	+55	+33	0

En la Tabla 5 se puede apreciar que los algoritmos AWNN [Ma, 2011] y AWNN Explicitada [Paredes-Cano, 2018], necesitan una mayor cantidad de iteraciones para el cálculo de la ruta. Al modelo AWNN [Ma, 2010] le toma una iteración por cada unidad de distancia. Es decir, si la distancia del camino más corto es 15 como en el ejercicio 1, requerirá de 15 iteraciones para resolver el problema. Al modelo AWNN Explicitada [Paredes-Cano, 2018] le toma menos iteraciones debido a que la velocidad de la auto-onda es dinámica en la primera iteración y posteriormente se vuelve constante.

Al analizar los resultados del modelo SAPCNN [Li, 2013], es notorio que puede encontrar el camino más corto en menos iteraciones, ya que cuenta con una auto-onda que se propaga de manera dinámica dependiendo del estado de la Red, lo que le permite a su actividad interna cumplir fácilmente con las condiciones necesarias para generar pulsos en las neuronas de la Red. Sin embargo, a pesar del mecanismo dinámico de este modelo, aún existen iteraciones donde no hay cambios en la Red, es decir, iteraciones donde ninguna de sus neuronas pulsa.

Por otro lado, el modelo propuesto SAM-PCNN presenta una menor cantidad de iteraciones para encontrar el camino más corto. El modelo cuenta con un mecanismo de auto-onda que adapta su velocidad de manera heurística, lo cual le permite optimizar el número de iteraciones necesarias para el cálculo de la ruta. En total el modelo propuesto SAM-PCNN hizo 86 iteraciones menos que el modelo AWNN [Ma, 2010], 55 iteraciones menos que el modelo AWNN Explicitada [Paredes-Cano, 2018] y 33 iteraciones menos que SAPCNN [Li, 2013]. En la Figura 48 se muestra de manera gráfica la comparativa de las iteraciones realizadas por cada modelo en cada uno de los experimentos.

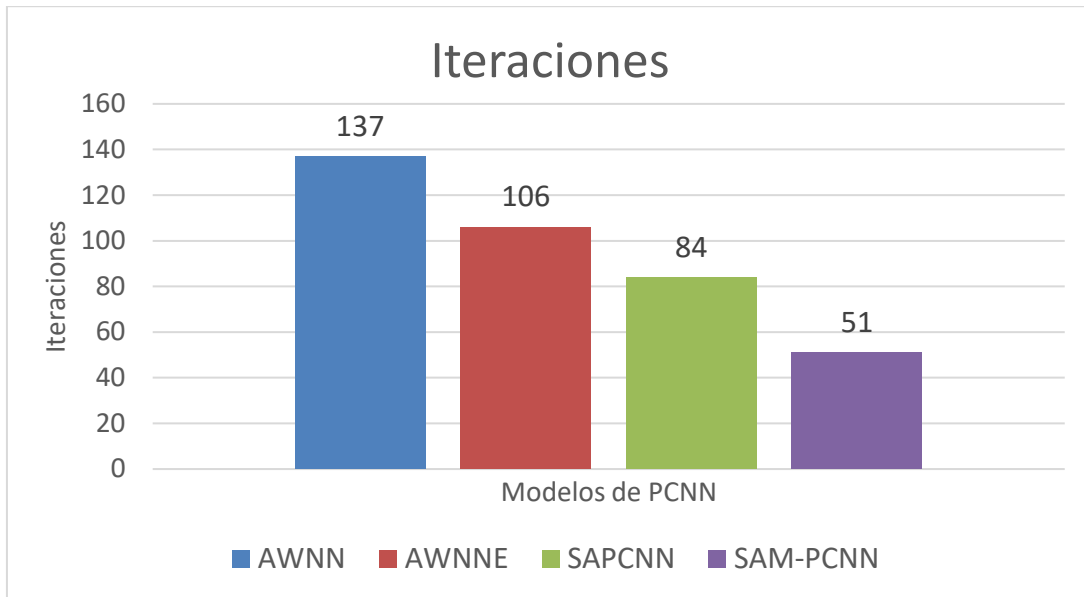


Figura 48. Gráfica comparativa del número de iteraciones.

5.1.2 Experimento 2

Para el segundo experimento se generaron aleatoriamente 4 grafos de 30, 50, 200 y 500 nodos respectivamente. Dichos grafos fueron generados tal como se especifica en [Qu, 2013]. El objetivo del experimento fue buscar la trayectoria más corto entre el nodo 1 y el último nodo de cada grafo. El resultado de las iteraciones de cada algoritmo en cada grafo se muestra en la Figura 49. En la Tabla 6 se presentan los tiempos de procesamiento en segundos obtenidos en cada una de las búsquedas.

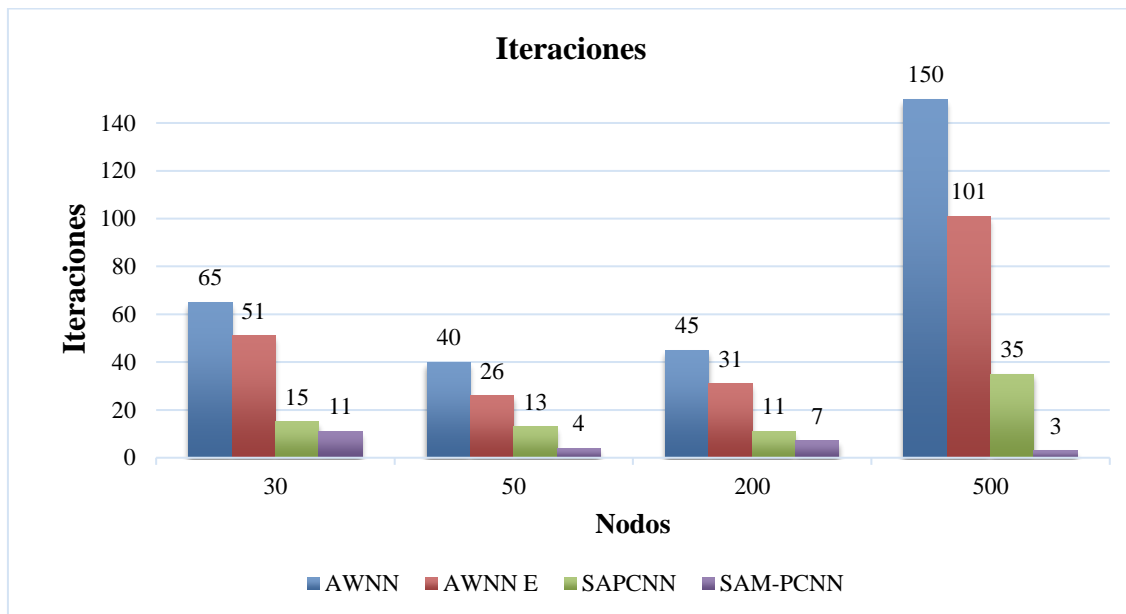


Figura 49. Comparativa de iteraciones del Experimento 2.

Tabla 6. Comparativa de tiempos de procesamiento en segundos del Experimento 2.

Algoritmo	Nodos			
	30	50	200	500
AWNN	0.005 s	0.0028 s	0.272 s	13.833 s
AWNN E	0.006 s	0.002	0.268 s	13.789 s
SAPCNN	0.002 s	0.001 s	0.111 s	4.204 s
SAM-PCNN	0.002 s	0.001 s	0.111 s	1.692 s

Los resultados de la Tabla 6, muestran que, con grafos con pocos nodos, la diferencia en tiempo es apenas perceptible, pero cuando se trata de grafos de mayores dimensiones, la diferencia es considerable. Por otro lado, se puede apreciar que el tiempo de procesamiento del modelo SAM-PCNN tiene un comportamiento semi-lineal con respecto al número de nodos.

5.2 Experimentos de Planeación de Trayectorias (Mapas)

Se presentan los experimentos realizados con los modelos de PCNN diseñados para la Planeación de Trayectorias Discretas de Robots Móviles en espacios de dos dimensiones. Se experimentó con 10 mapas presentados en la Figura 50 los cuales son parte de un conjunto utilizado por *The Department of Cybernetics Intelligent and Mobil Robotics Group, University of Praga* [Vanasek,2009]. Los modelos evaluados en los experimentos son:

- MPCNN (*Modified Pulse-Coupled Neural Network*) [Qu, 2009]
- BSPCNN (*Bidirectional Search Pulse-Coupled Neural Network*)

A continuación, se dará una descripción de los experimentos realizados.

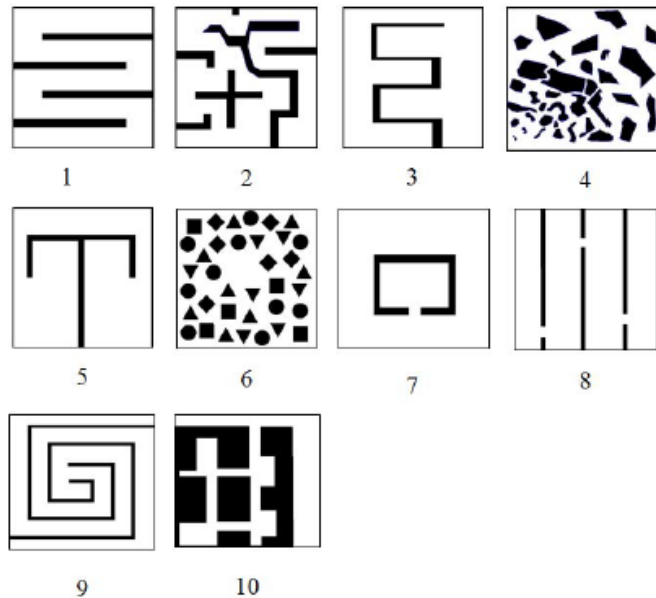


Figura 50. Mapas utilizados para los experimentos [Vanasek,2009].

5.2.1 Experimentos con Mapas de 200×200 pixeles

Se experimentó con 10 mapas de [Vanasek,2009] con una extensión de 200x200 pixeles presentados en la Figura 50. Las rutas que se eligieron para estos experimentos son las rutas que representan el mayor reto en cuanto a complejidad de los obstáculos que hay que evadir o distancia que se debe recorrer.

Se utilizaron los siguientes valores de los parámetros del modelo MPCNN para realizar los experimentos:

- $B = 1, C = 10$
- $A^l = C(1 - e^{-1}) = 6.32120559$
- $A^r = \sqrt{2}C(1 - e^{-1}) = 8.93953467$

Los resultados de estos experimentos se presentan en la Tabla 7, en la Figura 51 se presentan las trayectorias obtenidas en cada búsqueda. Finalmente se presenta una Grafica con los resultados obtenidos en la Figura 52.

Tabla 7. Resultados de la búsqueda en mapas de 200 x 200 nodos.

Mapa	Nodo Inicio	Nodo Final	# Iteraciones	Ejecución (Segundos)
1	199	39,800	80,848	278
2	199	39,800	41,249	138
3	39,924	39,944	53,557	180
4	199	39,800	30,342	106
5	9,293	8903	47,025	161
6	0	39,999	30,605	104
7	20,100	199	27,463	94
8	0	199	62,335	215
9	199	39,800	129,685	448
10	38,009	199	33,416	105

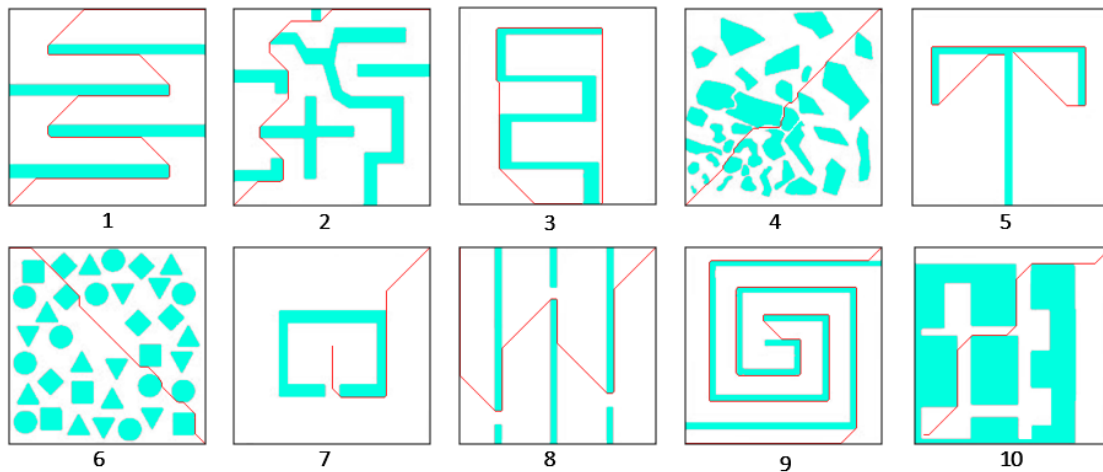


Figura 51. Mapas con las trayectorias obtenidas.

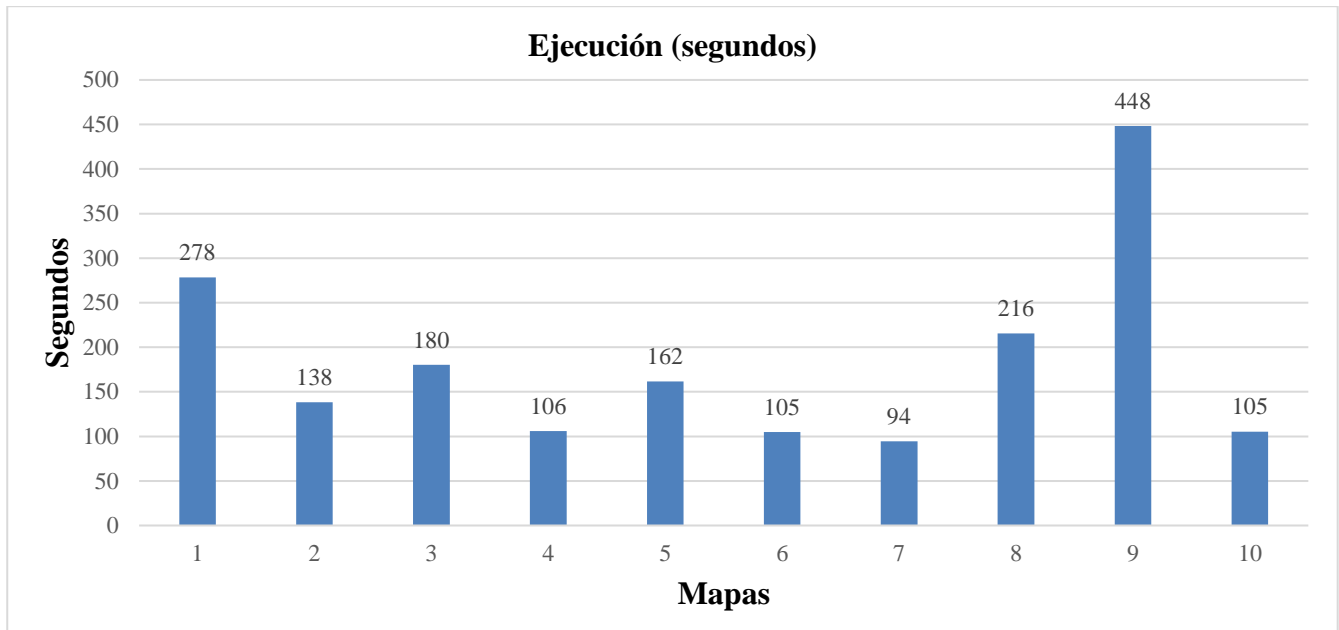


Figura 52. Grafica con los resultados de la búsqueda en mapas de 200 x 200 nodos.

5.2.2 Sintonización de parámetros del modelo MPCNN

Por otro lado, los autores del modelo MPCNN [Qu, 2009] indican que es posible sintonizar los parámetros para obtener una mayor velocidad de propagación de auto-onda. La sensibilidad de un sistema a las variaciones de parámetros es un factor de primordial importancia que debe considerarse al proponer un modelo.

Los parámetros en el modelo son: A^r, A^l, B, C y V . Los autores [Qu, 2009] proponen un rango razonable para estos parámetros el cual es: $0 < A^l < \frac{C}{B}, 0 < A^r < \frac{\sqrt{2}C}{B}$, y $V = (\sqrt{2}C)/B$. En la Figura 53, se muestra el rango propuesto en [Qu, 2009].

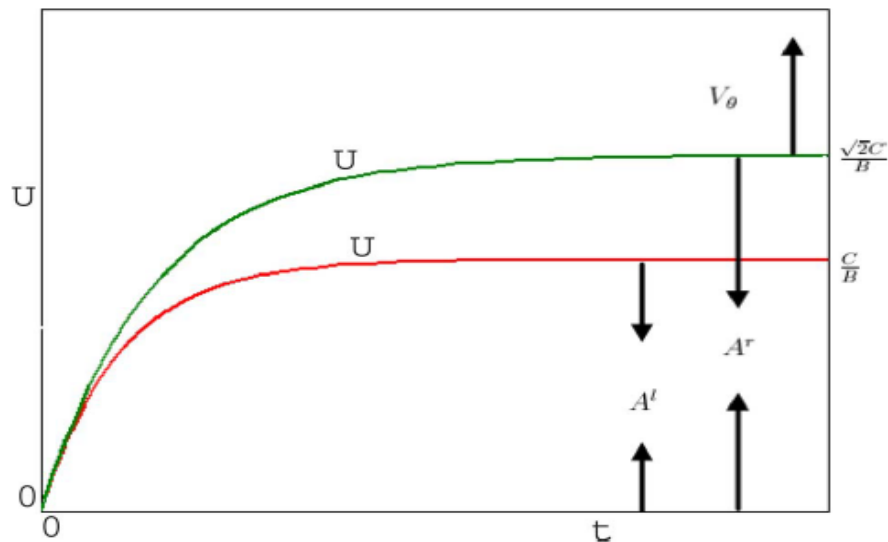


Figura 53. rangos para los parámetros del MPCNN [Qu, 2009].

De la desigualdad restringe los parámetros A , B y C , pero, en general, los valores pequeños de A^l y A^r conducen a velocidades de onda más rápidas. Tomando en cuenta los rangos propuestos en [Qu, 2009] para los parámetros, se utilizaron 7 combinaciones de parámetros, los cuales se muestran en la Tabla 8, además se muestran las iteraciones y tiempo de ejecución obtenidos en los experimentos. Adicionalmente se muestra una Grafica con los datos del experimento en la Figura 54. Se utilizó un mapa con obstáculos 200 x 200 pixeles [Vanasek,2009] donde se buscó la trayectoria óptima entre el nodo superior derecho y el inferior izquierdo.

Tabla 8. Resultados con diferentes combinaciones de parámetros.

Combinación	Parámetros				# Iteraciones	Ejecución (Segundos)
	Al	Ar	B	C		
A	6.32	8.93	1	10	30,342	106
B	1	1.4142	1	10	3,433	13
C	1	1.4142	10	50	1,089	4
D	1	1.4142	10	100	691	2.7
E	1	1.4142	10	200	461	2.3
F	1	1.4142	1	100	691	2.7
G	1	1.4142	50	100	691	2.7

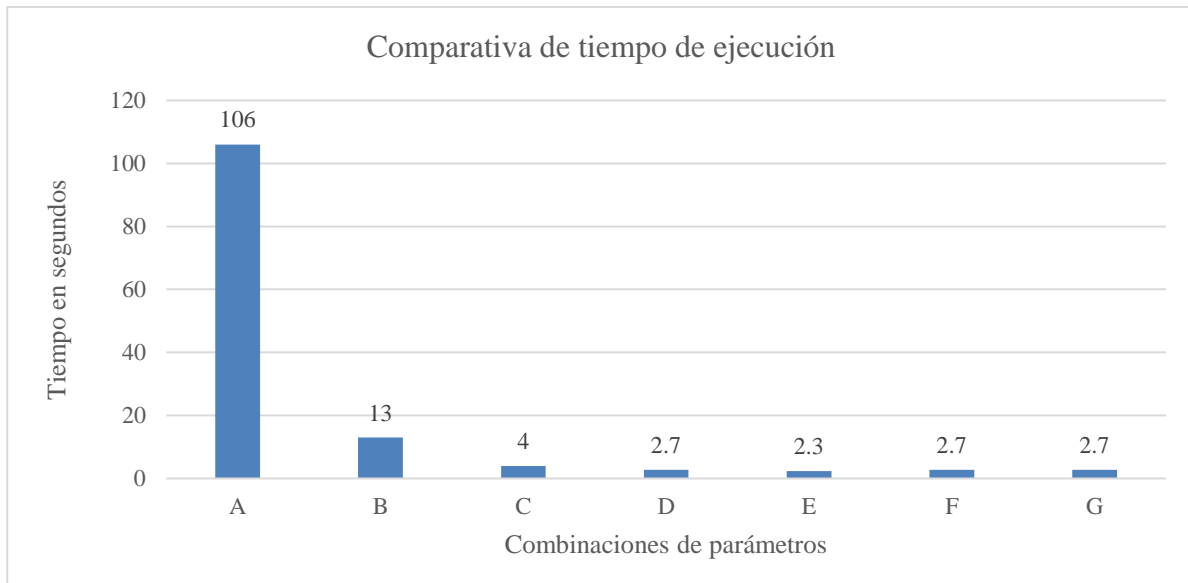


Figura 54. Tiempos de ejecución de cada una de las combinaciones de parámetros.

La combinación A tiene un tiempo de ejecución muy alto, además de que se encuentra la trayectoria en un número muy elevado de iteraciones. La combinación B ya proporciona una baja considerable en el número de iteraciones y el tiempo de ejecución. Para el resto de las combinaciones los tiempos son bastante adecuados para el cálculo de la trayectoria óptima en tiempo real, sin embargo, las combinaciones D , E , F y G generan trayectorias que no corresponden con la trayectoria más corta.

Finalmente, la combinación C presenta los mejores resultados, tanto en el número de iteraciones como en el tiempo de ejecución. Utilizando la combinación de parámetros C se ejecutaron nuevamente

las búsquedas en los diez mapas de [Vanasek,2009]. Los resultados de estos experimentos se muestran en la Tabla 9.

Tabla 9. Resultados con la combinación de parámetros C.

Mapa	Nodo inicio	Nodo final	# Iteraciones	Ejecución (segundos)
1	199	39,800	3,091	12
2	199	39,800	1,544	5.9
3	39,924	39,944	2,088	8.4
4	199	39,800	1,089	4
5	9,293	8,903	1,786	6.7
6	0	39,999	1,104	4
7	20,100	199	1,048	4
8	0	199	2,344	9
9	199	39,800	5,089	18
10	38,009	199	1,263	4

En la Figura 55 se presenta una comparación entre los tiempos de ejecución obtenidos con la combinación A y la combinación C.

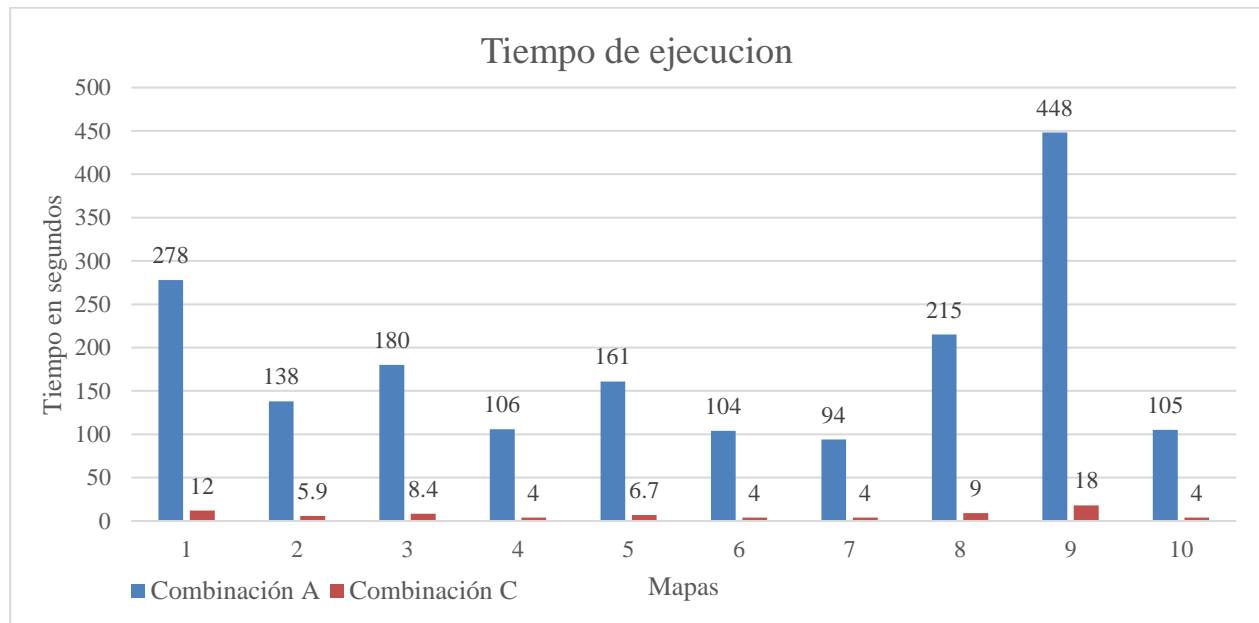


Figura 55. Comparativa entre los resultados de la combinación A y la combinación B.

5.2.3 Tiempos de un agente de búsqueda

Una vez implementado el modelo BSPCNN, se experimentó con los 10 mapas de una extensión de 200x200 pixeles, los cuales son parte de un conjunto utilizado por *The Department of Cybernetics Intelligent and Mobil Robotics Group, University of Praga* [Vanasek,2009]. En la Tabla 10 se muestran los resultados obtenidos por las búsquedas A y B. La búsqueda del nodo de inicio al nodo final será llamada la “búsqueda A”, y la búsqueda del nodo final al nodo de inicio se llamará “búsqueda B”.

Tabla 10. Resultados de las búsquedas A y B.

Mapa	Distancia	Tiempo en segundos		Iteraciones
		Búsqueda A	Búsqueda B	
1	798.1	5.34	5.44	1,546
2	412.02	2.52	2.55	785
3	528.81	3.27	3.29	1,103
4	321.33	1.95	1.92	601
5	464.14	3.06	3.09	893
6	305.83	2.03	1.92	602
7	270.78	2.49	2.30	729
8	616.04	4.08	3.98	1,215
9	1,281.51	8.28	8.68	2,545
10	329.62	1.97	2.10	632

Teniendo en cuenta que el tiempo de ejecución es diferente para cada una de las búsquedas por que el espacio de búsqueda no es el mismo en cada iteración. Se toma el tiempo más alto obtenido por alguna de las dos búsquedas como el tiempo de ejecución del algoritmo. En la Tabla 11 se presenta la comparativa de los resultados obtenidos por el modelo MPCNN y los resultados obtenidos por el modelo BSPCNN en las búsquedas con los mapas de 200x200 pixeles.

Tabla 11. Comparativa entre MPCNN y BSPCNN con sólo un agente de busqueda.

Mapa	Distancias		Tiempo en segundos		Iteraciones	
	MPCNN	BSPCNN	MPCNN	BSPCNN	MPCNN	BSPCNN
1	798.1	798.1	12	5.44	3,091	1,546
2	407.26	412.02	5.9	2.55	1,544	785
3	528.81	528.81	8.4	3.29	2,088	1,103
4	299.58	302.31	4	1.95	1,089	601
5	464.14	464.14	6.7	3.09	1,786	893
6	302.51	305.83	4	2.03	1,104	602
7	270.78	270.78	4	2.49	1,048	729
8	615.8	616.04	9	4.08	2,344	1,215
9	1,281.5	1,281.5	18	8.68	5,089	2,545
10	329.62	329.62	4	2.10	1,263	632

Para analizar con detenimiento los datos de la Tabla 11 se generaron gráficas donde se compara con mayor claridad las distancias, los tiempos de ejecución y el número de iteraciones. En la Figura 56 se presenta una gráfica donde se comparan las distancias obtenidas por los algoritmos. En la Figura 57 se

muestra la gráfica con la comparación de los tiempos de ejecución de cada algoritmo. Finalmente, en la Figura 58 se presenta una gráfica con los datos de las iteraciones en cada experimento.

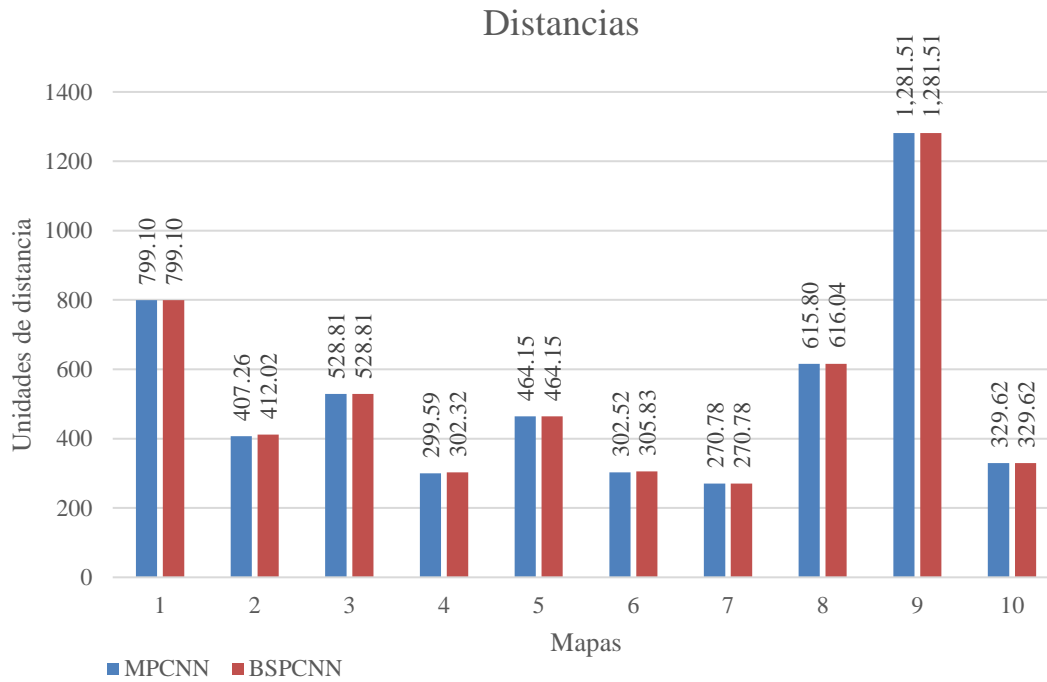


Figura 56. Comparación de distancias obtenidas por los algoritmos.

De acuerdo con los resultados el modelo propuesto BSPCNN logró calcular la trayectoria más corta en un 60% de los experimentos, en el resto de los experimentos se obtuvo una trayectoria sub-óptima.

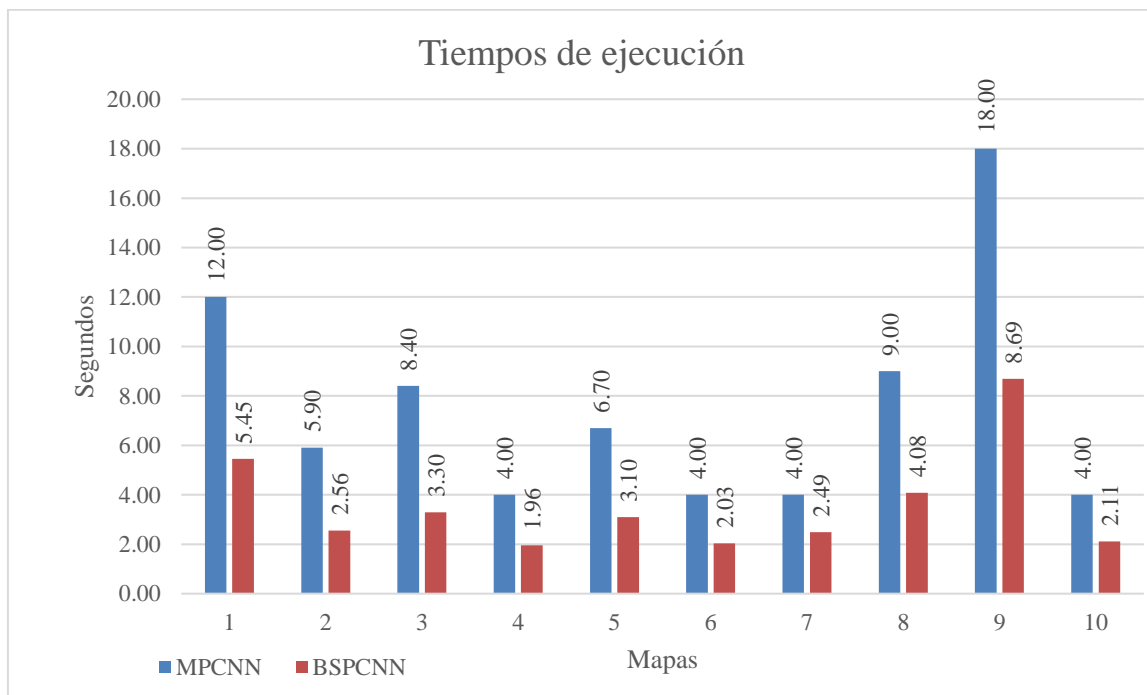


Figura 57. Comparación de los tiempos de ejecución de cada algoritmo.

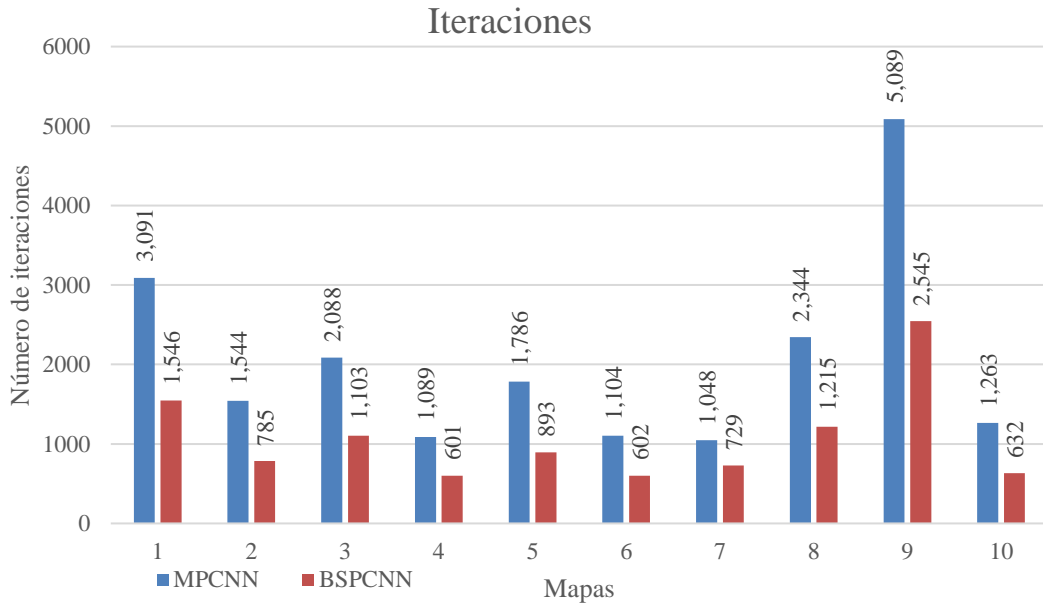


Figura 58. Las iteraciones en cada experimento.

En la Figura 57 se puede apreciar que los tiempos de ejecución obtenidos por el algoritmo propuesto BSPCNN fueron menores a los conseguidos por el algoritmo MCPNN [Qu, 2009]. El modelo BSPCNN fue en promedio un 50% más rápido en estos experimentos, superando en el 100% de los casos al modelo MPCNN [Qu, 2009]. El número de iteraciones del modelo BSPCNN es aproximadamente la mitad que el número de iteraciones del modelo MPCNN [Qu, 2009] como se puede apreciar en la Figura 58. Esto se debe a que una iteración cuenta para ambas auto-ondas, por lo que se encuentran en la mitad iteraciones comparado con una sola auto-onda recorriendo el camino desde el punto de inicio hasta el punto final.

Posteriormente se utilizaron los mismos mapas de [Vanasek,2009] con una dimensión de 500x500 pixeles para comparar los resultados del modelo propuesto BSPCNN y del modelo MPCNN [Qu, 2009]. la información obtenida en las búsquedas se presenta en la Tabla 12.

Tabla 12. Resultados con mapas de 250 mil pixeles.

Mapa	Distancias		Tiempo en segundos		Iteraciones	
	MPCNN	BSPCNN	MPCNN	BSPCNN	MPCNN	BSPCNN
1	1,966.63	2000.36	187.49	84.50	7,616	3,873
2	1,013.81	1016.23	95.66	41.74	3,852	1,933
3	1,330.32	1330.32	112.63	54.24	5,252	2,626
4	746.1	746.1	62.65	28.57	2,703	1,352
5	1,185.25	1185.25	99.47	45.89	4,564	2,282
6	754.9	804.86	95.44	43.64	2,748	1,505
7	689.08	689.08	58.77	26.37	2,670	1,337
8	1,536.46	1572.85	221.87	107.90	5,848	3,018
9	3,199.68	3199.68	519.75	134.01	12,717	6,359
10	826.78	826.78	74.17	34.66	3,165	1,583

De acuerdo con los datos generados por los experimentos con los mapas de 500x500, el mapa 2 fue donde la distancia obtenida por el modelo BSPCNN es un 0.5% más grande que la distancia obtenida por el MPCNN. En la Figura 59 se pueden observar las trayectorias obtenidas por los algoritmos de PCNN.

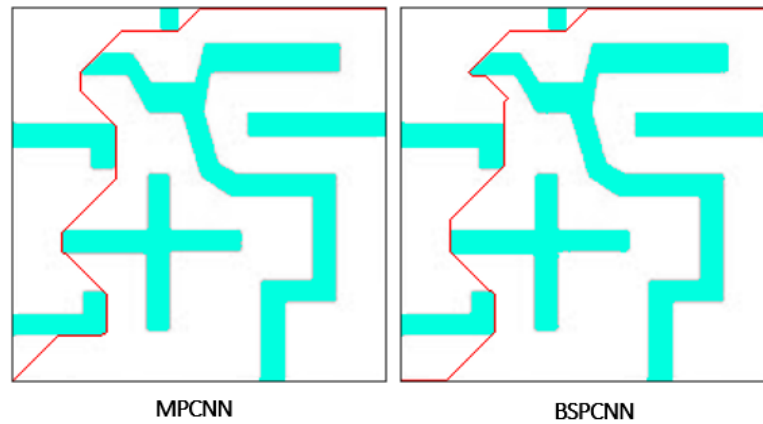


Figura 59. Trayectorias obtenidas por las PCNN en el mapa 2.

Posteriormente se experimentó con diferentes magnitudes del mapa 2 para comparar los resultados obtenidos en mapas de 100x100, 200x200, 300x300, 400x400, 500x500, 600x600 y 700x700. En la Tabla 13 se presentan los resultados obtenidos en dichos experimentos.

Tabla 13. Comparativa con mapas de distintas magnitudes.

# Nodos	Mapa	Distancias		Tiempo en segundos		Iteraciones	
		MPCNN	BSPCNN	MPCNN	BSPCNN	MPCNN	BSPCNN
10,000	100x100	207.44	215.02	0.67	0.36	792	414
40,000	200x200	407.26	412.02	5.46	2.55	1,544	785
90,000	300x300	624.17	631.27	18.83	8.88	2,388	1,213
160,000	400x400	831.71	837.64	45.18	20.77	3,184	1,608
250,000	500x500	1,013.81	1,016.23	95.66	41.74	3,852	1,933
360,000	600x600	1,246.44	1,254.37	149.95	68.79	4,772	2,406
490,000	700x700	1,453.63	1,459.57	239.94	115.82	5,564	2,798

Se puede ver que en todos los experimentos las distancias obtenidas por el BSPCNN son mayores en un 0.5% al modelo MPCNN, debido a que el nodo donde se encuentran las auto-ondas puede ser un nodo que no pertenezca a la trayectoria óptima, pero se encuentra cerca de los nodos que componen la trayectoria óptima. Por otro lado, el tiempo de procesamiento requerido para cada una de las pruebas sugiere una complejidad casi lineal. En la Figura 60 se aprecia una gráfica donde se compara el tiempo de ejecución de cada modelo.

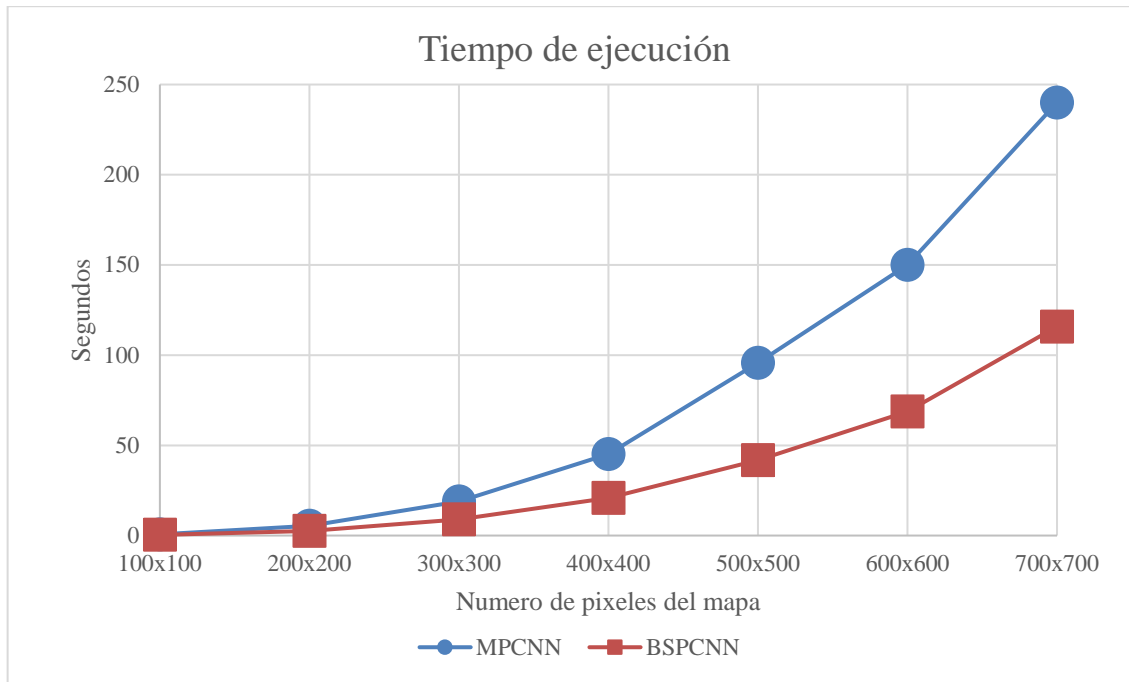


Figura 60. Comparativa de los tiempos de ejecución con diferentes resoluciones del mapa 2.

5.3 Discusión

Se implementaron dos modelos de PCNN orientados para la Optimización de Trayectorias presentados en la literatura. Adicionalmente, se propuso el modelo SAM-PCNN el cual tiene una velocidad de propagación dinámica lo cual le permite mayor eficiencia en el número de iteraciones que requiere para calcular una ruta. Del Experimento 1: el modelo SAM-PCNN es hasta 39.28% más eficiente que SAPCNN y hasta un 62.77% más eficiente que AWNN.

Por otro lado, se implementaron dos modelos de PCNN diseñados para específicamente para la Planeación de Trayectorias. Se utilizó El Modelo MPCNN presentado en [Hong, 2009], para realizar experimentos de búsqueda de trayectorias de manera exitosa, sin embargo, se utilizaron diferentes combinaciones de parámetros con la finalidad de encontrar una combinación que permitiera resultados confiables en poco tiempo de ejecución.

Adicionalmente, se propone un modelo BSPCNN, el cual realiza dos búsquedas de manera simultánea. Durante los experimentos se realizó una búsqueda desde un punto a otro punto dentro de un grafo inconexo y la búsqueda no se pudo completar, sin embargo, es posible aplicar el algoritmo Floyd-Warshall [Cormen,1990] para determinar si existe una ruta entre un punto y otro. Los resultados obtenidos del modelo BSPCNN muestran una mejora de 50% menos iteraciones con respecto de los resultados MPCNN en el 100% de los casos. Tomando en cuenta el tiempo más alto de las búsquedas A y B se obtuvieron mejoras en el tiempo de ejecución en el 100% de los casos. Las trayectorias obtenidas son óptimas en el 50% de los casos.

Capítulo 6. Conclusiones

En este Capítulo final se muestra el logro de los objetivos y alcances mediante actividades y productos específicos, los productos publicados y enviados, las aportaciones y las principales conclusiones del trabajo realizado.

6.1 Objetivos logrados

6.1.1 Objetivo general

Implementar, experimentar y evaluar la RNA tipo Pulso-Acoplada en el dominio de la Planeación de Trayectorias discretas para robots móviles.

El objetivo general se cumplió satisfactoriamente, se implementaron seis modelos de PCNN, cuatro modelos descritos en la literatura, y se propusieron dos nuevas variantes de PCNN. El algoritmo propuesto en el dominio de la Optimización de Trayectorias sobre grafos, se logra una eficiencia en un 39% sobre el modelo que menos iteraciones realiza para obtener una trayectoria. En el campo de la Planeación de Trayectorias sobre mapas, se presenta un modelo que puede calcular una trayectoria en 50% menos iteraciones y en aproximadamente un 50% menos de tiempo de procesamiento.

6.1.2 Objetivos específicos

De acuerdo con el objetivo general y con los objetivos específicos, se llevaron a cabo tareas propuestas para el cumplimiento de cada uno de ellos, como se describe en la Tabla 14.

Tabla 14. Objetivos específicos de la tesis

Objetivos específicos	Actividades
Estudiar los métodos convencionales empleados en el campo de la planeación de trayectorias para robots móviles.	Se revisaron 39 documentos sobre el tema donde fueron estudiados diversos métodos empleados en la planeación de trayectorias.
Estudiar el área de las RNAs tradicionales.	Se cursó la materia de Redes Neuronales Artificiales donde se estudiaron e implementaron modelos de RNA como el Perceptron, Redes <i>Hopfield</i> , Redes de <i>Hamming</i> , <i>Multilayer Perceptrón Back Propagation</i> , Red <i>BAM</i> , <i>Kohonen</i> , las Redes Neuronales de Tercera Generación como PCNN, el modelo ICM e <i>Izhikevich</i> .
Revisar y entender los problemas típicos de las técnicas tradicionales y de las RNAs en el dominio de la Planeación de Trayectorias.	En el campo de la Planeación de Trayectorias, los algoritmos de búsqueda tradicionales pueden llegar a tener tiempos largos de procesamiento, problemas de mínimos locales y en algunos casos no poder encontrar una ruta existente.

Implementar la RNA Pulsante aplicada al problema de la planificación de trayectorias.	Se implementaron 6 modelos de RNA Pulsante aplicados al problema de la planificación de trayectorias.
Experimentar y evaluar con bases de datos de uso común en la comunidad de Inteligencia artificial y de robótica.	Se experimentó con distintos mapas y grafos utilizados en investigaciones sobre trayectorias, además, se utilizaron 10 mapas los cuales son parte de un conjunto utilizado por <i>The Department of Cybernetics Intelligent and Mobil Robotics Group, University of Praga</i> .
Comparar el algoritmo pulsante con respecto a otros algoritmos previos.	Se realizó una comparación entre el algoritmo <i>Branch and Bound</i> y los modelos de PCNN implementados.
Realizar un estudio comparativo con por lo menos 3 variantes de la RNA de tipo Pulso-Acoplada.	Se elaboró un estudio comparativo entre los modelos AWNN [Ma, 2011], AWNN Explicitada [Paredes-Cano, 2018], SAPCNN [Li, 2013] y SAM-PCNN en el campo de la Optimización de Trayectorias. Además, se compararon los modelos MPCNN [Qu, 2009] y BSPCNN en el campo de la Planeación de Trayectorias.

6.1.3 Alcances y limitaciones

En la Tabla 15 se presentan los alcances propuestos para esta tesis y cómo fueron cubiertos por cada una de las actividades realizadas.

Tabla 15. Alcances y limitaciones de la tesis.

Alcances	Actividades
Experimentar con casos de problemas planteados en la literatura.	Se experimentó con 10 casos planteados en [Li, 2013] y 3 casos planteados en [Qu, 2009].
Realizar el estudio comparativo entre las RNAPs y las técnicas tradicionales basado en el tiempo de procesamiento y resultados presentados en la literatura de optimización de trayectorias.	Se presentó en el Congreso Mexicano de Inteligencia Artificial un artículo [Espinosa, 2019] donde se describe un estudio comparativo entre el modelo propuesto SAM-PCNN y AWNN [Ma, 2011], AWNN Explicitado [Paredes-Cano, 2018], SAPCNN [Li, 2013].

6.2 Productos y actividades extra

6.2.1 Participación como ponente en Escuela de Inteligencia Artificial y Robótica 2018 (octubre - 2018)

Se realizó la presentación de un poster en la Escuela de Inteligencia Artificial y Robótica 2018, que se llevó a cabo los días 25 y 26 de octubre de 2018, en la ciudad de Cuernavaca, Morelos; México. La constancia es mostrada en el Anexo A.

6.2.2 Presentación de poster 1ra Jornada de Ciencia y Tecnología Aplicada (noviembre - 2018)

Se realizó la presentación de un poster en la 1ra Jornada de Ciencia y Tecnología Aplicada, que se llevó a cabo los días 15 y 16 de noviembre de 2018, en la ciudad de Cuernavaca, Morelos; México. La constancia es mostrada en el Anexo B [Espinosa, 2018].

6.2.3 Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2018)

Se realizó la presentación de un poster en el ICMEAE (*International Conference on Mechatronics, Electronics and Automotive Engineering*) 2018. El reconocimiento se muestra en el Anexo C.

6.2.4 Presentación de poster 2da Jornada de Ciencia y Tecnología Aplicada (abril - 2018)

Se realizó la presentación de un poster en la 2da Jornada de Ciencia y Tecnología Aplicada, que se llevó a cabo los días 4 y 5 de abril de 2019, en la ciudad de Cuernavaca, Morelos; México. La constancia es mostrada en el Anexo D.

6.2.5 Presentación para el 1er coloquio “Artificial Intelligence” (mayo - 2019)

Se participó como ponente con la conferencia llamada: “Redes Neuronales Pulsantes para la planeación de Trayectorias de Robots Móviles”, realizada en el Instituto Tecnológico de Iguala. El reconocimiento se muestra en Anexo E.

6.2.6 Participación como ponente en el XI Congreso Mexicano de Inteligencia Artificial (junio -2019)

Se presentó el artículo “Variante Eficiente de Red Neuronal Pulsante con Auto-Onda Dinámica para Resolver el Problema del Camino más Corto” en el XI Congreso Mexicano de Inteligencia Artificial llevado a cabo del 4 al 7 de junio del 2019. El reconocimiento se muestra en el Anexo F. [Espinosa, 2019]

6.2.7 Participación como ponente en la 1er Jornada de Tecnologías de la información 2019 (junio -2019)

Se participó como ponente con la conferencia llamada: “Problema del Camino más Corto resuelto con una Red Neuronal de Tercera Generación”, realizada en la Universidad tecnológica del Sur del Estado de Morelos. El reconocimiento se muestra en Anexo G.

6.2.8 Participación como ponente en 3ra Jornada de Ciencia y Tecnología Aplicada (noviembre -2019)

Se presentó el artículo “El problema del Camino más Corto Resuelto con una Alternativa Eficiente Basada en Redes Neuronales” en la 3da Jornada de Ciencia y Tecnología Aplicada, que se llevó a cabo los días 14 y 15 de noviembre de 2019, en la ciudad de Cuernavaca, Morelos; México. La constancia es mostrada en el Anexo H.

6.2.9 Participación como ponente en Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2019)

Se presentó el artículo “Spiking Neural Network to Solve the Shortest Path NP Problem” en el ICMEAE (*International Conference on Mechatronics, Electronics and Automotive Engineering*) 2019, en la ciudad de Cuernavaca, Morelos; México. La constancia es mostrada en el Anexo I.

6.2.10 Publicación de Artículo en la revista “*Research in Computing Science*” (noviembre - 2019)

Se publicó el artículo “Variante Eficiente de Red Neuronal Pulsante con Auto-Onda Dinámica para Resolver el Problema del Camino más Corto” en la revista “*Research in Computing Science*” en el volumen 148(8): Advances in Intelligent Systems. La carta de aceptación se muestra en el Anexo J. [Espinosa, 2019]

6.2.11 Estancia en Instituto Nacional de Electricidad y Energías Limpias

A través del uso de la tecnología de Redes Neuronales Pulso Acopladas, se diseñó e implementó un sistema que permite optimizar la construcción de un modelo de negocio mediante la metodología Canvas, de tal forma que se minimizan los costos y se maximizan las ganancias.

6.3 Aportaciones

- Se propusieron dos nuevas variantes de Red Neuronal Pulso Acoplada. La variante SAM-PCNN se diseñó para resolver el Problema del Camino más Corto dentro de grafos ponderados no dirigidos. Los resultados de los experimentos donde se comparó su rendimiento frente a otros algoritmos, muestran que tiene la capacidad de calcular el Camino más Corto en un 39% menos iteraciones y con 61% menos tiempo de procesamiento en grafos donde existe una distancia considerable entre cada nodo.
- Adicionalmente se propuso la variante BSPCNN, diseñada para la Planeación de Trayectorias en Mapas bidimensionales discretizados. BSPCNN realiza una búsqueda bidireccional produciendo dos auto-ondas, la primera comienza desde la neurona de inicio y la segunda comienza desde la neurona objetivo.
- Los resultados obtenidos en los experimentos indican que la variante propuesta le toma un 50% menos de tiempo de procesamiento encontrar una Trayectoria óptima. Además, requiere de 50% menos iteraciones que el modelo MPCNN [Qu, 2012].
- Se encontró la trayectoria óptima en la mayoría de las búsquedas, en algunos casos se encontró una trayectoria sub-óptima, sin embargo, la distancia de la trayectoria encontrada no superaba el 0.5% de diferencia con respecto de la óptima.
- En resumen, las variantes de PCNN propuestas en esta tesis, son una aportación novedosa en el campo de las Redes Neuronales Pulso Acopladas y en el campo de la Planeación y Optimización de Trayectorias.
- Se presentó un análisis de diferentes variantes de PCNN encontradas en la literatura, y se realizó un estudio que comparó el rendimiento de cada una de las variantes presentadas, lo que aporta al conocimiento producido en campo de las Redes Neuronales Pulso Acopladas dentro de la comunidad científica.

6.4 Conclusiones

- Se implementaron 3 modelos pulsantes orientados para la Optimización de Trayectorias sobre grafos presentados en la literatura.
- Se propuso SAM-PCNN con incrementos dinámicos.
- De acuerdo con los resultados obtenidos el modelo propuesto SAM-PCNN presenta mayor eficiencia al calcular el Camino más Corto en grafos donde la distancia entre los nodos es mayor a uno. Además, se mostró que el modelo propuesto es una buena alternativa en el campo de la Optimización de Trayectoria.
- En el Experimento 1, el modelo SAM-PCNN es hasta 39.28% más eficiente que SAPCNN [Li, 2013] y hasta un 62.77% más eficiente que AWNN [Ma, 2011].
- Se implementaron 2 modelos de PCNN orientados para la Planeación de Trayectorias, el modelo MPCNN [Qu, 2009] encontrado en el estado del arte, y el un nuevo modelo llamado BSPCNN.

- De acuerdo con los experimentos realizados en la investigación, el número de interacciones realizadas por el modelo BSPCNN es en promedio un 50% menor que las del modelo MPCNN [Qu, 2009].
- El modelo BSPCNN encontró la trayectoria óptima en la mayoría de las búsquedas, en algunos casos se encontró una trayectoria sub-óptima, sin embargo, la distancia de la trayectoria encontrada no superaba el 0.5% de diferencia con respecto de la óptima.
- Para el modelo BSPCNN se tienen dos escenarios:
 - a. Tomando en cuenta la suma del tiempo de ejecución de las dos búsquedas realizadas por las Redes del modelo BSPCNN, se puede afirmar que:
 - En el mejor de los casos el tiempo mejora hasta un 12.44%, mientras que en el peor de los casos se registró una mejora del 1.87%.
 - El modelo BSPCNN fue en promedio un 7.39% más rápido, superando en el 100% de los casos al modelo MPCNN [Qu, 2009].
 - b. Tomando en cuenta la búsqueda de la red del modelo BSPCNN con el tiempo de ejecución más largo, se puede decir que:
 - El tiempo de ejecución del modelo BSPCNN es en promedio un 50% más rápido que el del modelo MPCNN [Qu, 2009].
- En esta investigación los algoritmos fueron implementados de manera secuencial, por lo que los resultados presentados pueden mejorar considerablemente con una implementación paralela.
- El tiempo de procesamiento requerido para cada una de las pruebas sugiere una complejidad menor a cuadrática tanto en el modelo MPCNN como el BSPCNN.

6.5 Trabajo futuro

- Como trabajo futuro se puede pensar en realizar algunas modificaciones al modelo BSPCNN para que funcione con una red con dos auto-ondas en lugar de dos redes superpuestas con una auto-onda cada una.
- Además, se podría adecuar el modelo BSPCNN para trabajar con Planeación de Trayectorias en espacios de tres dimensiones.
- Otro objetivo que se puede alcanzar es proponer un modelo de PCNN que realice una búsqueda dirigida similar al modelo GAPCNN [Usman, 2014] pero sin la necesidad de ocupar las heurísticas que dicho modelo utiliza, de tal manera que no sea necesario conocer todas las trayectorias a priori.
- También se puede proponer un algoritmo que pueda verificar si existe una trayectoria entre un punto y otro en grafos disjuntos.
- Por otro lado, es importante mencionar que aún falta probar con un robot real los algoritmos propuestos en esta investigación.
- Finalmente, se debe aprovechar la naturaleza paralela de los algoritmos de PCNN implementando los algoritmos propuestos utilizando un paradigma de programación paralelo.

Referencias

- [Aguilar, 2019] Aguilar Domínguez K. S. (2019). Mejoramiento de la Luminosidad de Imágenes Digitales del Cerebro Humano mediante Redes Neuronales Pulso-Acopladas . Tesis de Maestría en Ciencias de la Computación, CENIDET.
- [Arista, 2012] Arista R. V. (2012). Implementación Configurable y Multipropósito de Redes neuronales de tercera Generación en GPUs. de *Memorias del primer concurso de Investigación, desarrollo e innovación tecnológic.*.
- [Arndt, 1990] Arndt, M., Dicke, P., Eckhorn, R., & Reitboeck, H. J. (1990). Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex. *Neural Computation*, 2, pp. 293–307.
- [Barraquand, 1992] Barraquand J., Langlois B., Latombe J. C. (1992). Numerical potential field techniques for robot path planning, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, n° 2, pp. 224-241,.
- [Carballo, 2017] Carballo E. S., Morales L., Trujillo-Ro F. (2017). Path planning for a mobile robot using genetic algorithm and artificial bee colony. *International Conference on Mechatronics, Electronics and Automotive Engineering*. pp. 8-12.
- [Cárdenas, 2015] Cárdenas S. Y. (2015). Implementación y Evaluación de Redes Neuronales Artificiales tipo “Pulse-Coupled Neural Networks” (PCNN) Aplicadas a Visión Artificial. Tesis de Maestría en Ciencias de la Computación, CENIDET.
- [Caulfield, 1999] Caulfield H. J., Kinser: Finding M. (1999). The shortest path in the shortest time using PCNNs, *IEEE Trans. Neural Netw*, vol. 10, número 3, pp. 604–606.
- [Cherkassky, 1996] Cherkassky B. V., Goldberg A. V., Radzik T. (1996). Shortest paths algorithms: theory and experimental evaluation, *Math. Programming* 73 (2) pp. 129–174.
- [Córtez, 2018] Córtes Martínez K. V. (2018). Red Neural Pulsante para el Reconocimiento de Imágenes de Cerebros Sanos y No-Sanos. Tesis de Maestría en Ciencias de la Computación, CENIDET.
- [Dijkstra, 1959] Dijkstra E. W. (1959). A note on two problems in connection with graphs, *Numer. Math.* 1(1) pp. 269–271.
- [Eckhorn, 1990] Eckhorn R., H. Reitboeck J., Arndt M., Dicke P. W. (1990). Feature linking via synchronization among distributed assemblies: Simulation of results from cat cortex.,» *Neural Comput*, vol. 2, n° 3.
- [Escamilla, 2013] A. C. Escamilla. (2013). Planeación de Trayectorias y Control por Platitude Diferencial de Múltiples Robots Móviles. Tesis de Maestría en Ingeniería en Electronica, Cenidet.
- [Espinosa, 2018] Espinosa O., Mejía-Lavalle M., (2018). Jornada de Ciencia y Tecnología Aplicada Tecnológico Nacional de México/CENIDET Vol. 1. pp. 64.
- [Espinosa, 2019] Espinosa O., Mejía-Lavalle M., Ruiz J., Reyes G., Martínez A. (2019). Variante eficiente de red neuronal pulsante con auto-onda dinámica para resolver el problema del camino más corto. *Research in Computing Science Issue* 148(8) pp. 144–166.
- [FitzHugh, 1961] FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysics Journal*, 1, pp. 445–466.
- [Qu, 2009] Hong Q., Yang S. X., Willms A., Yi Z.. (2009). Real-Time Robot Path Planning Based on a Modified, *IEEE Transactions on Neural Networks*, vol. 20, n° 11, pp. 1724-1739.


- [Harris, 2015] Harris M.A., Van A.N., Malik B.H., Jabbour J.M, Maitland K.C. (2015). A pulse coupled neural network segmentation algorithm for reflectance confocal images of epithelial tissue. *Plos One*, vol. 10, issue 3, p e0122368, ISSN 19326203. ISBN 19326203
- [Hodgkin, 1952] Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117, 500–544.
- [Jio, 2019] Jiao K., Pan Z. (2019). *A Novel Method for Image Segmentation Based on Simplified Pulse Coupled Neural Network and Gbest Led Gravitational Search Algorithm*. *IEEE Access*, 7, pp. 21310–21330. doi:10.1109/access.2019.2894301
- [Johnson, 1993] Johnson, J. L., & Ritter, D. (1993). Observation of periodic waves in a pulse-coupled neural network. *Optical Letter*, 18, 1253.
- [Khouil, 2016] Khouil M., Sanou I., Mestari M., Aitelmahjoub A. (2016). Planification of an Optimal Path for a Mobile Robot Using Neural Networks, *Applied Mathematical Sciences*, Vol. 10, no. 13, pp. 637-652, doi:10.12988/ams.2016.510653
- [Kroumov, 2011] Kroumov V. Yu J. (2011), *Neural Networks Based Path Planning and Navigation of Mobile Robots*, Rijeka, Croatia: InTech Europe.
- [Latombe, 1991] Latombe J.-C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, Norwell.
- [Lei, 2019] Lei L., Xi F., Chen S. (2019). *Finger-Vein Image Enhancement Based on Pulse Coupled Neural Network*. *IEEE Access*, 7, 57226–57237. Doi: 10.1109/ access.2019.2914229
- [Li, 2013] Li X., Ma Y., Feng X. (2013). Self-adaptive autowave pulse-coupled neuralnet work for shortest-path problem. *Neurocomputing*. vol. 115. pp. 63–71
- [Lindblad, 2013] Lindblad T., Kinser J. (2013). *Image Processing using Pulse-Coupled Neural Networks*, Springer-Verlag Berlin Heidelberg.
- [Lopéz, 2017] Lopéz Borreguero D. (2017). Evaluación de técnicas SLAM disponibles en ROS. Tesis de Maestría en Ciencias de la Computacion, CENIDET.
- [Lozano-Pérez, 1990] Lozano-Pérez T. (1990). Foreword: Mobile Robot and Robotics. *Autonomous Robot Vehicles*, Editores I.J. Cox y G.T. Wilfong. Springer-Verlag.
- [Ma, 2011] Ma, Y., Zhan, K., Wang, Z. (2011), *Pulse-Coupled Neural Networks. Applications of Pulse-Coupled Neural Networks*, pp. 447-465. Springer Heidelberg Dordrecht London New York.
- [Martínez, 2012] Martínez Ramírez F. (2012). Control embebido de un vehículo guiado automáticamente mediante Redes Neuronales Artificiales. Tesis de Maestría, Departamento de Electrónica, CENIDET.
- [Mathivet. 2018] Mathivet V. (2018). Capítulo 3 Búsqueda de rutas, de *Inteligencia Artificial para desarrolladores - Conceptos e implementación en C# (2e edición)*, Ediciones ENI , pp. 162-225.
- [Morrison, 2016] Morrison D. R., Jacobson S. H., Sauppe J. J., & Sewell E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19, pp. 79–102.
- [Muralidharan, 2019] Muralidharan A., Mostofi Y. (2019). Path Planning for Minimizing the Expected Cost Until Success. *IEEE TRANSACTIONS ON ROBOTICS*. pp. 1–16. doi:10.1109/tro.2018.2883829

- [Qu, 2007] Qu H., Yi Z. (2007). A new algorithm for finding the shortest paths using PCNNs. *Chaos, Solitons & Fractals*, vol 33 n.4, pp. 1220–1229. doi: 10.1016/j.chaos.2006.01.097
- [Qu, 2009] Qu H., Yang S. X., Willms A., Yi Z. (2009). Real-Time Robot Path Planning Based on a Modified, *IEEE transactions on neural networks*, vol. 20, n° 11, pp. 1724-1739.
- [Ortigoza, 2012] Ortigoza R. S. *et al.* (2012). Wheeled Mobile Robots: A review. in *IEEE Latin America Transactions*. vol. 10, no. 6, pp. 2209-2217. doi: 10.1109 / TLA. 2012. 6418124
- [Ortiz, 2016] Ortiz E., Mejía-Lavalle M., Sossa H. (2016). Uso de Redes neuronales pulsantes para mejorar el filtrado de imágenes contaminadas con ruido Gaussiano, *Research in Computing Science*, Vol 114, pp. 45–58.
- [Ortiz, 2017] Ortiz E. (2017). Detección de Ruido Impulsivo o Gaussiano en Imágenes Monocromáticas mediante Redes Neuronales Pulso-Acopladas, Tesis de Maestría en Ciencias de la Computación.
- [Paredes-Cano, 2018] Paredes-Cano J. J., Mejía-Lavalle M., Mújica-Va D. y Reyes-Salgado G. (2018). Red Neuronal Pulsante Explicitada para el Problema del Camino más Corto. *XIV Congreso Internacional sobre Innovación y Desarrollo Tecnológico*.
- [Rios, 2008] Rios L. H., Buno M., Sanchez S. (2008). Generación de Trayectorias para un Robot Móvil empleando Redes. *Scientia et Technica*, vol. XIV, n° 39, pp. 94-99.
- [Rojas, 2018] Rojas Fernández A. M. (2018). Evaluación del algoritmo Theta* para planeación de trayectorias. Tesis de Maestría en Ciencias de la Computación, CENIDET.
- [Rybak, 1992] Rybak, I. A., Sandler, V. A., & Shevtsova, N. A. (1992). The model of a neural network visual processor. *Neurocomputing*, 4, pp. 93–102.
- [Silva, 2007] Silva O. R., García S. J. R., Barrientos S. V. R., Molina V. M. A., V. M. y Silva O. G. 2007. Una panorámica de los robots móviles,» *Revista Telematique*, vol. 6, n° 3.
- [Sabashini, 2014] Sabashini M., Sahoo S. (2014). Pulse coupled neural networks and its applications. *Expert systems with applications*, n° 41, pp. 3965-3974.
- [Syed, 2014] Syed U. A., Kunwar F., Iqbal M. (2014). Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots. *Robotics and Autonomous Systems* Vol. 62 pp. 474–486.
- [Thulasiraman, 2016] Thulasiraman K., Arumugam S., Brandstädt A., Nishizeki T. (2016). *Handbook of graph theory, combinatorial optimization, and algorithms*. CRC Press Taylor & Francis Group, LLC.
- [Usman, 2014] Usman Ahmed Syed , Faraz Kunwar, Mazhar Iqbal. Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots. *Robotics and Autonomous Systems* Vol. 62 pp. 474-486 2014.
- [Vanasek, 2019] Vanásek V. (2009). Intelligent and Mobile Robotics Group, [Online], Disponible: <http://imr.felk.cvut.cz/planning/maps.xml>.
- [Wang, 2009] Wang X., Qu H., Yi Z. (2009). A modified pulse coupled neural network for shortest-path problem. *Neurocomputing*, Volumen 72 Números (13-15), pp. 3028–3033. doi:10.1016/j.neucom.2009.04.005.

- [Wu, 2016] Wu, Z., Fu, W., Xue, R., Wang, W. (2016). A Novel Global Path Planning Method for Mobile Robots Based on Teaching-Learning-Based Optimization. *Information*, 7(3), 39. doi:10.3390/info7030039.
- [Yang, 2016] Yang L., Qi J., Song D., Xiao J., Han J., Xia Y. (2016). Survey of Robot 3D Path Planning Algorithms. *Journal of Control Science and Engineering*. Volume 2016, Article ID 7426913.
- [Yang, 2019] Yang G., Lu Z., Yang J., Wang Y. (2019). An Adaptive Contourlet HMM-PCNN Model of Sparse Representation for Image Denoising. *IEEE Transactions on Image Processing*, Volume: 7 pp. 88242-88253 ISSN: 2169-3536.
- [Yao, 2019] Yao, Z., Zhang, W., Shi, Y., Li, M., Liang, Z., & Huang, Q. (2019). ReinforcedRimJump: Tangent-based shortest-path planning for two-dimensional maps. *IEEE Transactions on Industrial Informatics*, 1-1. doi: 10.1109/tii.2019.2918589.
- [Zhang, 2011] Zhang Y., Wu L., Wei G., Wang S. (2011). A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. *Digital Signal Processing* vol. 21, número 4, pp. 517-521

Anexos


Anexo A. Poster presentado en Escuela de Inteligencia Artificial y Robótica 2018 (octubre - 2018)



TECNM
TECNOLOGICO NACIONAL DE
MEXICO

Algoritmo SAM-PCNN para Eficientar la Solución del Problema del Camino más Corto

Oliver Eder Espinosa Meneses
Asesor: Dr. Manuel Mejía Lavalle



cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico

RESUMEN

- La red Neuronal de Pulso Acoplado (PCNN) es adecuada para tratar el problema de la ruta más corta.
- Algunas PCNN requieren muchas iteraciones para encontrar la ruta más corta.
- EL modelo Self Adaptive Autowave Pulse-Coupled Neural Network (SAPCNN) propone una manera de llegar a la solución en menos iteraciones.
- Inspirado en la SAPCNN, se presenta el SAM-PCNN, un modelo modificado que busca la ruta más corta de una manera más eficiente.

INTRODUCCIÓN:

- El problema de la ruta más corta en una red ponderada consiste en encontrar las rutas de longitud más corta desde un nodo de origen hasta un nodo destino.
- La Red Neuronal Pulso Acoplada (PCNN), conocida como la tercera generación de Redes Neuronales Artificiales (RNA) se puede aplicar para resolver este tipo de problemas.
- Se han presentado diversas variantes de PCNN para resolver el problema de la ruta más corta, entre las que se encuentran:
 - AWNN [Ma, 2011]. Velocidad constante de onda
 - AWNN Explícita [Paredes-Cano, 2018]. Velocidad constante con explicitación del conocimiento
 - SAPCNN [U, 2013]. Velocidad auto adaptable.
- Se propone el modelo **SAM-PCNN** el cual es un modelo modificado de PCNN que es capaz de adaptar la velocidad de onda de manera más eficiente que la SAPCNN

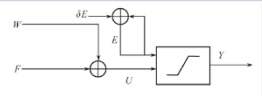
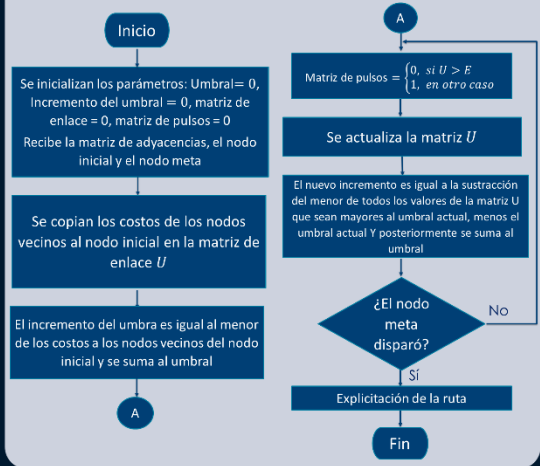


Diagrama de Auto Wave Neural Network

OBJETIVO:

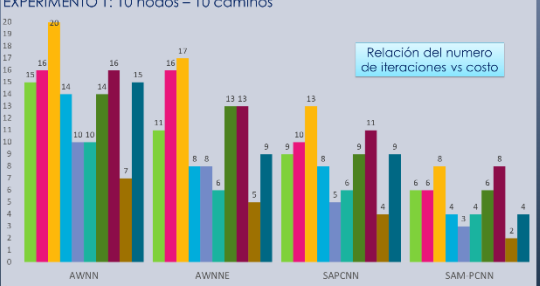
- Implementar, experimentar y evaluar la eficiencia de diferentes modelos de RNA de tipo Pulso-Acoplada en el dominio de la optimización de rutas.

METODOLOGÍA de SAM-PCNN:



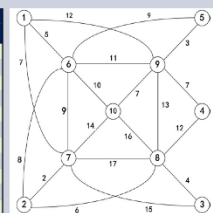
RESULTADOS:

EXPERIMENTO 1: 10 nodos – 10 caminos

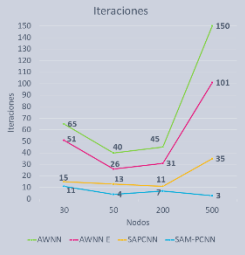


Relación del número de iteraciones vs costo

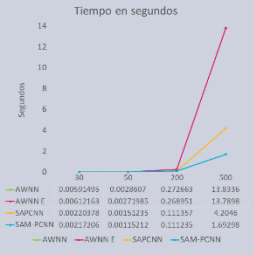
Ruta	Iteraciones				
	AWNN	AWNN E	SAPCNN	SAM-PCNN	
1-6-10	15	15	11	9	6
2-7-10	16	16	10	6	6
3-8-10	20	20	17	13	8
4-9-10	14	14	8	8	4
5-9-10	10	10	8	5	3
6-10	10	10	6	6	4
7-10	14	14	13	9	6
8-10	16	16	13	11	8
9-10	7	7	5	4	2
10-6-1	15	15	9	9	4
Total	137	106	84	51	



EXPERIMENTO 2: 30 a 500 nodos



Iteraciones



Tiempo en segundos

Nodos	AWNN	AWNN E	SAPCNN	SAM-PCNN
30	15	18	11	3
50	21	26	18	7
100	45	31	11	3
200	85	45	11	3
300	101	35	11	3
500	150	35	11	3

CONCLUSIÓN:

- Se implementaron tres modelos de PCNN para la optimización de la ruta mas corta.
- Se propuso el modelo SAM-PCNN.
- De acuerdo con los resultados de los experimentos, la mejora propuesta es hasta 39.28% más eficiente con respecto de la SAPCNN y hasta un 62.77% mas eficiente que la AWNN.

REFERENCIAS:

[Li, 2013]

[Ma, 2011]

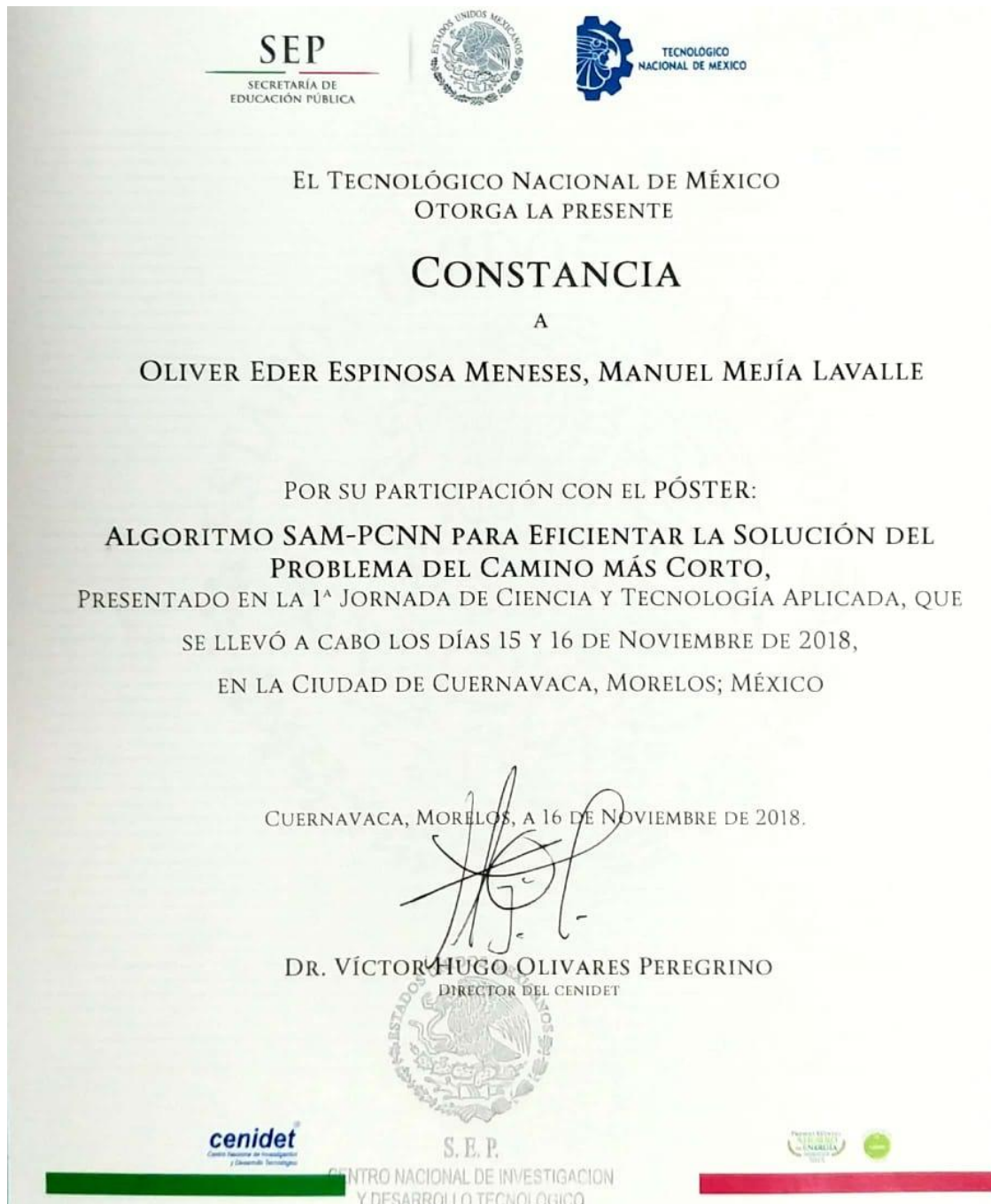
[Paredes-Cano, 2018]

X. Li, Y. Ma y X. Feng. «Self-adaptive autowave pulse-coupled neural network for shortest-path problem», Neurocomputing, vol. 115, p. 63–71, 2013.

Y. Ma, K. Zhan, Z. Wang, Applications of pulse-coupled neural networks, Springer, 2011.

J. J. Paredes-Cano, M. Mejía-Lavalle, D. Mújica y G. Reyes-Salgado, «Red Neuronal Pulsante Explícita para el Problema del Camino más Corto», XIV Congreso Internacional sobre Innovación y Desarrollo Tecnológico , 2018.

Anexo B. Presentación de poster 1ra Jornada de Ciencia y Tecnología Aplicada (noviembre - 2018)



Anexo C. Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2018)



ICMEAE
INTERNATIONAL CONFERENCE ON MECHATRONICS, ELECTRONICS AND AUTOMOTIVE ENGINEERING

El Centro de Investigación en Ingeniería y Ciencias Aplicadas de la Universidad Autónoma del Estado de Morelos y El Instituto de Ingenieros Electrónicos y Eléctricos de Morelos A.C.

otorgan el presente

RECONOCIMIENTO

A: **Oliver Eder G. Espinosa Castro**

Por su destacada participación como: **Concursante:**

En la categoría Innovación en el Advanced Robotics and Drone Competition.

En el marco del Congreso Internacional de Ingeniería Mecatrónica, Electrónica y Automotriz realizado del 27 al 30 de Noviembre del 2018 en la ciudad de Cuernavaca, Morelos, México.


Ing. Leoncio Aguilar Négrete
Presidente del Instituto de Ingenieros Electrónicos y Eléctricos de Morelos A.C.

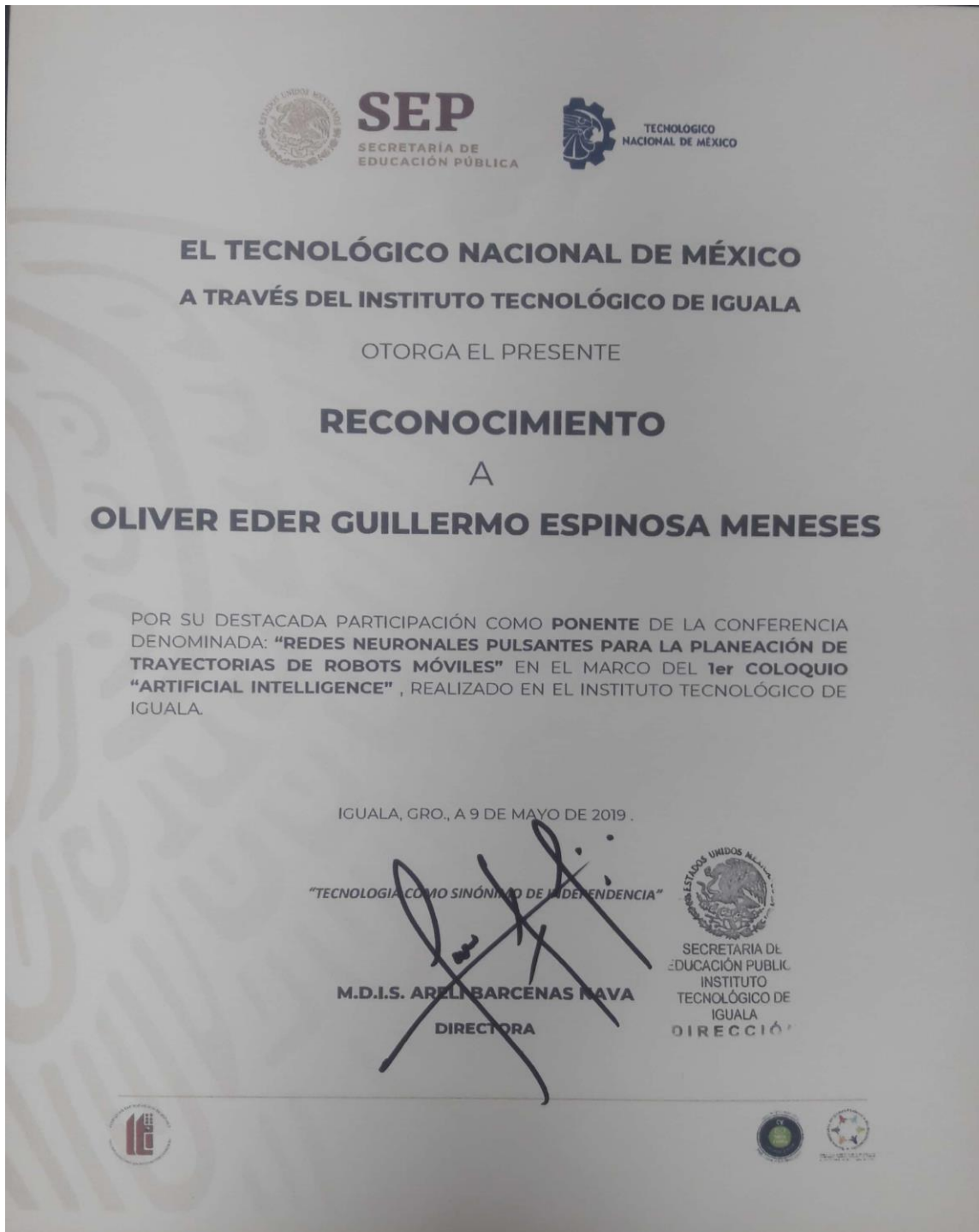

Dra. Elsa Carmina Menchaca Campos
Directora Interina del Centro de Investigación en Ingeniería y Ciencias Aplicadas.



Anexo D. Presentación de poster 2da Jornada de Ciencia y Tecnología Aplicada (abril - 2018)



Anexo E. Presentación para el 1er coloquio “Artificial Intelligence” (mayo - 2019)



Anexo F. Participación como ponente en el XI Congreso Mexicano de Inteligencia Artificial (junio -2019)



Anexo G. Participación como ponente en la 1er Jornada de Tecnologías de la información 2019



Anexo H. Participación como ponente en 3ra Jornada de Ciencia y Tecnología Aplicada (noviembre -2019)



Anexo I. Participación como ponente en Presentación de poster ICMEAE con ISBN: 978-1-5386-9190-8 (noviembre - 2019)



Anexo J. Publicación de Artículo en la revista “Research in Computing Science”
(noviembre - 2019)

ISSN 1870-4069

Variante eficiente de red neuronal pulsante con auto-onda dinámica para resolver el problema del camino más corto

Oliver Espinosa, Manuel Mejía-Lavalle, José Ruiz, Gerardo Reyes, Alicia Martínez

Tecnológico Nacional de México / CENIDET,
Cuernavaca, Morelos, México
{oliver.espinosa18ce, mlavalle,
josera, greyes, amartinez}@cenidet.edu.mx

Resumen. Desde hace aproximadamente 10 años se ha estudiado el comportamiento de las Redes Neuronales Artificiales Pulsantes en el campo de la Optimización de Trayectorias, específicamente en el problema de la ruta más corta. Sin embargo, existen modelos de Redes Neuronales Pulsantes que necesitan un gran número de iteraciones antes de encontrar la ruta más corta entre un punto y otro. En este artículo se presenta una variante de Red Neuronal Pulsante para resolver el problema de la ruta más corta de manera eficiente. Esta variante cuenta con una velocidad dinámica de propagación de auto-onda, la cual se ajusta de manera heurística para evitar iteraciones donde no hay cambios en la Red. Para mostrar la eficiencia del modelo, se realizan experimentos y se comparan los resultados contra los obtenidos por otros modelos de Redes Neuronales Pulsantes. Para la comparativa se utilizan paradigmas que emplean una velocidad de auto-onda estática y modelos con una velocidad de auto-onda dinámica.

Palabras clave: red neuronal pulsante, problema del camino más corto, optimización, auto-onda dinámica.