



Asignación Automática de Rutas en Redes de Subestaciones Eléctricas Virtualizadas

Wilmar Alesander Vásquez Barrientos

Tesis de maestría presentada para optar al título de Magíster en Ingeniería de Telecomunicaciones

Director

Juan Felipe Botero Vega, Doctor (PhD) en Ingeniería Telemática

Universidad de Antioquia

Facultad de Ingeniería

Maestría en Ingeniería de Telecomunicaciones

Medellín, Antioquia, Colombia

2024

Cita	(Vásquez Barrientos, 2024)
Referencia	Vásquez Barrientos, W. A. (2024). <i>Asignación Automática de Rutas en Redes de Subestaciones Eléctricas Virtualizadas</i> [Tesis de maestría].
Estilo APA 7 (2020)	Universidad de Antioquia, Medellín, Colombia.



Maestría en Ingeniería de Telecomunicaciones, Cohorte VII.

Grupo de Investigación GITA

Centro de Investigación Ambientales y de Ingeniería (CIA).



Centro de documentación de Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.



**UNIVERSIDAD
DE ANTIOQUIA**
1 8 0 3

Universidad de Antioquia
Departamento de Ingeniería Electrónica y Telecomunicaciones
Grupo de Investigación en Telecomunicaciones Aplicadas (GITA Lab)

Asignación Automática de Rutas en Redes de Subestaciones Eléctricas Virtualizadas

Ing. Wilmar Alesander Vásquez Barrientos

En cumplimiento de los requisitos para obtener el grado:
Magister en Ingeniería de Telecomunicaciones

1. Revisor

2. Revisor

Director Ph.D. Juan Felipe Botero Vega
Universidad de Antioquia

27 de abril de 2024

Ing. Wilmar Alesander Vásquez Barrientos

Asignación Automática de Rutas en Redes de Subestaciones Eléctricas Virtualizadas

27 de abril de 2024

Director: Ph.D. Juan Felipe Botero Vega

Universidad de Antioquia

Grupo de Investigación en Telecomunicaciones Aplicadas (GITA Lab)

Departamento de Ingeniería Electrónica y Telecomunicaciones

Calle 67 No. 53 - 108

Índice de figuras

1.1	Red eléctrica	2
1.2	Red de subestación eléctrica	4
2.1	Equipo físico con 3 máquinas virtuales	10
2.2	Equipo físico con 6 contenedores	11
2.3	Red virtual	12
2.4	Arquitectura SDN	13
2.5	Arquitectura NFV	14
2.6	Solución gráfica del problema ILP.	18
2.7	Grafo inicial.	20
2.8	Actualización Feromona.	21
2.9	Arquitectura subestación eléctrica tipo T1-1	23
2.10	Arquitectura subestación eléctrica digital	24
4.1	Modelo S3N (Imagen tomada para la edición de [LB16a])	30
4.2	Modelo S3N ampliado(Imagen tomada para la edición de [LB16a])	30
5.1	Red Spiderweb general	39
5.2	Parejas de prueba por cada servicio	41
5.3	Imagen ‘manager’ 324MB.	43
5.4	Servicio GOOSE 7.22MB.	43
5.5	Red A con tres servicios diferentes Vs red B con múltiples publicadores P y múltiples suscriptores S para cada servicio sin interferencia.	44
6.1	Red Spiderweb 8 Nodos	52
6.2	Red spiderweb con 20 Nodos	53
6.3	Red spiderweb con 30 Nodos	55
6.4	Escenarios de prueba	56
6.5	Distribución de tiempos de ruta (segundos) para la red de 8 nodos	65
6.6	Distribución de tiempos de ruta (segundos) para la red de 20 nodos	65
6.7	Distribución de tiempos de ruta (segundos) para la red de 30 nodos	66
6.8	Tiempos de ejecución	67

Índice de tablas

2.1	Parámetros del algoritmo ACO.	22
2.2	Descripción de los diferentes tipos de servicio.	25
4.1	Propuestas de virtualización en subestaciones eléctricas.	32
4.2	Propuestas de optimización de rutas.	38
5.1	Parejas <i>Publisher-Subscriber</i>	41
5.2	Imágenes base.	42
5.3	Parámetros de comunicación para cada servicio.	44
5.4	Parámetros obtenidos para cada servicio.	45
5.5	Índices, variables y parámetros del modelo propuesto.	46
5.6	Parámetros iniciales del algoritmo ACO.	50
5.7	Parámetros secundarios del algoritmo ACO.	50
6.1	Vector de retardos para un enlace	51
6.2	Matriz de retardos para la red de 8 nodos y 14 enlaces	52
6.3	Matriz de retardos para la red de 20 nodos y 38 enlaces	53
6.4	Matriz de retardos para la red de 30 nodos y 58 enlaces	54
6.5	Enlaces con falla para cada topología	57
6.6	Promedio de enlaces usados en cada demanda ILP	58
6.7	Promedio de nodos usados en cada demanda ILP	59
6.8	Métricas ILP con solución exacta	60
6.9	Promedio de enlaces usados en cada demanda ACO	62
6.10	Promedio de nodos usados en cada demanda ACO	62
6.11	Métricas con algoritmo ACO	63
6.12	Comparación de resultados de las Tablas 6.11 y 6.8.	64
6.13	Resultados prueba U de Mann-Whithney ACO VS ILP	67

Abstract

Electrical substations are a key element in the transmission, control, and transformation of energy. In addition to having transformation, protection, and control equipment, substations have communication networks to control and monitor the entire system. However, this infrastructure requires (i) backup equipment in a quantity similar to that operating in the substation network, and (ii) expert personnel to monitor the network 24 hours a day, 7 days a week to react immediately in case of a failure.

The virtualization of the data network in electrical substations has emerged as a technological upgrade alternative. This transformation facilitates overcoming challenges related to flexibility, resilience, and the performance of the underlying communication network. The integration of all substation components into the data network, signal digitalization, and the use of ethernet for controlling protection devices and managing the substation make the communication network a critical component for the system, as it concentrates the flow of data from all elements.

Since electric grids, in general, are considered critical infrastructures, it is of utmost importance to maintain continuous service, aiming to minimize any downtime. To achieve this, redundant systems are implemented in equipment, networks, and platforms, and highly trained technical personnel are available to ensure on-demand operation and to guarantee the required service level, even in failure situations.

A connectivity failure can consequently lead to the system's collapse. Conversely, the performance and the condition of the substation depend significantly on the stability of the communication network and its tolerance to link failures.

In this research, two different approaches are presented to find an alternative route in case of a failure in a communication link of a virtualized electrical substation: (i) an exact integer linear programming model and (ii) a metaheuristic algorithm based on ant colonies.

To replicate functions specific to the substation, virtualized services (GOOSE, SV, and MMS) were implemented in a substation network, and failures were simulated

on some links. Two different performance metrics were measured: (i) the running time each algorithm took to find alternative routes and (ii) the delay of the path for the resulting alternative route.

According to the results, the metaheuristic algorithm, Ant Colony Optimization, is faster at finding alternative routes in small instances. This is possibly because the algorithm focuses on finding a viable route without necessarily considering if it is the optimal one. In addition, convergence in this algorithm becomes challenging for larger networks due to the disappearance of pheromone imprints in the random process of choosing the path. However, adjustments are needed for the algorithm to ensure its convergence in the case of large instances. Meanwhile, the solutions for every service are similar between the exact model and the metaheuristic algorithm in all scenarios.

Further research is required to assess the suitability of the metaheuristic algorithm to work in more realistic scenarios, where physical and virtualized devices work together.

Keywords: Service virtualization, containers, route optimization, network restoration, NFV, electrical substations

Resumen

Las subestaciones eléctricas son un elemento clave en la transmisión, control y transformación de la energía. Además de contar con equipos de transformación, protección y control, las subestaciones cuentan con redes de comunicación para controlar y monitorear todo el sistema. Sin embargo, esta infraestructura requiere tener (i) equipos de respaldo en algunas ocasiones en cantidad similar a los que operan en la red de comunicaciones de la subestación, y (ii) personal experto que vigile la red las 24 horas del día, los 7 días de la semana para reaccionar inmediatamente en caso de una falla.

La virtualización de la red de datos de las subestaciones eléctricas ha surgido como una alternativa de actualización tecnológica. Esta transformación facilita la superación de los desafíos relacionados con la flexibilidad, resiliencia y rendimiento de la red de comunicación subyacente. La integración de todos los componentes de la subestación a la red de datos, la digitalización de las señales y el uso de la red ethernet para el control de los dispositivos de protección y el manejo de la subestación hacen que la red de comunicación sea un componente crítico para el sistema, ya que concentra el flujo de datos de todos los elementos.

Dado que las redes eléctricas en general son consideradas infraestructuras críticas, resulta de suma importancia mantener el servicio en funcionamiento de manera continua, buscando reducir al mínimo cualquier tiempo de interrupción. Para lograrlo, se implementan sistemas redundantes en equipos, redes y plataformas, y se cuenta con personal técnico altamente capacitado para garantizar el funcionamiento bajo demanda y asegurar el nivel de servicio requerido, incluso en situaciones de fallo.

La importancia de encontrar rutas alternativas ante un fallo en la red de datos de una subestación eléctrica impacta directamente en el funcionamiento del sistema, reduce el costo operativo de mantener el servicio en alto y, en consecuencia, evita el colapso del sistema. Por lo tanto, el rendimiento y el estado de la subestación dependen en gran medida de la estabilidad de la red de comunicación y su tolerancia a fallos en los enlaces.

En esta investigación, se presentan dos enfoques diferentes para encontrar una ruta alternativa en caso de fallo en un enlace de la red de comunicaciones de una subestación eléctrica virtualizada: (i) un modelo exacto de programación lineal entera y (ii) un algoritmo metaheurístico basado en colonias de hormigas.

Con el fin de reproducir funciones propias de la subestación, se implementaron servicios virtualizados (GOOSE, SV y MMS) en una red de subestaciones y se simuló fallas en algunos enlaces, midiendo dos métricas de desempeño distintas: (i) el tiempo de cálculo que cada algoritmo tomó para encontrar rutas alternativas, y (ii) el retardo del camino para la ruta alternativa resultante.

Según los resultados, el algoritmo metaheurístico es más rápido para encontrar rutas alternativas en pequeñas instancias, posiblemente porque este algoritmo se centra en encontrar una ruta viable, sin necesariamente considerar si es la óptima. En cuanto a topologías con mayor número de nodos, se debe hacer ajustes al algoritmo para garantizar su convergencia. En la búsqueda de las mejores rutas entre el modelo exacto y el algoritmo ACO, la prueba estadística evidencia que no se encuentra una diferencia significativa entre ambas soluciones. Se requiere mayor investigación para evaluar la idoneidad del algoritmo metaheurístico en escenarios más realistas, donde los dispositivos físicos y virtualizados trabajen juntos.

Palabras Claves: Virtualización de servicios, contenedores, optimización de rutas, restablecimiento de la red, NFV, subestaciones eléctricas

Agradecimientos

Este proceso ha sido largo y de gran aprendizaje. Hubo personas que me impulsaron a ingresar a la universidad con el propósito de cambiar mi vida y ampliar mi visión del mundo. Aunque ya no me acompañe físicamente, se le recuerda con mucho amor por su sacrificio y dedicación para contribuir a mi desarrollo personal y académico ¡Muchas gracias, papá!. También quiero manifestar mi mas sincero agradecimiento al profesor Juan Rafael Orozco, quien sin proponérselo, hizo una contribución significativa en la reorientación de objetivos personales y profesionales. En especial, fue el 'gruñón' incansable que mantuvo viva la motivación que me ayudó a culminar este proceso con éxito.

Índice general

1	Introducción	1
1.1	Motivación	4
1.2	Organización	6
2	Marco teórico	9
2.1	Virtualización	9
2.1.1	Virtualización de equipos	9
2.1.2	Virtualización de red	11
2.2	Redes Definidas por Software (SDN)	12
2.3	Virtualización de las funciones de red (NFV)	13
2.4	Optimización de rutas	15
2.4.1	Optimización de rutas usando métodos exactos	16
2.4.2	Métodos heurísticos y metaheurísticos	18
2.5	Subestaciones eléctricas	22
2.6	Subestaciones eléctricas digitales	23
3	Planteamiento del problema	27
3.1	Definición del problema de investigación	27
3.2	Objetivos	28
4	Estado del arte	29
4.1	Virtualización en redes de subestaciones eléctricas	29
4.2	Optimización de rutas usando métodos exactos y metaheurísticos	32
4.3	Algoritmos misceláneos	34
5	Implementación de algoritmos de solución del problema de rutas alternativas usando ILP y ACO	39
5.1	Virtualización de servicios en redes eléctricas virtualizadas	40
5.1.1	Creación de imagen ‘manager’	42
5.1.2	Creación de servicios	43
5.2	Modelo ILP	45
5.3	Algoritmo Metaheurístico ACO	49
6	Experimentos y resultados	51
6.1	Experimento 1: método exacto	57

6.2	Experimento 2: método metaheurístico	61
6.3	Comparación entre método metaheurístico y método exacto	63
7	Conclusiones y trabajo futuro	69
7.1	Conclusiones	69
7.2	Trabajo futuro	70
8	Resumen	73
	Bibliografía	77

Introducción

La economía del siglo XXI tiene un énfasis en la búsqueda de eficiencias operativas, apertura de nuevos mercados y una conciencia ambiental mucho más rigurosa y dinámica, lo cual contrasta con los modelos económicos de períodos anteriores. Esta tendencia se hace más notoria en el sector de los servicios públicos que se ha caracterizado por ser una industria estática y conservadora. A modo de ilustración en el sector eléctrico, muchos de los modelos de operación siguen utilizando iguales o semejantes técnicas a las establecidas hace más de 70 años por Thomas Edison, George Westinghouse, y Nicola Tesla [AKV10]. A pesar de que no es tan evidente, esta industria se esfuerza constantemente por adaptarse a los paradigmas de la sociedad moderna.

Dentro del panorama de cambio, es esencial reconocer que una red eléctrica opera como un conjunto de subsistemas interdependientes que se acoplan para asegurar el debido funcionamiento del sistema eléctrico en general. Entre estos subsistemas se destacan: (i) la generación a partir de diferentes fuentes como plantas de energía hidroeléctricas, térmicas, nucleares, eólicas o solares, (ii) la transmisión de la electricidad desde las plantas generadoras a un usuario final, (iii) la transformación de la energía a diferentes niveles de tensión para facilitar su distribución y adecuarla a las necesidades de los usuarios y (iv) la distribución en áreas locales y regionales, hasta llegar a los hogares, empresas e industrias (ver Figura 1.1).

Las *Smart Grids*, o Redes Eléctricas Inteligentes, representan una transformación esencial en la infraestructura eléctrica, aprovechando tecnologías avanzadas para mejorar la eficiencia y confiabilidad del sistema. Estas redes permiten una comunicación bidireccional en tiempo real, integran energías renovables de manera eficiente, permiten a los consumidores autogestionar su consumo, promueven la resiliencia ante crisis y mejoran la calidad de la energía. Este enfoque representa una importante alternativa para abordar los desafíos energéticos actuales y garantizar un suministro eléctrico sostenible y confiable.

Dentro de la estructura de la red eléctrica, las subestaciones eléctricas desempeñan un papel fundamental. Básicamente, son instalaciones divididas en varios componentes esenciales. En su núcleo se encuentran los transformadores, que desempeñan un papel importante al convertir la electricidad de alta tensión en baja tensión y

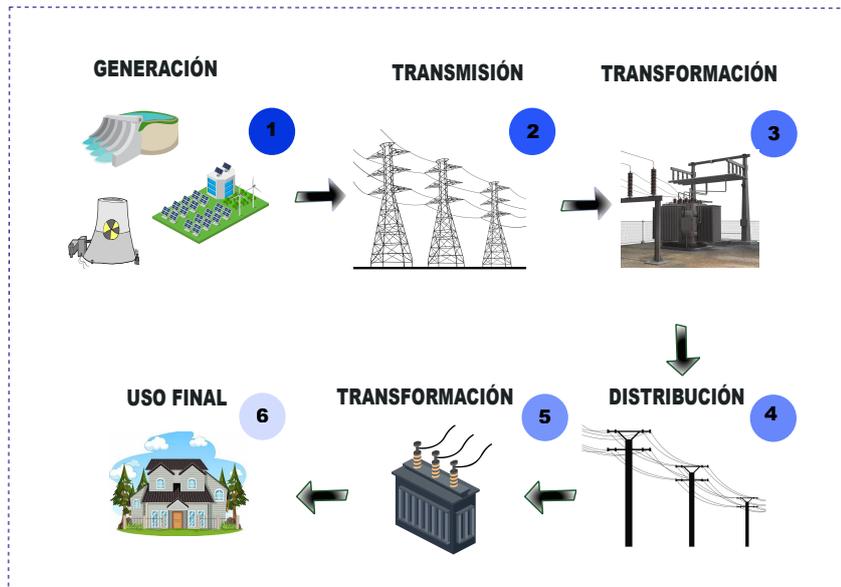


Figura. 1.1: Red eléctrica

viceversa, permitiendo su distribución óptima y segura. Esto requiere de interruptores que controlan el flujo de electricidad, equipos de medición para registrar datos importantes, aisladores que previenen cortocircuitos, y dispositivos de protección, como fusibles y relés¹, que se activan en caso de sobrecargas o fallos [Gri18].

Dentro del contexto de las subestaciones eléctricas, es importante destacar que las comunicaciones en las redes tradicionales solían ser de naturaleza analógica. Sin embargo, gracias a la tecnología ethernet, se ha producido una transición hacia la digitalización, un paso significativo en el camino hacia la *Smart Grid*. En este escenario, se emplean dos tipos de redes de comunicaciones: una red Ethernet, que es ampliamente utilizada para la transmisión de datos, y una red punto a punto, que se utiliza para establecer conexiones específicas entre dispositivos.

La red Ethernet se encarga de facilitar la comunicación eficiente entre todos los sistemas de la subestación, permitiendo el reenvío de datos esenciales para el sistema de alertas, el monitoreo constante y la transmisión de señales a los relés de protección y control. Por otro lado, la red punto a punto se compone de sistemas especializados que supervisan las señales analógicas entre los equipos de patio, como los transformadores de corriente (CT), transformadores de voltaje (VT) y transformadores de potencia (PT), además de los equipos de protección y control (Ver Figura 1.2). Esta red garantiza la precisión en la transmisión de señales críticas, lo que resulta fundamental para el funcionamiento seguro y eficaz de la subestación.

¹Un relé es un dispositivo que detecta condiciones anormales o fallas en el sistema eléctrico y toma medidas para evitar daños o interrupciones en el suministro de energía, su función es garantizar la seguridad y confiabilidad del sistema eléctrico.

En conjunto, estas dos redes permiten una supervisión en tiempo real del estado de la subestación, la detección y notificación inmediata de posibles problemas, la ejecución de acciones de control automático para mantener la integridad y la continuidad del suministro eléctrico, consolidando así su importancia en la operatividad fiable y segura de la infraestructura eléctrica.

Dado que las redes eléctricas en general son consideradas infraestructuras críticas [LB16a], resulta de suma importancia mantener el servicio en funcionamiento de manera continua, buscando reducir al mínimo cualquier tiempo de interrupción. Para lograrlo, se implementan sistemas redundantes en equipos, redes y plataformas, y se cuenta con personal técnico altamente capacitado para garantizar el funcionamiento bajo demanda y asegurar el nivel de servicio requerido, incluso en situaciones de fallo.

Según lo anterior, el costo de operar y mantener en funcionamiento la red de una subestación eléctrica es una tarea que implica una inversión significativa en capital (CAPEX, por las siglas en Inglés de CAPital EXpenditure) destinado a equipos de respaldo, alta densidad de cableado (conexiones punto a punto) entre equipos de sensado y equipos de protección y control. La gestión de estos elementos representa también un alto costo operativo (OPEX, por las siglas en Inglés de OPerational EXpenditure) debido a la necesidad de contar con personal altamente capacitado para manejar, programar, establecer y brindar asistencia en tiempos cortos a imprevistos en la subestación eléctrica, sin afectar la provisión del servicio. La Figura 1.2 ilustra las componentes de una red en una subestación eléctrica típica.

Dada la problemática mencionada anteriormente (OPEX y CAPEX), los operadores del servicio de energía eléctrica se han visto en la obligación de adelantar procesos de actualización tecnológica en busca de mejorar la eficiencia operativa, la agilidad, la escalabilidad y la seguridad, al mismo tiempo que reducir los costos y simplificar la gestión de recursos. Un cambio similar han experimentado los centros de datos [SDF20], los cuales incorporaron tecnologías de virtualización como las basadas en redes definidas por software (SDN, por las siglas en Inglés de Software Defined Networking) [Kre+14], virtualización de equipos mediante el uso de máquinas virtuales (VM) [VMw23] o máquinas virtuales ligeras (contenedores) (LVM) [Doc23], y la virtualización de servicios NFV (por las siglas en Inglés de Network functions virtualization) [Mij+15].

Las subestaciones eléctricas digitales representan una evolución que combina la eficiencia de las redes de datos con las operaciones del sistema eléctrico. Esto se logra mediante la digitalización, que implica conectar dispositivos a través de tecnología ethernet y seguir el estándar IEC 61850 para garantizar la interoperabilidad. Este estándar se estructura en tres niveles: estación, bahía y proceso.

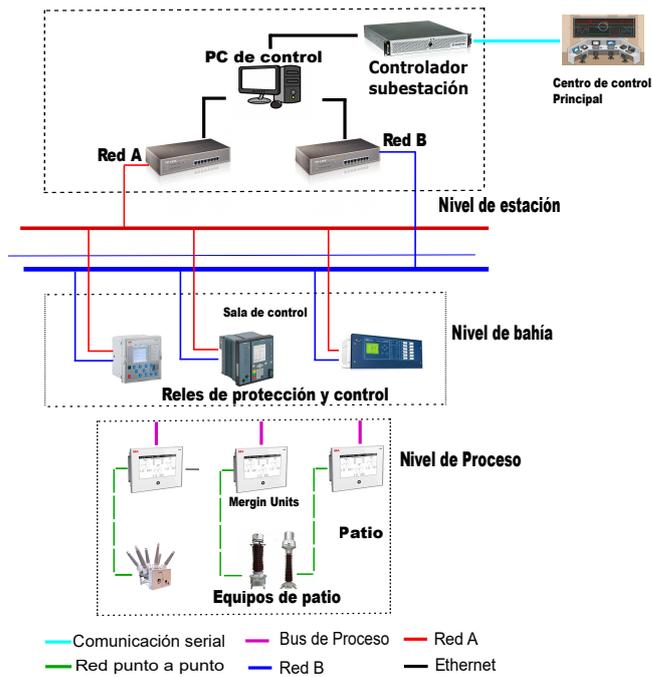


Figura. 1.2: Red de subestación eléctrica

En el nivel de proceso, se encuentran equipos como seccionadores, dispositivos de protección y conversión de señales. El nivel de bahía comprende dispositivos electrónicos inteligentes que pueden tomar decisiones cuando se detectan problemas. El nivel de estación es la capa superior y supervisa y controla todo el sistema. Estos niveles se conectan a través de dos redes ethernet de comunicación, el bus de proceso y el bus de estación, que utilizan topologías en anillo para garantizar redundancia y resiliencia del sistema.

Así como SDN y NFV entraron en las redes de los centros de datos y en la nube, algunos trabajos científicos [LB16d; WKR17; LDB20; Rös+22] que proponen el uso de SDN y NFV para gestionar las redes de comunicación de las subestaciones eléctricas. La virtualización y tecnologías como NFV y SDN ofrecen oportunidades en el contexto de subestaciones virtualizadas, abriendo un amplio campo para la innovación en la gestión y optimización de estos sistemas. La optimización de rutas y la restauración de servicios después de fallas en los enlaces de la red de datos se convierten en aspectos clave para garantizar la continuidad y eficiencia operativa en entornos de subestaciones eléctricas digitales.

1.1 Motivación

En el ámbito de las subestaciones eléctricas, la modernización de las redes de comunicaciones está encabezada por la Comisión Electrotécnica Internacional (IEC)

a través del estándar IEC 61850 en sus diversas versiones. Este protocolo establece las reglas necesarias para la estandarización de las redes de comunicaciones en las subestaciones y proporciona directrices para avanzar en la migración hacia subestaciones digitales. Existen propuestas en la literatura que sugieren el uso de SDN para mejorar la gestión de la red [LB16d; AAU15; LB16b; LB19; RNB21], así como el uso de funciones de red virtualizadas [RNB21], y la implementación de varios modelos matemáticos que permiten extrapolar las ventajas y desafíos de la red de datos a la red de comunicaciones de una subestación eléctrica.

La importancia de establecer rutas alternativas en respuesta a fallos en la red de datos de una subestación eléctrica se refleja directamente en la funcionalidad del sistema. Esto no solo ayuda a reducir los costos operativos asociados con el mantenimiento del servicio, sino que también previene el colapso del sistema en su totalidad. En consecuencia, el rendimiento y la estabilidad de la subestación están estrechamente ligados a la fiabilidad de su red de comunicación y su capacidad para tolerar fallos en los enlaces.

Ante la necesidad de proveer fiabilidad y tolerancia a fallos en la red de comunicaciones de las subestaciones eléctricas, en [LDB20] se plantean algoritmos que proporcionan rutas alternativas en caso de falla, minimizan la cantidad de saltos en la red para las nuevas rutas y proporcionan rutas alternativas para mantener la comunicación. A pesar de las alternativas planteadas en la literatura para la virtualización de subestaciones, aún no se ha presentado una solución que permita restablecer las demandas que generan las funciones de red en una subestación, cuando se producen fallas en los enlaces físicos o virtuales.

Para restablecer el servicio rápidamente después de un fallo, se formula un modelo matemático mediante un Programa Lineal Entero (ILP, por las siglas en Inglés de Integer Linear Programming). Un ILP que escoge la mejor ruta, soluciona el problema de funcionamiento y el restablecimiento de las demandas ante un fallo de enlace. Sin embargo, la complejidad computacional de resolver un ILP aumenta de manera exponencial a medida que crece el número de variables, restricciones, elementos en la red, y el número de posibles soluciones enteras o binarias. Esto hace que la búsqueda sea exhaustiva e impracticable en términos de tiempo.

Dado que se ha demostrado que el ILP no es eficaz en redes medianas y grandes, y no es versátil para manejar parcialmente las demandas, se propone una metaheurística basada en optimización por colonia de hormigas (ACO, por las sigla sen inglés de Ant colony optimization). Esta metaheurística no solo proporciona soluciones de alta calidad, sino que también equilibra la carga, entrega las demandas que pueden ser transportadas y revela las demandas que no pueden ser atendidas después de una falla cuando el ILP no es factible.

Una de las principales contribuciones de este trabajo de investigación radica en la virtualización de funciones de equipos de protección y control de subestaciones eléctricas (IED, por sus siglas en inglés de Intelligent Electronic Device), mediante el uso de contenedores optimizados en tamaño, tiempo de ejecución y uso de recursos. La virtualización de estas funciones se implementa en una red virtual de prueba, donde se miden los tiempos máximos permitidos para la comunicación de cada servicio, su tiempo de lanzamiento y el tráfico generado en cada petición. Estos parámetros desempeñan un papel fundamental, al establecer los límites en las restricciones de los modelos de optimización para encontrar rutas alternas.

1.2 Organización

A lo largo de esta sección, se detalla la estructura que guiará la presentación de los objetivos, metodología, resultados y conclusiones, proporcionando así una visión general de cómo se desarrollará la investigación.

Capítulo 2: En esta sección, se analiza la teoría propuesta para la virtualización de los servicios, se explora la manera en que la virtualización de funciones se implementa mediante el uso de contenedores, y se destaca el impacto significativo que puede resultar de la incorporación de estas funciones en contextos de gran envergadura. La base de esta investigación se centra en la optimización de rutas, lo que involucra la evaluación de métodos exactos y su progresión a lo largo del tiempo. Se examinan sus limitaciones para hallar soluciones viables en situaciones de alta complejidad del problema. Además, se presentan alternativas mediante el empleo de métodos metaheurísticos, como el enfoque de la colonia de hormigas.

Capítulo 3: En esta sección se define el problema de investigación en conjunto con los objetivos propuestos para el desarrollo de este trabajo de investigación.

Capítulo 4: En este capítulo se presenta el estado del arte de la virtualización de la red de datos de las subestaciones eléctricas, los métodos de virtualización de funciones de red propuestos en la literatura y los algoritmos encontrados para encontrar rutas cuando se experimentan fallos en los enlaces.

Capítulo 5: En esta sección se presenta la descripción detallada de los algoritmos implementados para resolver el problema de rutas en caso de fallo de enlace en la red de datos de una subestación eléctrica virtualizada. Además se presentan las cotas máximas establecidas de acuerdo con las emulaciones realizadas con base en la virtualización de funciones de red usando contenedores.

Capítulo 6: En esta sección se detallan los experimentos llevados a cabo, se presentan diversas instancias del problema y se describe el proceso de desarrollo de las pruebas realizadas, la evaluación y presentación de resultados. A partir de los experimentos realizados, se analiza en qué instancias el método exacto muestra un mejor rendimiento y se examina la relevancia de implementar la técnica metaheurística propuesta. Se exploran sus ventajas y desventajas, así como las soluciones apropiadas ante la ocurrencia de fallos en los enlaces de la red.

Capítulo 7: Se elaboran las conclusiones de este estudio tomando como base los resultados presentados en el capítulo previo. Asimismo, se inicia la discusión sobre la relevancia de la virtualización de funciones de red, que fue implementada en los experimentos. Se exploran las ventajas que se obtuvieron a través de la convergencia de estas propuestas en el contexto de las subestaciones eléctricas modernas.

Capítulo 8: En este capítulo se presenta un resumen conciso de cada uno de los capítulos anteriores, brindando una visión panorámica de la estructura y contenido del trabajo de investigación. Se establece una contextualización general sobre la importancia de la optimización de rutas y el impacto positivo de la virtualización en el contexto de las subestaciones eléctricas y la eficiencia operativa en redes eléctricas.

Marco teórico

Dado que las contribuciones principales de este trabajo de investigación hacen parte de dos campos del conocimiento: la virtualización de servicios y la optimización de rutas en subestaciones eléctricas, las siguientes secciones presentan los conceptos necesarios para los experimentos y resultados que se presentan en los capítulos 5 y 6.

2.1 Virtualización

La virtualización es un concepto tecnológico que implica crear una versión virtual de un componente físico. Esta versión virtual puede cumplir la misma función que el componente original y coexistir con otras instancias virtualizadas en el mismo equipo físico anfitrión. Permite ejecutar múltiples sistemas operativos y aplicaciones de forma simultánea en un mismo enrutador, conmutador, switch, servidor, etc. Con base en este principio se garantiza la escalabilidad, flexibilidad y agilidad en el ámbito de la infraestructura de tecnologías de la información, permitiendo una gestión eficiente de los recursos, la adaptación rápida a cambios y la implementación ágil de servicios y aplicaciones, todo ello sin requerir cambios significativos en el hardware subyacente [HS16; Bar+03].

2.1.1 Virtualización de equipos

La virtualización de equipos es llevada a cabo principalmente mediante dos tecnologías: máquinas virtuales y contenedores. Una máquina virtual es una representación completa de un sistema de computadora, incluyendo hardware, sistema operativo y aplicaciones. Ésta se ejecuta en un software de virtualización mediante el uso de un hipervisor [Hwa+13]. La figura 2.1 ilustra las componentes principales de una máquina virtual.

La capa de software, conocida como hipervisor, desvincula las máquinas virtuales de la máquina física y asigna dinámicamente los recursos a cada ordenador virtual según las necesidades. El sistema operativo y las aplicaciones no pueden distinguir un equipo virtual de un equipo físico porque los ordenadores virtuales se comportan

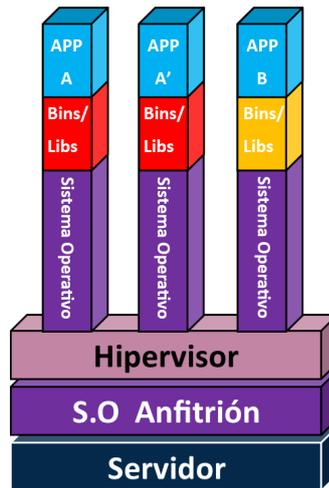


Figura. 2.1: Equipo físico con 3 máquinas virtuales

como ordenadores físicos reales, los procesos que ejecuta un ordenador virtual están limitados por los recursos entregados por la máquina anfitrión [VMw23].

De la misma forma que una máquina física, cada máquina virtual completa es totalmente independiente; esto permite que en un único ordenador físico, se puedan instalar de forma simultánea varios sistemas operativos y aplicaciones al configurar varias máquinas virtuales. Adicionalmente, este tipo de elementos virtuales son totalmente compatibles con las distribuciones Linux, versiones de Windows y entornos Mac (ver Figura 2.1).

A diferencia de las máquinas virtuales, los contenedores son una tecnología de virtualización a nivel de sistema operativo. Los contenedores comparten el mismo kernel del sistema operativo del equipo físico y permiten la ejecución de aplicaciones como tareas independientes dentro de la máquina anfitrión. Por lo tanto, pueden ser definidos como máquinas virtuales ligeras, menos exigentes con los recursos de los equipos donde se ejecutan, y con un mejor desempeño en el despliegue de servicios, requieren menor tiempo de lanzamiento y presentan una rápida ejecución.

Dado que pueden ser desplegados en cualquier sistema operativo, estos elementos usan como base imágenes livianas y complementan su funcionamiento con el uso del kernel de la máquina donde se corren los servicios. Esta característica hace que se minimice el consumo de memoria, dado que se encuentran imágenes del orden de los 4 Megabytes, esto hace que los sistemas sean eficientes, dinámicos y autónomos, a su vez garantiza que el software siempre se ejecute de la misma manera, independiente de dónde se despliegue (ver Figura 2.2) [Doc23].

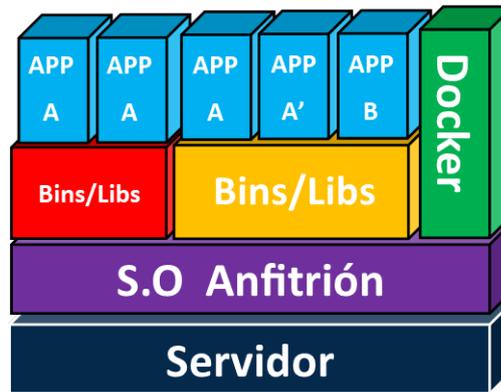


Figura. 2.2: Equipo físico con 6 contenedores

2.1.2 Virtualización de red

La virtualización de red representa un paradigma que facilita la coexistencia de múltiples redes virtuales en una misma infraestructura física. Dentro de un entorno de virtualización de redes, cada red virtual se compone de nodos virtuales y enlaces virtuales, formando así un subconjunto de los recursos de la red física subyacente [CB10]. Entre los ejemplos de virtualización de redes más comunes se destacan:

Redes Virtuales de Área Local (VLAN, por las siglas en inglés de Virtual Local Area Network): es una técnica de segmentación de redes que permite dividir una red de área local (LAN, por las siglas en inglés de Local Area Network) física en múltiples redes lógicas más pequeñas. Estas redes lógicas funcionan como si fueran LAN separadas, a pesar de compartir la misma infraestructura física, como cables y switches. Se utilizan en entornos de virtualización para aislar grupos de máquinas virtuales o contenedores, evitando la difusión de paquetes en la red física [ONC15].

Red Privada Virtual (VPN, por las siglas en inglés de Virtual Private Network): Es una tecnología que crea una conexión virtual segura y cifrada entre dos puntos en una red, a menudo a través de una red pública como Internet. Su principal objetivo es garantizar la privacidad y la seguridad de la comunicación, permitiendo a los usuarios acceder a recursos de red de forma remota como si estuvieran físicamente presentes en la ubicación de la red privada [PG02].

Redes Virtuales en la Nube: Son redes lógicas y aisladas que operan en infraestructuras de nube pública. Estas redes permiten a las organizaciones crear y administrar entornos de red personalizados y privados en la nube, lo que les da control sobre la conectividad, la seguridad y la distribución de recursos en la nube [Lee14].

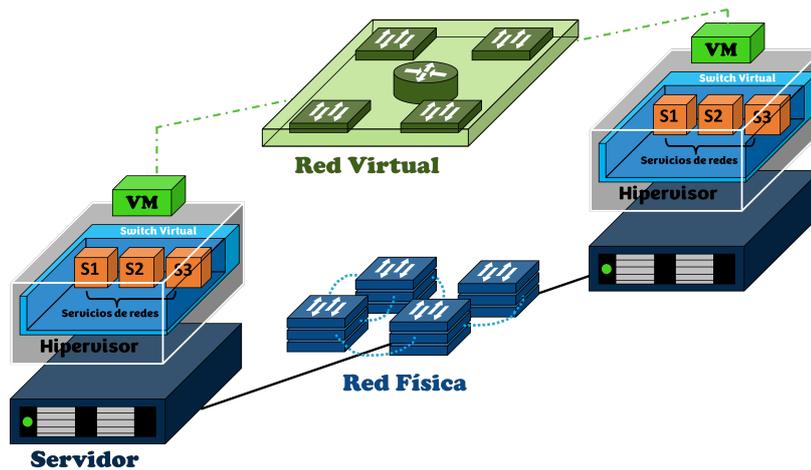


Figura. 2.3: Red virtual

Conforme se muestra en la Figura 2.3, cada equipo físico puede alojar diversas máquinas virtuales. Estas máquinas virtuales, a su vez, integran una serie de funciones subyacentes que replican las tareas de un equipo físico real. La red física se segmenta en múltiples redes virtuales que interconectan los dispositivos virtualizados, lo que posibilita la creación de redes distintas dentro de la misma infraestructura física.

La presente investigación se ha enfocado en aspectos específicos relacionados con las redes de subestaciones eléctricas, la virtualización y el restablecimiento de la red de datos en caso de falla. Sin embargo, se reconoce la creciente importancia de las Redes Definidas por Software (SDN) en la evolución de las infraestructuras eléctricas. Aunque SDN no ha sido el foco central de este estudio, se planea dedicar una página a este paradigma, que formará parte del trabajo futuro, ya que la integración de SDN podría potenciar y complementar las soluciones propuestas.

2.2 Redes Definidas por Software (SDN)

SDN es una arquitectura de red que utiliza software para controlar y gestionar de manera centralizada la infraestructura de red, incluyendo enrutadores, conmutadores y otros dispositivos. Las SDN permiten la separación del plano de control y el plano de reenvío de datos en los equipos físicos a diferencia de las redes tradicionales, donde el control y el reenvío de datos están integrados en los dispositivos de red [BEFEE16].

SDN proporciona una gran flexibilidad, ya que permite programar y ajustar el comportamiento de la red de manera centralizada y dinámica. También brinda agilidad al permitir la rápida implementación de servicios y políticas de red. Además,

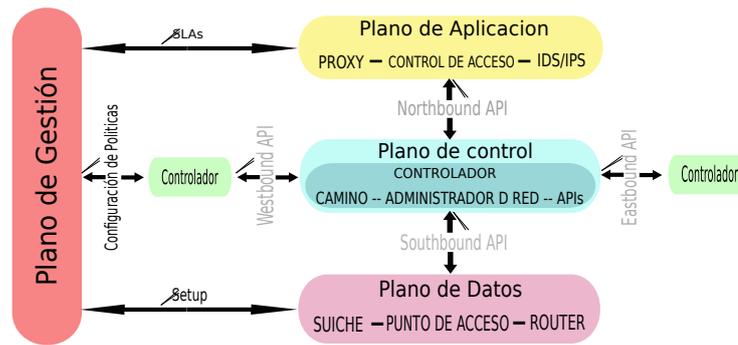


Figura. 2.4: Arquitectura SDN

optimiza el tráfico y simplifica la gestión de la red, lo que conlleva a mejoras en la seguridad.

El corazón de SDN es el controlador, un elemento central que toma decisiones sobre cómo se debe comportar la red y cómo se enrutan los datos. Este controlador es accesible a través de una API que permite a las aplicaciones y servicios comunicarse con él para programar el comportamiento de la red según sus necesidades.

En general, la arquitectura de SDN está basada en cuatro pilares fundamentales: (i) la capa de control es removida de los equipos de red, convirtiéndolos en simples equipos de reenvío, (ii) la decisión de reenvío de la información está basada en el flujo (es una secuencia de paquetes entre dos nodos (origen, destino)), (iii) el control lógico es trasladado a una entidad externa (servidor, maquina virtual, contenedor) denominada controlador, el cual está encargado de realizar la gestión y la programación de las reglas en cada uno de los equipos de reenvío y (iv) la red es programable a través de aplicaciones programadas en el controlador. Note que este proceso hace que las redes tradicionales se conviertan en redes auto gestionables, dado que pueden tomar decisiones y realizar acciones en tiempo real, adaptarse a las necesidades cambiantes de la red y reducir la carga de trabajo manual para los administradores de redes [Kre+14; LB16a].

Las ventajas ofrecidas por SDN convierten las redes de datos en estructuras dinámicas, flexibles y escalables; con una reducción aproximada de 58 % del CAPEX durante el proceso de creación y mantenimiento de la red [Nau+12] [BEFEE16], .

2.3 Virtualización de las funciones de red (NFV)

La NFV representa una propuesta cuyo objetivo es convertir las funciones de red, tradicionalmente enlazadas a hardware específico, en soluciones de software. Estas

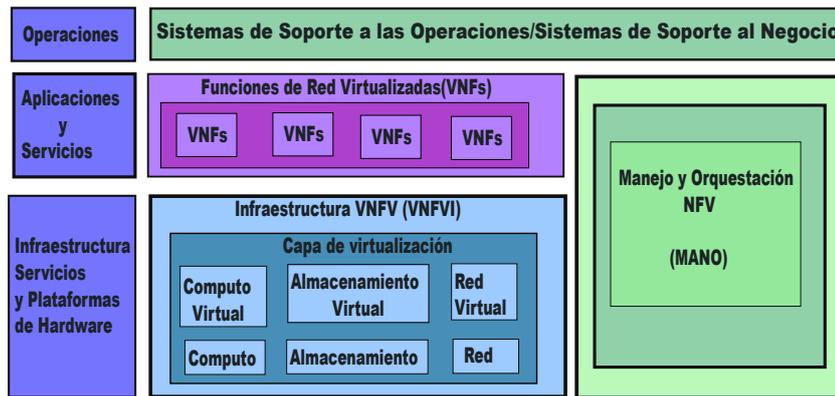


Figura. 2.5: Arquitectura NFV

soluciones pueden operar en una diversidad de dispositivos de propósito general (switches, enrutadores, middleboxes, servidores, etc) permitiendo así una mayor flexibilidad y eficiencia en la implementación de las redes [HB16].

La NFV reduce la dependencia del hardware dedicado, permitiendo a las compañías gestionar, administrar, crear e instalar servicios de manera ágil y eficiente [MHS21]. Las implementaciones con NFV permiten el uso eficiente de la infraestructura y ofrecen libertad a los Proveedores de Servicios de Internet (ISPs, por sus siglas en Inglés de Internet service providers) para crear, implementar y gestionar servicios de red sin preocuparse por la configuración de dispositivos de red específicos del proveedor. Las funciones de red se virtualizan de tal manera que pueden encadenarse para crear y desplegar servicios de red sobre la marcha, ofreciendo flexibilidad para adaptarse rápidamente a la innovación tecnológica y proporcionar un mejor retorno de inversión para los ISPs en comparación con los dispositivos basados en hardware [RAB19].

Dadas las ventajas de la NFV y su alcance en la modernización de la red, el Instituto Europeo de Normas de Telecomunicaciones (ETSI, por las siglas en Inglés de European Telecommunications Standards Institute) plantea una arquitectura básica que integra de forma general los principales componentes de NFV (funciones, orquestadores y administración) [KMK22; HC22]. En la figura 2.5 se puede evidenciar la estructura propuesta por ETSI, la cual puede ser implementada sobre un hardware físico o un elemento virtualizado [RAB19].

Dentro del marco de la arquitectura de Virtualización de Funciones de Red (NFV), las tres capas fundamentales: Operaciones, Aplicaciones y Servicios, e Infraestructura de Servicios y Plataformas de Hardware, delinean de manera lógica los componentes que integran el entorno de NFV.

La capa de Operaciones se orienta hacia la eficiente gestión y operación de la infraestructura NFV y los servicios desplegados. En ella, se encuentran el Orquestador NFV, encargado de coordinar la implementación y escalado de las Funciones de Red Virtual (VNFs), y la Gestión y Orquestación de Red (MANO), que proporciona herramientas para gestionar el ciclo de vida de las VNFs y servicios.

Por otro lado, la capa de Aplicaciones y Servicios se centra en las aplicaciones y servicios específicos desplegados en la infraestructura NFV. Aquí, las Funciones de Red Virtual (VNFs) representan aplicaciones específicas, como cortafuegos o enrutadores, implementadas como funciones virtualizadas en máquinas virtuales. Además, el Catálogo de Servicios alberga descripciones y configuraciones de servicios específicos.

Finalmente, la capa de Infraestructura de Servicios y Plataformas de Hardware proporciona la base física y virtual para la ejecución de aplicaciones y servicios NFV. Incluye recursos virtualizados, como computacionales, de almacenamiento y de red, que se asignan dinámicamente a las VNFs según las necesidades. Esta capa también integra el hipervisor, encargado de gestionar las máquinas virtuales y asignar recursos físicos, así como la red virtualizada, que segmenta y conecta dispositivos virtualizados permitiendo la creación de redes distintas dentro de la infraestructura física.

Esta organización en capas facilita la comprensión y administración de los diversos aspectos de la virtualización de funciones de red, desde la operación y gestión hasta la ejecución de servicios específicos en una infraestructura flexible y escalable. La implementación de NFV puede representar una reducción del OPEX y el CAPEX en valores superiores al 50 % cuando se incrementa el número de elementos en la red [AS16].

2.4 Optimización de rutas

La optimización de rutas se refiere a la práctica de descubrir las rutas o caminos más eficientes para llevar datos en una red, desde un punto de origen hasta un destino. En el contexto de redes de datos, la optimización de rutas se emplea para elegir las rutas ideales que posibiliten la transmisión eficaz de datos, tráfico y recursos, con el propósito de maximizar el rendimiento de la red y disminuir tanto la latencia como los tiempos de espera [MGK21].

El criterio para determinar la ruta óptima está condicionado por la estrategia de optimización empleada, y esta elección puede diferir según el tipo de red y los

objetivos concretos de la implementación, entre estas se destacan las métricas de ruta, la carga de tráfico, la topología de la red o algoritmos de enrutamiento usados. En este marco teórico el foco se centra en los modelos ILP, las técnicas heurísticas y las metaheurísticas.

2.4.1 Optimización de rutas usando métodos exactos

Los métodos exactos usados para resolver problemas de optimización lineales están basados en la programación lineal y sus variantes. ILP es una técnica de optimización matemática empleada para abordar problemas en los cuales se busca encontrar la mejor solución posible de un conjunto de opciones, sujetas a restricciones específicas, con variables que toman valores enteros. En particular, se enfoca en problemas en los que el objetivo es maximizar o minimizar una función lineal (llamada función objetivo), diseñada para reflejar de manera cuantitativa las metas que se quieren lograr; se define en función de las variables de decisión, que son las incógnitas del problema que se pueden ajustar o cambiar para obtener un resultado óptimo [Mug98]. Estas variables pueden tomar valores enteros (programación lineal entera), o valores enteros y reales (programación lineal real) dependiendo del contexto del problema.

Los valores que pueden tomar las variables de decisión están limitadas por las restricciones impuestas en el problema; las cuales definen las limitaciones o reglas que restringen las posibles soluciones que se pueden considerar en el problema de optimización. Las restricciones pueden ser expresadas como ecuaciones o desigualdades y se utilizan para modelar las limitaciones físicas, operativas o de recursos del problema en cuestión [Hua+23].

Cuando la función objetivo y las restricciones en un problema combinatorio consisten en números enteros y son de naturaleza lineal, se le denomina un problema de Programación Lineal Entera (ILP).

Cualquier problema general de programación lineal puede presentarse de la siguiente forma:

$$\begin{array}{ll}
\text{Minimizar} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\
& a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1 \\
& a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\
\text{Sujeto a} & \vdots + \vdots + \dots + \geq \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \\
& x_1, \quad x_2, \quad \dots, \quad x_n \geq 0
\end{array} \tag{2.1}$$

en este planteamiento $c_1x_1 + c_2x_2 + \dots + c_nx_n$ es la función objetivo que debe minimizarse. Los coeficientes c_1, c_2, \dots, c_n son los coeficientes de costo (escalares conocidos), y x_1, x_2, \dots, x_n son las variables de decisión que deben determinarse. La desigualdad $\sum_{j=1}^n a_{ij}x_{ij} \leq b_i$ indica la i -ésima restricción. Los coeficientes a_{ij} para $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ se llaman coeficientes tecnológicos y forman la matriz de restricciones **A**.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

El vector columna cuya i -ésimo componente es b_i , al cual se le denomina vector del lado derecho, representa los requerimientos mínimos que deben satisfacerse. las restricciones $x_1, x_2, \dots, x_n \geq 0$ son las restricciones de no negatividad. Un conjunto de variables x_1, \dots, x_n que satisfacen todas las restricciones se denomina punto factible o vector factible. el conjunto de todos estos puntos se llama región factible o espacio factible. De acuerdo con la descripción anterior, podemos afirmar que un problema de programación lineal es un método que busca, entre todos los vectores factibles, aquel que minimice o maximice la función objetivo.

El siguiente ejemplo ilustra la descripción anterior.

$$\begin{array}{ll}
\text{Minimizar} & 2x_1 + 5x_2 \\
& x_1 + x_2 \geq 6 \\
\text{Sujeto a} & -x_1 - 2x_2 \geq -18 \\
& x_1, \quad x_2, \geq 0
\end{array} \tag{2.2}$$

En este ejemplo se cuenta con dos variables de decisión, x_1, x_2 , y la función objetivo que debe minimizarse es $2x_1 + 5x_2$. La solución se presenta gráficamente en la Figura 2.6, donde x_1 estará representado en el eje de las abscisas y x_2 ocupa el eje de las ordenadas. En este gráfico se ilustran las restricciones y la región factible. El problema de optimización consiste en encontrar un punto dentro de la región factible que minimice el valor de la función objetivo.

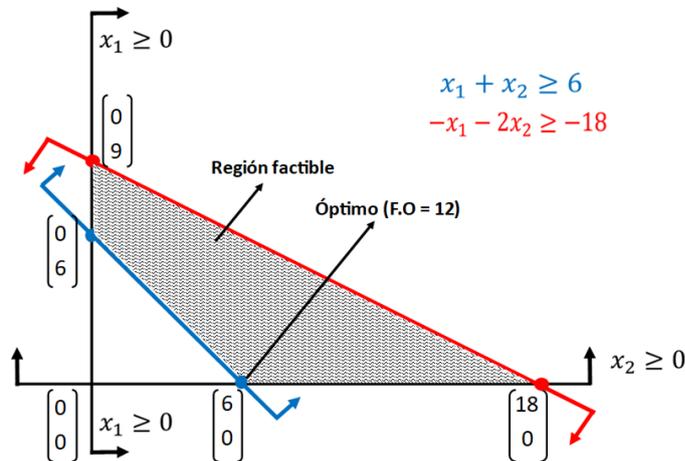


Figura. 2.6: Solución gráfica del problema ILP.

Los programas lineales son capaces de proporcionar una gran cantidad de información, en vez de una sola solución. Los ILP son resueltos con algoritmos exactos debido a la naturaleza de su enfoque de resolución. El uso de un método exacto para resolver un problema, significa que se busca encontrar la solución óptima de manera precisa y sin aproximaciones.

Los métodos exactos se implementan a través de herramientas de software (solvers) que ejecutan algoritmos típicos como *simplex*, *branch and bound* y *branch and cut*. Algunos solvers disponibles, como GLPK [PRO23], son de código abierto, mientras que otros como GUROBI [GUR23] y CPLEX [IBM23] son propietarios, aunque GUROBI ofrece una licencia académica con amplio margen de trabajo.

2.4.2 Métodos heurísticos y metaheurísticos

Las heurísticas y metaheurísticas son herramientas poderosas para abordar problemas de optimización en los casos donde los métodos exactos son ineficientes. Un método heurístico es una técnica de optimización basada en la intuición, la creatividad, la experiencia y el conocimiento del problema. Los métodos heurísticos usan reglas no estandarizadas para guiar la búsqueda hacia una mejor solución; aunque carecen de elaboración, estructura y matemática sistemática, se presentan como una importante alternativa en la obtención de buenas soluciones en tiempos cortos para pequeños problemas, sin garantizar el óptimo global. Entre la técnicas

heurísticas comunes encontramos los algoritmos voraces (greedy), el enfoque de dividir un problema en pequeñas instancias y solucionarlo por partes (*Divide et vinci*), algoritmos de búsqueda local y algoritmos constructivos [Tal09].

A diferencia de las heurísticas, las metaheurísticas son técnicas que enmarcan un problema de optimización de manera general y abstracta. Generalmente, se aplican en problemas grandes de difícil solución o en casos donde faltan detalles del problema. Estas técnicas incorporan funciones de exploración e intensificación claramente definidas, junto con un componente de aleatoriedad que evita que la búsqueda quede estancada en óptimos locales. Similar a las heurísticas, esta técnica no garantiza la solución óptima, pero ofrece soluciones de calidad en intervalos de tiempo considerablemente más cortos. Las técnicas metaheurísticas han experimentado una notable aceptación en los últimos años. En su mayoría están basadas en procesos industriales, comportamiento de enjambres y colonias, procesos naturales y sistemas físicos. Entre los métodos más comunes se encuentra la optimización por colonia de hormigas (ACO, por las siglas en inglés de Ant Colony Optimization), la búsqueda tabú (TS, por las siglas en Inglés Tabu Search), la optimización por enjambre de partículas (PSO, por las siglas en Inglés de Particle Swarm Optimization), los algoritmos genéticos (GA, por las siglas en inglés de Genetic Algorithm), etc [Tal09].

Optimización de rutas usando Metaheurísticas: caso de colonia de hormigas

La optimización de rutas basada en colonias de hormigas y sus variantes, ha sido ampliamente utilizada para resolver problemas complejos que requieren soluciones rápidas [Che+18; Che+19; Zha+19; SRK21; SOH22; Den+22]. Por este motivo, en este documento se describe de manera detallada el método y se resalta la importancia de sus variables.

La teoría de la optimización por colonia de hormigas fue introducida por Marco Dorigo en su tesis de doctorado como un enfoque para encontrar soluciones a problemas complejos relacionados con la búsqueda de caminos o rutas más cortas en un grafo dado, con base en el comportamiento de las hormigas [DG97]. Se inspira en gran medida en el comportamiento de una hormiga real mientras busca su fuente de alimento, desde el momento en que la encuentra hasta que regresa a su nido.

Descripción del algoritmo

Durante la búsqueda, cada hormiga explora caminos de manera aleatoria o al azar. Una vez que encuentra su alimento, regresa a su nido dejando un rastro

de feromona. Este rastro actúa como un canal de comunicación que guiará a las siguientes hormigas hacia el lugar donde encontró su alimento, a continuación se define el escenario que soporta el algoritmo.

Hormiga artificial: Una hormiga artificial es un agente diseñado para buscar soluciones computacionales de menor costo en la resolución de diversos problemas.

Feromona artificial: Las feromonas artificiales son valores numéricos que se emplean para expresar la preferencia o calidad de las rutas seguidas por las hormigas artificiales en un grafo.

Grafo: Los grafos son una herramienta matemática esencial para modelar relaciones complejas en diversos campos. Al usar nodos para representar objetos y aristas para representar conexiones entre ellos, los grafos permiten analizar y resolver problemas de una manera más intuitiva y efectiva. Por ejemplo, en la planificación de rutas en una red, los nodos pueden representar equipos y las aristas los enlaces entre ellos. Los tipos comunes de grafos incluyen grafos dirigidos y no dirigidos, grafos ponderados, grafos bipartitos, grafos cíclicos y grafos acíclicos [Wes+01].

El entorno inicial se puede observar en la figura 2.7.

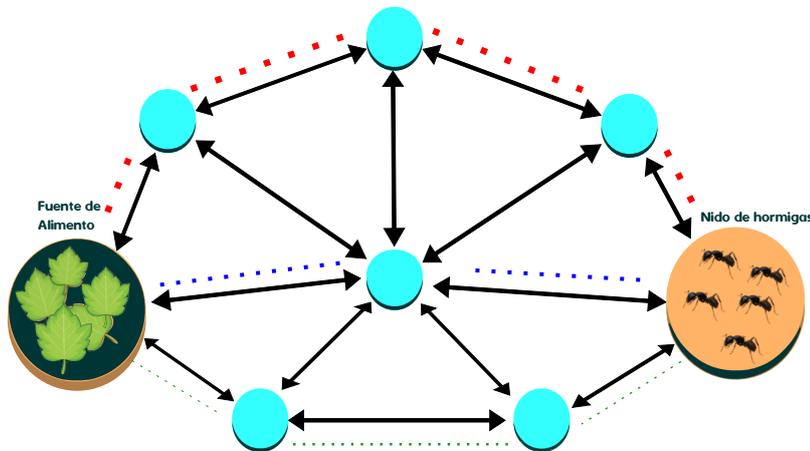


Figura. 2.7: Grafo inicial.

Una vez que las hormigas exploradoras han completado sus recorridos, ya sea hacia la derecha o hacia la izquierda según su elección inicial, y han alcanzado su principal objetivo de encontrar alimento, regresan a su nido. En su trayecto de regreso, depositan una sustancia llamada feromona, la cual actúa como una guía para las futuras hormigas exploradoras (ver Figura 2.8 lado A). En la Figura 2.8 B se observa la gran cantidad de feromonas depositadas en los enlaces y cómo pueden influenciar en la decisión de las hormigas al elegir su camino. Se puede decir que

si un camino cuenta con más feromonas, representa el camino más corto que las hormigas deben recorrer para llegar a su fuente de alimento con éxito.

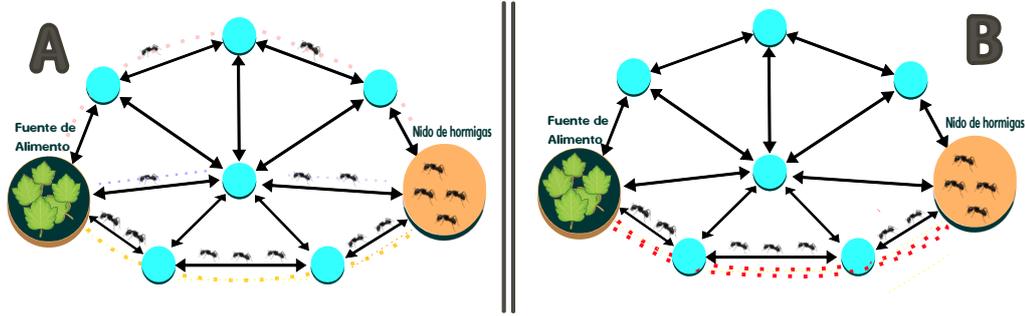


Figura. 2.8: Actualización Feromona.

La decisión de cada hormiga depende de dos variables fundamentales: la información heurística, que representa la preferencia inherente de las hormigas por un camino basada en la información del problema, y la información de rastros de feromona que refleja la deseabilidad aprendida a partir de los movimientos de otros individuos.

El algoritmo requiere del cálculo de la probabilidad que la hormiga k -th seleccione el nodo j partiendo desde el nodo i esta determinada por:

$$P_{ij}^k = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{j \in N_i^k} (\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}$$

Donde α Es un parámetro que representa la influencia relativa de feromonas, η_{ij} representa la preferencia de la hormiga para moverse del nodo i al nodo j y es definido como $\eta_{ij} = \frac{1}{d_{ij}}$ y d_{ij} es la longitud del enlace o el retardo entre los nodos i y j . $\beta > 0$ es un parámetro heurístico que indica la influencia de la información heurística. N_i^k Es el número de nodos disponibles que tiene la hormiga k -th para moverse al nodo i . K Es el número total de hormigas en el algoritmo.

Dado que por cada recorrido que realiza una hormiga por un camino determinado se deposita feromona en el camino, el proceso de actualización de feromona τ_{ij} para la nueva hormiga k -th en el enlace entre i y j esta definido por la siguiente expresión:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^K \Delta\tau_{ij}^k$$

Donde $0 < \rho < 1$ representa el coeficiente de evaporación de feromonas y $\Delta\tau_{ij}^k$ representa la cantidad de feromonas que depositadas por la k -ésima hormiga del nodo i al nodo j . Los valores de $\Delta\tau_{ij}^k$ son $\frac{1}{L_K}$ si la hormiga k transitó del nodo i al nodo j y 0 en otros casos.

A continuación, se presenta el resumen de los parámetros del algoritmo.

Tabla. 2.1: Parámetros del algoritmo ACO.

Parámetro	Descripción
K	Número de hormigas en la colonia.
N_i^k	Número de nodos disponibles para la hormiga k partiendo del nodo i .
d_{ij}	Distancia (o peso) en el enlace que une el nodo i con el nodo j .
η_{ij}	Preferencia de una hormiga para moverse del nodo i al nodo j con base en la información heurística y es definida como $\eta_{ij} = \frac{1}{d_{ij}}$.
τ_{ij}	Rastro de feromonas del nodo i al nodo j .
α	Parámetro que controla la influencia relativa del rastro de feromona ($0 < \alpha < 1$).
β	Parámetro que controla la influencia de la información heurística ($0 < \beta$).
P_{ij}^k	Probabilidad de que la hormiga k elija el nodo j partiendo del nodo i .
L_k	Es la longitud del camino en términos de número de enlaces recorridos por la hormiga k -th desde el origen al destino.
$\Delta\tau_{ij}^k$	Parámetro que indica la cantidad de feromonas que depositadas por la k -ésima hormiga del nodo i al nodo j .
ρ	Coefficiente de evaporación de feromonas.

2.5 Subestaciones eléctricas

Las subestaciones eléctricas son sistemas robustos compuestos por transformadores de alta tensión, transformadores de baja tensión, equipos de medida, sistemas de protección y control y sistemas de monitoreo. Su función principal es la transformación de tensión a niveles adecuados para su distribución mediante VTs, CTs y PTs. Esta característica permite hacer la distribución de energía para llevar la electricidad a áreas locales y regionales.

Además de transformar y distribuir la electricidad generada en centrales convencionales, las subestaciones son usadas para conectar fuentes de energía renovable, como parques eólicos y plantas solares a la red eléctrica. La subestación permite el control y regulación del flujo de energía eléctrica. Cuenta con equipos de control como interruptores y disyuntores que se utilizan para administrar y proteger la red eléctrica contra cortocircuitos y sobrecargas, asegurando un suministro confiable y seguro.

En la figura 2.9 se presenta un modelo simplificado de una subestación eléctrica, específicamente una subestación tipo TI-1. En este esquema, los elementos están

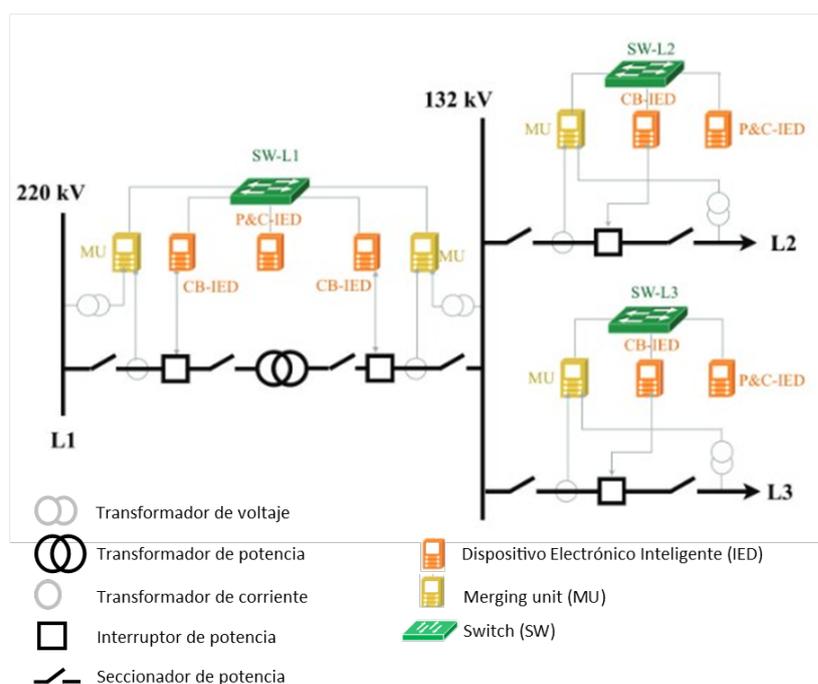


Figura. 2.9: Arquitectura subestación eléctrica tipo T1-1

interconectados mediante switches de red. La subestación incluye dispositivos electrónicos inteligentes (IEDs) que supervisan y alertan sobre posibles problemas en la infraestructura. Además, se observan equipos de monitoreo de señales analógicas (MU, por sus siglas en inglés de Merging Unit), transformadores y seccionadores, que cumplen funciones especiales en la conversión y finalización del trabajo para este caso particular.

2.6 Subestaciones eléctricas digitales

Una subestación eléctrica digital representa un nuevo paradigma que fusiona las ventajas de las redes de datos con las operaciones del sistema, abarcando áreas clave como protección, control y supervisión [Far09]. La digitalización de esta infraestructura conlleva la incorporación de diversos dispositivos interconectados mediante tecnología Ethernet. La automatización de la red de las subestaciones se rige por el estándar IEC 61850 [NPH14], que asegura la interoperabilidad entre todos los dispositivos, independientemente de su fabricante (esto incluye paradigmas emergentes como SDN, NFV y VM).

Dentro del estándar IEC 61850, se presenta un modelo jerárquico resumido en tres niveles principales: (i) Nivel de estación, (ii) Nivel de bahía y (iii) Nivel de proceso.

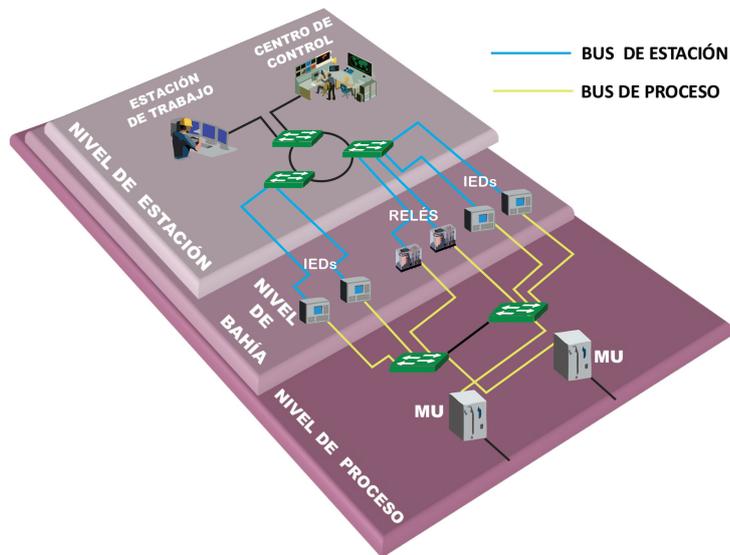


Figura. 2.10: Arquitectura subestación eléctrica digital

El nivel de proceso está compuesto por equipos de patio, como seccionadores, equipo de transformación (CT, VT, PT), elementos de protección (interruptores) y dispositivos de conversión y posterior transmisión de señales analógicas (MU, por las siglas en Inglés de Merging Unit).

El nivel de bahía abarca elementos de protección y control denominados dispositivos electrónicos inteligentes (IED, por las siglas en inglés de Intelligent electronic device). Su función consiste en alertar si se detecta una inconsistencia en los datos enviados por la MU. Estos dispositivos contribuyen a tomar decisiones en caso de posibles sobrecargas, y en situaciones de emergencia, se encargan de aislar secciones del sistema.

El nivel de estación incluye los componentes necesarios para supervisar y controlar todo el sistema. Dispone de múltiples interfaces de máquina-humano (HMI) que interactúan con diversos procesos de la subestación, controlando variables de todos los niveles. En este nivel se ubican los elementos que permiten la interconexión y el enrutamiento de la subestación digital en su conjunto [LB16a].

La interconexión entre estos niveles se lleva a cabo mediante dos redes denominadas **bus de proceso** y **bus de estación** (ver Figura 2.10). El bus de estación establece la conexión entre el nivel de bahía y el nivel de estación mediante una topología en anillo. Por su parte, el nivel de bahía se conecta con el nivel de proceso a través del bus de proceso, empleando una red en anillo que garantiza la redundancia y la resiliencia del sistema.

Tabla. 2.2: Descripción de los diferentes tipos de servicio.

Servicio	Descripción
GOOSE	Evento Genérico Orientado a Objetos de Subestación. Comunicación Publicador-Suscriptor definida en la Parte 8-1 de la norma IEC 61850. Principalmente utilizada para enviar información crítica en tiempo real entre dispositivos electrónicos inteligentes.
SV	Valor Muestreado. Comunicación Publicador-Suscriptor definida en la Parte 9-2 de la norma IEC 61850. Principalmente utilizada para enviar información de corriente y voltaje muestreada digitalmente a 4000 muestras por segundo para sistemas de 50Hz y 4800 muestras por segundo para sistemas de 60Hz.
MMS	Especificación de mensaje del fabricante. Comunicación Cliente-Servidor definida en la Parte 8-1 de la norma IEC 61850. Principalmente utilizada para enviar estados, alarmas y comandos de control entre el cliente (por ejemplo, HMI) y el servidor (por ejemplo, IED)

Para asegurar la operatividad integral de los distintos niveles como una única estructura, se implementa una comunicación eficaz entre los diversos equipos que conforman el sistema, haciendo uso de la red Ethernet subyacente. En esta red, se despliegan diversas funciones y servicios destinados a supervisar y llevar a cabo acciones específicas en la red. Entre los servicios mas importantes se encuentran, los mensajes de alta prioridad GOOSE (Eventos de Subestación Orientados a Objetos Genéricos), utilizados para abrir circuitos, transmitir el estado de equipos que requieren atención inmediata, y configurar el sistema, según se describe en [XXH15]. También se cuentan los servicios MMS (MMS, por sus siglas en ingles de Manufacturing Message Specification), responsables de mantener una comunicación continua sobre el estado de la red como se explica en [Rid+21]. Por último, se encuentran los valores de muestreo (SV, por sus siglas en inglés de Samples Values), diseñados para supervisar el estado de los equipos de patio y transmitir mediciones digitalizadas de tensión, corriente y potencia, recopiladas por las MU, según se detalla en [Hon+21].

En la tabla 2.2 se presenta un resumen de la descripción de los servicios en el estándar [Mis+15]. Dado que todas estas funciones pueden ser virtualizadas, se abre un amplio campo para las tecnologías NFV, SDN y propuestas de optimización de rutas en este novedoso concepto de subestación virtualizada [Gut+23; LB16a].

Planteamiento del problema

Con base en [DS19], en este trabajo de investigación se propone utilizar la técnica de optimización colonia de hormigas para abordar el desafiante problema de optimización de rutas alternativas en la red de una subestación eléctrica cuando falla un enlace. La elección de la colonia de hormigas se basa en su capacidad para adaptarse a restricciones complejas, su amplio uso en la optimización de rutas [SOH22; Den+22; SRK21; SRK21], la adaptabilidad a restricciones de tiempo y capacidad, así como su historial positivo en aplicaciones similares de enrutamiento.

3.1 Definición del problema de investigación

La evolución de las subestaciones eléctricas tradicionales a subestaciones eléctricas digitales destaca la importancia crucial de la red de datos en la infraestructura de la subestación. Esta red se convierte en un elemento central que requiere atención inmediata, ya que sustenta la gestión, el control y el monitoreo de la subestación. Un fallo en la red de datos se traduce directamente en un fallo del sistema. Por lo tanto, en la presente investigación se propone un modelo ILP destinado a encontrar rutas alternativas para cada enlace de forma offline, con el fin de lograr el restablecimiento de la comunicación en caso de fallos en los enlaces (uno a la vez) de la red de datos, con el objetivo de garantizar una comunicación efectiva entre los diversos equipos presentes en la subestación.

Además, se limita la implementación al escenario con una topología tipo SpiderWeb. Esta elección se justifica debido a (i) su reciente inclusión en el panorama de investigaciones [LB19], (ii) la conveniencia que ha demostrado en comparación con otras topologías y (iii) la falta de sentido en optimizar rutas en redes con topología de anillo doble dada la obviedad en la elección de rutas alternativas, donde la existencia de dos caminos distintos hace que la selección de la ruta secundaria sea evidente y directa. La implementación de una topología en malla en este contexto sería tanto costosa como poco realista, debido a la naturaleza altamente interconectada de esta estructura y a los considerables gastos asociados con su establecimiento y mantenimiento.

3.2 Objetivos

Objetivo general

Diseñar y llevar a cabo una estrategia de asignación de rutas en un entorno virtualizado, con el propósito de satisfacer la demanda de servicios, especialmente en el caso de fallos en los enlaces de una subestación eléctrica. Este proceso se llevará a cabo mediante la implementación de un testbed a escala, permitiendo la validación práctica de la eficacia de la estrategia propuesta.

Objetivos específicos

- Realizar una revisión y posterior clasificación de la literatura existente en el ámbito de virtualización y asignación de rutas en las redes de datos de las subestaciones eléctricas.
- Definir los servicios que serán desplegados en el caso de uso de una subestación eléctrica.
- Formular el problema para encontrar rutas alternativas eficientes para el despliegue de los servicios definidos en la infraestructura física de la subestación en caso de fallo de un enlace.
- Implementar una estrategia de solución basada en técnicas de optimización para el modelo propuesto.
- Evaluar el desempeño de los algoritmos propuestos, comparando los resultados entre ambas propuestas en un entorno emulado, simulando fallas en la red de datos de una subestación eléctrica.

Estado del arte

La transición de las subestaciones eléctricas convencionales a las digitales implica cambios significativos y la adopción de diversos métodos existentes, esta investigación estructura el estado del arte en dos secciones principales. En un primer momento, se examina el proceso de virtualización en las redes de subestaciones eléctricas, considerando las arquitecturas propuestas, la flexibilidad de los enfoques planteados y el progreso en las implementaciones basadas en el estándar IEC 61850. En una segunda parte, se aborda la optimización de las rutas a través de una combinación de métodos heurísticos, metaheurísticos y exactos.

4.1 Virtualización en redes de subestaciones eléctricas

La virtualización en las redes de subestaciones eléctricas es un enfoque que permite la aplicación de técnicas de virtualización como SDN, NFV, VM y LVM en las subestaciones. La virtualización en las redes de subestaciones eléctricas ha sido poco tratado en la literatura debido a la creación reciente del comité para la modernización de la infraestructura (IEC 61850), por lo tanto, las primeras investigaciones en este tema y las posibles soluciones se han desarrollado en años recientes. Estas soluciones están relacionadas con el tipo de virtualización que se quiera aplicar, ya sea virtualizando la red misma o las funciones de red.

En [LB16c] sugieren actualizar la red incorporando los paradigmas SDN y NFV con el fin de mejorar la automatización, gestión y operación de las subestaciones eléctricas. Aunque el estándar sugiere una arquitectura jerárquica, los autores presentan una propuesta donde todas las categorías de la subestación pueden ser representadas en un modelo concéntrico llamado solución inteligente para la red de la subestación (S3N) [LB16a].

En la arquitectura S3N el elemento central corresponde a todos los dispositivos de interconexión, principalmente switches Ethernet y cableado. Este componente es llamado S3N-CONNECT. El S3N-CONNECT permite la comunicación entre todos los componentes que están directamente relacionados con la operación de la red de comunicaciones, entre los que se tienen: (i) dispositivos de protección y control



Figura. 4.1: Modelo S3N (Imagen tomada para la edición de [LB16a])

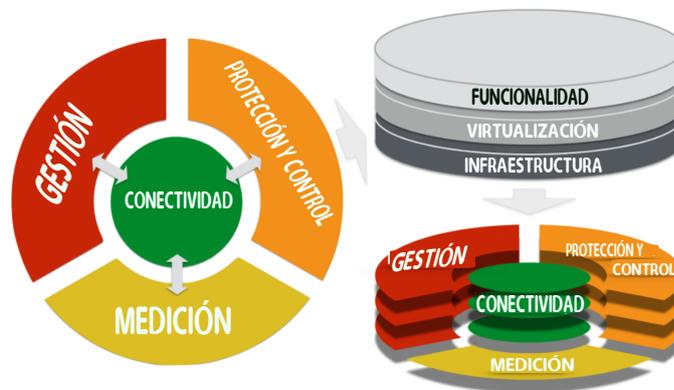


Figura. 4.2: Modelo S3N ampliado (Imagen tomada para la edición de [LB16a])

(IEDs, actuadores) agrupados en un componente llamado protección y control o S3N-PROTECT, (ii) dispositivos de administración (Recopilador de eventos, monitoreo) los cuales son agrupados en el componente denominado administración o S3N-MANAGE y (iii) dispositivos de medida (MUs) agrupados en el componente llamado mediciones o S3N-MEASURE.

De acuerdo con [LB16a], el modelo presentado en la Figura 4.1 puede representar la interacción de la red de comunicaciones en su estado actual, sin embargo, este modelo puede ampliarse a través de capas de abstracción, con el objetivo de ofrecer mayor flexibilidad a la estructura propuesta y concebir los posibles cambios que pueda experimentar esta red de comunicaciones en el futuro.

La figura 4.2 muestra una serie de capas denominadas (i) capa de infraestructura, la cual esta compuesta de una fuente física de computación de recursos, instrumentación, red y recursos de almacenamiento a través del uso de hardware abierto (con soporte de virtualización), (ii) la capa de virtualización que proporciona un entorno

de tiempo de ejecución para recursos virtualizados, reduce los tiempos de provisión y mantenimiento y aumenta la flexibilidad del sistema. (iii) La capa de funcionalidad corresponde al dominio de aplicaciones virtuales (funciones) que pueden ser consumidas por recursos virtualizados (servidores virtuales, IEDs virtuales o red virtual), ubicados en la capa de virtualización. Este aporte sugiere ver la subestación como una red controlada desde un punto central; aunque es una propuesta innovadora, no entrega detalles de la implementación.

Con base en la necesidad de incorporar modelos virtualizados en 2017 se plantea el uso de switches virtuales y máquinas virtuales. En [WKR17] los autores detallan los posibles escenarios para la virtualización, además de la creación e implementación de (IED) virtuales. La contribución principal de esta iniciativa se centra en utilizar los IEDs virtualizados para interactuar con equipos físicos. Sin embargo, la solución propuesta enfrenta dificultades al evaluar las funciones propias de subestaciones en los IED virtuales, dado que aunque la comunicación no se ve interrumpida, los tiempos obtenidos superan las pautas establecidas en el estándar.

En 2020 [LDB20] propone hacer uso de SDN en las redes de comunicación de las subestaciones eléctricas para implementar innovadoras técnicas de redundancia basadas en árboles multicast redundantes. Este enfoque busca reemplazar los mecanismos de redundancia tradicionales presentes en las redes de comunicación de subestaciones eléctricas y reducir el CAPEX y el OPEX.

La estrategia de redundancia propuesta se basa en árboles redundantes, acompañada de un modelo ILP que permite el despliegue de árboles multicast redundantes en la red, con el objetivo de minimizar la cantidad de enlaces de red utilizados. Estos árboles redundantes facilitan la transmisión de tráfico duplicado en la misma LAN subyacente de manera aislada, lo que asegura una recuperación instantánea en caso de fallos y mejora la utilización de los recursos de la red. Los resultados de la simulación demuestran que este enfoque alcanza un rendimiento similar al de los mecanismos de redundancia tradicionales, utilizando solamente la mitad de los recursos de red (es decir, en comparación con el Protocolo de Redundancia Paralela, PRP). El modelo planteado precalcula dos caminos diferentes para cada demanda de forma predeterminada, pero no permite llevar parcialmente las demandas.

Los modelos presentados se han probado bajo simulación sin tener en cuenta redes y tráfico real de la red. En [Rös+22] los autores presentan un modelo para una co-simulación (simulación y equipos reales funcionando sobre el mismo escenario) de retardos que combina arquitectura SDN y virtualización usando contenedores. Para el modelo general, se combinaron descripciones validadas individualmente y se derivó un modelo relevante para el tiempo de transferencia IEC 61850.

El modelo validado se usa para estimar el tamaño de red virtual para co-simulación y se concluye que para no violar el tiempo de transferencia, se deben usar hasta 4 conmutadores de bus de proceso con 3 flujos de señales cada uno. Es importante destacar que el modelo utilizado y adaptado es válido solamente en estado estable, donde la tasa de servicio supera la tasa de llegada. Este estado estable describe cómo se calcula el retardo estimado y advierte que en otro estado, el retardo podría volverse infinito debido al exceso en las colas de llegada.

En la Tabla 4.1, se muestra la categorización de las propuestas encontradas que incorporan tecnologías como SDN y NFV en el contexto de subestaciones eléctricas.

Tabla. 4.1: Propuestas de virtualización en subestaciones eléctricas.

Artículo	Contribución	Virtualización	Método
[LB16a]	Mejora la gestión, la automatización y la operación de las subestaciones eléctricas	SDN - NFV	Modelo de red centralizado, categorizando funciones dentro de la subestación.
[WKR17]	Optimiza el uso de recursos en la red de la subestación virtualizada haciendo uso de IEDs virtuales	SDN - VM	Creación de red híbrida entre IEDs reales, IEDs virtuales y Switches virtuales.
[LDB20]	Algoritmo que mejora la redundancia en la red, garantizando su recuperación cuando falla un enlace	SDN	Presenta un modelo que permite encontrar dos árboles multicast para cada demanda.
[Rös+22]	Flexibiliza la red de la subestación con equipos reales y virtuales dentro de la misma red usando contenedores	SDN - NFV - LMV	Escenario híbrido de red con SDN y NFV implementadas en contenedores

4.2 Optimización de rutas usando métodos exactos y metaheurísticos

La optimización de rutas es un campo de estudio que se encarga de plantear métodos que permitan encontrar la mejor forma de recorrer un camino desde un punto origen a un destino, evaluando múltiples posibilidades y teniendo en cuenta ciertas condiciones o restricciones con el fin de maximizar o minimizar un objetivo.

En general el campo de la optimización de rutas tiene aplicaciones en una amplia gama de industrias y situaciones como: logística, transporte, distribución, planificación urbana, gestión de flotas, operaciones militares, rutas de vehículos autónomos

y más. Los objetivos de la optimización de rutas pueden variar y pueden incluir minimizar el tiempo de viaje, la distancia recorrida, el costo operativo, el consumo de combustible o maximizar la eficiencia en la asignación de los recursos.

Una metaheurística muy usada para resolver problemas de enrutamiento en los últimos años es ACO [DS19], algunas variaciones son planteadas para superar las deficiencias del algoritmo de hormigas clásico, como elegir repetidamente caminos de óptimo local, convergencia lenta y convergencia fácil a una solución óptima local. En [Che+18] los autores presentan una combinación entre ACO y el algoritmo de búsqueda tabú (TS) para optimizar el Problema del agente viajero (TSP). El artículo adopta una lista tabú para registrar los caminos recientemente explorados por las hormigas y usa esta información para formar un algoritmo de hormigas con memoria.

La comparación entre las mejores rutas encontradas entre el algoritmo tradicional y el propuesto en [Che+18], arrojan una mejora de un 4.85 %, y en la comparación de las peores rutas halladas se encuentra una diferencia a favor de la nueva propuesta en un 7.73 %. Bajo los mismos parámetros, el tiempo de búsqueda se acorta significativamente y se puede obtener una mejor solución. Los resultados experimentales demuestran la efectividad de la optimización de colonia de hormigas con la lista tabú.

En 2019 [JYG19] plantea una estrategia de optimización combinatoria neuronal basada en el aprendizaje profundo por refuerzo. Específicamente, se transforma el problema de enrutamiento en línea en un problema de generación de tours de vehículos, mediante la propuesta de una red de punteros con incrustación gráfica estructural para desarrollar recorridos de manera iterativa. Dado que la construcción de datos de entrenamiento supervisado para la red neuronal no es práctica no es práctico debido al alto costo computacional y sobre todo, al alto costo que traería el etiquetado de los datos, los autores proponen un mecanismo de aprendizaje profundo por refuerzo con una red auxiliar no supervisada para entrenar los parámetros del modelo.

Aunque la estrategia del modelo presentado por lo autores encuentra mejores rutas, su trabajo no contempla un posible fallo de enlace. Adicionalmente, la estrategia planteada en el artículo no funciona con información en tiempo real.

4.3 Algoritmos misceláneos

A continuación se presentará una revisión breve de trabajos que, si bien han contribuido en áreas diferentes a las que enmarcan este trabajo de investigación, constituyen una buena referencia conceptual para este informe.

Como se había mencionado, la obtención de la mejor ruta es un problema que aqueja a la industria en múltiples disciplinas. En [Che+19] se plantean los desafíos de la logística de cadena de frío entre los que se destacan ofrecer alimentos frescos minimizando las emisiones de CO₂. El grado de frescura y las emisiones de CO₂ están involucrados en el problema de optimización de enrutamiento de vehículos (VRP) en la logística de cadena de frío. Con el fin de cumplir con los requisitos de calidad para productos agrícolas frescos y la logística baja en carbono, en [Che+19] se estableció un modelo de optimización de enrutamiento que considera los costos de deterioro de calidad y emisiones de carbono, definido como problema de enrutamiento de vehículos bajos en carbono y grados de Frescura (LCFD-VRP).

Este modelo tiene en cuenta el costo fijo, el costo de combustible y el costo de penalización por ventana de tiempo. Se utiliza un algoritmo mejorado de colonia de hormigas (IACA) para optimizar todo el enrutamiento de distribución de vehículos debido a su sólida capacidad de búsqueda global. El algoritmo de TS se utiliza para buscar el enrutamiento de distribución de vehículos individuales debido a su buena capacidad de búsqueda local. Se propuso una combinación de IACA y TS (IACATS) para resolver el modelo LCFD-VRP mencionado anteriormente. La practicidad del modelo y la efectividad del algoritmo mejorado se verifican mediante un estudio de caso real.

Los resultados de la empresa Zhoushan Dayang Refrigerated Logistics Co. Ltd. mostraron que, en comparación con el algoritmo tradicional, IACATS pudo reducir la asignación de dos vehículos refrigerados, lo que redujo el costo total en un 4.94%, acortó la distancia de transporte real en un 5.50% y redujo las emisiones totales de CO₂ en un 8.9%. Por lo tanto, el modelo LCFD-VRP puede ayudar efectivamente a lograr bajas emisiones de carbono, distribución de múltiples variedades y bajo costo de productos agrícolas frescos. Los autores no contemplan una solución alternativa que contemple problemas en las vías, o distribución parcial de productos en caso de registrar carreteras con problemas.

Para mejorar el enrutamiento de vehículos [Zha+19] presenta un problema de enrutamiento multiobjetivo con ventanas de tiempo flexibles (MOVRPFlexTW). En este problema, una flota de vehículos puede atender a un conjunto de clientes antes y después del tiempo requerido con una tolerancia dada. Esta flexibilidad

permite a una empresa de logística ahorrar costos de distribución a expensas de la satisfacción del cliente. Los autores proponen una interpretación directa del problema de enrutamiento de vehículos con ventanas de tiempo flexibles (VRPFlexTW) como un problema multiobjetivo, donde se minimizan los costos totales de distribución (incluidos los costos de viaje y los costos fijos de vehículos) y se maximiza la satisfacción general del cliente.

En VRPFlexTW se permite que los clientes sean atendidos fuera de su intervalo de tiempo con una cierta tolerancia. Este problema también tiene en cuenta la satisfacción del cliente relacionada con la ruta, que se representa como una función convexa dependiente del tiempo de llegada del vehículo. En comparación con VRPTW, VRPFlexTW es más flexible y aplicable a problemas de la vida real. En este documento, se considera VRPFlexTW como un problema multiobjetivo, con el fin de minimizar los costos generales de enrutamiento y maximizar la satisfacción general del cliente.

El marco básico del enfoque propuesto para resolver MOVRPFlexTW se inspiró en la metaheurística ACO clásica. Sin embargo, este enfoque se extendió utilizando tres operadores de mutación que mejoran la capacidad de búsqueda local y permiten exploraciones globales aleatorias. Estos operadores de mutación evitan que el ACO quede atrapado en una solución óptima local. Mientras tanto, utilizando el concepto de optimización de Pareto, se obtienen mejores soluciones no dominadas para MOVRPFlexTW.

Los resultados implican que la propuesta es bastante eficiente en comparación con los mejores resultados publicados. Dado que el MOVRPFlexTW utiliza ventanas de tiempo más flexibles, se puede lograr más ahorro para la empresa logística a expensas de la satisfacción del cliente. Este equilibrio debe ser cuidadosamente considerado por los tomadores de decisiones con respecto a las preferencias de la empresa logística y las preocupaciones de sus clientes.

Las soluciones obtenidas utilizando MOVRPFlexTW fueron iguales o mejores que las soluciones obtenidas utilizando el VRPFlexTW de objetivo único. Aunque la distancia obtenida utilizando MOVRPFlexTW fue ligeramente peor que la distancia obtenida utilizando VRPFlexTW de objetivo único, la diferencia fue solo del 0.83 %.

En enrutamiento de tráfico en 2020 los autores de [Rus+20] proponen RouteNet. RouteNet es un modelo de red basado en Redes Neuronales Basadas en Gráficas (GNN) que es capaz de comprender la compleja relación entre la topología, el enrutamiento y el tráfico de entrada, para producir estimaciones precisas de la distribución de retraso por paquete por origen/destino. RouteNet aprovecha la capacidad de las GNN para aprender y modelar información con estructura de grafo

y, como resultado, el modelo es capaz de generalizar sobre topologías arbitrarias, esquemas de enrutamiento e intensidades de tráfico. RouteNet es capaz de predecir con precisión la distribución de retraso (retraso medio y fluctuación) y la pérdida incluso en topologías, enrutamientos y tráfico no visto en el entrenamiento. En particular, el problema de optimización se limita a: (i) generar un conjunto de configuraciones candidatas (por ejemplo, esquemas de enrutamiento), (ii) evaluar el rendimiento resultante para cada una de ellas y (iii) seleccionar la que mejor se adapte al objetivo de optimización. El rendimiento logrado por el optimizador basado en RouteNet fue comparado con los resultados obtenidos por optimizadores clásicos basados en la utilización de enlaces, la política de enrutamiento de camino más corto y la solución óptima. Los resultados muestran una reducción del retraso en promedio de un 44.511 % y una mejora en el jitter de 30.4 % en promedio.

En 2021, los autores de [SRK21] plantean una solución que tienen como objetivo minimizar el consumo total de energía y maximizar la vida útil de la red de sensores inalámbricos (WSNs). Los autores emplean el algoritmo de optimización de mariposa (BOA) para seleccionar un nodo líder de grupo, entre un grupo de nodos. La selección del nodo líder de grupo se optimiza mediante la energía residual de los nodos, la distancia a los vecinos, la distancia a la estación base, el grado del nodo y la centralidad del nodo. La ruta entre el nodo líder del grupo y la estación base se identifica mediante ACO, que selecciona la ruta óptima en función de la distancia, la energía residual y el grado del nodo. Las medidas de rendimiento de esta metodología propuesta se analizan en términos de nodos vivos, nodos muertos, consumo de energía y paquetes de datos recibidos por la estación base (BS), para luego evaluar los paquetes totales transmitidos a la BS en dos escenarios diferentes con nodos variables. El rendimiento de los paquetes transmitidos a la BS en la metodología propuesta es comparado con los algoritmos LEACH (jerarquía de agrupamiento adaptativo de baja energía) y DEEC (agrupamiento distribuido de eficiencia energética). Donde la metodología propuesta logra un mayor número de paquetes recibidos en los escenarios de prueba. La razón de lograr un mayor número de paquetes de datos en la BS es la eficiencia de las funciones de aptitud de BOA y ACO. Estas funciones de aptitud se consideran para preservar la energía residual de los nodos, lo que lleva a maximizar el número de nodos activos en toda la WSN, lo que a su vez aumenta el número de paquetes de datos transmitidos a la BS.

Para mejorar el enrutamiento en redes satelitales [Den+22] propone un algoritmo de enrutamiento de optimización de colonia de hormigas con reducción de ventana (ACORA-WR) para redes satelitales LEO (órbita terrestre baja), con el objetivo de lograr un balanceo de carga. ACORA-WR limita el movimiento de la colonia de hormigas a un rango específico y considera de manera integral la distancia de la ruta, la dirección de transmisión y la carga del enlace para encontrar una ruta con bajo retraso y menor sobrecarga. Cuando los paquetes se envían a través de enlaces

intersatelitales, se agrega información heurística para evitar la congestión a las reglas de transición de estado, y las hormigas detectan la ruta a través de las reglas de transición de estado mejoradas de manera periódica para ajustar la información local a tiempo. Además, se propone un mecanismo de reducción de ventana para limitar la dirección de transmisión de datos y eliminar rutas redundantes, lo que acelera la velocidad de convergencia de la colonia de hormigas y reduce la sobrecarga de transmisión de la red.

El algoritmo ACORA-WR fue comparado con tres protocolos de enrutamiento populares: LBRA-CP [Wan+19], SPR y LCRA [Liu+15] bajo las métricas de tasa de entrega de datos, retardo promedio, proporción de sobrecarga de transmisión y tasa de transferencia. Los resultados experimentales demuestran que el esquema ACORA-WR propuesto, realiza el balanceo en la entrega de datos y presenta un rendimiento de red más altos en cada una de las métricas.

En [SOH22] los autores proponen un algoritmo que mejora los resultados obtenidos al usar ACO para optimizar rutas. La mejora del ACO denominada algoritmo AACO-NC, implementa tres técnicas para mejorar el rendimiento general en las rutas y reducir el tiempo de ejecución con métodos de diversificación y la configuración de parámetros de control. Estas técnicas incluyen: (i) el concepto de agrupación de nodos, donde los nodos de transición se organizan en un conjunto de clústeres, (ii) la evaporación adaptativa de feromonas controlada dinámicamente basada en la entropía de la información y (iii) la formulación de una nueva condición de terminación basada en la diversidad de soluciones en la población. El algoritmo compara el rendimiento y el tiempo de optimización del AACO-NC en las instancias pequeñas y medianas seleccionadas para diferentes configuraciones de los parámetros de agrupación de control ($n_{cluster}$, $n_{tamaño}$, $n_{sectores}$), así como con el enfoque original (sin agrupación de nodos).

Los experimentos se llevaron a cabo utilizando 30 instancias de referencia (que van desde 51 hasta 2,392 nodos con diversas topologías) tomadas de los conocidos bancos de pruebas de TSPLIB, y los resultados se comparan con varios métodos de última generación basados en ACO; en la mayoría de los casos, el algoritmo propuesto supera a estos métodos rivales. El impacto de las nuevas técnicas en el comportamiento del algoritmo se discute en relación con el rendimiento general, el tiempo de ejecución y la convergencia.

Tabla. 4.2: Propuestas de optimización de rutas.

Artículo	Contribución	Algoritmo usado	Tipo	Métrica
[Che+18]	Mejora la convergencia del algoritmo ACO para cualquier caso de uso	ACO - TS	Metaheurística	Distancia - tiempo de convergencia
[JYG19]	Reduce las rutas con búsqueda de parámetros off-line para entrenar modelo.	Redes neuronales	Metaheurística	Tiempo ruta
[Che+19]	Reduce la distancia y minimiza el CAPEX	ACO modificado - TS	Metaheurística	Distancia de rutas
[Zha+19]	Minimiza OPEX y CAPEX	ACO	Metaheurística	Distancia de rutas
[Rus+20]	Reduce el retraso y el jitter	Redes neuronales	Metaheurística	Retraso - Fluctuación
[SRK21]	Reduce el CAPEX	ACO modificado	Metaheurística	Tasa de entrega de datos - Retraso promedio
[SOH22]	Reduce las rutas	ACO modificado	Metaheurística	Tiempo de ruta
[Den+22]	Balanza la carga y reduce el encolamiento	ACO modificado	Metaheurística	Jitter - Ocupación de enlaces

Después de llevar a cabo una exhaustiva revisión de la literatura sobre la virtualización en subestaciones eléctricas, se ha llegado a la conclusión de que hasta la fecha no se han encontrado modelos que aborden de manera efectiva el desafío de gestionar el enrutamiento en situaciones de fallos de enlace. A pesar de los avances significativos en la virtualización de subestaciones y la optimización de la infraestructura eléctrica, la gestión de la conectividad y el flujo de datos en condiciones de fallo sigue siendo un área poco explorada en la investigación. Este hallazgo resalta la necesidad de futuras investigaciones y desarrollo de modelos que aborden específicamente esta problemática crítica en este contexto.

Implementación de algoritmos de solución del problema de rutas alternativas usando ILP y ACO

En esta sección se presentarán los métodos matemáticos y computacionales que han sido implementados para el desarrollo del trabajo descrito. El proceso inicial implica la virtualización de servicios para obtener los parámetros del modelo. En este contexto, se introduce una formulación ILP diseñada para abordar el desafío de asignar rutas en situaciones donde se produzcan fallos de enlace en la red de comunicaciones. Este proceso de asignación de rutas se lleva a cabo en paralelo con la tarea de equilibrar la carga en los enlaces. Además, se implementa una metaheurística como parte integral de la solución.

A continuación, se presenta la topología Spiderweb, seleccionada para la implementación de las soluciones propuestas en este trabajo de investigación. Esta topología consta de una estructura en forma de estrella y un anillo externo. Las ventajas específicas de esta configuración sobre otras han sido definidas en la literatura por los autores en [LB19].

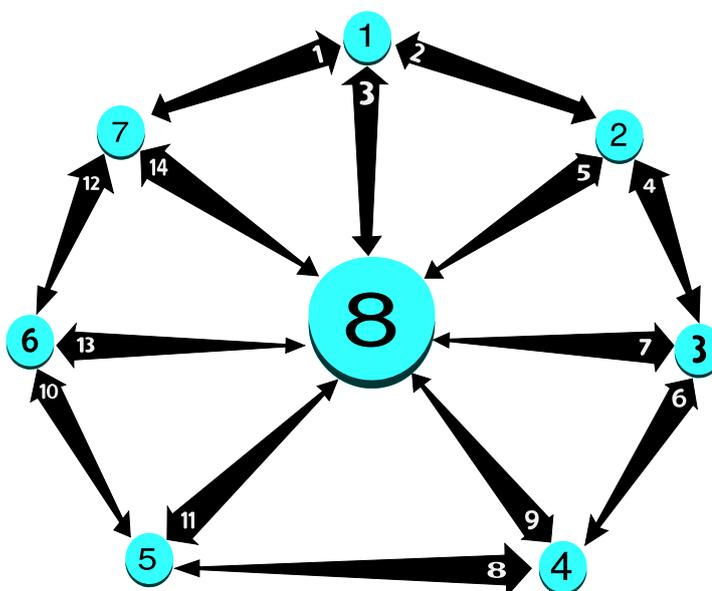


Figura. 5.1: Red Spiderweb general

5.1 Virtualización de servicios en redes eléctricas virtualizadas

Los desarrolladores de sistemas eléctricos ofrecen bibliotecas de funciones y kits de desarrollo de software (SDK) que contienen herramientas predefinidas, destinadas a facilitar la implementación del estándar IEC 61850 en dispositivos y sistemas. Estas bibliotecas proporcionan múltiples ejemplos basados en el modelo *Publisher-Subscriber*, implementados en el lenguaje de programación C [Zil13]. Al utilizar estas bibliotecas, los desarrolladores pueden crear sistemas que cumplen con el estándar en entornos de software, simplificando el desarrollo de aplicaciones que permiten a los investigadores comprender el uso de este estándar y realizar pruebas en entornos emulados antes de la implementación en entornos reales.

Un servicio virtualizado en el contexto de subestaciones eléctricas (GOOSE, MMS y SV), es un conjunto de funciones y capacidades proporcionadas por dispositivos virtuales en una red de subestación eléctrica simulada. Este servicio implica la comunicación eficiente y precisa de eventos, mensajes y valores de medida entre estos dispositivos, utilizando los protocolos GOOSE para eventos, MMS para mensajes y SV para la transmisión de valores de medida. La implementación y configuración de este servicio se llevan a cabo en un entorno virtualizado, lo que permite simular interacciones realistas entre los elementos de la subestación eléctrica para propósitos de prueba, desarrollo y análisis.

Las pruebas fueron realizadas en un equipo portátil marca ASUS, sistema operativo Ubuntu 22.04.3 LTS, Procesador Intel(R) Core(TM) i7-5500, CPU @ 2.4GHz dos núcleos de 4 hilos, memoria RAM 8GB, Motherboard 1.0/X555LB(ASUSTeC COMPUTER INC), tarjeta gráfica nvidia GEFORCE 940M.

En la Tabla 5.1, se detallan las diversas parejas *Publisher-Subscriber* que simulan distintos servicios para la comunicación entre equipos dentro de la subestación. En el proceso de implementación, se priorizaron pruebas con parejas específicas, como *Client example3 – Server example control* (MMS), *goose subscriber – goose publisher* (GOOSE), y *sv subscriber – sv publisher* (SV). Estas pruebas tienen como objetivo medir el tráfico enviado, probar la eficiencia de las funciones dentro de los contenedores, cambiar los identificadores para un modelo *Publisher* múltiples *Subscribers*, determinar la demanda usando Wireshark y calcular el tiempo máximo requerido para cada servicio. Estos parámetros serán usados tanto en las restricciones del modelo ILP planteado, como en la metaheurística propuesta en este capítulo.

Tabla. 5.1: Parejas *Publisher-Subscriber*.

MMS	
<i>Subscriber</i>	<i>Publisher</i>
Client-example3	Server-example-control
Client-example4	Server-example3
Client-example4	Server-example-goose
Client-example-reporting	Server-example3
Client-example-reporting	Server-example-goose
Client-example-log	Server-example-loggin
GOOSE	
<i>Subscriber</i>	<i>Publisher</i>
goose-observer	server-example-goose
goose-subscriber	goose-publisher
SV	
<i>Subscriber</i>	<i>Publisher</i>
sv-subscriber	sv-publisher

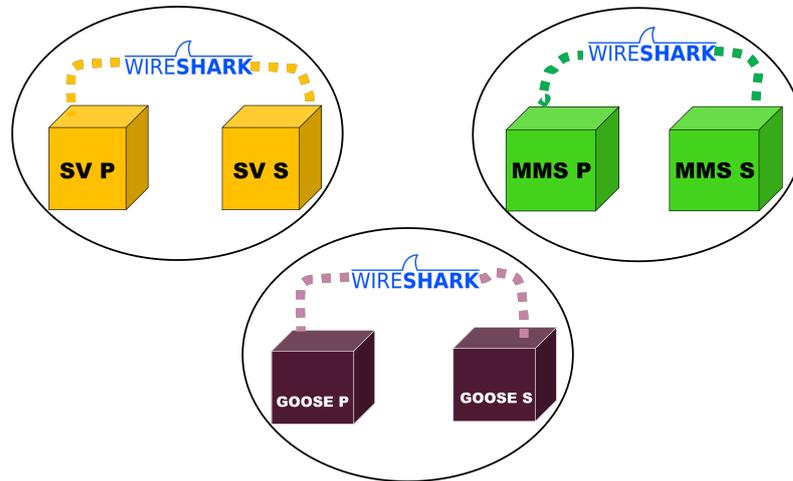


Figura. 5.2: Parejas de prueba por cada servicio

La implementación de servicios basados en contenedores requiere el uso de plataformas de software que faciliten la creación, gestión y ejecución de aplicaciones de manera coherente y eficiente. Entre las diversas opciones disponibles, se optó por utilizar Docker como plataforma para virtualizar los servicios, debido a su amplia documentación y su fácil integración en el sistema.

Para el caso de uso de una subestación eléctrica virtualizada se consideran dos criterios importantes para comenzar la creación del servicio: (i) el tamaño de la imagen base y (ii) la facilidad para encontrar librerías compatibles con la imagen base que facilite el funcionamiento de cada servicio. La tabla 5.2 muestra la categorización de las imágenes evaluadas.

Tabla. 5.2: Imágenes base.

Imagen	Tamaño	Características diferenciales
BusyBox	2.71MB	Es la imagen base más liviana de los servicios virtualizados, contiene solo un archivo binario en su interior que permite agregar software mediante código para su posterior compilación. Su desventaja radica en que no dispone de un gestor de paquetes.
Alpine	5MB	Esta imagen se basa en Busybox y se complementa con musl libc. Se utiliza comúnmente para despliegues rápidos y viene con el gestor de paquetes APK, lo que facilita la instalación de software.
Ubuntu 22.04	77.8MB	Ubuntu es una imagen que proporciona una amplia y variada colección de paquetes, herramientas y documentación. Destaca por su sólido soporte y actualizaciones periódicas, además de contar con una interfaz de usuario amigable.
CentOS	172MB	CentOS es una imagen que se deriva de Red Hat y se caracteriza por tener un mayor número de archivos de configuración para la creación y el funcionamiento del sistema operativo. Cada imagen de CentOS cuenta con un soporte continuado durante un período de 10 años.

5.1.1 Creación de imagen ‘manager’

De acuerdo con lo observado en la tabla 5.2, se crea una imagen llamada ‘manager’ basada en Alpine. Posteriormente, se complementa en varios pasos: i) se instalan los encabezados de Linux, necesarios para compilar el software sin problemas; ii) se instala un editor de texto para modificar los archivos de la librería IEC61850; iii) se agrega la librería SQLite para gestionar los datos y registros obtenidos durante la ejecución de cada contenedor; iv) se incluye el software make para automatizar la compilación de los códigos fuente; v) se instala el compilador de C++ y se descarga la librería IEC61850-1.5.1. Sin embargo, es importante tener en cuenta que al actualizar los paquetes y llevar a cabo la compilación de la librería, el tamaño del contenedor aumenta hasta alcanzar los 324MB. Esto lo excluye como una opción viable para funcionar como un servicio, ya que su proceso de inicio requiere más tiempo y consume una parte significativa de los recursos de la máquina anfitriona. No obstante, sí sirve como servidor para los ejecutables necesarios para crear los diferentes servicios. La clave para optimizar el contenedor radica en que solo albergue los ejecutables esenciales para la implementación de cada servicio.

La Figura 5.3 muestra la estructura de la imagen base ‘manager’ y los archivos utilizados para el funcionamiento de la librería en el entorno virtualizado.

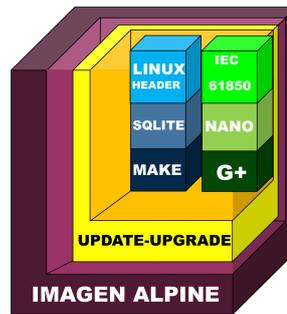


Figura. 5.3: Imagen 'manager' 324MB.

5.1.2 Creación de servicios

En el proceso de creación de los servicios, se sigue un procedimiento similar al realizado en el paso anterior. Sin embargo, en esta etapa, el enfoque principal es mantener los contenedores lo más ligeros posible. En otras palabras, se carga un ejecutable específico en cada contenedor desde el contenedor manager para definir su función concreta dentro de la subestación eléctrica. En el contexto de las pruebas, al incorporar únicamente el ejecutable de cada servicio a la imagen base, se logran imágenes funcionales con tamaños reducidos: 7.22MB para servicios GOOSE, 7.23MB para servicios SV y 9.44MB para servicios MMS.

En la Figura 5.4 se presenta un servicio GOOSE en funcionamiento mientras se le analiza el tráfico y la cantidad de Bytes por ejecución.

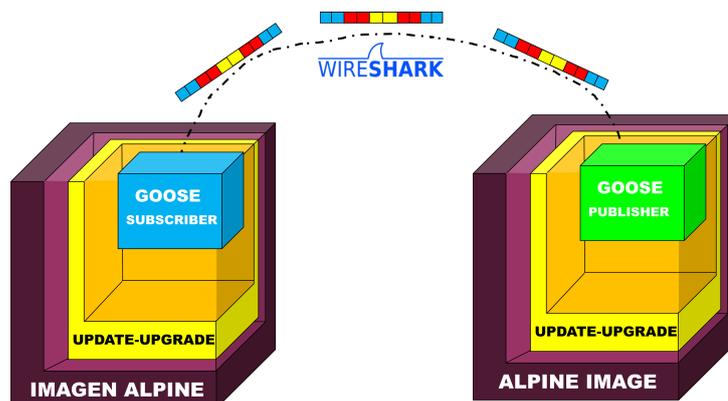


Figura. 5.4: Servicio GOOSE 7.22MB.

Para la adquisición de parámetros, se implementaron dos redes virtuales de prueba utilizando un switch virtual, varios nodos virtuales y los servicios respectivamente virtualizados. En la Figura 5.5, se presenta una red A con transmisión de información punto a punto (1 publisher - 1 subscriber) y una red B con transmisión de información punto a multipunto (1 publisher - multiples subscribers).

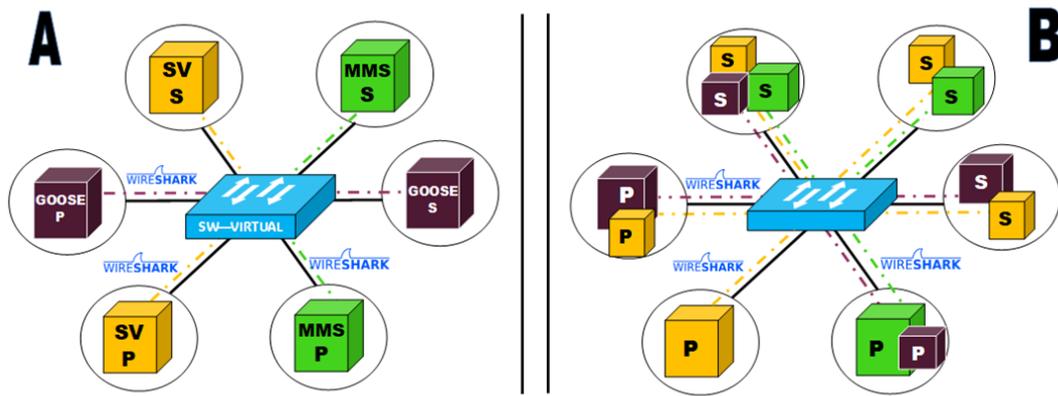


Figura. 5.5: Red A con tres servicios diferentes Vs red B con múltiples publicadores P y múltiples suscriptores S para cada servicio sin interferencia.

La implementación de la red A no presentó grandes desafíos, ya que el modelo tenía una configuración predeterminada para las parejas presentadas en la Tabla 5.1. Sin embargo, en la red B se realizaron modificaciones en la librería para adaptarse a cada servicio. Esto permitió presentar múltiples servicios virtualizados del mismo tipo sin generar problemas en la comunicación entre el publicador y el suscriptor. Esto es, los servicios pueden ser del mismo tipo pero etiquetados de forma diferente, por lo tanto es posible generar comunicaciones de un publicador a múltiples suscriptores dentro de una red virtualizada o se puede establecer una comunicación publicador suscriptor dentro del mismo equipo. Un resumen de los cambios realizados en la librería para garantizar el funcionamiento de los servicios es presentado en la Tabla 5.3.

Tabla. 5.3: Parámetros de comunicación para cada servicio.

MMS	
Parámetro	Descripción
simpleIOGenericIO	Este parámetro representa un identificador que hace referencia a un dispositivo lógico (equipo que tiene varias funciones en su interior "IEDs") donde se encuentra un conjunto de datos.
LLN0.AnalogueValues	Hace referencia al nodo lógico (función dentro de un dispositivo lógico que representa una tarea o un conjunto de datos).
GOOSE	
Parámetro	Descripción
appId	Identificador del tipo de mensaje GOOSE que se envía. De acuerdo con el valor de la appId, los dispositivos receptores interpretan el mensaje o no lo reconocen si no es compatible con su appId programada.
dstAddress	Dirección de destino del mensaje GOOSE, es decir, la dirección MAC del destinatario.
vlanId	Identifica la VLAN a la que pertenece el tráfico GOOSE.
vlanPriority	Especifica la prioridad de tráfico dentro de la VLAN.
SV	
Parámetro	Descripción
AppId	Similar al servicio GOOSE, el appId en los SV debe ser igual entre el <i>Publisher</i> y el <i>Subscriber</i> para que pueda establecerse la comunicación. La configuración de este parámetro debe estar respaldada por la misma interfaz de red entre los dispositivos a conectar (eth0, wlan0, docker0, etc.).

Una vez desplegados los servicios se revisan las métricas expuestas en cada escenario y el comportamiento del mismo cuando incrementa la cantidad de servicios en la red de la topología presentada en la figura 5.5 A.

La Tabla 5.4 presenta los resultados obtenidos para cada servicio ejecutado. En la primera fila, se especifica el retraso máximo permitido por demanda d de cada servicio s . Las filas 2-3 muestran el tráfico generado por la demanda d del servicio s en un enlace e , mientras que la fila 4 indica el tiempo necesario para inicializar un contenedor antes de establecer un servicio. Es importante destacar que los datos presentados en la tabla representan los límites de las restricciones del modelo propuesto.

Tabla. 5.4: Parámetros obtenidos para cada servicio.

Métricas	Goose	SV	MMS
Retraso máximo permitido (T_{ds} en segundos)	3e-3	10e-3	250e-3
Cantidad de demanda (h_{ds} en bits)	1640	920	712
Cantidad de demanda (h_{ds} en bytes)	205	115	89
Despliegue contenedor local (segundos)	590e-3	570e-3	518e-3

5.2 Modelo ILP

El enfoque para modelar la red implica considerarla como un grafo dirigido $G = (N, E)$, donde N es el conjunto de nodos y E es el conjunto de enlaces de la red. Se formula un modelo de optimización lineal entera que busca determinar las rutas óptimas en caso de una falla en alguno de los enlaces de la red de datos de una subestación eléctrica. Para facilitar la comprensión del modelo, se proporciona un resumen de las variables y parámetros relevantes en la Tabla 5.5.

Tabla. 5.5: Índices, variables y parámetros del modelo propuesto.

Índices	
ID	Descripción
n	Representa los nodos de la red $n = 1, 2, \dots, N$
e	Representa los enlaces de la red $e = 1, 2, \dots, E$
d	Representa las demandas a llevar entre nodos $d = 1, 2, \dots, D$
s	Representa los servicios de subestación $s = 1, 2, \dots, S$
s_d	Nodo origen para la demanda d
t_d	Nodo destino para la demanda d
Variables	
δ_{eds}	Variable binaria que indica si el enlace e es usado para llevar la demanda d del servicio s
K_e	Variable que almacena las demandas enviadas por el enlace e
R	Variable a optimizar
Parámetros	
a_{en}	Parámetro que indica si el nodo n es el origen del enlace e
b_{en}	Parámetro que indica si el nodo n es el destino del enlace e
T_e	Retraso del enlace e
T_{ds}	Retraso máximo permitido para la demanda d del servicio s (ver Tabla 5.4)
h_{ds}	Cantidad de demanda d para el servicio s (ver Tabla 5.4)
C_e	Capacidad del enlace e

Este modelo de optimización incluye una función objetivo destinada a minimizar la carga en el enlace más utilizado de la red, al mismo tiempo que asegura que la asignación de nuevas rutas cumple con las restricciones de tiempo y demanda para cada servicio.

$$\text{mín}R \tag{5.1}$$

Dado que K_e es un vector acumulador que registra las demandas que atraviesan cada enlace y R representa la variable que adquiere el valor del enlace más utilizado (se asigna el valor a R como el máximo valor de k_e).

$$K_e \leq R \quad \forall e \tag{5.2}$$

Las restricciones del problema limitan el espacio de búsqueda, y son representadas de la siguiente manera.

$$\begin{aligned}
K_e &= \sum_{s=1}^S \sum_{d=1}^D \delta_{eds} h_{ds} \quad \forall e \\
\sum_{e=1}^E a_{en} \cdot \delta_{eds} - \sum_{e=1}^E b_{en} \cdot \delta_{eds} &= \begin{cases} 1, & \text{Si } n = s_d \\ 0, & \text{Si } n \neq (s_d \text{ y } t_d) \\ -1, & \text{Si } n = t_d. \end{cases} \quad \forall d, n, s \\
\text{Sujeto a } \sum_{e=1}^E T_e \cdot \delta_{eds} &\leq T_{ds} \quad \forall d, s \\
\sum_{d=1}^D \delta_{eds} &\leq 1 \quad \forall e, s \\
\sum_{s=1}^S \sum_{d=1}^D h_{ds} &\leq C_e \quad \forall e
\end{aligned} \tag{5.3}$$

A continuación se definirán y explicarán cada unas de las restricciones que se han añadido al problema del ejemplo ILP.

Demandas que pasan por cada enlace

$$K_e = \sum_{s=1}^S \sum_{d=1}^D \delta_{eds} h_{ds} \quad \forall e \tag{5.4}$$

Donde K_e es la variable que acumula las demandas, también conocidas como el flujo de datos generado al enviar un mensaje o un servicio por cada enlace e . δ_{eds} es una variable binaria definida como:

$$\delta_{eds} = \begin{cases} 1, & \text{si el enlace } e \text{ es usado para llevar la demanda } d \text{ del servicio } s \\ 0, & \text{en otros casos} \end{cases}$$

h_{ds} es un parámetro que indica la cantidad de demanda d para el servicio s .

Restricción de conservación de flujo

$$\sum_{e=1}^E a_{en} \cdot \delta_{eds} - \sum_{e=1}^E b_{en} \cdot \delta_{eds} = \begin{cases} 1, & \text{Si } n = s_d \\ 0, & \text{Si } n \neq (s_d \text{ y } t_d) \\ -1, & \text{Si } n = t_d. \end{cases} \quad \forall d, n, s \tag{5.5}$$

Esta restricción garantiza la conservación de flujo, es decir, el flujo que entra al nodo n es igual al flujo que sale del mismo nodo. Se mantiene la indexación para determinar cual es el nodo origen y el nodo destino de cada demanda. a_{en} es un parámetro usado para definir si el nodo es un nodo origen, tal que: $a_{en} = 1$ si n es el origen del enlace e y $a_{en} = 0$ en otros casos. b_{en} es un parámetro usado para definir si el nodo n es un nodo destino, tal que: $b_{en} = 1$ si n es el destino del enlace e y $b_{en} = 0$ en otros casos.

s_d Es el nodo origen para la demanda d , y t_d es el nodo destino para una demanda d . Los valores de s_d y t_d son asignados de acuerdo con las siguientes condiciones:

$$s_d = \begin{cases} 1 & \text{Si es el nodo que origina la demanda} \\ 0 & \text{en otros casos} \end{cases}$$

$$t_d = \begin{cases} -1 & \text{Si es el nodo destino de la demanda} \\ 0 & \text{en otros casos} \end{cases}$$

Restricción de retardo máximo

$$\sum_{e=1}^E T_e \cdot \delta_{eds} \leq T_{ds} \quad \forall d, s \quad (5.6)$$

Esta restricción garantiza que la suma de todos los retrasos en los enlaces de una ruta dada entre s_d y t_d esté por debajo, o a lo sumo igual al retardo máximo T_{ds} definido para la demanda y el servicio correspondiente.

Restricción de camino para cada demanda de un servicio

$$\sum_{d=1}^D \delta_{eds} \leq 1 \quad \forall e, s \quad (5.7)$$

Esta restricción garantiza que una demanda dada no utilice el mismo enlace dos veces. Esta condición también ayuda a asegurar que no haya una sobrecarga en un enlace, ya que distribuye las demandas entre diferentes rutas.

Restricción de capacidad de enlace

$$\sum_{s=1}^S \sum_{d=1}^D h_{ds} \leq C_e \quad \forall e \quad (5.8)$$

Esta restricción asegura que la suma de todas las demandas en un enlace dado no supere su capacidad máxima C_e .

5.3 Algoritmo Metaheurístico ACO

Dado que los métodos exactos pueden ser computacionalmente intensivos y a menudo ineficientes cuando se enfrentan a problemas muy complejos (muchas variables y restricciones). En problemas con un espacio de búsqueda extremadamente grandes, pueden requerir mucho tiempo y demandar demasiados recursos para explorar todas las posibles soluciones. Por lo tanto, se propone un algoritmo para resolver el problema en instancias de tamaño mediano a grande. Como una eficiente combinación entre la intensificación y diversificación, las metaheurísticas se convierten en una de las estrategias de solución más efectivas para resolver problemas de optimización combinatoria y aplicables a una amplia gama de problemas académicos y del mundo real [GK06].

ACO es un método que se fundamenta en una colonia inicial de hormigas artificiales que exploran caminos de manera aleatoria desde un punto de origen hasta un destino, depositando feromonas a lo largo de su trayecto [DS19]. La convergencia del algoritmo está determinada por la cantidad de feromonas presentes en cada camino después de cada iteración, considerando el camino con mayor acumulación de feromonas como la ruta más corta entre los nodos de origen y destino. Inicialmente para la implementación del algoritmo, todos los enlaces poseen la misma cantidad de feromonas; sin embargo, dado que los pesos de los enlaces son diferentes, existe una alta probabilidad de que la colonia seleccione una buena ruta en su búsqueda hacia el nodo destino.

A continuación, se presenta el resumen de los parámetros iniciales del algoritmo para la primera implementación con la red de 8 nodos.

Dado que el algoritmo presenta dificultades de convergencia en instancias extensas, especialmente cuando la red supera los 20 nodos, se hace necesario ajustar las condiciones iniciales en función de las soluciones obtenidas en cada iteración. Por lo tanto, se incrementa el tamaño de la colonia de manera proporcional a la tasa de 2, es decir, si el algoritmo no logra converger en 5 iteraciones, la colonia se expande a 20; después de 10 iteraciones, se amplía a 40, y así sucesivamente. Dado que la

Tabla. 5.6: Parámetros iniciales del algoritmo ACO.

Parámetro	Valor inicial	Descripción
K	10	Número de hormigas en la colonia.
d_{ij}	$1e-05 < d_{ij} < 2e-03$	Distancia (o peso) en el enlace que une el nodo i con el nodo j .
τ_{ij}	0.01	Rastro de feromona inicial del nodo i al nodo j .
α	0.7	Parámetro que controla la influencia relativa del rastro de feromona ($0 < \alpha < 1$).
β	1.2	Parámetro que controla la influencia de la información heurística ($0 < \beta$).
ρ	0.01	Coefficiente de evaporación de feromonas.

información heurística permanece constante (es el estado de la red), se incrementa el parámetro α en 0,1, se reduce β en 0,1 y se reduce el coeficiente de evaporación ρ , facilitando así que las hormigas sigan más fácilmente el rastro de feromonas. Las condiciones iniciales cuando se presentan problemas de convergencia se muestran en la tabla 5.7.

Tabla. 5.7: Parámetros secundarios del algoritmo ACO.

Parámetro	Valor inicial	Descripción
K	20	Número de hormigas en la colonia.
d_{ij}	$1e-05 < d_{ij} < 2e-03$	Distancia (o peso) en el enlace que une el nodo i con el nodo j .
τ_{ij}	0.02	Rastro de feromona inicial del nodo i al nodo j .
α	0.8	Parámetro que controla la influencia relativa del rastro de feromona ($0 < \alpha < 1$).
β	1.1	Parámetro que controla la influencia de la información heurística ($0 < \beta$).
ρ	0.008	Coefficiente de evaporación de feromonas.

Experimentos y resultados

En este capítulo se realizaron simulaciones para evaluar de manera apropiada el rendimiento de las soluciones propuestas, incluyendo el modelo ILP y el algoritmo metaheurístico ACO presentado en el capítulo anterior. En esta sección, se describen en detalle los escenarios de simulación y se presentan los resultados experimentales.

Se implementaron dos experimentos principales orientados a evaluar la eficiencia de los métodos propuestos para encontrar la ruta óptima en caso falla de alguno de los enlaces de una red de datos de una subestación eléctrica con servicios virtualizados. Los algoritmos mencionados fueron evaluados en una topología de red de tipo “Spiderweb” dadas las ventajas expuestas por los autores en comparación con otras topologías [LB19], con el objetivo de determinar su idoneidad para restaurar el servicio en un tiempo asequible y encontrar rutas alternas manteniendo los enlaces balanceados con la menor cantidad de demandas posible, es decir, la ruta óptima.

Dado que las redes de datos en subestaciones eléctricas suelen implementarse con enlaces de fibra óptica y, considerando que los retardos en estos enlaces tienden a ser mínimos, se plantea la utilización de un vector de tiempo. Este vector inicia con valores cercanos a cero hasta valores que se aproximan al máximo permitido por el servicio de alta prioridad, especialmente en el caso del protocolo GOOSE, que establece un límite de tiempo de camino de 3 milisegundos entre origen y destino de la demanda. Los tiempos fueron establecidos de forma aleatoria entre los límites mencionados como se pueden evidenciar en la Tabla 6.1.

Tabla. 6.1: Vector de retardos para un enlace

Enlaces	Iteraciones								
	E	ITE1	ITE2	ITE3	ITE4	ITE5	ITE6	ITE7	ITE8
1	1e-05	5e-05	0.0001	0.0003	0.001	0.0015	0.002	0.0025	

Con el propósito de obtener datos con significancia estadística, se diseñaron redes de diferentes tamaños. Se estableció un vector de 8 tiempos para cada enlace, considerando que la metaheurística ACO incorpora un componente aleatorio que, en el transcurso del experimento, produce respuestas variadas, y teniendo en cuenta que para garantizar resultados fiables se requiere un número sustancial de muestras,

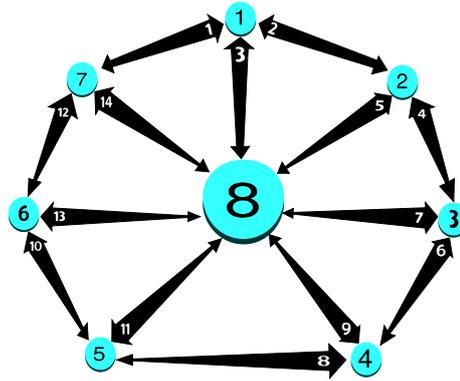


Figura. 6.1: Red Spiderweb 8 Nodos

se observó que valores inferiores a 100 podrían generar respuestas sesgadas. De acuerdo con lo anterior, se concluye que la red debe contener mínimo de 8 nodos (14 enlaces) para garantizar que el numero mínimo de muestras sobrepase los 100 valores $n = (N_{enlaces} + 1) * Tiempos$.

Inicialmente se simuló una red con 8 nodos y 14 enlaces (ver Figura 6.4(a), donde se evaluaron dos escenarios diferentes: (i) la red funcionando completamente con los mismos retardos para cada enlace pero distribuidos de forma aleatoria en cada vector de tiempos, y (ii) la red funcionando sin uno de sus enlaces (seleccionado al azar), lo cual se definió como “enlace roto” con el propósito de evaluar la efectividad de los algoritmos de optimización para encontrar rutas alternativas.

La tabla 6.2 presenta los tiempos de la red inicial de 8 nodos, la cual se muestra en la figura 6.4(a).

Tabla. 6.2: Matriz de retardos para la red de 8 nodos y 14 enlaces

Enlaces		Iteraciones						
<i>E</i>	ITE1	ITE2	ITE3	ITE4	ITE5	ITE6	ITE7	ITE8
1	0.0015	0.002	0.001	0.0025	1e-05	5e-05	0.0003	0.0001
2	5e-05	0.0025	0.0001	0.002	0.0003	0.0015	1e-05	0.001
3	0.0001	0.002	0.0003	1e-05	5e-05	0.0025	0.001	0.0015
4	0.001	5e-05	1e-05	0.0025	0.0001	0.0015	0.002	0.0003
5	0.0003	0.0025	0.0001	1e-05	0.002	0.0015	0.001	5e-05
6	0.002	0.0001	0.0025	1e-05	5e-05	0.0003	0.001	0.0015
7	1e-05	0.0003	0.0001	0.0025	0.002	5e-05	0.001	0.0015
8	1e-05	0.0001	0.002	0.001	0.0003	0.0025	0.0015	5e-05
9	0.0025	0.001	1e-05	0.0003	0.0015	0.0001	0.002	5e-05
10	0.002	1e-05	0.0003	0.001	0.0015	0.0001	5e-05	0.0025
11	1e-05	0.0003	0.001	0.002	0.0025	0.0001	0.0015	5e-05
12	1e-05	0.001	0.0025	0.0001	0.0003	5e-05	0.002	0.0015
13	0.0025	0.0003	0.0015	0.002	1e-05	0.0001	0.001	5e-05
14	0.002	1e-05	0.0003	0.0025	0.001	0.0015	0.0001	5e-05

Al igual que en el experimento anterior, se asignan tiempos a los enlaces de la red de 20 nodos, siguiendo los valores especificados en la Tabla 6.1. Para facilitar la implementación, se mantiene el orden de los enlaces, siendo los pares dispuestos en

sentido horario en el anillo y los impares en la disposición de la estrella (ver Figura 6.4(b))

Tabla. 6.3: Matriz de retardos para la red de 20 nodos y 38 enlaces

Enlaces	Iteraciones							
	E	ITE1	ITE2	ITE3	ITE4	ITE	ITE6	ITE7
1	0.0015	0.002	0.001	0.0025	1e-05	5e-05	0.0003	0.0001
2	5e-05	0.0025	0.0001	0.002	0.0003	0.0015	1e-05	0.001
3	0.0001	0.002	0.0003	1e-05	5e-05	0.0025	0.001	0.0015
4	0.001	5e-05	1e-05	0.0025	0.0001	0.0015	0.002	0.0003
5	0.0003	0.0025	0.0001	1e-05	0.002	0.0015	0.001	5e-05
6	0.002	0.0001	0.0025	1e-05	5e-05	0.0003	0.001	0.0015
7	1e-05	0.0003	0.0001	0.0025	0.002	5e-05	0.001	0.0015
8	1e-05	0.0001	0.002	0.001	0.0003	0.0025	0.0015	5e-05
9	0.0025	0.001	1e-05	0.0003	0.0015	0.0001	0.002	5e-05
10	0.002	1e-05	0.0003	0.001	0.0015	0.0001	5e-05	0.0025
11	1e-05	0.0003	0.001	0.002	0.0025	0.0001	0.0015	5e-05
12	1e-05	0.001	0.0025	0.0001	0.0003	5e-05	0.002	0.0015
13	0.0025	0.0003	0.0015	0.002	1e-05	0.0001	0.001	5e-05
14	0.002	1e-05	0.0003	0.0025	0.001	0.0015	0.0001	5e-05
15	0.001	0.0015	0.0001	0.002	5e-05	0.0003	1e-05	0.0025
16	0.0025	5e-05	1e-05	0.002	0.0003	0.001	0.0001	0.0015
17	0.0015	5e-05	1e-05	0.001	0.002	0.0003	0.0025	0.0001
18	0.0003	1e-05	0.002	0.0001	0.0015	0.0025	0.001	5e-05
19	0.0001	0.0015	0.002	0.001	0.0003	5e-05	1e-05	0.0025
20	0.002	0.001	0.0001	1e-05	0.0015	0.0003	0.0025	5e-05
21	0.0001	5e-05	0.0003	0.0025	0.0015	1e-05	0.001	0.002
22	0.0003	0.0015	1e-05	0.001	0.0025	5e-05	0.002	0.0001
23	0.001	1e-05	0.0015	0.0003	0.002	0.0025	0.0001	5e-05
24	0.002	0.0001	1e-05	0.0003	5e-05	0.001	0.0015	0.0025
25	1e-05	5e-05	0.0015	0.0003	0.001	0.002	0.0001	0.0025
26	0.0015	0.001	0.002	1e-05	0.0001	0.0025	0.0003	5e-05
27	0.0025	0.001	1e-05	0.002	5e-05	0.0003	0.0001	0.0015
28	0.0001	0.0025	0.001	5e-05	0.0003	1e-05	0.002	0.0015
29	0.0015	0.0025	0.0001	5e-05	0.002	0.0003	1e-05	0.001
30	0.0001	0.0015	0.0025	1e-05	0.001	0.0003	5e-05	0.002
31	0.0025	1e-05	0.0015	0.0001	5e-05	0.0003	0.001	0.002
32	5e-05	0.0015	0.0001	1e-05	0.0025	0.001	0.002	0.0003
33	0.002	0.001	1e-05	0.0015	0.0001	5e-05	0.0003	0.0025
34	0.0003	0.0015	0.001	0.002	5e-05	1e-05	0.0025	0.0001
35	0.0025	5e-05	0.0001	0.0003	1e-05	0.001	0.002	0.0015
36	0.0003	0.0001	0.0025	5e-05	0.002	1e-05	0.001	0.0015
37	1e-05	5e-05	0.0015	0.0003	0.0025	0.002	0.001	0.0001
38	0.001	0.0001	0.0025	0.0003	1e-05	5e-05	0.0015	0.002

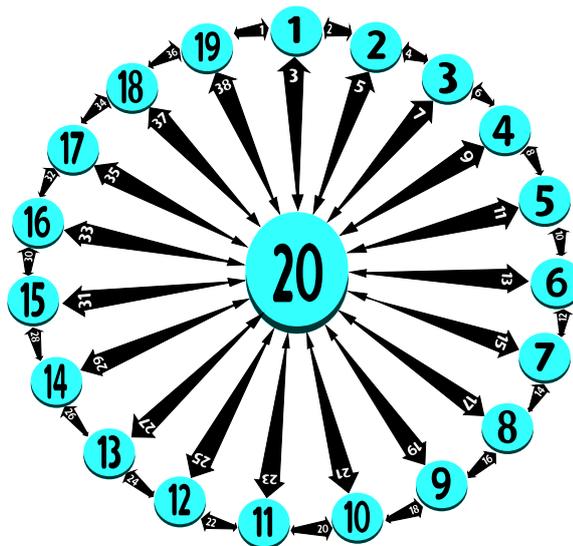


Figura. 6.2: Red spiderweb con 20 Nodos

La tabla 6.4 Contiene la configuración de tiempos para la red de 30 nodos y 58 enlaces (ver figura 6.4(c)), donde se asignaron tiempos a los enlaces siguiendo la lógica de los experimentos anteriores.

Tabla. 6.4: Matriz de retardos para la red de 30 nodos y 58 enlaces

Enlaces <i>E</i>	Iteraciones							
	ITE1	ITE2	ITE3	ITE4	ITE5	ITE6	ITE7	ITE8
1	0.0015	0.002	0.001	0.0025	1e-05	5e-05	0.0003	0.0001
2	5e-05	0.0025	0.0001	0.002	0.0003	0.0015	1e-05	0.001
3	0.0001	0.002	0.0003	1e-05	5e-05	0.0025	0.001	0.0015
4	0.001	5e-05	1e-05	0.0025	0.0001	0.0015	0.002	0.0003
5	0.0003	0.0025	0.0001	1e-05	0.002	0.0015	0.001	5e-05
6	0.002	0.0001	0.0025	1e-05	5e-05	0.0003	0.001	0.0015
7	1e-05	0.0003	0.0001	0.0025	0.002	5e-05	0.001	0.0015
8	1e-05	0.0001	0.002	0.001	0.0003	0.0025	0.0015	5e-05
9	0.0025	0.001	1e-05	0.0003	0.0015	0.0001	0.002	5e-05
10	0.002	1e-05	0.0003	0.001	0.0015	0.0001	5e-05	0.0025
11	1e-05	0.0003	0.001	0.002	0.0025	0.0001	0.0015	5e-05
12	1e-05	0.001	0.0025	0.0001	0.0003	5e-05	0.002	0.0015
13	0.0025	0.0003	0.0015	0.002	1e-05	0.0001	0.001	5e-05
14	0.002	1e-05	0.0003	0.0025	0.001	0.0015	0.0001	5e-05
15	0.001	0.0015	0.0001	0.002	5e-05	0.0003	1e-05	0.0025
16	0.0025	5e-05	1e-05	0.002	0.0003	0.001	0.0001	0.0015
17	0.0015	5e-05	1e-05	0.001	0.002	0.0003	0.0025	0.0001
18	0.0003	1e-05	0.002	0.0001	0.0015	0.0025	0.001	5e-05
19	0.0001	0.0015	0.002	0.001	0.0003	5e-05	1e-05	0.0025
20	0.002	0.001	0.0001	1e-05	0.0015	0.0003	0.0025	5e-05
21	0.0001	5e-05	0.0003	0.0025	0.0015	1e-05	0.001	0.002
22	0.0003	0.0015	1e-05	0.001	0.0025	5e-05	0.002	0.0001
23	0.001	1e-05	0.0015	0.0003	0.002	0.0025	0.0001	5e-05
24	0.002	0.0001	1e-05	0.0003	5e-05	0.001	0.0015	0.0025
25	1e-05	5e-05	0.0015	0.0003	0.001	0.002	0.0001	0.0025
26	0.0015	0.001	0.002	1e-05	0.0001	0.0025	0.0003	5e-05
27	0.0025	0.001	1e-05	0.002	5e-05	0.0003	0.0001	0.0015
28	0.0001	0.0025	0.001	5e-05	0.0003	1e-05	0.002	0.0015
29	0.0015	0.0025	0.0001	5e-05	0.002	0.0003	1e-05	0.001
30	0.0001	0.0015	0.0025	1e-05	0.001	0.0003	5e-05	0.002
31	0.0025	1e-05	0.0015	0.0001	5e-05	0.0003	0.001	0.002
32	5e-05	0.0015	0.0001	1e-05	0.0025	0.001	0.002	0.0003
33	0.002	0.001	1e-05	0.0015	0.0001	5e-05	0.0003	0.0025
34	0.0003	0.0015	0.001	0.002	5e-05	1e-05	0.0025	0.0001
35	0.0025	5e-05	0.0001	0.0003	1e-05	0.001	0.002	0.0015
36	0.0003	0.0001	0.0025	5e-05	0.002	1e-05	0.001	0.0015
37	1e-05	5e-05	0.0015	0.0003	0.0025	0.002	0.001	0.0001
38	0.001	0.0001	0.0025	0.0003	1e-05	5e-05	0.0015	0.002
39	0.0003	5e-05	1e-05	0.001	0.0025	0.002	0.0015	0.0001
40	0.002	0.0001	0.0015	0.0003	1e-05	5e-05	0.0025	0.001
41	0.0015	0.0003	5e-05	0.0025	0.002	1e-05	0.001	0.0001
42	0.0001	0.002	5e-05	0.0003	0.0025	0.0015	1e-05	0.001
43	0.0003	0.001	0.002	5e-05	0.0001	1e-05	0.0015	0.0025
44	0.001	1e-05	0.002	0.0003	5e-05	0.0025	0.0001	0.0015
45	0.002	5e-05	0.0001	0.0015	0.0003	0.001	1e-05	0.0025
46	0.001	0.0001	5e-05	0.002	0.0003	0.0025	1e-05	0.0015
47	0.001	0.0025	0.0003	5e-05	0.002	1e-05	0.0015	0.0001
48	0.001	1e-05	0.0001	0.002	5e-05	0.0015	0.0025	0.0003
49	0.0003	1e-05	0.0015	0.0025	0.002	5e-05	0.001	0.0001
50	0.0025	0.001	5e-05	0.0003	0.0001	1e-05	0.002	0.0015
51	0.0003	0.002	0.0025	0.0015	0.001	5e-05	1e-05	0.0001
52	0.0001	0.0003	0.002	5e-05	0.001	1e-05	0.0025	0.0015
53	0.002	5e-05	0.0015	0.0025	0.0001	0.001	0.0003	1e-05
54	0.0025	0.0001	0.0003	0.0015	0.002	0.001	1e-05	5e-05
55	0.001	0.0015	5e-05	1e-05	0.0003	0.0025	0.0001	0.002
56	0.0025	1e-05	0.002	0.001	0.0003	0.0001	5e-05	0.0015
57	0.0001	5e-05	0.0015	0.0025	0.0003	1e-05	0.001	0.002
58	0.0003	0.002	0.0001	5e-05	0.0025	1e-05	0.001	0.0015

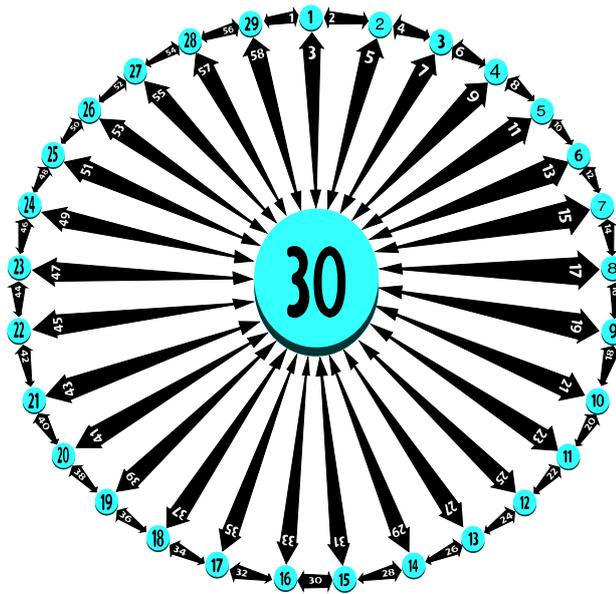


Figura. 6.3: Red spiderweb con 30 Nodos

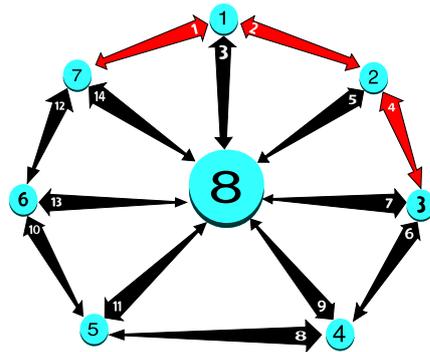
En el experimento con 30 nodos y 58 enlaces, se asignaron tiempos a los enlaces siguiendo la lógica de los experimentos anteriores.

En el caso de la red de 8 nodos y también en la de 20 nodos, se plantearon tres escenarios de fallo cada una. Sin embargo, para la red de 30 nodos, bastaron únicamente dos enlaces rotos para obtener los resultados esperados. Es importante señalar que se simuló una sola caída de enlace a la vez. Esto significa que cada vez que se simula una falla, se ejecutan los algoritmos pertinentes para determinar su rendimiento. Es relevante destacar que no se simularon redes con un número mayor de nodos, dado que el estándar sugiere redes independientes para el bus de proceso y el bus de estación con una cantidad de switches inferior a la establecida en el experimento; por lo tanto, el valor de 30 nodos resulta ser bastante elevado para este tipo de redes.

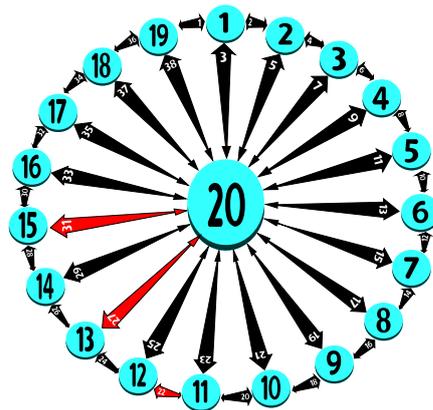
Teniendo en cuenta la topología de la red y con el objetivo de diversificar las soluciones encontradas, la elección de las demandas se llevó a cabo evitando que los nodos contiguos actuaran como origen y destino del mismo servicio, a excepción de la red de 8 nodos, donde debido al tamaño y la cantidad de servicios se utilizaron los nodos 2-3. Bajo esta premisa, se seleccionó aleatoriamente un nodo para cada servicio y luego se complementó asegurando un mínimo de 2 saltos.

Dado que los tiempos de los enlaces cambian en cada iteración (consultar Tablas 6.2, 6.3, 6.4), se seleccionaron enlaces de prueba para generar las fallas en cada experimento. La Tabla 6.5 detalla los enlaces que presentan fallas en cada topología,

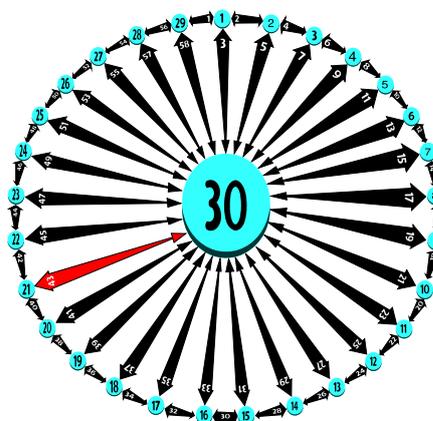
así como los servicios a desplegar entre cada par de nodos en las redes representadas en la Figura 6.4.



((a)) Red spiderweb 8 Nodos y 3 fallas



((b)) Red spiderweb 20 Nodos y 3 fallas



((c)) Red spiderweb 30 Nodos y 2 fallas

Figura. 6.4: Escenarios de prueba

Tabla. 6.5: Enlaces con falla para cada topología

Red	Enlaces rotos	Servicios	Nodo-(inicio, final)
8 nodos	1-2-4	GOOSE	(4, 1)
		SV	(8, 4)
		MMS	(1, 5)
		MMS	(2, 8)
		SV	(2, 3)
20 nodos	22-27-31	MMS	(5, 16)
		SV	(11, 13)
		MMS	(9, 19)
		SV	(14, 8)
		GOOSE	(19, 17)
30 nodos	16-43	GOOSE	(20, 1)
		SV	(8, 21)
		MMS	(2, 23)
		MMS	(10, 19)
		GOOSE	(7, 30)

Para evaluar la eficiencia de los algoritmos propuestos, se desactivaron ciertos enlaces de la topología asignándoles tiempos del orden de los 100 segundos. De esta manera, los algoritmos propuestos no considerarían estos enlaces como parte de la solución. Los resultados se presentan en tablas que muestran una primera exploración de rutas alternativas con todos los enlaces en la red (sin fallas), iterando sobre la matriz de retardo según el tamaño de la red (consultar Tablas 6.2, 6.3, 6.4). Posteriormente, se repitió el experimento desactivando los enlaces donde se simulaban las fallas (ver Tabla 6.5), y se realizó otra iteración sobre las tablas de retardo previas, cambiando la línea que representaba el enlace en estado de falla. Tomando como ejemplo la red de 8 nodos, se reportaron resultados **sin fallas** (todos los enlaces funcionando), así como para **falla 1** (reemplazando la fila 1 de la Tabla 6.2), **falla 2** (reemplazando la fila 2 de la Tabla 6.2) y **falla 4** (reemplazando la fila 4 de la Tabla 6.2).

6.1 Experimento 1: método exacto

Para implementar el método exacto, se utilizó el software gurobi-python [GUR23], evaluando tres métricas principales: (i) el tiempo necesario para encontrar una solución en cada escenario con sus diferentes fallas, (ii) el número de nodos utilizados para enviar la demanda, y (iii) el tiempo de la ruta resultante. Es fundamental resaltar que si no es posible satisfacer alguna de las demandas entre el nodo de origen y el destino, el modelo se torna inviable, lo que conlleva a la cancelación de todas las solicitudes de envío.

A continuación, se presenta la tabla resumen que muestra en cada fila el promedio de los enlaces utilizados para llevar las demandas de cada servicio. Al final de cada

columna se muestra el promedio de enlaces usados por todos los servicios para transportar la demanda entre el origen y el destino.

Tabla. 6.6: Promedio de enlaces usados en cada demanda ILP

8 nodos				
Demanda	Sin falla	Falla 1	Falla 2	Falla 4
GOOSE ->(4, 1)	3.375 \approx 4	2.875 \approx 3	3.375 \approx 4	2.37 \approx 3
SV ->(8, 4)	1.125 \approx 2	1.375 \approx 2	1.375 \approx 2	1.5 \approx 2
MMS ->(1, 5)	3	2.75 \approx 3	3.125 \approx 4	4.5 \approx 5
MMS ->(2, 8)	1.75 \approx 2	1.5 \approx 2	1.625 \approx 2	1.75 \approx 2
SV ->(2, 3)	1	1.625 \approx 2	1.5 \approx 2	1.12 \approx 2
Promedio	2.4 \approx 3	2.4 \approx 3	2.8 \approx 3	2.8 \approx 3
20 nodos				
Demanda	Sin falla	Falla 22	Falla 27	Falla 31
MMS ->(5, 16)	2.286 \approx 3	2.286 \approx 3	2.143 \approx 3	2
SV ->(11, 13)	2	3.57 \approx 4	3.286 \approx 4	2
MMS ->(9, 19)	6.43 \approx 7	7.286 \approx 8	7.286 \approx 8	5.57 \approx 6
SV ->(14, 8)	4	3.286 \approx 4	2.86 \approx 3	4
GOOSE ->(19, 17)	2.286 \approx 3	2.143 \approx 3	2.714 \approx 3	2.28 \approx 3
Promedio	3.8 \approx 4	4.4 \approx 4	4.2 \approx 5	3.4 \approx 4
30 nodos				
Demanda	Sin falla	Falla 16	Falla 43	—
GOOSE ->(20, 1)	3.375 \approx 4	3.625 \approx 4	3.5 \approx 4	—
SV ->(8, 21)	6.5 \approx 7	7.25 \approx 8	6	—
MMS ->(2, 23)	2.25 \approx 3	3.25 \approx 4	2.5 \approx 3	—
MMS ->(10, 19)	11.5 \approx 12	3	10	—
GOOSE ->(7, 30)	1	1.5 \approx 2	1	—
Promedio	5.4 \approx 6	4.2 \approx 5	4.8 \approx 5	—

Dado que en cada escenario solucionado con un algoritmo exacto se halla el óptimo, la Tabla 6.6 refleja el número mínimo promedio de enlaces necesarios en cada solución. Es importante tener en cuenta que este número se aproxima al entero superior más cercano, dado que tanto los nodos como los enlaces deben ser números enteros positivos. Se puede observar que el modelo ILP conserva casi en su totalidad el mismo número de enlaces en estado de falla, comparado con la red funcionando sin enlaces rotos.

Se debe tener en cuenta que a medida que aumenta el número de enlaces utilizados, se requiere un mayor número de nodos de tránsito. En una red real, es importante mantener las rutas lo más cortas posible para reducir el tiempo de retardo debido al procesamiento en los diferentes nodos. Sin embargo, teniendo en cuenta que el modelo propuesto tiene como objetivo minimizar la carga del enlace más utilizado cumpliendo un mínimo de tiempo, el uso de un mayor número de nodos y enlaces muestra un mejor balanceo de carga en cada escenario de prueba. La Tabla 6.7 muestra en cada fila el promedio de nodos utilizado en cada servicio y, al final de cada columna, el promedio de nodos utilizados por todos los servicios de la red en cada escenario planteado, por lo tanto, el número promedio de nodos aumenta

cuando se incrementa el tamaño de la red, dado que para la misma cantidad de demandas se cuentan con una mayor cantidad de caminos disponibles, lo que permite balancear la carga en los enlaces. Teniendo en cuenta lo anterior, la Tabla 6.7 representa la mínima cantidad de nodos necesaria para llevar las demandas entre origen y destino, garantizando el mejor balanceo de carga, por lo tanto, en este contexto entre mayor sea la cantidad de nodos mejor es la solución encontrada.

Tabla. 6.7: Promedio de nodos usados en cada demanda ILP

8 nodos				
Demanda	Sin falla	Falla 1	Falla 2	Falla 4
GOOSE ->(4, 1)	2.375 \approx 3	1.875 \approx 2	2.375 \approx 3	1.375 \approx 2
SV ->(8, 4)	0.125 \approx 1	0.375 \approx 1	0.875 \approx 1	0.375 \approx 1
MMS ->(1, 5)	2	1.75 \approx 2	1.625 \approx 2	3.5 \approx 4
MMS ->(2, 8)	0.75 \approx 1	0.5 \approx 1	0.625 \approx 1	0.75 \approx 1
SV ->(2, 3)	0	0.625 \approx 1	0.437 \approx 1	0.125 \approx 1
Promedio	1.4 \approx 2	1.4 \approx 2	1.6 \approx 2	1.8 \approx 2
20 nodos				
Demanda	Sin falla	Falla 22	Falla 27	Falla 31
MMS ->(5, 16)	1.286 \approx 2	1.428 \approx 2	1.143 \approx 2	1
SV ->(11, 13)	1	2.570 \approx 3	2.146 \approx 3	1
MMS ->(9, 19)	5.43 \approx 6	6.286 \approx 7	6.286 \approx 7	4.57 \approx 5
SV ->(14, 8)	3	2.428 \approx 3	2.143 \approx 3	3
GOOSE ->(19, 17)	1.286 \approx 2	1.143 \approx 2	1.714 \approx 2	1.286 \approx 2
Promedio	2.6 \approx 3	3.4 \approx 4	3.4 \approx 4	2.4 \approx 3
30 nodos				
Demanda	Sin falla	Falla 16	Falla 43	—
GOOSE ->(20, 1)	2.5 \approx 3	2.625 \approx 3	2.5 \approx 3	—
SV ->(8, 21)	5.75 \approx 6	6.25 \approx 7	5	—
MMS ->(2, 23)	1.25 \approx 2	2.25 \approx 3	1.5 \approx 2	—
MMS ->(10, 19)	10.5 \approx 11	2	9	—
GOOSE ->(7, 30)	0	0.5 \approx 1	0	—
Promedio	4.4 \approx 5	3.2 \approx 4	3.8 \approx 4	—

Los tiempos para encontrar una solución desempeñan un papel fundamental en las infraestructuras críticas, donde cada milisegundo es crucial. Si una ruta es viable, varios métodos pueden utilizarse indistintamente considerando las diferencias observadas. En la tabla 6.8 se presentan las métricas establecidas para determinar la respuesta del modelo ILP. En la primera columna, \overline{T}_e , se presenta el tiempo promedio en segundos que tarda el algoritmo exacto en encontrar una solución en cada red con diferentes fallas. La columna \overline{T}_r muestra el tiempo promedio en segundos que tardarán las demandas en ser llevadas en cada escenario, de acuerdo con las rutas halladas por el algoritmo. \overline{N}_e representa la suma de los promedios de enlaces usados por todos los servicios en la solución, \overline{N}_n indica la suma del promedio del número de nodos (saltos) empleados para garantizar los servicios en cada escenario de prueba, y \overline{B}_e representa la demanda en bits por el enlace mas usado en cada uno de los experimentos (es la representación de la variable R en el modelo planteado). Estas métricas ofrecen información sobre el rendimiento y la eficiencia del algoritmo en

diversos escenarios, incluyendo el tiempo empleado, los recursos utilizados y la satisfacción de las demandas de servicios en redes con diferentes tipos de fallas. Es importante resaltar que al final de cada columna se presenta promedio de cada una de las métricas en cada escenario de prueba.

Tabla. 6.8: Métricas ILP con solución exacta

		\overline{T}_e	\overline{T}_r	\overline{N}_e	\overline{N}_n	\overline{B}_e
8 nodos	Sin fallas	0.0402	0.00873	12	7	2456
	Falla 1	0.0443	0.00799	12	7	2430
	Falla 2	0.0377	0.00831	14	8	2508
	Falla 4	0.0426	0.00863	14	9	2482
	Promedio	0.0412	0.00841	13	7.75 \approx 8	2469
20 nodos	Sin fallas	0.0626	0.01098	19	14	1827
	Falla 22	0.0480	0.01288	22	17	1856
	Falla 27	0.0535	0.00953	21	17	1958
	Falla 31	0.0604	0.00821	17	12	1726
	Promedio	0.0561	0.0104	19.75 \approx 20	15	1842
30 nodos	Sin falla	0.1922	0.02096	27	22	2163
	Falla 16	0.8997	0.01488	21	16	2304
	falla 43	0.0816	0.01861	24	19	2111
	Promedio	0.3919	0.01814	24	19	2193

\overline{T}_e : Tiempo promedio de ejecución(segundos).
 \overline{N}_n : Número promedio de nodos.

\overline{T}_r : Tiempo promedio de rutas(segundos).
 \overline{B}_e : Bits promedio por el enlace mas usado.

\overline{N}_e : Número promedio de enlaces.

La tabla resumen presentada permite determinar el escenario óptimo en el uso de recursos de la red, incluyendo la cantidad de nodos y enlaces utilizados en caso de fallo de enlace en diferentes escenarios. Además, facilita la comparación de las métricas de tiempo de rutas y tiempo de ejecución con las obtenidas mediante el uso del algoritmo ACO presentado en el capítulo 6.2. Esta comparación nos permitirá evaluar la eficiencia de la solución propuesta en términos de rendimiento, tiempo de procesamiento y diversificación de rutas, lo que contribuirá a una toma de decisiones consciente sobre la selección del método más adecuado para resolver problemas similares en el futuro.

La solución exacta exhibe un promedio de ocupación del enlace más crítico similar en los diferentes escenarios, lo que valida el equilibrio de carga óptimo esperado en cada experimento.

La red de 30 nodos presenta variabilidad en los tiempos de ejecución, esto se debe a la cantidad de nodos que el modelo debe recorrer para alcanzar la solución óptima. Según lo indicado en la tabla 6.7, se necesitan 11 saltos para dirigir la demanda de uno de los servicios desde el origen hasta su destino.

6.2 Experimento 2: método metaheurístico

En este informe, se presentan los resultados de la aplicación del algoritmo ACO en la optimización de rutas en redes de datos de subestaciones eléctricas. Destacaremos su eficacia en la resolución de desafíos complejos relacionados con la planificación de rutas en las redes de datos que manejan la protección y el control de infraestructuras críticas para la distribución de energía.

Para la implementación del algoritmo, se ajustaron los parámetros heurísticos, estableciendo $\beta = 0,02$ y la influencia de la feromona en $\alpha = 0,7$. El algoritmo se probó inicialmente con una colonia definida utilizando la función exponencial 2^m , donde el valor de m se incrementa si el algoritmo no converge después de dos iteraciones del hormiguero. Es importante mencionar que, para instancias pequeñas con hasta 20 nodos, una colonia de 8 hormigas y 3 iteraciones resulta suficiente para lograr la convergencia. Sin embargo, en escenarios que contienen entre 20 y 40 nodos, se hace necesario aumentar el tamaño de la colonia a 16 individuos para lograr resultados adecuados. Por último, en redes con más de 40 nodos, la convergencia se vuelve más desafiante.

La siguiente tabla muestra el número promedio de enlaces utilizados para transportar cada demanda utilizando el algoritmo ACO. Se varía el número de nodos en la red y se observa cómo las fallas de los enlaces externos (enlaces pares) incrementan el número promedio de enlaces utilizados por el algoritmo para encontrar una solución, lo que implica un aumento en el tamaño de las rutas.

Cuando se produce un cambio en el número de enlaces en la red, la cantidad de nodos disponibles también cambia. Como consecuencia, se observa una variación en el uso de los nodos por parte del algoritmo ACO para cada demanda. Es importante mencionar la tendencia que tiene el algoritmo para encontrar soluciones haciendo uso de la misma cantidad de nodos para cada configuración en los diferentes estados de falla propuestos (ver Tabla 6.10)..

Tabla. 6.9: Promedio de enlaces usados en cada demanda ACO

8 nodos				
Demanda	Sin falla	Falla 1	Falla 2	Falla 4
GOOSE ->(4, 1)	3	2.50 ≈ 3	2.75 ≈ 3	2.625 ≈ 3
SV ->(8, 4)	1.62 ≈ 2	1.62 ≈ 2	1.75 ≈ 2	1.625 ≈ 2
MMS ->(1, 5)	2.75 ≈ 3	2.75 ≈ 3	2.62 ≈ 3	2.750 ≈ 3
MMS ->(2, 8)	1.25 ≈ 2	1.25 ≈ 2	1.25 ≈ 2	1.50 ≈ 2
SV ->(2, 3)	1.25 ≈ 2	1.50 ≈ 2	1.37 ≈ 2	1.125 ≈ 2
Promedio	2.4 ≈ 3	2.4 ≈ 3	2.4 ≈ 3	2.4 ≈ 3
20 nodos				
Demanda	Sin falla	Falla 22	Falla 27	Falla 31
MMS ->(5, 16)	3.14 ≈ 4	4	4.28 ≈ 5	4.42 ≈ 5
SV ->(11, 13)	2	5.42 ≈ 6	2	2.71 ≈ 3
MMS ->(9, 19)	5.14 ≈ 6	5	4.85 ≈ 5	4.85 ≈ 5
SV ->(14, 8)	4.57 ≈ 5	3.42 ≈ 4	3.42 ≈ 4	3.42 ≈ 4
GOOSE ->(19, 17)	3	2.85 ≈ 3	2.43 ≈ 3	2
Promedio	4	4.4 ≈ 5	3.8 ≈ 4	3.8 ≈ 4
30 nodos				
Demanda	Sin falla	Falla 16	Falla 43	—
GOOSE ->(20, 1)	3.25 ≈ 4	3.25 ≈ 4	4	—
SV ->(8, 21)	4.37 ≈ 5	6.62 ≈ 7	5.125 ≈ 6	—
MMS ->(2, 23)	4.50 ≈ 5	4.87 ≈ 5	4.125 ≈ 5	—
MMS ->(10, 19)	3.87 ≈ 4	5.75 ≈ 6	4.875 ≈ 5	—
GOOSE ->(7, 30)	1.37 ≈ 2	1.37 ≈ 2	1.375 ≈ 2	—
Promedio	4	4.8 ≈ 5	4.4 ≈ 5	—

Tabla. 6.10: Promedio de nodos usados en cada demanda ACO

8 nodos				
Demanda	Sin falla	Falla 1	Falla 2	Falla 4
GOOSE ->(4, 1)	2.375 ≈ 3	1.875 ≈ 2	2.375 ≈ 3	1.375 ≈ 2
SV ->(8, 4)	0.125 ≈ 1	0.375 ≈ 1	0.875 ≈ 1	0.375 ≈ 1
MMS ->(1, 5)	2	1.750 ≈ 2	1.625 ≈ 2	3.5 ≈ 4
MMS ->(2, 8)	0.750 ≈ 1	0.500 ≈ 1	0.625 ≈ 1	0.750 ≈ 1
SV ->(2, 3)	0	0.625 ≈ 1	0.437 ≈ 1	0.125 ≈ 1
Promedio	1.4 ≈ 2	1.4 ≈ 2	1.6 ≈ 2	1.8 ≈ 2
20 nodos				
Demanda	Sin falla	Falla 22	Falla 27	Falla 31
MMS ->(5, 16)	1.286 ≈ 2	1.428 ≈ 2	1.143 ≈ 2	1
SV ->(11, 13)	1	2.570 ≈ 3	2.146 ≈ 3	1
MMS ->(9, 19)	5.430 ≈ 6	6.286 ≈ 7	6.286 ≈ 7	4.57 ≈ 5
SV ->(14, 8)	3	2.428 ≈ 3	2.143 ≈ 3	3
GOOSE ->(19, 17)	1.286 ≈ 2	1.143 ≈ 2	1.714 ≈ 2	1.28 ≈ 2
Promedio	2.8 ≈ 3	3.4 ≈ 4	3.4 ≈ 4	2.4 ≈ 3
30 nodos				
Demanda	Sin falla	Falla 16	Falla 43	—
GOOSE ->(20, 1)	2.250 ≈ 3	2.250 ≈ 3	3	—
SV ->(8, 21)	3.375 ≈ 4	5.625 ≈ 6	4.125 ≈ 5	—
MMS ->(2, 23)	3.50 ≈ 4	3.875 ≈ 4	3.125 ≈ 4	—
MMS ->(10, 19)	2.875 ≈ 3	4.750 ≈ 5	3.875 ≈ 4	—
GOOSE ->(7, 30)	0.375 ≈ 1	0.375 ≈ 1	0.5 ≈ 1	—
Promedio	3	3.8 ≈ 4	3.4 ≈ 4	—

La implementación de algoritmos metaheurísticos ofrece rutas rápidas que cumplen con restricciones, aunque no garantiza la optimización absoluta. En la primera columna de la Tabla 6.11 \overline{T}_e , se muestra el tiempo promedio en segundos necesario para que el algoritmo ACO encuentre una solución en todas las redes afectadas por diversas fallas. La columna \overline{T}_r representa el tiempo promedio en segundos que lleva satisfacer las demandas en cada escenario, basándose en las rutas determinadas por el algoritmo. \overline{N}_e indica el promedio de enlaces utilizados en la solución, \overline{N}_n señala la cantidad de nodos o saltos empleados para asegurar los servicios en cada escenario de prueba, mientras que \overline{B}_e representa la demanda en bits por el enlace mas usado en cada uno de los experimentos. Estas métricas entregan información acerca del desempeño y la eficacia del algoritmo en una variedad de situaciones, incluyendo el tiempo invertido, los recursos empleados y la capacidad para satisfacer las demandas de servicios en redes con diversos fallos de enlace.

Tabla. 6.11: Métricas con algoritmo ACO

		\overline{T}_e	\overline{T}_r	\overline{N}_e	\overline{N}_n	\overline{B}_e
8 nodos	Sin fallas	0.00759	0.00881	12	7	2953
	Falla 1	0.01398	0.00948	12	7	2889
	Falla 2	0.01051	0.00100	12	8	2686
	Falla 4	0.00655	0.01020	12	9	2496
	Promedio	0.00965	0.00737	12	7.75 \approx 8	2756
20 nodos	Sin fallas	0.03167	0.015360	20	14	2279
	Falla 22	0.05381	0.017198	22	17	2602
	Falla 27	0.07835	0.015561	19	17	2047
	Falla 31	0.05618	0.015311	19	12	2207
	Promedio	0.05500	0.015857	20	15	2284
30 nodos	Sin falla	0.13650	0.009648	20	15	2508
	Falla 16	0.08057	0.013240	24	19	2419
	falla 43	0.15110	0.010073	22	17	2367
	Promedio	0.12272	0.010987	22	17	2432

\overline{T}_e : Tiempo promedio de ejecución(segundos). \overline{T}_r : Tiempo promedio de rutas(segundos). \overline{N}_e : Número promedio de enlaces.
 \overline{N}_n : Número promedio de nodos. \overline{B}_e : Bits promedio por el enlace mas usado.

El cambio de tamaño de la topología adoptada genera un nuevo desafío para el algoritmo ACO, dado que hace que la colonia de hormigas encuentres múltiples alternativas para llevar una demanda entre origen y destino, lo que contribuye a reducir la eficiencia del algoritmo por baja tasa de convergencia (dispersión en la colonia y en consecuencia, bajos niveles de feromona en los caminos).

6.3 Comparación entre método metaheurístico y método exacto

Los algoritmos propuestos fueron comparados en la tabla 6.12, donde se presenta el promedio de recursos utilizados en cada experimento por cada uno de los algoritmos.

El tiempo de ejecución promedio ($\overline{T_e}$) del algoritmo ACO es significativamente menor que el empleado por la solución al modelo de forma exacta en todos los escenarios, aunque muestra una variabilidad en los tiempos de ruta ($\overline{T_r}$), mostrando peor desempeño solo en la red de 20 nodos. Estos resultados son esperados teniendo en cuenta la naturaleza del modelo ILP planteado, dado que como objetivo se presenta el balanceo de carga y la diversificación de rutas para facilitar el restablecimiento del servicio en caso de falla.

Las columnas 5 y 6 representan el promedio de nodos y enlaces utilizados en cada escenario. En este caso, el modelo ILP presenta el número óptimo de enlaces que deben ser utilizados para garantizar la distribución de las demandas, representadas en la columna 7 (variable R del modelo propuesto), con el promedio del número de bits enviados por el enlace más cargado en cada uno de los escenarios. Es importante resaltar que la diferencia entre los resultados obtenidos no es lo suficientemente significativa como para descartar el uso de alguno de los métodos implementados.

Tabla. 6.12: Comparación de resultados de las Tablas 6.11 y 6.8.

		$\overline{T_e}$	$\overline{T_r}$	$\overline{N_e}$	$\overline{N_n}$	$\overline{B_e}$
8 nodos	ILP	0.0412	0.00841	13	$7.75 \approx 8$	2462
	ACO	0.00965	0.00737	12	$7.75 \approx 8$	2756
20 nodos	ILP	0.0561	0.0104	$19.75 \approx 20$	15	1842
	ACO	0.05500	0.015857	20	15	2284
30 nodos	ILP	0.3919	0.01814	24	19	2193
	ACO	0.12272	0.010987	22	17	2432

$\overline{T_e}$: Tiempo promedio de ejecución(segundos).
 $\overline{T_r}$: Tiempo promedio de rutas(segundos).
 $\overline{N_e}$: Número promedio de enlaces.
 $\overline{N_n}$: Número promedio de nodos.
 $\overline{B_e}$: Bits promedio por el enlace mas usado.

Los resultados obtenidos se presentan resumidos en varios diagramas que muestran la distribución de los datos, lo que proporciona una representación visual de la variabilidad y la tendencia de los resultados en cada caso.

La figura 6.5 presenta la distribución de los tiempos para llevar las demandas que tardan las rutas escogidas por los algoritmos en el escenario spiderweb 8. En este contexto, se aprecia una menor variabilidad en los datos entre los distintos escenarios del algoritmo ACO.

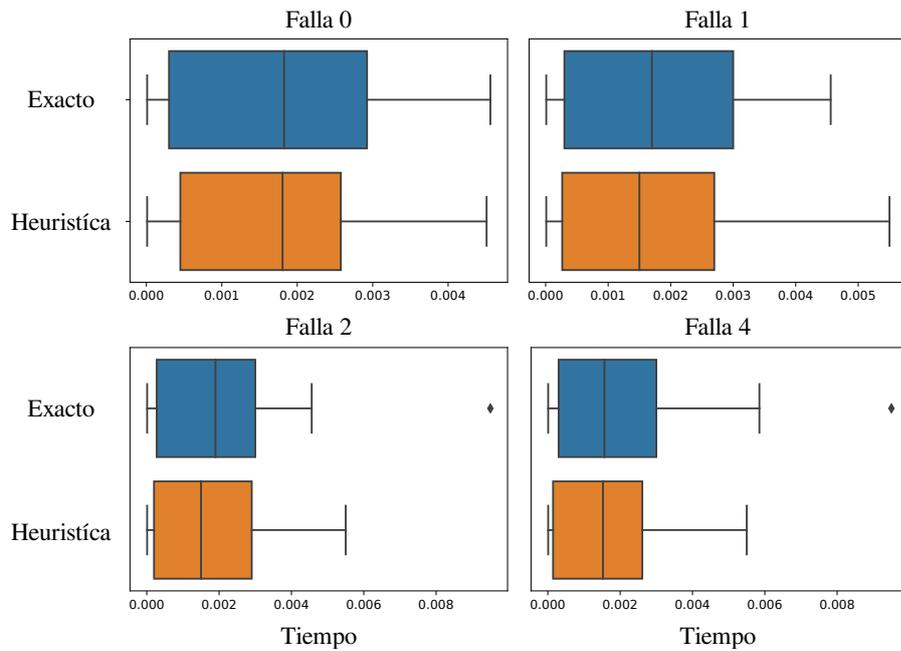


Figura. 6.5: Distribución de tiempos de ruta (segundos) para la red de 8 nodos

En el escenario de 20 nodos (ver figura 6.6), se observa una menor dispersión en los datos en ambas soluciones. El algoritmo ACO muestra convergencia en tiempos similares independientemente de las variaciones en los pesos de los enlaces. A diferencia del diagrama anterior, donde la distribución entre los métodos usados mantuvo una proporción constante en cada experimento, en este caso se evidencia una menor dispersión de los datos en cada escenario del algoritmo ACO.

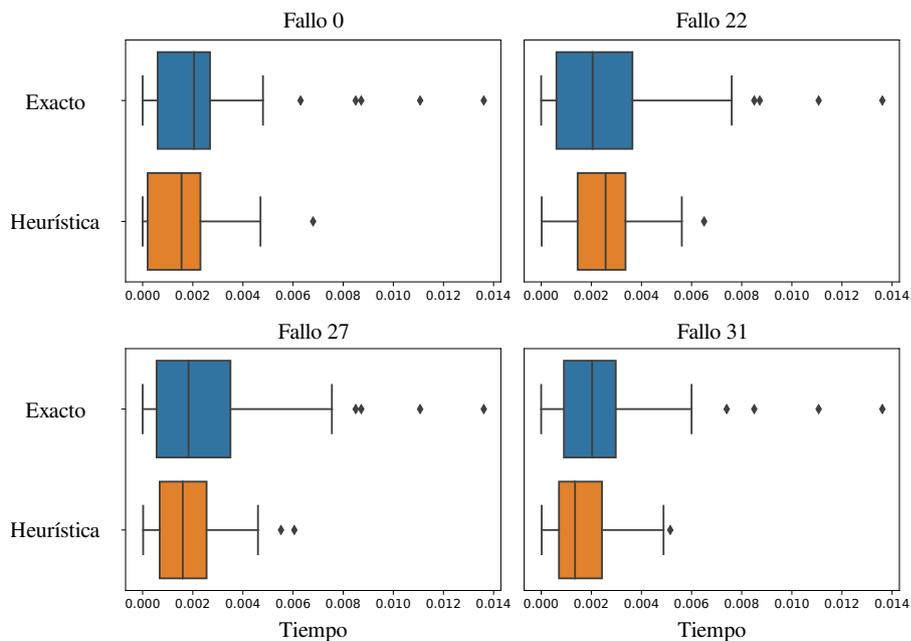


Figura. 6.6: Distribución de tiempos de ruta (segundos) para la red de 20 nodos

En la Figura 6.7 se presentan los datos obtenidos del escenario spiderweb 30, es evidente que, dependiendo de la asignación de pesos a cada enlace, el algoritmo exacto enfrenta dificultades al buscar soluciones. En este contexto, se observan tiempos atípicos y situaciones en las que el modelo exacto supera el tiempo razonable para encontrar una solución. Esta variabilidad en los resultados subraya la influencia crítica de la ponderación de los enlaces en el rendimiento del algoritmo.

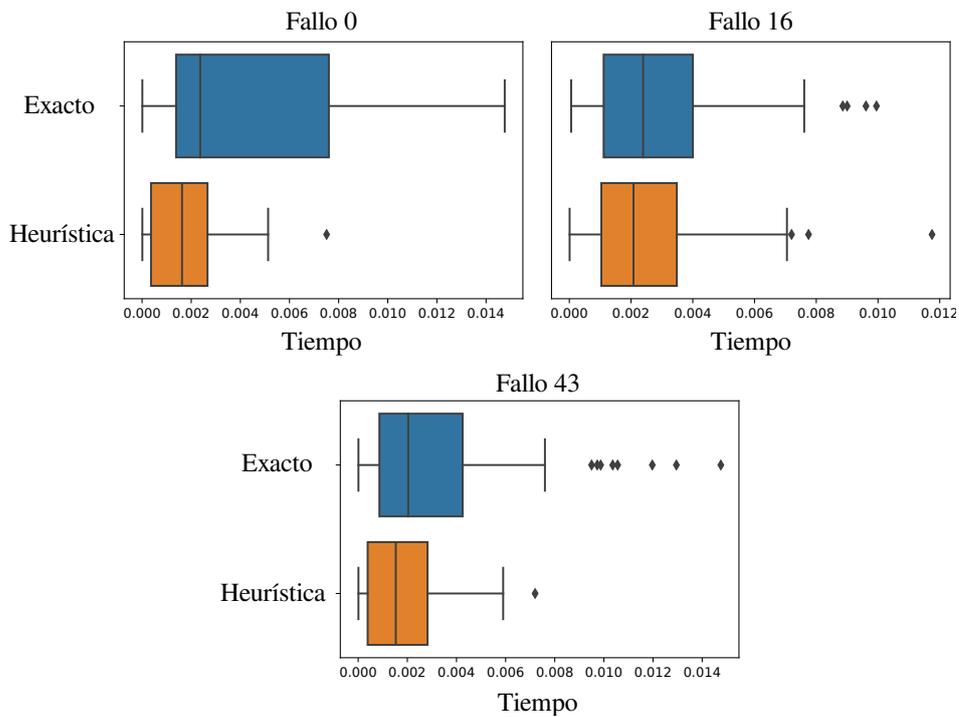


Figura. 6.7: Distribución de tiempos de ruta (segundos) para la red de 30 nodos

Para determinar si existe una diferencia significativa en los datos se realizó la prueba estadística U de Mann-Whitney, donde un $p - value > 0,05$ muestra que no hay una diferencia significativa. La prueba estadística de los datos en este escenario destaca de manera evidente la importancia del algoritmo ACO en la búsqueda de rutas óptimas. Además ACO demuestra su eficacia al ofrecer una mayor consistencia y un desempeño más robusto en comparación con el algoritmo exacto a esta escala. Es importante destacar que no se aumentó la cantidad de nodos en este escenario, ya que en redes con un mayor número de nodos, el algoritmo ACO experimentó problemas de convergencia, lo que resultó en tiempos de solución inviables de varios segundos. En contraste, el algoritmo exacto convergió, pero con tiempos promedio significativamente largos.

Tabla. 6.13: Resultados prueba U de Mann-Whithney ACO VS ILP

Resultados prueba estadística		
Red	Escenario	<i>P</i> – value
8 nodos	Sin fallas	0.8171
	Falla 1	0.4156
	Falla 2	0.4495
	Falla 4	0.5698
20 nodos	Sin fallas	0.1380
	Falla 22	0.6064
	Falla 31	0.3497
30 nodos	Sin falla	0.3342
	Falla 16	0.5379
	Falla 43	0.1732

La Figura 6.8 muestra el tiempo promedio requerido por cada algoritmo para encontrar una solución en los diversos escenarios.

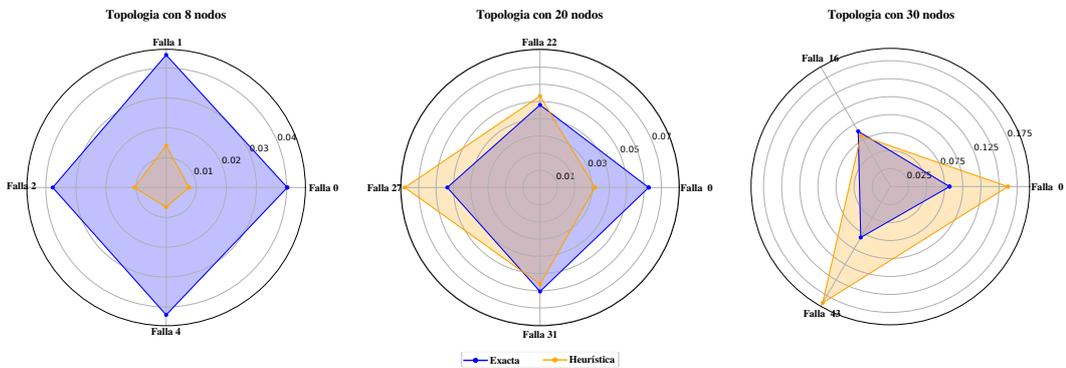


Figura. 6.8: Tiempos de ejecución

Dado que no se encuentran diferencias significativas en las rutas encontradas por ambos algoritmos, y considerando que los tiempos promedio para encontrar soluciones son más favorables con el algoritmo ACO, se recomienda la implementación de este método. Esto se debe a que el ACO logra soluciones en tiempos más cortos en comparación con el modelo ILP, y al mismo tiempo, las soluciones encontradas no muestran diferencias significativas en relación a las generadas por el modelo exacto en el contexto de la restauración de fallas en subestaciones eléctricas virtualizadas.

Conclusiones y trabajo futuro

El escenario de este proyecto de investigación fue diseñado considerando una infraestructura virtualizada. La rigidez de las redes convencionales en las subestaciones eléctricas, la necesidad de un personal operativo altamente capacitado y la intervención manual requerida para programar las funciones en los elementos de protección y control impulsaron el desarrollo de funciones virtualizadas utilizando contenedores en el contexto de las subestaciones eléctricas. Al mismo tiempo, la necesidad de garantizar la redundancia en los equipos físicos, conexiones y la disponibilidad de personal motivaron la formulación de los algoritmos propuestos.

En la elaboración de este capítulo, se tuvieron en cuenta los resultados observados en el capítulo 3. Se identificaron posibles escenarios de uso y se plantearon alternativas para futuros trabajos de investigación haciendo uso de paradigmas emergentes.

7.1 Conclusiones

El método exacto consideró la mayoría de las restricciones principales relacionadas con la naturaleza del problema de asignación de rutas, como las restricciones de capacidad en enlaces, caminos disjuntos y la minimización del máximo enlace ocupado para llevar las demandas.

El método exacto conduce a un resultado óptimo global. No obstante, de acuerdo con los resultados obtenidos, su tiempo de convergencia lo pone en desventaja en comparación con el método metaheurístico ACO para topologías pequeñas. Es importante tener en cuenta que los tiempos encontrados para los caminos son similares, pero el tiempo total empleado para encontrar la solución es superior. Considerando lo anterior, en las redes con menos de 20 nodos, se sugiere utilizar el algoritmo metaheurístico.

A pesar de que el algoritmo metaheurístico es más eficiente en términos de tiempo de cómputo, este método enfrenta problemas a medida que aumenta la cantidad de nodos. En particular, en los experimentos realizados en este trabajo de investigación se observó que, en el caso de más de 20 nodos, es necesario aumentar el tamaño de la colonia para mantener la feromona en el camino. De manera similar, cuando

el número de nodos se acerca a 50 (resultado no reportado en este trabajo, pero observado en pruebas de laboratorio), el método basado en ACO presenta problemas de convergencia, lo que justifica la tendencia en el estado del arte de combinar este método con otros [Den+22; DS19]. Por otro lado, el método exacto converge en todos los escenarios, aunque presenta tiempos muy elevados, por lo cual no es una opción viable en situaciones donde se deben cumplir los tiempos establecidos por el estándar IEC61850 [Zil13].

Se observaron casos donde el modelo exacto no encuentra una solución factible debido a que no fue posible transportar, de manera óptima, una o más demandas. De manera contraria, para estos mismos casos, el algoritmo metaheurístico permitió llevar al menos aquellas demandas para las cuales había encontrado un camino óptimo. Las demás demandas fueron señaladas por el algoritmo como no entregadas. Este aspecto hace que, comparativamente hablando, en casos donde no se encuentra una solución factible, el algoritmo metaheurístico sea el recomendado.

Esta solución ofrece la capacidad de ser utilizada de manera offline dado que requiere menor carga de procesamiento. En este enfoque, el algoritmo puede ejecutarse periódicamente, con una frecuencia determinada por un sistema de gestión, para monitorear el estado de la red, calcular rutas alternativas anticipándose a posibles fallos de enlace y menor riesgo de impacto en la operación del sistema online. La implementación del modelo online no se contempla debido a los breves tiempos requeridos para determinar la mejor ruta ante un estado de falla; además sería necesario simular la caída del enlace afectado en la red original para garantizar la eficiencia del modelo planteado.

7.2 Trabajo futuro

Las contribuciones presentadas en este trabajo de investigación han sido estructuradas de manera que pueden ser aplicadas en futuros proyectos de virtualización de redes tales como aquellos donde el sistema de gestión pueda aprovechar los resultados proporcionados por estas implementaciones en el controlador. En esta sección, se resumen las posibles mejoras y las aplicaciones futuras.

Dada la necesidad de garantizar un soporte operativo y de capital redundante, se puede considerar la implementación de un escenario híbrido funcional basado en las funciones virtualizadas en contenedores propuestas en el capítulo 1. Este enfoque abarcaría todos los servicios y aseguraría la comunicación entre los servicios virtualizados y los equipos existentes en las subestaciones eléctricas tradicionales. Si bien este proyecto de investigación ha demostrado el funcionamiento de las

funciones virtualizadas y su conectividad virtual entre un *publisher* y múltiples *subscribers*, es importante tener precaución al intervenir en el código proporcionado por las librerías IEC61850, ya que la arquitectura de cada solución dependerá de las necesidades específicas de la implementación.

Es importante realizar la integración de los resultados del algoritmo con el funcionamiento real de la red de comunicaciones en la subestación eléctrica

El algoritmo ACO deberá ser ajustado para mejorar su eficiencia en topologías con grandes números de nodos (> 50). De acuerdo con la literatura, resulta recomendable complementarlo con una metaheurística que almacene soluciones aceptables en cada iteración, lo que a su vez mejoraría la convergencia y permitiría su aplicación en escenarios de mayor complejidad.

EL enfoque expuesto sería particularmente beneficioso si se integra con tecnologías como SDN y NFV, que permitirían una gestión más dinámica y eficiente de la red. La combinación de estas tecnologías proporcionaría una solución integral para anticipar y gestionar los fallos de enlace en la red de datos de una subestación eléctrica virtualizada.

Dado que los modelos aquí presentados están diseñados para proporcionar datos comprensibles para un controlador SDN, se podría considerar la ejecución de estos algoritmos en un contenedor que pueda conocer la topología de la red mediante consultas periódicas en intervalos de tiempo reducidos. Esto habilitaría al controlador para hacer uso de rutas alternativas almacenadas en memoria con datos actualizados en caso de una falla, optimizando así el proceso de restablecimiento del servicio en tiempos más cortos.

Finalmente, al tener la flexibilidad de gestionar parcialmente la demanda mediante rutas sub óptimas utilizando el algoritmo meta heurístico, se podría implementar la instanciación de las demandas que no puedan ser acomodadas en un nodo cercano al destino, mientras se restaura el servicio. Esto podría llevarse a cabo haciendo uso de un controlador SDN, ampliando aún más las capacidades de la solución planteada.

Resumen

La economía del siglo XXI se enfoca en la eficiencia operativa, la apertura de nuevos mercados y una conciencia ambiental más estricta, a diferencia de los modelos económicos del pasado. Esta tendencia se observa claramente en el sector de los servicios públicos, especialmente en la industria eléctrica, que históricamente ha sido estática y conservadora en sus métodos.

Una red eléctrica opera como un conjunto de subsistemas interdependientes, que incluyen la generación de energía, la transmisión, la transformación de la energía a diferentes niveles de tensión y la distribución hasta llegar a los usuarios finales. Se hace hincapié en que la industria eléctrica se esfuerza por adaptarse a los paradigmas modernos, a pesar de mantener en uso técnicas que datan de hace más de 70 años.

Las subestaciones eléctricas tradicionales utilizan redes ethernet y redes punto a punto para el monitoreo y control de equipos, y que estas redes son esenciales para el funcionamiento continuo de la infraestructura eléctrica. Dado que las redes eléctricas se consideran infraestructuras críticas, se deben implementar sistemas redundantes y contar con personal altamente capacitado para garantizar la continuidad del servicio.

El costo de operar y mantener estas redes es considerable, tanto en términos de inversión de capital (CAPEX) como en costos operativos (OPEX). Esto se debe a la necesidad de equipos de respaldo, conexiones de alta densidad y personal capacitado. Ante esta problemática y la falta de flexibilidad en las redes de datos tradicionales en las subestaciones eléctricas, los operadores del servicio eléctrico han comenzado procesos de actualización tecnológica. Se experimenta una tendencia similar a la evolución de los centros de datos, que han adoptado tecnologías de virtualización como las redes definidas por software (SDN), máquinas virtuales (VM) y la virtualización de servicios (NFV).

La actualización tecnológica propuesta permite a las subestaciones eléctricas ser más flexibles y adaptables a las demandas cambiantes del sector eléctrico, lo que a su vez optimiza los recursos y mejora la calidad del servicio eléctrico ofrecido.

La Comisión Electrotécnica Internacional (IEC) lidera el proceso de modernización de las subestaciones eléctricas a través del estándar IEC61850. Donde se destaca la importancia de garantizar una comunicación eficiente en las subestaciones eléctricas y mediante la virtualización de funciones de subestación con NFV.

Una de las principales contribuciones de este trabajo de investigación, es la virtualización de funciones de red utilizando contenedores optimizados en tamaño, tiempo de ejecución y uso de recursos. Se plantea la problemática de los fallos en los enlaces de la red de comunicaciones en las subestaciones eléctricas y se menciona la formulación de un modelo matemático para abordar este problema, así como el uso de la optimización por colonia de hormigas como una meta-heurística para encontrar soluciones efectivas en instancias medianas y grandes.

En el contexto de la optimización de rutas en subestaciones eléctricas virtualizadas, se emplean dos enfoques fundamentales. En primer lugar, se utiliza la Programación Lineal Entera (ILP), con el solver Gurobi que usa un método exacto para buscar soluciones óptimas a problemas de optimización. Este enfoque se basa en representar el problema como una función objetivo lineal con restricciones enteras. Los ILP son idóneos cuando se requiere una solución óptima y son implementados mediante solvers como GLPK, CPLEX o GUROBI.

Por otro lado, se emplea una metaheurística inspirada en el comportamiento de las colonias de hormigas. En este modelo, las "*hormigas artificiales*" exploran rutas en un grafo, dejando feromonas que guían a otras. Este enfoque se basa en la probabilidad y la preferencia de las rutas, no garantiza soluciones exactas pero es efectivo para problemas complejos y ampliamente utilizado en la optimización de rutas en redes eléctricas y otros campos.

En la implementación, se virtualizan servicios como GOOSE, SV y MMS en contenedores, utilizando imágenes base livianas como BusyBox o Alpine. La elección de la imagen base se basa en el tamaño y la facilidad de encontrar librerías compatibles. Los servicios se ejecutan como contenedores independientes, optimizados para mantener un tamaño reducido y un inicio eficiente.

El modelo ILP se utiliza para asignar rutas y equilibrar la carga en la red virtualizada, garantizando un rendimiento óptimo. Mientras que la metaheurística de colonia de hormigas se integra para encontrar rutas eficientes y resolver problemas complejos de optimización de rutas en tiempos cortos.

Se llevaron a cabo varias simulaciones para evaluar el rendimiento de dos enfoques clave: (i) el desempeño del modelo ILP presentado en el capítulo 1 y (ii) un algoritmo metaheurístico ACO presentado en el capítulo 2. Estos experimentos se enfocaron en

restaurar el servicio en una red de subestación eléctrica con servicios virtualizados en caso de fallos en los enlaces. Se realizaron pruebas en topologías de red Spiderweb con diferentes números de nodos y enlaces.

En el primer experimento, se evaluó el método exacto (ILP) utilizando Gurobi-Python. Se analizaron tres métricas principales: el tiempo requerido para encontrar una solución, el número de nodos utilizados para enviar la demanda y el tiempo de la ruta resultante. El segundo experimento se basó en el método metaheurístico ACO. Se ajustaron parámetros heurísticos y se evaluaron métricas similares, como el número de enlaces utilizados y el tiempo de respuesta.

La comparación entre ambos métodos reveló que el algoritmo ACO superó al modelo exacto en términos de tiempo de respuesta y estabilidad en redes de menor complejidad. Sin embargo, a medida que la complejidad de la red aumentaba, el modelo exacto demostró una tendencia a encontrar soluciones más rápidamente. Se planteó la posibilidad de un modelo híbrido que combine ambos enfoques para aprovechar sus fortalezas en diferentes situaciones.

Estos experimentos proporcionaron información valiosa sobre la eficiencia y aplicabilidad de los enfoques ILP y ACO en la restauración de servicios en redes de subestación eléctrica, destacando la importancia de la elección de la estrategia en función de la complejidad de la red y los requisitos de tiempo críticos.

En conclusión este proyecto de investigación se enfocó en la infraestructura virtualizada en subestaciones eléctricas, lo cual se convierte en una solución clave para superar la rigidez de las redes convencionales y automatizar procesos de protección y control. Se desarrollaron funciones virtualizadas usando contenedores, y se consideró disminuir la redundancia de equipos y conexiones en las subestaciones tradicionales. En la evaluación de los dos métodos de solución propuestos para la asignación de rutas, se observó que el método exacto cumplía con restricciones críticas pero tenía tiempos de convergencia que no permiten su implementación, especialmente en topologías pequeñas. Sin embargo, el método ACO enfrenta desafíos a medida que aumentaba el número de nodos y permite encontrar soluciones parciales cuando el modelo es infactible, lo que sugiere la necesidad de ajustes y la posibilidad de combinar ambos enfoques.

Como trabajo futuro, se planteó la implementación de un escenario híbrido que integre funciones virtualizadas con equipos en subestaciones tradicionales, asegurando la comunicación entre ambos. Es importante mejorar la eficiencia del algoritmo ACO en topologías con un gran número de nodos y se recomienda complementarlo con una metaheurística que almacene en memoria soluciones aceptables. La ejecución de estos algoritmos en contenedores que permitan al controlador SDN conocer la

topología de la red y optimizar el restablecimiento del servicio en caso de fallas se plantea como una vía para reducir los tiempos de recuperación. Además, se sugiere instanciar demandas en diferentes nodos para cumplir con los servicios en nodos cercanos al destino mientras se restauran las fallas, haciendo uso de un controlador SDN. Estos enfoques abren posibilidades interesantes para futuros proyectos de virtualización y automatización en subestaciones eléctricas.

Bibliografía

- [AAU15] Abdullah Aydeger, Kemal Akkaya y A. Selcuk Uluagac. *SDN-based resilience for smart grid communications*. 2015 (vid. pág. 5).
- [AKV10] Amit Aggarwal, Swathi Kunta y Pramode K Verma. “A proposed communications infrastructure for the smart grid”. En: *2010 Innovative Smart Grid Technologies (ISGT)*. IEEE. 2010, págs. 1-5 (vid. pág. 1).
- [AS16] MD Ananth y Rinki Sharma. “Cloud management using network function virtualization to reduce capex and opex”. En: *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE. 2016, págs. 43-47 (vid. pág. 15).
- [Bar+03] Paul Barham et al. “Xen and the art of virtualization”. En: *ACM SIGOPS operating systems review* 37.5 (2003), págs. 164-177 (vid. pág. 9).
- [BEFEE16] Kamal Benzekki, Abdeslam El Fergougui y Abdelbaki Elbelrhiti Elalaoui. “Software-defined networking (SDN): a survey”. En: *Security and communication networks* 9.18 (2016), págs. 5803-5833 (vid. págs. 12, 13).
- [CB10] NM Mosharaf Kabir Chowdhury y Raouf Boutaba. “A survey of network virtualization”. En: *Computer Networks* 54.5 (2010), págs. 862-876 (vid. pág. 11).
- [Che+18] Haijun Chen et al. “Ant colony optimization with tabu table to solve TSP problem”. En: *2018 37th Chinese Control Conference (CCC)*. IEEE. 2018, págs. 2523-2527 (vid. págs. 19, 33, 38).
- [Che+19] Jing Chen et al. “Optimization of transportation routing problem for fresh food by improved ant colony algorithm based on tabu search”. En: *Sustainability* 11.23 (2019), pág. 6584 (vid. págs. 19, 34, 38).
- [Den+22] Xia Deng et al. “An ant colony optimization-based routing algorithm for load balancing in leo satellite networks”. En: *Wireless Communications and Mobile Computing* 2022 (2022) (vid. págs. 19, 27, 36, 38, 70).
- [DG97] Marco Dorigo y Luca Maria Gambardella. “Ant colony system: a cooperative learning approach to the traveling salesman problem”. En: *IEEE Transactions on evolutionary computation* 1.1 (1997), págs. 53-66 (vid. pág. 19).
- [DS19] Marco Dorigo y Thomas Stützle. *Ant colony optimization: overview and recent advances*. Springer, 2019 (vid. págs. 27, 33, 49, 70).
- [Far09] Hassan Farhangi. “The path of the smart grid”. En: *IEEE power and energy magazine* 8.1 (2009), págs. 18-28 (vid. pág. 23).

- [GK06] Fred W Glover y Gary A Kochenberger. *Handbook of metaheuristics*. Vol. 57. Springer Science & Business Media, 2006 (vid. pág. 49).
- [Gri18] Leonard L Grigsby. *The electric power engineering handbook-five volume set*. CRC press, 2018 (vid. pág. 2).
- [Gut+23] Sergio Armando Gutiérrez et al. “Next-generation power substation communication networks: IEC 61850 meets programmable networks”. En: *IEEE Power and Energy Magazine* 21.5 (2023), págs. 58-67 (vid. pág. 25).
- [HB16] Juliver Gil Herrera y Juan Felipe Botero. “Resource allocation in NFV: A comprehensive survey”. En: *IEEE Transactions on Network and Service Management* 13.3 (2016), págs. 518-532 (vid. pág. 14).
- [HC22] Yong-Xuan Huang y Jerry Chou. “A survey of NFV network acceleration from ETSI perspective”. En: *Electronics* 11.9 (2022), pág. 1457 (vid. pág. 14).
- [Hon+21] Junho Hong et al. “Implementation of secure sampled value (SeSV) messages in substation automation system”. En: *IEEE Transactions on Power Delivery* 37.1 (2021), págs. 405-414 (vid. pág. 25).
- [HS16] Jun-Ho Huh y Kyungryong Seo. “Design and test bed experiments of server operation system using virtualization technology”. En: *Human-centric Computing and Information Sciences* 6 (2016), págs. 1-21 (vid. pág. 9).
- [Hua+23] Meng-Yu Huang et al. “MILP Acceleration: A Survey from Perspectives of Simplex Initialization and Learning-Based Branch and Bound”. En: *Journal of the Operations Research Society of China* (2023), págs. 1-55 (vid. pág. 16).
- [Hwa+13] Jinho Hwang et al. “A component-based performance comparison of four hypervisors”. En: *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE. 2013, págs. 269-276 (vid. pág. 9).
- [JYG19] JQ James, Wen Yu y Jiatao Gu. “Online vehicle routing with neural combinatorial optimization and deep reinforcement learning”. En: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), págs. 3806-3817 (vid. págs. 33, 38).
- [KMK22] Karamjeet Kaur, Veenu Mangat y Krishan Kumar. “A review on Virtualized Infrastructure Managers with management and orchestration features in NFV architecture”. En: *Computer Networks* 217 (2022), pág. 109281 (vid. pág. 14).
- [Kre+14] Diego Kreutz et al. “Software-defined networking: A comprehensive survey”. En: *Proceedings of the IEEE* 103.1 (2014), págs. 14-76 (vid. págs. 3, 13).
- [LB16a] Erwin Alexander Leal y Juan Felipe Botero. “S3N-Smart Solution for Substation Networks, an architecture for the management of communication networks in power substations”. En: *Management and Security in the Age of Hyperconnectivity: 10th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2016, Munich, Germany, June 20-23, 2016, Proceedings 10*. Springer. 2016, págs. 52-56 (vid. págs. 3, 13, 24, 25, 29, 30, 32).
- [LB16b] Erwin Alexander Leal y Juan Felipe Botero. “Software defined power substations: An architecture for network communications and its control plane”. En: *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*. 2016, págs. 1-6 (vid. pág. 5).

- [LB16c] Erwin Alexander Leal y Juan Felipe Botero. “Software defined power substations: An architecture for network communications and its control plane”. En: *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE. 2016, págs. 1-6 (vid. pág. 29).
- [LB16d] Erwin Alexander Leal y Juan Felipe Botero. “Transforming communication networks in power substations through SDN”. En: *IEEE Latin America Transactions* 14.10 (2016), págs. 4409-4415 (vid. págs. 4, 5).
- [LB19] Alexander Leal y Juan Felipe Botero. “Defining a reliable network topology in software-defined power substations”. En: *IEEE Access* 7 (2019), págs. 14323-14339 (vid. págs. 5, 27, 39, 51).
- [LDB20] Alexander Leal, Mario Durán y Juan Felipe Botero. “Reliability provision in software defined power substations communication networks”. En: *Computer Networks* 181 (2020), pág. 107560 (vid. págs. 4, 5, 31, 32).
- [Lee14] Gary Lee. *Cloud networking: Understanding cloud-based data center networks*. Morgan Kaufmann, 2014 (vid. pág. 11).
- [Liu+15] Xinmeng Liu et al. “A low-complexity routing algorithm based on load balancing for LEO satellite networks”. En: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE. 2015, págs. 1-5 (vid. pág. 37).
- [MGK21] Priti Maratha, Kapil Gupta y Pratyay Kuila. “Energy balanced, delay aware multi-path routing using particle swarm optimisation in wireless sensor networks”. En: *International Journal of Sensor Networks* 35.1 (2021), págs. 10-22 (vid. pág. 15).
- [MHS21] Seyedakbar Mostafavi, Vesal Hakami y Maryam Sanaei. “Quality of service provisioning in network function virtualization: a survey”. En: *Computing* 103 (2021), págs. 917-991 (vid. pág. 14).
- [Mij+15] Rashid Mijumbi et al. “Network function virtualization: State-of-the-art and research challenges”. En: *IEEE Communications surveys & tutorials* 18.1 (2015), págs. 236-262 (vid. pág. 3).
- [Mis+15] Normi Salwana Miswan et al. “Interoperability testing for Digital Substation in Smart Grid domain: A power utility perspective”. En: *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*. IEEE. 2015, págs. 154-158 (vid. pág. 25).
- [Mug98] Stephen Muggleton. “Advances in ILP theory and implementations”. En: *Inductive Logic Programming: 8th International Conference, ILP-98 Madison, Wisconsin, USA, July 22–24, 1998 Proceedings* 8. Springer. 1998, págs. 9-9 (vid. pág. 16).
- [Nau+12] Bram Naudts et al. “Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network”. En: *2012 European workshop on software defined networking*. IEEE. 2012, págs. 67-72 (vid. pág. 13).
- [NPH14] Jeyasingam Nivethan, Mauricio Papa y Peter Hawrylak. “Modeling and simulation of electric power substation employing an IEC 61850 network”. En: *Proceedings of the 9th annual cyber and information security research conference*. 2014, págs. 89-92 (vid. pág. 23).

- [ONC15] Agwu Chukwuemeka Odi, Nweso Emmanuel Nwogbaga y NO Chukwuka. “The proposed roles of VLAN and inter-VLAN routing in effective distribution of network services in Ebonyi State University”. En: *International Journal of Science and Research* 4 (2015), págs. 2608-2615 (vid. pág. 11).
- [PG02] Ivan Pepelnjak y Jim Guichard. *MPLS and VPN architectures*. Vol. 1. Cisco press, 2002 (vid. pág. 11).
- [RAB19] AU Rehman, Rui L Aguiar y Joao Paulo Barraca. “Network functions virtualization: The long road to commercial deployments”. En: *IEEE Access* 7 (2019), págs. 60439-60464 (vid. pág. 14).
- [Rid+21] M Ridwan et al. “Review of digital substation equipment and technical specification in Indonesia”. En: *IOP Conference Series: Materials Science and Engineering*. Vol. 1098. 4. IOP Publishing. 2021, pág. 042053 (vid. pág. 25).
- [RNB21] Dennis Rösch, Steffen Nicolai y Peter Bretschneider. “Combined simulation and virtualization approach for interconnected substation automation”. En: *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE. 2021, págs. 1-6 (vid. pág. 5).
- [Rös+22] Dennis Rösch et al. “Delay modeling for virtualization-based Co-simulation of IEC 61850 substations”. En: *Journal of Cleaner Production* 368 (2022), pág. 133179 (vid. págs. 4, 31, 32).
- [Rus+20] Krzysztof Rusek et al. “Routenet: Leveraging graph neural networks for network modeling and optimization in sdn”. En: *IEEE Journal on Selected Areas in Communications* 38.10 (2020), págs. 2260-2270 (vid. págs. 35, 38).
- [SDF20] Rafael Souza, Kelvin Dias y Stenio Fernandes. “NFV data centers: A systematic review”. En: *IEEE Access* 8 (2020), págs. 51713-51735 (vid. pág. 3).
- [SOH22] Petr Stodola, Pavel Otřisal y Kamila Hasilová. “Adaptive ant colony optimization with node clustering applied to the travelling salesman problem”. En: *Swarm and Evolutionary Computation* 70 (2022), pág. 101056 (vid. págs. 19, 27, 37, 38).
- [SRK21] GA Senthil, Arun Raaza y N Kumar. “Internet of things multi hop energy efficient cluster-based routing using particle swarm optimization”. En: *Wireless Networks* 27 (2021), págs. 5207-5215 (vid. págs. 19, 27, 36, 38).
- [Tal09] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009 (vid. pág. 19).
- [Wan+19] Houtian Wang et al. “A load balanced routing algorithm based on congestion prediction for LEO satellite networks”. En: *Cluster Computing* 22 (2019), págs. 8025-8033 (vid. pág. 37).
- [Wes+01] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001 (vid. pág. 20).
- [WKR17] Robert Wojtowicz, Ryszard Kowalik y Desire D Rasolomampionona. “Next generation of power system protection automation—virtualization of protection systems”. En: *IEEE Transactions on Power Delivery* 33.4 (2017), págs. 2002-2010 (vid. págs. 4, 31, 32).

- [XXH15] Fei Xia, Zongze Xia y Xiaobo Huang. “Summary of GOOSE substation communication”. En: *MATEC Web of Conferences*. Vol. 25. EDP Sciences. 2015, pág. 01007 (vid. pág. 25).
- [Zha+19] Huizhen Zhang et al. “A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows”. En: *Information Sciences* 490 (2019), págs. 166-190 (vid. págs. 19, 34, 38).

Websites

- [Doc23] Inc Docker. *Use containers to Build, Share and Run your applications*. 2023. URL: <https://www.docker.com/resources/what-container/> (visitado 3 de ago. de 2023) (vid. págs. 3, 10).
- [GUR23] GUROBI. *GUROBI OPTIMIZATION*. 2023. URL: <https://www.gurobi.com/downloads/gurobi-software/> (visitado 10 de sep. de 2023) (vid. págs. 18, 57).
- [IBM23] IBM. *IBM CPLEX Optimizer*. 2023. URL: <https://www.ibm.com/mx-es/analytics/cplex-optimizer> (visitado 10 de sep. de 2023) (vid. pág. 18).
- [PRO23] GNU PROYECT. *GLPK (GNU Linear Programming Kit)*. 2023. URL: <https://www.gnu.org/software/glpk/> (visitado 10 de sep. de 2023) (vid. pág. 18).
- [VMw23] Inc VMware. *Virtualización de servidores*. 2023. URL: <https://www.vmware.com/es/products/vsphere/data-center-consolidation.html> (visitado 3 de ago. de 2023) (vid. págs. 3, 10).
- [Zil13] Michael Zillgith. *libIEC61850/lib60870*. 2013 - 2022. URL: <https://libiec61850.com/downloads/> (visitado 10 de sep. de 2023) (vid. págs. 40, 70).

Declaracion

Por la presente confirmo que soy el único autor del trabajo escrito aquí adjunto y que lo he compilado con mis propias palabras y no he utilizado fuentes o medios excepto los que se enumeran en la Bibliografía.

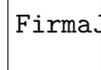
Medellín, Colombia - 27 de abril de 2024

Estudiante,

 sign.pdf

Ing. Wilmar Alesander Vásquez Barrientos

Asesor,

 Firma.JFB1.png

