

# Tabu Search for Service Function Chain Composition in NFV

Juliver Gil-Herrera, and Juan Felipe Botero

**Abstract**—One of the main challenges for service providers when they try to deploy Network Function Virtualization based networks is to efficiently make use of substrate network resources to facilitate the design, delivery and operation of network services in a dynamic and scalable manner. This challenge is known as Network Function Virtualization Resource Allocation, which can be addressed in two stages called Service Function Chain Composition and Service Function Chain Embedding. The main objective of the first stage is to compose an optimized chain of Virtual Network Functions, which offers a complete network service. The second stage is responsible for the embedding of such network service onto a physical infrastructure based on virtualization technologies. In this paper, we propose a metaheuristic algorithm to face the scalability challenges of exact approaches, when solving the composition stage. The performance of our metaheuristic approach shows that large, complex and heterogeneous service requests can be processed in reasonable running times, finding better results compared with existing heuristics in the state of the art.

**Index Terms**—communication networks, network function virtualization, virtual network functions, resource allocation.

## I. INTRODUCCIÓN

Las funciones de red son tradicionalmente llevadas a cabo por equipos dedicados de red (*middleboxes*), tales como balanceadores de carga, corta fuegos, puertas de enlaces, inspectores de paquetes, optimizadores de tráfico, entre otros. Las *middleboxes* son plataformas físicas de propósito particular, que normalmente son costosas y difíciles de mantener y actualizar. Además, estas no permiten agregar nuevas funcionalidades y tienen un corto ciclo de vida útil [1]. Esto significa que el despliegue de nueva infraestructura de red se ve comprometido en términos de gastos operativos y de capital para los operadores y proveedores de servicios de telecomunicaciones. Recientes estudios demuestran que, en una red corporativa, el número de *middleboxes* es comparable con la cantidad de enrutadores y conmutadores necesarios para mantener el funcionamiento de la red [2].

This work was supported by Tecnológico de Antioquia IU under the research project CODEI-206001179.

Juliver Gil-Herrera está con la Facultad de Ingeniería del Tecnológico de Antioquia IU, Calle 78b #72A-220, Medellín, CO (juliver.gil@udea.edu.co).

Juan Felipe Botero está con la Facultad de Ingeniería de la Universidad de Antioquia, Calle 67 #53-108, Medellín, CO (juanf.botero@udea.edu.co).

Virtualización de funciones de red (NFV, siglas en inglés) y funciones de red virtualizadas (VNFs, siglas en inglés) son dos conceptos diferentes y no pueden usarse indistintamente, para evitar confusión entre estos conceptos aclaramos lo siguiente,

NFV es un paradigma tecnológico propuesto por el Instituto Europeo de Normas de Telecomunicaciones que propone revolucionar la infraestructura de las redes, optando por soluciones basadas en virtualización, para reducir costos de despliegue y facilitar su gestión y operación. El propósito principal es reemplazar el hardware de red por aplicaciones que realicen el tratamiento o procesamiento del tráfico, pero a nivel de software. En otras palabras, es reemplazar los *middleboxes* por VNFs, las cuales son componentes virtualizados que proporciona una funcionalidad bien definida en una infraestructura de red, como por ejemplo filtrar tráfico, balancear carga, enrutar, etc. Estas VNFs pueden ser instaladas en servidores de propósito general, ubicados en centros de datos. La Fig. 1 ilustra una perspectiva sobre la virtualización de funciones de red.

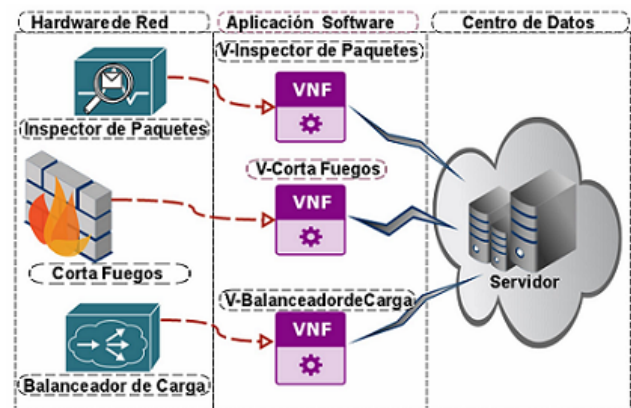


Fig. 1. Perspectiva sobre la virtualización de funciones de red.

NFV introduce modularidad y aislamiento entre las VNFs, para que sean administradas independientemente; de esta manera, se evita la proliferación de dispositivos hardware en la red [3-6]. NFV, desde una perspectiva de alto nivel, promueve y acelera la innovación en redes y servicios, permitiendo dinamizar el mercado de las telecomunicaciones a través de nuevos enfoques operativos, que proporcionan el despliegue ágil de nuevos servicios de red [7, 8].

Otra característica que aporta la tecnología NFV es la posibilidad de orquestar y ubicar funciones de red virtualizadas donde realmente se necesitan, y migrarlas, duplicarlas o eliminarlas dinámicamente en tiempo real. Este enfoque aumenta la flexibilidad y la escalabilidad de la red. De hecho, cada servicio puede ser asignado y reasignado bajo demanda, lo que brinda la oportunidad a los proveedores de servicios de utilizar sus recursos de manera eficiente [9-11].

Uno de los principales desafíos técnicos para la implementación de NFV es la administración de recursos. Este desafío se conoce como el problema de asignación de recursos en NFV (NFV-RA, siglas en inglés). La solución de NFV-RA comprende una composición de cadenas y asignación rápida, escalable y dinámica de funciones virtuales en una infraestructura basada en NFV. En un trabajo publicado en [12], se dividió NFV-RA en los siguientes sub-problemas: composición, asignación y planificación. Las entradas del problema NFV-RA son las solicitudes de encadenamiento y las salidas son las cadenas de funciones construidas.

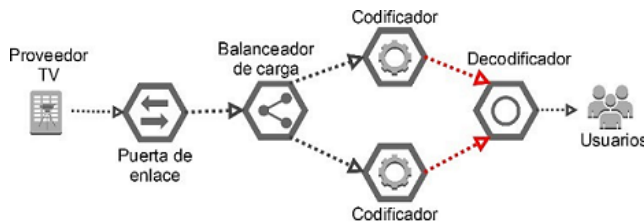


Fig. 2. Servicio de red.

Una cadena de funciones de red virtualizadas (SFC, siglas en inglés), se compone de un conjunto de VNFs concatenadas, que cumplen con un conjunto de dependencias entre sí. Diferentes cadenas pueden describir el mismo servicio de red, aunque en una cadena el orden de las VNFs puede ser flexible; es decir, algunas VNFs deben ubicarse en un orden específico (por ejemplo, una VNF que codifica tráfico debe colocarse primero que la VNF que lo decodifica, ver Fig. 2), mientras que otras VNFs son flexibles en ese sentido (es decir, una VNF no depende de otra, como en el caso de las funciones puerta de enlace y balanceador de carga de la Fig. 2, que pueden ir en cualquier orden en la cadena). Por lo tanto, la composición de la mejor cadena posible para cada servicio de red es muy importante para el operador. Al resolver esta primera fase del problema NFV-RA, se obtiene una cadena de funciones, la cual representa un servicio de red optimizado, esta puede ser atendida inmediatamente por la etapa de incrustación, la cual, toma el servicio y lo asigna a una Infraestructura física basada en NFV (NFVI) [12].

El trabajo que se presenta en este artículo se encarga de solucionar la primera etapa del problema de investigación, la cual es la composición de cadenas de servicios de red. Las propuestas de investigación existentes para resolver esta etapa utilizan enfoques exactos, poco escalables en redes de gran tamaño, o heurísticas simples que logran soluciones alejadas del óptimo. En este artículo, se propone un enfoque metaheurístico escalable basado en búsqueda tabú, para solucionar la etapa de composición de cadenas. El enfoque metaheurístico propuesto obtiene mejores soluciones, que otros trabajos presentados en el estado del arte, basados en enfoques exactos y metaheurísticos, cuando se busca la minimización de ancho de banda en las cadenas resultantes. Este artículo es una extensión del trabajo presentado en [13], una de las principales diferencias es que en este trabajo se realizó una generación dinámica de 12 solicitudes de encadenamiento (VNFRs) que incluyeron de 8 a 30 VNFs, es

decir, se evaluaron escenarios más grandes y de mayor complejidad, mientras que en [13] se evaluaron solo 4 escenarios estáticos, con un bajo nivel de complejidad, los cuales incluían menos de 12 VNFs. Además, aquí, se presenta un análisis más detallado de los resultados de la evaluación del algoritmo propuesto.

Las principales contribuciones de este trabajo son las siguientes: i) se propone un algoritmo basado en búsqueda tabú para resolver el problema de composición de cadenas de servicios de red, y ii) se diseña e implementa un entorno de simulación que permita una evaluación ágil y flexible del rendimiento del enfoque propuesto.

Este artículo está organizado de la siguiente manera: En la Sección II, se presenta una descripción general de los trabajos relacionados. En la Sección III, se describe la etapa de composición de cadenas de funciones de red bajo la perspectiva del problema de asignación de recursos. En la Sección IV, se propone un algoritmo metaheurístico rápido y eficiente para abordar la etapa de composición. La Sección V presenta la evaluación del desempeño del enfoque propuesto. Finalmente, las conclusiones y trabajo futuro se presentan en la sección VI.

## II. TRABAJO RELACIONADO

La mayoría de la literatura existente sobre la asignación de recursos en el entorno NFV se ha centrado en la fase de asignación, es decir, en la segunda fase del problema NFV-RA [14-22]. Además, se han realizado pocas investigaciones sobre la fase de planificación [23, 24], que es aún más limitada. En esta sección, se revisa el trabajo relacionado en el área de la composición de cadenas de servicios de red, pues es el objeto de esta investigación.

Las propuestas que trabajan la fase de asignación consideran las cadenas de funciones de red como una entrada ya definida del problema, es decir, se descuida la etapa de composición de la cadena. Se han propuesto pocos enfoques para resolver independientemente la etapa de composición de cadenas. Por ejemplo, Mehraghdam et al., realizan las etapas de composición y asignación de manera separada [25]; en este trabajo, se emplea un lenguaje libre de contexto para formalizar las solicitudes de encadenamiento. Además, se propone una heurística voraz rápida que intenta minimizar el ancho de banda del servicio resultante, ordenando las VNFs en orden ascendente de acuerdo con la proporción de tráfico saliente y entrante, ubicando adelante, las funciones que reducen el tráfico en la cadena.

Por el contrario, Beck y Botero presentan un enfoque coordinado para realizar las dos primeras etapas del NFV-RA en un solo paso [26, 27]. El enfoque propone una heurística recursiva que resuelve las etapas de composición y asignación de forma coordinada, es decir, la cadena se compone y se asigna simultáneamente a la infraestructura física. De esta forma, el enfoque propuesto mejora la probabilidad de incrustación, ya que la construcción de la cadena depende de su asignación exitosa en la red sustrato.

Recientemente, los autores de este artículo participaron en [28], donde se propone un enfoque de Programación Lineal

Entera para solucionar la etapa de encadenamiento (ILP-SFCC), evaluando una función objetivo que busca minimizar el ancho de banda total utilizado por la cadena resultante.

La revisión anterior muestra que, hasta ahora, se han realizado pocas o escasas investigaciones sobre la etapa de composición de cadenas de servicios de red, del problema NFV-RA. Los enfoques discutidos en esta sección presentan algunas limitaciones; por ejemplo, [28] es poco escalable y [25] no tiene un buen desempeño al ser heurístico.

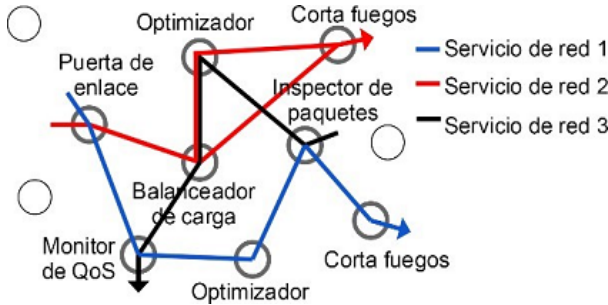


Fig. 3. Cadena de VNFs.

### III. DESCRIPCIÓN DEL PROBLEMA DE ENCADENAMIENTO DE FUNCIONES DE RED

Para comprender mejor el enfoque propuesto, esta sección presenta los términos más importantes utilizados a lo largo de este documento.

**Servicio de red** (NS, siglas en inglés): es una oferta proporcionada por un operador de red, que se entrega utilizando una o más funciones de red. Un servicio de red es una funcionalidad completa de extremo a extremo. Un ejemplo de servicio de red puede estar conformado por: puerta de enlace, balanceador de carga, codificadores y decodificador, como el servicio de televisión por internet que se ilustra en la Fig. 2.

**Función de red virtualizada** (VNF, siglas en inglés): es una aplicación software que es responsable de un tratamiento específico de flujos de paquetes a través de una red. Una VNF puede actuar en varias capas de la pila de protocolos de red. Como componentes lógicos, una o más VNFs se pueden programar como elementos virtuales e incrustar en un mismo dispositivo físico de red. Varias ocurrencias de VNF pueden existir en el mismo equipo físico. En el escenario NFV, los *proxies*, balanceadores de carga, puertas de enlaces, optimizadores de tráfico son ejemplos de VNFs.

**Cadena de VNFs** (SFC, siglas en inglés): es un conjunto ordenado o parcialmente ordenado de VNFs. El orden implícito de las VNFs en la SFC puede ser no lineal, ya que se permiten VNFs que envían tráfico a más de una rama (por ejemplo, la SFC roja de la Fig. 3 tiene una bifurcación en la VNF de balanceo de carga), y también se presentan casos en los que hay flexibilidad en el orden en que se deben ubicar las VNFs en la cadena. Las SFC pueden ser unidireccionales o bidireccionales. Muchas VNFs comunes como inspectores de paquetes y cortafuegos requieren a menudo un

encadenamiento bidireccional para garantizar que el flujo de tráfico sea estable y constante [12].

En NFV, los servicios de red eficientes requieren una asignación óptima de recursos. El problema de asignación de recursos mencionado anteriormente puede solucionarse resolviendo la etapa de composición y la etapa de incrustación o asignación. Sin embargo, la comunidad científica ha pasado por alto la composición de cadenas de red, que generalmente se suponen como una entrada al problema de asignación [12].

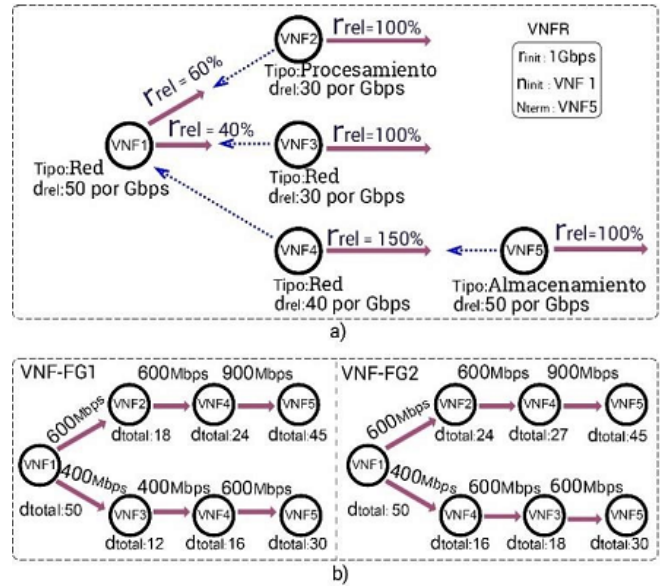


Fig. 4. Composición de cadena de servicio de red (SFC).

Una cadena debe componerse de forma inteligente y asignarse a una infraestructura física, tal que se proporcionen al servicio de red garantías de calidad de servicio de extremo a extremo. Sin embargo, dadas las dependencias entre VNFs, la composición eficiente de cadenas de servicios de red es extremadamente desafiante.

La fig. 4a), muestra cómo está constituida una solicitud de encadenamiento (es decir, una VNFR, la cual es una petición por parte de los clientes a los proveedores de servicios, donde se describen las características del servicio de red requerido). Los clientes, en lugar de proporcionar la estructura de la cadena como un todo, deben ingresar solo la información necesaria que permita a los proveedores, obtener la mejor cadena, de acuerdo con un objetivo predefinido (por ejemplo, minimizar el número de instancias de VNF o el ancho de banda).

Esencialmente, una VNFR tiene cinco elementos: i) la velocidad inicial del flujo de datos ( $r_{init}$ ), ii) el conjunto de VNFs que componen el servicio, cada una con sus respectivos requerimientos de procesamiento ( $d_{rel}$ ), iii) las VNFs donde el flujo inicia ( $n_{init}$ ) y termina ( $n_{term}$ ), iv) un número de enlaces de salidas en cada VNF (flechas sólidas), y v) las dependencias entre VNFs (flechas punteadas) [12, 28].

Los enlaces de salida se pueden usar para representar escenarios donde el tráfico se divide (por ejemplo, bifurcaciones). Si una VNF tiene más de un enlace de salida significa que esta divide el tráfico en diferentes subflujos. Como ejemplo, podemos considerar un balanceador de carga que puede separar el tráfico TCP del tráfico no TCP. Cada subflujo tiene una tasa de tráfico relativa ( $r_{rel}$ ), que puede ser mayor de 100% si la función replica o genera tráfico adicional como en el caso de la codificación de paquetes.

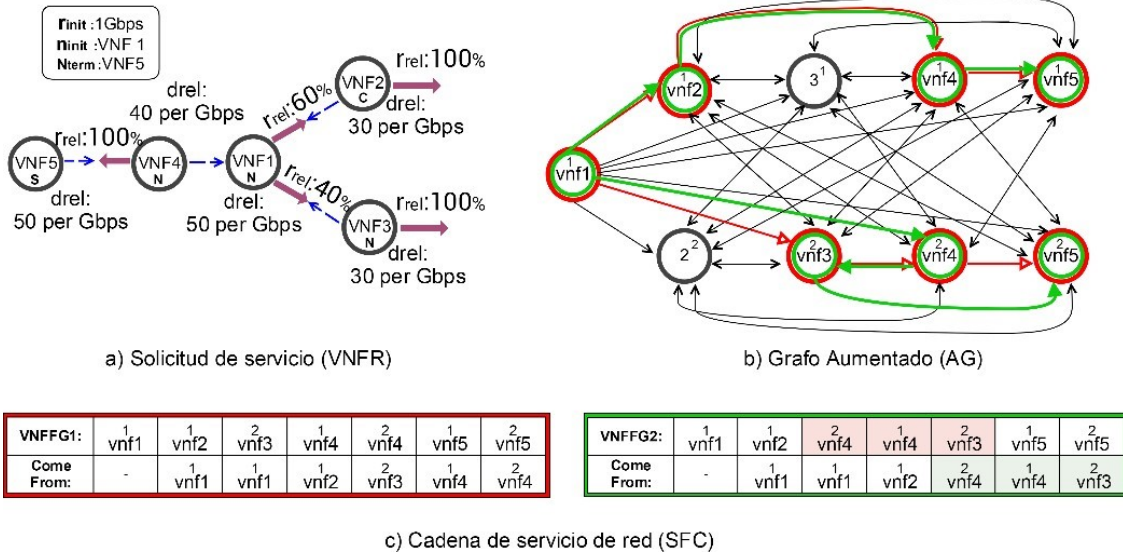


Fig. 5. Composición de cadena de servicio de red (SFC).

Las dependencias, a su vez, pueden ser de dos tipos diferentes: entre una VNF y un enlace de salida, o entre dos VNFs distintas, como se ilustra en la Fig. 4a). Para el primer caso, se observa que las VNF 2 y VNF 3 dependen del tráfico de salida de los enlaces de la VNF 1, con  $r_{rel}$  de 60% y 40% respectivamente; igualmente la VNF 5 depende del tráfico del único enlace de salida de la VNF 4. Para el caso de dependencias entre VNFs, se indica que una función dependiente debe estar obligatoriamente presente en todos y los subflujos de la cadena resultante (por ejemplo, se observa que la VNF 4 depende de VNF 1, esta última tiene dos enlaces de salida, por tal motivo es que la VNF 4, aparece en cada subflujo de las dos posibles soluciones de encadenamiento que se ilustran en la Fig. 4b). Sin embargo, las demandas de ancho de banda y de procesamiento se determinan de acuerdo con el tráfico relativo de cada enlace de salida.

Aunque estructuralmente parezcan similares, la cadena uno o grafo de reenvío uno (VNF-FG1, sigla en inglés) requiere 200Mbps menos de ancho de banda y 15 unidades menos de procesamiento en comparación con la cadena dos (VNF-FG2). Calculando estos valores se tiene que en la rama superior de la cadena uno se requiere (600 + 600 + 900) Mbps, es decir 2.1 Gbps de ancho de banda, y en la rama inferior se requieren (400 + 400 + 600) Mbps, es decir 1.4Gbps de procesamiento; entonces la suma de la demanda de ancho de banda de ambas ramas da como resultado 3.5Gbps. Análogamente, se realiza el mismo procedimiento para la cadena dos y se obtiene como

resultado 3.7Gbps. Para los cálculos de la capacidad de procesamiento de la cadena uno, se suman los valores del parámetro  $d_{rel}$  de cada VNF en ambas ramas (50 + 18 + 24 + 45 + 12 + 16 + 30), como resultado se obtiene 195 unidades de procesamiento. Análogamente se realizan los mismos cálculos para la cadena dos y se obtiene 210 unidades de procesamiento en total. Con los resultados anteriores se demuestra que seleccionando la cadena uno se incurre en menores costos asociados a la utilización de recursos tanto

para los clientes como para los proveedores. El servicio construido en esta primera fase puede ser enviado para su asignación o incrustación en la infraestructura física del proveedor.

Las principales ventajas de la composición de cadenas de servicio de red son: i) la reutilización de instancias de VNFs, ii) la posibilidad de generar nuevos y más complejos servicios de red y iii) la posibilidad de solicitar servicios sin preocuparse de los detalles de implementación. Estas ventajas brindan oportunidades para que los proveedores de servicios reduzcan los costos de operación y los gastos de capital. Si se considera la cadena como un parámetro dado por el proveedor de red, implica que, los clientes deben lidiar con complejas dependencias entre VNFs cuando generan las solicitudes de servicios, por lo tanto, los proveedores no deben estructurar las cadenas solo para encontrar la mejor opción para su infraestructura, sino, encontrar un balance de costo/beneficio tanto para ellos como para los clientes. El resultado de lo anterior podría redundar en asignaciones de cadenas sub-óptimas, que pueden requerir muchas más instancias de VNFs o mayor ancho de banda de lo necesario [27].

#### IV. ALGORITMO BASADO EN BÚSQUEDA TABÚ PARA LA COMPOSICIÓN DE SERVICIOS DE RED

En esta sección presentamos el algoritmo TS-SFCC de composición de cadenas de servicio de red. TS-SFCC se basa en una metaheurística de búsqueda tabú y resuelve la primera

etapa del problema NFV-RA. En resumen, el algoritmo propuesto, elige una cadena de servicio de red factible para una determinada solicitud de encadenamiento, que consuma el mínimo ancho de banda total. TS-SFCC está diseñado para trabajar en condiciones de alta eficiencia computacional, y lleva a cabo controles adicionales para garantizar que no se violen las limitaciones de recursos.

#### A. Metaheurística Búsqueda Tabú

Búsqueda tabú (TS, siglas en inglés) es una metaheurística propuesta por Glover [29], basada en métodos de búsqueda local y utilizada para resolver problemas de optimización combinatorios. La búsqueda tabú utiliza un procedimiento de búsqueda local para iterativamente pasar de una solución  $S$  a otra solución mejor  $S_0$ .  $N(S)$  es el vecindario de  $S$ , por lo tanto,  $S \in N(S)$ . Para explorar las regiones del espacio de búsqueda que quedarían sin evaluar por el procedimiento de búsqueda local, la búsqueda tabú modifica la estructura del vecindario de cada solución a medida que avanza la búsqueda. Además, la búsqueda tabú utiliza estructuras de memoria que describen las soluciones visitadas o los conjuntos de reglas provistos por el usuario.

Si una solución potencial ha sido visitada previamente dentro de un cierto período de tiempo o si se ha violado una regla, entonces se marca como tabú (prohibida) para que el algoritmo no considere moverse a esa solución posteriormente. Un movimiento como este se llama movimiento tabú. Sin embargo, cuando un movimiento tabú tiene una evaluación suficientemente atractiva en comparación con la mejor solución encontrada, entonces su clasificación tabú puede ser anulada. Una condición que permite que ocurra dicha anulación se llama criterio de aspiración.

#### B. Algoritmo Propuesto Basado en Búsqueda Tabú

En esta subsección, discutimos algunos componentes importantes que son parte de la heurística basada en TS, tales como: solución inicial, vecindario o soluciones vecinas, lista tabú, criterio de aspiración y criterio de parada. Al final, explicamos el pseudocódigo del algoritmo propuesto.

*Solución inicial:* En un modelo basado en búsqueda tabú, la solución inicial dependerá del algoritmo específico que la genera, se puede usar una heurística voraz para proporcionar una buena solución inicial o se puede partir de una solución obtenida aleatoriamente, pero, si no se tiene ninguna buena solución inicial, entonces se puede inicializar de cualquier resultado, con el propósito de mejorarlo en el proceso iterativo de búsqueda. Nuestro algoritmo comienza usando una solución inicial  $S_0$ , aportada por la heurística voraz propuesta en [25] que en este trabajo llamaremos HEU1. Cabe resaltar que las métricas de tiempo de ejecución y ancho de banda para HEU1 y el algoritmo TS-SFCC se evalúan por separado, obteniendo resultados independientes que luego comparamos en el análisis de resultados. La idea principal de HEU1 es encontrar rápidamente una cadena de VNFs que minimice el ancho de banda, y para esto clasifica las VNFs en orden ascendente de acuerdo con la relación entre el tráfico saliente y entrante, ubicando en las primeras posiciones de la cadena, a las VNFs que más reducen el ancho de banda. Se selecciona  $S_0$

como la solución inicial de nuestro enfoque metaheurístico (Algoritmo 1). Por tanto, la solución actual  $S$  se establece como  $S_0$ .

*Soluciones vecinas:* para encontrar otra solución  $S^*$  que sea mejor que la solución actual  $S$ , es necesario evaluar las soluciones vecinas  $S \in N(S)$ . Para este fin, primero se define el vecindario  $N(S)$ . Idealmente, todas las soluciones que implican mover cada VNF de una posición a otra, podrían producir una nueva y diferente solución. Sin embargo, esto requiere de un gran espacio de búsqueda. El vecindario  $N(S)$  se genera de acuerdo con un grafo aumentado (AG, siglas en inglés), que se ilustra en la Fig. 5b. El AG está diseñado para explorar todo el espacio de solución del problema de encadenamiento, y es similar al utilizado en el modelo de programación lineal entera (ILP, siglas en inglés) que se presentó en [28].

El AG se crea a partir de una VNFR (ver Fig. 5a) de la siguiente manera: i) para la primera VNF ( $n_{init}$ ) de la VNFR, un nodo se coloca en el grafo aumentado y, ii) para cada una de las VNFs restantes, se crean en el AG tantos nodos como la cantidad máxima de instancias que puede tener cada VNF. En la Fig. 5 se ilustra un ejemplo, donde el número máximo de instancias de las VNFs es de 2, ya que la VNF 1 divide el tráfico en dos subflujos. Finalmente, se generan enlaces dirigidos entre cada par de nodos del grafo, excepto para instancias de la misma VNF y para enlaces dirigidos al nodo inicial  $n_{init}$ .

El grafo aumentado es un conjunto de nodos conectados de acuerdo con reglas descritas en el párrafo anterior. En tales nodos se pueden instanciar VNFs, para construir una cadena (SFC) la cual representa un servicio de red. Entonces los números en los nodos del grafo aumentado indican que se pueden hospedar allí diferentes instancias de VNFs.

La orientación de los arcos está dada desde el nodo origen al nodo destino que hospeda la función sucesiva o siguiente, de acuerdo con la solicitud de encadenamiento, lo que significa que el arco está dirigido en un sentido que depende de cómo están definidas las dependencias entre VNFs.

El  $N(S)$  involucra todas las posibles soluciones que resultarían de la migración de una  $VNF_i$  desde una posición actual a otra segunda posición en la cadena, respetando las restricciones de dependencias. Si no hay una posición candidata para la  $VNF_i$ , entonces se elige la próxima VNF a evaluar.

En la Fig. 5c, se ilustran dos posibles cadenas de funciones de red, que requieren ser evaluadas en el AG para determinar cuál mantiene un mínimo consumo de ancho de banda en el servicio que ofrece. Estas dos soluciones hacen parte del vecindario y cumplen con las dependencias planteadas para el problema de encadenamiento.

*Lista tabú:* Como se discutió en la sección 3, una cadena tiene diferentes instancias de VNFs, ubicadas en posiciones específicas que cumplen con las restricciones de dependencias. Para generar una nueva solución, se intercambian dos VNFs entre sí en la misma cadena, generando de este modo una nueva cadena a evaluar. Entonces, si una  $VNF_i$  se mueve de una posición  $p_a$  a una segunda posición  $p_b$ , entonces, declaramos como tabú el

movimiento de regreso de esta función a la posición  $p_a$ , durante las siguientes  $m$  iteraciones. El movimiento se declara tabú para posteriormente no volver a evaluar las soluciones que ya han sido evaluadas con anterioridad. También es tabú elegir una solución (cadena) con un ancho de banda mayor al que posee la mejor solución conocida.

*Criterios de aspiración:* El algoritmo propuesto define un criterio de aspiración para permitir un movimiento tabú si se obtiene como resultado una solución (servicio construido) con un ancho de banda inferior al de la mejor solución conocida  $S^*$ . Ahora, si todos los movimientos disponibles se clasifican como tabú, debido a tener un ancho de banda mayor que el de la mejor solución conocida, entonces se selecciona el movimiento menos tabú, es decir, el movimiento con el más bajo ancho de banda de todos.

*Criterio de parada:* Se han definido dos condiciones de parada, que determinan cuándo el algoritmo debe detenerse: i) si no se encuentran mejores soluciones después de  $m$  iteraciones consecutivas y ii) si ninguna solución es factible en el vecindario de la solución  $S$ .

---

#### Algoritmo 1: TS-SFCC

---

```

1. procedure
2.  $S_0 \leftarrow S_{HEU1}$       solución inicial obtenida de HEU1
3.  $S \leftarrow S_0$           estableciendo la solución corriente
4.  $S^* \leftarrow S$         se define la mejor solución conocida
5.  $TL \leftarrow \{ \}$       se inicializa la lista tabú
6.  $f^* \leftarrow f^*(S)$     se evalúa la F.O para la mejor solución
7. while (el criterio de parada no se cumpla) do
8.    $N(S) \leftarrow generateNeighborhood(AG)$ 
9.   if  $N(S) \neq \emptyset$  then
10.     $N^*(S) \leftarrow getBestNeighbors(N(S))$ 
11.   end if
12.   for  $S' \in N^*(S)$  do     $S'$  solución candidata de  $N^*(S)$ 
13.     while ( $S'$  no esté en  $TL$ ) do
14.       if  $f(S') < f(S^*)$  then    evaluando F.O
15.          $S^* \leftarrow S'$ 
16.       end if
17.        $Update(TL)$               actualización de lista tabú
18.     end while
19.   end for
20. end while
21. if  $S^* \neq \emptyset$  then
22.   return  $S^*$                 mejor solución encontrada
23. else
24.   return null
25. end if
26. end procedure

```

---

El pseudocódigo para la solución propuesta se muestra en Algoritmo 1. Una solución inicial se determina a partir de HUE1 en la línea 2, los pasos de inicialización (líneas 3 y 4) establecen a HUE1 como la solución actual  $S$  y  $S^*$  como la mejor solución conocida, respectivamente. Además, la lista tabú  $TL$  se inicializa como vacía (línea 5). Para concluir la fase de inicialización, se establece el valor de la función objetivo para la mejor solución conocida  $f(S^*)$  (línea 6). El bloque *mientras* entre la línea 7 y 20 intenta encontrar una solución hasta que se cumpla el criterio de detención.

Entre las líneas 8 a 11, se crean las soluciones vecinas  $N(S)$  y  $N(S^*)$ . Como  $N(S)$  se genera a partir del grafo aumentado, entonces se pueden presentar soluciones vecinas que no cumplan las restricciones de dependencias. Por esta razón, se evalúan solo las soluciones de  $N(S^*)$ , las cuales cumplen con las restricciones de dependencias.

La solución con el ancho de banda más bajo se elige en la línea 15 y la lista tabú se actualiza en la línea 17. Si hay una solución con un ancho de banda inferior a la mejor solución actual (línea 21), entonces, el algoritmo devuelve la nueva mejor solución en la línea 22 o nulo en la línea 24.

## V. EVALUACIÓN Y RESULTADOS

En esta sección, se presenta una evaluación del desempeño de TS-SFCC. El algoritmo se enfoca principalmente en minimizar el ancho de banda total requerido por la cadena resultante. Un mecanismo dinámico de generación de solicitudes se emplea para generar las peticiones de servicio, con el que se construyen solicitudes donde se asignan aleatoriamente el número de enlaces de salidas de cada VNF, la tasa de tráfico relativo, las dependencias entre funciones y otros parámetros asociados a la capacidad de procesamiento de cada VNF. Bajo estas condiciones se puede analizar la escalabilidad y el desempeño del TS-SFCC.

TS-SFCC se compara contra el enfoque heurístico HEU1 descrito en [25], que ha sido uno de los primeros trabajos que formalmente ha enfrentado la etapa de composición de cadenas de servicios de red, y ha proporcionado valiosos resultados iniciales como primer paso para resolver el problema de asignación de recursos en arquitecturas de red basadas en NFV. Como se ha mencionado anteriormente, en [28], se propuso un modelo exacto de programación lineal entera para solucionar la fase de composición de cadenas. Sin embargo, este modelo presenta problemas de escalabilidad, dado que, debido a los altos tiempos de ejecución, no se pueden obtener resultados para peticiones con más de 7 VNFs. Por tal motivo, nosotros en este artículo proponemos TS-SFCC, con el objetivo de procesar solicitudes de encadenamiento heterogéneas y encontrar soluciones a requerimientos con dependencias complejas y con mayor número de VNFs. Finalmente, se comparan los valores de tiempo de ejecución, así como los valores de ancho de banda total requerido en las soluciones obtenidas por el algoritmo TS-SFCC frente a los resultados obtenidos por el enfoque HEU1. Cabe resaltar que, para problemas reales de gran escala, este es solo un acercamiento de lo que se puede hacer con la utilización de metodologías de optimización para la asignación de recursos en redes, debido a que el tamaño de las pruebas realizadas, no son lo suficientemente grandes comparadas con las que se requieren en escenarios reales.

### A. Escenario de Simulación y Configuración de los Experimentos

Para el escenario de simulación, se generaron dinámicamente 12 solicitudes de encadenamiento, cada una requiere entre 8 a 30 VNFs respectivamente con incremento de dos en dos, es decir, la VNFR 1 requiere 8 VNFs, la VNFR

2 requiere 10 VNFs y así sucesivamente. Se estableció que una VNF puede tener a lo sumo 2 dependencias, esto significa que a una función se le asigna entre 0, 1 o 2 dependencias siguiendo una distribución uniforme con una o varias VNFs. La velocidad de datos inicial de las peticiones de encadenamiento  $r_{init}$  se estableció en 1Gbps (unidades de ancho de banda) y la cantidad de capacidad de procesamiento relativa  $d_{rel}$  para cada VNF se distribuye uniformemente entre 10 y 50 unidades de ancho de banda. El número mínimo de instancias  $MI(i)$  para cada función se configura en 1. Para cada enlace de salida  $L_{out}^i$ , el parámetro de tasa de tráfico relativo se distribuye uniformemente entre 10 y 150%. La tasa de tráfico relativa del enlace  $(i, b)$  identifica el enlace fuente de la  $VNF_i$ , donde  $b$  es el número del enlace. Los elementos tabúes poseen la misma *tenure* por lo que permanecen en la lista por 5 iteraciones consecutivas. La tabú *tenure* o tamaño de la lista tabú se definió en 5. La configuración de parámetros en los experimentos es acorde al estado del arte presentado en [28].

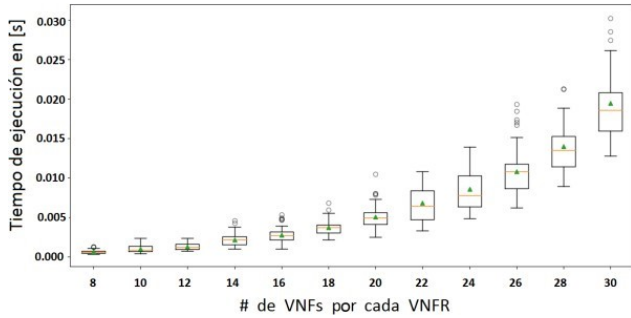


Fig. 6. Intervalos de confianza del tiempo de ejecución del TS-SFCC.

Se evaluó el tiempo de ejecución del algoritmo para verificar su escalabilidad, estableciendo el criterio de parada con 10, 20, 50 y 100 iteraciones respectivamente. La evaluación del TS-SFCC de acuerdo con el número de iteraciones definido por el criterio de parada, arrojó como resultado que, a medida que aumentaba el número de iteraciones, también aumentaba de manera no uniforme el tiempo promedio de ejecución. El algoritmo emplea en promedio 33211.42 milisegundos, cuando procesa 100 iteraciones. En la Tabla 1 se relacionan los promedios de los tiempos de ejecución del algoritmo para 20, 30, 50 y 100 iteraciones.

TABLA I  
TIEMPO DE EJECUCIÓN DEL ALGORITMO TS-SFCC

Iteraciones	Tiempo promedio (ms)
20	4101.06
30	8917.38
50	14987.75
100	33211.42

Los experimentos se realizaron utilizando un computador portátil equipado con un procesador Intel i7-4510U de doble núcleo con CPU de 2,6 GHz y 8 GB de memoria RAM. Las figuras que se presentan a continuación se generaron cuando el

algoritmo realizaba 50 iteraciones para cada VNFR, con el fin de garantizar la interpretación y estabilidad de los resultados que se muestran con un intervalo de confianza del 95%.

### B. Comparación de los Resultados de Tiempo de Ejecución y Ancho de Banda

Para evaluar el desempeño de la solución propuesta, se comparó TS-SFCC contra el enfoque HEU1 y no contra el ILP-SFCC porque este modelo exacto no es escalable para redes medianamente grandes. La comparación se realiza en términos de tiempo de ejecución y ancho de banda total requerido por la cadena resultante. En cuanto a la escalabilidad, la Fig. 6 muestra que el TS-SFCC, procesa todas las VNFRs generadas; y que el tiempo de ejecución promedio requerido para encontrar soluciones a tales solicitudes, aumenta a medida que se incrementa el número de VNFs en cada VNFR. Este comportamiento se debe al aumento del nivel de complejidad de las VNFRs, dado que es posible configurar mayor cantidad de dependencias entre VNFs.

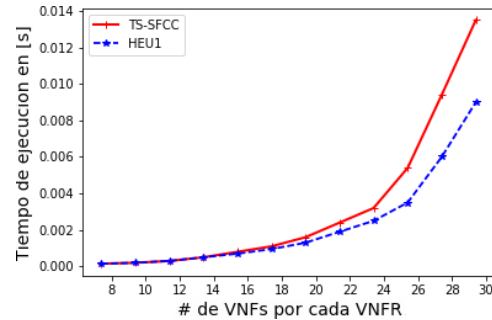


Fig. 7. Comparación de los promedios de los tiempos de ejecución de los enfoques.

Analizando la Fig. 7, se observa que los tiempos que requiere TS-SFCC son iguales o similares en las primeras cinco solicitudes y levemente superior en las solicitudes de más de 18 VNFs; en este caso, la HEU1 es en promedio levemente más rápida que TS-SFCC en el orden de 10.25 milisegundos. El análisis anterior evidencia que el algoritmo propuesto encuentra soluciones a las solicitudes de encadenamiento en tiempo factible, en el orden de los milisegundos.

TS-SFCC se desempeña adecuadamente cuando procesa VNFRs que requieren de bifurcaciones del tráfico en funciones posteriores a  $n_{init}$ . Esto es un aspecto crítico que afecta el desempeño del modelo exacto presentado en [28]. Como se había mencionado anteriormente, a medida que aumenta la cantidad de requerimientos de bifurcaciones, aumenta también el nivel de complejidad de esta, este hecho es lo que hace que el modelo exacto no solucione este tipo de solicitudes. Mientras que con TS-SFCC se logra superar ese aspecto crítico.

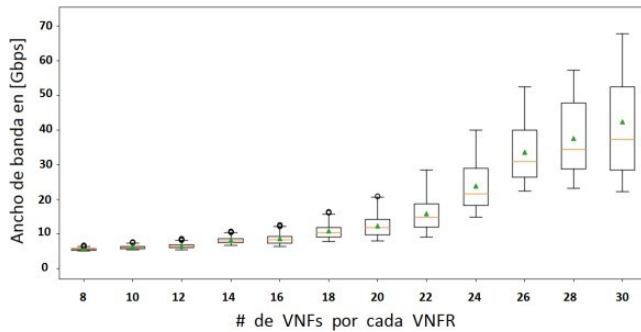


Fig. 8. Intervalos de confianza del ancho de banda del TS-SFCC.

En los experimentos, se configuró que una VNF puede bifurcar el tráfico hasta en cuatro ramas, esto se establece con el número de enlaces de salida de la VNF. Este parámetro es clave a la hora de analizar los resultados de ancho de banda, dado que, entre más bifurcaciones en la cadena, aumenta el consumo de ancho de banda. Otro aspecto importante es la cantidad de funciones (VNFs) que demanda la solicitud de encadenamiento. En la Fig. 8 se observa que, a medida que aumenta el número de VNFs en la VNRF, aumentan también las demandas de ancho de banda total para cada solicitud de encadenamiento. Esta tendencia se debe a los aspectos discutidos en el párrafo anterior, que son consecuencia del aumento del nivel del tamaño y complejidad de las VNFRs.

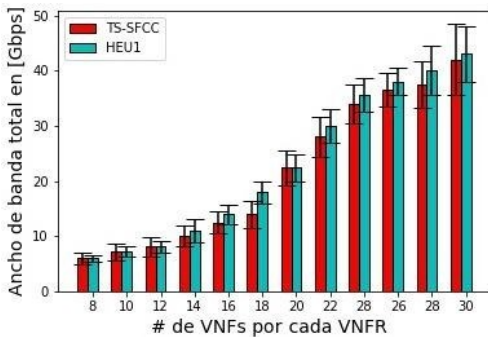


Fig. 9. Comparación del consumo de ancho de banda de los enfoques.

La Fig. 9 muestra los consumos de ancho de banda de cada VNFR para cada enfoque. Aquí, observamos algunas diferencias en los resultados. En la mayoría de los casos, TS-SFCC obtiene menores demandas de ancho de banda que HEU1. El mejor caso de reducción de ancho de banda se presenta en la VNFR 6 con un porcentaje de 22.23%.

Para finalizar la evaluación de los resultados, se puede determinar que nuestro algoritmo es escalable y eficiente dado que ofrece mejores resultados de ancho de banda en la mayoría de los escenarios estudiados.

## VI. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo, se estudia la composición de cadenas de funciones de red. Se propone un enfoque metaheurístico basado en búsqueda tabú (TS-SFCC), para resolver la fase de encadenamiento del problema de asignación de recursos en NFV. Este enfoque encuentra soluciones en tiempo razonable,

ya que el modelo exacto (ILP-SFCC) propuesto por nosotros, no es escalable para peticiones de encadenamiento (VNFR) complejas y que involucran un número significativo de funciones de red virtualizadas. Después de analizar los datos experimentales se evidencia que TS-SFCC supera a HEU1 en términos de calidad de la solución, eso significa que encontramos una solución de alta calidad en un tiempo factible, cuando el objetivo es minimizar el ancho de banda total utilizado por la cadena resultante. Como trabajo futuro, se pretende formular y evaluar una solución al problema de asignación de recursos en NFV, abordando conjuntamente y de manera coordinada las fases de composición de cadenas y la fase de incrustación de VNFs, en una infraestructura física que soporte la tecnología de virtualización de funciones de red.

## REFERENCIAS

- [1] V. Sekar, S. Ratnasamy, M. K. Reiter, N. Egi, and G. Shi. "The middlebox manifesto: Enabling innovation in middlebox deployment". In Proc. HotNets, pages 21:1-21:6, 2011.
- [2] J. Sherry and S. Ratnasamy, "A survey of enterprise middlebox deployments," Dept. EECS, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2012-24, Feb. 2012.
- [3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," IEEE Communication Magazine., vol. 53, no. 2, pp. 90-97, Feb. 2015.
- [4] Gil-Herrera, Juliver, and Botero, Juan Felipe. "Network functions virtualization: A survey." IEEE Latin America Transaction, p983-997, 2016.
- [5] Mijumbi, Rashid, et al. "Network function virtualization: State-of-the-art and research challenges." IEEE Communication Surveys & Tutorials, 2016.
- [6] E. S. Correa, L. A. Fletscher, and J. F. Botero, "Virtual data center embedding: A survey," IEEE Latin America Transaction, vol. 13, no. 5, pp. 1661-1670, May 2015.
- [7] J. Batalle, J. Ferrer Riera, E. Escalona, E. Grasa and J.A. García Espin. Virtual Network Function Scheduling-Concept and Challenges. Smart Communications in Network Technologies, p. 1-5, Jun 2014.
- [8] A. Barakabitze, A. Arslan, M. Rashid, and H. Andrew, "5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, pp. 1-40, February 2020.
- [9] F. Callegati, W. Ceroni, C. Contoli, G. Santandrea, "Implementing dynamic chaining of virtual network functions in open stack platform", in: Proceedings of the 17th International Conference on Transparent Optical Networks (ICTON), pp. 14 July 2015.
- [10] Zheng, Z., Bi, J., Yu, H., Wang, H., Sun, C., Hu, H., & Wu, J. (2019). Octans: Optimal Placement of Service Function Chains in Many-Core Systems. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. doi:10.1109/infocom.2019.8737544
- [11] Xie, Y., Wang, S., & Dai, Y. (2020). Revenue-maximizing virtualized network function chain placement in dynamic environment. Future Generation Computer Systems, 108, 650-661. doi:10.1016/j.future.2020.03.011
- [12] J. Gil-Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," in IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 518-532, Sept. 2016. doi: 10.1109/TNSM.2016.2598420.
- [13] J. Gil-Herrera and J. F. Botero, "A scalable metaheuristic for service function chain composition," 2017 IEEE 9th Latin-American Conference on Communications (LATINCOM), Guatemala City, 2017, pp. 1-6. doi: 10.1109/LATINCOM.2017.8240194.
- [14] H. Moens and F. De Turck, "VNF-P: A model for efficient placement of virtualized network functions," Proceedings of the 10th International Conference on Network and Service Management, CNSM14, 2015.
- [15] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 1346-1354.



- [16] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte "Orchestrating virtualized network functions," IEEE Transactions on Network and Service Management, 2016.
- [17] A. Gupta, M. F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee, On service chaining using virtual network functions in network-enabled cloud systems, in Proc. IEEE International Conference Adv. Network Telecommunication System (ANTS), Kolkata, India, Dec. 2015, pp. 13.
- [18] R. Riggio, T. Rasheed, and R. Narayanan, Virtual network functions orchestration in enterprise WLANs, in Proc. IFIP/IEEE Int. Symposium on Integrated Network Management. (IM), Ottawa, ON, Canada, May 2015, pp. 1220-1225.
- [19] Bo Yi, Xingwei Wang, Min Huang, Design and evaluation of schemes for provisioning service function chain with function scalability, In Journal of Network and Computer Applications, Volume 93, 2017.
- [20] Taixin Li, Huachun Zhou, Hongbin Luo, A new method for providing network services: Service function chain, In Optical Switching and Networking, vol 26, 2017, pp 60-68, ISSN 1573-4277, <https://doi.org/10.1016/j.osn.2015.09.005>.
- [21] Antonio Marotta, Enrica Zola, Fabio D'Andreagiovanni, Andreas Kessler, A fast robust optimization-based heuristic for the deployment of green virtual network functions, In Journal of Network and Computer Applications vol 95, 2017.
- [22] X. Chen, W. Ni, I.B. Collings, X. Wang, S. Xu, Automated function placement and online optimization of network functions virtualization, IEEE Transactions on Communication 67 (2) (2019) 1225-1237, <http://dx.doi.org/10.1109/TCOMM.2018.2877336>.
- [23] R. Mijumbi et al., "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in Proc. 1st IEEE Conference on Network Softwarization (NetSoft), London, U.K., Apr. 2015, pp. 19.
- [24] J. F. Riera, X. Hesselbach, E. Escalona, J. A. Garca-Espin, and E. Grasa, "On the complex scheduling formulation of virtual network functions over optical networks," in Proc. 16th International Conference on Transparent Optical Network. (ICTON), Graz, Austria, Jul. 2014, pp. 15.
- [25] Mehraghdam, S., Keller, M., Karl, H.: Specifying and placing chains of virtual network functions. In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), October 2014, pp. 713.
- [26] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in Proc. IEEE Global Communications Conf. (GLOBECOM), Sel. Areas Communication Software Defined Network Functions (GC- SACSDN), San Diego, CA, USA, Dec. 2015, pp. 16.
- [27] Michael Till Beck, Juan Felipe Botero, "Scalable and coordinated allocation of service function chains," in Computer Communications Journal, vol 102, 2017, pp. 78-88.
- [28] Ocampo, A. F., Gil-Herrera, J., Isolani, P. H., Neves, M. C., Botero, J. F., Latre, S., Zambenedetti, L., Barcellos, M. P., and Gaspary, L. P. "Optimal Service Function Chain Composition in Network Functions Virtualization". in Proc. 11th IFIP International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2017, Zurich, July 2017, pp. 62-76.
- [29] F. Glover and M. Laguna, "Tabu Search". Norwell, MA, USA: Kluwer Academic Publishers, 1997.



**Juan Felipe Botero** received the Computer Science Degree from the University of Antioquia, Colombia, in 2006, and the M.Sc. and Ph.D. degrees in telematics engineering from the Telematics Department, Technical University of Catalonia, UPC, Barcelona, Spain, in 2008 and 2013, respectively. He is an Assistant Professor with the University of Antioquia. In 2013, he joined the Applied Telecommunications Research Group, University of Antioquia. His main research topics are energy efficiency, network virtualization, SDN, and network functions virtualization.



**Juliver Gil-Herrera** received his B.S. degree in Telecommunications Engineering in 2013. Also, he received his M.Sc. degree in Computer Engineering from the Polytechnic University of Turin, Italy in 2013. In 2018, he received his Ph.D. degree in Electronic and Computer engineering from the University of

Antioquia. His main research and professional interests involve algorithms, programming, SDN, and network functions virtualization, also he is interested in AI, Blockchain, Data Science and Internet of Things (IoT).