



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Cinvestav Tamaulipas

**Selección Conjunta de Rutas y
Divisiones Funcionales para
Arquitecturas Cloud-RAN**

Tesis que presenta:

Cristian Camilo Erazo Agredo

Para obtener el grado de:

**Maestro en Ciencias
en Ingeniería y Tecnologías
Computacionales**

Dr. Javier Rubio Loyola, Co-Director
Dr. Mario Garza Fabre, Co-Director

© Derechos reservados por
Cristian Camilo Erazo Agredo
2020

La tesis presentada por Cristian Camilo Erazo Agredo fue aprobada por:

Dr. Hiram Galeana Zapién

Dr. Carlos Alberto Lara Álvarez

Dr. Javier Rubio Loyola, Co-Director

Dr. Mario Garza Fabre, Co-Director

Ciudad Victoria, Tamaulipas, México., 30 de Junio de 2020

A mi familia y amigos

Agradecimientos

Le agradezco a mi familia, a mi madre, mi padre, mi hermana, mi hermano y mi sobrino, pues sin su apoyo y cuidado, hoy no estaría aquí.

Le agradezco a mis viejos y nuevos amigos, gracias a esos consejos inesperados y al apoyo brindado.

Le agradezco al personal administrativo del CINVESTAV Unidad Tamaulipas que desde el inicio de este proceso, me han brindado ayuda y asesoramiento.

Le agradezco a mis revisores, los doctores Hiram Galeana Zapién y Carlos Alberto Lara Álvarez, quienes validaron y ayudaron a mejorar el trabajo realizado y presentado en este documento.

Le agradezco a los doctores del CINVESTAV Unidad Tamaulipas, quienes me brindaron nuevos conocimientos y en especial a los doctores Javier Rubio Loyola y Mario Garza Fabre quienes me asesoraron y dirigieron para poder cumplir con este trabajo de tesis.

Le agradezco al CONACyT por el apoyo económico parcial que me brindó, gracias a su ayuda pude terminar mis estudios en el CINVESTAV Unidad Tamaulipas.

Le agradezco a Karina Gazca, sin sus consejos, alegría y cariño, esta historia no sería igual.

Finalmente, le agradezco a todas aquellas personas que en algún momento me brindaron de su ayuda, a todos muchas gracias.

Índice General

Índice General	I
Índice de Figuras	III
Índice de figuras	III
Índice de Tablas	v
Índice de tablas	v
Índice de Algoritmos	vii
Publicaciones	ix
Resumen	xi
Abstract	xiii
Nomenclatura	xv
1. Introducción	1
1.1. Nueva generación de redes celulares	1
1.1.1. Cloud Radio Access Network	2
1.1.2. Virtualización de redes	6
1.2. Motivación de la tesis	6
1.3. Planteamiento del problema	8
1.4. Pregunta de investigación e hipótesis	9
1.5. Objetivos generales y particulares	10
1.6. Metodología	11
1.7. Contribuciones	12
1.8. Organización del documento	13
2. Marco teórico y estado del arte	15
2.1. Marco teórico	15
2.1.1. Divisiones funcionales	16
2.1.1.1. Funciones de la estación base	16
2.1.1.2. Divisiones de las funcionalidades	17
2.1.2. Problema de incrustación de redes virtuales	21
2.1.2.1. Enfoques del problema de VNE	22
2.1.3. Optimización e inteligencia computacional	23

2.1.3.1.	Optimización	23
2.1.3.2.	Clasificación de los problemas de optimización	24
2.1.3.3.	Técnicas de optimización	25
2.1.3.4.	Algoritmos evolutivos	27
2.2.	Estado del arte	29
2.2.1.	Trabajos relacionados con el enfoque propuesto	36
3.	Modelado formal del problema	39
3.1.	Modelo propuesto	39
3.1.1.	Red física	40
3.1.2.	Peticiones de recursos virtuales	42
3.1.3.	Mapeo de recursos físicos y virtuales	43
3.1.4.	Solución candidata	44
3.2.	Formulación del problema de optimización	46
3.2.1.	Función objetivo	46
3.2.2.	Restricciones	47
3.3.	Espacio de búsqueda	50
3.4.	Resumen y conclusiones	51
4.	Métodos de solución propuestos	53
4.1.	Algoritmo de búsqueda exhaustiva	54
4.2.	Algoritmo avaro	58
4.3.	Algoritmo evolutivo	61
4.3.1.	Representación de las soluciones	61
4.3.2.	Inicialización de la población	62
4.3.3.	Criterio de optimización	63
4.3.4.	Selección para variación	63
4.3.5.	Operadores de variación	63
4.3.6.	Selección para supervivencia y manejo de restricciones	64
4.4.	Resumen y conclusiones	65
5.	Evaluación y resultados	67
5.1.	Parámetros de evaluación	67
5.1.1.	Escenarios de prueba	68
5.1.2.	Algoritmos evaluados y configuración de parámetros	70
5.1.3.	Evaluación de desempeño	72
5.1.3.1.	Calidad de la solución	72
5.1.3.2.	Tasa de aceptación	72
5.1.3.3.	Tiempo de ejecución	73
5.1.3.4.	Análisis de significancia estadística	73
5.2.	Resultados	74
5.2.1.	Desempeño general	75
5.2.1.1.	Ejemplo de una instancia de prueba y una solución encontrada	77

5.2.2.	Comportamiento convergente	80
5.2.3.	Capacidad de producir soluciones cercanas al óptimo	82
5.2.4.	Eficiencia computacional	84
5.2.5.	Análisis estadístico	86
5.3.	Resumen y conclusiones	87
6.	Conclusiones y trabajo futuro	89
6.1.	Conclusiones	89
6.2.	Trabajo futuro	92
A.	Resultados obtenidos del proceso de ajuste de parámetros	93
A.1.	Resultados del ajuste de parámetros para los escenarios pequeños	94
A.2.	Resultados obtenidos del ajuste de parámetros para los escenarios medianos.	95
A.3.	Resultados obtenidos para el ajuste de parámetros de los escenarios grandes.	96

Índice de figuras

1.1.	Escenarios posibles del concepto WNC	3
1.2.	Ejemplo de una RAN virtualizada con NGFI	5
2.1.	Clasificación de los algoritmos metaheurísticos más comúnmente utilizados en la literatura	27
2.2.	Esquema general del funcionamiento de un algoritmo evolutivo	28
4.1.	Ejemplo de la representación para una solución del problema	55
4.2.	Ejemplo de una red física o red de sustrato	56
4.3.	Ejemplo de una red virtual o petición de recursos de un vMNO. La petición consiste de tres DUs virtuales (r_1^v , r_2^v y r_3^v), dos CUs virtuales (c_1^v y c_2^v) y tres enlaces virtuales ($e_{c_1^v, r_1^v}$, $e_{c_1^v, r_2^v}$ y $e_{c_2^v, r_3^v}$).	56
4.4.	Ejemplo de una asignación para una petición de recursos	56
4.5.	Diseño del algoritmo de búsqueda exhaustiva para identificar soluciones óptimas del problema de SCRDF	57
4.6.	Diseño del algoritmo avaro propuesto para abordar el problema de SCRDF	59
4.7.	Diseño del algoritmo evolutivo que se propone para abordar el problema de SCRDF	62
5.1.	Resumen de los resultados obtenidos por los enfoques AA , AE^{gre} y AE^{rnd}	76
5.2.	Ejemplo de una instancia de prueba de tamaño pequeño	78
5.3.	Mejor solución encontrada por el algoritmo AE^{gre} para la instancia de prueba de tamaño pequeño ilustrada en la Figura 5.2	79
5.4.	Gráficas de convergencia de los algoritmos AE^{gre} y AE^{rnd}	81
5.5.	Comparación de los resultados obtenidos por los enfoques AA , AE^{gre} y AE^{rnd} con respecto a la solución óptima (OPT)	83
5.6.	Tiempos de ejecución registrados durante los experimentos de este trabajo de tesis	85
A.1.	Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias pequeñas seleccionadas	95
A.2.	Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias medianas seleccionadas	96
A.3.	Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias grandes seleccionadas	97

Índice de tablas

2.1.	Requerimientos de ancho de banda y latencia para cada división funcional	21
2.2.	Tabla de resumen del estado del arte	36
2.3.	Tabla comparativa entre los trabajos seleccionados de la literatura y el modelo propuesto	38
3.1.	Notación usada para los componentes principales del modelo propuesto	41
5.1.	Parámetros empleados para generar las redes físicas o redes de sustrato	70
5.2.	Parámetros empleados para generar los conjuntos de peticiones de recursos virtuales	70
5.3.	Valores empleados para las restricciones de ancho de banda y latencia para cada una de las divisiones funcionales	70
5.4.	Configuración de parámetros seleccionados para los algoritmos AE^{gre} y AE^{rnd} . . .	71
5.5.	Resumen de los resultados del análisis de significancia estadística realizado mediante el uso de la prueba de Mann-Whitney	86
6.1.	Resumen del cumplimiento de los objetivos particulares de este trabajo de tesis . . .	91
A.1.	Valores empleados en el proceso de ajuste de parámetros	94

Índice de Algoritmos

1.	Algoritmo de Jerarquización Estocástica (SR)	65
----	--	----

Publicaciones

Cristian Erazo-Agredo, Mario Garza-Fabre, Ramón Agüero, Luis Diez, Joan Serrat, and Javier Rubio-Loyola. Joint Route Selection and Split Level Management for 5G C-RAN Architectures. IEEE Transactions on Network and Service Management, 2020. (actualmente en proceso de revisión)

Selección Conjunta de Rutas y Divisiones Funcionales para Arquitecturas Cloud-RAN

por

Cristian Camilo Erazo Agredo

Unidad Cinvestav Tamaulipas

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2020

Dr. Mario Garza Fabre, Co-Director

Dr. Javier Rubio Loyola, Co-Director

Las redes celulares de quinta generación (5G) han sido propuestas para afrontar las crecientes demandas de tráfico de red de alta velocidad y servicios de alta calidad. La parte de las redes celulares que deben modificarse para manejar dicho crecimiento se llaman redes de acceso por radio (RAN). *Cloud Radio Access Network* (C-RAN) es un paradigma que ha sido propuesto con la intención de virtualizar las funcionalidades de las estaciones base para aumentar la flexibilidad de la red móvil, al agrupar las unidades de procesamiento en grupos de recursos. En C-RAN, las funcionalidades de las estaciones base están separadas, dando lugar a las unidades centrales (CUs), que están a cargo del procesamiento de la banda base, y a las unidades distribuidas (DUs), que están a cargo de las funciones de radiofrecuencia. Recientemente, se han propuesto divisiones funcionales flexibles para proporcionar la posibilidad de colocar dinámicamente las funcionalidades de CUs y DUs en la red de *fronthaul*¹, para cumplir con los requisitos de servicio. Sin embargo, esta evolución hacia la C-RAN ha aumentado la complejidad de la red *fronthaul*, ya que es necesario, entre otros asuntos, desarrollar técnicas para definir dónde ubicar las funcionalidades de CUs y DUs en los nodos de la red *fronthaul*, explotando los recursos de la red de manera eficiente, considerando las limitaciones de la red y cumpliendo los requisitos para soportar los servicios desplegados en la C-RAN. En esta tesis, nos enfocamos en abordar conjuntamente el problema de determinar las mejores rutas para el tráfico de red *fronthaul* y la división óptima

¹Red que interconecta las DUs y CUs en el borde de una red celular.

de funcionalidades entre CUs y DUs, para soportar los requisitos de calidad de los operadores de redes móviles virtuales. Se propone un modelo de sistema novedoso que reúne todos los elementos y características del problema, y utilizamos este modelo para formular un problema de optimización que se caracteriza por el tamaño exponencial de su espacio de búsqueda. Proponemos dos enfoques heurísticos para abordar este problema: (1) un algoritmo avaro, y (2) un algoritmo evolutivo que emplea una rutina de inicialización especializada que permite comenzar el proceso de búsqueda a partir de soluciones factibles de calidad razonable. Realizamos extensos experimentos para investigar la efectividad de los métodos propuestos. Estos experimentos consideran una colección diversa de escenarios de prueba, y evalúan la capacidad de nuestros algoritmos para producir soluciones de alta calidad, así como también aspectos relacionados con su comportamiento y eficiencia computacional. También realizamos comparaciones con respecto a la solución óptima (obtenida mediante búsqueda exhaustiva) empleando un subconjunto de instancias pequeñas. Nuestros resultados indican que las técnicas propuestas representan compromisos apropiados entre la calidad de la solución y el tiempo de ejecución, y pueden servir a objetivos complementarios: la calidad de los resultados obtenidos por nuestro método evolutivo es mejor, pero a costa de tiempos de ejecución más largos. Por otro lado, nuestro algoritmo avaro ofrece un desempeño razonablemente apropiado, con un tiempo de ejecución notablemente menor. Nuestros experimentos demuestran que es posible producir resultados cercanos al valor óptimo para el problema a través de soluciones algorítmicas computacionalmente eficientes.

Joint Selection of Route and Functional Split for C-RAN Architectures

by

Cristian Camilo Erazo Agredo

Cinvestav Tamaulipas

Center for Research and Advanced Studies of the National Polytechnic Institute, 2020

Dr. Mario Garza Fabre, Co-advisor

Dr. Javier Rubio Loyola, Co-advisor

The fifth-generation (5G) cellular networks have been promised to cope with the highly increasing demands for high-speed network traffic and high-quality services. The part of the cellular networks that needs to be modified to handle such growth is called the Radio Access Network (RAN). The *Cloud Radio Access Network* (C-RAN) is a paradigm that has been proposed with the intention to virtualize the functionalities of the base stations to increase the flexibility of the mobile network, by grouping processing units into pools of resources. In C-RAN, the functionalities of the base stations are separated, giving rise to the central units (CUs) that are in charge of the baseband processing and to the distributed units (DUs) that are in charge of the radio frequency functions. Recently, flexible functional divisions have been proposed to provide the possibility of dynamically placing the functionalities of CUs and DUs in the fronthaul network², in order to meet service requirements. Nevertheless, this architectural evolution has increased the complexity of the fronthaul network as it is necessary, among other issues, to develop techniques to define where to place the functionalities of CUs and DUs in the nodes of the fronthaul network, exploiting the network resources efficiently, considering the constraints of the network, and meeting the requirements to support the services deployed on the C-RAN. In this thesis, we focus on jointly addressing the problem of determining the best routes for network traffic and the optimal division of functionalities between central and distributed units, to support the quality requirements of virtual mobile network operators. We build

²Network that connects the DUs and CUs at the "edge" of a cellular network

a novel system model that brings together all the elements and features of the problem, and use this model to formulate an optimization problem which is characterized by the exponential size of its search space. We propose two heuristic approaches to address this problem: (1) a greedy algorithm, and (2) an evolutionary algorithm which uses a specialized initialization routine to seed the search process with a set of feasible solutions of reasonable quality. We conducted extensive experiments to investigate the suitability of the proposed methods. These experiments consider a diverse collection of test scenarios, and evaluate the capability of our algorithms to produce high-quality solutions as well as their behavior and computational efficiency. We also performed comparisons with the optimal solution (obtained by exhaustive search) using a subset of small-size instances. Our results indicate that the proposed techniques represent appropriate trade-offs between solution quality and execution time, and can serve complementary goals: the quality of the results yielded by our evolutionary method is better, but at the cost of longer execution times. On the other hand, our greedy algorithm offers a reasonably appropriate performance, with an execution time that is notably lower. Our experiments demonstrate that it is possible to produce near-optimal results to the above complex problem through computationally efficient algorithmic solutions.

Nomenclatura

5G	Quinta generación de tecnologías de red celular.
<i>AA</i>	Algoritmo avaro.
AE	Algoritmo evolutivo.
<i>AE^{rnd}</i>	Algoritmo evolutivo con inicialización aleatoria.
<i>AE^{gre}</i>	Algoritmo evolutivo con inicialización avara.
BBU	Unidad de procesamiento de banda base.
BS	Estación base.
C-RAN	Cloud Radio Access Network.
CU	Unidad central.
D-RAN	Distributed Radio Access Network.
DU	Unidad distribuida.
GA	Algoritmo genético.
IC	Inteligencia computacional.
InP	Proveedor de infraestructura.
MAC	Control de acceso al medio.
NV	Virtualización de redes.
PDCP	Protocolo de convergencia de paquetes de datos.
PHY	Capa física.
QoS	Calidad de servicio.
RH	Cabezal de radio.
RLC	Control de enlace de radio.
RRH	Cabezal de radio remoto.
RRC	Control de recursos de radio.
RU	Unidad de radio.
SCRDF	Selección conjunta de rutas y divisiones funcionales.
VLiM	Mapeo de enlaces virtuales.
vMNOs	Operadores de redes móviles virtuales.
VNE	Incrustación de redes virtuales.
VNoM	Mapeo de nodos virtuales.
WNC	Wireless Network Cloud.

1

Introducción

En este capítulo se brinda una introducción a los conceptos y elementos esenciales para el entendimiento del presente trabajo de tesis. En la parte inicial de este capítulo, se presentan conceptos relacionados con la nueva generación de redes celulares, en especial de la arquitectura C-RAN. Posteriormente, se presenta la motivación de la tesis, el planteamiento del problema y la pregunta de investigación. Seguido de esto, se presentan los objetivos generales y particulares, así como la metodología y las contribuciones o aportaciones principales obtenidas en el desarrollo del presente trabajo de tesis.

1.1 Nueva generación de redes celulares

La demanda de redes y dispositivos móviles sigue en aumento. De acuerdo al reporte realizado por Cisco para el periodo 2017-2022 [1], el tráfico de datos móviles creció un 71 % en 2017 y se espera un aumento a una tasa de crecimiento anual compuesta del 46 % del tráfico de datos móviles a nivel mundial para el año 2022. De igual manera, el número de dispositivos móviles proyectados para el

año 2022 es de 1.5 por persona; esto significa que para el año 2022 habría más de 12 mil millones de dispositivos, generando un tráfico mensual de más de 77 exabytes (EB). Las redes celulares de quinta generación (5G) han sido propuestas para afrontar a las crecientes demandas de tráfico de red de alta velocidad y servicios de alta calidad. La parte de las redes celulares que deben modificarse para manejar dicho crecimiento se llaman redes de acceso por radio (RAN). Estas redes deben de modificarse debido a que son las encargadas de proveer la conexión entre los usuarios y el núcleo de la red. Las redes celulares de quinta generación se caracterizan, entre otras características, por proveer mayor versatilidad y adaptabilidad de las tecnologías de acceso por radio [2, 3].

Para proveer mayor versatilidad y adaptabilidad, *Cloud Radio Access Network* (C-RAN) es considerada como la propuesta más importante para las redes actuales y futuras [2, 3, 4, 5]. C-RAN es un paradigma que ha sido propuesto en [6], el cual fue adaptado de *Wireless Network Cloud* (WNC) [7], con la intención de virtualizar las funcionalidades de las estaciones base para aumentar la flexibilidad de la red móvil, al agrupar las unidades de procesamiento en grupos de recursos. Dada la importancia que tiene C-RAN en este trabajo de tesis, a continuación se dan algunos detalles de este importante paradigma.

1.1.1 Cloud Radio Access Network

En 2010 se presenta un diseño de red móvil denominado WNC [7], el cual permitiría comunicar diferentes estaciones base (BS, por sus siglas en inglés) al desacoplar las antenas y los cabezales de radio (RH, por sus siglas en inglés) de las estaciones base, así como ubicar los equipos de las BSs en centros de datos provistos con mayores capacidades de cómputo en los que se implementaban las unidades de banda base (BBU, por sus siglas en inglés). Estos centros de datos implementan el procesamiento de banda base por medio de software, permitiendo abordar los problemas de compatibilidad con múltiples estándares sin requerir de actualizaciones de hardware. Las antenas y los RHs de las BSs son desacoplados por medio del uso de cabezales de radio remotos (RRH, por sus siglas en inglés), permitiendo la instalación de RRHs en torres remotas distribuidas. Ante la

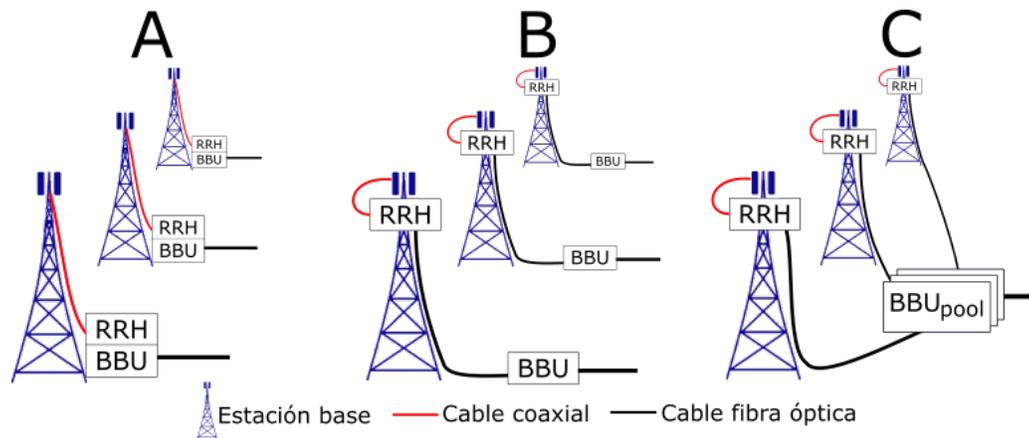


Figura 1.1: Escenarios posibles del concepto WNC. Los elementos referidos como RRH corresponden a las unidades de procesamiento de radio frecuencia, mientras que las unidades referidas como BBU corresponden a las unidades de procesamiento de banda base.

posibilidad de desacoplar las funcionalidades de la red de acceso, se plantearon tres escenarios para el concepto WNC, tal como se muestra en la Figura 1.1, estos escenarios se describen brevemente a continuación.

En el escenario **A**, las estaciones base constan de un bloque conformado por la unidad de procesamiento de banda base y la unidad de procesamiento de radio frecuencia, la cual se conecta por medio de cable coaxial a las antenas de transmisión. El procesamiento de señal se realiza de manera local a unos pocos metros de la antena. En el escenario **B**, las estaciones base se dividen en una unidad de procesamiento de radio frecuencia y una unidad de procesamiento de banda base, las cuales están conectadas por medio de fibra óptica. Este diseño de infraestructura se conoce como *Distributed-RAN* (D-RAN). Finalmente, en el escenario **C**, se presenta el diseño de red móvil WNC, donde el procesamiento de banda base se realiza en una piscina de recursos (BBU_{pool}).

Posteriormente, en [6] el concepto de WNC fue detallado y renombrado como C-RAN, y se planteó como el diseño de una arquitectura centralizada para las redes de acceso que se caracteriza por la disminución de los costos de operación de la red (aunque los costos de despliegue aumentarían debido a la necesidad de aumentar la capacidad en la red de acceso), el aumento del ancho de

banda disponible, la adaptabilidad al tráfico no uniforme y la división de las funcionalidades de la BS, compuesta por la BBU y la RRH. Estas divisiones establecen dos escenarios posibles:

- **Centralización completa:** las RRHs proveen únicamente de funciones de radio, mientras que las BBUs proveen todas las funcionalidades de banda base.
- **Centralización parcial:** las RRHs proveen funciones de radio y algunas funcionalidades de procesamiento de banda base, y las BBUs proveen las funcionalidades restantes.

El escenario completamente centralizado tiene la ventaja de ser escalable a costa de la cantidad de elementos que se deben configurar y los requerimientos que se deben satisfacer. También permite compartir el procesamiento de las funciones de red, mejorar el uso de los recursos y reducir los costos de operación de la red debido a que al estar agrupadas las BBUs, se reducen los costos de renta, de refrigeración, entre otros. Este escenario requiere más ancho de banda para las conexiones entre las BBUs y las RRHs. A estas conexiones entre las BBUs y las RRHs se les conoce como *fronthaul*; de manera similar, a las conexiones entre las BBUs y el *backbone* o núcleo de la red se le conoce como *backhaul*.¹ Las conexiones entre las BBUs y las RRHs para este escenario, son realizadas (típicamente) con fibra óptica para poder satisfacer los requerimientos de ancho de banda y latencia del *fronthaul*. Por otro lado, el escenario parcialmente centralizado tiene la ventaja de requerir menos ancho de banda en el *fronthaul* dado que parte del procesamiento de banda base se realiza en la RRH, pero pierde las ventajas provistas por la centralización completa (por ejemplo, procesamiento compartido). C-RAN requiere un alto despliegue de enlaces de fibra óptica, lo cual podría no ser factible en algunos escenarios. Para afrontar esto, se ha propuesto el uso de diferentes divisiones funcionales, conocidas en la literatura como *Functional Splits* [4, 8, 9]. La BBU es dividida en una unidad distribuida (DU, por sus siglas en inglés), la cual es ubicada cerca de la RRH, ahora denominada unidad de radio (RU, por sus siglas en inglés), y una unidad central (CU, por sus siglas en

¹El *backbone* es una línea o conjunto de líneas a las que las redes de área local se conectan para tener conexión de red de área amplia; en este contexto, para conectarse a Internet.

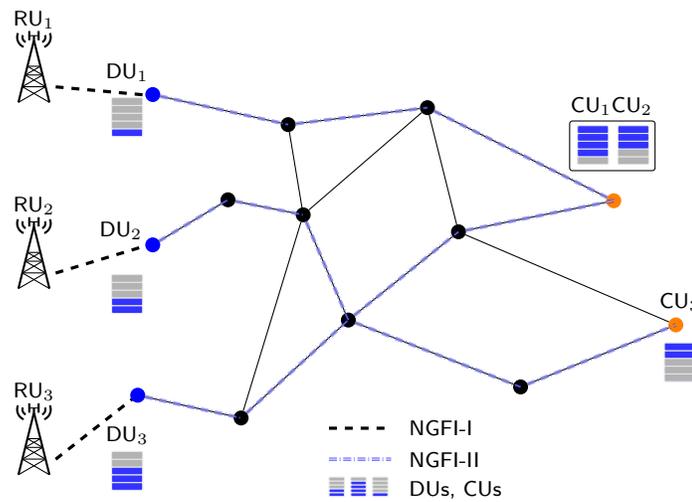


Figura 1.2: Ejemplo de una RAN virtualizada con NGFI.

inglés), la cual se beneficiaría del procesamiento centralizado y de los altos poderes de procesamiento del BBU_{pool} .

Debido a los requerimientos de ancho de banda y latencia, actualmente se está estudiando la idoneidad de enlaces de fibra óptica, enlace cableado y enlace de radio en las conexiones de *fronthaul* y *backhaul* [10, 11, 12, 13], con lo cual es necesario estudiar esquemas de gestión de recursos para estos enlaces. Por otra parte, una interfaz de comunicación denominada *Next Generation Fronthaul Interface* (NGFI) ha sido propuesta en [9]. De acuerdo con [14], el segmento convencional de *fronthaul* en NGFI se divide en dos interfaces, NGFI-I y NGFI-II. Por un lado, NGFI-I conecta DUs con las RUs. Por otro lado, la interfaz NGFI-II interconecta DUs con CUs, la conexión entre las DUs y las CUs se conoce como *midhaul*. Como ejemplo, la Figura 1.2 representa una red que posee tres estaciones base virtualizadas, cuyos componentes están conectados a través de una red NGFI. Como se puede ver, la NGFI-I conecta las RU y las DU, que a su vez implementan algunas de las funciones de la estación base. Para este caso, las funcionalidades restantes se implementarían en la CU con la que se conectara la DU en cuestión. La NGFI-II es una red basada en paquetes que conecta DUs y las CUs que implementan las funciones restantes, de acuerdo con la división funcional seleccionada. Teniendo en cuenta la naturaleza no determinista de una red basada en paquetes, los nodos intermedios en

el NGFI pueden necesitar emplear técnicas de almacenamiento tipo búfer, lo que podría aumentar el retraso y pérdidas de paquetes, poniendo en riesgo la calidad del servicio (QoS). Ahora bien, la cantidad y el tipo de funcionalidades que se implementen en las DUs y las correspondientes CUs en interconexión, depende de la división funcional que se elija para el despliegue.

1.1.2 Virtualización de redes

La tecnología de virtualización de redes (NV por sus siglas en inglés) permite que múltiples redes virtuales (VN por sus siglas en inglés) se implementen en una infraestructura de red física (red de sustrato) [15, 16]. El desarrollo de la tecnología de NV presenta como principal problema el incorporar las VNs en una red de sustrato de forma que se aprovechen al máximo o de manera óptima los recursos de la infraestructura de la red física. Existen diferentes enfoques para llevar a cabo dicha optimización, es decir se puede realizar tomando en cuenta diversos objetivos, tales como garantizar la QoS, optimizar el uso de la energía eléctrica al utilizar los recursos físicos, incrustar el mayor número de redes, etc. Al problema anteriormente mencionado se le conoce como incrustación de redes virtuales (VNE, del inglés Virtual Network Embedding), este problema es descrito con mayor detalle en la Sección 2.1.2 del Capítulo 2. En este trabajo de tesis, los requisitos de QoS de los Operadores de Redes Móviles Virtuales (vMNOs por sus siglas en inglés) son representadas como VNs y el poseedor de la red física o red de sustrato, se le conoce como proveedor de infraestructura (InP por sus siglas en inglés).

1.2 Motivación de la tesis

En el contexto de la arquitectura C-RAN descrito anteriormente, las funciones de las estaciones base pueden ser procesadas o bien, en la DU, o bien en la CU, dependiendo de los requerimientos de transporte y el nivel de centralización que se requiera para algún despliegue. De modo que, para un despliegue de una red celular de acceso en el contexto de la arquitectura C-RAN, es necesario

determinar qué funciones se implementan en la DU y qué funciones se implementan en la CU, es decir, es necesario determinar las divisiones funcionales de toda la red, la cual está integrada por un número determinado de estaciones base. Ahora bien, lo anterior tiene varias problemáticas fundamentales. Primeramente, la división de funciones de red está restringida a un número de divisiones funcionales. En la literatura se han propuesto diez divisiones funcionales [4] (presentadas con mayor detalle en el Capítulo 2) que podrían ser empleadas en la red *midhaul* [5]. También, dados los requisitos de ancho de banda y latencia que se deben cumplir para un despliegue determinado por las necesidades de un operador de red móvil, cada división funcional afecta a la cantidad de CUs que se pueden ubicar cerca de las DUs.

En este sentido, dos de los principales retos a resolver por los administradores de red son: (1) determinar la ubicación de cada DU y CU que componen la estación base, así como la distribución óptima de funcionalidades entre ellos, de acuerdo a dicha ubicación; y (2) determinar las mejores rutas para el tráfico proveniente de cada una de las DUs. Cada uno de estos retos es desafiante debido a la cantidad de elementos y posibles configuraciones que se deben considerar. Ante este escenario, surge la necesidad de contar con mecanismos que permitan automatizar la toma de decisiones para que el InP pueda soportar las peticiones de los vMNOs con la QoS requerida.

Aunado a la naturaleza desafiante que individualmente presentan los dos retos antes mencionados, la motivación principal de este trabajo de tesis es abordar estos dos retos de manera conjunta. Como se detalla en la Sección 1.3, esto se debe a que la elección de determinadas rutas impone limitaciones en cuanto a las posibles funcionalidades que se pueden establecer en una DU. Del mismo modo, una determinada asignación de funcionalidades implica demandas específicas de recursos de red que deben considerarse durante la planificación de rutas. Por lo tanto, este es un problema computacionalmente complejo, que demanda una solución integral que pueda producir soluciones satisfactorias de manera eficiente.

1.3 Planteamiento del problema

El problema de Selección Conjunta de Rutas y Divisiones Funcionales (SCRDF) para arquitecturas C-RAN se compone de los siguientes dos subproblemas: (1) el problema de determinar una asignación DU-CU, así como una división óptima de funcionalidades de acuerdo a esta asignación; y (2) el problema de seleccionar las mejores rutas para el tráfico proveniente de cada una de las DUs. El primer subproblema consiste en asignar, a cada una de las DUs, una división funcional que permita aprovechar los recursos de la red, y posteriormente asociar esta DU con la CU que implementará las funciones virtualizadas restantes. El segundo subproblema tiene como objetivo establecer la mejor ruta para conectar cada DU con su respectiva CU, que luego procesará y reenviará el tráfico al núcleo de la red. Las rutas deben cumplir con los requisitos de retardo y ancho de banda de las divisiones funcionales presentadas en la Tabla 5.3, y los requerimientos de QoS de los vMNOs.

Aún cuando se consideran de manera individual, cada uno de los subproblemas antes descritos representa un reto computacional significativo dado que se ha demostrado que cada uno de éstos son NP-difícil [17]. No obstante, debido a su interrelación con el ecosistema global de una red de acceso, estos retos no deben ser abordados individualmente. La asignación óptima de una división funcional depende de la capacidad y latencia de la ruta que conecta la DU con la CU en la red *midhaul*. Estos valores son desconocidos hasta que todas las rutas hayan sido calculadas. Por otra parte, la identificación de la mejor ruta para el flujo de tráfico proveniente de una DU requiere del conocimiento de la división funcional elegida, ya que ésta determina el ancho de banda y latencia que se deben garantizar.

Finalmente, se ha demostrado también que abordar estos problemas de manera jerárquica es decir, resolviendo primero uno de los problemas y utilizando la solución obtenida como punto de partida para resolver el segundo problema, resultaría en un enfoque que produce soluciones sub-óptimas o soluciones inviables [17]. En este documento de tesis, al problema anterior lo definimos como

el problema de Selección Conjunta de Rutas y Divisiones Funcionales (SCRDF) para arquitecturas C-RAN.

1.4 Pregunta de investigación e hipótesis

A partir de la descripción del problema de Selección Conjunta de Rutas y Divisiones Funcionales (SCRDF) para arquitecturas C-RAN, se hace evidente la necesidad de contar con enfoques computacionalmente eficientes, que permitan abordar de manera integral los retos que este problema de optimización involucra. Por otro lado, la literatura reporta numerosas evidencias sobre la aplicación exitosa de técnicas heurísticas de optimización y técnicas de inteligencia computacional (IC) en la resolución de problemas de optimización desafiantes, problemas que presentan características diversas y que se originan en distintas áreas. Este trabajo de tesis de maestría, por lo tanto, busca responder a la siguiente pregunta de investigación:

¿Es posible abordar el problema SCRDF de forma integral, y emplear técnicas heurísticas de optimización y de inteligencia computacional para definir compromisos entre la calidad de las soluciones y el tiempo de ejecución requerido para producirlas?

La hipótesis subyacente de este trabajo de tesis es que el planteamiento integral del problema SCRDF, así como la utilización de técnicas heurísticas de optimización e inteligencia computacional (basadas en algoritmos evolutivos, más específicamente), permitirá identificar de manera eficiente soluciones adecuadas de asignación de rutas y divisiones funcionales en arquitecturas C-RAN.

Como se presenta más adelante en la Sección 2.2, los algoritmos heurísticos se han utilizado ampliamente en problemas relacionados. Estos algoritmos se caracterizan por sus bajos tiempos de ejecución y su capacidad para encontrar soluciones satisfactorias. Por otra parte, en la literatura se pueden encontrar numerosas aplicaciones de los algoritmos evolutivos (AEs), que han reportado resultados competitivos al enfrentarse a problemas de optimización complejos del mundo real [18].

Dada la complejidad que presenta el problema de estudio de este trabajo de tesis, se considera que los AEs representan un enfoque prometedor que puede conducir al diseño de una solución algorítmica adecuada para abordar este problema.

1.5 Objetivos generales y particulares

Objetivo general

Abordar el problema de Selección Conjunta de Rutas y Divisiones Funcionales para redes *midhaul* en arquitecturas C-RAN de manera integral, y demostrar la viabilidad de obtener soluciones de asignación eficiente de los recursos de la red mediante la distribución adecuada de funciones.

Objetivos particulares

1. Definir un modelo del problema de Selección Conjunta de Rutas y Divisiones Funcionales para arquitecturas C-RAN, que permita formalizar la función objetivo y las restricciones del problema.
2. Diseñar e implementar una técnica heurística de optimización y una técnica de inteligencia computacional, basada en algoritmos evolutivos, que permitan obtener de manera eficiente soluciones satisfactorias al problema de Selección Conjunta de Rutas y Divisiones Funcionales para arquitecturas C-RAN.
3. Validar las técnicas propuestas y demostrar la viabilidad de proveer soluciones computacionales que permitan definir compromisos entre eficiencia (tiempos de ejecución) y la capacidad de identificar soluciones (configuraciones de recursos) de alta calidad.

1.6 Metodología

Esta sección describe la metodología que se ha usado en este trabajo de tesis para alcanzar el objetivo general y los objetivos particulares planteados en la Sección 1.5.

- **Etapa 1, Estado del arte.** La primer etapa se enfoca en una revisión de la literatura y en la obtención de los elementos que permiten caracterizar el problema de SCRDF para arquitecturas C-RAN.
- **Etapa 2, Modelado del problema.** En la segunda etapa de este trabajo de tesis se investigaron los elementos de una red móvil, haciendo énfasis en aquellos que determinan o caracterizan la división de las funciones de las estaciones base. Esto permite determinar qué factores influyen en la elección de divisiones funcionales particulares. Se definió un planteamiento formal que permite modelar y abstraer los distintos componentes del problema, esto involucra la formulación matemática de los criterios específicos a utilizarse como función objetivo, así como la definición de las restricciones del problema.
- **Etapa 3, Diseño e implementación.** La tercera etapa plantea el diseño e implementación de nuevos algoritmos para abordar el problema de estudio de este trabajo de tesis. Inicialmente se consideraron heurísticas simples, específicamente un algoritmo avaro. Este algoritmo avaro sirve también como punto de referencia y comparación, y fue considerado para su integración como parte del diseño de un algoritmo más sofisticado. Finalmente, se desarrolló un nuevo algoritmo, basado en técnicas de IC, más específicamente AEs, que es capaz de producir soluciones de mayor calidad, pero aumentando el tiempo de ejecución.
- **Etapa 4, Experimentación y evaluación.** La etapa final de este trabajo de tesis contempló el diseño y ejecución de los experimentos necesarios para evaluar la efectividad e idoneidad de los enfoques resultantes de la etapa anterior. Los resultados preliminares obtenidos durante

la parte inicial de esta etapa permitieron refinar el diseño de los algoritmos propuestos. La experimentación de este trabajo de tesis requirió el diseño de un conjunto de instancias de prueba que consideraron diferentes escenarios, los cuales permitieron comparar los algoritmos propuestos.

1.7 Contribuciones

Con el desarrollo de este trabajo de tesis se logran las siguientes aportaciones principales:

- Un modelo novedoso que reúne los elementos y características para abordar el problema de selección conjunta de rutas y divisiones funcionales para arquitecturas C-RAN y manejar su complejidad. El problema se modela como un problema de incrustación de redes virtuales, que admite múltiples solicitudes de vMNOs, heterogeneidad en el uso de recursos, diversidad de asignaciones de DUs y CUs, selección de rutas con reconocimiento de restricciones y selección de divisiones funcionales.
- La formulación del problema de optimización que tiene como objetivo evaluar la calidad de las soluciones potenciales en función del grado de centralización dado por las divisiones funcionales, así como del número y tipo de enlaces de red usados.
- Dos enfoques algorítmicos para abordar el problema antes mencionado. Debido a la complejidad del problema, los algoritmos exactos pueden encontrar soluciones óptimas, pero su tiempo de ejecución aumenta drásticamente con instancias prácticas, es decir, instancias que consisten en redes de sustrato y redes virtuales de tamaño grande. Por lo tanto, los algoritmos exactos se pueden usar de manera eficiente solo para casos pequeños del problema. En este trabajo de tesis se contribuye con dos algoritmos heurísticos: un enfoque avaro diseñado para optimizar el tiempo de ejecución, y un algoritmo evolutivo personalizado para proporcionar soluciones más prometedoras.

1.8 Organización del documento

El presente documento de tesis está organizado de la siguiente manera:

- **Capítulo 2. Marco teórico y estado del arte:** en este capítulo se describen los principales conceptos que permiten entender el contexto del problema y la solución propuesta. También se presenta la revisión de la literatura, la cual presenta los trabajos relacionados al problema de SCRDF para arquitecturas C-RAN.
- **Capítulo 3. Modelado formal del problema:** en este capítulo se detalla el modelo que se propone en este trabajo de tesis para formalizar y abstraer los principales componentes del problema.
- **Capítulo 4. Métodos de solución propuestos:** en este capítulo se describen los diseños de los algoritmos propuestos que permiten encontrar soluciones al problema de SCRDF para arquitecturas C-RAN.
- **Capítulo 5. Pruebas y resultados:** en este capítulo se describe el diseño experimental realizado y presenta los resultados obtenidos al realizar la experimentación. La experimentación realizada tiene como objetivo principal poner a prueba la hipótesis planteada en este trabajo de tesis.
- **Capítulo 6. Conclusiones y trabajo futuro:** en este capítulo se describen las conclusiones obtenidas del análisis de los resultados de este trabajo de tesis. También se presentan los posibles trabajos o áreas de oportunidad que derivan del trabajo realizado.

2

Marco teórico y estado del arte

Este capítulo presenta antecedentes y conceptos esenciales que son necesarios para el entendimiento y correcta interpretación del resto de los capítulos de este documento de tesis (Sección 2.1). Asimismo, este capítulo incluye una revisión y discusión de trabajos representativos de la literatura especializada, enfocándose particularmente en aquellos trabajos que se relacionan estrechamente con el problema abordado en el presente trabajo de tesis (Sección 2.2).

2.1 Marco teórico

En esta sección se detallan los conceptos que permiten al lector familiarizarse con el problema de estudio. Primero se presentan los conceptos relacionados a las divisiones funcionales. Seguido de esto, se presentan conceptos de optimización. Finalmente, se presentan conceptos del problema de incrustación de redes virtuales.

2.1.1 Divisiones funcionales

En esta sección se describe el concepto de división funcional. Para ello, primeramente presentamos las funciones de una estación base que son objeto de división y posteriormente describimos las posibles combinaciones de divisiones funcionales.

2.1.1.1. Funciones de la estación base

La idea de dividir una estación base en un conjunto de funciones obedece a que, dichas funciones pueden ser procesadas o bien, en la DU, o bien en la CU, dependiendo de los requerimientos de transporte y el nivel de centralización que se requiera para algún despliegue. De acuerdo con [4, 19, 5], las funciones de red son:

- **Radio Resource Control (RRC):** la subcapa RRC se encarga del control de los recursos de radio sobre la interfaz de radio, del intercambio de mensajes de señalización que son esenciales para gestionar la conexión entre la estación base y el equipo de usuario, y de la gestión de portadores de radio.
- **Packet Data Convergence Protocol (PDCP):** la subcapa PDCP se encarga de realizar funciones de seguridad y de la compresión del encabezado. La función de seguridad proporciona cifrado para evitar que partes no autorizadas miren los datos y proporciona protección de integridad para maximizar la eficiencia de los recursos de radio al evitar la transmisión de información redundante a través de la interfaz de radio.
- **Radio Link Control (RLC):** la subcapa RLC provee funciones de segmentación y reensamblaje, y transmisiones libres de errores. Para esto, emplea mecanismos de corrección de errores mediante *Automatic Repeat ReQuest* (ARQ) y de entrega ordenada de unidades de datos de protocolo (PDU).

- **Medium Access Control (MAC):** la subcapa MAC se encarga de controlar el acceso al canal de radio. La capa MAC soporta funciones de planificación para alojar recursos de radio a cada equipo de usuario. Para lograr esto, la capa MAC realiza informes de estado de búfer para indicar la cantidad de datos en cada canal lógico, multiplexación y de-multiplexación para combinar y desensamblar PDUs de capa superior o inferior, y *Hybrid Automatic Repeat Request* (HARQ) para transferir PDUs sobre la interfaz de radio.
- **Physical (PHY):** Es la capa encargada de realizar la transmisión a través de la interfaz de radio. Alberga funciones de codificación de canal, modulación, procesado asociado a las técnicas de múltiples antenas de transmisión/recepción, y mapeo de la señal a los recursos físicos frecuencia-tiempo apropiados.

2.1.1.2. Divisiones de las funcionalidades

Las funciones descritas anteriormente pueden ser procesadas o bien, en la DU, o bien en la CU, dependiendo de los requerimientos de transporte y el nivel de centralización necesarios para algún despliegue. De tal modo que, un aspecto relevante para un despliegue de una red celular de acceso en el contexto de la arquitectura C-RAN, es determinar qué funciones se implementan en la DU y qué funciones se implementan en la CU, a ésta definición se refiere el término división funcional. En la literatura se han propuesto diez divisiones funcionales [4, 20] que determinan las tecnologías empleadas en el *midhaul* [5]. Cada división funcional afecta a la cantidad de CUs de la red que se pueden ubicar cerca de las DUs, esto debido a las restricciones de ancho de banda y latencia que se deben cumplir. Las divisiones funcionales propuestas son las siguientes:

- **Opción 1 (RRC/PDCP):** Las funciones de la subcapa RRC se encuentran en la CU, y las funciones de las subcapas PDCP, RLC, MAC, PHY y RF están en la DU. Como beneficios se cuenta con una tasa de bits baja y dependiente de la carga de las celdas en el enlace *fronthaul*,

los datos del usuario están cerca del punto de transmisión y baja latencia. Como desventajas se tiene la complejidad de la DU, pocas funciones aprovechan el procesamiento compartido.

- **Opción 2 (PDCP/RLC):** Las funciones de las subcapas RRC y PDCP están en la CU, mientras que las funciones de las subcapas RLC, MAC, PHY y RF están en la DU. Como ventajas se cuentan con aspectos en tiempo real ubicados en la DU, esto hace que los requisitos de enlace para esta división sean bajos, otra ventaja es el gran potencial para la coordinación de los procedimientos de movilidad. Como desventaja esta división funcional tiene un potencial limitado para la coordinación entre múltiples DUs.
- **Opción 3 (High RLC/Low RLC):** La subcapa RLC es dividida en subcapa alta RLC (compuesta de ARQ y otras funciones RLC) y subcapa baja RLC (compuesta de funciones de segmentación). Las funciones de las subcapas baja RLC, MAC, PHY y RF son ubicadas en la DU, mientras que las funciones de las subcapas alta RLC y PDCP son implementadas en la CU. Esta división tiene como ventajas la robustez sobre condiciones no ideales de transmisión, bajos requerimientos de ancho de banda en el *fronthaul*, posible reducción en el procesamiento y requerimientos de almacenamiento temporal en la DU. Como desventaja se tiene que la latencia se ve afectada debido a la ubicación del ARQ.
- **Opción 4 (RLC-MAC):** Las funciones de las subcapas MAC, PHY y RF están en la DU, y las funciones de las subcapas RRC, PDCP y RLC están en la CU. Esta división tiene como ventajas los beneficios de uso compartido de recursos para el almacenamiento y la utilización del procesador. Como desventaja, esta opción se considera poco práctica, dada la estrecha relación entre las subcapas RLC y MAC.
- **Opción 5 (intra MAC):** La subcapa MAC es dividida en subcapa baja MAC y subcapa alta MAC. Las funciones de las subcapas RF, PHY y baja MAC se encuentran en la DU. Las funciones de las subcapas alta MAC, RLC y PDCP están en la CU. Esta división tiene como

ventajas que la DU solo tiene las funciones que requieren comunicación en tiempo real, gestión eficiente de interferencias en múltiples celdas y tecnologías de planificación mejoradas. Como desventaja se tiene que la interfaz entre la CU y la DU es más compleja.

- **Opción 6 (MAC-PHY):** Las funcionalidades de las subcapas PHY y RF están en la DU, las demás funcionalidades están en la CU. Esta división tiene como ventajas que la transmisión conjunta es posible debido a que las funciones MAC están en la CU. Esto también permite la planificación centralizada y la agrupación de recursos para subcapas MAC y superiores. Como desventajas se tienen posibles problemas debido a retraso en el *fronthaul* que afecta las capacidades de las subtramas más cortas.
- **Opción 7 (intra PHY):** Parte de las funciones de subcapa física y RF están en la DU. Las funciones de las demás subcapas están en la CU. Tiene como ventajas que permite centralizar la agregación de tráfico desde los puntos de transmisión, puede facilitar la gestión de la carga de tráfico y el procesamiento conjunto (tanto de transmisión como de recepción) es posible con esta opción ya que las funciones MAC están en la CU. Como desventaja, esta división puede requerir interacciones de sincronización a nivel de subtrama entre parte de la capa PHY en la CU y parte de la capa PHY en la DU. En esta opción, tres sub-divisiones han sido propuestas. Estas sub-divisiones establecen la ubicación de las funciones de la subcapa física y son descritas a continuación:
 - **Opción 7-1:** Las funciones de transformada rápida de Fourier (FFT), la transformada rápida inversa de Fourier (iFFT) y de plano de control (CP) residen en la DU. El resto de las funciones de la subcapa PHY están en la CU.
 - **Opción 7-2:** Las funciones FFT, iFFT, CP, des-mapeo de recursos y funciones de pre-filtrado residen en la DU. El resto de las funciones de la subcapa PHY están en la CU.
 - **Opción 7-3:** Sólo la función del codificador reside en la CU. El resto de las funciones de la subcapa PHY residen en la DU.

- **Opción 8 (PHY-RF):** Las funcionalidades de RF están en la DU, las demás funcionalidades están en la CU. Esta opción corresponde a la centralización completa. Esta división tiene como ventajas los altos niveles de centralización y coordinación en toda la pila de protocolos, lo que puede permitir una gestión de recursos y un rendimiento de radio más eficiente. También permite la reutilización de los componentes de RF para servir a las subcapas PHY de diferentes tecnologías de acceso de radio (por ejemplo, GSM, 3G, LTE) y permite a los operadores compartir componentes de RF, lo que puede reducir los costos del sistema y del sitio [4]. Como desventajas se tienen los altos requisitos en el ancho de banda de *fronthaul* (lo que puede implicar un mayor consumo de recursos y costos) y altos requisitos en la latencia del *fronthaul*, lo que puede causar restricciones en las implementaciones de red con respecto a la topología de red y las opciones de transporte disponibles.

En la Tabla 2.1 se presentan las diferentes opciones posibles para las divisiones funcionales, el requerimiento de ancho de banda (medido en mega bits por segundo) y el requerimiento de latencia (medido en microsegundos) que se debe garantizar para cada división funcional. Estos valores son obtenidos de [17]. La metodología para obtener estos elementos, se describe en [4, 5, 20]. En este trabajo, los valores considerados corresponden al caso de bajada, y la restricción de latencia para cada división funcional, corresponde al retardo de bajada.

En el contexto de la arquitectura C-RAN, la posibilidad de virtualización de las funciones de la red y en concreto, la posibilidad de implementar divisiones funcionales en los despliegues de las redes de acceso, ha abierto un gran debate en la academia e industria sobre el desarrollo de conceptos, métodos y tecnologías de soporte para la red *midhaul*. Sin duda, este escenario plantea nuevos retos para los administradores de red. Uno de los retos principales radica en determinar la asignación de cada DU con su respectiva CU y encontrar la disposición adecuada de las funciones de red, de manera que maximice el aprovechamiento del uso de recursos y maximizar la calidad de servicio de los servicios que se proveen.

Tabla 2.1: Requerimientos de ancho de banda y latencia para cada división funcional. Los valores son obtenidos de [17].

División funcional	Restricción de ancho de banda en Mb/s		Restricción de latencia en μs
	bajada	subida	
opción 1	151	48	30000
opción 2	151	48	30000
opción 3	151	48	6000
opción 4	151	49	6000
opción 5	152	49	250
opción 6	173	452	250
opción 7-1	933	903	250
opción 7-2	1075	922	250
opción 7-3	1966	1966	250
opción 8	2457.6	2457.6	250

2.1.2 Problema de incrustación de redes virtuales

El problema que se aborda en este trabajo de tesis tiene similitudes con el problema de incrustación de redes virtuales (VNE, por sus siglas en inglés). Por este motivo, se presenta una descripción del problema de VNE, así como los enfoques que abordan este problema clásico.

El problema de VNE consiste en optimizar el uso de los recursos de la red de sustrato, de manera que se logre incrustar un conjunto de redes virtuales [21, 22]. Esta optimización de recursos se puede efectuar empleando diferentes objetivos, que van desde la QoS, ganancia o costos, eficiencia energética, entre otros. Para incrustar una red virtual en la red física es necesario asignar tanto los nodos virtuales como los enlaces, por lo que VNE se puede dividir en dos subproblemas. El primer

subproblema consiste en el de mapeo de nodos virtuales (VNoM, por sus siglas en inglés), que consiste en la asignación de nodos virtuales a nodos físicos. El segundo subproblema consiste en el de mapeo de enlaces virtuales (VLiM, por sus siglas en inglés), donde los enlaces virtuales son asignados a rutas que conectan los nodos correspondientes de la red de sustrato a los que se encuentran asignados los nodos virtuales. Estos dos subproblemas deben ser considerados de forma conjunta, debido a que las asignaciones de los nodos afectarán las asignaciones de los enlaces y, de manera similar, las asignaciones de los enlaces afectarán las asignaciones de los nodos [22].

2.1.2.1. Enfoques del problema de VNE

Hay dos enfoques que determinan la manera de abordar el problema de VNE. Estos enfoques corresponden al dinámico y estático, como se describe a continuación:

- *Dinámico*: el enfoque dinámico, también denominado enfoque *online* (en línea), se caracteriza por introducir una componente temporal para las VNs. Más específicamente, las VNs involucran un tiempo de llegada y un tiempo de vida (o tiempo de uso de los recursos), por lo que se requiere de una supervisión constante de las VNs, así como de la actualización de los recursos disponibles en todos los nodos y enlaces de la red de sustrato. En este enfoque se requiere, por lo tanto, que los algoritmos obtengan soluciones rápidamente, antes de la llegada de otra VN. Esto hace que el tiempo de ejecución represente un factor crítico en este tipo de enfoques.
- *Estático*: el enfoque estático, también conocido como enfoque *offline* (fuera de línea), considera una cantidad fija de VNs, por lo que no se requiere de un proceso de supervisión de las VNs, ni se involucran tiempos de llegada y tiempos de vida como en el enfoque dinámico. En este sentido, se podría decir que el enfoque estático es menos complejo, pero aún así se ha demostrado que representa un problema NP-Difícil [23] (este problema puede plantearse como el problema del flujo no divisible [24]). En el enfoque estático los tiempos de ejecución no juegan un papel tan crítico como en el caso del enfoque dinámico. Tiempos de ejecución relativamente altos (pero

razonables) pueden aún considerarse aceptables, siempre que esto se traduzca en una mejora significativa en términos de calidad de solución.

El modelo del problema que se propone en el Capítulo 3 se relaciona estrechamente con el problema de VNE, donde una VN es el equivalente a una petición de recursos virtuales de un vMNO. Este trabajo de tesis plantea un modelo que involucra múltiples peticiones de recursos virtuales, considerando un enfoque de asignación de recursos estático (fuera de línea).

2.1.3 Optimización e inteligencia computacional

Como se describe a detalle en la Sección 1.3, este trabajo de tesis aborda un problema desafiante. Este problema se plantea como un problema de optimización combinatoria. Por lo tanto, esta sección proporciona una breve introducción de conceptos esenciales de optimización, así como de técnicas heurísticas de optimización que representan el enfoque de solución planteado en este trabajo.

2.1.3.1. Optimización

La optimización se refiere al proceso de búsqueda del mejor resultado posible para una función, dadas ciertas circunstancias [25]. Dado un conjunto finito de soluciones potenciales, denominado *espacio de búsqueda*, la tarea es identificar aquella solución (o soluciones) que nos permita satisfacer de la mejor manera (maximizando o minimizando) un cierto número de criterios de optimización, más formalmente conocidos como *funciones objetivo*, ajustándose a ciertas restricciones.

Formalmente, y sin pérdida de generalidad, un problema de optimización puede definirse de la siguiente manera:

$$\begin{aligned} \text{Minimizar} \quad & f(X), \\ \text{sujeto a} \quad & X \in \mathcal{F}, \end{aligned} \tag{2.1}$$

donde X es un vector de *variables de decisión* que representa una *solución candidata* (es decir, una configuración potencial de los parámetros para los que se requiere encontrar el valor más conveniente); \mathcal{F} denota el *espacio o región factible*, que corresponde al subconjunto de todas las soluciones factibles en el espacio de búsqueda \mathcal{X} , $\mathcal{F} \subset \mathcal{X}$; y $f : \mathcal{F} \rightarrow \mathbb{R}$ es la función objetivo a optimizarse. A la configuración específica de X para la cual f toma su valor mínimo se le conoce como *solución óptima*. La meta es, por lo tanto, encontrar una solución óptima $X^* \in \mathcal{F}$, tal que $f(X^*) = \min\{f(X) \mid X \in \mathcal{F}\}$.

2.1.3.2. Clasificación de los problemas de optimización

Existen diferentes formas de clasificar los problemas de optimización [25]. De acuerdo con el rango de valores que pueden tomar las variables de decisión, y por lo tanto con la finitud del espacio de búsqueda, los problemas de optimización pueden clasificarse como:

- *Optimización numérica*: todos aquellos problemas que son definidos sobre subespacios de espacios numéricos infinitos no contables (por ejemplo, vectores reales).
- *Optimización combinatoria*: todos aquellos problemas que son definidos sobre un espacio finito (o infinito numerable) discreto.

Por otro lado, los problemas de optimización también pueden clasificarse de acuerdo con el número de funciones objetivo que presentan:

- *Problema de optimización mono-objetivo*: como su nombre lo indica, un problema de optimización mono-objetivo involucra una única función objetivo y, por lo tanto, resolverlo implica encontrar una solución capaz de producir el mejor valor al ser evaluada utilizando esta función.
- *Problema de optimización multiobjetivo*: un problema de optimización multiobjetivo presenta múltiples criterios de optimización. A diferencia de la optimización mono-objetivo, para un

problema de optimización multiobjetivo usualmente no es posible encontrar una única solución que logre satisfacer todos sus objetivos simultáneamente. Dado el conflicto que generalmente se presenta entre las diferentes funciones objetivo, en optimización multiobjetivo la meta es encontrar un conjunto de *soluciones compromiso*.

Finalmente, es posible también clasificar a los problemas de optimización con base en la existencia de restricciones que determinan la factibilidad de las soluciones candidatas:

- *Problema de optimización sin restricciones*: no existen restricciones y, por lo tanto, todas las soluciones del espacio de búsqueda son consideradas factibles, es decir, $\mathcal{F} = \mathcal{X}$.
- *Problema de optimización con restricciones*: existe un conjunto de restricciones que las soluciones candidatas deben satisfacer para ser consideradas factibles. Usualmente existen soluciones que no cumplen con estas restricciones y, por lo tanto, $\mathcal{F} \subsetneq \mathcal{X}$.

Dadas las características del problema de optimización abordado en este trabajo de tesis, planteado formalmente en el Capítulo 3, este problema puede clasificarse como un problema de optimización combinatoria, mono-objetivo, con restricciones.

2.1.3.3. Técnicas de optimización

En la literatura se ha reportado una gran diversidad de métodos para resolver problemas de optimización [26, 27, 28]. Estos enfoques de solución pueden clasificarse de manera general como métodos exactos y métodos heurísticos. La principal característica de los métodos exactos es que garantizan la obtención de una solución óptima. Algunos ejemplos de este tipo de enfoque son los algoritmos de búsqueda exhaustiva, búsqueda con retroceso, y ramificación y acotamiento, que realizan una enumeración total o parcial del espacio de búsqueda para identificar una solución óptima. Dado el crecimiento exponencial del tamaño del espacio de búsqueda para muchos problemas de interés práctico, el uso de este tipo de algoritmos puede resultar prohibitivo y estar limitado a instancias relativamente pequeñas de estos problemas.

Por otro lado, los métodos de optimización heurísticos no garantizan la obtención de una solución óptima, pero usualmente son capaces de producir soluciones satisfactorias (generalmente soluciones sub-óptimas) a un costo computacional razonable. A través de la incorporación de conocimiento específico sobre el problema abordado, estos algoritmos realizan una búsqueda guiada, evitando realizar una exploración sistemática y exhaustiva del espacio de soluciones. Un ejemplo de estos métodos heurísticos son los algoritmos avaros (AA), también conocidos en la literatura como algoritmos voraces o golosos [29]. Los AA construyen iterativamente una solución para un problema determinado, explorando a cada paso un conjunto de posibles alternativas y seleccionando siempre aquella alternativa que parezca ser la más prometedora. La motivación principal de este tipo de técnicas es que una serie de decisiones localmente óptimas podrían permitir alcanzar el óptimo global del problema (aunque esto no suele ocurrir comúnmente en la práctica).

Finalmente, las metaheurísticas son algoritmos heurísticos que representan esquemas de solución generales, que pueden adaptarse a una gran variedad de problemas al otorgarle una definición específica a cada uno de sus componentes [30, 31]. Estos algoritmos emplean una o varias heurísticas para realizar una exploración más eficiente del espacio de soluciones. Las metaheurísticas han demostrado gran potencial al resolver problemas desafiantes que surgen en diversas áreas [18]. La Figura 2.1 ilustra la gran variedad de metaheurísticas que se han propuesto en la literatura especializada, incluyendo los algoritmos evolutivos que se tratan con mayor detalle en la Sección 2.1.3.4.

2.1.3.4. Algoritmos evolutivos

Los algoritmos evolutivos (AEs) son técnicas metaheurísticas del área de inteligencia computacional (IC), que toman inspiración de los principios y mecanismos del proceso natural de la evolución. Existen cuatro variantes principales de los AEs: estrategias evolutivas [32, 33, 34], programación evolutiva [35, 36], algoritmos genéticos [37, 38], y programación genética [39, 40]. Aunque históricamente estas cuatro variantes se propusieron y desarrollaron de manera independiente,

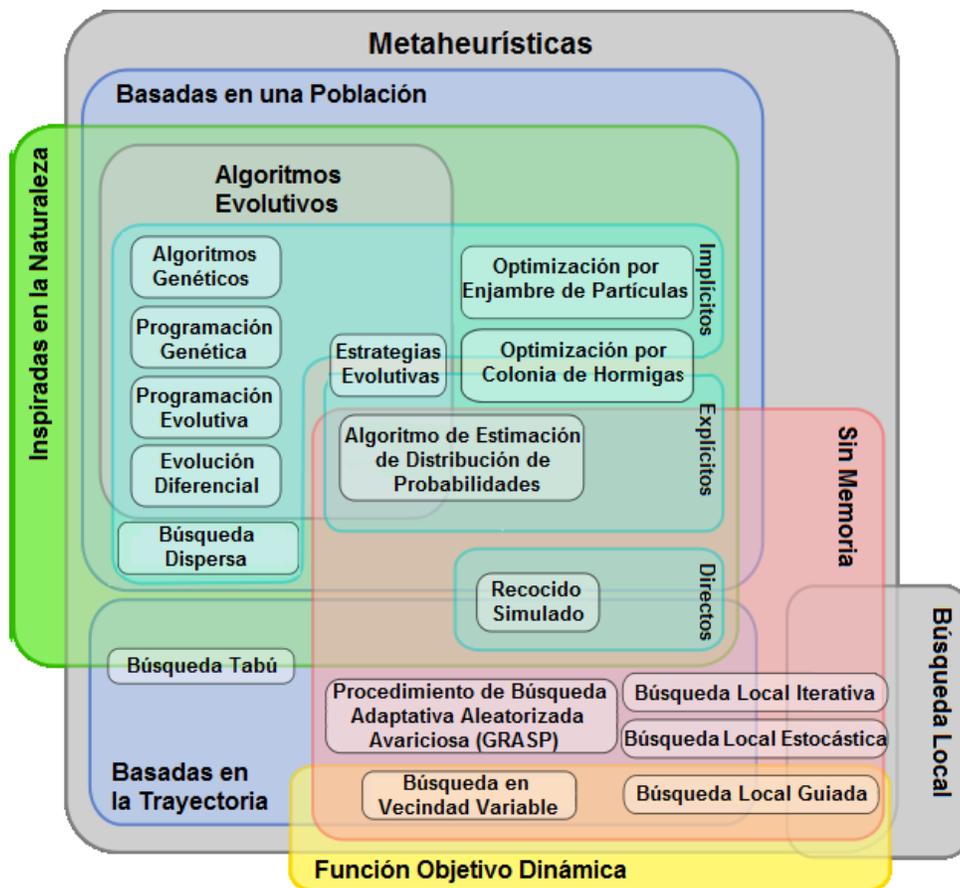


Figura 2.1: Clasificación de los algoritmos metaheurísticos más comúnmente utilizados en la literatura. Esta figura ha sido tomada de: https://commons.wikimedia.org/wiki/File:Metaheur%C3%ADsticas_clasificaci%C3%B3n.png

hoy en día forman parte de la misma familia de técnicas de computación evolutiva, y se conocen de manera general como AEs.

De acuerdo con Eiben [41], el funcionamiento general de un AE es el ilustrado en la Figura 2.2. En el contexto de los AEs, una solución candidata es entendida como un individuo dentro de una población (conjunto de soluciones). Los individuos de la población evolucionan mediante la aplicación de operadores genéticos como la recombinación (también conocida como cruce), que permite la mezcla de información entre dos o más individuos, y la mutación, que altera la información de un individuo de manera aleatoria. Los individuos de la población compiten unos contra otros, de modo que los individuos más aptos (aquellas soluciones que permiten obtener un mejor valor para la función



Figura 2.2: Esquema general del funcionamiento de un algoritmo evolutivo.

objetivo) tienen mayor probabilidad de ser seleccionados como padres y sometidos a los operadores genéticos para la creación de nuevos individuos hijo. Los individuos mejor adaptados también tienen mejores oportunidades de prevalecer y sobrevivir de una generación a la siguiente. Normalmente, este proceso evolutivo se repite hasta alcanzar un cierto número de generaciones, encontrar una solución con determinada calidad, o satisfacer algún otro criterio de terminación.

Los AEs se han utilizado con gran éxito gracias a su flexibilidad para adaptarse a problemas de optimización con características diversas, mostrando resultados prometedores. En la literatura se pueden encontrar numerosas aplicaciones de los AEs [18], donde estos algoritmos han reportado resultados competitivos al resolver problemas de optimización complejos del mundo real. Dada la complejidad que presenta el problema de estudio de este proyecto, se considera que los AEs representan un enfoque prometedor que puede conducir al diseño de una solución algorítmica adecuada para abordar este reto.

2.2 Estado del arte

El problema de seleccionar conjuntamente rutas de *midhaul* y determinar las divisiones funcionales apropiadas para satisfacer los requisitos de calidad de servicio de los vMNOs que aprovechen las

infraestructuras físicas es un problema que no ha sido abordado en la literatura a la fecha. En este capítulo se mencionan los trabajos relacionados a este problema, para los cuales se presenta una breve descripción, resultados principales y limitaciones:

- **2016, Xu y Wang, Algoritmo eficiente para la planificación de agrupaciones de unidades de banda base en Cloud-RAN [42]:**
 - **Descripción:** en este trabajo, los autores afrontan el problema de establecer la mejor ubicación para una piscina de recursos (BBU_{pool}) en C-RAN, este problema consiste en determinar la ubicación de las CUs de manera tal que den soporte a la mayor cantidad de DUs, teniendo en consideración la restricción de latencia en la ruta que conecta cada DU a la piscina de recursos. Los autores tratan de minimizar los costos de despliegue y, a la par, satisfacer la demanda de las estaciones base conectadas. Debido a que este problema es NP-difícil, los autores proponen un modelo aproximado con el cual disminuyen la complejidad del problema y proponen un algoritmo heurístico para solucionarlo. Los autores realizan una comparación entre el algoritmo propuesto, un algoritmo genético y un algoritmo de búsqueda tabú.
 - **Hallazgos / resultados principales:** la evaluación de los algoritmos fue realizada sobre diferentes diseños de red generados aleatoriamente, donde las DUs y las posibles piscinas de recursos están separadas uniformemente. Los resultados muestran que para las instancias del problema, el algoritmo propuesto es mejor en comparación con las metaheurísticas, logrando reducir los costos en un 16 % en comparación con el algoritmo genético. Con la evaluación de los experimentos se logra evidenciar que a medida en que la distancia entre las piscinas de recursos y las DUs aumenta, se logra dar servicio a una menor cantidad de DUs, esto debido a los requerimientos de latencia.
 - **Limitaciones / áreas de oportunidad:** en este trabajo los autores sólo consideran el escenario completamente centralizado, por lo cual las conexiones entre las DUs y la

BBU_{pool} , se establecen con fibra óptica. Se asume que los requerimientos de ancho de banda son cubiertos y el modelo propuesto por los autores se enfoca en cumplir la restricción de latencia en las rutas que conectan cada DU con la piscina de recursos. En los resultados se logra observar que para instancias grandes del problema (40 sitios posibles y 260 DUs), la calidad de las soluciones obtenidas por el algoritmo heurístico decrementa.

- 2017-2018, Harutyunyan y Riggio, Flex5G: división funcional flexible en redes 5G [43, 44]:
 - **Descripción:** en este trabajo, los autores se enfocan en resolver el problema de selección del grado de centralización. Este problema consiste en seleccionar la división funcional que permita minimizar la interferencia entre celdas y la utilización del ancho de banda del *fronthaul*. Esta tarea se vuelve compleja para el administrador de la red, debido a la cantidad de parámetros que se deben considerar. Los autores proponen un algoritmo heurístico que permite afrontar este problema. El algoritmo consta de tres pasos: 1) seleccionar un grupo de RRHs candidatos por cada celda del nodo virtual RRH (*vRRH*), considerando la configuración de la antena, recursos físicos y ubicación deseada; 2) crear una lista de vecinos de cada uno de los nodos *vRRH* candidatos, esta lista contiene a todos los RRHs cuya señal puede interferir con la señal del *vRRH*; y 3) asignar recursos y elegir la división funcional apropiada para el *vRRH*.
 - **Hallazgos / resultados principales:** este trabajo fue evaluado usando una topología de red en estrella con 8 RRHs y una piscina BBU, los resultados muestran que el nivel de división de funcionalidades cambia sustancialmente las restricciones del ancho de banda y los requerimientos de procesamiento.
 - **Limitaciones / áreas de oportunidad:** este trabajo toma en consideración tres posibles divisiones funcionales y sólo considera las conexiones de *fronthaul*, no se realiza la selección

de rutas y no se consideran las conexiones de *backhaul*. Considerar todo esto incrementaría la complejidad del problema afrontado.

■ 2018, Garcia-Saavedra y colaboradores, FluidRAN: Orquestación optimizada de vRAN/MEC [45]:

- **Descripción:** en este trabajo, los autores abordan el problema de diseño de una vRAN (*Virtualized Radio Access Network*), proponiendo un modelo que considera los costos de la RAN, las divisiones funcionales y las rutas entre las DUs y CUs. Los costos de la RAN se derivan del costo de cada ruta seleccionada, el costo de instanciar una máquina virtual (ya sea en las CUs o en las DUs) y los costos de darle servicio a cada petición. Las divisiones funcionales que se consideran son tres: (1) PDCP/RLC (opción 2); (2) MAC-PHY (opción 6); (3) PHY-RF (opción 8). Estas divisiones funcionales las representan como una cadena de funciones (f_0, f_1, f_2, f_3) , donde la primera función (f_0) se ejecuta en la DU y corresponde a las tareas básicas de radio, la cuarta función (f_3) corresponde a las funciones que siempre se establecen en la CU (funciones de las subcapas PDCP y RRC), la función f_1 corresponde a las funciones de las subcapas PHY que no son parte de f_0 y la función f_2 , corresponde a las funciones de las subcapas RLC y MAC. A las funciones f_1 y f_2 , se les debe elegir el lugar donde serán implementadas (en las CUs o las DUs).
- **Hallazgos / resultados principales:** los autores evalúan el modelo propuesto usando tres redes reales y emplean un algoritmo exhaustivo inspirado en el método de Benders [46] para encontrar las soluciones óptimas. Los autores demuestran que el problema afrontado es NP-difícil y determinan que la centralización completa (C-RAN) para las redes evaluadas es inviable. Por otra parte, determinan que al elegir un diseño distribuido (D-RAN), los costos aumentan en comparación a un diseño flexible (vRAN) determinado por la selección de las divisiones funcionales y las rutas para cada DU-CU.

- **Limitaciones / áreas de oportunidad:** el modelo propuesto por los autores sólo considera tres divisiones funcionales, considerar más funcionalidades haría más complejo el problema. Dado que el algoritmo empleado es un algoritmo de búsqueda exhaustiva, el problema se vuelve intratable para redes más grandes (redes con más de 200 DUs, más de una CU y con más de 6 rutas que conectan a cada una de las DUs con la CU).
- **2018, Garcia-Saavedra y colaboradores, WizHaul: Sobre el grado de centralización de Cloud RAN Next Generation Fronthaul [17]:**
 - **Descripción:** en este trabajo, los autores hacen notar el reto significativo al que se enfrentan los administradores de red y proponen un algoritmo de ramificación y poda con retroceso que permite obtener una solución cercana al óptimo para el problema de SCRDF, con el fin de poder aprovechar los recursos actuales de la red y ayudar a migrar hacia las nuevas tecnologías 5G. También se proponen dos algoritmos voraces para problemas de mayor escala. Se realiza una comparación entre los tres algoritmos propuestos y tres algoritmos que afrontan los subproblemas de forma individual; esta comparación se realiza usando diferentes topologías de red.
 - **Hallazgos / resultados principales:** los autores determinan que este nuevo problema es NP-difícil y, por simplicidad, toman en consideración un modelo basado en paquetes y restricciones de ancho de banda y latencia. Los resultados demuestran que este problema debe ser afrontado de manera conjunta, esto debido a la cantidad de soluciones no válidas obtenidas por los algoritmos que afrontan los subproblemas de manera individual.
 - **Limitaciones / áreas de oportunidad:** dado que los enfoques propuestos en este trabajo sólo son capaces de producir soluciones aproximadas, no es posible determinar qué tan cercanas se encuentran estas soluciones con respecto a la solución óptima. El modelo presentado por los autores no considera (de manera directa o indirecta) las ventajas obtenidas al centralizar las CUs en piscinas de recursos.

- 2019, Molner y colaboradores, **Optimización de una red fronthaul/backhaul integrada bajo restricciones de ruta y retardo [47]:**
 - **Descripción:** en este trabajo, los autores proponen una heurística para afrontar el problema de direccionamiento (o ruteo) y ubicación de recursos, teniendo en cuenta restricciones del *midhaul*. El algoritmo heurístico intenta encontrar las mejores ubicaciones para los XPU (conjunto de unidades centrales o piscinas de BBUs en el *midhaul*) tratando de acomodar la mayor cantidad de unidades distribuidas (DU) o RRHs, que un XPU pueda soportar.
 - **Hallazgos / resultados principales:** la heurística fue evaluada usando una topología de pequeña escala, la cual contiene siete nodos conectados en anillo, donde cada nodo posee tres RRH. Dicha evaluación mostró que para esa configuración, fue necesario un solo XPU con 21 DUs. En la topología a gran escala, se cuenta con 339 nodos y las soluciones requirieron de 12 XPUs. Dada la complejidad de la primer heurística, los autores proponen otra heurística computacionalmente más eficiente que permite afrontar problemas grandes, pero sacrificando calidad en la solución.
 - **Limitaciones / áreas de oportunidad:** en este trabajo, los niveles de división funcional no son tomados en cuenta de manera explícita (se asume que se puede establecer una división de capa física en cada DU/CU). Los autores tratan de establecer la menor cantidad de nodos CU y buscan que las CUs se encuentren en grupos (piscinas de recursos). Esto se traduce en ubicar a cada CU cerca de las otras CUs.
- 2019, Ojaghi y colaboradores, **Sliced-RAN: Unión de rebanadas y división funcional en redes de acceso de radio 5G futuras [48]:**
 - **Descripción:** en este trabajo, los autores proponen un modelo que une las divisiones funcionales con segmentos de RAN (los segmentos de RAN son denominados *slices*),

este modelo incluye la selección de la ruta que deben seguir los flujos del tráfico proveniente de los usuarios. Este modelo tiene como objetivo la minimización de los costos computacionales. Los costos computacionales miden la cantidad de funcionalidades virtualizadas en las CUs y la cantidad de funcionalidades implementadas en las DUs, siendo más costoso implementar funcionalidades en las DUs. Debido a esta relación entre los costos computacionales, el grado de centralización corresponde al inverso de los costos obtenidos. Debido a la complejidad del problema, los autores emplean aproximaciones para resolver el modelo propuesto y realizan un conjunto de experimentos usando simulación de Monte Carlo.

- **Hallazgos / resultados principales:** los experimentos realizados muestran que los costos computacionales usando los segmentos de RAN son más elevados en comparación con las diferentes divisiones funcionales. Sin embargo, el rendimiento de la red se ve aumentado al usar segmentos de RAN en comparación con las divisiones funcionales.
- **Limitaciones / áreas de oportunidad:** en este trabajo no se tienen en consideración las piscinas de recursos, las cuales tienen como objetivos principales la disminución de costos y la cooperación entre las CUs. Esta consideración debería mostrar un comportamiento similar entre la propuesta de los autores y las divisiones funcionales de capa física, puesto que los costos computacionales disminuyen y el desempeño de la red aumenta. Esto se debe a que se ha establecido [5] que por cada CU con funcionalidad de capa física se puede dar soporte de entre 5 y 2395 DUs (dependiendo del ancho de banda, latencia y capacidad de procesamiento) y, como los autores resaltan, el desempeño de las divisiones de capa física no depende del tráfico de los usuarios.

En resumen, en la Tabla 2.2 se presentan las propuestas encontradas en la literatura, donde cada renglón corresponde a uno de los trabajos antes descritos. La segunda columna de la tabla establece el tráfico de la red que se consideró en cada propuesta. La tercer columna describe si en la propuesta

Tabla 2.2: Tabla de resumen del estado del arte. Cada renglón de la tabla muestra una propuesta encontrada en la literatura. Las columnas representan los aspectos que se tomaron en cuenta durante la revisión realizada.

Propuesta	Tráfico de red	Selección de rutas	Niveles de split	Enfoque de solución
2016, Xu y Wang [42]	fronthaul	✓	✗	algoritmo heurístico
2017, Harutyunyan y Riggio [43]	fronthaul	✗	✓	algoritmo heurístico
2018, Harutyunyan y Riggio [44]	fronthaul	✗	✓	algoritmo heurístico
2018, Garcia-Saavedra y col. [45]	midhaul	✓	✓	algoritmo exhaustivo
2018, Garcia-Saavedra y col. [17]	fronthaul	✓	✓	algoritmo heurístico
2019, Molner y col. [47]	midhaul	✓	✗	algoritmo heurístico
2019, Ojaghi y col. [48]	midhaul	✓	✓	No aplica

se abordó el problema de selección de rutas. La cuarta columna determina si la propuesta considera las divisiones funcionales. Finalmente, la quinta columna indica el enfoque de solución, es decir, los tipos de algoritmos propuestos para solucionar el problema.

El trabajo realizado por Garcia-Saavedra y colaboradores en [17], es el primer trabajo encontrado que aborda el problema de SCRDF en arquitecturas Cloud-RAN. También se puede apreciar a partir de esta revisión que los trabajos enfocados al problema de SCRDF [45, 17] no abordan de manera directa la agrupación de CUs en piscinas de recursos. Este es un factor que ayudaría a determinar el grado de centralización de la red, ya que agrupar las CUs permite disminuir los costos operativos (renta, refrigeración, entre otros) y aumentar el aprovechamiento de los recursos disponibles. Esta revisión de la literatura también hace evidente que el problema de SCRDF en redes móviles aún no ha sido abordado desde el punto de vista de IC, de manera que la idoneidad de este tipo de enfoques en este contexto específico permanece como una ruta de investigación aún sin explorarse. El presente

trabajo de tesis representa un aporte a la literatura del área mediante la exploración de tal ruta de investigación.

2.2.1 Trabajos relacionados con el enfoque propuesto

El presente trabajo de tesis aborda el problema de seleccionar conjuntamente las rutas de la red de *midhaul* para conectar las DUs y CUs de la NGFI-II, y determinar las divisiones funcionales apropiadas para satisfacer los requisitos de QoS de los vMNOs. En este trabajo de tesis, dicho problema es modelado como un problema de incrustación de redes virtuales. Como se detalla en el Capítulo 3, el modelo propuesto se basa en elementos y consideraciones que se han adoptado previamente en la literatura especializada, principalmente en los trabajos [17, 43, 44, 45]. Estos trabajos han sido seleccionados debido a que abordan el problema de la elección adecuada de las divisiones funcionales. Por lo tanto, estos trabajos que aportan representaciones de las divisiones funcionales, nos permiten determinar los elementos de la red que se deben considerar y nos brindan información para seleccionar el nivel de centralización adecuado.

En los trabajos [43, 44], los autores modelan los recursos de una red física (o red de sustrato) y un conjunto de peticiones de recursos virtuales que serán servidos por la red física. Estos dos trabajos modelan múltiples peticiones de recursos, así como el consumo o uso de los recursos de la red física. Los autores no realizan una búsqueda de la mejor asignación de parejas DU/CU. Por el contrario, esta tarea se delega al usuario que realiza la petición de los recursos (proveedor de servicio), quien debe dar una ubicación deseada (o aproximada) para las DUs y las CUs. Debido a que el modelo no considera los nodos intermedios o nodos de transmisión (es decir, nodos que no son CUs o DUs), el problema de elegir una de las rutas posibles no se aborda en estos trabajos.

El modelo presentado en [45] no aborda las peticiones de recursos por parte del (o los) proveedor(es) de servicio, pero modela el uso de los recursos computacionales que se requieren para dar servicio a las divisiones funcionales. Las divisiones funcionales son modeladas como distintas cadenas de funciones que se pueden virtualizar. El modelo considera las posibles asignaciones para

Tabla 2.3: Tabla comparativa entre los trabajos seleccionados de la literatura y el modelo propuesto. Las filas de la tabla corresponden a los trabajos seleccionados de la literatura, la última fila corresponde al modelo propuesto en este trabajo de tesis. Las columnas corresponden a las características o elementos que considera el modelo.

Enfoque	Peticiones de recursos	Uso de recursos	Asignación DU/CU	Elección de rutas	Piscinas de recursos
Flex5G [43, 44]	✓	✓	✗	✗	✗
FluidRAN [45]	✗	✓	✓	✓	✗
WizHaul [17]	✗	✗	✓	✓	✗
Enfoque propuesto	✓	✓	✓	✓	✓

cada pareja DU/CU, aunque solo consideran una única CU. Dado que el modelo considera los nodos intermedios, el modelo aborda el problema de la elección de la mejor ruta posible.

Finalmente, el modelo descrito en [17] considera la asignación para cada pareja DU/CU. Debido a que las divisiones funcionales se realizan sobre las capas del protocolo E-UTRAN, las divisiones funcionales son representadas como el conjunto de funciones que se ejecutan en cada subcapa. Los autores también consideran el modelado de las rutas y toman en consideración los nodos intermedios. Sin embargo, no se consideran las peticiones ni el uso de los de recursos de la red.

A modo de resumen, la Tabla 2.3 presenta una comparación entre los elementos modelados en cada uno de los trabajos antes descritos. Es posible resaltar a partir de esta tabla, que ninguno de estos trabajos considera el modelado de las piscinas de recursos a nivel de CUs. Es decir, los autores parten de que ya existe un grupo de CUs, y no se considera la posibilidad de disminuir o aumentar la cantidad de grupos o la cantidad de elementos que forman parte de ese grupo. La última fila en la Tabla 2.3 corresponde a la propuesta de este trabajo de tesis, misma que se detallará en el siguiente capítulo. Como puede observarse, lo que buscamos con el modelo propuesto es considerar escenarios que tomen en cuenta todas las características antes mencionadas.

3

Modelado formal del problema

En este capítulo se describe a detalle el modelo que se propone para formalizar y abstraer los principales componentes del problema de SCRDF. Primeramente, la Sección 3.1 define los componentes principales del modelo, así como la notación utilizada. Basándose en estas definiciones, la Sección 3.2 plantea formalmente el problema de optimización, incluyendo la función objetivo a optimizarse y las restricciones que determinan la factibilidad de las soluciones. Finalmente, la Sección 3.3 presenta un análisis del tamaño del espacio de búsqueda para este problema.

3.1 Modelo propuesto

En esta sección se definen formalmente los componentes del modelo propuesto, así como la notación que será utilizada para hacer referencia a estos componentes a lo largo de este documento. La Tabla 3.1 muestra un resumen de la notación utilizada para representar los componentes principales de este modelo. Las definiciones que a continuación se presentan han sido organizadas con base a la correspondencia de los elementos del modelo con recursos físicos, con recursos virtuales,

con el mapeo entre recursos físicos y virtuales, o con la definición de una solución candidata para el problema abordado.

3.1.1 Red física

Los elementos que conforman la red física, también conocida como red de sustrato, se describen formalmente a continuación:

- La red física es representada por un grafo no dirigido, $G = (V, E)$, donde V y E corresponden a los conjuntos de nodos y enlaces físicos de la red, respectivamente.
- El conjunto de nodos de la red física V se define como $V = C \cup R \cup W$, donde C corresponde al conjunto de posibles CUs, R corresponde al conjunto de DUs, y W corresponde al conjunto de los nodos intermedios o nodos de transmisión.
- El conjunto de posibles CUs corresponde a los nodos con capacidades de procesamiento y en los cuales se pueden virtualizar las funcionalidades de las diferentes subcapas del protocolo E-UTRAN. El conjunto de CUs está dado por: $C = \{c_i \mid c_i = (x_i, y_i, w_{prc_i}), x_i, y_i \in \mathbb{R}^+, w_{prc_i} \in \mathbb{N}, 1 \leq i \leq n\}$, donde x_i, y_i representa las coordenadas de la i -ésima CU y w_{prc_i} corresponde a la cantidad máxima de recursos computacionales de este nodo físico.
- Las DUs son los nodos que implementan las funciones de radio frecuencia. El conjunto de todas las DUs de la red física se define como: $R = \{r_i \mid r_i = (x_i, y_i, \theta_i, w_{prc_i}, w_{ant_i}, w_{prb_i}), x_i, y_i \in \mathbb{R}^+, \theta_i, w_{prc_i}, w_{ant_i}, w_{prb_i} \in \mathbb{N}, 1 \leq i \leq m\}$. Cada DU física r_i está representada por sus coordenadas x_i, y_i , su radio de cobertura θ_i , su capacidad máxima de recursos computacionales w_{prc_i} , su cantidad de antenas w_{ant_i} , y su cantidad de bloques de recursos físicos w_{prb_i} .
- El conjunto de nodos de transmisión W se define como: $W = \{w_i \mid w_i = (x_i, y_i), x_i, y_i \in \mathbb{R}^+, 1 \leq i \leq l\}$, donde las coordenadas del i -ésimo nodo están dadas por x_i, y_i .

Tabla 3.1: Notación usada para los componentes principales del modelo propuesto.

Parámetro	Descripción
G	Red física o red de sustrato.
V	Conjunto de nodos físicos.
E	Conjunto de conexiones entre nodos físicos.
C	Conjunto de posibles CUs físicas, $ C = n$.
R	Conjunto de posibles DUs físicas, $ R = m$.
W	Conjunto de nodos intermedios (o nodos de transmisión), $ W = l$.
$P_{a,b}^j$	La j -ésima ruta entre los nodos a y b .
$P_{a,b}^*$	Conjunto de todas las rutas entre los nodos a y b .
F	Conjunto de todas las divisiones funcionales.
Q	Conjunto de todas las peticiones de recursos virtuales.
G_i^v	La i -ésima petición de recursos virtuales.
V_i^v	Conjunto de nodos virtuales solicitados por la i -ésima petición de recursos virtuales.
C_i^v	Conjunto de CUs virtuales solicitadas por la i -ésima petición de recursos virtuales, $ C_i^v = n'_i$.
R_i^v	Conjunto de DUs virtuales solicitadas por la i -ésima petición de recursos, $ R_i^v = m'_i$.
E_i^v	Conjunto de enlaces virtuales solicitados por la i -ésima petición de recursos virtuales.
$\Omega_{r_j}^i$	Conjunto de DUs físicas candidatas para la j -ésima DU virtual.
$\Delta_{c_k}^i$	Conjunto de CUs físicas candidatas para la k -ésima CU virtual.
$\Lambda_{e_{c_k^v, r_j^v}}^i$	Conjunto de todas las rutas candidatas para el enlace virtual $e_{c_k^v, r_j^v}$.
$s_{j,k}^i$	Asignación de ruta y división funcional para la i -ésima petición, j -ésima DU virtual y k -ésima CU virtual.
BBU_{pool_ρ}	Conjunto de CUs que representan a la ρ -ésima piscina de recursos.
$R_{L_1}^v$	Conjunto de asignaciones de divisiones funcionales de capa física para cada DU virtual.
$R_{L_2}^v$	Conjunto de asignaciones de divisiones funcionales de capa de enlace a datos para cada DU virtual
$R_{L_3}^v$	Conjunto de asignaciones de divisiones funcionales de capa de red para cada DU virtual.
X	Configuración para todas las peticiones de recursos.

- Un enlace entre dos nodos de la red física se define como: $e_{v_i, v_j} = (v_i, v_j, b_{i,j}, d_{i,j}, \gamma_{i,j}) \mid v_i, v_j \in V, v_i \neq v_j, b_{i,j}, d_{i,j} \in \mathbb{N}, \gamma_{i,j} \in \mathbb{R}^+$, donde v_i, v_j son los nodos conectados, $b_{i,j}$ corresponde a la capacidad máxima del enlace, $d_{i,j}$ corresponde al retardo inducido por el enlace, y $\gamma_{i,j}$ corresponde a un costo relativo de acuerdo con el tipo de enlace físico (fibra óptica, cable de cobre o enlace inalámbrico).
- El conjunto de todos los enlaces de la red se define como: $E = \{e_{v_i, v_j}\}$.
- Una ruta entre dos nodos a y b se define como el conjunto de k enlaces y $k + 1$ vértices, donde el primer vértice corresponde al nodo a y el último vértice corresponde al nodo b . Formalmente, una ruta se define como: $P_{a,b}^j = \{e_{v_i, v_{i+1}} \mid e_{v_i, v_{i+1}} \in E, 1 \leq i \leq k, v_1 = a, v_{k+1} = b\}$.
- El conjunto de todas las rutas posibles entre los nodos a y b , contiene todas las p posibles rutas que inician en el nodo a y terminan en el nodo b . Formalmente, el conjunto está dado por: $P_{a,b}^* = \{P_{a,b}^1, \dots, P_{a,b}^p\}$.

3.1.2 Peticiones de recursos virtuales

El modelo propuesto considera un escenario en el que se busca satisfacer un total de q peticiones de recursos virtuales. El término petición de recursos virtuales se utiliza en este trabajo de tesis para hacer referencia a los requerimientos de recursos (es decir, CUs, DUs y enlaces) por parte de un vMNO. Estas peticiones de recursos virtuales se definen formalmente como sigue:

- El conjunto de todas las q peticiones de recursos virtuales se define como: $Q = \{G_1^v, \dots, G_q^v\}$.
- La i -ésima petición de recursos virtuales se representa como un grafo no dirigido, $G_i^v = (V_i^v, E_i^v)$, donde V_i^v corresponde a los nodos virtuales solicitados y E_i^v corresponde a los enlaces virtuales solicitados.

- El conjunto de todos los nodos virtuales solicitados en la i -ésima petición de recursos virtuales se define como: $V_i^v = C_i^v \cup R_i^v$, donde C_i^v corresponde al conjunto de CUs virtuales solicitados y R_i^v corresponde al conjunto de DUs virtuales de la i -ésima petición.
- El conjunto de CUs virtuales solicitadas se define formalmente como: $C_i^v = \{c_j^v \mid c_j^v = (w_{prc_j}^v), w_{prc_j}^v \in \mathbb{N}, 1 \leq j \leq n_i^v\}$, donde $w_{prc_j}^v$ corresponde al procesamiento solicitado para la j -ésima CU virtual.
- El conjunto de DUs virtuales solicitadas para la i -ésima petición se define como: $R_i^v = \{r_j^v \mid r_j^v = (x_j, y_j, w_{prc_j}^v, w_{ant_j}^v, w_{prb_j}^v), x_j, y_j \in \mathbb{R}^+, w_{prc_j}^v, w_{ant_j}^v, w_{prb_j}^v \in \mathbb{N}, 1 \leq j \leq m_i^v\}$, donde x_j, y_j representa las coordenadas deseadas, $w_{prc_j}^v$ corresponde a la cantidad de recursos computacionales, $w_{ant_j}^v$ corresponde a la cantidad de antenas, y $w_{prb_j}^v$ corresponde a la cantidad de bloques de recursos físicos especificados para la j -ésima DU virtual.
- Un enlace virtual se define formalmente como: $e_{c_k^v, r_j^v} = (c_k^v, r_j^v, b_{j,k}^v, d_{j,k}^v)$, donde r_j^v corresponde a la DU virtual, la cual se desea que se conecte con la CU virtual c_k^v , $b_{j,k}^v$ corresponde al mínimo ancho de banda solicitado, y $d_{j,k}^v$ corresponde al retardo máximo aceptable. Los enlaces virtuales permiten modelar las necesidades del proveedor de servicio para garantizar la calidad de sus servicios.
- El conjunto de todos los enlaces virtuales solicitados en la i -ésima petición se define como: $E_i^v = \{e_{c_k^v, r_j^v}\}$.

3.1.3 Mapeo de recursos físicos y virtuales

A continuación se definen formalmente los componentes del modelo que se relacionan con la asignación de recursos virtuales a recursos de la red de sustrato:

- El conjunto de todas las DUs candidatas que pueden dar servicio a la j -ésima DU virtual se define formalmente como: $\Omega_{r_j^v}^i = \{r_k \in R \mid r_j^v \in R_i^v, \text{distancia}(r_j^v, r_k) \leq \theta_k\}$, donde la

inecuación distancia $(r_j^v, r_k) \leq \theta_k$ permite determinar las DUs físicas cuya área de cobertura es adecuada de acuerdo con la posición deseada para la j -ésima DU virtual.

- El conjunto de todas las CUs candidatas para la j -ésima CU virtual que pertenece a la i -ésima petición, se define formalmente como: $\Delta_{c_k^v}^i = \{c_y \in C \mid c_k^v \in C_i^v, w_{prc_k}^v \leq w_{prc_y}\}$, donde la inecuación $w_{prc_k}^v \leq w_{prc_y}$ especifica que la cantidad de recursos computacionales de la CU física debe ser igual o mayor a la cantidad de recursos computacionales solicitados.
- El conjunto de todas las posibles rutas que pueden representar al enlace virtual $e_{c_k^v, r_j^v}$ se define formalmente como: $\Lambda_{e_{c_k^v, r_j^v}}^i = \{P_{r_x, c_y}^l \in P_{r_x, c_y}^* \mid r_x \in \Omega_{r_j^v}^i, c_y \in \Delta_{c_k^v}^i, 1 \leq l \leq p\}$.
- El conjunto de posibles divisiones funcionales se define como: $F = \{f_i \mid f_i = (\beta_i, \delta_i), \beta_i, \delta_i \in \mathbb{N}, 1 \leq i \leq 10\}$, donde β_i y δ_i corresponden a los requerimientos de ancho de banda y retardo para la i -ésima división funcional, respectivamente.
- La asignación de la ruta física y división funcional para la i -ésima petición, j -ésima DU virtual y k -ésima CU virtual, es representada como una tupla de cuatro elementos y se define formalmente como: $s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l)$.
- El conjunto que contiene todas las asignaciones de recursos para la i -ésima petición, es representado por S_i y se define formalmente como: $S_i = \{s_{j,k}^i \mid 1 \leq j \leq m_i', 1 \leq k \leq n_i'\}$.

3.1.4 Solución candidata

Una solución candidata para el problema de estudio de este trabajo de tesis, consiste en una configuración específica de asignaciones de recursos para las q peticiones virtuales. Formalmente:

- Una solución potencial será denotada por X , y corresponde a un conjunto particular de asignaciones de recursos para todas las q peticiones. Formalmente, $X = \{S_i \mid 1 \leq i \leq q\}$.

- El conjunto que contiene todos los enlaces físicos que han sido asignados se define como:

$$E^a = \{e_{v_\pi, v_{\pi+1}} \mid \exists s_{j,k}^i \in S_i, s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), e_{v_\pi, v_{\pi+1}} \in P_{r_x, c_y}^l, S_i \in X\}.$$

- El conjunto de asignaciones de división funcional de capa física se define como:

$$R_{L_1}^v = \{s_{j,k}^i \in S_i \mid s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), 1 \leq i \leq q, f_z \in \{f_6, \dots, f_{10}\}, S_i \in X\}.$$

- El conjunto de asignaciones de división funcional de capa de enlace a datos se define como:

$$R_{L_2}^v = \{s_{j,k}^i \in S_i \mid s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), 1 \leq i \leq q, f_z \in \{f_2, \dots, f_5\}, S_i \in X\}.$$

- El conjunto de asignaciones de división de capa de red se define como:

$$R_{L_3}^v = \{s_{j,k}^i \in S_i \mid s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), 1 \leq i \leq q, f_z \in \{f_1\}, S_i \in X\}.$$

- El conjunto que contiene las asignaciones que usan el enlace e_{v_a, v_b} se define formalmente como:

$$\mathcal{E}_{v_a, v_b} = \{s_{j,k}^i \in S_i \mid e_{v_a, v_b} \in P_{r_x, c_y}^l, P_{r_x, c_y}^l \in \Lambda_{e_{c_k^v, r_j^v}}^i, S_i \in X\}.$$

- El conjunto que contiene todas las DUs físicas asignadas para la i -ésima petición se define formalmente como: $R_i^a = \{r_x \mid \exists s_{j,k}^i \in S_i, s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), S_i \in X\}$.

- El conjunto que contiene las CUs físicas asignadas para la i -ésima petición se define formalmente como: $C_i^a = \{c_y \mid \exists s_{j,k}^i \in S_i, s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l), S_i \in X\}$.

- El conjunto que contiene todas las CUs que han sido asignadas está dado por: $\Gamma = \bigcup_{i=1}^q C_i^a$, donde C_i^a es el subconjunto de CUs físicas asignadas para la i -ésima petición (definido anteriormente).

- Un grupo de CUs (o piscina de recursos) es representado como el conjunto de CUs cuyos elementos han sido asignados, y que para todo par de elementos que hacen parte del conjunto existe una ruta disponible. Formalmente: $BBU_{pool_\rho} = \{c_i \mid \forall c_j \in BBU_{pool_\rho} (\exists P_{c_i, c_j}^l = \{e_{v_x, v_{x+1}} \mid v_x \in \Gamma, 1 \leq x < t, t \leq n, v_1 = c_i, v_t = c_j\}), c_i \neq c_j\}$.

- El conjunto de todas las piscinas de recursos está dado por: $BBU_{pools} = \{BBU_{pool_1}, \dots, BBU_{pool_\lambda}\}$.

3.2 Formulación del problema de optimización

Con los elementos considerados y definidos en la Sección 3.1, a continuación se presenta un planteamiento formal del problema SCRDF como un problema de optimización.

3.2.1 Función objetivo

La meta en nuestro problema es obtener una solución X , es decir, una asignación de recursos para todas las peticiones virtuales de los vMNOs, que permita maximizar el grado de centralización y optimizar el uso de los recursos de la red. Por lo tanto, definimos este problema de optimización formalmente como la tarea de determinar la mejor asignación posible X^* , tal que:

$$X^* = \arg \max_{X \in \mathcal{F}} f(X), \quad (3.1)$$

donde $\mathcal{F} \subset \mathcal{X}$ denota el conjunto de todas las asignaciones válidas posibles en el espacio de búsqueda \mathcal{X} , mismo que está definido por un conjunto de restricciones (ver Sección 3.2.2), y $f : \mathcal{F} \rightarrow \mathbb{R}$ es la función objetivo (criterio de optimización), a maximizarse, y está dada por:

$$f(X) = [DoC(X) \cdot (1 - UoL(X))], \quad (3.2)$$

donde

$$DoC(X) = \frac{w_1 \cdot |R_{L_1}^v| + w_2 \cdot |R_{L_2}^v| + w_3 \cdot |R_{L_3}^v|}{|BBU_{pools}| \cdot \sum_{i=1}^q m'_i}, \quad (3.3)$$

$$UoL(X) = \sum_{e_{v_a, v_b} \in E^a} \frac{\gamma_{a,b} \cdot \beta(v_a, v_b)}{|E| \cdot b_{a,b}}, \quad (3.4)$$

$$\beta(v_a, v_b) = \sum_{s_{j,k}^i \in \mathcal{E}_{v_a, v_b}} \beta_z. \quad (3.5)$$

La función objetivo definida en (3.2) está compuesta por dos factores. El primer factor (3.3) permite evaluar el nivel de centralización (DoC) de la red por medio de las distintas divisiones funcionales de las asignaciones de la solución X . Para lograr esto, se emplea una suma ponderada de la cantidad de asignaciones a cada capa del protocolo E-UTRAN. Los pesos w_1 , w_2 y w_3 se utilizan para definir el grado con el que cada capa de la pila de protocolos contribuye al nivel de centralización de las divisiones funcionales de la red. Dado que el objetivo es maximizar el nivel de centralización, estos tres pesos deben respetar la siguiente relación: $w_1 > w_2 > w_3$.

Por otra parte, el segundo factor, $(1 - UoL(X))$, permite determinar la cantidad y uso de los enlaces de las asignaciones de la solución X . El objetivo es disminuir el uso de los enlaces de la red física, minimizando la cantidad de enlaces necesarios para satisfacer las solicitudes de los vMNOs. Debido a la diversidad de los enlaces en la red *midhaul*, se adopta un enfoque jerárquico para evaluar el costo de los enlaces. Teniendo en cuenta el costo relativo del tipo de conexiones físicas ($\gamma_{a,b}$), en este trabajo de tesis se considera que el costo de los enlaces de fibra óptica es mayor que el de los enlaces por cable, y que a su vez el costo de los enlaces por cable es mayor que el de los enlaces inalámbricos. En (3.4), restamos $UoL(X)$ de 1 para que el factor pueda maximizarse, según lo requerido por nuestra función objetivo.

3.2.2 Restricciones

Las restricciones del problema, que determinan las características que debe cumplir una solución candidata para ser considerada válida, están dadas formalmente por las siguientes expresiones:

$$\forall e_{v_a, v_b} \in P_{r_x, c_y}^l (\exists s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l) : \beta_z \leq b_{a,b}). \quad (3.6)$$

La expresión en (3.6) corresponde a la restricción de ancho de banda asignado a cada enlace, y establece que para cada enlace que pertenece a la ruta que ha sido asignada, se debe cumplir que la capacidad del enlace sea mayor o igual al ancho de banda requerido por la división funcional.

$$\forall P_{r_x, c_y}^l (\exists s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l) : \sum_{e_{v_a, v_b} \in P_{r_x, c_y}^l} d_{a,b} \leq \delta_z). \quad (3.7)$$

La expresión en (3.7) corresponde a la restricción de retardo para cada ruta, y establece que para cada ruta que ha sido asignada, se debe cumplir que la suma de los retardos producidos por cada enlace sea menor o igual al retardo máximo requerido por la división funcional asignada.

$$\forall r_x \in \Omega_{r_j^v}^i (w_{prc_j}^v \leq w_{prc_x}). \quad (3.8)$$

La expresión en (3.8) corresponde a la restricción de la capacidad de recursos computacionales para cada DU física. Esta restricción establece que por cada DU física que pueda satisfacer a una DU virtual, se debe cumplir que los recursos computacionales requeridos por la DU virtual sean menores o iguales a los recursos computacionales disponibles en la DU física.

$$\forall c_y \in \Delta_{c_k^v}^i (w_{prc_j}^v \leq w_{prc_y}). \quad (3.9)$$

La expresión en (3.9) corresponde a la restricción de la capacidad de recursos computacionales para cada CU física. Esta restricción establece que por cada CU física que pueda implementar a una CU virtual, se debe cumplir que los recursos computacionales requeridos por la CU virtual sean menores o iguales a los recursos computacionales disponibles en la CU física.

$$\forall r_x \in \Omega_{r_j^v}^i (w_{ant_j}^v \leq w_{ant_x}). \quad (3.10)$$

La expresión en (3.10) corresponde a la restricción de la antenas disponibles para cada DU física, y establece que para cada DU física que puede ser asignada a una DU virtual, se debe cumplir que la cantidad de antenas solicitadas por cada DU virtual sea menor o igual a la cantidad de antenas disponibles en la DU física.

$$\forall r_x \in \Omega_{r_j^v}^i (w_{prb_j}^v \leq w_{prb_x}). \quad (3.11)$$

La expresión en (3.11) corresponde a la restricción de la capacidad de bloques de recursos físicos de cada DU física, y establece que por cada DU física que pueda implementar a una DU virtual, se debe cumplir que los bloques de recursos físicos requeridos por la DU virtual sean menores o iguales a los recursos computacionales disponibles en la DU física.

$$\forall S_i \in A (\exists s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l) : b_{j,k}^v \leq \beta_z). \quad (3.12)$$

La expresión en (3.12) corresponde a la restricción del mínimo ancho de banda asignado, y establece que para cada petición de ancho de banda de un enlace virtual, el ancho de banda asignado en la división funcional debe ser mayor o igual al ancho de banda solicitado en el enlace virtual.

$$\forall S_i \in A (\exists s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l) : \delta_z \leq d_{j,k}^v). \quad (3.13)$$

La expresión en (3.13) corresponde a la restricción de retardo máximo asignado, y establece que para cada petición de retardo máximo de un enlace virtual, el requerimiento de retardo de la división funcional debe ser menor o igual al retardo máximo solicitado en el enlace virtual.

$$\forall G_i^v (|C_i^a| = |C_i^v|). \quad (3.14)$$

La expresión en (3.14) corresponde a la restricción de CUs asignadas, y establece que por cada petición de recursos virtuales, la cantidad de CUs físicas asignadas debe ser igual a la cantidad de CUs virtuales solicitadas.

$$\forall G_i^v (|R_i^a| = |R_i^v|). \quad (3.15)$$

La expresión en (3.15) corresponde a la restricción de DUs asignadas, y establece que por cada petición de recursos virtuales, la cantidad de DUs físicas asignadas debe ser igual a la cantidad de DUs virtuales solicitadas.

$$|BBU_{pools}| > 0. \quad (3.16)$$

Finalmente, la expresión en (3.16), corresponde a la restricción de piscinas de recursos y establece que debe existir al menos una piscina de recursos.

3.3 Espacio de búsqueda

El espacio de búsqueda del problema de optimización descrito en la Sección 3.2, denotado por \mathcal{X} , representa el conjunto de todas las posibles soluciones al problema. Con los elementos del modelo descritos anteriormente podemos, por lo tanto, determinar el tamaño de dicho conjunto:

$$|\mathcal{X}| = \prod_{i=1}^q \left[\left(|\Delta_{c_k^v}^i| \right)^{n_i'} \cdot \left(|\Omega_{r_j^v}^i| \cdot |F| \cdot |P_{rx, cy}^*| \right)^{m_i'} \right]. \quad (3.17)$$

Como se puede observar a partir de la expresión en (3.17), la cardinalidad de \mathcal{X} está determinada por el número total de posibles asignaciones que pueden adoptarse, de manera independiente, para satisfacer cada una de las peticiones virtuales. Es decir, cada posible solución $X \in \mathcal{X}$ representa una asignación potencial de recursos para cada una de las q peticiones de recursos virtuales. Tal como

se describe en la Sección 3.1.2, la i -ésima petición involucra un total de n'_i CUs virtuales, para las que existen un total de $|\Delta_{c_k^v}^i|$ alternativas de asignación disponibles. De manera similar, la i -ésima petición involucra un total de m'_i DUs virtuales, para las que una asignación consiste en seleccionar una de las $|\Omega_{r_j^v}^i|$ DUs físicas disponibles, una de las $|F|$ divisiones funcionales posibles, y una de las $|P_{r_x, r_y}^*|$ rutas físicas existentes.

El análisis anterior revela que el problema de optimización estudiado en este trabajo de tesis define un espacio de búsqueda exponencial. Es decir, el tamaño del espacio de búsqueda \mathcal{X} crece exponencialmente con el número de peticiones de vMNOs, así como con la cantidad de recursos virtuales especificados en estas peticiones (es decir, con el número de CUs y DUs virtuales). La consideración de redes físicas grandes también plantea un reto significativo, ya que la red física define los conjuntos de recursos $(|\Delta_{c_k^v}^i|, |\Omega_{r_j^v}^i| \text{ y } |P_{r_x, c_y}^*|)$ que están disponibles y que pueden ser asignados para satisfacer los recursos virtuales solicitados.

Tal complejidad hace evidente la necesidad de desarrollar enfoques de solución capaces de explorar eficientemente el espacio de búsqueda exponencial de este problema, con la finalidad de identificar soluciones prometedoras a un costo computacional razonable.

3.4 Resumen y conclusiones

En este capítulo se presentó el modelado formal de los diferentes componentes que definen el problema de estudio de este trabajo de tesis. El modelo propuesto emplea elementos del problema de VNE para describir la manera en la que se relacionan la red física y las peticiones de recursos virtuales. La formalización de los elementos de la red física y de las peticiones virtuales permitió la definición de un mapeo entre recursos físicos y virtuales, que da lugar a una solución candidata para el problema abordado. A diferencia de otros modelos propuestos en la literatura, nuestro modelo es el primero en considerar simultáneamente múltiples peticiones de recursos, el uso eficiente de los

recursos de red, las asignaciones de DUs y CUs, la selección de rutas que conectan las DUs con las CUs, y el uso de piscinas de recursos, favoreciendo la reducción de la cantidad de CUs asignadas.

En este capítulo también se plantea formalmente nuestro problema como un problema de optimización combinatoria, mono-objetivo, con restricciones. Este planteamiento involucra la definición de la función objetivo a optimizarse, que pretende favorecer un buen compromiso entre el grado de centralización y el uso de los enlaces de la red. Del mismo modo, se han definido una serie de restricciones que determinan la factibilidad de las soluciones potenciales.

Finalmente, este capítulo concluye con un análisis que ha permitido resaltar la complejidad del problema estudiado. Dicho análisis revela que el espacio de búsqueda crece exponencialmente con el tamaño del problema, lo que hace evidente la necesidad de contar con algoritmos eficientes, que permitan obtener soluciones satisfactorias en tiempos razonables. El Capítulo 4 del presente documento describe nuestras dos propuestas algorítmicas para hacer frente a este reto.

4

Métodos de solución propuestos

En este capítulo se describe el diseño de los algoritmos propuestos para abordar el problema de SCRDF. El primer diseño considerado, presentado en la Sección 4.1, corresponde a un algoritmo de búsqueda exhaustiva. Este algoritmo permite identificar la solución óptima mediante la enumeración sistemática del espacio de búsqueda. Dado que el problema estudiado presenta un espacio de búsqueda exponencial (ver análisis del espacio de búsqueda en la Sección 3.3), el uso de este enfoque exhaustivo estará limitado a instancias pequeñas del problema, y servirá únicamente como punto de referencia para evaluar la capacidad de nuestras otras propuestas algorítmicas para producir soluciones que representen una buena aproximación al óptimo. El segundo diseño, presentado en la Sección 4.2, corresponde a un algoritmo avaro. El algoritmo avaro propuesto permite obtener soluciones satisfactorias rápidamente. Finalmente, en la Sección 4.3 se describe el diseño de un algoritmo evolutivo. Este enfoque (meta-)heurístico implementa una rutina de inicialización especializada, y cuenta operadores de variación que le permiten lograr una exploración más efectiva

del espacio de soluciones. Los algoritmos descritos en este capítulo se proponen para lograr distintos compromisos entre eficiencia computacional y la capacidad de producir soluciones de alta calidad.

4.1 Algoritmo de búsqueda exhaustiva

Como se mencionó anteriormente, el algoritmo de búsqueda exhaustiva garantiza encontrar la mejor solución posible, es decir, la solución óptima del problema abordado. Esto se logra mediante la evaluación sistemática de todas y cada una de las soluciones del espacio de búsqueda, eligiendo aquella solución factible (es decir, que cumpla con todas las restricciones planteadas en la Sección 3.2.2) que presente el mayor valor para la función objetivo (descrita en la Sección 3.2.1).

En este algoritmo, una solución candidata se representa como una matriz de q filas y $\max(n'_i + 3 \cdot m'_i)$ columnas, $1 \leq i \leq q$, donde q es el número de peticiones virtuales, y n'_i y m'_i representan el número de CUs y DUs virtuales en la i -ésima petición (ver Sección 3.1). Por lo tanto, cada fila de la matriz representa a la configuración de la asignación de recursos para una petición virtual.

La Figura 4.1 muestra un ejemplo de la representación de una solución. En este ejemplo, tenemos tres peticiones de recursos virtuales ($q = 3$), y tenemos un total de once columnas. Estas columnas se obtienen a partir de la cantidad de CUs y DUs solicitadas, la asignación de la división funcional y la asignación de ruta física (estos dos últimos elementos se deben de determinar para cada DU). Reemplazando los valores para n'_i y m'_i de las diferentes peticiones, obtenemos la cantidad de columnas: $(\max(n'_1 + 3 \cdot m'_1, \dots, n'_3 + 3 \cdot m'_3)) = \max(2 + 3 \cdot 3, \dots, 2 + 3 \cdot 3) = 11$.

Como se ilustra en la Figura 4.1, el identificador de la CU física (color azul), corresponde a la CU física asignada a la petición y toma valores desde 1 hasta $|C|$. De manera similar, el identificador de la DU física (color rojo) corresponde a la asignación de la DU que dará soporte a la DU virtual y toma valores desde 1 hasta $|R|$. El identificador de la ruta elegida (color verde) corresponde a la elección de una de las rutas que conectan a la DU física con la CU física y toma valores desde 1

Petición 1	1	2	1	2	5	2	3	2	3	2	3
Petición 2	2	3	3	3	4	1	1	3	2	1	1
Petición 3	1	1	1	7	5	1	3	2	3	1	5

Identificador de la CU física
 Identificador de la DU física
 Identificador de la ruta elegida
 Identificador de la división funcional

Figura 4.1: Ejemplo de la representación para una solución del problema. En este ejemplo, se consideran 3 peticiones de recursos virtuales, cada una de las cuales solicitando 2 CUs y 3 DUs.

hasta $|P_{r_i, c_j}^*|$. Finalmente, el identificador de la división funcional (color amarillo), toma valores desde 1 hasta $|F|$ y corresponde a la división funcional específica a utilizarse.

Con la finalidad de ejemplificar el uso de esta representación, la Figura 4.2 presenta una red física (red de sustrato), la Figura 4.3 presenta una petición de recursos (red virtual), y la Figura 4.4 muestra la representación de una posible configuración que permite asignar los recursos de la red física a la petición de recursos virtuales. La red de sustrato se compone de cinco DUs, dos CUs y tres nodos de transmisión. Esta red incluye una única ruta (en color rojo) que conecta la CU c_1 con la DU r_1 , y una única ruta (en color magenta) que conecta la CU c_1 con la DU r_2 . Por otro lado, la red incluye dos posibles rutas que permiten conectar la CU c_2 con la DU r_5 . Una de estas rutas (en color azul) pasa por un nodo de transmisión, mientras que la otra ruta (en color verde) pasa por dos nodos de transmisión. La red virtual o petición de recursos, se compone de dos grafos, el primer grafo contiene dos DUs virtuales (r_1^v y r_2^v), que están conectadas a una CU virtual (c_1^v) a través de los enlaces virtuales $e_{c_1^v, r_1^v}$ y $e_{c_1^v, r_2^v}$, respectivamente. El segundo grafo contiene una CU virtual (c_2^v) y una DU virtual (r_3^v), conectadas por el enlace virtual $e_{c_2^v, r_3^v}$.

La Figura 4.4 representa la incrustación de los elementos virtuales ilustrados en la Figura 4.3 sobre la red física de la Figura 4.2. Cada celda del vector representa una asignación realizada. En el ejemplo, tenemos que a la CU virtual c_1^v se le asignó la CU física c_1 , a la DU virtual r_1^v se le asignó la DU física r_1 , al enlace virtual $e_{c_1^v, r_1^v}$ se le asignó la ruta que conecta c_1 con r_1 , y se asignó la segunda división funcional (PDCP/RLC) para esta pareja de nodos físicos. Por otra parte, a la CU virtual c_2^v

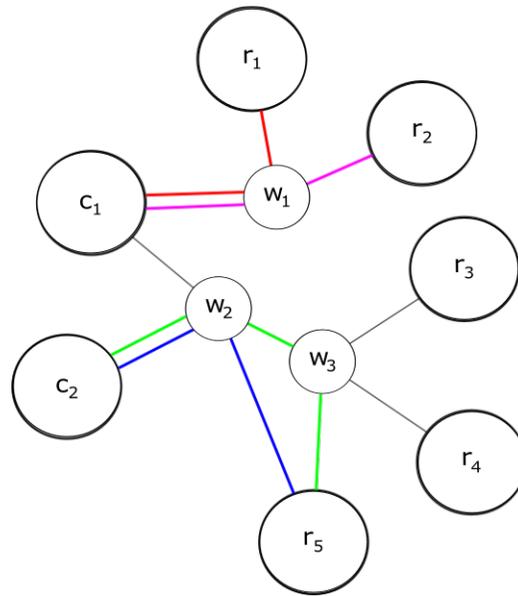


Figura 4.2: Ejemplo de una red física o red de sustrato. La red contiene dos CUs (c_1 y c_2), cinco DUs (r_1, r_2, r_3, r_4 y r_5) y tres nodos intermedios (w_1, w_2 y w_3).

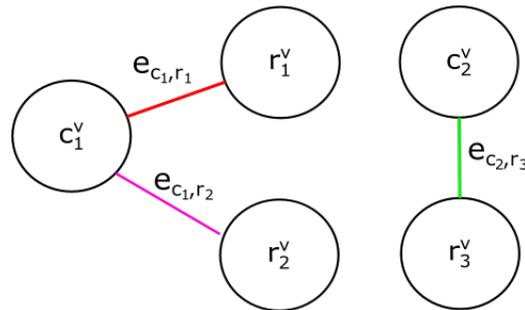


Figura 4.3: Ejemplo de una red virtual o petición de recursos de un vMNO. La petición consiste de tres DUs virtuales (r_1^v, r_2^v y r_3^v), dos CUs virtuales (c_1^v y c_2^v) y tres enlaces virtuales ($e_{c_1^v, r_1^v}, e_{c_1^v, r_2^v}$ y $e_{c_2^v, r_3^v}$).

c_1^v	r_1^v	$e_{c_1^v, r_1^v}$		r_2^v	$e_{c_1^v, r_2^v}$		c_2^v	r_3^v	$e_{c_2^v, r_3^v}$	
1	1	1	2	2	1	3	2	5	2	3
Identificador de la CU física	Identificador de la DU física	Identificador de la ruta elegida	Identificador de la división funcional	Identificador de la DU física	Identificador de la ruta elegida	Identificador de la división funcional	Identificador de la CU física	Identificador de la DU física	Identificador de la ruta elegida	Identificador de la división funcional

Figura 4.4: Ejemplo de una asignación para la petición de recursos de la Figura 4.3, sobre la red física de la Figura 4.2.

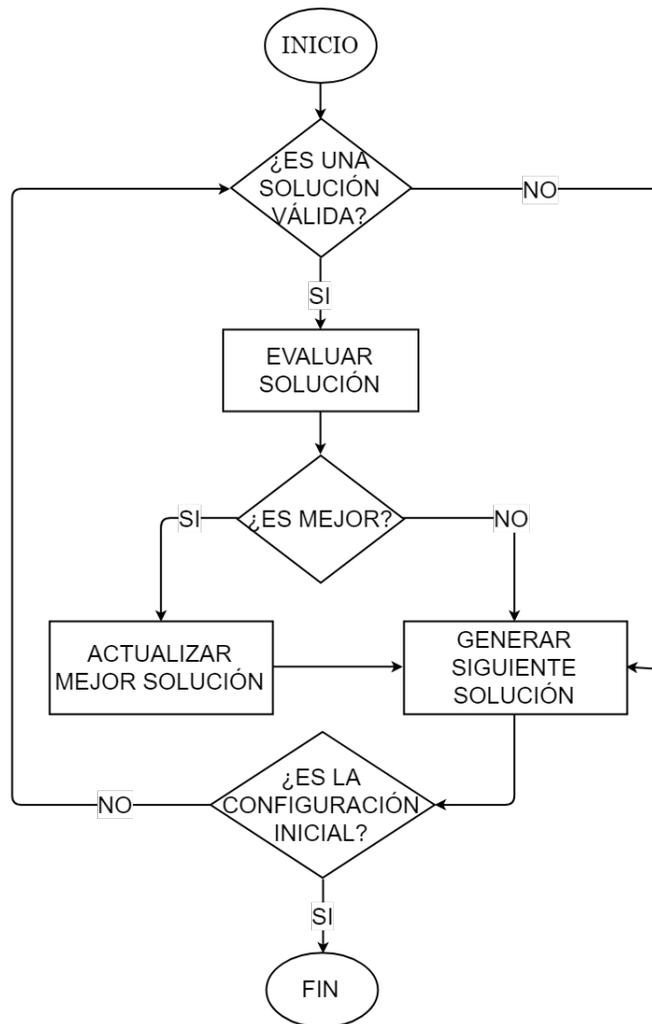


Figura 4.5: Diseño del algoritmo de búsqueda exhaustiva para identificar soluciones óptimas del problema de SCRDF.

se le asignó la CU física c_2 , a la DU virtual r_3^v se le asignó la DU física r_5 , al enlace virtual e_{c_2, r_3^v} se le asignó la segunda ruta (ruta de color azul en la red física) que conecta la c_2 con la r_5 y, finalmente, se le asignó la tercer división funcional (High RLC/Low RLC).

El algoritmo de búsqueda exhaustiva hace uso de esta representación, siguiendo un orden lexicográfico para recorrer las diferentes opciones de los posibles valores de cada posición de la matriz. El diseño del algoritmo se ilustra en la Figura 4.5. Este algoritmo parte de una configuración inicial (configuración base), la cual será modificada de manera iterativa. En cada iteración, se determina la

factibilidad de la nueva configuración (es decir, se verifica si cumple con todas las restricciones del problema). En caso de ser una configuración válida, se determina el valor de la función objetivo y se actualiza la mejor solución encontrada hasta el momento. De lo contrario, las configuraciones que no son válidas se descartan y se procede a generar la siguiente configuración. El algoritmo termina cuando se identifica que la nueva configuración generada corresponde a la configuración inicial (esto ocurre cuando se terminan de explorar todas las alternativas del espacio de búsqueda).

A pesar de la capacidad del algoritmo de búsqueda exhaustiva para garantizar la identificación de una solución óptima, la aplicación de este tipo de algoritmos está limitada a instancias relativamente pequeñas del problema. Esta limitante radica en la cantidad de posibles soluciones que se deben explorar, considerando que el espacio de búsqueda del problema SCRDF crece de manera exponencial (ver Sección 3.3). No obstante, este algoritmo representa un punto de referencia objetivo para la evaluación de algoritmos más sofisticados (que no garantizan encontrar una solución óptima), como los que se proponen en secciones posteriores de este mismo capítulo. El diseño de los algoritmos descritos en las siguientes secciones permite aprovechar el conocimiento que se ha obtenido sobre el problema para realizar una exploración eficiente del espacio de búsqueda, que permite identificar regiones prometedoras y generar soluciones satisfactorias en tiempos razonables.

4.2 Algoritmo avaro

En esta sección se describe el diseño del algoritmo avaro (AA) propuesto para abordar el problema de SCRDF. Los algoritmos avaros construyen soluciones de forma iterativa, tomando las decisiones que parecen ser las mejores en cada paso del proceso. Aunque tales elecciones localmente óptimas no suelen conducir al óptimo global, estas heurísticas tienden a producir resultados satisfactorios en tiempos razonables, y se han aplicado con éxito en una gran variedad de escenarios prácticos.

La Figura 4.6 presenta el diseño de nuestro AA. Este algoritmo implementa una política de prioridad basada en la cantidad de recursos solicitados, por lo que las peticiones virtuales son

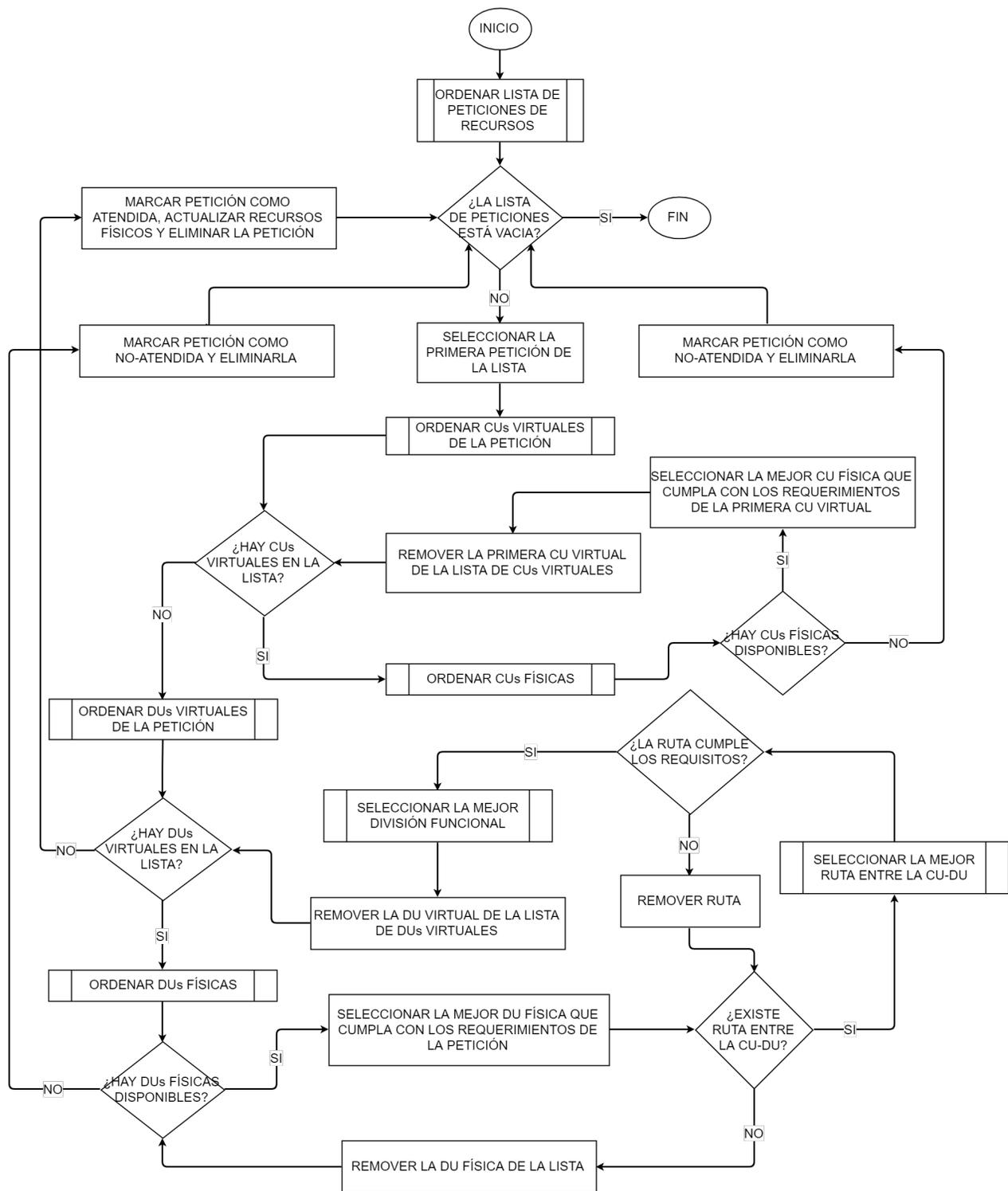


Figura 4.6: Diseño del algoritmo avaro propuesto para abordar el problema de SCRDF.

inicialmente ordenadas de manera descendente de acuerdo con este criterio. Esto permite que las peticiones sean atendidas, una por una, priorizando aquellas que requieren la mayor cantidad de recursos. Tales peticiones, demandantes de recursos, podrían ser difíciles de satisfacer si se consideraran después de que algunos recursos de la red ya se hayan asignado a otras peticiones.

El procesamiento de una petición empieza por la asignación de las CUs virtuales a CUs físicas. Para esto, las CUs virtuales son ordenadas (descendentemente) de acuerdo con la cantidad de recursos solicitados, y son procesadas para su asignación a una CU física en dicho orden. Las CUs físicas, por su parte, son ordenadas de acuerdo con su capacidad para satisfacer los requerimientos de la CU virtual. En particular, cuando dos o más CUs físicas que puedan satisfacer la demanda de la CU virtual se encuentran disponibles, el algoritmo favorece a la CU física que quedaría con la cantidad mínima de recursos disponibles después de dicha asignación. En otras palabras, el algoritmo elegiría una CU física que cumpla con los requerimientos, pero que a su vez los exceda en la menor medida posible. Esta estrategia conservadora pretende reservar CUs físicas ricas en recursos para otras CUs virtuales más demandantes que puedan ser parte de futuras peticiones de recursos. Una vez se realice la asignación, se procede con el procesamiento de la siguiente CU virtual, repitiéndose este proceso hasta que todas las CUs virtuales sean asignadas a un nodo físico.

Una vez asignadas todas las CUs virtuales, el AA centrará sus esfuerzos en la asignación de las DUs virtuales. De manera similar a las CUs, las DUs virtuales son ordenadas y procesadas con base en la cantidad de recursos que solicitan. La asignación de una DU física favorece (al igual que ocurre con las CUs) aquellos nodos que cumplan con los requerimientos de la DU virtual y que, al mismo tiempo, minimicen la cantidad de recursos disponibles después de la asignación. Una vez que la DU virtual es asignada a una DU física, se procede con la asignación de la mejor ruta disponible. En este caso, la mejor ruta será aquella ruta válida (es decir, que respeta los requerimientos de mínimo ancho de banda y máximo retardo definidos en por enlace virtual) que minimice el ancho de banda disponible después de la asignación (esto último adhiriéndose a la estrategia conservadora

mencionada anteriormente). Asimismo, se selecciona la mejor división funcional, que en este caso será aquella que permita lograr el máximo grado de centralización.

Finalmente, si todas las DUs virtuales logran ser asignadas, la petición se marca como atendida, se actualizan los recursos físicos disponibles, y se repite el proceso hasta procesar todas las peticiones virtuales. Las peticiones se marcan como rechazadas (o no atendidas), si no existen recursos físicos adecuados para satisfacer los requerimientos de cualquier recurso virtual solicitado. En caso de que esto último suceda, la petición rechazada no será configurada (i.e., embebida) en la red física.

4.3 Algoritmo evolutivo

En esta sección se describe el diseño del algoritmo evolutivo (AE) que se propone para abordar el problema de SCRDF. Como se explica en la Sección 2.1.3.4, los AEs son metaheurísticas inspiradas en el proceso natural de la evolución. El AE propuesto sigue el esquema generalizado que se ilustra en la Figura 4.7. El algoritmo comienza inicializando una población de individuos (soluciones candidatas), y posteriormente entra en un proceso iterativo que permite evolucionar estos individuos hasta que se cumple un criterio particular de terminación (en este caso, hasta que se alcanza un número máximo de generaciones, G_{max}). El proceso evolutivo consta de tres componentes principales: (i) el proceso de *selección para variación*, que identifica individuos padre adecuados para fines reproductivos; (ii) los *operadores de variación*, que crean nuevos individuos hijo a partir de los padres seleccionados; y (iii) la estrategia de *selección para supervivencia*, que determina los individuos padre y los individuos hijo que sobreviven de una generación a la siguiente. Todos los elementos que nos permiten instanciar este esquema generalizado para abordar nuestro problema se describen por separado a continuación.

4.3.1 Representación de las soluciones

El AE propuesto adopta la representación entera que fue descrita previamente en el contexto del algoritmo de búsqueda exhaustiva. Como se describe a detalle en la Sección 4.1, el genotipo de un

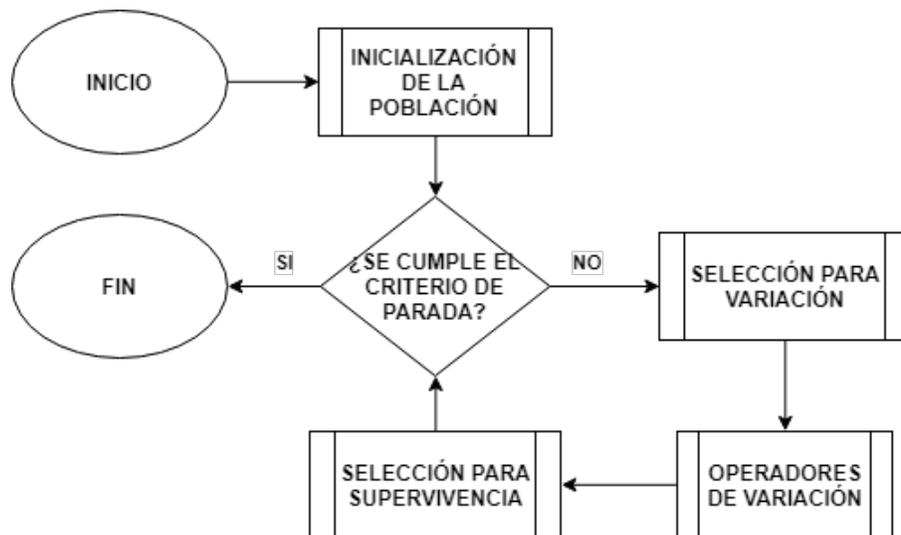


Figura 4.7: Diseño del algoritmo evolutivo que se propone para abordar el problema de SCRDF.

individuo está dado por una matriz de q renglones y $\max(n'_i + 3 \cdot m'_i)$ columnas, $1 \leq i \leq q$. En esta matriz, el i -ésimo renglón codifica la asignación de recursos para la i -ésima petición virtual.

4.3.2 Inicialización de la población

El AE propuesto emplea una rutina de inicialización especializada. Esta rutina tiene como objetivo generar una población inicial conformada por un conjunto diverso de N individuos potencialmente buenos. Para tales efectos, nuestra rutina de inicialización toma ventaja del AA propuesto en la Sección 4.2, como se describe a continuación.

El primer individuo de la población se construye utilizando el AA, exactamente como se describe en la Sección 4.2. Sin embargo, los $N - 1$ individuos restantes se generan utilizando una versión modificada de este AA. En lugar de procesar, seleccionar y asignar recursos siguiendo un orden particular de preferencia (que, como se explica en la Sección 4.2, se establece de acuerdo con la cantidad total de recursos solicitados y/o disponibles, según sea el caso), la versión modificada de este algoritmo toma todas estas decisiones de manera aleatoria. A pesar de tales decisiones aleatorias (poco usuales en un AA), la versión modificada del algoritmo conserva la característica del algoritmo

original de asignar nodos virtuales únicamente cuando se puedan encontrar nodos físicos adecuados (que cumplan con los requerimientos especificados por la CU o DU virtual). Por lo tanto, esta modificación transforma al AA, que originalmente sigue un proceso determinista, en una estrategia no determinista capaz de generar una solución diferente con cada una de sus ejecuciones.

4.3.3 Criterio de optimización

La función objetivo definida en (3.2), Sección 3.2.1, se adopta como función de aptitud (función de evaluación) para medir la calidad de las soluciones candidatas en el AE propuesto. Es importante notar que la expresión definida en (3.2) representa un criterio de optimización a maximizarse.

4.3.4 Selección para variación

Nuestro AE utiliza una estrategia de selección para variación en la que se otorga a todos los individuos de la población la misma probabilidad de ser seleccionados como padres, para la creación de nuevas soluciones candidatas. En esta estrategia panmíctica, los pares de padres a los que se aplicará el operador de recombinación (ver Sección 4.3.5) se seleccionan aleatoriamente (sin reemplazo) de la población. Esto permite reducir la presión de selección y promover la diversidad [49].

4.3.5 Operadores de variación

Nuestro método utiliza recombinación y mutación como operadores de variación. La recombinación toma como entrada dos individuos padre y produce como salida dos individuos hijo por medio del operador de *cruza uniforme* [50, 51]. Este operador funciona creando copias exactas de los padres de entrada, y posteriormente decidiendo si cada pieza de información genética se preserva o se intercambia entre estas copias con la misma probabilidad. La recombinación se aplica con una probabilidad dada p_r .

A todos los individuos hijo producidos mediante recombinación se les aplica *mutación uniforme*. Este operador altera independientemente cada posición en el genotipo del individuo de entrada con una probabilidad dada p_m . Cuando una posición es alterada, su valor original se reemplaza por otro, elegido uniformemente del conjunto de opciones disponibles para esa posición particular del genotipo.

4.3.6 Selección para supervivencia y manejo de restricciones

Nuestro método de selección para supervivencia (o estrategia de reemplazo) se basa en el uso de la jerarquización estocástica (SR) [52], una técnica efectiva para el manejo de restricciones [53]. Una vez que se ha creado una población de individuos hijo utilizando los operadores de variación, estos nuevos individuos compiten contra los individuos padre por un lugar en la población de la próxima generación. La técnica de SR provee de un mecanismo conveniente para jerarquizar a todos los individuos (padres e hijos), tomando en cuenta las restricciones que involucra nuestro problema.

El funcionamiento de la técnica de SR se describe en el Algoritmo 1.

Como puede apreciarse, SR emplea un procedimiento que muestra cierta similitud con el ordenamiento por el método de la burbuja, con la finalidad de ordenar una población de λ individuos. En este caso, λ corresponde al número total de individuos padre e individuos hijo que compiten por un lugar en la nueva población. SR realiza un total de n iteraciones, en las que se recorre la población comparando parejas de individuos consecutivos. En este trabajo de tesis, se ha adoptado un valor de $N = \lambda$, de acuerdo con las recomendaciones proporcionadas en [52]. La característica particular de la técnica de SR es el uso de un parámetro definido por el usuario, que en este trabajo se denota por p_s , que representa la probabilidad de usar la función objetivo (f) o el grado de violación de las restricciones (ϕ) como base para la comparación de la solución. Con la finalidad de promover la diversidad en la población, en nuestro algoritmo todos los individuos duplicados se filtran y descartan antes de aplicar SR.

Algoritmo 1 Algoritmo de Jerarquización Estocástica (SR) [52]

```

1: Función STOCHASTICRANKING( $Pop, \lambda, n, p_s$ )
    $Pop$  - población de individuos
    $\lambda$  - cantidad de individuos en la población
    $n$  - cantidad máxima de iteraciones
    $p_s$  - parámetro definido por el usuario
2:   para  $i = 1$  hasta  $n$  hacer
3:     para  $j = 1$  hasta  $\lambda - 1$  hacer
4:        $u = \text{aleatorio}(0, 1)$ 
5:       si  $(\phi(Pop_j) = \phi(Pop_{j+1}) = 0)$  o  $(u < p_s)$  entonces
6:         si  $f(Pop_j) > f(Pop_{j+1})$  entonces
7:            $\text{intercambiar}(Pop_j, Pop_{j+1})$ 
8:         fin si
9:       si no
10:        si  $\phi(Pop_j) > \phi(Pop_{j+1})$  entonces
11:           $\text{intercambiar}(Pop_j, Pop_{j+1})$ 
12:        fin si
13:      fin si
14:    fin para
15:    si no se realizaron intercambios entonces
16:      terminar ciclo
17:    fin si
18:  fin para
19:  devolver  $Pop$ 
20: fin Función

```

4.4 Resumen y conclusiones

En este capítulo se presentó una descripción detallada del diseño de los algoritmos que se consideran en este trabajo de tesis para abordar el problema de SCRDF. El primer diseño corresponde a un algoritmo de búsqueda exhaustiva, el cual realiza una exploración completa del espacio de búsqueda y, por lo tanto, garantiza la identificación de una solución óptima. El uso de este algoritmo está limitado a instancias muy pequeñas debido al crecimiento exponencial del espacio de búsqueda. Por lo tanto, el enfoque exhaustivo se ha considerado en este trabajo únicamente como punto de referencia, durante la evaluación de nuestras otras propuestas algorítmicas en los experimentos reportados en el Capítulo 5.

El segundo diseño corresponde al algoritmo avaro (AA) propuesto, que se caracteriza por la construcción rápida e iterativa de una solución candidata. El proceso iterativo del AA se basa en elegir la asignación de recursos que parece más prometedora en cada momento. Este algoritmo se propone como una alternativa que permite encontrar soluciones razonablemente buenas de manera muy eficiente (comportamiento que se confirma más adelante en el Capítulo 5). Finalmente, este capítulo presentó el diseño de un algoritmo evolutivo (AE). Se describieron a detalle los diferentes componentes del AE propuesto, incluyendo los mecanismos implementados para los procesos de selección, de variación, y de manejo de restricciones. El propósito de esta propuesta es proveer de una herramienta capaz de generar soluciones de mayor calidad, a cambio de un incremento en el tiempo de cómputo en comparación con el AA. De esta manera, nuestras propuestas AA y AE representan dos compromisos distintos entre eficiencia computacional y calidad de solución.

5

Evaluación y resultados

Este capítulo presenta los resultados de la experimentación realizada, cuyo objetivo es investigar la efectividad del algoritmo avaro (AA), descrito en la Sección 4.2, y el algoritmo evolutivo con inicialización avara (AE^{gre}), descrito en la Sección 4.3. Primeramente, la Sección 5.1 describe los escenarios de prueba, medidas de desempeño, y parámetros utilizados durante esta experimentación. Finalmente, la Sección 5.2 analiza los resultados obtenidos desde diferentes perspectivas.

5.1 Parámetros de evaluación

A continuación se proporcionan detalles sobre los experimentos que se realizaron en este trabajo de tesis, incluyendo los escenarios de prueba considerados, la configuración de parámetros adoptada para los algoritmos evaluados, y las medidas de desempeño utilizadas.

5.1.1 Escenarios de prueba

Definimos una instancia del problema como una tripleta, que consiste de una red de sustrato, de un conjunto de peticiones de recursos de vMNOs, y de un conjunto de rutas precalculadas. Para nuestros experimentos, generamos un total de 3,024 instancias de prueba, mismas que hemos clasificado en tres categorías diferentes: instancias pequeñas, instancias medianas, e instancias grandes. Cada una de estas categorías incluye un subconjunto de 1,008 instancias del problema.

Para las instancias de prueba de tamaño pequeño, puede adoptarse una de 42 topologías de red posibles. Las redes de sustrato de todas las instancias se generaron usando el kit de herramientas BRITE,¹ con valores aleatorios dentro de los siguientes rangos: la red más pequeña incluye 3 nodos y 3 enlaces, y la red más grande incluye 10 nodos y 12 enlaces. Las redes de tamaño pequeño se generaron con un número aleatorio de DUs entre 2 y 8, y con un número aleatorio de CUs entre 1 y 3. Las instancias de tamaño pequeño consideran uno de 8 posibles conjuntos de peticiones de recursos. Cada uno de estos conjuntos se generó con un número aleatorio de peticiones de recursos, entre 1 y 5 peticiones por conjunto. Cada petición de recursos se genera con un número aleatorio de DUs virtuales entre 2 y 7, y con un número aleatorio de CUs virtuales entre 1 y 3.

Para las instancias de tamaño mediano, la topología de red puede ser una de 56 opciones posibles. Las redes de sustrato se generaron aleatoriamente considerando los siguientes rangos: la red más pequeña tiene 100 nodos y 197 enlaces, y la red más grande tiene 500 nodos y 1,000 enlaces. Las redes se generaron con un número aleatorio de DUs entre 20 y 50, y un número aleatorio de CUs entre 14 y 26. Estas instancias incluyen uno de 6 conjuntos de peticiones de recursos posibles, cada uno de los cuales se generó con un número aleatorio de peticiones de recursos que varían de 5 a 50. Cada petición de recursos se generó con un número aleatorio de DUs virtuales en el rango de 2 a 10, y con un número aleatorio de CUs virtuales en el rango de 1 a 5.

¹BRITE - Boston University Representative Internet Topology Generator, software para generar topologías de red. Código fuente: <https://github.com/unly/brite>, página de inicio BRITE: <https://www.cs.bu.edu/brite/index.html>

Para instancias de tamaño grande, la topología de red puede ser una de 56 opciones posibles. Las redes de sustrato se generaron al azar utilizando los siguientes rangos: la red más pequeña tiene 600 nodos y 1,197 enlaces, y la red más grande tiene 1,000 nodos y 2,000 enlaces. Las redes se generaron con un número aleatorio de DUs entre 25 y 50, y un número aleatorio de CUs entre 40 y 200. Estas instancias incluyen uno de los 6 conjuntos de peticiones de recursos, cada uno generado con un número aleatorio de peticiones en el rango de 50 a 100. Cada petición de recursos tiene un número aleatorio de DUs virtuales entre 2 y 10, y un número aleatorio de CUs virtuales entre 1 y 5.

Para los tres grupos de instancias de prueba (pequeñas, medianas y grandes), generamos tres conjuntos de rutas precalculadas. Se usó la versión de k -rutas-más-cortas del algoritmo de Dijkstra, donde cada conjunto corresponde a las k -rutas entre cada par de nodos CU-DU, variando k desde 1 hasta 3.

Para las tres categorías de instancias, las cantidades de recursos disponibles y de recursos solicitados se generaron aleatoriamente utilizando los siguientes rangos de valores: para cada CU, w_{prc} toma un valor entre 10 y 20 (procesadores); para cada enlace físico, $b_{i,j}$ toma un valor entre 1,000 y 10,000 (Mbps), y $\gamma_{i,j}$ toma uno de los siguientes valores: 0.1, 0.2 o 0.7; para cada DU, w_{ant} toma un valor entre 10 y 50 (antenas), w_{prc} toma un valor entre 10 y 20 (procesadores), w_{prb} toma un valor entre 50 y 100 (bloques), y θ toma un valor entre 10 y 100 (km); para cada enlace virtual, $b_{i,j}^v$ toma un valor entre 100 y 1,000 (Mbps), y $d_{i,j}^v$ toma un valor entre 100 y 500 (ms); para cada CU virtual, w_{prc}^v toma un valor entre 2 y 6 (procesadores); para cada DU virtual, w_{ant}^v toma un valor entre 2 y 10 (antenas), w_{prc}^v toma un valor entre 1 y 5 (procesadores), y w_{prb}^v toma un valor entre 6 y 15 (bloques).

Las Tablas 5.1 y 5.2 presentan un resumen de los parámetros utilizados para generación de las redes de sustrato y de las peticiones de recursos virtuales, respectivamente. Los valores usados para las restricciones de ancho de banda (β_i) y latencia (δ_i) de cada división funcional (f_i) del modelo, corresponden a los presentados en la Sección 2.1.1, los cuales se describen (nuevamente) en la Tabla 5.3, enfocándonos en el caso de bajada (downlink).

Tabla 5.1: Parámetros empleados para generar las redes físicas o redes de sustrato.

Escenario	# nodos	# enlaces	# DUs	# CUs	# redes de
	mín, máx	mín, máx	mín, máx	mín, máx	sustrato
Pequeño	3, 10	3, 12	2, 8	1, 3	42
Mediano	100, 500	197, 1000	20, 50	14, 26	56
Grande	600, 1000	1197, 2000	40, 200	25, 50	56

Tabla 5.2: Parámetros empleados para generar los conjuntos de peticiones de recursos virtuales.

Escenario	# peticiones por conjunto	# DUs virtuales	# CUs virtuales	# conjuntos de peticiones
	mín, máx	mín, máx	mín, máx	de recursos
Pequeño	1, 5	2, 7	1, 3	8
Mediano	5, 50	2, 10	1, 5	6
Grande	50, 100	2, 10	1, 5	6

Tabla 5.3: Valores empleados para las restricciones de ancho de banda y latencia para cada una de las divisiones funcionales.

División funcional	β_i (Mbps)	δ_i (μ s)	División funcional	β_i (Mbps)	δ_i (μ s)
f_1	151	30000	f_6	173	250
f_2	151	30000	f_7	933	250
f_3	151	6000	f_8	1075	250
f_4	151	6000	f_9	1966	250
f_5	152	250	f_{10}	2457.6	250

5.1.2 Algoritmos evaluados y configuración de parámetros

Los experimentos realizados buscan investigar y comparar la efectividad de nuestros dos métodos propuestos: el algoritmo avaro (AA) descrito en la Sección 4.2, y el enfoque evolutivo (AE^{gre}) que

Tabla 5.4: Configuración de parámetros seleccionados para los algoritmos AE^{gre} y AE^{rnd} . La tabla muestra los valores específicos de las probabilidades de cruce (p_c), mutación (p_m) y jerarquización estocástica (p_s) que se utilizan para los escenarios de tamaño pequeño, mediano y grande.

Algoritmo	Instancias pequeñas	Instancias medianas	Instancias grandes
AE^{gre}	$p_r = 0,5$ $p_m = 3/\ell$ $p_s = 0,425$	$p_r = 0,7$ $p_m = 1/\ell$ $p_s = 0,425$	$p_r = 0,7$ $p_m = 1/\ell$ $p_s = 0,425$
AE^{rnd}	$p_r = 0,5$ $p_m = 3/\ell$ $p_s = 0,475$	$p_r = 0,5$ $p_m = 1/\ell$ $p_s = 0,475$	$p_r = 0,7$ $p_m = 1/\ell$ $p_s = 0,475$

ℓ se refiere al número total de posiciones (genes) en el genotipo de un individuo.

se introduce en la Sección 4.3. Adicionalmente, en nuestros experimentos consideramos una versión modificada del enfoque evolutivo, en la que se implementa una rutina de inicialización aleatoria. Esta variante, a la que se hará referencia como AE^{rnd} , nos permitirá evaluar en qué medida el uso de una rutina de inicialización especializada contribuye al desempeño de nuestra propuesta.

Se realizó un análisis preliminar para determinar una configuración de parámetros apropiada para los algoritmos AE^{gre} y AE^{rnd} . Los resultados detallados de este análisis se incluyen en el Anexo A. Con base en estos experimentos preliminares, se seleccionaron las configuraciones más prometedoras para los parámetros de cada uno de estos algoritmos, mismas que se usaron durante los experimentos cuyos resultados se reportan en la Sección 5.2. La Tabla 5.4 presenta un resumen de las configuraciones adoptadas para las probabilidades de cruce, mutación y jerarquización estocástica. En todos los casos, se utiliza un tamaño de población de $N = 100$, y un número máximo de generaciones de $G_{max} = 300$.

5.1.3 Evaluación de desempeño

Debido a la naturaleza estocástica de los enfoques AE^{gre} y AE^{rnd} , se realizaron un total de 31 ejecuciones independientes de estos algoritmos por cada instancia del problema considerada. Dado que nuestro método AA es determinista, se consideró una única ejecución de este algoritmo en todos los casos. En nuestros experimentos, el desempeño de estos algoritmos se evalúa en términos de la calidad de las soluciones, la tasa de aceptación de peticiones, y el tiempo de ejecución, como se describe más ampliamente a continuación.

5.1.3.1. Calidad de la solución

En este trabajo de tesis, la calidad de una solución se refiere al grado de centralización alcanzado mediante una determinada asignación de recursos. Esto es capturado adecuadamente por la función objetivo definida en (3.2), Sección 3.2.1, que debe ser maximizada. En nuestros experimentos, por lo tanto, la calidad de la solución corresponde al valor de la función objetivo de la mejor solución candidata encontrada por un algoritmo. Aunque los valores de nuestra función objetivo se definen en el rango $[0, 1]$, el mejor valor alcanzable puede variar de una instancia a otra. Por esta razón, los valores de la función objetivo obtenidos para una instancia del problema se han normalizado dividiéndolos entre el mejor valor (valor más alto) observado para esa instancia particular (considerando todos los algoritmos evaluados y las ejecuciones realizadas). En nuestro análisis de resultados, reportamos la calidad de solución promedio, así como también la calidad de solución más alta lograda por un algoritmo en el conjunto de todas las ejecuciones independientes realizadas.

5.1.3.2. Tasa de aceptación

Esta medida se calcula como la razón del número de peticiones de recursos virtuales que se asignaron con éxito a recursos físicos, que se denota por q' , con respecto al número total de peticiones,

q . Formalmente, la tasa de aceptación se calcula como q'/q . Esta medida se define en el rango $[0, 1]$, y los valores más altos indican un mejor desempeño.

5.1.3.3. Tiempo de ejecución

Para realizar una comparación justa basada en el tiempo de ejecución, nos aseguramos de que todos los algoritmos evaluados se ejecutaran bajo condiciones similares. Nuestros experimentos para medir el tiempo de ejecución fueron realizados utilizando un servidor dedicado exclusivamente para estos propósitos. Este servidor se encuentra equipado con 2 procesadores Intel E5-2630 (2.30 GHz), 32 GB de RAM, y el sistema operativo Ubuntu 18.04.4 LTS.

5.1.3.4. Análisis de significancia estadística

Para realizar el análisis estadístico de los resultados obtenidos en cada una de las métricas empleadas, se hizo uso de la herramienta IBM SPSS Statistics². Se realizó la prueba no-paramétrica (no se asume una distribución normal de los datos observados) de Kruskal-Wallis[54], la cual nos permite determinar si un conjunto de muestras independientes, siguen la misma distribución. Es decir, nos permiten determinar si existe (o no) una diferencia estadísticamente significativa entre los resultados obtenidos por los algoritmos evaluados. Esta prueba plantea las siguientes hipótesis:

- Hipótesis nula (H_0): las muestras observadas siguen la misma distribución.
- Hipótesis alternativa (H_1): existe al menos una muestra que posee una distribución diferente.

Para un valor de significancia α , esta prueba determina un valor- p de los datos observados. Este valor se interpreta de la siguiente forma:

- Si $\text{valor-}p \leq \alpha$, entonces H_0 es rechazada y se infiere que existe diferencia estadísticamente significativa entre las muestras observadas (se acepta la hipótesis alternativa H_1).

²IBM SPSS Statistics, pagina de inicio: <https://www.ibm.com/analytics/spss-statistics-software>

- Si $\text{valor-}p > \alpha$, entonces H_0 es aceptada y se infiere que no existe diferencia estadísticamente significativa entre las muestras observadas.

Si la prueba de Kruskal-Wallis llega a determinar una diferencia estadísticamente significativa entre las muestras observadas, se realiza la prueba de Mann-Whitney[55] por parejas, aplicando la corrección de Bonferroni. La corrección de Bonferroni consiste en ajustar el nivel de significancia estadística (α) de la siguiente manera:

$$\alpha^* = \frac{\alpha}{k},$$

siendo k la cantidad de comparaciones que se realizan para alguna de las muestras involucradas.

Para un valor ajustado de significancia α^* , esta prueba determina un valor- p para la pareja de datos observados. Este valor se interpreta de la siguiente forma:

- Si $\text{valor-}p \leq \alpha^*$, entonces H_0 es rechazada y se infiere que existe diferencia estadística significativa entre el par de muestras evaluadas.
- Si $\text{valor-}p > \alpha^*$, entonces H_0 es aceptada y se infiere que no existe diferencia estadística significativa entre el par de muestras evaluadas.

El valor de significancia α elegido para realizar el análisis estadístico se establece en $\alpha = 0,05$. Estos análisis estadísticos se aplicaron a las tres medidas (calidad de la solución, tasa de aceptación y tiempo de ejecución) descritas anteriormente. La cantidad de comparaciones que se realizan por cada muestra corresponde a $k = 2$, obteniendo un valor ajustado de significancia $\alpha^* = 0,025$.

5.2 Resultados

Esta sección presenta los resultados de una serie de experimentos realizados para investigar la efectividad y el comportamiento de los dos algoritmos propuestos en el Capítulo 4: nuestro algoritmo avaro denotado por AA , y nuestro método evolutivo denotado por AE^{gre} . Además, como se indica

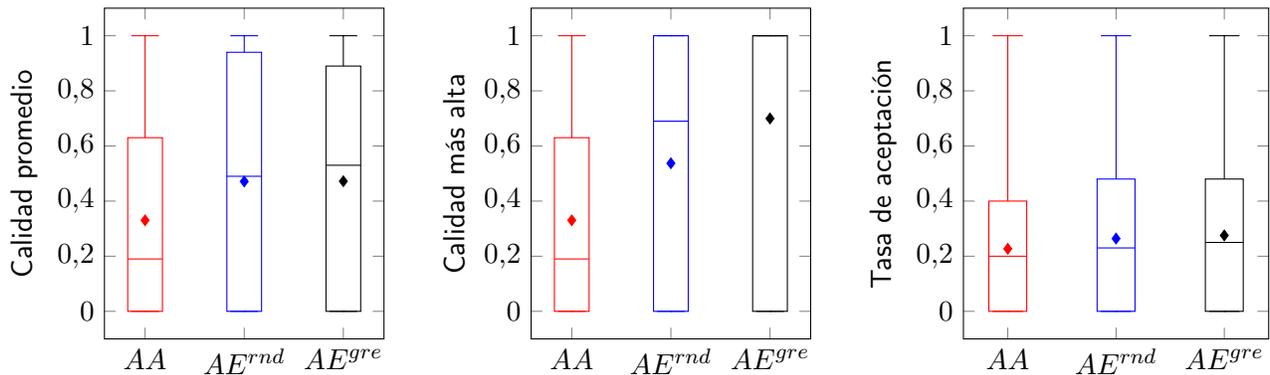
en la Sección 5.1.2, una versión modificada de nuestro algoritmo evolutivo, denotada por AE^{rnd} , se incluye en nuestro análisis comparativo como punto de referencia. Los tres algoritmos se investigan en el contexto de una colección diversa de escenarios de prueba, que se organizan en tres categorías de acuerdo con su tamaño, como se describe en la Sección 5.1.1. El desempeño de estos algoritmos se evalúa en función de la calidad de solución promedio, la calidad de solución más alta observada, la tasa de aceptación de peticiones, y el tiempo de ejecución (medidas descritas en la Sección 5.1.3). Los resultados obtenidos se analizan y discuten desde diferentes perspectivas a continuación.

5.2.1 Desempeño general

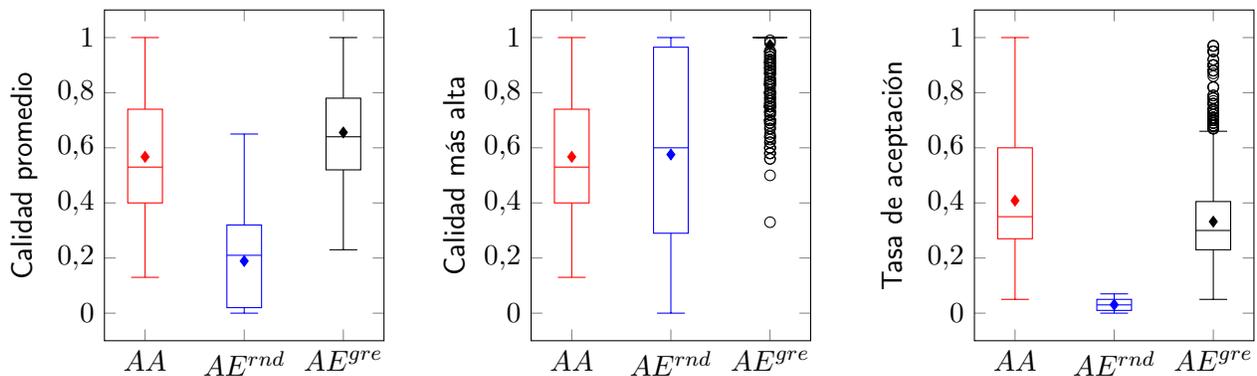
Los resultados obtenidos durante la evaluación de los enfoques AA , AE^{gre} , y AE^{rnd} se presentan de manera resumida en la Figura 5.1. Como se puede observar, nuestro algoritmo AE^{gre} es el método que en general obtiene los mejores resultados en nuestros experimentos, ofreciendo el mejor compromiso para el conjunto de medidas de desempeño utilizadas. En las tres categorías de problemas, AE^{gre} supera a los otros dos algoritmos, o al menos obtiene resultados comparables, en términos de calidad de solución promedio, calidad de solución más alta, y tasa de aceptación.

El método AA exhibe un desempeño más bajo para las instancias pequeñas en comparación con los enfoques evolutivos AE^{gre} y AE^{rnd} . Sin embargo, enfocándonos en las medidas de calidad de solución promedio y tasa de aceptación, es posible ver en la Figura 5.1 que AA claramente supera a AE^{rnd} en los escenarios medianos y grandes (con una diferencia estadísticamente significativa, como se muestra en la Sección 5.2.5), compitiendo estrechamente con AE^{gre} .³ Como se describe en la Sección 4.3, el método AA es el fundamento de la rutina de inicialización utilizada por AE^{gre} . El hecho de que AA puede producir, por sí solo, resultados satisfactorios, confirma que nuestra rutina de inicialización especializada proporciona al algoritmo AE^{gre} una clara ventaja competitiva y explica, hasta cierto punto, el éxito de este método en nuestros experimentos.

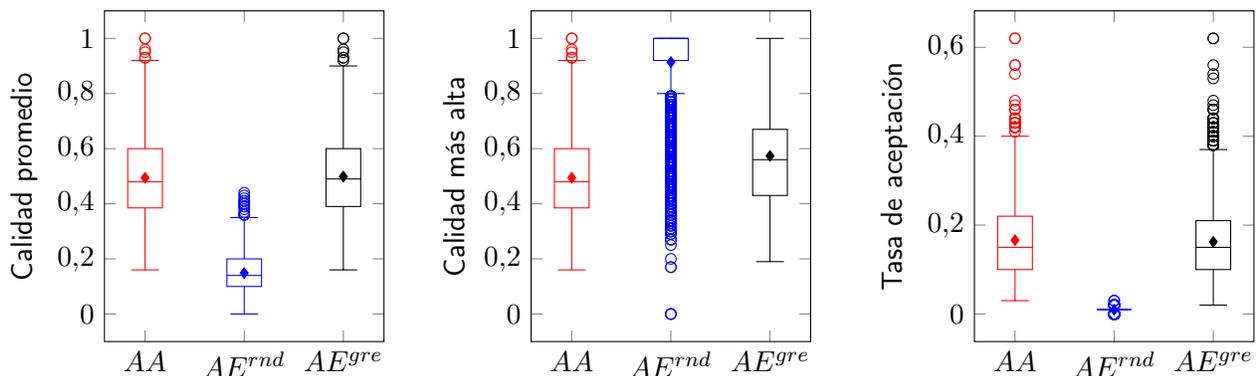
³Como se ha mencionado previamente en la Sección 5.1.3, AA es un método determinista. Por lo tanto, la calidad de solución promedio es la misma que la calidad de solución más alta para este algoritmo.



(a) Instancias de tamaño pequeño



(b) Instancias de tamaño mediano



(c) Instancias de tamaño grande

Figura 5.1: Resumen de los resultados obtenidos por los enfoques AA , AE^{gre} y AE^{rnd} . Los resultados se muestran por separado para instancias de tamaño pequeño (a), instancias de tamaño mediano (b) e instancias de tamaño grande (c). Para cada una de las categorías de problemas, los resultados se evalúan en términos de la calidad de solución promedio (izquierda), la calidad de solución más alta (centro), y la tasa promedio de aceptación de peticiones (derecha).

A pesar de los resultados prometedores obtenidos para las instancias pequeñas, el algoritmo AE^{rnd} obtiene los valores más bajos de las medidas de calidad de solución promedio y tasa de aceptación para los escenarios de tamaño mediano y grande. Este decremento en el desempeño de AE^{rnd} es estadísticamente significativo al compararse con AA y AE^{gre} (ver Sección 5.2.5). Dado que AE^{rnd} comienza con una población de soluciones aleatorias, que en la gran mayoría de los casos no son factibles, el algoritmo requiere invertir una cantidad significativa de esfuerzo simplemente para lograr descubrir la región factible. Sin embargo, a pesar de tal esfuerzo, AE^{rnd} generalmente fracasa en su búsqueda por una solución factible (que satisfaga los requerimientos de al menos una de las peticiones de los vMNOs), como lo indican las bajas tasas de aceptación reportadas por este algoritmo. No obstante, es interesante apreciar en la Figura 5.1 que para las instancias grandes AE^{rnd} logra superar a nuestras dos propuestas en términos de la calidad de solución más alta observada. Esto sugiere que la naturaleza aleatoria, sin sesgo, de la rutina de inicialización usada por AE^{rnd} , tiene cierto mérito y permite que el algoritmo realice algunas ejecuciones exitosas para varios escenarios de prueba.

Para ilustrar mejor el desempeño de nuestro algoritmo AE^{gre} , a continuación se incluye un ejemplo de una instancia de prueba, así como la solución de mayor calidad producida por este algoritmo.

5.2.1.1. Ejemplo de una instancia de prueba y una solución encontrada

En esta sección se presenta un ejemplo de una instancia de prueba y una solución encontrada por el algoritmo AE^{gre} . Más concretamente, utilizamos una instancia de prueba de tamaño pequeño, que incluye un conjunto de peticiones de recursos, una red de sustrato con 10 nodos y un conjunto de rutas predefinidas. Finalmente, se presenta la mejor solución encontrada por el algoritmo AE^{gre} .

La Figura 5.2 ilustra la instancia de prueba de tamaño pequeño considerada, misma que involucra una red de sustrato con 10 nodos y 12 enlaces. En esta instancia, se seleccionaron 4 nodos como DUs y 3 nodos como CUs. El conjunto Q contiene 3 peticiones de recursos: G_1^v , G_2^v y G_3^v . Cada petición de recursos requiere una CU virtual y dos DUs virtuales. Para cada pareja CU-DU, se generó una única ruta ($k = 1$), de modo que se obtuvieron un total de 12 rutas para esta instancia de prueba.

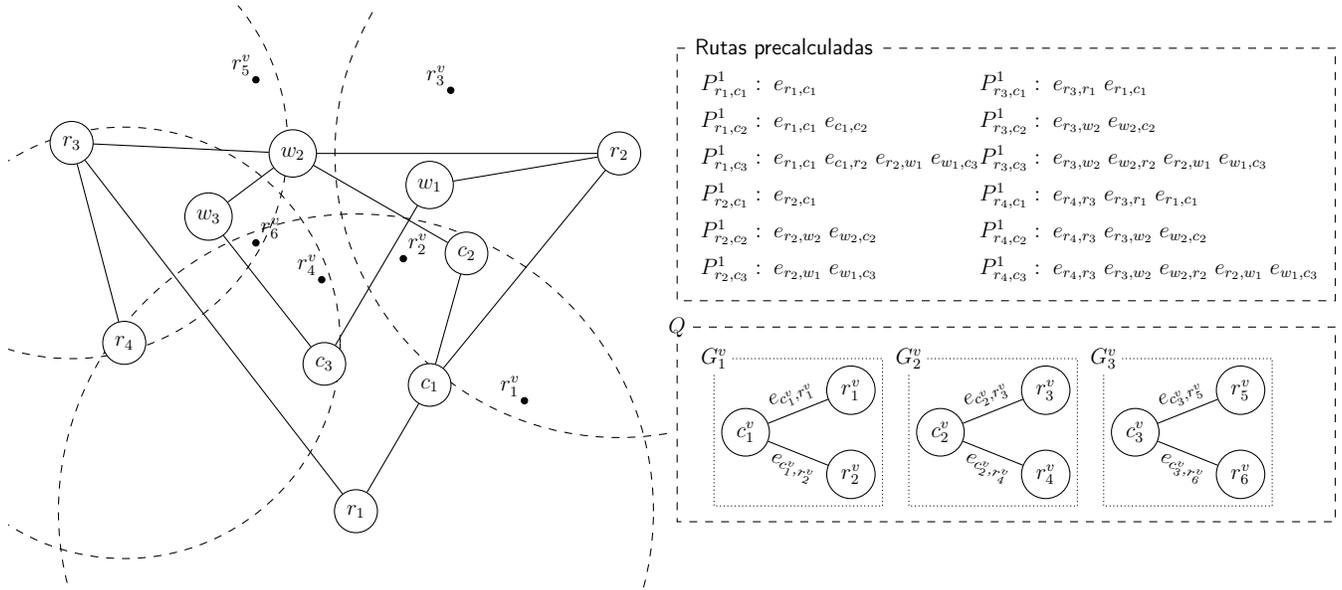


Figura 5.2: Ejemplo de una instancia de prueba de tamaño pequeño. La instancia se compone de una red física (grafo izquierdo), un conjunto de peticiones de recursos (Q , parte inferior derecha) y un conjunto de rutas precalculadas (P_{r_i,c_j}^1 , parte superior derecha).

La figura muestra las posiciones deseadas para todas las DUs virtuales, así como la cobertura de cada DU física. Como puede verse, las DUs virtuales r_1^v , r_2^v , r_4^v y r_6^v pueden asignarse a más de una DU física, mientras que las DUs virtuales r_5^v y r_3^v tienen solo una asignación posible de DU física.

La Figura 5.3 muestra la mejor solución obtenida por nuestro algoritmo AE^{gre} para esta instancia de prueba. Se puede observar en la figura que el algoritmo coloca todas las CUs virtuales en una única CU física (esta CU es la única que forma parte de la piscina de recursos). Las DUs virtuales r_1^v y r_6^v se asignaron a la DU física r_1 , los enlaces virtuales $e_{c_1^v,r_1^v}$ y $e_{c_3^v,r_6^v}$ fueron asignados a la ruta P_{r_1,c_1}^1 . Del mismo modo, las DUs virtuales r_2^v y r_3^v se asignaron a la DU física r_2 , y los enlaces virtuales $e_{c_1^v,r_2^v}$ y $e_{c_2^v,r_3^v}$ fueron asignados a la ruta P_{r_2,c_1}^1 . Las DUs virtuales r_4^v y r_5^v fueron asignados a las DUs r_4 y r_3 , respectivamente. Los enlaces virtuales $e_{c_2^v,r_4^v}$ y $e_{c_3^v,r_5^v}$ se asignaron a las rutas P_{r_4,c_1}^1 y P_{r_3,c_1}^1 respectivamente. Finalmente, a todas las DUs virtuales se les asignó la división funcional f_7 (intra PHY, opción 7-1).

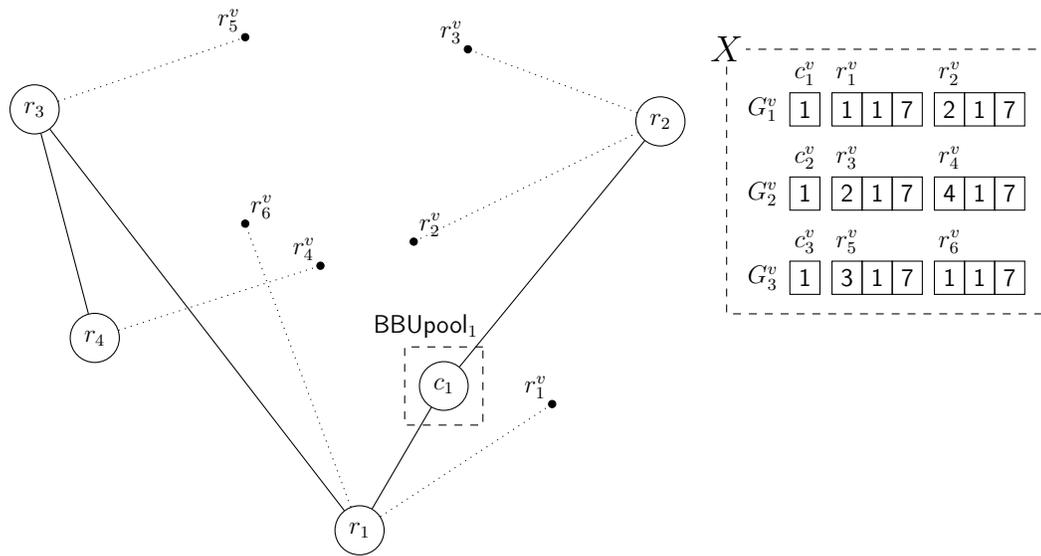


Figura 5.3: Mejor solución encontrada por el algoritmo AE^{gre} para la instancia de prueba de tamaño pequeño ilustrada en la Figura 5.2. Los elementos no asignados de la red física se han eliminado para resaltar mejor las asignaciones realizadas (solución candidata X). Cada DU virtual ha sido asignada a una DU física. Todas las CUs virtuales se asignaron a una única CU física, siendo esta la piscina de recursos BBU_{pool_1} . Cada DU virtual implementa la división funcional f_7 (intra PHY).

Esta asignación es posible debido a que en el momento en que se valida una asignación, nuestra solución primero valida que la división funcional que se va a asignar cumpla con las restricciones de mínimo ancho de banda y retardo máximo (expresiones (3.12) y (3.13), Sección 3.2.2, respectivamente). Si la división funcional cumple con los requisitos de calidad del vMNO, nuestra solución recorre cada enlace de la ruta a asignar. Para cada enlace, el ancho de banda disponible es usado para validar si se cumple la restricción de mínimo ancho de banda asignado a cada enlace (expresión (3.6), Sección 3.2.2). Del mismo modo, se verifica y garantiza el cumplimiento de la restricción de retardo para cada ruta (expresión (3.7), Sección 3.2.2). Si la asignación se determina como válida, se actualiza el ancho de banda disponible en cada enlace.

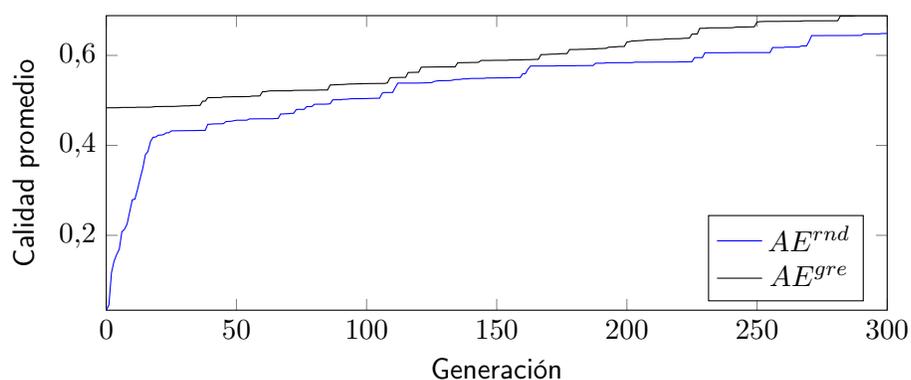
5.2.2 Comportamiento convergente

Esta sección analiza la capacidad de convergencia de los algoritmos AE^{gre} y AE^{rnd} , con la finalidad de explicar las diferencias observadas en la Sección 5.2.1 entre el desempeño de estos dos algoritmos. El análisis presentado en esta sección pretende también destacar el impacto que tiene el proceso de inicialización en el comportamiento de estos enfoques. La Figura 5.4 ilustra la evolución de la solución de más alta calidad observada en los algoritmos AE^{gre} y AE^{rnd} a lo largo del proceso evolutivo, considerándose un subconjunto de ejemplos de nuestros escenarios de prueba.

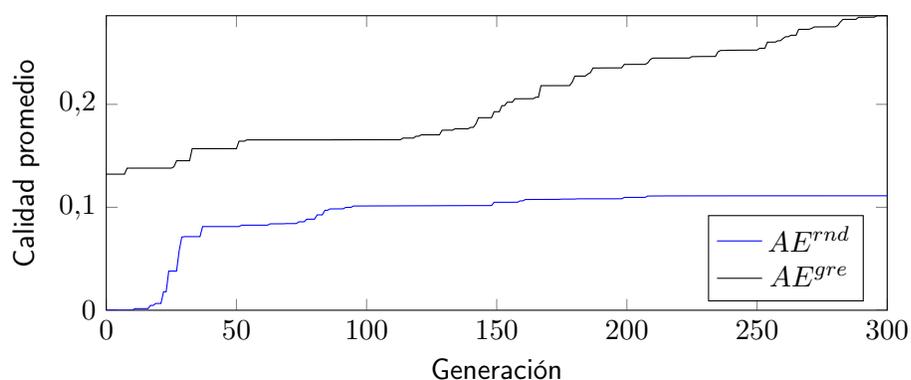
Los ejemplos mostrados en la Figura 5.4 confirman que los esfuerzos iniciales del algoritmo AE^{rnd} se invierten principalmente en la localización de la región factible, como se mencionó anteriormente en la Sección 5.2.1. Como se muestra en la figura, no es sino hasta después de que han pasado varias generaciones que el algoritmo AE^{rnd} logra identificar la región factible, y puede entonces obtener un progreso en términos de calidad de solución.

Como se describe en la Sección 5.1.2, el proceso de inicialización es el único factor distintivo entre los enfoques AE^{gre} y AE^{rnd} . Por lo tanto, nuestro análisis confirma que este componente juega un papel fundamental y contribuye de manera importante al desempeño de nuestro algoritmo AE^{gre} . A partir de la Figura 5.4, es posible observar que nuestra rutina de inicialización especializada es efectiva y logra incluir en la población inicial de nuestro algoritmo AE^{gre} soluciones factibles de buena calidad. De esta manera, AE^{gre} inicia el proceso evolutivo con una clara ventaja significativa con respecto al método de referencia AE^{rnd} (ver las notables diferencias entre el desempeño de las poblaciones iniciales para estos dos métodos, en la generación 0).

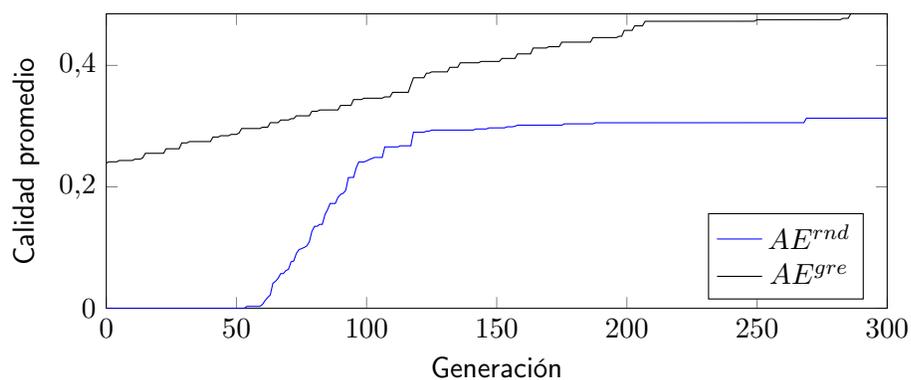
Vale la pena señalar, finalmente, que aunque la inicialización es un componente clave, no es el único factor que contribuye al desempeño favorable del algoritmo AE^{gre} . Las curvas de convergencia que se muestran en la Figura 5.4 ilustran la capacidad del método para evolucionar gradualmente las soluciones a lo largo de las generaciones. Si bien la inicialización es responsable de establecer un



(a) Ejemplo de instancia de tamaño pequeño



(b) Ejemplo de instancia de tamaño mediano



(c) Ejemplo de instancia de tamaño grande

Figura 5.4: Gráficas de convergencia de los algoritmos AE^{gre} y AE^{rnd} . Las gráficas muestran la calidad de solución más alta observada en cada generación, promediando los resultados de todas las ejecuciones independientes realizadas. Los resultados se ejemplifican para una instancia de tamaño pequeño (a), mediano (b), y grande (c).

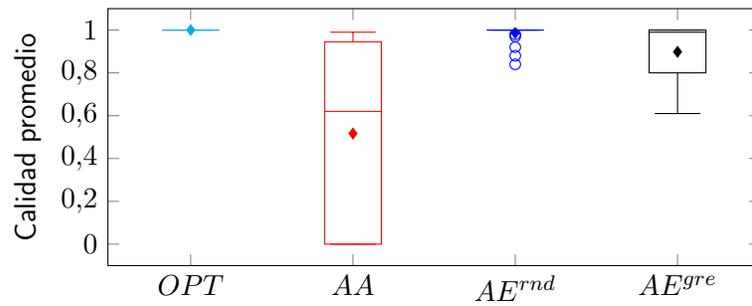
buen punto de partida para la búsqueda, el proceso evolutivo en sí es el responsable de capitalizar en dicho estado inicial tan favorable para alcanzar una solución final de alta calidad.

5.2.3 Capacidad de producir soluciones cercanas al óptimo

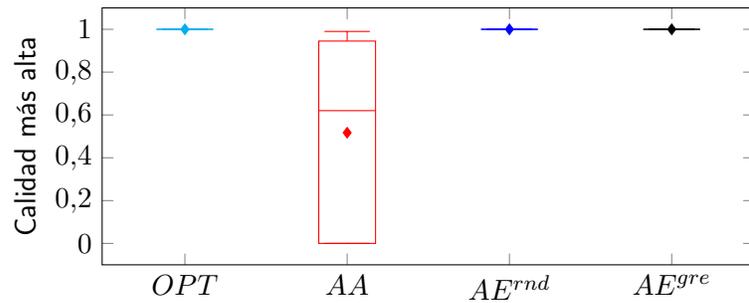
Los resultados presentados anteriormente permiten evaluar y comparar los algoritmos estudiados entre ellos, en función de su desempeño relativo. Sin embargo, debido a la naturaleza heurística de estos enfoques, es importante analizar qué tan cerca se encuentran las soluciones producidas por estos algoritmos con respecto la solución óptima de las instancias de prueba consideradas.

El principal desafío al abordar la pregunta anterior radica en el hecho de que las soluciones óptimas para nuestras instancias de prueba (creadas aleatoriamente) son desconocidas. De manera aún más importante, tales soluciones óptimas son computacionalmente costosas de calcular. Por lo tanto, nuestro esfuerzo para hacer frente a esta pregunta se centra necesariamente en un subconjunto reducido de instancias de tamaño pequeño. Para este subconjunto de instancias pequeñas, la solución óptima se ha determinado mediante la enumeración sistemática del espacio de soluciones, empleando el algoritmo de búsqueda exhaustiva descrito en la Sección 4.1. Es preciso resaltar que este análisis se limita a instancias muy pequeñas, debido al crecimiento exponencial del espacio de búsqueda para este problema (como se analiza a detalle en la Sección 3.3).

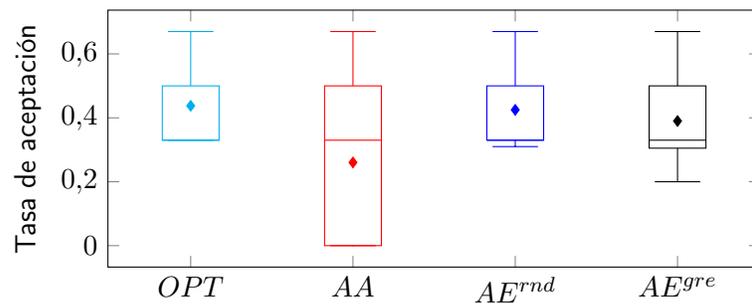
Los resultados de este análisis se muestran en la Figura 5.5. Como se puede observar en la figura, tanto AE^{gre} como AE^{rnd} obtienen valores para las medidas de calidad de solución y tasa de aceptación que en la mayoría de los casos son comparables con los de la solución óptima. Esto sugiere que los dos algoritmos pueden localizar el óptimo o, por lo menos, logran producir una aproximación cercana. A diferencia de los resultados reportados anteriormente en la Figura 5.1 para la colección completa de instancias de prueba, la Figura 5.5 muestra que el algoritmo de referencia AE^{rnd} obtiene resultados ligeramente mejores que el algoritmo propuesto AE^{gre} para el subconjunto particular de instancias considerado en esta prueba.



(a) Calidad promedio de las soluciones



(b) Calidad de la mejor solución



(c) Tasa de aceptación promedio

Figura 5.5: Comparación de los resultados obtenidos por los enfoques AA , AE^{gre} y AE^{rnd} con respecto a la solución óptima (OPT). Se considera un subconjunto de 30 instancias de prueba de tamaño pequeño. Los resultados se evalúan en términos de la calidad de solución promedio (a), la calidad de solución más alta observada (b), y la tasa de aceptación promedio (c).

Finalmente, nuestro algoritmo AA logra aproximar correctamente la solución óptima en algunos casos. En algunos otros casos, sin embargo, AA no logra identificar una solución factible (como lo indican las bajas tasas de aceptación de este algoritmo), por lo que el desempeño general de AA en este experimento se considera bajo, en comparación con los enfoques evolutivos AE^{gre} y AE^{rnd} .

5.2.4 Eficiencia computacional

Sin duda, un aspecto fundamental que determina la aplicabilidad de un algoritmo en la práctica es su eficiencia computacional. Si bien los resultados presentados hasta este momento revelan que nuestros métodos son capaces de producir soluciones de alta calidad, en esta sección nos concentraremos en investigar la eficiencia con la que se logran obtener dichas soluciones.

Los tiempos de ejecución que se midieron durante nuestros experimentos se reportan en la Figura 5.6. Por un lado, es evidente a partir de los resultados mostrados en la sub-figura (a) que el tamaño del espacio de búsqueda hace que una exploración exhaustiva sea poco práctica, incluso para instancias muy pequeñas del problema. Aunque este análisis se centra en un subconjunto de nuestros escenarios de prueba más pequeños, como se discutió en la Sección 5.2.3, es posible observar que los tiempos de ejecución del enfoque exhaustivo son relativamente altos. Más importante aún, estos tiempos crecerían exponencialmente con el aumento en el tamaño de las instancias.

Por otro lado, la Figura 5.6 permite resaltar la eficiencia de nuestro algoritmo AA , que produce soluciones en tiempos del orden de milisegundos, incluso para los casos de prueba más grandes. La figura también muestra que el uso de los enfoques evolutivos, AE^{gre} y AE^{rnd} , provoca un aumento significativo en el tiempo de ejecución con respecto al algoritmo AA (ver resultados del análisis de significancia estadística en la Sección 5.2.5). Sin embargo, los tiempos de ejecución que reportan los enfoques AE^{gre} y AE^{rnd} son del orden de segundos en la mayoría de los casos. Por lo tanto, tal incremento en los tiempos de ejecución puede considerarse aceptable, principalmente si se toman en cuenta las ventajas significativas que el algoritmo AE^{gre} aporta en términos de producir soluciones de mayor calidad, como se observó en la Sección 5.2.1. Vale la pena destacar que este trabajo de tesis sigue un enfoque estático (fuera de línea) para abordar el problema de SCRDF (ver Sección 2.1.2.1), de manera que tiempos de ejecución del orden de segundos (o incluso mayores) no comprometen la viabilidad y aplicabilidad de nuestro algoritmo AE^{gre} .

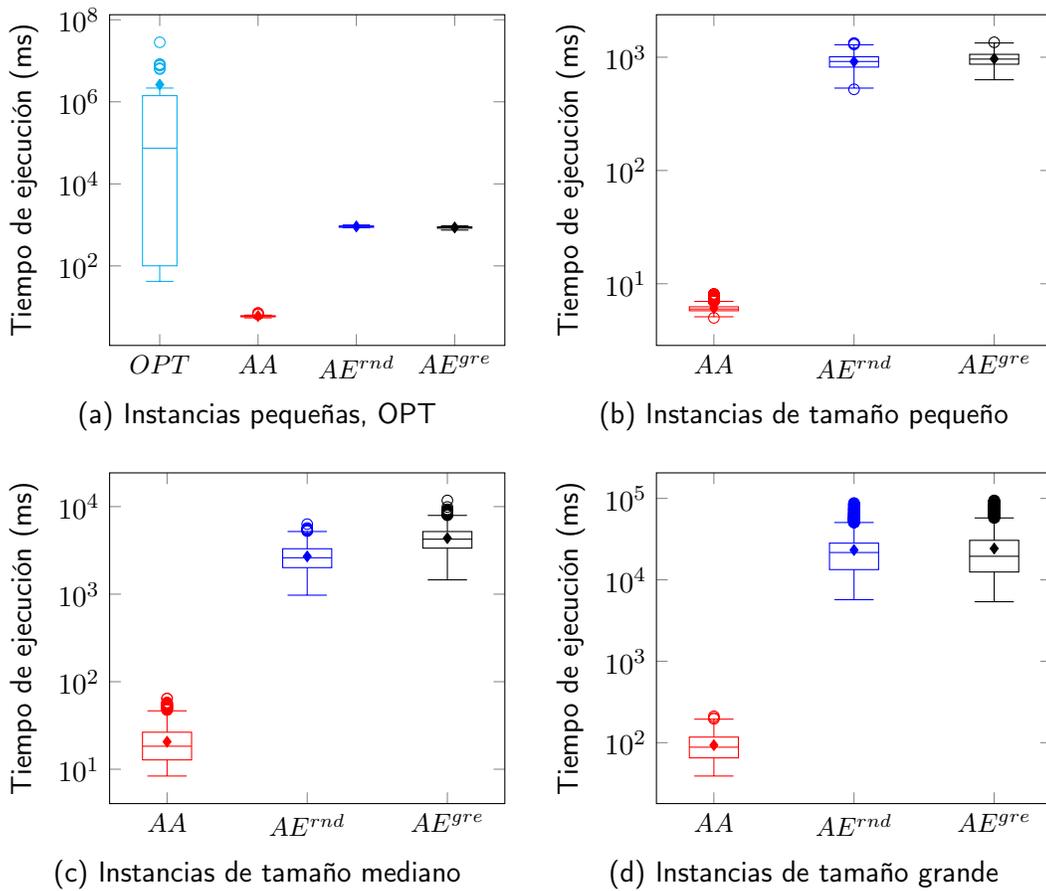


Figura 5.6: Tiempos de ejecución registrados durante los experimentos de este trabajo de tesis. La gráfica en (a) muestra los tiempos obtenidos durante el análisis de la Sección 5.2.3, incluyendo los tiempos necesarios para encontrar el óptimo mediante la enumeración exhaustiva del espacio de búsqueda (*OPT*). Las gráficas (b), (c) y (d) muestran los tiempos obtenidos durante el análisis de la Sección 5.2.1, considerando el conjunto completo de instancias de tamaño pequeño, mediano y grande, respectivamente. Es preciso notar que los tiempos se expresan en milisegundos (ms) y se muestran en escala logarítmica.

Finalmente, gracias al bajo costo computacional del algoritmo *AA*, el uso repetido de este algoritmo dentro de nuestra inicialización especializada no ha causado un impacto notable en el tiempo de ejecución del algoritmo *AE^{gre}*. A partir de la Figura 5.6, solamente se pueden observar diferencias menores en el tiempo de ejecución del algoritmo *AE^{gre}* con respecto al uso de una inicialización aleatoria en el enfoque *AE^{rnd}*.

Tabla 5.5: Resumen de los resultados del análisis de significancia estadística realizado mediante el uso de la prueba de Mann-Whitney. La medida de desempeño M1 corresponde a la calidad promedio de las soluciones obtenidas por cada algoritmo. La medida de desempeño M2 corresponde a la tasa de aceptación promedio obtenida por cada algoritmo. La medida de desempeño M3 corresponde al tiempo de ejecución de cada algoritmo. Cuando existe una diferencia estadísticamente significativa entre una pareja específica de algoritmos, para alguna de las tres medidas de desempeño, la celda correspondiente indica dicho resultado con el símbolo \checkmark . En caso contrario, la celda indica que no existe una diferencia estadísticamente significativa utilizando el símbolo \times .

Instancias	$AA - AE^{rnd}$			$AA - AE^{gre}$			$AE^{rnd} - AE^{gre}$		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
Pequeñas	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\times	\checkmark	\checkmark
Medianas	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
Grandes	\checkmark	\checkmark	\checkmark	\times	\times	\checkmark	\checkmark	\checkmark	\times

5.2.5 Análisis estadístico

Tal como se describe en la Sección 5.1.3.4, se realizó un análisis estadístico para investigar si las diferencias observadas entre los algoritmos estudiados son significativas. Las pruebas de Kruskal-Wallis realizadas nos indican que se observa una diferencia estadísticamente significativa ($\text{valor-}p \leq 0,05$) entre los resultados obtenidos por el conjunto de algoritmos evaluados, para cada una de las medidas de desempeño utilizadas. Por tal motivo, se utilizó la prueba de Mann-Whitney (con corrección de Bonferroni) para determinar entre qué parejas específicas de algoritmos se presentaron tales diferencias significativas. A manera de resumen, la Tabla 5.5 presenta los resultados obtenidos durante el análisis de las muestras provenientes de cada pareja de algoritmos, para las diferentes medidas de desempeño.

Por un lado, a partir de la Tabla 5.5 se logra observar que en la mayoría de los casos se encontró una diferencia estadísticamente significativa entre los resultados obtenidos por cada pareja de algoritmos, para las medidas de desempeño consideradas. Por otro lado, es posible observar las

siguientes excepciones: no hay una diferencia estadísticamente significativa entre la calidad promedio de las soluciones obtenidas por los algoritmos AE^{rnd} y AE^{gre} para las instancias pequeñas. Del mismo modo, para las instancias grandes las diferencias observadas entre nuestros algoritmos AA y AE^{gre} no son estadísticamente significativas en términos de calidad de solución y tasa de aceptación. Finalmente, tampoco existe una diferencia estadísticamente significativa en el tiempo de ejecución de los algoritmos AE^{rnd} y AE^{gre} para las instancias grandes.

5.3 Resumen y conclusiones

En este capítulo se analizaron los resultados de una serie de experimentos, cuyo objetivo es la validación y comparación de nuestras propuestas algorítmicas presentadas en el Capítulo 4. Nuestros experimentos evaluaron la capacidad de los algoritmos propuestos para producir soluciones de alta calidad, así como también aspectos relacionados con su comportamiento y eficiencia computacional. De manera general, estos experimentos han permitido demostrar la efectividad de nuestros algoritmos. Los resultados obtenidos indican que nuestros algoritmos ofrecen distintos compromisos entre la calidad de las soluciones que producen y el tiempo de ejecución que requieren para hacerlo.

Por un lado, los resultados de nuestros experimentos confirman que un algoritmo de búsqueda exhaustiva resulta prohibitivo para instancias razonablemente realistas del problema. Esto se debe al crecimiento exponencial del espacio de búsqueda, que este tipo de algoritmos debe enumerar para lograr identificar una solución óptima. Por otro lado, nuestro algoritmo avaro logra mejorar significativamente los tiempos de ejecución, pero también fue posible observar que este algoritmo no suele proporcionar los mejores resultados en términos de calidad de solución.

Finalmente, nuestros experimentos revelan que el algoritmo evolutivo propuesto logra un balance más adecuado entre calidad de solución y eficiencia computacional. Se observó que nuestro algoritmo evolutivo genera soluciones de mayor calidad, pero registrando un incremento significativo en los tiempos de ejecución en comparación con el algoritmo avaro. Sin embargo, los tiempos registrados

por el algoritmo evolutivo fueron aún relativamente bajos, y se consideran aceptables dado que este trabajo se centra en una versión estática (fuera de línea) del problema abordado (ver Sección 2.1.2.1).

Los resultados obtenidos son alentadores y abren la posibilidad de considerar, en nuestro trabajo futuro, el problema de SCRDF siguiendo un enfoque dinámico (en línea). También, consideramos que el diseño de nuevos operadores genéticos especializados para el algoritmo evolutivo propuesto representa una área de oportunidad relevante. Tales operadores explotarían el conocimiento del problema y permitirían aumentar la tasa con la que se generan soluciones candidatas factibles y de buena calidad.

6

Conclusiones y trabajo futuro

En los capítulos anteriores se diseñaron, implementaron, ejecutaron experimentos y se presentaron los resultados obtenidos. En este capítulo partimos del análisis de dichos resultados y describimos, en la Sección 6.1, las conclusiones a las que se llegó en este trabajo de tesis. Finalmente, en la Sección 6.2 se describen las áreas de oportunidad para futuros desarrollos que puedan partir de la conclusión de este trabajo de tesis.

6.1 Conclusiones

En este trabajo de tesis se ha abordado el problema de seleccionar conjuntamente rutas de *midhaul* y determinar las divisiones funcionales apropiadas sobre arquitecturas C-RAN, para satisfacer los requisitos de calidad de servicio de los vMNOs. Hemos demostrado que este es un problema complejo, que se abordó considerando holísticamente los siguientes dos subproblemas : (1) determinar la ubicación más apropiada de DUs y CUs que componen la estación base, así como la distribución de la división funcional óptima entre ellos; y (2) seleccionar las mejores rutas para el tráfico proveniente

de cada una de las DUs. Hasta donde sabemos, este es el primer trabajo que aborda los problemas mencionados de manera combinada. De hecho, existen pocos trabajos que los hayan considerado aún por separado.

Hemos definido un modelo del problema para abordar dicho desafío, empleando un modelado de incrustación de redes virtuales, que soporta múltiples solicitudes de vMNOs, heterogeneidad en el uso de recursos, diversidad de asignaciones de DUs y CUs, selección de rutas con reconocimiento de restricciones y divisiones funcionales. Se han propuesto dos soluciones algorítmicas que permiten afrontar el problema: un enfoque avaro y una solución evolutiva, que se ha implementado con dos mecanismos de inicialización, un enfoque evolutivo de referencia con inicialización aleatoria y una segunda versión mejorada con un enfoque de inicialización inteligente. Después de una validación exhaustiva, que se ha realizado usando diferentes topologías de red, hemos demostrado que las técnicas propuestas proporcionan compromisos adecuados entre la calidad de las soluciones, la tasa de aceptación y el tiempo de ejecución. También hemos comparado las soluciones obtenidas por nuestros métodos con respecto a la solución óptima para un conjunto de instancias pequeñas, y los resultados fueron satisfactorios.

Por lo anterior, con base en los resultados obtenidos se concluye que se ha cumplido el objetivo general de este trabajo de tesis, el cual consiste en “Abordar el problema de Selección Conjunta de Rutas y Divisiones Funcionales para redes *midhaul* en arquitecturas C-RAN de manera integral, y demostrar la viabilidad de obtener soluciones de asignación eficiente de los recursos de la red mediante la distribución adecuada de funciones”. En la Tabla 6.1 se muestra un resumen del cumplimiento de los objetivos particulares del presente trabajo de tesis.

Tabla 6.1: Resumen del cumplimiento de los objetivos particulares de este trabajo de tesis.

Objetivo	Descripción del cumplimiento del objetivo
Definir un modelo del problema de Selección Conjunta de Rutas y Divisiones Funcionales para arquitecturas C-RAN, que permita formalizar la función objetivo y las restricciones del problema.	La formalización del problema se ha documentado en el Capítulo 3. Esta formalización consiste en la descripción del modelo (Sección 3.1), la formulación del problema de optimización (Sección 3.2), compuesto de la función objetivo y de las restricciones del problema. Se presentó también un análisis que demuestra que el problema de optimización planteado involucra un espacio de búsqueda exponencial.
Diseñar e implementar una técnica heurística de optimización y una técnica de inteligencia computacional, basada en algoritmos evolutivos, que permitan obtener de manera eficiente soluciones satisfactorias al problema de Selección Conjunta de Rutas y Divisiones Funcionales para arquitecturas C-RAN.	Los diseños de los dos algoritmos propuestos se presentan en el Capítulo 4. El diseño del algoritmo heurístico corresponde al algoritmo avaro, descrito en la Sección 4.2, y el enfoque de inteligencia computacional corresponde al algoritmo evolutivo descrito en la Sección 4.3.
Validar las técnicas propuestas y demostrar la viabilidad de proveer soluciones computacionales que permitan definir compromisos entre eficiencia (tiempos de ejecución) y la capacidad de identificar soluciones (configuraciones de recursos) de alta calidad.	La evaluación de los algoritmos propuestos, documentada en el Capítulo 5, ha permitido la validación de su capacidad para producir soluciones satisfactorias. El conjunto de soluciones algorítmicas desarrolladas en este trabajo de tesis permite definir compromisos de acuerdo con las necesidades de los administradores de red. Por un lado, se demuestra que un algoritmo exhaustivo logra encontrar soluciones óptimas a costa de un tiempo de ejecución elevado, que puede ser prohibitivo para instancias realistas del problema. Por otro lado, el algoritmo avaro logra mejorar los tiempos de ejecución, con un decremento en la calidad de las soluciones. Finalmente, finalmente el algoritmo evolutivo logra un balance más adecuado entre el tiempo de ejecución y la calidad de las soluciones.

6.2 Trabajo futuro

El modelo, los diseños algorítmicos y las instancias de prueba presentadas en este trabajo de tesis, representan los primeros esfuerzos para abordar el problema de selección conjunta de rutas y divisiones funcionales sobre arquitecturas C-RAN. En este punto, los resultados obtenidos por nuestras propuestas algorítmicas en términos de calidad de soluciones, tasas de aceptación y tiempos de ejecución, son alentadores y nos motivan a considerar la versión dinámica (en línea) de este complejo problema como parte de nuestro trabajo futuro.

Desde el punto de vista algorítmico, el trabajo futuro podría ser dedicado a diseñar operadores genéticos novedosos que exploten el conocimiento del problema para mejorar la generación de soluciones viables de mayor calidad. Finalmente, aprovecharemos este marco para analizar las técnicas de segmentación (slicing) en la temática de redes posteriores a la quinta generación (del inglés, Beyond 5G Networks), donde la gestión adecuada de los recursos será más relevante.



Resultados obtenidos del proceso de ajuste de parámetros

Como se menciona en la Sección 4.3, los operadores de variación (cruza, mutación y selección), sirven para guiar el proceso de búsqueda del algoritmo genético. Estos operadores cuentan con algunos parámetros que permiten controlar la exploración del espacio de búsqueda y la generación de soluciones. Encontrar los mejores parámetros para estos operadores no siempre es una tarea simple, ya que el valor de estos parámetros dependerá de la interacción con los demás operadores y del problema abordado. Por esta razón, realizar un proceso de ajuste de parámetros permitirá encontrar aquellos valores que permitan mejorar el desempeño del algoritmo evolutivo propuesto.

En este trabajo de tesis se optó por utilizar tres posibles valores para cada uno de los parámetros de los operadores de: mutación (p_m), cruza (p_c) y selección p_s . Los valores del operador de mutación (probabilidad de mutación) dependen de la cantidad de celdas (ℓ) de la matriz que representa a una solución para una instancia del problema, los valores para el operador de mutación corresponden a:

Tabla A.1: Valores empleados en el proceso de ajuste de parámetros. La columna Grupo corresponde al identificador del grupo de valores para los parámetros de probabilidad de cruce, probabilidad de mutación y los valores de probabilidad de comparación de soluciones no-factibles usando la función objetivo (0,425, 0,45 y 0,475).

Grupo	Probabilidad de cruce	Probabilidad de mutación	Grupo	Probabilidad de cruce	Probabilidad de mutación	Grupo	Probabilidad de cruce	Probabilidad de mutación
P1	0.5	$1/\ell$	P4	0.7	$1/\ell$	P7	0.9	$1/\ell$
P2	0.5	$2/\ell$	P5	0.7	$2/\ell$	P8	0.9	$2/\ell$
P3	0.5	$3/\ell$	P6	0.7	$3/\ell$	P9	0.9	$3/\ell$

$1/\ell$, $2/\ell$ y $3/\ell$. Esto quiere decir que de manera aleatoria se altera (aproximadamente) una, dos o tres celdas de la matriz en el proceso de mutación. Los valores para el operador de cruce (probabilidad de cruce), sirven para ajustar el grado de semejanza de las soluciones (generadas) respecto de los padres seleccionados. Los valores para el operador de cruce corresponden a: 0,5, 0,7 y 0,9.

Finalmente, el valor para el parámetro del operador de selección corresponde a la probabilidad de comparar soluciones no-factibles del espacio de búsqueda, utilizando el valor de la función objetivo. Los valores para este operador corresponden a: 0,425, 0,45 y 0,475. Con estos valores, se obtienen 27 posibles combinaciones de parámetros (ver Tabla A) para cada versión del algoritmo genético (algoritmo genético usando inicialización aleatoria, algoritmo genético usando inicialización avara), estas 27 configuraciones se evaluaron usando 300 instancias del problema (100 instancias seleccionadas de manera aleatoria por cada uno de los escenarios).

A.1 Resultados del ajuste de parámetros para los escenarios pequeños

En la Figura A.1, se logra apreciar que los mejores resultados para los dos algoritmos corresponden a P_3 ($p_c = 0,5$ y $p_m = 3/\ell$), para la inicialización aleatoria, el mejor resultado se obtiene con $p_s = 0,475$, y para la inicialización avara con $p_s = 0,425$.

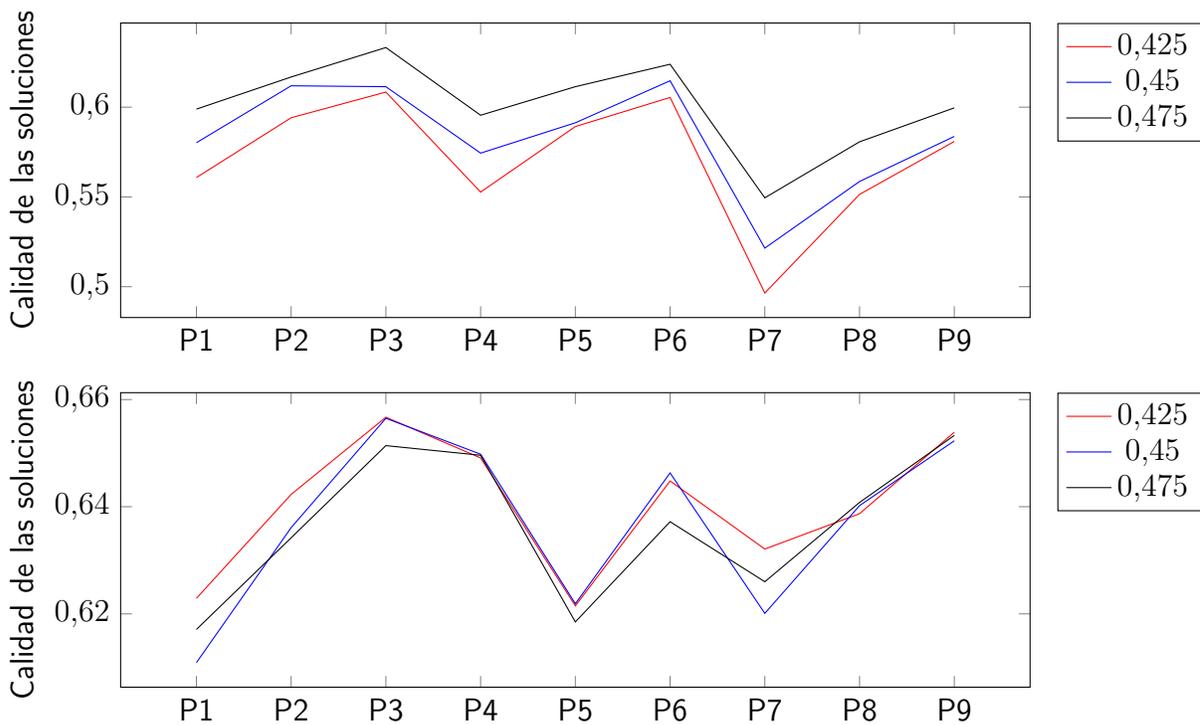


Figura A.1: Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias pequeñas seleccionadas. La primera gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando inicialización aleatoria. La segunda gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando inicialización avara.

A.2 Resultados obtenidos del ajuste de parámetros para los escenarios medianos.

En la Figura A.2, se logra apreciar que la calidad de las soluciones obtenidas por los algoritmos genéticos que utiliza la inicialización aleatoria, es peor en comparación con las soluciones obtenidas por los algoritmos genéticos que emplea la representación completa. También se logra apreciar que los mejores resultados para los dos algoritmos corresponden a P_4 ($p_c = 0,7$ y $p_m = 1/\ell$); para la

inicialización aleatoria, el mejor resultado se obtiene con $p_s = 0,475$, y para la inicialización avara con $p_s = 0,425$.

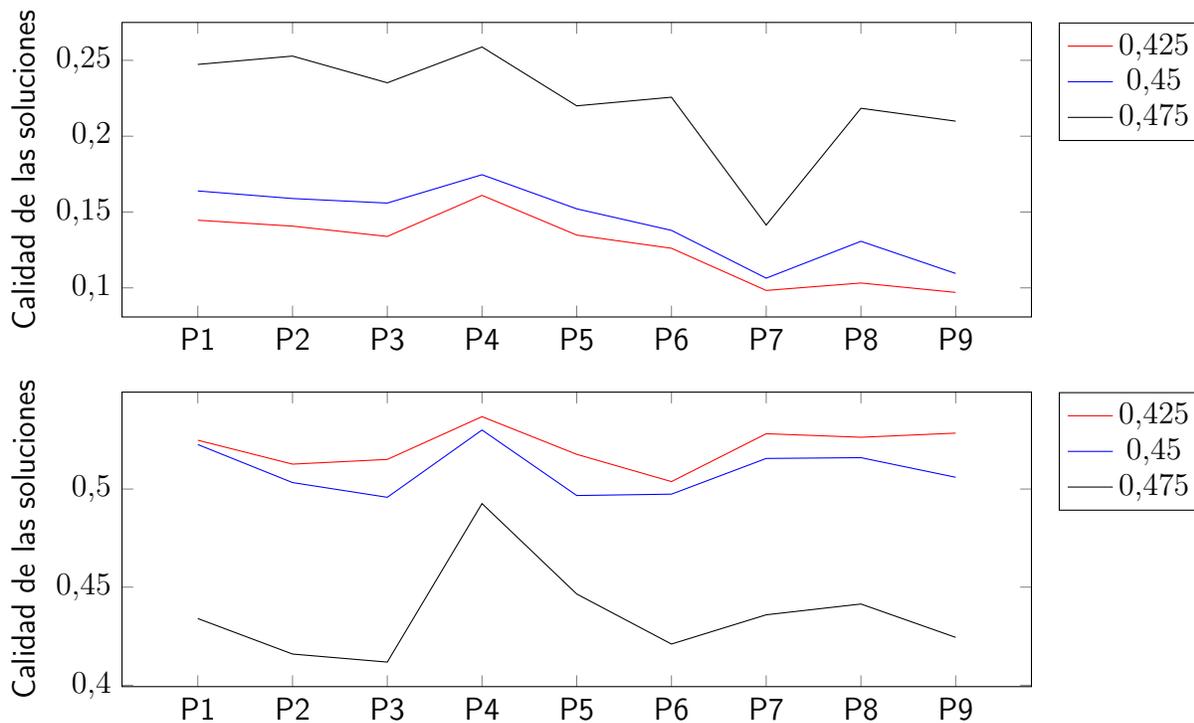


Figura A.2: Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias medianas seleccionadas. La primer gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando la inicialización aleatoria. La segunda gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando inicialización avara.

A.3 Resultados obtenidos para el ajuste de parámetros de los escenarios grandes.

En la Figura A.3, se logra apreciar que la calidad de las soluciones obtenidas por los algoritmos genéticos que utiliza la inicialización aleatoria, es peor en comparación con las soluciones obtenidas

por los algoritmos genéticos que emplea la inicialización avara. Se logra apreciar que los mejores resultados para los dos algoritmos corresponden a P_4 ($p_c = 0,7$ y $p_m = 1/\ell$). Similar a los resultados obtenidos anteriormente, para la inicialización avara el mejor resultado se obtiene con $p_s = 0,475$ y para la inicialización avara con $p_s = 0,425$.

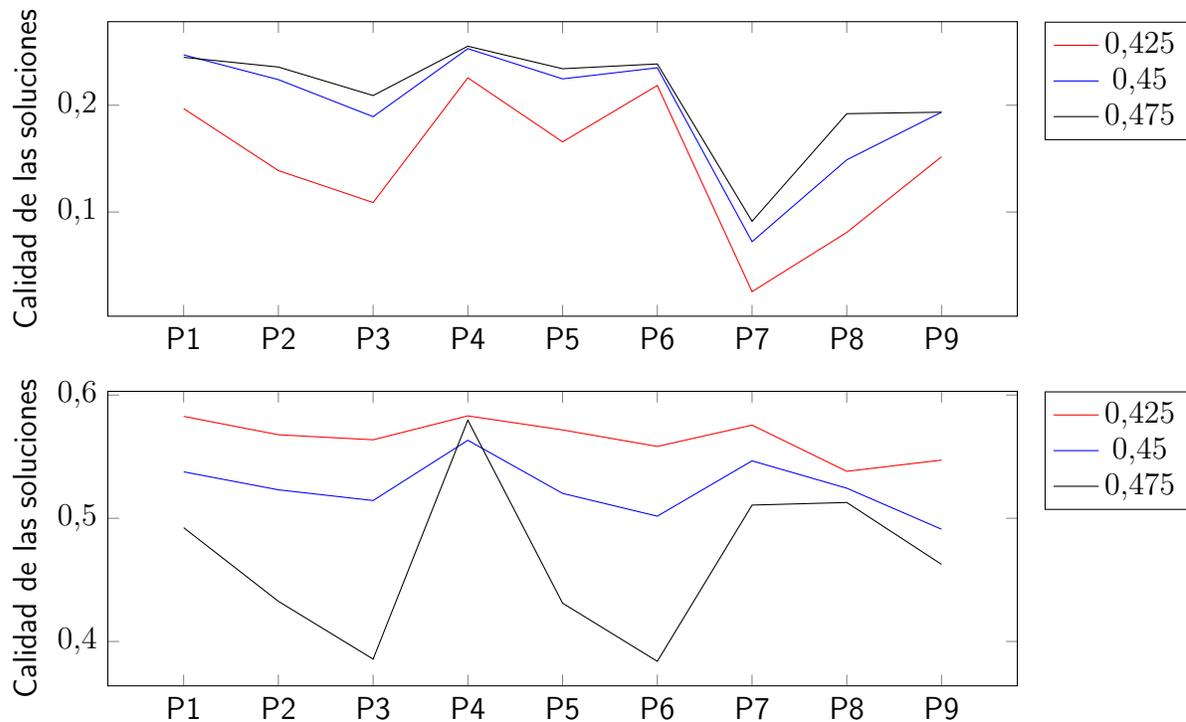


Figura A.3: Resultados obtenidos del ajuste de parámetros para la calidad de las soluciones obtenidas por los algoritmos genéticos, para las instancias grandes seleccionadas. La primera gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando la inicialización aleatoria. La segunda gráfica corresponde a los resultados obtenidos por el algoritmo genético empleando la inicialización avara.

Bibliografía

- [1] Cisco Visual Networking Index. «Global mobile data traffic forecast update, 2017–2022». En: *Cisco White paper* (2019).
- [2] Mamta Agiwal, Abhishek Roy y Navrati Saxena. «Next generation 5G wireless networks: A comprehensive survey». En: *IEEE Communications Surveys & Tutorials* 18.3 (2016), págs. 1617-1655.
- [3] Conor Sexton y col. «5G: Adaptable networks enabled by versatile radio access technologies». En: *IEEE Communications Surveys & Tutorials* 19.2 (2017), págs. 688-720.
- [4] 3GPP TR 38.801. *Study on new radio access technology: Radio access architecture and interfaces*. Technical report. 3GPP Organizational Partners, 2017.
- [5] L. M. P. Larsen, A. Checko y H. L. Christiansen. «A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks». En: *IEEE Communications Surveys Tutorials* 21.1 (ene. de 2019), págs. 146-172. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2868805.
- [6] Kulin Chen y Run Duan. «C-RAN: the road towards green RAN». En: *China Mobile Research Institute, White Paper 2* (2011), págs. 1-12.
- [7] Y. Lin y col. «Wireless network cloud: Architecture and system requirements». En: *IBM Journal of Research and Development* 54.1 (ene. de 2010), 4:1-4:12. ISSN: 0018-8646. DOI: 10.1147/JRD.2009.2037680.
- [8] A. Maeder y col. «Towards a flexible functional split for cloud-RAN networks». En: *2014 European Conference on Networks and Communications (EuCNC)*. Jun. de 2014, págs. 1-5. DOI: 10.1109/EuCNC.2014.6882691.

- [9] Chih-Lin I y col. «Rethink fronthaul for soft RAN». En: *IEEE Communications Magazine* 53.9 (sep. de 2015), págs. 82-88. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7263350.
- [10] Orawan Tipmongkolsilp, Said Zaghloul y Admela Jukan. «The evolution of cellular backhaul technologies: Current issues and future trends». En: *IEEE communications surveys & tutorials* 13.1 (2010), págs. 97-113.
- [11] NGMN Alliance. «Small cell backhaul requirements». En: *white paper, June* (2012).
- [12] Amir H Jafari y col. «Small cell backhaul: challenges and prospective solutions». En: *EURASIP Journal on Wireless Communications and Networking* 2015.1 (2015), pág. 206.
- [13] Dominic Schulz y col. «Robust optical wireless link for the backhaul and fronthaul of small radio cells». En: *Journal of Lightwave technology* 34.6 (2016), págs. 1523-1532.
- [14] IEEE 1914 Working Group. *Next generation Fronthaul Interface*. URL: <http://sites.ieee.org/sagroups-1914/> (visitado 03-04-2019).
- [15] N. M. Mosharaf Kabir Chowdhury y R. Boutaba. «Network virtualization: state of the art and research challenges». En: *IEEE Communications Magazine* 47.7 (2009), págs. 20-26.
- [16] Rashid Mijumbi y col. «Network function virtualization: State-of-the-art and research challenges». En: *IEEE Communications surveys & tutorials* 18.1 (2015), págs. 236-262.
- [17] A. Garcia-Saavedra y col. «WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul». En: *IEEE Transactions on Mobile Computing* 17.10 (oct. de 2018), págs. 2452-2466. ISSN: 1536-1233. DOI: 10.1109/TMC.2018.2793859.
- [18] D. Dasgupta y Z. Michalewicz. *Evolutionary Algorithms in Engineering Applications*. Springer Berlin Heidelberg, 2014. ISBN: 9783662034248.
- [19] SeungJune Yi y col. *Radio Protocols for LTE and LTE-advanced*. John Wiley & Sons, 2012.
- [20] Small Cell Forum. «Small cell virtualization: Functional splits and use cases». En: (2015).

- [21] NM Mosharaf Kabir Chowdhury y Raouf Boutaba. «A survey of network virtualization». En: *Computer Networks* 54.5 (2010), págs. 862-876.
- [22] Aun Haider, Richard Potter y Akihiro Nakao. «Challenges in resource allocation in network virtualization». En: *20th ITC specialist seminar*. Vol. 18. 2009. ITC. 2009.
- [23] Yong Zhu y Mostafa H Ammar. «Algorithms for assigning substrate network resources to virtual network components.» En: *INFOCOM*. Vol. 1200. 2006. Citeseer. 2006, págs. 1-12.
- [24] S. G. Kolliopoulos y C. Stein. «Improved approximation algorithms for unsplittable flow problems». En: *Proceedings 38th Annual Symposium on Foundations of Computer Science*. 1997, págs. 426-436.
- [25] S Singiresu y col. *Engineering optimization: theory and practice*. John Wiley & Sons, 1996.
- [26] Richard Bellman. *Mathematical optimization techniques*. Univ of California Press, 1963.
- [27] Carlos A Coello. «An updated survey of GA-based multiobjective optimization techniques». En: *ACM Computing Surveys (CSUR)* 32.2 (2000), págs. 109-143.
- [28] Godfrey C Onwubolu y BV Babu. *New optimization techniques in engineering*. Vol. 141. Springer, 2013.
- [29] Thomas H. Cormen y col. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844, 9780262033848.
- [30] E. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009. ISBN: 978-0-470-27858-1.
- [31] C. Blum y A. Roli. «Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison». En: *ACM Computing Surveys* 35.3 (2003), págs. 268-308. ISSN: 0360-0300.
- [32] I. Rechenberg. *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.

- [33] H-P Schwefel. *Numerische optimierung von computer-modellen mittels der evolutionsstrategie*. Vol. 26. ISR. Birkhäuser, 1977.
- [34] Hans-Georg Beyer y Hans-Paul Schwefel. «Evolution Strategies –A Comprehensive Introduction». En: *Natural Computing* 1.1 (mayo de 2002), págs. 3-52. ISSN: 1567-7818.
- [35] L.J. Fogel, A.J. Owens y M.J. Walsh. *Artificial intelligence through simulated evolution*. Chichester, WS, UK: Wiley, 1966.
- [36] David B. Fogel y Lawrence J. Fogel. «An introduction to evolutionary programming». En: *Artificial Evolution*. Ed. por Jean-Marc Alliot y col. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, págs. 21-33.
- [37] J. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975.
- [38] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.
- [39] John R. Koza. *Genetic Programming*. Cambridge, MA, USA: MIT Press, 1992. ISBN: 9780262527910.
- [40] John R. Koza. *Genetic Programming II*. Cambridge, MA, USA: MIT Press, 1994. ISBN: 9780262111898.
- [41] A. E. Eiben y J. E. Smith. *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg, 2015. ISBN: 978-3-662-44874-8.
- [42] S. Xu y S. Wang. «Efficient Algorithm for Baseband Unit Pool Planning in Cloud Radio Access Networks». En: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. Mayo de 2016, págs. 1-5. DOI: 10.1109/VTCSpring.2016.7504207.

- [43] D. Harutyunyan y R. Riggio. «Flexible functional split in 5G networks». En: *2017 13th International Conference on Network and Service Management (CNSM)*. Nov. de 2017, págs. 1-9. DOI: 10.23919/CNSM.2017.8255992.
- [44] D. Harutyunyan y R. Riggio. «Flex5G: Flexible Functional Split in 5G Networks». En: *IEEE Transactions on Network and Service Management* 15.3 (sep. de 2018), págs. 961-975. ISSN: 2373-7379. DOI: 10.1109/TNSM.2018.2853707.
- [45] A. Garcia-Saavedra y col. «FluidRAN: Optimized vRAN/MEC Orchestration». En: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. Abr. de 2018, págs. 2366-2374. DOI: 10.1109/INFOCOM.2018.8486243.
- [46] Ragheb Rahmaniani y col. «The Benders decomposition algorithm: A literature review». En: *European Journal of Operational Research* 259.3 (2017), págs. 801-817. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2016.12.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0377221716310244>.
- [47] Nuria Molner y col. «Optimization of an integrated fronthaul/backhaul network under path and delay constraints». En: *Ad Hoc Networks* 83 (2019), págs. 41-54. ISSN: 1570-8705. DOI: <https://doi.org/10.1016/j.adhoc.2018.08.025>. URL: <http://www.sciencedirect.com/science/article/pii/S1570870518306206>.
- [48] B. Ojaghi y col. «Sliced-RAN: Joint Slicing and Functional Split in Future 5G Radio Access Networks». En: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. Mayo de 2019, págs. 1-6. DOI: 10.1109/ICC.2019.8761081.
- [49] T. Back. «Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms». En: *IEEE Congress on Evolutionary Computation*. 1994, págs. 57-62.
- [50] David H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Vol. 28. Springer Science & Business Media, 2012.

-
- [51] Gilbert Syswerda. «Uniform Crossover in Genetic Algorithms». En: *Proceedings of the 3rd International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, págs. 2-9. ISBN: 1558600663.
- [52] T. P. Runarsson y Xin Yao. «Stochastic ranking for constrained evolutionary optimization». En: *IEEE Transactions on Evolutionary Computation* 4.3 (2000), págs. 284-294.
- [53] E. Mezura-Montes y C. A. Coello Coello. «Constraint-handling in Nature-inspired Numerical Optimization: Past, Present and Future». En: *Swarm and Evolutionary Computation* 1.4 (2011), págs. 173-194.
- [54] William H Kruskal y W Allen Wallis. «Use of ranks in one-criterion variance analysis». En: *Journal of the American statistical Association* 47.260 (1952), págs. 583-621.
- [55] Henry B Mann y Donald R Whitney. «On a test of whether one of two random variables is stochastically larger than the other». En: *The annals of mathematical statistics* (1947), págs. 50-60.