



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PROTOCOLO DE AHORRO DE ENERGÍA PARA MEJORAR LA VIDA ÚTIL EN UNA RED INALÁMBRICA DE SENSORES
USANDO EL AUTÓMATA CELULAR DEL JUEGO DE LA VIDA

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
JOEL REYES VARGAS

TUTOR:
LUIS FRANCISCO GARCÍA JIMÉNEZ / Facultad de Ingeniería UNAM

CIUDAD UNIVERSITARIA, DICIEMBRE DE 2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mi familia, gracias por su dedicación y confianza, y principalmente por procurarme las condiciones que me permitieron cursar una educación universitaria, así como estudios de posgrado. Agradezco especialmente a mis padres y hermanas, todo lo que soy es gracias a ustedes, por lo que todos mis logros son también suyos.

Gracias a las personas que han formado parte de mi desarrollo académico, tanto profesores como compañeros. Agradezco también a mis amigos por el apoyo y las risas, gracias por estar siempre cuando los necesito.

De la misma forma agradezco al Dr. Luis Francisco García Jiménez por su ayuda en el desarrollo de este trabajo, así como las aportaciones y recomendaciones para mejorarlo por parte de la Dra. María Elena Lárraga Ramírez y el Dr. Javier Gómez Castellanos.

Finalmente, agradezco el apoyo brindado por parte de CONACYT y especialmente al proyecto DGAPA-PAPIIT IA102822.

Índice general

Resumen	IX
1. Introducción	1
1.1. Definición del problema	4
1.2. Hipótesis	4
1.3. Metas	4
1.3.1. Meta general	4
1.3.2. Metas particulares	5
1.4. Metodología	5
1.5. Contribución	5
1.6. Descripción del contenido	5
2. Trabajo relacionado	7
3. Autómatas celulares	11
3.1. Espacio celular	11
3.2. Conjunto de estados	13
3.3. Vecindad	13
3.4. Reglas de transición	16
3.4.1. Clase 1	17
3.4.2. Clase 2	17
3.4.3. Clase 3	17
3.4.4. Clase 4	17
3.5. Autómatas síncronos	17
3.6. Autómatas asíncronos	18
3.7. Sistema complejo	18
3.8. Game of Life	18
3.8.1. Espacio celular	18

3.8.2.	Conjunto de estados	18
3.8.3.	Vecindad	19
3.8.4.	Reglas de transición	19
4.	Game of Sensors	25
4.1.	Reglas de transición	27
4.2.	Algoritmo A-star	30
4.3.	Análisis de desempeño	36
5.	Experimentos y resultados	45
5.1.	Modelo de energía	46
5.2.	GoS contra EERP	47
5.3.	GoS contra LPA-star	51
5.4.	GoS contra Maximizing	53
6.	Conclusiones	57
6.1.	Validación de la hipótesis	57
6.2.	Conclusiones	57
6.3.	Perspectivas de investigación	58

Índice de figuras

3.1. Espacios celulares Abelianos.	12
3.2. Espacio celular no Abeliano.	12
3.3. Espacio celular con diferentes dimensiones.	13
3.4. Vecindad de <i>Moore</i> con diferentes radios.	14
3.5. Vecindad de <i>von Neuman</i> con diferentes radios.	14
3.6. Vecindad de <i>Moore</i> de radio 1 en espacio celular no regular.	15
3.7. Patrón de radiación omnidireccional.	15
3.8. Vecindad definida de acuerdo a un patrón de radiación omnidireccional. . .	16
3.9. Vecindad de patrón de radiación omnidireccional en geometría no regular. .	16
3.10. CA con patrón: <i>Symmetric die hard</i>	23
4.1. Patrón estático bajo las reglas del juego de la vida.	27
4.2. Regla de calendarización.	29
4.3. Regla de probabilidad, donde los sensores que deben pasar a estado activo respecto a las reglas anteriores realizan el cambio de estado solo con una probabilidad p , por lo que en este caso el sensor central no realiza el cambio de estado.	30
4.4. Funcionamiento del algoritmo A-star en un escenario con obstáculo.	31
4.5. Funcionamiento del algoritmo <i>A-star</i> en el modelo GoS, donde la ruta óptima se calcula localmente en cada paso de tiempo.	33
4.6. Nodos generadores de paquetes y nodo <i>sink</i>	35
4.7. Posible escenario inicial.	36
4.8. Convergencia de resultados.	37
4.9. Impacto de la variación del tiempo de calendarización <i>RT</i> en el promedio de celdas activas y la vida de la red.	39
4.10. Promedio de nodos activos respecto a la vida de la red.	40
4.11. Estado de la red en distintos tiempos.	41
4.12. Vecinos activos respecto al porcentaje de nodos en la red con batería.	42

4.14. Impacto de la variación del parámetro p	43
5.1. Escenario con nodos distribuidos aleatoriamente.	47
5.2. Número de nodos activos respecto a la vida de la red con una distribución aleatoria de nodos.	49
5.3. Número de nodos activos respecto al número de paquetes transmitidos, entre GoS, EERP y A&F.	50
5.4. Energía restante promedio respecto al número de paquetes transmitidos entre EERP, A&F y GoS.	51
5.5. Número de nodos con batería respecto a los paquetes transmitidos, entre EERP, LPA-star y GoS.	53
5.6. Energía promedio restante respecto a los paquetes transmitidos, comparando EERP, LPA-star y GoS.	54
5.7. Porcentaje de nodos con batería respecto al número de paquetes transmitidos, entre q – <i>switch</i> , Maximizing y GoS.	56

Índice de tablas

4.1. Comparación de desempeño del algoritmo A-star bajo el modelo de GoL. . .	36
4.2. Desempeño del CA GoL.	43
4.3. Desempeño del modelo GoS.	43
4.4. Desempeño del modelo GoS cuando 20 % de los nodos han agotado su batería. . .	44
5.1. Parámetros de simulación.	49
5.2. Parámetros de simulación.	52
5.3. Parámetros de simulación.	55

Resumen

Las redes inalámbricas de sensores (*Wireless Sensor Network*) han incrementado su popularidad debido a que desempeñan un papel muy relevante en múltiples aplicaciones, algunas de estas son la monitorización ambiental, detección de incendios, protección de fauna y el cuidado de la salud. Estas aplicaciones recolectan información de forma periódica a través de varios sensores, donde esta información es enviada hacia un nodo especial llamado nodo *sink* o nodo sumidero, para ello, usan un modelo de comunicación conocido como *many-to-one*, el cual va enviando la información de la periferia de la red hacia el nodo sumidero. Este modelo de comunicación requiere que los protocolos de encaminamiento se enfoquen en minimizar el consumo energético, debido a que el número de paquetes que un sensor debe retransmitir se incrementa conforme su distancia al nodo *sink* disminuye. Esto provoca que los sensores más cercanos al nodo *sink* consuman más energía que aquellos sensores que se encuentran más alejados.

Con la finalidad de evitar interrupciones en la comunicación causada por el agotamiento de la energía bajo este modelo de comunicación, en la actualidad se han propuesto múltiples técnicas. Por ejemplo, algunas de ellas proponen incrementar la energía de los sensores más cercanos al nodo *sink*, para ello, dividen la red en anillos concéntricos, donde el nodo *sink* se encuentra justo en el centro de todos, sin embargo, estas técnicas requieren de un ordenamiento predefinido de los dispositivos que componen la red. Esta disposición no siempre se adapta a todos los escenarios en los que se puede desplegar una red de sensores, especialmente en lugares remotos, donde se suelen arrojar los sensores mediante avionetas de manera aleatoria. Otros trabajos usan lógica difusa (*fuzzy logic*) para reducir el consumo de energía en los nodos de la red, sin embargo, estas propuestas no siempre garantizan un consumo homogéneo de la energía, lo que puede provocar que ciertas áreas se queden sin energía y haya interrupciones en la comunicación. Como consecuencia de estos problemas, en esta tesis se propone un modelo de ahorro energético basado en el autómata celular del *juego de la vida*, que combinado con el algoritmo de encaminamiento *A-star* proporciona un consumo homogéneo de la energía entre los sensores. Esto provoca que la vida útil de la red se extienda, ya que la energía es utilizada de manera óptima. Para ello, se consideran factores como la energía residual, el número de vecinos activos y un calendario

prendido/apagado en los nodos de la red. Nuestras simulaciones demuestran que el modelo propuesto mejora la vida útil de la red en comparación con los trabajos más recientes en el tema.

Capítulo 1

Introducción

Las redes inalámbricas de sensores (*Wireless Sensor Networks*) se han destacado como una tecnología capaz de realizar tareas complejas tales como monitorización ambiental, seguimiento de animales, y diversas tareas para la implementación de ciudades inteligentes. En estas tareas, usualmente los nodos que componen a la red son arrojados de forma aleatoria sobre un área geográfica sin una infraestructura preestablecida. Estos nodos pueden recolectar información sobre algún fenómeno físico y reportarlo hacia un nodo sumidero, también llamado nodo *sink*, a través de comunicación directa (*single-hop*) o a través de otros nodos (*multi-hop*) mediante reenvíos de la información. En un modelo de comunicación *multi-hop*, los nodos sensores alrededor del nodo *sink* son sometidos a una mayor carga de tráfico de información que los nodos que se encuentran en áreas alejadas del *sink*, esto es debido a que entre más alejado del nodo sumidero se encuentren, menos probabilidad existe de que estos nodos deban retransmitir paquetes de otros nodos. Esta situación implica que los sensores cercanos al nodo *sink* consuman su energía más rápido, provocando que la información pueda no ser entregada de forma correcta al nodo *sink*. Por esto, implementar protocolos de encaminamiento con alta eficiencia energética es una tarea crucial en las WSN. Especialmente, porque en este contexto la energía que alimenta a los nodos sensores es suministrada a través de pequeñas baterías que pueden requerir de procedimientos complejos para ser recargadas o cambiadas en entornos hostiles como volcanes, zonas de riesgos e inundaciones.

Recientemente, se han propuesto múltiples técnicas para lidiar con el problema del consumo energético en WSN. Estas técnicas pueden ser clasificadas en cuatro principales categorías: optimización en el uso de los radios, implementación de esquemas de calendarización, cosecha de energía, y protocolos de encaminamiento con consumo energético eficiente. Por ejemplo, los esquemas de optimización en el uso de los radios, modifican parámetros del módulo interno tales como la potencia de transmisión [1] o la modulación, con la fina-

lidad de disminuir el consumo de energía en la batería. Por otro lado, los esquemas de calendarización consisten en llevar a los sensores a un estado de reposo periódicamente con base en un esquema que define los tiempos en los que un nodo debe permanecer en reposo o activo basado en parámetros como la latencia, la carga de tráfico, o las características topológicas de la red. Finalmente, los esquemas de cosecha de energía usan los avances más recientes en el campo de la microelectrónica para desarrollar nuevos sensores capaces de recolectar energía de su entorno por medio de la energía eólica, la radiación solar, o incluso la energía cinética.

Por otro lado, es bien sabido que la transmisión de información está clasificada como la actividad que requiere mayor consumo de energía en las WSN [2]. Es por esto que crear esquemas de encaminamiento que sean eficientes en términos energéticos puede aumentar la vida útil de la red. Con este propósito, varios mecanismos han sido propuestos recientemente, la mayoría de estos se basan en *clustering*, donde esta técnica consiste en dividir a los nodos que componen a la red en grupos llamados *clusters*, la técnica para formar estos grupos puede cambiar de acuerdo a la propuesta. Cada *cluster* selecciona un nodo líder entre todos ellos comúnmente llamado *cluster head* (CH), este nodo es elegido de acuerdo a diversos criterios, por ejemplo, su cantidad de energía disponible. La principal tarea de este nodo es ser el responsable de recolectar la información generada por su grupo para enviarla al nodo *sink*, ya sea directamente o a través de otros CH. Aunque estas propuestas disminuyen el consumo energético a través de la red, suelen usar técnicas complicadas para seleccionar a los CH. Además de que no siempre garantizan un balance en el consumo energético entre los nodos. Por otro lado, en los patrones de comunicación *many-to-one*, los nodos más cercanos al nodo *sink* agotan su batería más rápido que el resto, por ende, existen mecanismos que dividen la red en anillos concéntricos con la finalidad de brindar mayores recursos energéticos a los nodos ubicados en las áreas con mayor carga de tráfico, es decir, las zonas cercanas al nodo *sink*, de tal forma que se logre balancear el consumo de la energía entre los sensores. Esto provoca que todos los nodos consuman la totalidad de su energía al mismo tiempo, en otras palabras, que la vida útil de la red (periodo en el que la red es capaz de transmitir información al nodo *sink*) termina sin energía residual en los nodos. Sin embargo, estos esquemas requieren de una distribución de nodos predefinida, lo cual no siempre se ajusta a todos los escenarios en los que puede ser desplegada una WSN. Adicionalmente, la mayoría de estas técnicas solo funcionan al considerar una tasa de información constante. Otros mecanismos que estudian este problema usan lógica difusa (*fuzzy logic*) combinada con técnicas de encaminamiento. Sin embargo, estas propuestas no siempre resultan en un consumo de energía balanceado.

Contrario a todos los métodos mencionados anteriormente, en este trabajo de tesis se propone un modelo simple basado en autómatas celulares en conjunto con el algoritmo *A-star* para crear un esquema de encaminamiento con consumo energético eficiente.

Un autómata celular o *Cellular Automaton* (CA) es un sistema discreto y descentralizado usado para modelar sistemas físicos, el cual está conformado por entidades conocidas como celdas. Cada una de estas celdas tiene un conjunto de estados posibles, una vecindad conformada por más celdas con las cuales puede interactuar y que a su vez comparten un número finito de reglas de transición, siendo estas últimas las que determinan el comportamiento del autómata celular. Con base a las reglas de transición, el autómata celular puede ser usado para modelar fenómenos complejos, por ejemplo; el tránsito urbano o el flujo del agua en un río. Un CA también puede ser usado para modelar una WSN, en la cual un sensor (celda) puede estar en estado activo o inactivo, mientras que la vecindad de cada celda está formada por los sensores dentro de su rango de comunicación. En esta tesis, el estado activo representa un periodo en el que un sensor puede monitorizar el medio, transmitir y/o recibir paquetes de información. Mientras, el estado inactivo representa un periodo en el que un sensor trata de mantenerse en reposo tanto como le es posible.

Con la finalidad de reducir el consumo energético en WSN, en esta tesis se usa un autómata celular conocido como *el juego de la vida* o *Game of Life* (GoL) [3, 4]. Para este autómata, las reglas de transición son las siguientes:

- Una celda viva (estado activo) con menos de dos vecinos vivos muere (estado inactivo) por aislamiento.
- Una celda viva sobrevive si tiene dos o tres vecinos vivos.
- Una celda viva con más de tres vecinos activos muere por sobre población.
- Una celda muerta con exactamente tres vecinos vivos cobra vida.

Bajo estas reglas, el juego de la vida evoluciona a través del tiempo, sin embargo, algunas celdas permanecerán en su estado actual, mientras que otras cambiarán. Esta evolución afecta el consumo energético y la vida útil de la red, ya que estas reglas evitan que todos los nodos se mantengan activos al mismo tiempo. De esta forma, las reglas presentadas pueden reducir el consumo energético en las WSN y a su vez pueden evitar que se envíe información redundante hacia el nodo *sink*, al mantener una cantidad controlada de nodos activos en una región o área de estudio específica. Sin embargo, conforme se va desarrollando el autómata celular, los complejos patrones formados por las reglas de GoL pueden formar agujeros en la red, es decir, puede existir una gran cantidad de nodos en estado inactivo, lo cual puede provocar fallos en los esquemas de encaminamiento y pérdida de información. Por lo que, para prevenir este problema, en esta tesis se proponen nuevas reglas basadas en las reglas de GoL, las cuales consideran variables como la energía restante en los nodos, el número de vecinos activos y un esquema de calendario encendido/apagado para extender la vida útil de la red.

Debido a que GoL es un CA basado en sistemas biológicos, no considera el reenvío de paquetes, por lo tanto se consideró adoptar el uso del algoritmo de encaminamiento *A-star* con la finalidad de minimizar el número de saltos desde el nodo origen del paquete hasta el nodo *sink*. La unión entre las nuevas reglas del juego de la vida y el algoritmo *A-star* lo llamamos *Game of Sensors (GoS)*.

Las simulaciones de los experimentos muestran que el método GoS mejora el consumo energético mientras que aumenta la vida útil de la red en comparación de las propuestas más recientes en la literatura.

1.1. Definición del problema

Las redes inalámbricas de sensores suelen estar formadas por pequeños dispositivos, los cuales cuentan con baterías de capacidades limitadas, por lo que es de gran importancia encontrar métodos que aprovechen al máximo estos recursos con la finalidad de extender la vida útil de la red, además de asegurar que se mantiene el correcto funcionamiento de la misma, es decir que siga siendo capaz de entregar paquetes al nodo *sink* a través de nodos de retransmisión (*relay*).

Se han hecho diversas propuestas para disminuir el impacto que implica este problema, algunas de estas hacen uso de autómatas celulares, *clustering*, lógica difusa, entre otras técnicas. Sin embargo, ninguna de ellas ha mostrado un consumo energético balanceado entre los nodos basado en propuestas simples que puedan adaptarse de manera rápida a sensores con poca memoria y capacidad de procesamiento.

1.2. Hipótesis

Añadir nuevas reglas al modelo del juego de la vida (GoL) e incluir un protocolo de encaminamiento basado en el algoritmo *A-star* permite extender la vida útil de la red, además de asegurar el envío de paquetes a través de la red hacia el nodo *sink*.

1.3. Metas

1.3.1. Meta general

Proponer un modelo para WSNs basado en el autómata celular del juego de la vida, que extienda la vida útil de la red y asegure el correcto envío de paquetes hacia el nodo *sink*.

1.3.2. Metas particulares

- Proponer un modelo para extender la vida útil de la red basado en el autómata GoL.
- Proponer un algoritmo de encaminamiento dinámico basado en *A-star* que se adapte al modelo GoL.
- Verificar el funcionamiento del modelo con una distribución de nodos aleatoria.
- Realizar simulaciones y comparaciones con los trabajos relacionados más recientes con el fin de cuantificar la efectividad de GoS.

1.4. Metodología

1. Estudiar el modelo para sistemas complejos basado en el autómata celular del juego de la vida (GoL) y su desempeño al usarse en una WSN.
2. Añadir nuevas reglas al modelo que le permitan adaptarse a las necesidades de una WSN, así como adaptar el algoritmo de encaminamiento A* para lograr la transmisión de paquetes dentro del modelo.
3. Crear un ambiente de simulación que permita comparar las principales características de GoS con algunos de los trabajos relacionados más recientes como los desarrollados en las referencias [5, 6] y [7], con la finalidad de validar el modelo propuesto.

1.5. Contribución

Se propone un modelo que extiende la vida útil de la red basado en el autómata celular del juego de la vida que garantiza el correcto envío de paquetes, y supera el desempeño de los trabajos relacionados más recientes.

1.6. Descripción del contenido

Con la finalidad de alcanzar la meta planteada previamente, la estructura de esta tesis se plantea a continuación:

- El capítulo 2 presenta los trabajos relacionados más relevantes sobre el consumo energético en redes de sensores usando autómatas celulares o sobre aquellos trabajos que aseguran el correcto envío de paquetes hacia el nodo sink.

- El capítulo 3 desarrolla el concepto del autómata celular como herramienta para el modelado de sistemas complejos, además se describe el autómata celular del juego de la vida.
- El capítulo 4 describe el funcionamiento del modelo propuesto en esta tesis, el cual recibe el nombre de Game of Sensors (GoS).
- El capítulo 5 presenta las simulaciones realizadas, así como comparaciones respecto a otros trabajos relacionados.
- El capítulo 6 presenta las conclusiones generales del trabajo, la verificación de la hipótesis y las perspectivas de investigación.

Capítulo 2

Trabajo relacionado

Desde el comienzo del desarrollo de WSN se le ha dado especial atención al problema de maximizar su vida útil. El escenario más estudiado en relación a este problema es la búsqueda de esquemas de encaminamiento energéticamente eficientes. La mayoría de estas estrategias dividen el área geográfica en secciones llamadas *clusters*. Cada *cluster* cuenta con un *cluster head* (CH), el cual es responsable de administrar las actividades locales y comunicarse con otros CH. Los CH son elegidos por parámetros como la energía residual, su ubicación respecto al nodo *sink*, cobertura y su carga de tráfico, entre otras. Las técnicas que usan *clusters* mejoran la eficiencia energética debido a que tienen la capacidad de apagar los nodos que no se utilizan localmente, esto gracias a la administración de recursos que realiza el CH [8, 9, 10]. De esto, se puede deducir que el consumo energético del CH es superior al del resto de los nodos, por lo que al rotar el papel del CH entre todos los nodos es posible extender la vida útil de la red. Sin embargo, las técnicas para rotar los CH utilizan mecanismos complicados para seleccionar al mejor nodo candidato.

Recientemente otros trabajos de investigación [7, 11, 12, 6] han atacado el problema de consumo de energía usando una combinación del algoritmo *A-star* y la lógica difusa (*fuzzy*) sin necesidad de usar las técnicas de *clusters* previamente mencionadas. De cualquier forma, aunque estos mecanismos extienden la vida útil de la red, no siempre garantizan un consumo de energía homogéneo entre los nodos. Otros trabajos usan estrategias de lógica difusa con algoritmos de encaminamiento basados en el comportamiento de las colonias de hormigas con la finalidad de lograr un consumo homogéneo de la energía [13, 14].

Adicionalmente, los modelos basados en autómatas con aprendizaje se han convertido en una gran alternativa para el estudio de WSN. El autómata con aprendizaje o *learning automata* (LA) es una técnica en la que el proceso de aprendizaje está basado tanto en recompensas como en penalizaciones de acuerdo al objetivo. Este proceso hace que las variables del LA alcancen los valores óptimos al interactuar con su entorno a través del desarrollo

en el tiempo del sistema. Por ejemplo, en las referencias [15] y [16] los autores usaron un modelo basado en autómatas con aprendizaje para seleccionar un CH óptimo. Por otro lado, en la investigación [17] se usa un modelo basado en LA para minimizar el número de retransmisiones de un paquete para alcanzar su destino en un esquema de encaminamiento usando diagramas de *Voronoi*. De la misma forma, la propuesta de la referencia [18] usa un modelo basado en LA para optimizar un algoritmo de encaminamiento en el escenario de una red Ad Hoc móvil MANET (mobile Ad Hoc network). También, los autores en [19] aseguran que los LA mejoran el consumo energético, aumentan la tasa de envío de paquetes y logran un consumo homogéneo de la energía cuando son combinados con otras técnicas. Sin embargo, la implementación de LA en WSN requiere de múltiples comunicaciones entre nodos para conocer el estado general de la red y de algunos nodos en específico para poder establecer los castigos y las recompensas con base en la información obtenida, en muchos casos incluso los nodos requieren de conocer la información completa de la red en todo momento para poder establecer los parámetros de aprendizaje, lo que en un escenario real requeriría del envío constante de mucha información y con ello un consumo alto de energía.

En los patrones de tráfico de información *many-to-one*, los sensores más cercanos al nodo *sink* consumen su batería más rápido que los sensores ubicados en otras posiciones debido a que se convierten en nodos de retransmisión constantemente. Esta característica puede crear *agujeros* (regiones de la red con varios nodos inactivos o sin batería restante) o interrupciones en el proceso de recolección de información. Con la finalidad de resolver este problema, se han propuesto estrategias de distribución de nodos, de tal forma que se concentre más capacidad de transmisión en las áreas con mayor carga de tráfico y de esta forma homogeneizar el consumo energético general en la red. Por ejemplo, los autores en [20] proponen una estrategia de distribución de nodos no uniforme modelada como una serie de coronas o circunferencias concéntricas, en cuyo centro se encuentra el nodo *sink*. Para balancear el consumo de energía entre los nodos, los autores cambian la densidad de nodos en cada corona en una distribución geométrica desde la corona exterior (la de mayor radio) hasta la corona central (la de menor radio). Adicionalmente los autores proponen un algoritmo de encaminamiento llamado *q-switch* para encontrar la ruta más corta hacia el nodo *sink* basándose en la energía residual de los nodos. Sin embargo, de acuerdo a las simulaciones realizadas por los autores, la corona exterior mantiene un 15% de energía residual cuando la vida útil de la red termina, es decir, cuando ya no es posible transmitir más paquetes al nodo *sink*, lo que significa que el consumo de energía en la red no es completamente homogéneo a pesar de la distribución propuesta de nodos. Además, para lograr estos resultados los nodos deben estar localizados en una posición predefinida que no siempre es adecuada para todas las aplicaciones o escenarios en los que se puede usar una WSN. Asimismo, los autores no consideran el consumo energético que conlleva el

procesamiento de la información, ni de los mensajes de control en los nodos.

En la referencia [21], los autores dividen la red en secciones uniformes para balancear el consumo de energía al emplear dos estrategias llamadas; *inter-slice* e *intra-slice*. En la estrategia *inter-slice*, los nodos modifican su potencia de transmisión usando un modelo de programación lineal basado en la distancia de cada nodo hacia el nodo *sink*. Esta estrategia permite que los nodos cercanos al *sink* conserven su energía al realizar sus transmisiones con un nivel de energía bajo, mientras que los nodos lejanos al *sink* pueden consumir más energía debido a que no son usados como nodos de retransmisión de paquetes frecuentemente. Por otro lado, la estrategia *intra-slice* considera la energía residual de los nodos, de tal forma que los nodos con un nivel de energía bajo le delegan a otros nodos con niveles de energía altos el envío de sus paquetes para conservar energía. Al combinar ambas estrategias, los autores proponen un protocolo de transmisión llamado ETP (*Energy-balanced Transmission Protocol*). Los experimentos realizados indican que el protocolo ETP mejora el desempeño de consumo energético comparado con algoritmos basados en clusters como el de la referencia [22]. Sin embargo, los autores no consideraron las limitaciones de hardware, las cuales no siempre permiten alcanzar altos niveles de potencia de transmisión, así como tampoco permiten regular fácilmente el nivel de potencia de transmisión deseado. Adicionalmente, en la estrategia *intra-slice* tampoco consideran el costo de enviar mensajes de control para conocer cuáles nodos tienen un nivel más alto de energía.

Basándose en [20], los autores de la referencia [23] proponen tres estrategias para balancear el consumo de energía en los nodos en una WSN basada en coronas. En estas estrategias se busca una ubicación óptima de los dispositivos para minimizar el número de nodos en cada una de las coronas. Para lograr esto, se usa una técnica de control de potencia de transmisión, de tal forma que las corona exteriores cubran áreas más amplias que las coronas más cercanas al nodo *sink*. Dado que los nodos cercanos al *sink* pueden usar un nivel de potencia de transmisión bajo, se logra balancear el consumo de energía entre las coronas, y con esto se reduce la energía residual en los nodos. Adicionalmente los autores proponen una técnica en la que cada sensor puede estar inactivo por un largo periodo de tiempo para conservar energía. Sin embargo, los autores no consideran que los dispositivos transmisores o radios comerciales tienen limitaciones de potencia, las cuales no siempre les permiten alcanzar el nivel deseado de potencia de transmisión [1]. Además, el balanceo de consumo de energía se logra únicamente en una distribución predefinida de nodos en la red, lo cual no siempre se ajusta a todas las aplicaciones de las WSN.

En los patrones de transmisión *many-to-one*, la corona central de una WSN basada en coronas no es siempre la región que más energía consume. Es por esto que los autores en [5] proponen una ecuación para buscar las regiones críticas en redes *many-to-one* basadas en coronas, con la finalidad de distribuir la energía total de la red entre los nodos de tal forma que ninguno quede con energía residual al terminar la vida útil de la red. Los autores

afirman que su propuesta de balanceo de consumo de energía mejora la vida útil de la red comparado con lo propuesto en las referencias [20] y [23].

Recientemente, en el campo de los robots autónomos se han usado modelos de autómatas celulares para planear rutas libres de colisiones en ambientes desconocidos u hostiles [24, 25, 26, 27, 28]. Estos mecanismos brindan ventajas en las redes de comunicación *many-to-one* dado que los algoritmos de encaminamiento con consumo de energía eficiente tales como los basados en el algoritmo *A-star* pueden ser usados para planear rutas. Por la misma razón, en esta tesis se propone un protocolo de encaminamiento basado en el algoritmo *A-star* con el objetivo de optimizar el consumo de energía en WSN usando un modelo basado en autómatas celulares.

Con la finalidad de evaluar la efectividad del protocolo propuesto (GoS) en esta tesis, en el capítulo 5 se presenta una comparación con los trabajos propuestos en las referencias [7], [6], [20] y [5]. Contrario a estos trabajos, los cuales usan lógica difusa o distribuciones predefinidas de nodos en coronas, en esta tesis se propone un método simple basado en el modelo del CA del juego de la vida, el cual intrínsecamente permite encender o apagar los nodos de acuerdo a las necesidades de la red. Esta característica incrementa la vida útil de la red mientras que balancea el consumo de energía entre los nodos. Adicionalmente, el algoritmo *A-star* en el modelo *Game of Sensors* puede crear rutas de encaminamiento dinámicas que se pueden adaptar para una distribución uniforme o aleatoria de nodos sensores en una WSN.

Capítulo 3

Autómatas celulares

Un autómata celular es un “*sistema, dinámico, discreto y descentralizado que puede ser usado para modelar sistemas físicos*” [29], este modelo consta de entidades conocidas como células. Cada célula tiene un estado, un vecindario y un número finito de reglas de transición. El desarrollo del sistema en el tiempo t depende del estado del sistema en el tiempo $t - 1$, por lo que se trata de un sistema causal y dinámico, el cual se puede expresar como una cuádrupla A , tal que [30]:

$$A = \{C, \Omega, V, f\},$$

donde C representa el espacio celular o malla, Ω es el conjunto de estados finitos, V representa la vecindad de cada célula, y f representa el conjunto de reglas de transición. A continuación se describen cada uno de estos elementos.

3.1. Espacio celular

El espacio celular es el universo sobre el que se establece el autómata, usualmente conocido como malla. Esta puede ser representada como un grafo con diferentes geometrías. Los espacios celulares están divididos en grupos con geometrías regulares, conocidos como Abelianos, y grupos con geometrías no regulares, conocidos como no Abelianos. Las figuras 3.1 y 3.2 muestran ambos ejemplos.

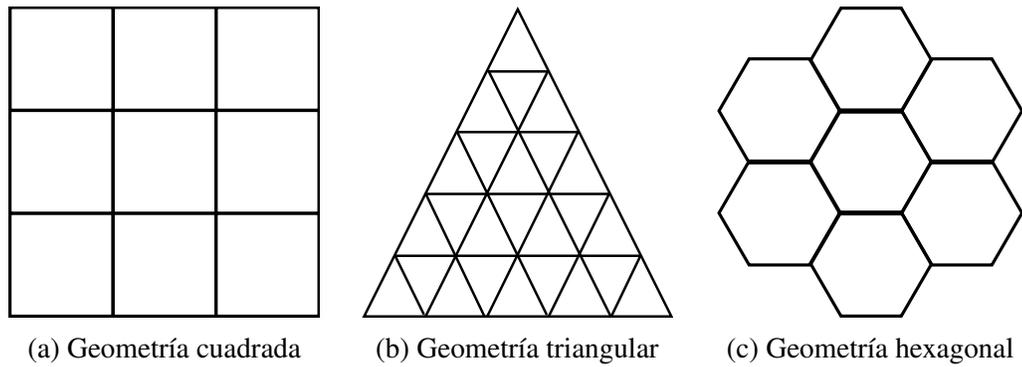


Figura 3.1: Espacios celulares Abelianos.

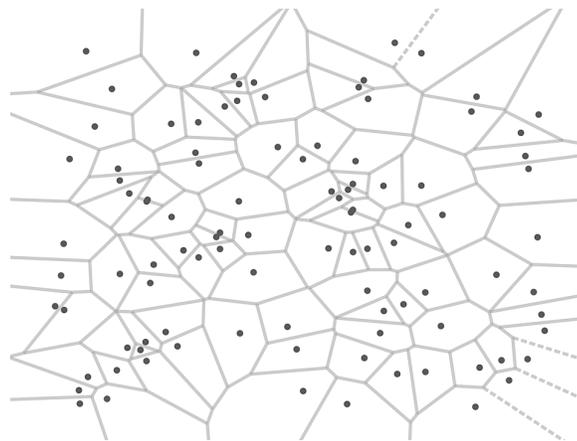


Figura 3.2: Espacio celular no Abeliano.

El espacio celular puede ser de dimensión 1 o superior, y puede extenderse de forma infinita, sin embargo, en este trabajo de tesis se usa un espacio de dos dimensiones. En la figura 3.3 se muestra un espacio celular finito con geometría regular cuadrada de dimensión 1, 2 y 3.

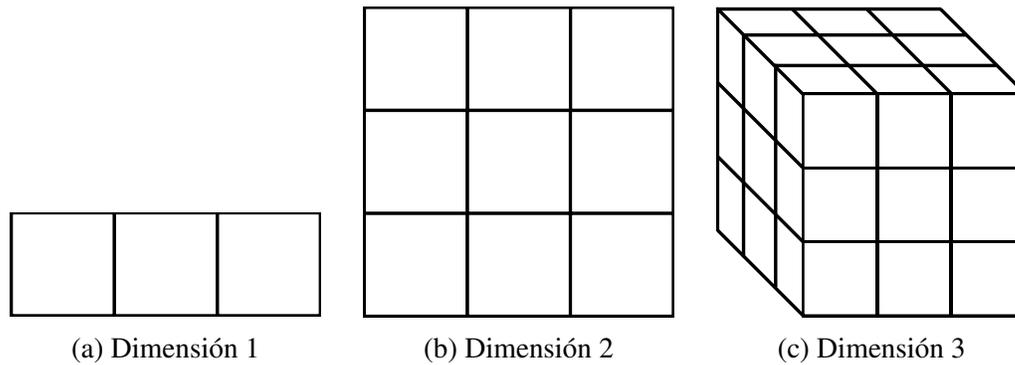


Figura 3.3: Espacio celular con diferentes dimensiones.

Un espacio celular está compuesto por un conjunto de celdas C , las cuales tienen una posición y geometría asignada de acuerdo a las necesidades del modelo que representa el autómata celular. Cuando estas celdas se encuentran en la frontera del espacio celular pueden tener propiedades específicas que modifiquen su comportamiento, ya que modelan las condiciones de frontera del sistema. En el caso de las figuras 3.3a y 3.3b, una celda puede ser cualquiera de los cuadrados que componen el espacio celular, mientras la figura 3.3c, una celda es representada como un cubo en el espacio celular.

3.2. Conjunto de estados

En un autómata celular, el estado de cada celda puede tomar un valor de entre un conjunto finito de valores definidos, este conjunto es conocido como el conjunto de estados. Cada estado tiene características específicas y permiten a cada celda interactuar con el resto de celdas en el espacio celular o exclusivamente con su vecindad, de acuerdo con el comportamiento que se busque replicar.

3.3. Vecindad

Cada una de las celdas que forma al autómata celular tiene la capacidad de interactuar con otras celdas en su entorno de acuerdo a las necesidades del sistema. Una vecindad de una celda es el conjunto de celdas con las que puede interactuar. Una vecindad puede tener diversas formas y radios de alcance. La vecindad más popular es conocida como vecindad de *Moore*, esta vecindad está compuesta por todas aquellas celdas que tengan contacto directo con la celda en cuestión de acuerdo al radio especificado. En la figura 3.4 se puede

observar esta vecindad con diferentes radios en una geometría Abeliiana cuadrada, la celda negra representa a la celda en referencia, mientras que su vecindad se muestra en color gris.

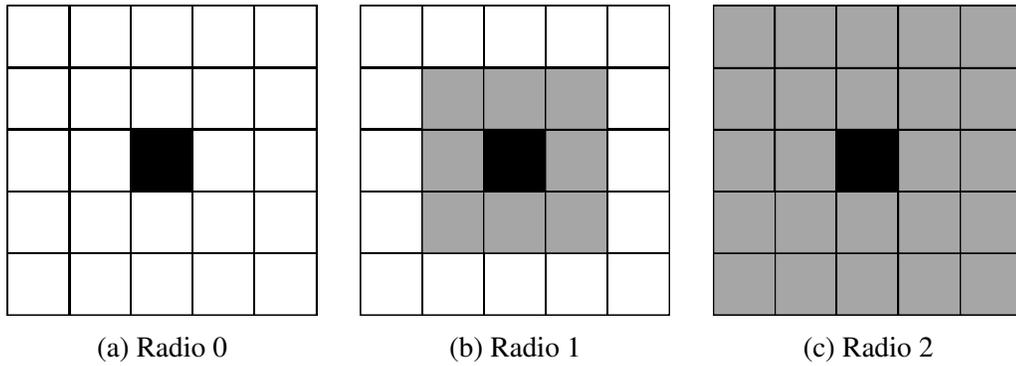


Figura 3.4: Vecindad de *Moore* con diferentes radios.

Otra vecindad popular es conocida como *von Neuman*. En la figura 3.5 se muestra la forma de esta vecindad con distintos radios.

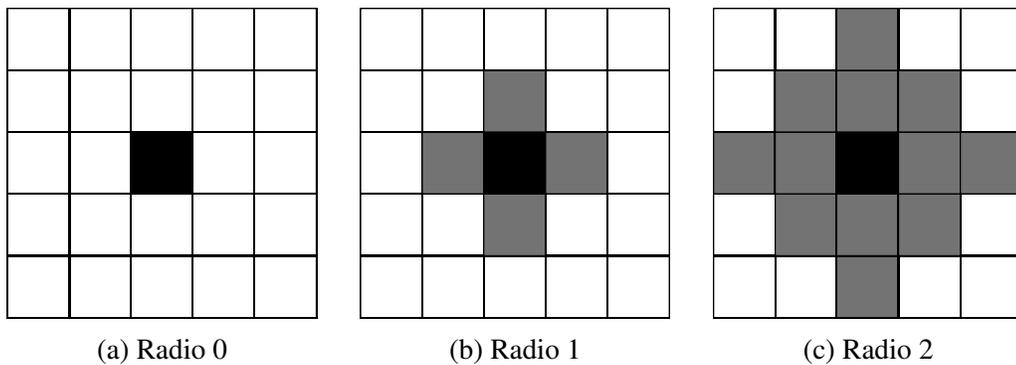


Figura 3.5: Vecindad de *von Neuman* con diferentes radios.

En un espacio celular no regular, la vecindad de *Moore* tiene una geometría variable, pero cumple con la condición de estar en contacto con la celda en cuestión o focal, tal como se muestra en la figura 3.6 para una vecindad de *Moore* de radio 1.

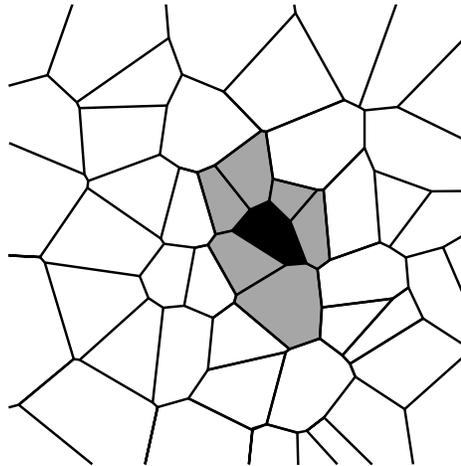


Figura 3.6: Vecindad de *Moore* de radio 1 en espacio celular no regular.

Además de las vecindades previamente establecidas, se puede definir una vecindad de acuerdo a las necesidades del sistema, por ejemplo, si se quisiera representar el rango de transmisión de una antena con un patrón de radiación omnidireccional, se puede definir a la vecindad como el conjunto de celdas que asemejen la cobertura de dicho patrón de radiación. En la figura 3.7 se muestra una representación polar del patrón de radiación omnidireccional.

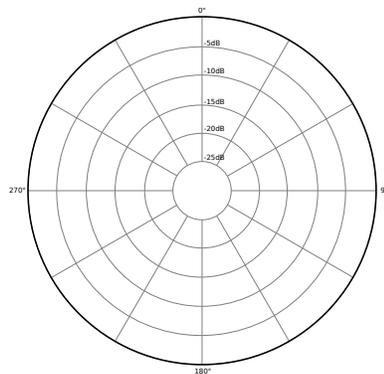


Figura 3.7: Patrón de radiación omnidireccional.

En la figura 3.8 se muestra una vecindad dada por un patrón de radiación omnidireccional con diferentes radios en un espacio celular finito regular. La celda central se representa en negro, la vecindad en gris y el patrón de radiación se representa por una circunferencia punteada. Como se puede observar, este patrón coincide con la vecindad de *Moore*,

por lo que se puede decir que en el contexto de las WSN, una vecindad de *Moore* puede representar un patrón de radiación omnidireccional.

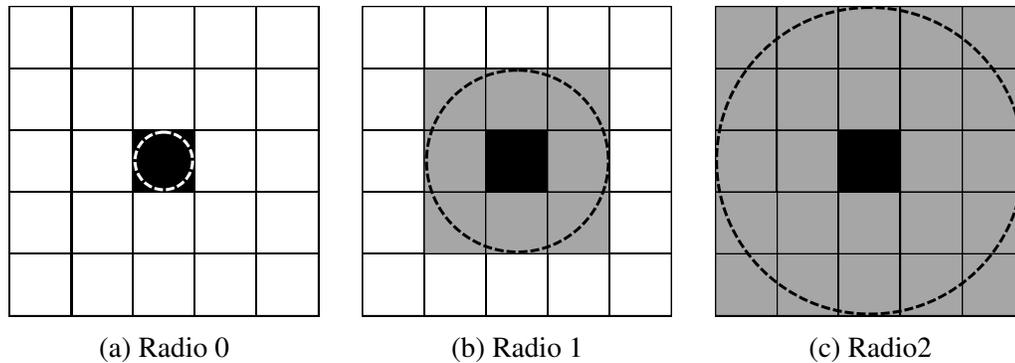


Figura 3.8: Vecindad definida de acuerdo a un patrón de radiación omnidireccional.

De la misma forma, en la figura 3.9 se muestra la misma vecindad, de radio 1 pero en un espacio celular finito no regular.

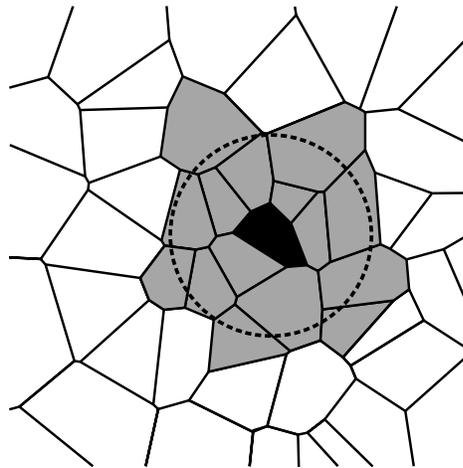


Figura 3.9: Vecindad de patrón de radiación omnidireccional en geometría no regular.

3.4. Reglas de transición

La evolución en el tiempo de los autómatas celulares está determinada por las reglas que se asignan a las celdas, estas reglas tienen como finalidad establecer el estado de cada celda para toda unidad de tiempo. Las reglas de transición también definen la forma en la

que interactúa cada celda con el resto. Dado que se trata de un sistema discreto, el efecto de aplicar las reglas en un tiempo t es perceptible hasta el tiempo $t + 1$.

Los autómatas celulares más usados para modelación son los de dos dimensiones, para los cuales actualmente existen muchas reglas de transición definidas [31], estas describen distintos comportamientos que permiten clasificar a los autómatas celulares de en cuatro clases.

3.4.1. Clase 1

Los autómatas celulares de clase 1 tiene como característica que sin importar el estado inicial del sistema, el autómata llega a un estado uniforme y estable.

3.4.2. Clase 2

Los autómatas celulares de clase 2 tienen como principal característica que se producen estructuras simples que mantienen su forma, las cuales se repiten periódicamente.

3.4.3. Clase 3

Los autómatas celulares de clase 3 tienen como principal característica que se producen patrones complejos que pueden parecer aleatorios, además se producen triángulos en alguna escala.

3.4.4. Clase 4

Los autómatas celulares de clase 4 presentan una combinación de comportamiento ordenado y aleatorio, las estructuras que se forman interactúan con el resto de estructuras formando nuevas estructuras.

3.5. Autómatas síncronos

En un autómata celular síncrono todas las celdas actualizan su estado al mismo tiempo de acuerdo a las reglas de transición, por lo que todo el espacio celular actualiza su estado a la vez. Este tipo de autómatas celulares son usados para modelar sistemas homogéneos, en los que las reglas pueden provocar interacciones locales que no afectan al resto del espacio celular en el tiempo inmediato, sino que los efectos pueden propagarse de una forma discreta manteniendo las propiedades del sistema que se busca modelar.

3.6. Autómatas asíncronos

En un autómata celular asíncrono las celdas pueden actualizar su estado individualmente, esto provoca que los efectos del cambio solo afectan a las celdas con las que interactúa, es decir con su vecindad.

3.7. Sistema complejo

Un sistema complejo es aquel que está compuesto por muchos elementos, los cuales pueden o no interactuar entre sí de una forma no lineal, generando comportamientos espontáneos y emergentes. Cualquier variación en el sistema puede generar cambios significativos en el comportamiento del mismo. En este trabajo de tesis se presenta el modelo de una red inalámbrica de sensores, la cual puede ser tratada como un sistema complejo debido a que su comportamiento obedece a una serie de reglas con un factor aleatorio, donde el estado de cada uno de los nodos depende del estado de los nodos con los que tiene comunicación, además de que la ubicación de los nodos es aleatoria.

3.8. Game of Life

Uno de los autómatas celulares síncronos más populares es conocido como “el juego de la vida” o Game of Life (GoL), este se describe a continuación.

3.8.1. Espacio celular

GoL es un autómata celular implementado en un espacio celular finito con geometría regular con dimensiones $L \times M$, las celdas que lo componen se encuentran posicionadas de forma consecutiva y uniforme. Se define a la celda de acuerdo a sus coordenadas i, j :

$$c = \{(i, j) \mid i, j \in \mathbb{N}, 1 \leq i \leq L, 1 \leq j \leq M\}. \quad (3.1)$$

3.8.2. Conjunto de estados

El conjunto de estados en el autómata celular del juego de la vida está formado por solo dos estados, los cuales son; ‘vivo’ representado con un número ‘uno’ (1) y ‘muerto’ representado con un número cero (0), por lo que el conjunto de estados de una celda se representa como:

$$\Omega_c = \{1, 0\}. \quad (3.2)$$

En esta tesis se representa gráficamente una celda en estado 1 en negro, mientras que las celdas en estado 0 se muestran en blanco.

De la misma forma, el estado de una celda ubicada en el punto (i, j) en el tiempo t se denota como $\omega_{i,j}(t)$.

3.8.3. Vecindad

La vecindad V del CA del juego de la vida es una vecindad de Moore de radio 1, tal como es descrita en la figura 3.4b. La relación entre una celda c y su vecindad V se denota como:

$$V_c(t), \quad (3.3)$$

donde el valor de $V_c(t)$ está dado por la cantidad de celdas en estado '1' que se encuentren dentro de la vecindad V definida para la celda c en el tiempo t .

3.8.4. Reglas de transición

Las reglas de transición pueden modificar el estado de una celda en el tiempo t de acuerdo al estado de la celda y su vecindad en el tiempo $t - 1$, por lo que se define a la aplicación del conjunto de reglas de transición sobre una celda en el tiempo t como:

$$\omega_{i,j}(t) = f(\omega_{i,j}(t-1), \{\omega_{k,l}(t-1)\}), (k,l) \in V_{i,j}, \quad (3.4)$$

donde f es una función de transición arbitraria que opera sobre la celda (i, j) y sobre $V_{i,j}$, siendo esta última la representación del conjunto de celdas (k, l) , las cuales conforman la vecindad de la misma celda (i, j) .

En el juego de la vida, las reglas de transición (f) establecen que, una celda en estado 1 en el tiempo t se mantiene en ese estado en el tiempo $t + 1$ siempre que dentro de su vecindad existan dos o tres celdas con estado 1 en el tiempo t , de lo contrario la celda cambia al estado 0 en el tiempo $t + 1$. Una celda en estado 0 en el tiempo t cambia a estado 1 en el tiempo $t + 1$ solo si en su vecindad hay exactamente tres celdas en estado 1 en el tiempo t , de lo contrario la celda permanece en estado 0 en el tiempo $t + 1$. Esto se puede expresar como:

$$\omega_{i,j}(t) = \begin{cases} 1, & \omega_{i,j}(t-1) = 1, \sum_{(k,l) \in V_{i,j}} V_c(t-1) = 2,3 & (3.5) \\ 0, & \omega_{i,j}(t-1) = 1, \sum_{(k,l) \in V_{i,j}} V_c(t-1) \neq 2,3 & (3.6) \\ 1, & \omega_{i,j}(t-1) = 0, \sum_{(k,l) \in V_{i,j}} V_c(t-1) = 3 & (3.7) \\ 0, & \omega_{i,j}(t-1) = 0, \sum_{(k,l) \in V_{i,j}} V_c(t-1) \neq 3, & (3.8) \end{cases}$$

donde $V_c(t-1)$ representa el número de vecinos activos (en estado 1) en el tiempo $t-1$ dentro de la vecindad $V_{i,j}$ de la celda de interés (i, j) . En el pseudocódigo 1 se muestra la implementación de las reglas del juego de la vida.

El autómata celular del juego de la vida le debe su nombre a que surgió con la finalidad de modelar sistemas biológicos, debido a que las reglas de transición simulan el comportamiento de una población con recursos limitados en un área geográfica definida. La ecuación 3.5 establece que si una celda está viva (1) y tiene dos o tres vecinos vivos, los recursos son suficientes para mantener a la celda viva. La ecuación 3.6 establece que si una celda está viva; en caso de tener más de tres vecinos vivos la celda muere por falta de recursos en la vecindad, por el contrario si se tienen menos de dos vecinos vivos la celda muere por falta de población en la vecindad. La ecuación 3.7 establece que si una celda está muerta (0) y tiene exactamente tres vecinos vivos en su vecindad se dan las condiciones para que la celda recupere la vida. Finalmente, la ecuación 3.8 establece que si una celda está muerta y no tiene tres vecinos vivos permanece muerta.

Aunque el autómata celular del juego de la vida surgió con la finalidad de modelar sistemas biológicos, este puede ser usado para modelar diversos tipos de sistemas, como son las redes inalámbricas de sensores (WSN). Por ejemplo, el conjunto de reglas de transición del juego de la vida presentadas en las ecuaciones 3.5-3.8 pueden evitar que los nodos envíen información redundante al nodo *sink* debido a que cuando existen nodos que se encuentran cercanos entre sí, estos recolectan y envían información similar, generando información redundante. Adicionalmente, estas reglas pueden disminuir el consumo energético, ya que un nodo activo tiene a lo más tres nodos activos en su vecindad. Sin embargo, estas reglas pueden causar estados en la red poco favorables, como nodos que se mantienen encendidos ininterrumpidamente, agotando rápidamente su batería, o nodos que se mantienen en estado inactivo que no permiten retransmitir mensajes, provocando fallas en los protocolos de encaminamiento.

El modelo basado en las reglas del juego de la vida no brinda a las celdas (nodos) la capacidad de transmitir paquetes a un nodo *sink*. Por lo que, para general un modelo válido de

Pseudocódigo 1 Algoritmo del juego de la vida

Entrada: celda_actual**Salida:** null*Revisa la vecindad de la celda para aplicar las reglas :*

```
1: vecinos_activos = 0
2: for celda in celda_actual.vecindad do
3:   if celda.estado == activo then
4:     vecinos_activos += 1
5:   end if
6: end for
7: if celda_actual.estado == activo then
8:   if vecinos_activos == 2 o 3 then
9:     celda_actual.estado_siguiete = activo
10:  else
11:    celda_actual.estado_siguiete = no_activo
12:  end if
13: end if
14: if celda_actual.estado == no_activo then
15:   if vecinos_activos == 3 then
16:     celda_actual.estado_siguiete = activo
17:   else
18:     celda_actual.estado_siguiete = no_activo
19:   end if
20: end if
21: return null
```

una WSN usando el juego de la vida es necesario incorporar un esquema para la transmisión de paquetes. Por ejemplo, en esta tesis se tomó el algoritmo A* o *A-star*. Este algoritmo es comúnmente utilizado para encontrar rutas óptimas en caminos con obstáculos para robots autónomos, así como en técnicas de planificación de rutas basadas en autómatas celulares [24, 25, 26, 27, 28], sin embargo, se pueden usar otro tipo de esquemas de encaminamiento. Como se mencionó, mejorar la vida útil de la red es una de una de las principales preocupaciones en el diseño de las WSN debido a que los sensores usualmente están equipados con pequeñas baterías que no son fáciles de recargar o sustituir, por lo que implementar protocolos de consumo energético eficiente se ha convertido en un tema de alta prioridad. Por ejemplo, se han propuesto estados iniciales del juego de la vida que logran patrones de larga duración sin presentar estados estáticos, uno de los más conocidos es el autómata celular llamado *symmetric die-hard* [32], el cual puede permanecer con actividad durante 1638 ciclos iniciando en un patrón de 32×32 celdas. La figura 3.10 muestra este CA en diferentes etapas de su desarrollo, representando a las celdas activas en negro y a las celdas inactivas en blanco. Aunque este modelo puede extender la vida en una malla de dos dimensiones, los patrones creados no se adaptan a las necesidades de una WSN debido a que un gran número de sensores permanecen apagados durante mucho tiempo, lo que puede provocar una gran cantidad de pérdida de información. Algunos ejemplos de estos patrones pueden verse a partir de la figura 3.10b hasta la figura 3.10f, donde existen muchos espacios (celdas inactivas) que afectarían a los protocolos de encaminamiento, ya que estos podrían no funcionar de manera apropiada debido a que no existen nodos cercanos entre sí que permitan el reenvío de paquetes. Debido a estos problemas, en esta tesis se proponen variaciones a las reglas originales del juego de la vida para permitir a las celdas adaptarse a las necesidades de una WSN, de tal forma que se asegure un consumo eficiente de la energía, y al mismo tiempo se asegure el envío de paquetes.

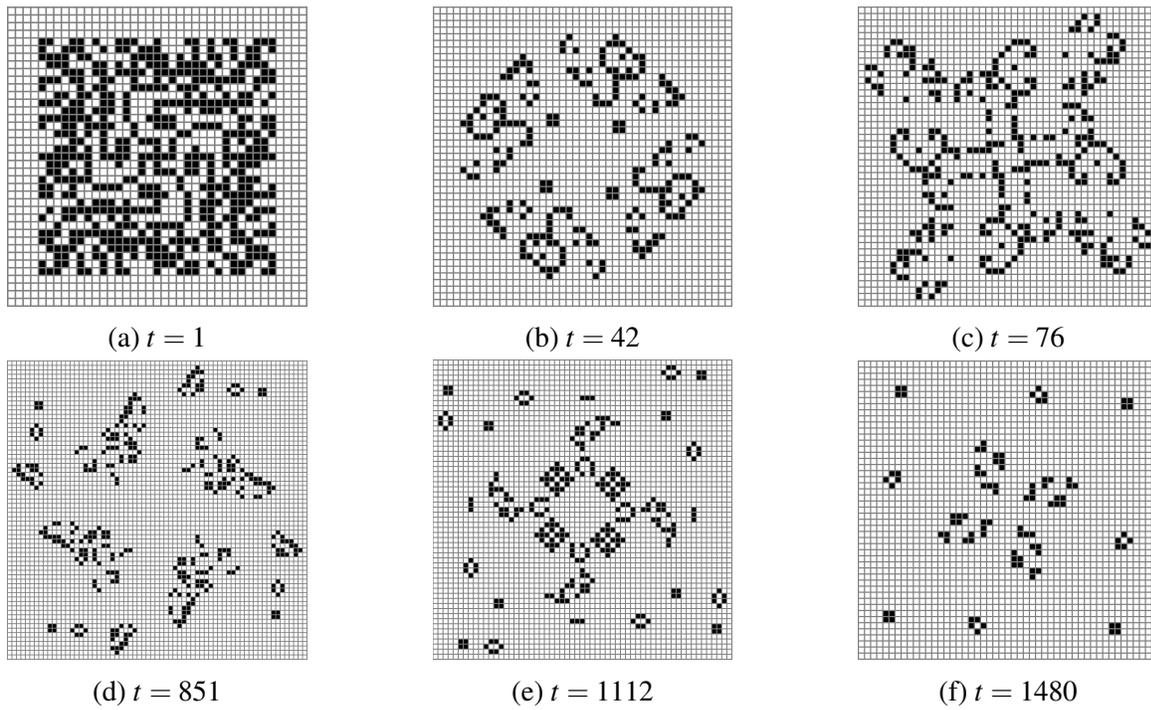


Figura 3.10: CA con patrón: *Symmetric die hard*.

Capítulo 4

Game of Sensors

En esta sección se presenta un nuevo conjunto de reglas de transición con la finalidad de extender la vida útil de la red y a la vez que sean capaz de crear un consumo energético homogéneo entre los nodos. Dado que se busca modelar una red inalámbrica de sensores, debe implementarse una forma de transmitir y retransmitir paquetes de información, por ello se adoptó el algoritmo A^* , el cual nos permite obtener rutas óptimas entre dos puntos en caminos que tienen obstáculos. A partir de este punto, al conjunto de reglas añadidas y la implementación del algoritmo *A-star* se le va a llamar *Game of Sensors* (GoS), el cual se resume en el pseudocódigo 2, mientras cada una de las partes que lo componen es detallada en el resto del capítulo. Finalmente, se presenta una evaluación del nuevo conjunto de reglas para medir la efectividad del modelo propuesto comparado con el modelo del juego de la vida. Posteriormente, en el capítulo 5 se harán comparaciones con los trabajos más recientes.

Pseudocódigo 2 *Game of Sensors*

Input: red**Output:** null*Ejecuta las reglas propuestas :*

- 1: Establecer el escenario inicial
 - 2: **while** Todos las celdas en la red estén vivas **do**
 - 3: **if** Si no hay un paquete en la red **then**
 - 4: Genera un paquete en una celda aleatoria
 - 5: **end if**
 - 6: **for** celda **in** red **do**
 - 7: Algoritmo del juego de la vida (celda)
 - 8: Algoritmo de calendarización (celda)
 - 9: Algoritmo de regla del umbral (celda)
 - 10: Algoritmo de encaminamiento (celda)
 - 11: **end for**
 - 12: **for** celda **in** red **do**
 - 13: celda.estado = celda.siguiete_estado
 - 14: **end for**
 - 15: **end while**
 - 16: **return** null
-

4.1. Reglas de transición

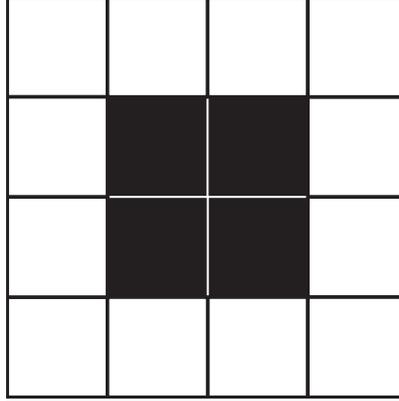


Figura 4.1: Patrón estático bajo las reglas del juego de la vida.

La primera regla tiene como finalidad de evitar estados estáticos de la red, es decir, que se reproduzcan patrones como el mostrado en la figura 4.1, en la cual se representan a los nodos activos en negro, mientras que los nodos en reposo se presentan en blanco. En esta figura se puede observar que las reglas del juego de la vida pueden mantener el estado de las celdas, lo que provocaría que algunos nodos de la red estén activos y por lo tanto consumiendo su batería, mientras que otros nodos permanecen en reposo sin enviar información.

La regla propuesta se expresa como:

$$\omega_{i,j}(t) = \begin{cases} 1, & \sum_{z=1}^{RT} \omega_{i,j}(t-z) = 0 \\ 0, & \sum_{z=1}^{RT} \omega_{i,j}(t-z) = RT, \end{cases} \quad (4.1)$$

donde RT (esquema de calendarización) es un periodo que se establece de acuerdo a las necesidades de cobertura y de vida útil en la red, este periodo expresa el número de pasos de tiempo que un sensor permanece en el mismo estado. El funcionamiento de esta regla consiste en que si un nodo ha permanecido en estado activo durante RT pasos de tiempo $\left(\sum_{z=1}^{RT} \omega_{i,j}(t-z) = RT\right)$, deberá de cambiar su estado a reposo ($\omega_{i,j}(t) = 0$) en el siguiente paso de tiempo. Por el contrario, si un nodo ha permanecido en reposo durante RT pasos

de tiempo $\left(\sum_{z=1}^{RT} \omega_{i,j}(t-z) = 0\right)$, en el siguiente paso de tiempo cambiará a estado activo ($\omega_{i,j}(t) = 1$). En ambos casos el estado de la celda cambiará sin importar el estado de las celdas dentro de su vecindad. Es importante destacar que esta regla se aplica exclusivamente cuando una celda ha permanecido en el mismo estado durante RT pasos de tiempo, en cualquier otro caso se aplican las reglas del juego de la vida tradicionales. A partir de este punto, esta regla va a ser identificada como la regla de calendarización. En el pseudocódigo 3 se muestra la implementación de esta regla:

Pseudocódigo 3 Algoritmo de calendarización

Entrada: celda

Salida: null

Verifica la actividad de cada celda para aplicar la regla de calendarización :

```

1: if celda.estado == activo then
2:   celda.bateria -= 1
3:   celda.tiempo_activo += 1
4:   celda.tiempo_no_activo = 0
5: else
6:   celda.tiempo_activo = 0
7:   celda.tiempo_no_activo += 1
8: end if
9: if celda.tiempo_activo  $\geq RT$  then
10:  celda.estado_siguiete = no_activo
11: end if
12: if celda.tiempo_no_activo  $\geq RT$  then
13:  celda.estado_siguiete = activo
14: end if
15: return null

```

Las figuras 4.2a y 4.2b muestran un conjunto de celdas siguiendo el mismo patrón desde un tiempo $t = 1$ hasta un tiempo $t = 10$, donde se considera el periodo $RT = 10$. Se puede ver en la figura 4.2c cómo el patrón se rompe debido a la implementación de la regla de calendarización.

En este trabajo se usa el valor de $RT = 10$ debido a los buenos resultados mostrados trabajos previos [33].

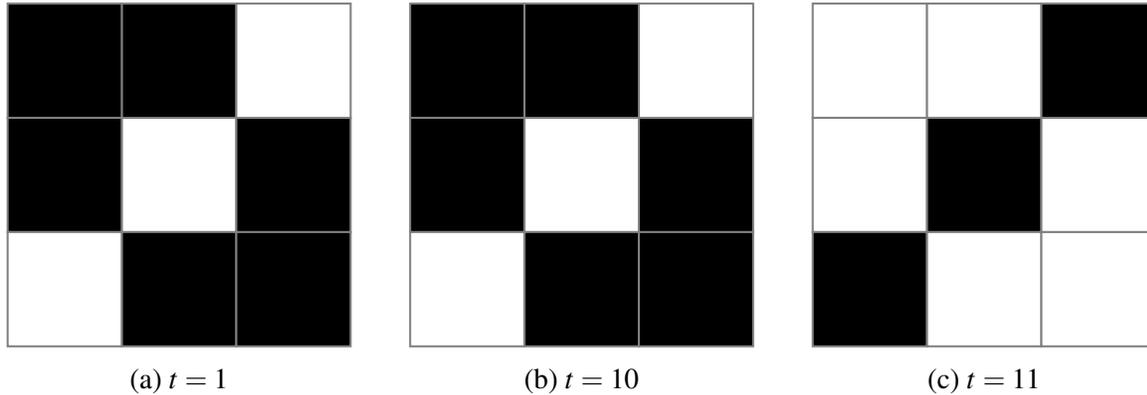


Figura 4.2: Regla de calendarización.

El agotamiento de las baterías es otro de los temas más importantes en WSN, por lo que los nodos con mayor carga restante de batería deberían permanecer activos con mayor probabilidad que aquellos nodos que tienen niveles más bajos de carga, esto con la finalidad de equilibrar el consumo energético en la red. Es por eso que se propone otra regla, la cual se aplica después de todas las reglas anteriormente propuestas y se expresa como:

$$\omega_{i,j}(t) = 1, \quad \text{si } \omega_{i,j}(t) = 1 \text{ (de acuerdo a las reglas anteriores) con probabilidad } p \quad (4.3)$$

Esta regla consiste en que una vez que los nodos alcanzan cierto umbral de carga en sus baterías, el cambio de estado de reposo a estado activo se hará solo con una determinada probabilidad p . Es decir, si un nodo ha estado inactivo durante RT pasos de tiempo, cambiará su estado a activo con una probabilidad p o permanecerá en su mismo estado con una probabilidad $1 - p$. La ecuación 4.3 solo es válida si el estado actual de la batería (e) de un sensor se encuentra debajo de un umbral (th), de otra forma esta regla no se aplica. El resto de reglas se aplican sin importar el nivel de batería de los sensores.

En el pseudocódigo 4 se muestra la implementación de esta regla.

En este trabajo se toma el valor $p = 0.6$, debido a los buenos resultados mostrados en trabajos previos [33], y por la misma razón el valor de th se estableció en 50% de la capacidad total de la batería e . De la misma forma, en la figura 4.3 se muestra cómo se aplicaría esta regla para el caso mostrado anteriormente en la figura 4.2.

Pseudocódigo 4 Algoritmo de la regla del umbral**Entrada:** celda**Salida:** null*Verifica la batería de las celdas para aplicar la regla :*

```

1: if celda.estado_siguiete == activo then
2:   if celda.batería < th then
3:     r = porcentaje aleatorio entre 0% a 100%
4:     if r > p then
5:       celda.estado_siguiete = no_activo
6:     end if
7:   end if
8: end if
9: return null

```

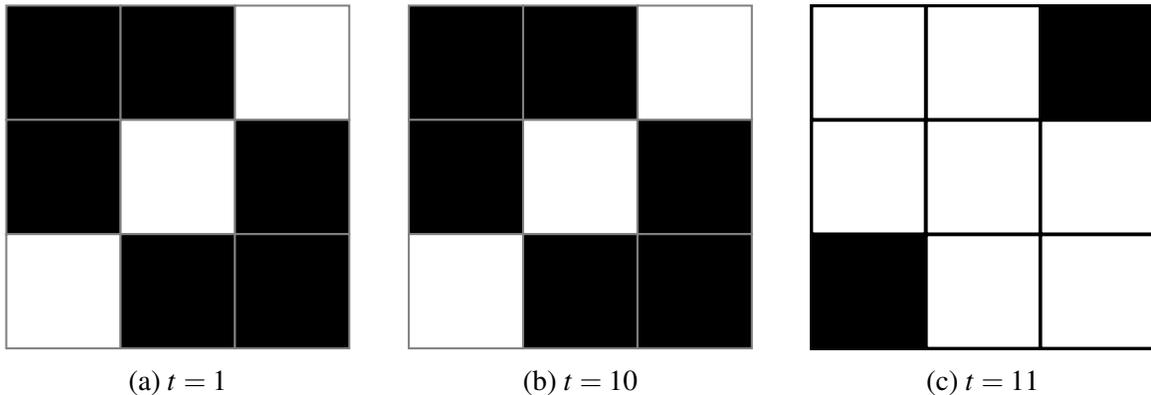


Figura 4.3: Regla de probabilidad, donde los sensores que deben pasar a estado activo respecto a las reglas anteriores realizan el cambio de estado solo con una probabilidad p , por lo que en este caso el sensor central no realiza el cambio de estado.

4.2. Algoritmo A-star

Como se mencionó, las reglas del juego de la vida no cumplen con la función de transmitir paquetes de información, por esto, se implementa el algoritmo *A-star* para realizar el proceso de encaminamiento [34]. *A-star* es un algoritmo heurístico de encaminamiento ampliamente conocido por su capacidad de crear rutas óptimas desde un punto fuente hasta un punto destino. Para ello, el algoritmo usa la ecuación 4.4 para determinar el siguiente

paso a tomar con el fin de alcanzar su destino en el menor número de pasos:

$$d(n) = g(n) + h(n), \quad (4.4)$$

donde $g(n)$ representa la distancia (costo) desde la ubicación del nodo fuente hasta la ubicación del nodo actual (n), mientras $h(n)$ es una función heurística que estima el costo desde el nodo actual hasta la ubicación del nodo destino (el nodo *sink*), finalmente $d(n)$ representa un peso aproximado para cada uno de las posibles rutas que pueda tomarse, por lo que el algoritmo elige aquella ruta en la que el valor de $d(n)$ sea el menor, repitiendo este proceso hasta completar la ruta.

En la figura 4.4a se muestra un escenario con un obstáculo en el centro de la cuadrícula. El nodo que desea transmitir un paquete o nodo fuente se representa como un círculo negro, mientras que el nodo destino o nodo *sink* se representa como un círculo blanco. En la figura 4.4b se muestra el funcionamiento del algoritmo A-star determinando una de las posibles rutas óptimas para llegar al destino, las flechas representan la dirección de transmisión del paquete y los círculos negros de menor tamaño representan a los nodos que servirán como transmisores cuando el paquete sea enviado.

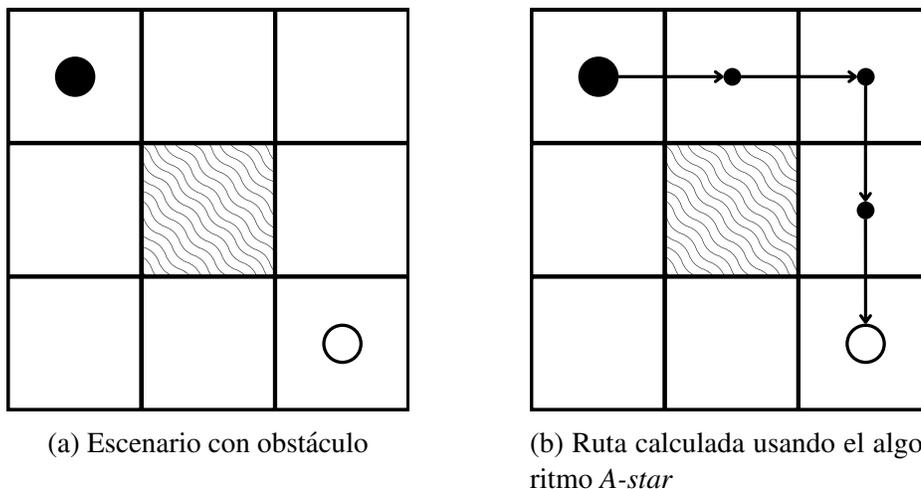


Figura 4.4: Funcionamiento del algoritmo A-star en un escenario con obstáculo.

El algoritmo *A-star* requiere de información previa (memoria) para estimar la mejor opción para enviar un paquete. Sin embargo, esto no puede ser trasladado a GoS directamente, esto se debe a que cada celda activa no puede estimar cuáles de sus vecinos se encontrarán activos en el siguiente paso de tiempo. Para resolver este problema, se propone un nuevo método en el que la ecuación 4.4 se usa de la siguiente forma: la componente $g(n)$ es obtenida al calcular la distancia Euclidiana desde la ubicación del nodo n hasta la ubicación del

nodo actual m . Por otro lado, $h(n)$ se obtiene al estimar una distancia desde el actual nodo m hasta el nodo destino o nodo *sink*.

El algoritmo *A-star* original forma rutas completas desde el origen hasta el destino; sin embargo, dado que el estado de la red cambia en cada paso de tiempo, las rutas calculadas en el tiempo t pueden dejar de ser válidas en el tiempo $t + 1$. Es por lo anterior que en este trabajo se propone una versión modificada de este algoritmo que permite calcular rutas óptimas locales (a un salto) en cada paso de tiempo. A continuación se describe dicho algoritmo:

Si un nodo desea transmitir un paquete:

- Calcula los pesos d de su vecindad a un salto.
- Determina cuáles de sus vecinos activos se encuentran activos.
- Determina cuáles de sus vecinos tiene menor peso d .
- Transmite el paquete al vecino activo con menor peso d .
- Si se transmitió el paquete al nodo *sink* el algoritmo termina, en caso contrario se repite el procedimiento.

El algoritmo *A-star* original tiene complejidad exponencial, mientras que la modificación propuesta tiene complejidad igual al tamaño de la vecindad de las celdas, dado que cada que se busca transmitir un paquete basta con verificar el estado de las celdas dentro de la vecindad.

En la figura 4.5 se muestra el funcionamiento del algoritmo *A-star* en el modelo GoS. Se puede observar que la ruta se calcula dinámicamente en cada paso de tiempo, anticipando de esta forma la aparición de nuevos obstáculos o de nuevas rutas de menor costo para llegar al nodo destino.

Para que sea posible la obtención de los datos necesarios para el algoritmo *A-star*, es necesario que cada nodo conozca su propia ubicación, así como la ubicación de sus vecinos y la ubicación del nodo *sink*. En este trabajo se usa la distancia Euclidiana por simplicidad, pero se puede usar cualquier otro tipo de estimador como la distancia *Manhattan*, la distancia *Minkowski* o el estimador usado en la referencia [7], de acuerdo a las necesidades del modelo.

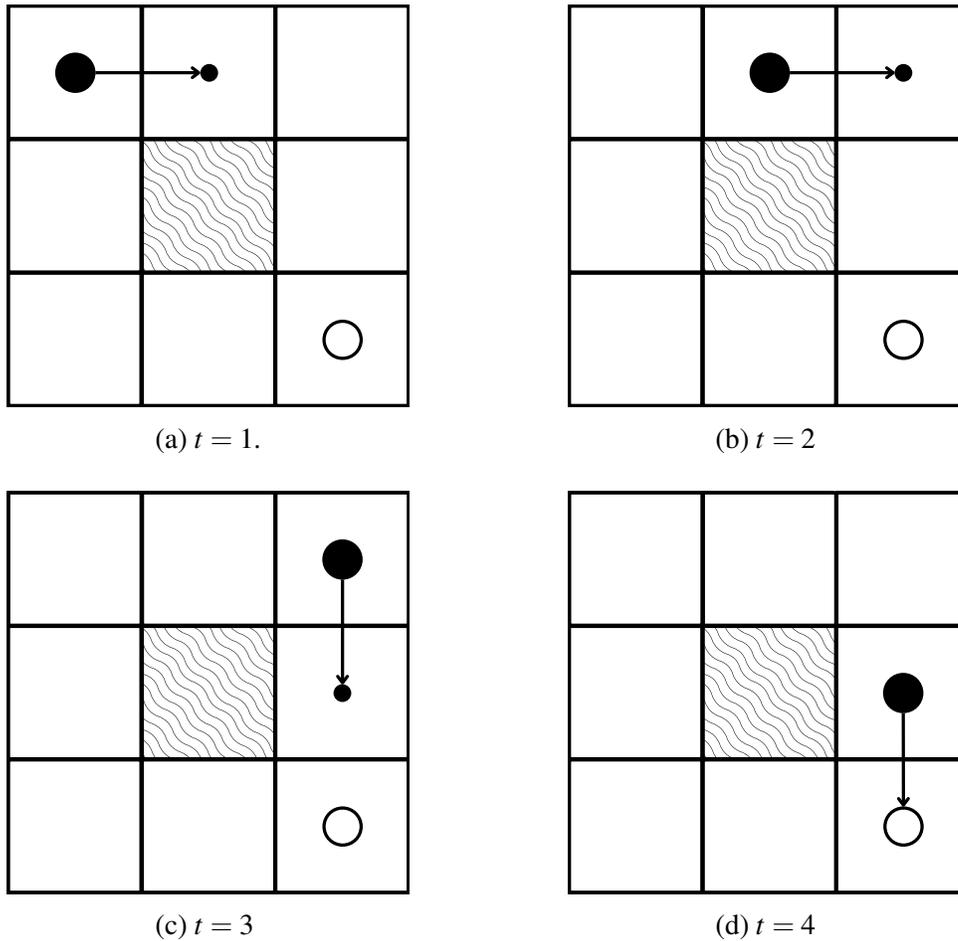


Figura 4.5: Funcionamiento del algoritmo *A-star* en el modelo GoS, donde la ruta óptima se calcula localmente en cada paso de tiempo.

Dado que en el modelo propuesto, el escenario cambia con cada paso de tiempo, el nodo que desea transmitir un paquete debe ejecutar el algoritmo para determinar cuál de sus vecinos activos es la mejor opción para retransmitir el paquete, el nodo con menor costo recibirá el paquete y se repetirá el proceso hasta llegar al nodo *sink*. En caso de que ninguno de los vecinos activos del nodo n en el tiempo actual tenga menor costo que el mismo nodo n el paquete no se retransmitirá, sino que el nodo n esperará al siguiente tiempo para encontrar una mejor ruta. Para asegurar esto, el nodo que desea transmitir debe permanecer en estado activo hasta que logre transmitir el paquete. Esto se expresa en la

siguiente regla, la cual es añade al modelo:

$$\omega_{i,j}(t) = \begin{cases} 1, & \omega_{i,j}(t-1) = 1, h(n) > d(n, DST) \\ 0, & \omega_{i,j}(t-1) = 1, h(n) < d(n, DST), \end{cases} \quad (4.5)$$

$$(4.6)$$

donde $d(n, DST)$ representa la distancia Euclidiana desde el nodo actual n hasta el nodo destino. Esta regla asegura que la distancia hasta el nodo destino disminuye al aumentar los pasos de tiempo, es decir, que el paquete nunca se va a alejar del nodo destino en busca de otras rutas.

Aún cuando un nodo ha transmitido un paquete de manera exitosa, no se puede reusar la ruta previa, debido a que dependiendo del estado de la red en el paso de tiempo actual puede encontrarse una nueva ruta óptima.

En el pseudocódigo 5 se muestra la implementación de este algoritmo de encaminamiento.

Pseudocódigo 5 Algoritmo de encaminamiento basado en A*

Entrada: celda_actual

Salida: celda_siguiente

Verifica la vecindad del nodo transmisor para elegir el siguiente salto :

```

1: distancia_actual = distancia desde celda_actual al sink
2: for celda in celda_actual.vecindad do
3:   if celda.estado == activo then
4:     celda.distancia = distancia desde la celda al sink
5:     if celda.distancia < distancia_actual then
6:       distancia_actual = celda.distancia
7:       celda_siguiente = celda
8:     end if
9:   end if
10: end for
11: if celda_siguiente == sink then
12:   Paquete recibido
13: end if
14: return celda_siguiente

```

Para comprobar el funcionamiento del algoritmo de encaminamiento dinámico *A-star* en autómatas celulares propuesto en esta tesis, se realizó un experimento que consiste en generar tráfico de información en forma de paquetes en los nodos que componen la periferia de la red, estos paquetes son transmitidos con la finalidad de llegar al nodo *sink*, el cual se

ubica en el centro de la malla. En la figura 4.6 se puede ver a los nodos generadores de paquetes en gris, mientras que el nodo *sink* se muestra en negro.

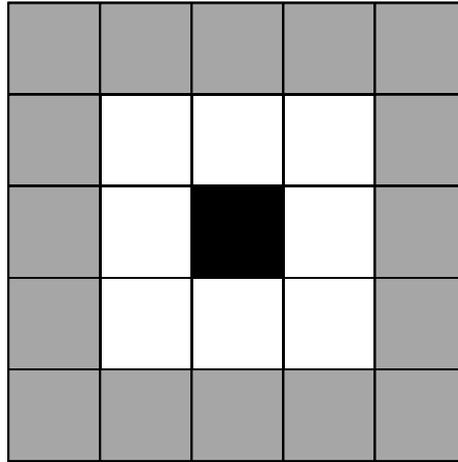


Figura 4.6: Nodos generadores de paquetes y nodo *sink*.

El escenario de simulación consiste en una red de 30×30 nodos, por lo que en una transmisión ideal con todos los nodos de la red activos el paquete realizaría máximo 15 saltos para llegar al nodo *sink* desde la periferia, dado que el nodo sumidero se encuentra en el centro de la red. En el escenario planteado para el experimento cada nodo cuenta con la batería suficiente para estar activo durante cien pasos de tiempo, cuando un nodo desea transmitir un paquete debe realizar los cálculos del algoritmo *A-star* para determinar la mejor ruta posible hacia el nodo *sink*. Si un nodo transmisor no cuenta con una ruta que mejore el recorrido del paquete, esperará sin transmitir la información hasta lograr una ruta óptima. Los paquetes se generan en la periferia de la red de forma aleatoria, mientras que el resto de nodos sigue las reglas del modelo GoS. Se considera que un nodo puede transmitir su paquete a otro nodo siempre que el receptor se encuentre activo y dentro de la vecindad de *Moore* de radio uno del nodo transmisor. Finalmente, el experimento se repitió cien veces para obtener resultados promedios.

En la tabla 4.1 se pueden ver los resultados obtenidos de este experimento, es importante destacar que si bien el algoritmo *A-star* usado en el modelo GoS tarda el doble de unidades de tiempo en entregar un paquete al nodo *sink* en promedio, también entrega en promedio casi el doble de paquetes en comparación con el algoritmo *A-star* puro.

Tabla 4.1: Comparación de desempeño del algoritmo A-star bajo el modelo de GoL.

Parámetro	GoS	A-star puro
Llegada a destino [Pasos de tiempo]	30	15
Desviación estándar [Pasos de tiempo]	5	0
Paquetes enviados	11	6

4.3. Análisis de desempeño

Con la finalidad de evaluar el impacto de las nuevas reglas de transición se presenta una comparación del modelo GoS con el modelo basado en el CA del juego de la vida. Para esta evaluación no se consideró el proceso de encaminamiento, aunque en el capítulo 5 se presentarán resultados considerando el modelo completo. Con este propósito se usó un simulador personalizado escrito en Python, la región a estudiar corresponde a una rejilla bidimensional cuadrada de 30×30 celdas. En la figura 4.7 se muestra uno de los posibles escenarios iniciales de la simulación, en los cuales siempre se debe cumplir que el 50% de los nodos (elegidos de forma aleatoria) se encuentren activos al iniciar la simulación, en este caso se representan los cuadros blancos como nodos en reposo y los cuadros oscuros como nodos activos.

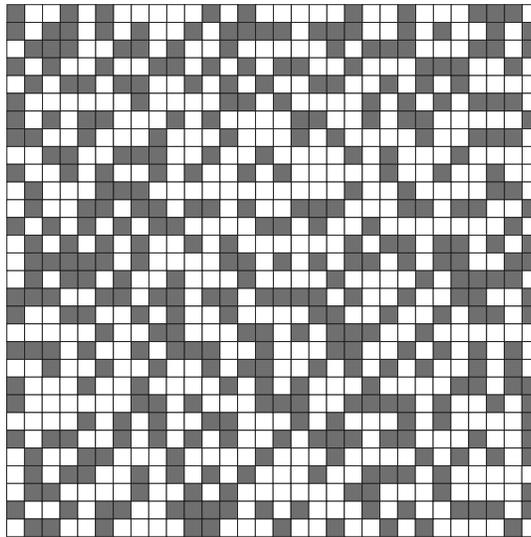


Figura 4.7: Posible escenario inicial.

Con motivo de simplificar el análisis, cada celda (nodo) cuenta con una batería de 100

unidades de carga, cada paso de tiempo en el que un nodo se encuentre activo consumirá una carga de su batería, en este caso cada paso de tiempo corresponde a un segundo. En el capítulo 5 se usa un modelo de energía más realista con la finalidad de comparar el método usado en GoS con algunos de los trabajos relacionados más importantes. El umbral de energía th se establece en un valor de 50%, mientras que la probabilidad de cambio se establece como $p = 0.6$. El impacto de la elección del valor de p se detallará más adelante. Cada métrica considerada en la simulación fue obtenida como promedio de 100 ejecuciones, debido a que como se muestra en la figura 4.8 en este valor se asegura la convergencia de los resultados.

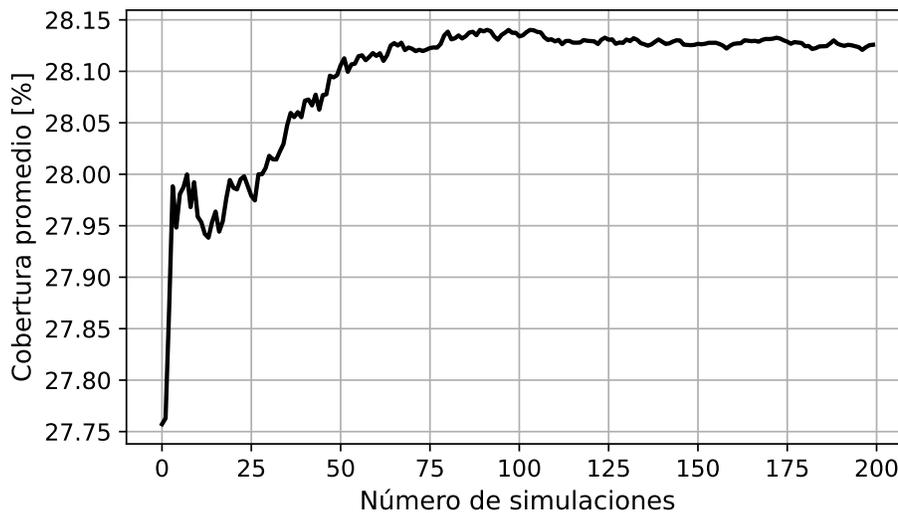
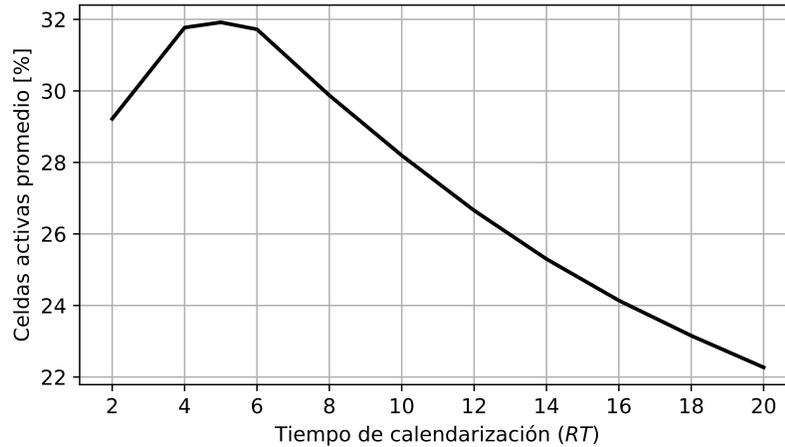


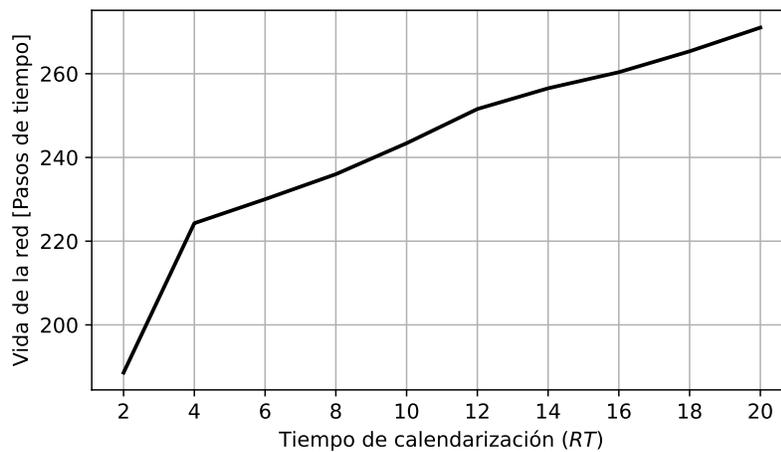
Figura 4.8: Convergencia de resultados.

Las variaciones en la duración de la calendarización (RT) implica cambios en el número de nodos activos. La figura 4.9a muestra cómo RT cambia el número promedio de nodos encendidos en la red. Por ejemplo, cuando RT es igual a 4, la cantidad de nodos activos en promedio es del 31.8% de toda la red, por lo que el 68.2% de nodos restante se encuentra en estado de reposo. Sin embargo, cuando RT es igual a 20, la cantidad de nodos activos en promedio es 22.3%. Esto se debe a que cuando el tiempo de calendarización RT es pequeño los nodos pasarán del reposo a estado activo con mayor frecuencia, aumentando así el porcentaje de nodos activos promedio, mientras que cuando el valor de RT es muy grande, los nodos permanecen en reposo por mayor tiempo, provocando así un menor porcentaje de nodos activos promedio. Por otro lado, la figura 4.9b muestra cómo el valor de RT impacta en la vida de la red, por ejemplo cuando RT es igual a 2, la vida de la red medida en pasos de tiempo es de 189, mientras que para un valor de RT igual a 20, la vida de la red incre-

menta a 271 pasos de tiempo, incrementando de esta forma la vida de la red en un 30%. De esta forma, al variar el valor de RT , la vida de la red cambia, así como la cantidad de nodos activos pueden aumentar o disminuir gradualmente en función del valor asignado a RT . Es importante mencionar que cada simulación termina cuando algún nodo se queda sin carga en su batería. Para esta tesis, el valor de RT se estableció en 10, dado que este valor crea un punto medio entre la vida de la red y el número de nodos activos que permiten garantizar el correcto funcionamiento del protocolo de encaminamiento propuesto.



(a) Promedio de celdas activas respecto al tiempo de calendarización RT .



(b) Vida de la red respecto al tiempo de calendarización RT .

Figura 4.9: Impacto de la variación del tiempo de calendarización RT en el promedio de celdas activas y la vida de la red.

Una vez que todos los parámetros fueron establecidos ($th = 50\%$, $p = 0.6$ y $RT = 10$), la figura 4.10 muestra el número de nodos activos a lo largo del tiempo. Como se mencionó previamente, al inicio de la simulación el 50% de los nodos se encuentra activos. Posteriormente, existe un breve periodo de oscilación hasta que el modelo converge en 28% de celdas activas. Esta figura también muestra que el consumo de energía dentro de la red es homogéneo, ya que al llegar a aproximadamente 300 pasos de tiempo, la cantidad de

nodos activos en promedio decae rápidamente. Por ejemplo, en el tiempo 350 (ver figura 4.10), solo el 11% de los nodos activos puede ser usado para establecer un protocolo de encaminamiento. Esta situación puede resultar en pérdidas de información debido a que hay muy pocos nodos que puedan retransmitir los paquetes. Sin embargo, el número de nodos activos puede incrementarse al reducir el valor de RT , a costa de reducir la vida útil de la red.

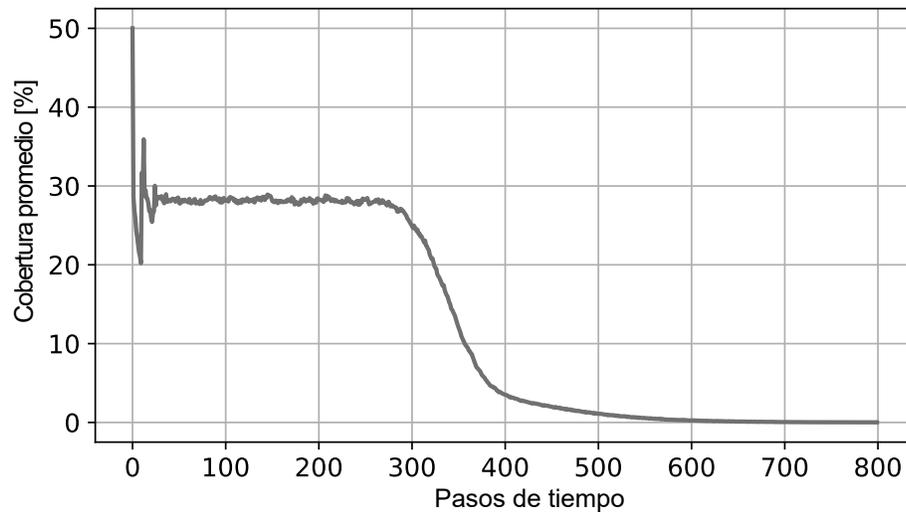


Figura 4.10: Promedio de nodos activos respecto a la vida de la red.

En la figura 4.11a se muestra el estado de la red cuando el primer nodo se queda sin batería, este nodo se identifica con el símbolo (X) en la esquina inferior derecha. Los cuadros negros representan nodos activos, mientras los blancos representan los nodos en reposo. La figura 4.11b muestra el estado de la red cuando el 20% de los nodos se quedó sin batería (ver nodos grises).

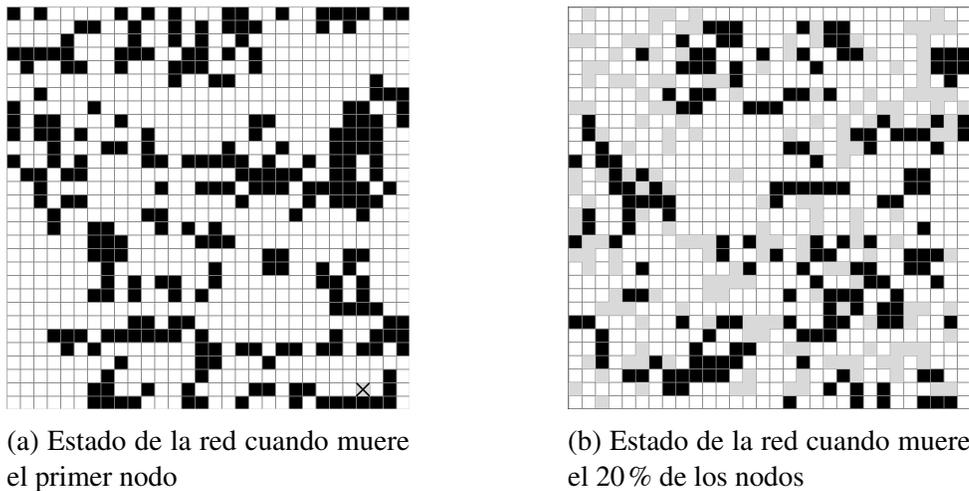


Figura 4.11: Estado de la red en distintos tiempos.

Uno de los parámetros más relevantes a considerar en este trabajo es el número de vecinos activos con los que cuenta cada nodo, esto con la finalidad de evaluar la posibilidad de implementar un esquema de encaminamiento que puede operar de forma correcta. La figura 4.12 muestra el número promedio de vecinos activos a partir de que el 1 % de los nodos se quedan sin batería. Se puede observar que para la vecindad a un salto (nodos con los que se puede tener comunicación directa), existen al menos dos puntos de reenvío para cada nodo hasta que el 20 % de los nodos se queda sin batería, además de tener al menos seis puntos de reenvío de paquetes por cada nodo en su vecindad a dos saltos (nodos a los que se llega a través de los nodos en la vecindad a un salto). En esta figura también se permite ver que después de que el 20 % de los nodos se quedan sin batería, el resto de nodos comienza a agotar su batería rápidamente.

La figura 4.14 muestra el impacto del valor elegido de p en el desempeño del modelo. Se puede ver que mientras el valor de p aumenta, la vida de la red disminuye. Por ejemplo, para $p = 0.4$ los nodos de la red agotan sus baterías en 335 pasos de tiempo, mientras que para $p = 0.8$ las baterías se agotan en 307 pasos de tiempo. De la misma forma se puede ver que en el escenario en el que $p = 0.8$, una vez que el primer sensor agota su batería (250), el resto de los nodos agota su batería más rápidamente en comparación al escenario donde $p = 0.4$. Esto ocurre debido a que en el escenario con $p = 0.4$ las reglas obligan a la celdas a cambiar al estado activo con mucha menos probabilidad, provocando que su batería se conserve por más tiempo. En el modelo de GoS se seleccionó un valor de $p = 0.6$ debido a que nos permite mantener un equilibrio entre la cantidad de nodos activos (cobertura) y la vida de la red.

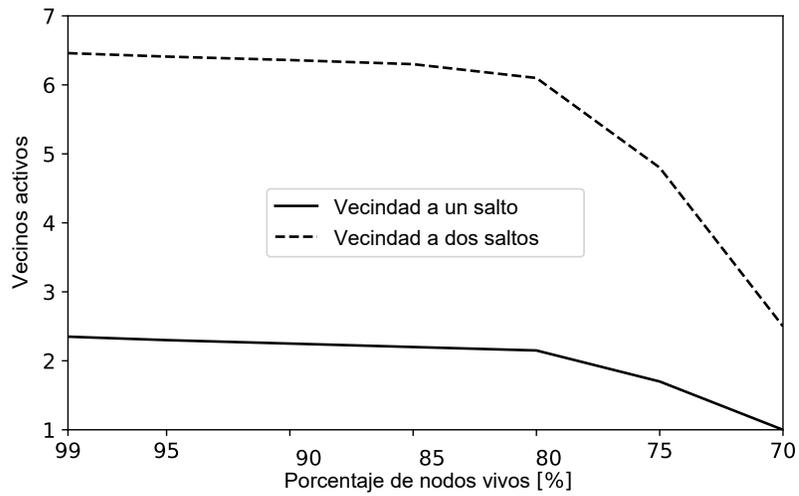
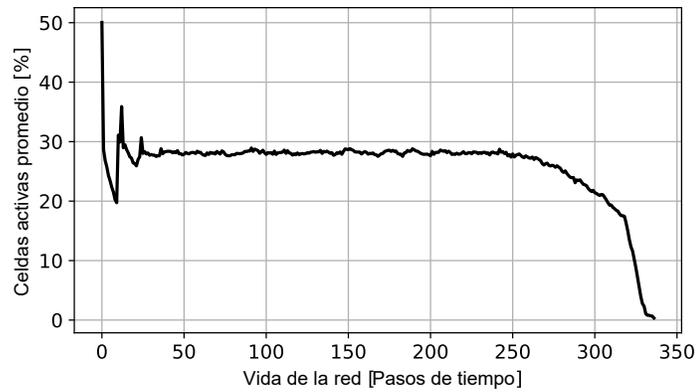
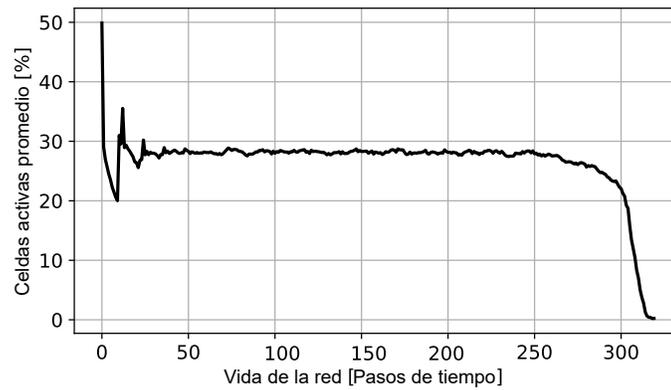


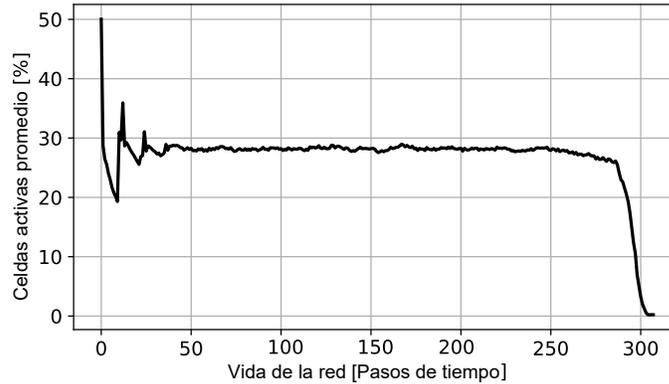
Figura 4.12: Vecinos activos respecto al porcentaje de nodos en la red con batería.



(a) $p = 0.4$.



(b) $p = 0.6$.

(a) $p = 0.8$.Figura 4.14: Impacto de la variación del parámetro p .

Ambos modelos (GoL y GoS) fueron evaluados bajo las mismas condiciones iniciales. Los resultados fueron promediados mediante 100 simulaciones (ver figura 4.8). La tabla 4.2 muestra la vida útil promedio de la red, el número promedio de celdas activas y el promedio de vecinos activos de cada nodo para el modelo basado en GoL.

Tabla 4.2: Desempeño del CA GoL.

GoL	Promedio	Desviación estándar
Vida de la red	283.18	95.24
Nodos activos	6.97	1.22
Vecindad a un salto	0.54	0.09
Vecindad a dos saltos	1.63	0.29

Por otro lado, en la tabla 4.3 podemos ver los resultados obtenidos para el modelo propuesto (GoS). Ambas tablas muestran los resultados obtenidos hasta que el primer nodo de la red agotó su batería.

Tabla 4.3: Desempeño del modelo GoS.

Game of Sensors	Promedio	Desviación estándar
Vida de la red	241.6	8.93
Nodos activos	28.2	0.3
Vecindad a un salto	2.2	0.02
Vecindad a dos saltos	6.5	0.07

Al comparar ambas tablas se puede ver que la desviación estándar en el campo vida de la red es 10.66 veces más grande en el modelo GoL comparado con el modelo propuesto. Este resultado muestra que GoL no es capaz de mantener la cantidad de nodos activos, tal como sí sucede en GoS. Este efecto puede verse en la figura 4.10, en el que los nodos activos permanecen constantes hasta que la batería de los mismos comienza a agotarse. Además, el modelo propuesto mantiene activos en promedio 28.2% de los nodos, mientras que GoL mantiene activo un promedio de 6.97% de los nodos en toda la red. Adicionalmente, el modelo propuesto cuenta en promedio con dos vecinos activos a un salto, lo que garantiza que en todo momento un nodo activo podrá comunicarse con al menos otros dos nodos, mientras que en el modelo basado en GoL, no se logra tener un vecino activo dentro de la vecindad a un salto (0.54). Este resultado tiene un fuerte impacto al considerar la implementación de un protocolo de encaminamiento debido a que el modelo GoL no tendrá nodos que cumplan el papel de reenviar los paquetes la mayoría del tiempo. De la misma forma se puede ver que para el modelo GoS se cuenta con 6.5 vecinos en la vecindad a dos saltos.

Finalmente, en la tabla 4.4 se muestran los resultados obtenidos con el modelo propuesto ampliando el tiempo de simulación hasta que el 20% de los nodos agota su batería. Se puede observar que el modelo propuesto mantiene al menos a dos nodos activos para retransmitir paquetes, lo que permitirá a los protocolos de encaminamiento operar de manera correcta.

Tabla 4.4: Desempeño del modelo GoS cuando 20% de los nodos han agotado su batería.

GoS	Promedio	Desviación estándar
Vida de la red	322.2	3.64
Nodos activos	27.58	0.29
Vecindad a un salto	2.01	0.02
Vecindad a dos saltos	6.08	0.06

Capítulo 5

Experimentos y resultados

En esta capítulo se presenta una evaluación del desempeño del modelo GoS. Para esto, las simulaciones se dividieron en tres grupos; el primer grupo es para realizar una comparación con los algoritmos EERP [13] y A&F [12]. El segundo grupo es para presentar una comparación con el algoritmo LPA-star [6]. Finalmente, el tercer grupo es para realizar una comparación con el algoritmo [5]. De ahora en adelante, este algoritmo será identificado como “Maximizing”. Es importante mencionar que la propuesta de “Maximizing” muestra mejor desempeño que los trabajos propuestos en las referencias [23] y [20].

Para estos experimentos se usó un simulador personalizado escrito en Python, a continuación se muestran las suposiciones y las propiedades que presentan los grupos de simulaciones:

- Todos los nodos cuentan con la misma batería inicial, excepto el último experimento.
- Todos los nodos tienen el mismo rango de transmisión máximo.
- Cada nodo está sincronizado con los nodos dentro de su vecindad a un salto.
- Cada nodo conoce la ubicación del nodo *sink*, así como la ubicación de sus vecinos.
- Todos los nodos permanecen en la misma ubicación durante toda la simulación
- El modelo se desarrolla cada paso de tiempo.

Un paso de tiempo representa el tiempo requerido para desarrollar el sistema, este se define como:

$$\text{Paso de tiempo} = C + \textit{Hello} + \textit{Unicast} + \textit{Paquete},$$

donde C es un periodo de contención (capa MAC del modelo OSI) para enviar los paquetes *hello*. El periodo *Hello* representa el tiempo necesario para enviar los paquetes *hello*, este

tiempo es necesario para anunciar cuáles nodos están activos durante el tiempo actual. Es importante mencionar que durante el periodo *Hello* todos los nodos vecinos deben recibir los paquetes *hello* para actualizar su estado en el tiempo $t + 1$. Con base en lo mostrado en la figura 4.12 el número promedio de vecinos activos en la vecindad a un salto es aproximadamente dos para cada paso de tiempo, esto significa que en promedio solo dos nodos intentan ganar el canal durante el periodo de contención (*C*) y enviar paquetes *hello* en cada paso de tiempo. La tabla 4.3 confirma este dato con una desviación estándar de apenas 0.02. El periodo de *Unicast* se da cuando el nodo que busca enviar un paquete selecciona el mejor punto de retransmisión para llegar al *sink*, el nodo transmisor envía un mensaje a toda su vecindad con la finalidad de que los nodos en la misma vecindad sepan si el paquete a enviar es para ellos o no. Finalmente el periodo *Paquete* es el tiempo en el que se envía el paquete.

Es importante mencionar que los paquetes son enviados de forma unicast por medio de un protocolo RTS-CTS o técnicas similares, de forma que solo el nodo que debe recibir el paquete se encuentra activo mientras el resto de nodos en el rango de transmisión se desconectan hasta el inicio del siguiente paso de tiempo.

5.1. Modelo de energía

El modelo de consumo de energía de los radios en los nodos fue tomado de los trabajos [5] y [7], en los cuales la energía consumida por transmitir un bit es calculada como se muestra a continuación:

$$E_{Tx}(l, d) = \begin{cases} l \cdot E_{elec} + l \cdot \epsilon_{fs} \cdot d^2 & \text{if } d \leq d_0 \\ l \cdot E_{elec} + l \cdot \epsilon_{amp} \cdot d^4 & \text{if } d > d_0, \end{cases} \quad (5.1)$$

$$(5.2)$$

donde l representa el tamaño en bits del paquete, mientras que d es la distancia Euclidiana entre el transmisor y el receptor. E_{elec} representa la disipación de energía en el circuito transmisor y receptor por bit, ϵ_{fs} representa el factor de amplificación para el modelo de espacio libre (*path loss* igual a 2), mientras que ϵ_{amp} representa el factor de amplificación para el modelo *multi-path* (*path loss* igual a 4). Finalmente, $d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{amp}}}$.

La energía consumida al recibir un bit se calcula como se muestra a continuación:

$$E_{Rx}(l) = l \cdot E_{elec}. \quad (5.3)$$

Debido a que el modelo propuesto en los trabajos [23], [20] y [5] no consideran el consumo de energía al procesar información, se toma el modelo de consumo propuesto en

[9], completando el modelo de energía para transmisión de la siguiente forma:

$$E_{Tx}(l, d) = \begin{cases} l \cdot E_{elec} + l \cdot E_{DA} + l \cdot \epsilon_{fs} \cdot d^2 & \text{if } d \leq d_0 \\ l \cdot E_{elec} + l \cdot E_{DA} + l \cdot \epsilon_{amp} \cdot d^4 & \text{if } d > d_0, \end{cases} \quad (5.4)$$

donde E_{DA} representa el consumo de energía al procesar un bit de información, por otro lado el modelo de energía para recepción esta dado por:

$$E_{Rx}(l) = l \cdot E_{elec} + l \cdot E_{DA}. \quad (5.6)$$

5.2. GoS contra EERP

Para estos experimentos, la región estudiada corresponde a un cuadrado de $200\text{ m} \times 200\text{ m}$. Aunque GoS fue modelado para funcionar con una distribución regular de nodos (ver figura 4.7), también puede ser usado en distribuciones de nodos irregulares, tal como se muestra en la figura 5.1, en la cual por cuestiones ilustrativas se describe un área de $100\text{ m} \times 100\text{ m}$, con uno de los nodos resaltado mostrando su rango de transmisión. En este caso la vecindad a usar ya no será la de *Moore* de radio 1 con ocho vecinos tal como se mostró en la figura 3.4b, sino que la vecindad estará conformada con todos los nodos dentro del rango de transmisión.

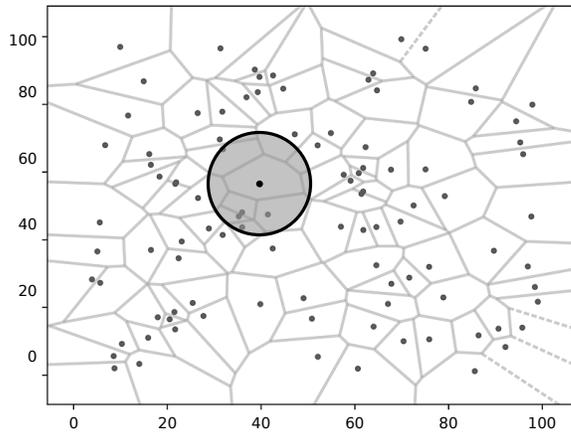


Figura 5.1: Escenario con nodos distribuidos aleatoriamente.

Al tratarse de una distribución de nodos aleatoria, también lo será la cantidad de nodos dentro de las vecindades, por lo cual las reglas del modelo deben modificarse. En el modelo

original se cumplen las reglas descritas en las ecuaciones 3.5-3.8, sin embargo, la distribución aleatoria de los nodos podrían provocar que existan vecindades con alta densidad de nodos o vecindades con baja densidad de nodos, lo que provocaría que las reglas nunca se cumplan y gran parte de la red permanezca en estado de reposo durante largos periodos de tiempo. Para resolver esto, las reglas se modifican de tal suerte que mantengan el porcentaje de nodos activos en la vecindad respecto a los nodos totales en la vecindad. Es decir, si en las reglas originales se necesitaban de 2 a 3 vecinos activos del total de 8 vecinos en la vecindad de *Moore* de radio 1 para que una regla se cumpliera, con estas nuevas reglas bastará con que se cumpla el porcentaje equivalente con respecto al número de vecinos, en el caso de dos a tres vecinos activos el porcentaje se traduciría como 25 % ($\frac{2}{8} = 0.25$) a 37.5 % ($\frac{3}{8} = 0.375$).

Un caso especial son las reglas que requieren de exactamente tres vecinos activos para ser ejecutadas, debido a que eso se traduce a un porcentaje específico de 37.5 % y en muchos casos este porcentaje específico en una vecindad con número de vecinos aleatoria resulta imposible, se adoptó un rango que va de 35 % a 40 %. Con esta modificación las reglas se expresan de la siguiente forma:

$$\omega_{i,j}(t) = \begin{cases} 1, & \omega_{i,j}(t-1) = 1, & \sum_{(k,l) \in V_{i,j}} \omega_{k,l}(t-1) = [25\%, 37.5\%] & (5.7) \\ 0, & \omega_{i,j}(t-1) = 1, & \sum_{(k,l) \in V_{i,j}} \omega_{k,l}(t-1) \neq [25\%, 37.5\%] & (5.8) \\ 1, & \omega_{i,j}(t-1) = 0, & \sum_{(k,l) \in V_{i,j}} \omega_{k,l}(t-1) = [35\%, 40\%] & (5.9) \\ 0, & \omega_{i,j}(t-1) = 0, & \sum_{(k,l) \in V_{i,j}} \omega_{k,l}(t-1) \neq [35\%, 40\%]. & (5.10) \end{cases}$$

En la figura 5.2 se muestra el número promedio de nodos activos respecto al tiempo de vida de la red en una disposición aleatoria de 100 nodos contenidos en una superficie bidimensional de forma cuadrada de 200 m \times 200 m. Se usaron los mismos parámetros que el modelo con distribución uniforme. El radio de cada nodo se establece en 15 m. De la misma forma las simulaciones fueron ejecutadas 100 veces con un estado inicial completamente aleatorio para cada una. En este experimento, el número promedio de nodos en la vecindad de un salto es de 2.1, mientras que el porcentaje promedio de nodos activos hasta que el primer nodo se queda sin batería es de 24 %. Al comparar estos valores con los obtenidos en el modelo con distribución de nodos regular (mostrados en la tabla 4.3), ambos valores disminuyeron levemente. Sin embargo, la distribución regular y la irregular muestran un comportamiento parecido, tal como se puede ver al comparar las figuras 4.10 y 5.2

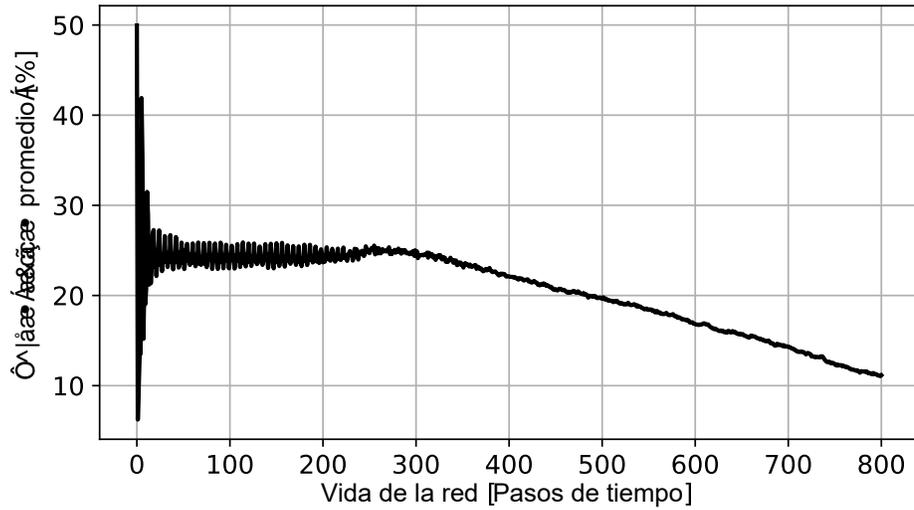


Figura 5.2: Número de nodos activos respecto a la vida de la red con una distribución aleatoria de nodos.

La tabla 5.1 contiene un resumen de los parámetros usados para esta serie de experimentos.

Tabla 5.1: Parámetros de simulación.

Parámetro	Valor
Número de nodos (N)	50
Radio de transmisión	80 m
Tamaño del paquete (l)	500 bytes
Energía inicial (E_0)	5 J
Disipación del circuito (E_{elec})	50 nJ/bit
Amplificación del espacio libre (ϵ_{fs})	100 pJ/bit/m ²
Factor de procesamiento de información (E_{DA})	5 nJ/bit
Número de actuadores (A)	20
Rango de transmisión de los actuadores	30 m
Tamaño de paquete de control	200 bits
Región estudiada	200 m \times 200 m
Posición del nodo sink	(200 m, 200 m)

Una característica a destacar del algoritmo EERP es que tiene veinte nodos actuadores que generan información. Estos nodos son cambiados aleatoriamente cada 100 rondas en

las simulaciones. Es importante mencionar que los mensajes de control (*hello* y *unicast*) en el modelo GoS son procesados debido a que tanto el nodo en cuestión como la vecindad necesitan saber el número de nodos activos durante el paso de tiempo actual para actualizar su estado en el siguiente paso de tiempo. Con este propósito se usan las ecuaciones del modelo de energía 5.4, 5.5 y 5.6.

La figura 5.3 muestra el número de paquetes transmitidos respecto al número de nodos vivos o con batería restante. Se puede ver que GoS tiene una mayor cantidad de paquetes transmitidos respecto a los métodos propuestos por EERP y A&F. Esto se debe a la simplicidad del modelo GoS debido a que intrínsecamente controla los periodos de actividad e inactividad con los tiempos de calendarización.

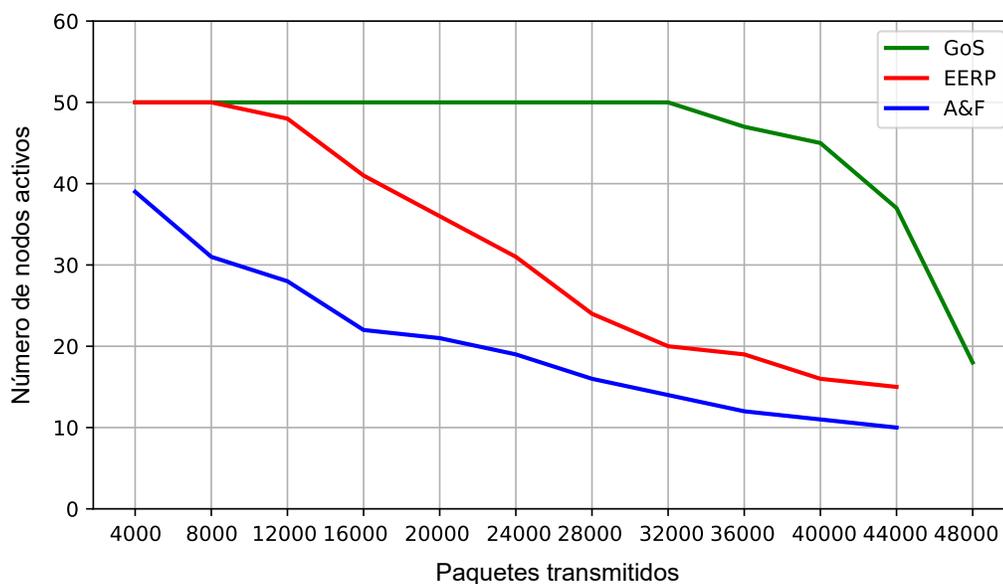


Figura 5.3: Número de nodos activos respecto al número de paquetes transmitidos, entre GoS, EERP y A&F.

Adicionalmente, en el modelo GoS se han transmitido 32,000 paquetes cuando el primer nodo se queda sin batería, mientras que en EERP se han transmitido 8,100 cuando el primer nodo se queda sin batería. En otras palabras, GoS logra triplicar el número de paquetes transmitidos antes de que el primer nodo se quede sin batería. Esta figura también muestra que en el modelo GoS después de que el primer nodo se queda sin batería, el número de nodos con batería restante comienza a disminuir lentamente en comparación con EERP y sobre todo en comparación con A&F. Este comportamiento puede verse también en la figura 5.2, en el cual el número de nodos con carga no cae instantáneamente después

de que el primer nodo se queda sin batería, resultando en un mayor tiempo de vida, y con ello una mayor cantidad de paquetes transmitidos.

Es importante notar que las simulaciones de los autores para EERP [7] solo muestran los resultados entre 4,000 y 44,000 paquetes transmitidos, mientras que en las simulaciones realizadas para GoS continúan hasta que ya no es posible seguir transmitiendo paquetes al nodo *sink* (48,000 paquetes transmitidos).

La figura 5.4 muestra cómo la energía promedio restante en A&F, EERP y GoS decrementa mientras el número de paquetes transmitidos incrementa. Se puede observar que GoS disminuye la energía disponible en los nodos de forma homogénea en comparación del comportamiento mostrado por A&F y EERP.

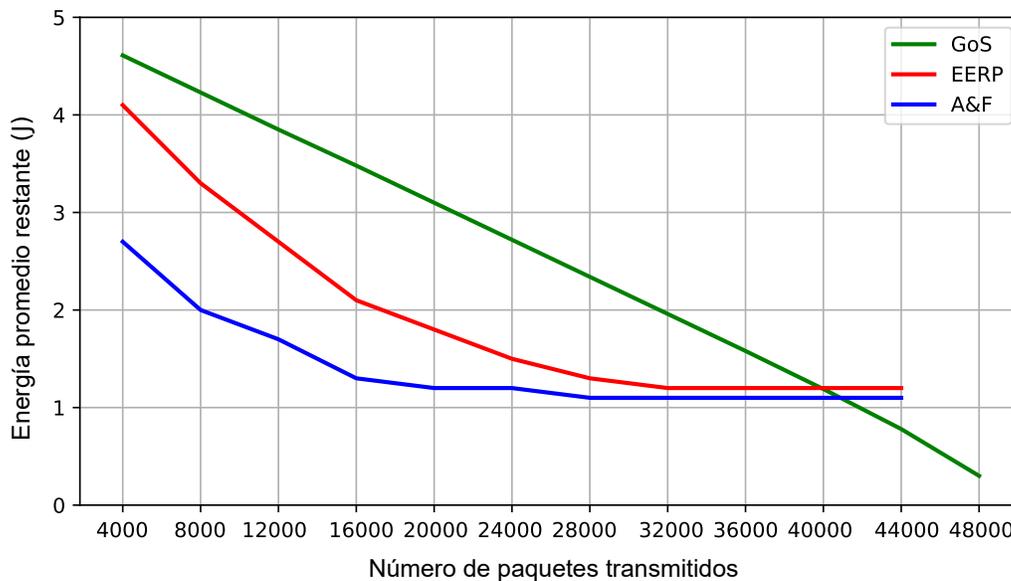


Figura 5.4: Energía restante promedio respecto al número de paquetes transmitidos entre EERP, A&F y GoS.

5.3. GoS contra LPA-star

LPA-star es un algoritmo que se beneficia de hacer búsquedas recursivas en el algoritmo *A-star* para encontrar una mejor ruta. El uso de este mecanismo mejora el rendimiento comparado con el algoritmo EERP. Para estos experimentos, la región estudiada corresponde a un área bidimensional cuadrada de $200 \text{ m} \times 200 \text{ m}$. Cincuenta nodos fueron colocados aleatoriamente en dicha área para crear un escenario similar al presentado en LPA-star [6]. Las simulaciones fueron ejecutadas 100 veces para obtener resultados promedio, cada una

de las simulaciones tiene un escenario inicial completamente aleatorio. Para cada experimento, en el modelo propuesto el nodo *sink* se colocó en la esquina inferior derecha del área de estudio. Es importante mencionar que para estos experimentos en el modelo propuesto se usaron los modelos de energía descritos en 5.4, 5.5 y 5.6, incluso cuando los autores de los trabajos [7] y [6] no incluyen el consumo de energía derivado del procesamiento de los mensajes y paquetes. De igual forma, los autores de [6] limitan el rango de estudio de sus experimentos de 1 a 20,000 paquetes transmitidos. La tabla 5.2 contiene un resumen de los parámetros usados para estos experimentos.

Tabla 5.2: Parámetros de simulación.

Parámetro	Valor
Número de nodos (N)	50
Rango de transmisión del radio	80 m
Tamaño del paquete (l)	4000 bits
Energía inicial (E_0)	5 J
Disipación del circuito (E_{elec})	50 nJ/bit
Amplificación en el espacio libre (ϵ_{fs})	100 pJ/bit/m ²
Factor de procesamiento de información (E_{DA})	5 nJ/bit
Tamaño del paquete de control	200 bits
Región estudiada	200 m \times 200 m
Posición del sink	(200 m, 200 m)

La figura 5.5 muestra cómo GoS es capaz de transmitir el mismo número de paquetes pero sin agotar la batería de ninguno de sus nodos, a diferencia de LPA-star y EERP. Nuevamente, esto se debe a las reglas que sigue el modelo GoS, las cuales provocan que intrínsecamente el modelo regule su propio consumo de energía. De acuerdo a los resultados obtenidos por los autores [6] en EERP se transmiten 2,947 paquetes antes de que el primer nodo se quede sin batería, en LPA-star se pueden transmitir 5,995 paquetes antes de agotar la batería de uno de los nodos, mientras que en GoS el primer nodo se queda sin batería cuando se han transmitido 24,779 paquetes.

Por otro lado, la figura 5.6 muestra cómo la energía promedio restante en LPA-star, EERP y GoS decremента mientras el número de paquetes transmitidos aumenta. Puede verse en la misma figura que LPA-star y GoS disminuyen sus recursos energéticos de manera homogénea en comparación a lo observado para EERP. De cualquier forma, GoS muestra un mejor desempeño en consumo energético que LPA-star y EERP.

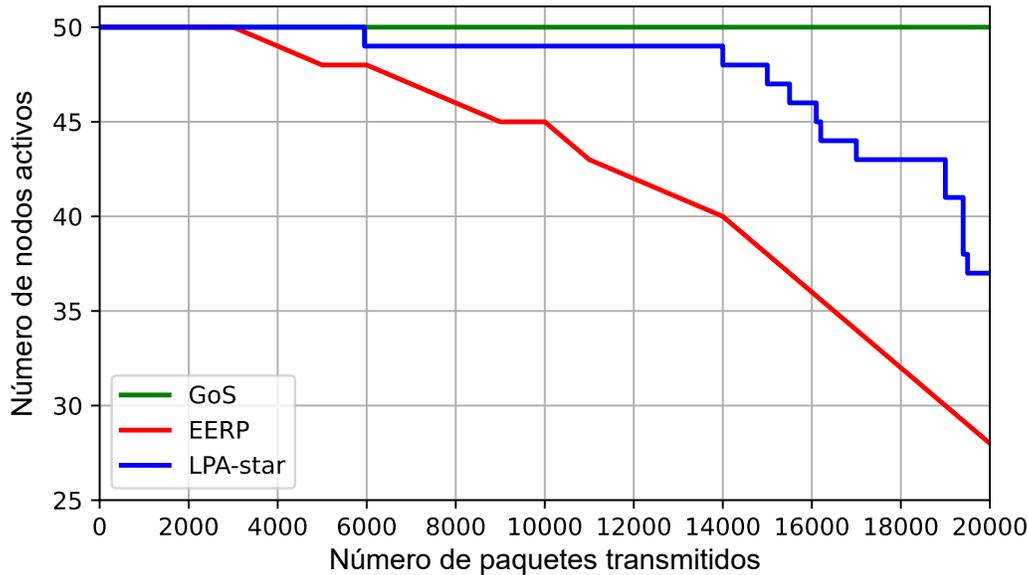


Figura 5.5: Número de nodos con batería respecto a los paquetes transmitidos, entre EERP, LPA-star y GoS.

5.4. GoS contra Maximizing

En estas simulaciones es importante destacar que los autores del protocolo Maximizing [5] y del protocolo *q-switch* [20] no considera el consumo energético de la transmisión de mensajes de control ni de procesamiento de datos. Debido a esto se realizó una nueva implementación de los algoritmos antes mencionados para que se consideren estos consumos energéticos, esto con la finalidad de poder realizar una comparación justa con el modelo GoS.

Para los algoritmos *q-switch* y Maximizing, la región estudiada corresponde a un área circular dividida en k circunferencias o coronas concéntricas, en la cual el nodo *sink* se encuentra en el centro de la corona más interna. Específicamente, en este conjunto de experimentos el valor de k se establece en cinco ($k = 5$), cada corona tiene un ancho de 40m. Todos los nodos tiene el mismo radio máximo de transmisión (80m). La corona i th se identifica como C_i , y los nodos en C_i se pueden comunicar directamente con C_{i-1} y con C_{i+1} . Los nodos en la corona exterior no reenvían ningún paquete, sino que solo se encargan de generar y transmitir sus propios paquetes. El número de nodos en cada corona incrementa en una progresión geométrica en una proporción de 2 desde la corona más externa hasta la corona más interna de la siguiente manera; 3, 3, 6, 12 y 24, para más detalles puede consultarse el funcionamiento de *q-switch* [20].

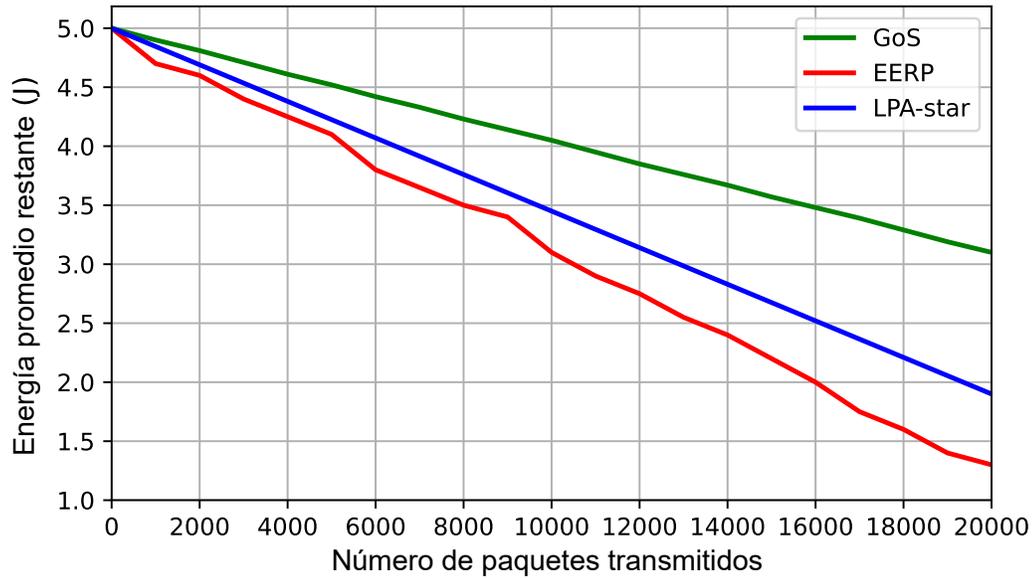


Figura 5.6: Energía promedio restante respecto a los paquetes transmitidos, comparando EERP, LPA-star y GoS.

Como se mencionó en el capítulo 2, en una transmisión *many-to-one*, los nodos cercanos al nodo *sink* agotan su batería más rápido que los nodos que se encuentran alejados del *sink*. Sin embargo, la corona más interna no siempre es la región que más energía consume, esto debido a que el funcionamiento de q-switch hace que otras corona consuma más energía [20]. Por esta situación, los autores de Maximizing proponen una ecuación para encontrar la corona con mayor consumo de energía [5], esta estrategia permite a la red balancear el consumo de energía en cada corona ajustando la energía inicial de los nodos respecto a su ubicación, de tal forma que todos los nodos agotan su batería al mismo tiempo. En otras palabras, la vida de la red termina tan pronto como su primer nodo se queda sin batería, debido a que esto significa que el resto de nodos también han agotado por completo su batería, resultando así en una red con energía residual nula.

La tabla 5.3 contiene un resumen de los parámetros usados en estos experimentos. En cada experimento la energía inicial de cada nodo está distribuida dependiendo de la corona donde se encuentre ubicado.

Con base en lo descrito en [5], y agregando el consumo energético de transmitir mensajes de control y de procesar la información, la cantidad de energía por nodo desde la corona más exterior hasta la más interior es la siguiente:

- $C_1 = 4.258J$

- $C_2 = 5.398J$
- $C_3 = 5.6845J$
- $C_4 = 5.6825J$
- $C_5 = 4.5325J$

Tabla 5.3: Parámetros de simulación.

Parámetro	Valor
Número de nodos (N)	48
Rango de transmisión del radio	80 m
Tamaño del paquete (l)	4000 bits
Energía total en la red (T_E)	240 J
Energía inicial (E_0)	5 J
Disipación del circuito (E_{elec})	50 nJ/bit
Amplificación en el espacio libre (ϵ_{fs})	100 pJ/bit/m ²
Factor de procesamiento de información (E_{DA})	5 nJ/bit
Tamaño del paquete de control	200 bits
Ancho de cada corona	40 m
Radio de la red	$R = 200$ m
Posición del sink	(0 m, 0 m)

Para el experimento usando el modelo GoS la región estudiada corresponde a un área circular con radio de 200 m. Cuarenta y ocho nodos fueron distribuidos aleatoriamente en el área a estudiar, en la cual se colocó también el nodo *sink* en el centro de dicha área. La energía inicial para cada nodo es 5 J. De igual forma, la tabla 5.3 contiene un resumen de los parámetros usados en esta simulación.

La figura 5.7 muestra cómo GoS supera a los algoritmos *q-switch* y Maximizing en cuanto a paquetes totales transmitidos, aún cuando el primer nodo en GoS se queda sin batería antes que en los métodos *q-switch* y Maximizing. Por ejemplo, el modelo Maximizing logra transmitir 34,123 paquetes, mientras que GoS logra transmitir 47,977. En otras palabras, la técnica GoS mejora la cantidad de paquetes transmitidos por 1.4 veces en comparación con el algoritmo Maximizing y 1.9 veces en comparación con *q-switch*.

Aunque *q-switch* como Maximizing son métodos que logran un mejor balance de energía, el método propuesto GoS puede transmitir más paquetes, alargando así la vida de la red. Además, GoS no está atado a funcionar en distribuciones fijas de nodos que no siempre son aplicables a múltiples escenarios donde se despliegan las WSN. Adicionalmente,

tanto *q-switch* como Maximizing requieren una tasa de transmisión de bits constante para garantizar un balance de energía entre todas las coronas. Finalmente, la técnica Maximizing requiere que los nodos tengan distintos niveles de capacidad en sus baterías, lo cual no siempre es posible. Contrario a estos algoritmos, GoS usa una técnica muy simple que puede ser fácilmente implementada en sensores con recursos limitados.

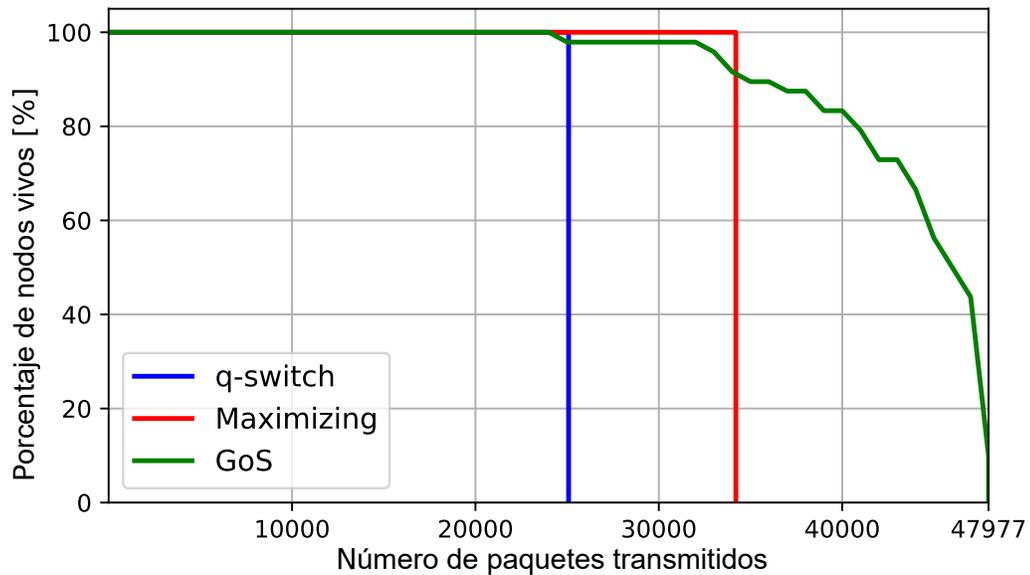


Figura 5.7: Porcentaje de nodos con batería respecto al número de paquetes transmitidos, entre *q-switch*, Maximizing y GoS.

Capítulo 6

Conclusiones

6.1. Validación de la hipótesis

De acuerdo a los resultados obtenidos en el capítulo 5, se puede afirmar que: “Añadir nuevas reglas al modelo del juego de la vida (GoL) e incluir un protocolo de encaminamiento basado en el algoritmo *A-star* permite extender la vida útil de la red, además de asegurar el envío de paquetes a través de la red hacia el nodo *sink*”, es cierta, ya que GoS puede transmitir hasta 32,000 paquetes antes de que el primer sensor agote su batería comparado con 8,100 paquetes que puede transmitir EERP (ver figura 5.3). Más aún, GoS es capaz de transmitir 24,779 paquetes comparado con tan solo 5,995 paquetes que puede transmitir LPA-star cuando su primer nodo agota su batería (ver figura 5.5). Finalmente, Gos es capaz de transmitir 1.4 veces más paquetes que Maximizing y 1.9 veces más que *q-switch* (ver figura 5.7).

6.2. Conclusiones

A diferencia de los trabajos presentados donde los autores usan métodos complejos para extender la vida útil de la red, GoS toma ventaja de uno de los conjuntos de reglas de autómatas celulares más populares, conocido como el juego de la vida o Game of Life. La unión de las reglas del CA GoL junto con un conjunto nuevo de reglas para evitar patrones indeseados y el algoritmo de encaminamiento *A-star* mostraron mejora la vida útil de la red en comparación a los trabajos relacionados más recientes tales como [7], [6], [12], [20] y [5], donde hay una mayor cantidad de paquetes transmitidos, así como una mayor vida útil de la red en general. Adicionalmente, GoS puede ser implementado fácilmente en sensores de recursos limitados dado que el conjunto de reglas de transición propuestas requieren

de técnicas de programación simple, ya que basta con hacer comparaciones para poder ser implementadas. Además, el algoritmo propuesto considera parámetros como la energía residual, el número de vecinos activos, y el tiempo de calendarización para garantizar un consumo balanceado de la energía mientras que minimiza el número de saltos necesarios para alcanzar al nodo *sink* en una red a través del algoritmo *A-star*. Asimismo, es importante destacar que el modelo GoS puede funcionar tanto en escenarios con distribución de nodos predeterminada así como en escenarios en los que la distribución de los nodos es aleatoria, recordando que aun cuando el escenario genere nuevos obstáculos, el algoritmo *A-star* es capaz de generar nuevas rutas que nos permitan evadir el obstáculo o esperar hasta que el obstáculo desaparezca, pero siempre encontrando una ruta hacia el nodo destino.

Los resultados de las simulaciones demuestran que el modelo propuesto balancea el consumo de energía de manera muy precisa, extendiendo así la vida útil de la red comparado con los trabajos relacionados más recientes. Por otra parte, el modelo propuesto tiene la capacidad de actuar como base o ser combinado con algoritmos de planificación de rutas para lograr un mejor desempeño.

Finalmente, aunque el consumo de energía en GoS es bastante homogéneo, es importante destacar que los resultados mostrados fueron obtenidos hasta que la vida útil de la red terminó, sin embargo puede darse el caso en el que aunque la red ya no tenga la capacidad de transmitir paquetes hacia el nodo *sink*, existan aún nodos con carga en sus baterías, por lo que un esquema de control de potencia de transmisión podría ser implementado para que los nodos con batería residual puedan intentar formar vecindades de mayor radio y así continuar con la dinámica del sistema.

6.3. Perspectivas de investigación

Como trabajo o investigación futura, se puede proponer el uso de la técnica empleada en este trabajo para implementar un modelo de autómatas celulares con *sink* móviles para maximizar el consumo de energía en la red, especialmente en WSN en las que los dispositivos se encuentran muy dispersos en el área de estudio, de tal forma que el *sink* móvil sea capaz de conseguir una ruta óptima usando el algoritmo A* o algún otro algoritmo de generación de rutas óptimas.

Bibliografía

- [1] O. Arana, F. Garcia, J. Gomez, Analysis of the effectiveness of transmission power control as a location privacy technique, *Computer Networks* 163 (2019) 106880. doi:<https://doi.org/10.1016/j.comnet.2019.106880>.
URL <https://www.sciencedirect.com/science/article/pii/S1389128618311034>
- [2] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks* 7 (3) (2009) 537–568. doi:<https://doi.org/10.1016/j.adhoc.2008.06.003>.
URL <https://www.sciencedirect.com/science/article/pii/S1570870508000954>
- [3] J. Lee, S. Adachi, F. Peper, K. Morita, Asynchronous game of life, *Physica D: Nonlinear Phenomena* 194 (3) (2004) 369–384. doi:<https://doi.org/10.1016/j.physd.2004.03.007>.
URL <https://www.sciencedirect.com/science/article/pii/S0167278904001198>
- [4] T. Fujita, K. Nakano, Y. Ito, Fast simulation of conway’s game of life using bitwise parallel bulk computation on a gpu, *International journal of foundations of computer science* 27 (08) (2016) 981–1003. doi:[10.1142/s0129054116500404](https://doi.org/10.1142/s0129054116500404).
URL <http://dx.doi.org/10.1142/s0129054116500404>
- [5] H. Asadollahi, S. Zandi, H. Asharioun, Maximizing network lifetime in many-to-one wireless sensor networks (wsns), *Wireless Pers Commun* 123 (2022) 2971–2983. doi:[10.1007/s11277-021-09271-9](https://doi.org/10.1007/s11277-021-09271-9).
URL <https://link.springer.com/article/10.1007/s11277-021-09271-9>
- [6] A. A. Alkadhawee, S. Lu, Prolonging the network lifetime based on lpa-star algorithm and fuzzy logic in wireless sensor network, in: 2016 12th World Con-

- gress on Intelligent Control and Automation (WCICA), 2016, pp. 1448–1453. doi: 10.1109/WCICA.2016.7578763.
- [7] A. Ghaffari, An energy efficient routing protocol for wireless sensor networks using a-star algorithm, *Journal of Applied Research and Technology* (2014). doi:10.1016/S1665-6423(14)70097-5.
URL <https://jart.icat.unam.mx/index.php/jart/article/view/196>
- [8] I. Gupta, D. Riordan, S. Sampalli, Cluster-head election using fuzzy logic for wireless sensor networks, in: *3rd Annual Communication Networks and Services Research Conference (CNSR'05)*, 2005, pp. 255–260. doi:10.1109/CNSR.2005.27.
- [9] Y. Yuan, C. Li, Y. Yang, X. Zhang, L. Li, Caf: Cluster algorithm and a-star with fuzzy approach for lifetime enhancement in wireless sensor networks, *Hindawi Publishing* (2014). doi:10.1155/2014/936376.
URL <https://www.hindawi.com/journals/aaa/2014/936376/>
- [10] G. Smaragdakis, I. Matta, A. Bestavros, Sep: A stable election protocol for clustered heterogeneous wireless sensor networks (2004).
URL <https://open.bu.edu/handle/2144/1548>
- [11] S. Soijoyo, R. Wardoyo, Wireless sensor network energy efficiency with fuzzy improved heuristic a-star method, *International Journal of Advanced Computer Science and Applications* 8 (4) (2017). doi:10.14569/IJACSA.2017.080443.
URL <http://dx.doi.org/10.14569/IJACSA.2017.080443>
- [12] I. S. AlShawi, L. Yan, W. Pan, B. Luo, Lifetime enhancement in wireless sensor networks using fuzzy approach and a-star algorithm, in: *IET Conference on Wireless Sensor Systems (WSS 2012)*, 2012, pp. 1–6. doi:10.1049/cp.2012.0611.
- [13] E. Amiri, H. Keshavarz, M. Alizadeh, M. Zamani, T. Khodadadi, Energy efficient routing in wireless sensor networks based on fuzzy ant colony optimization, *International Journal of Distributed Sensor Networks* 10 (7) (2014) 768936. doi:10.1155/2014/768936.
URL <https://doi.org/10.1155/2014/768936>
- [14] A. J. Obaid, Wireless sensor network (wsn) routing optimization via the implementation of fuzzy ant colony (faco) algorithm: Towards enhanced energy conservation, in: R. Kumar, B. K. Mishra, P. K. Pattnaik (Eds.), *Next Generation of Internet of Things*, Springer Singapore, Singapore, 2021, pp. 413–424. doi:10.1007/978-981-16-0666-3_33.

-
- [15] S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, La-mhr: Learning automata based multilevel heterogeneous routing for opportunistic shared spectrum access to enhance lifetime of wsn, *IEEE Systems Journal* 13 (1) (2019) 313–323. doi:10.1109/JSYST.2018.2818618.
- [16] C. Subha, S. Malarkkan, Optimization of energy efficient cellular learning automata algorithm for heterogeneous wireless sensor networks, in: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–6. doi:10.1109/ISCO.2016.7727094.
- [17] A. H. F. Navid, Selarp: Scalable and energy-aware learning automata-based routing protocols for wireless sensor networks, in: *2010 Fourth International Conference on Sensor Technologies and Applications*, 2010, pp. 570–576. doi:10.1109/SENSORCOMM.2010.91.
- [18] S. Hao, H. Zhang, M. Song, A stable and energy-efficient routing algorithm based on learning automata theory for manet, *Journal of Communications and Information Networks* 3 (2018) 52–66. doi:10.1007/s41650-018-0012-7.
URL <https://link.springer.com/article/10.1007/s41650-018-0012-7>
- [19] M. Kamarei, A. Patooghy, Z. Shahsavari, M. Javad Salehi, Lifetime expansion in wsns using mobile data collector: A learning automata approach, *Journal of King Saud University - Computer and Information Sciences* 32 (1) (2020) 65–72. doi:10.1016/j.jksuci.2018.03.006.
URL <https://www.sciencedirect.com/science/article/pii/S1319157818300880>
- [20] X. Wu, G. Chen, S. K. Das, Avoiding energy holes in wireless sensor networks with nonuniform node distribution, *IEEE Transactions on Parallel and Distributed Systems* 19 (5) (2008) 710–720. doi:10.1109/TPDS.2007.70770.
- [21] T. Liu, T. Gu, N. Jin, Y. Zhu, A mixed transmission strategy to achieve energy balancing in wireless sensor networks, *IEEE Transactions on Wireless Communications* 16 (4) (2017) 2111–2122. doi:10.1109/TWC.2016.2642098.
- [22] G. Chen, C. Li, M. Ye, An unequal cluster-based routing protocol in wireless sensor networks, *Wireless Networks* 15 (2009) 193–207. doi:10.1007/s11276-007-0035-8.
URL <https://link.springer.com/article/10.1007/s11276-007-0035-8>

-
- [23] H.-W. Ferng, M. Hadiputro, A. Kurniawan, Design of novel node distribution strategies in corona-based wireless sensor networks, *IEEE Transactions on Mobile Computing* 10 (9) (2011) 1297–1311. doi:10.1109/TMC.2010.241.
- [24] G. M. B. Oliveira, R. G. O. Silva, G. B. S. Ferreira, M. S. Couceiro, L. R. do Amaral, P. A. Vargas, L. G. A. Martins, A cellular automata-based path-planning for a cooperative and decentralized team of robots, in: *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 739–746. doi:10.1109/CEC.2019.8790205.
- [25] L. G. A. Martins, R. d. P. Cândido, M. C. Escarpinati, P. A. Vargas, G. M. B. de Oliveira, An improved robot path planning model using cellular automata, in: M. Giuliani, T. Assaf, M. E. Giannaccini (Eds.), *Towards Autonomous Robotic Systems*, Springer International Publishing, Cham, 2018, pp. 183–194.
- [26] S. C. S. Nametala, L. G. A. Martins, G. M. B. Oliveira, A new distance diffusion algorithm for a path-planning model based on cellular automata, in: *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8. doi:10.1109/CEC48606.2020.9185824.
- [27] G. B. S. Ferreira, P. A. Vargas, G. M. B. Oliveira, An improved cellular automata-based model for robot path-planning, in: M. Mistry, A. Leonardis, M. Witkowski, C. Melhuish (Eds.), *Advances in Autonomous Robotics Systems*, Springer International Publishing, Cham, 2014, pp. 25–36.
- [28] J. Tariq, A. Kumaravel, Construction of cellular automata over hexagonal and triangular tessellations for path planning of multi-robots, in: *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCI)*, 2016, pp. 1–6. doi:10.1109/ICCI.2016.7919686.
- [29] C. Banerjee, S. Saxena, Energy conservation in wireless sensor network using block cellular automata, in: *2013 International Conference on Computer Communication and Informatics*, 2013, pp. 1–6. doi:https://doi.org/10.1109/ICCCI.2013.6466126.
URL <https://ieeexplore.ieee.org/document/6466126>
- [30] S. Choudhury, S. G. Akl, K. Salomaa, Energy efficient cellular automaton based algorithms for mobile wireless sensor networks, in: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 2341–2346. doi:10.1109/WCNC.2012.6214185.

-
- [31] S. Wolfram, *A New Kind of Science*, Wolfram Media, 2002.
URL <https://www.wolframscience.com>
- [32] A. Adamatzky, *Game of Life Cellular Automata*, Springer London, 2010.
URL <https://doi.org/10.1007/978-1-84996-217-9>
- [33] J. Reyes, *Modelado de consumo energético en redes iot basado en autómatas celulares*, Tesis de licenciatura, Universidad Nacional Autónoma de México (2020).
URL <http://132.248.52.100:8080/xmlui/handle/132.248.52.100/17281>
- [34] T. Shankar, G. Eappen, S. Rajalakshmi, Optimized routing algorithm for wireless sensor networks, in: M. R. Nalim, R. Vasudevan, S. Rahatekar (Eds.), *Advances in Automotive Technologies*, Springer Singapore, Singapore, 2021, pp. 83–96.