Estudio de estrategias constructivas para un problema de embebido de redes virtuales

Jorge Sánchez-Díaz[0009−0004−6280−0510], Leticia Hernando[0000−0003−2828−0748], Sergio Cavero[0000−0002−5258−5915], and Eduardo G. Pardo[0000−0002−6247−5269]

Universidad Rey Juan Carlos
{jorge.sdiaz, sergio.cavero, eduardo.pardo}@urjc.es

Universidad del País Vasco UPV/EHU
leticia.hernando@ehu.eus

Resumen. La virtualización de redes flexibiliza las infraestructuras al permitir el despliegue concurrente de múltiples redes virtuales sobre una misma red física. Este modelo define el Problema de Embebido de Redes Virtuales (VNEP, del inglés, *Virtual Network Embedding Problem*), que consiste en mapear demandas lógicas de una red virtual, sobre recursos físicos, satisfaciendo restricciones de capacidad y minimizando costes. Esta investigación aborda la variante dinámica (*online*) del problema, caracterizada por la llegada secuencial de múltiples solicitudes. Dada su naturaleza NP-difícil, el VNEP suele resolverse mediante heurísticas constructivas, sin embargo, los enfoques tradicionales sufren de una miopía algorítmica que fragmenta prematuramente la red. Para mitigar esta deficiencia, se propone integrar mecanismos con consciencia topológica para seleccionar el nodo anclaje inicial y funciones de desempate basadas en métricas de centralidad estructural. La evaluación experimental, realizada sobre 72 escenarios dinámicos, analiza el impacto de estas alternativas en la búsqueda de la configuración algorítmica más efectiva para maximizar el rendimiento del sistema.

Palabras Clave: Embebido de Redes Virtuales · Heurísticas · Constructivos

1 Introducción

La virtualización de redes (NV, del inglés, *Network Virtualization*) constituye un avance tecnológico esencial en los entornos de computación en la nube y telecomunicaciones de nueva generación, tales como las redes 5G y 6G. Al posibilitar el despliegue concurrente de múltiples redes virtuales (VN, del inglés, *Virtual Networks*) sobre una infraestructura física compartida, la NV dota a los operadores de red de flexibilidad y escalabilidad para gestionar demandas dinámicas y heterogéneas.

El problema de optimización subyacente a esta flexibilidad se formaliza mediante el Problema de Embebido de Redes Virtuales (VNEP, del inglés, *Virtual*

Network Embedding Problem). Este problema de optimización combinatoria consiste en asignar de manera eficiente los recursos lógicos demandados por las peticiones de redes virtuales (VNR, del inglés, *Virtual Network Requests*), compuestas por nodos y enlaces, sobre una red física o sustrato (SN, del inglés, *Substrate Network*). Este proceso debe garantizar el cumplimiento estricto de las restricciones de capacidad de la infraestructura, fundamentalmente procesamiento (también denotado como CPU) y ancho de banda, persiguiendo simultáneamente la minimización del coste operativo derivado de dicha asignación.

El VNEP representa un marco general que abarca una amplia familia de variantes en función de la naturaleza del escenario operativo [4, 9]. Estas variantes difieren fundamentalmente en el número de solicitudes procesadas (únicas o múltiples) [2, 7], en su modelo temporal de llegada (*offline*, *online* o por lotes) [5,6,8] y en la capacidad de reconfiguración de los mapeos preexistentes [3]. Independientemente de la variante abordada, el VNEP constituye un problema computacionalmente intratable; incluso en su versión más fundamental, consistente en el mapeo de una única VNR, ha demostrado ser NP-difícil [1]. En consecuencia, la aplicación de métodos exactos resulta inabordable para instancias de dimensiones realistas, haciendo indispensable el diseño de enfoques heurísticos y metaheurísticos avanzados.

Este trabajo se centra en la variante *online* del VNEP con múltiples peticiones, denotada como MR-VNEP. Concretamente, las VNRs llegan al sistema de forma asíncrona y secuencial, debiendo ser procesadas (aceptadas o rechazadas) en tiempo real, careciendo por completo de información *a priori* sobre el flujo de solicitudes futuras. Históricamente, las metodologías constructivas voraces han sido ampliamente empleadas en este dominio debido a su agilidad computacional. Sin embargo, estas estrategias adolecen de un comportamiento inherentemente miope: al priorizar de forma exclusiva la compacidad y la minimización del coste local inmediato de una única solicitud, tienden a agotar prematuramente los recursos topológicamente críticos de la red sustrato. Esta saturación localizada induce una severa fragmentación temporal y espacial de la infraestructura, penalizando drásticamente la capacidad del sistema para alojar peticiones futuras.

Para mitigar esta limitación, el presente artículo propone la extensión de un marco algorítmico constructivo previamente publicado. El objetivo principal radica en dotar a dicho algoritmo de una perspectiva global que trascienda la mera eficiencia local. Para ello, se diseñan e integran estrategias topológicas anticipatorias, orientadas a la selección inteligente de los nodos de la red física, junto con el desarrollo de funciones de evaluación alternativas para la ruptura de empates. El propósito de estos mecanismos es preservar la integridad estructural del núcleo de la red física, distribuyendo la carga operativa hacia zonas perimetrales para minimizar la fragmentación de recursos y, en última instancia, maximizar la tasa global de aceptación de las VNRs.

El resto de este artículo se organiza de la siguiente manera. La Sección 2, define formalmente el escenario *online* del MR-VNEP, detallando las variables de decisión, las restricciones matemáticas y las funciones objetivo. La Sección 3,

expone el marco constructivo base y presenta las estrategias heurísticas propuestas para mitigar la miopía del algoritmo. En la Sección 4, se exponen y discuten los resultados computacionales obtenidos en el entorno de experimentación. Finalmente, la Sección 5, resume las principales aportaciones del trabajo y plantea futuras líneas de investigación.

2 Definición del problema

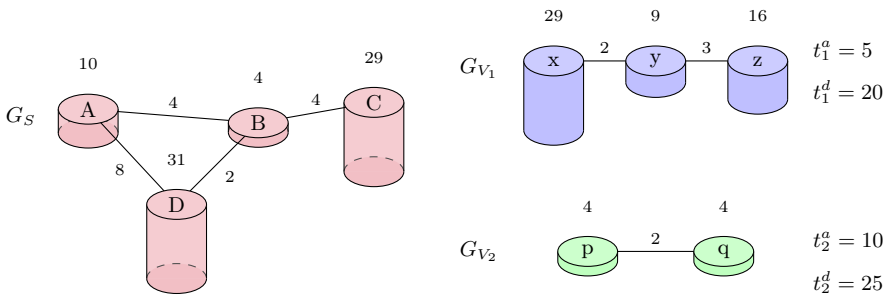
El MR-VNEP se modela como un sistema dinámico caracterizado por el flujo asíncrono de peticiones lógicas sobre una red física estática y compartida. Para formalizar este entorno, esta sección define, en primera instancia, los recursos del sustrato y la topología subyacente a las solicitudes entrantes. A continuación, se formula matemáticamente el proceso de embebido. Finalmente, se formulan las métricas de rendimiento que rigen el proceso de optimización.

La red física se modela mediante un grafo no dirigido $G_S = (N_S, L_S)$, donde N_S denota el conjunto de nodos servidores y L_S representa el conjunto de enlaces que los interconectan. Cada nodo $s \in N_S$ cuenta con una capacidad máxima de CPU, $C(s) \in \mathbb{R}^+$, mientras que cada enlace $l \in L_S$ dispone de una capacidad máxima de ancho de banda, $B(l) \in \mathbb{R}^+$. Asimismo, se define $\mathcal{P}_S$ como el conjunto conformado por todos los caminos simples, exentos de ciclos, viables entre cualquier par de vértices en G_S.

Sobre esta red opera un flujo temporal de solicitudes, caracterizado por un conjunto ordenado de índices $\mathcal{V} = \{1, 2, \ldots, M\}$. Cada VNR individual $i \in \mathcal{V}$ se define formalmente mediante la tupla (G_{V_i}, t_i^a, t_i^d). El primer componente, $G_{V_i} = (N_{V_i}, L_{V_i})$, es un grafo no dirigido que representa la topología lógica de la petición, donde cada nodo virtual $v \in N_{V_i}$ exige una demanda de CPU $C(v) \in \mathbb{R}^+$ y cada enlace virtual $e \in L_{V_i}$ requiere un ancho de banda $B(e) \in \mathbb{R}^+$. Por su parte, $t_i^a \in \mathbb{R}^+$ representa el instante de llegada al sistema, momento en el que se emite una decisión de admisión irrevocable. Finalmente, $t_i^d \in \mathbb{R}^+$ determina el instante de expiración, marcando el punto exacto en el que la red virtual es desmantelada y los recursos físicos comprometidos durante su ciclo de vida ($\Delta t_i = t_i^d - t_i^a$) son liberados íntegramente.

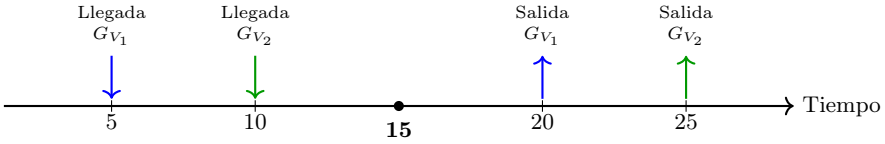
Para asimilar visualmente estos componentes fundamentales, la Fig. 1 expone el planteamiento de un escenario representativo. La red sustrato inicial, G_S, y sus capacidades nativas se detallan en la Fig. 1a. Sobre esta infraestructura, el sistema recibe un flujo compuesto por las dos solicitudes entrantes, G_{V_1} y G_{V_2}, mostradas en la Fig. 1b, cuya dinámica de llegada y expiración obedece a la estricta cronología esquematizada en la Fig. 1c.

Conforme el flujo de peticiones virtuales avanza, el sistema debe evaluar en tiempo real la viabilidad de cada solicitud entrante y determinar su admisión. Para representar las peticiones admitidas a lo largo del horizonte temporal, se define el conjunto global $\mathcal{A} \subseteq \mathcal{V}$, el cual contiene exclusivamente los índices de las soluciones aceptadas en su instante de llegada t_i^a. Consecuentemente, el subconjunto de redes virtuales que coexisten activas en la infraestructura en cualquier instante $t \in \mathbb{R}^+$ queda definido como $\mathcal{A}_t = \{i \in \mathcal{A} : t_i^a \leq t < t_i^d\}$.



(a) Red sustrato inicial.

(b) Solicitudes virtuales entrantes.



(c) Flujo secuencial de eventos.

Fig. 1: Elementos fundamentales del escenario VNEP *online*: recursos físicos base (1a), peticiones virtuales entrantes (1b) y cronología de eventos (1c).

Para que estas solicitudes puedan ser admitidas, el sistema debe establecer un mapeo lógico-físico $f^i = (f_N^i, f_L^i)$ factible, que puede ser entendido como la definición de una solución para este problema de optimización. El mapeo de nodos se define como una función inyectiva $f_N^i : N_{V_i} \to N_S$ que asigna cada nodo virtual v a un único servidor físico s. Esta inyectividad ($\forall u, v \in N_{V_i}, u \neq v \implies f_N^i(u) \neq f_N^i(v)$) constituye una restricción estructural para preservar la tolerancia a fallos. Por su parte, el mapeo de enlaces, $f_L^i : L_{V_i} \to \mathcal{P}_S$, enruta cada enlace virtual $e = (u, v)$ a través de un camino físico $p \in \mathcal{P}_S$ que conecta de forma ininterrumpida los servidores anfitriones $f_N^i(u)$ y $f_N^i(v)$.

Cualquier embebido propuesto debe respetar los límites físicos de la infraestructura. En todo momento t, el estado de ocupación generado por las peticiones activas $\mathcal{A}_t$ debe cumplir con las restricciones de procesamiento y transmisión. Formalmente, la suma de las demandas lógicas alojadas no puede exceder las capacidades físicas nativas:

$$\sum_{i \in \mathcal{A}_t} \sum_{v \in N_{V_i} : f_N^i(v) = s} C(v) \leq C(s), \quad \forall s \in N_S, \tag{1}$$

$$\sum_{i \in \mathcal{A}_t} \sum_{e \in L_{V_i} : l \in f_L^i(e)} B(e) \leq B(l), \quad \forall l \in L_S. \tag{2}$$

La calidad de esta asignación se fundamenta en maximizar la rentabilidad del proveedor. El beneficio teórico de aceptar una VNR equivale a la sumatoria de los recursos lógicos explícitamente demandados:

$$\texttt{Beneficio}(G_{V_i}) = \sum_{v \in N_{V_i}} C(v) + \sum_{e \in L_{V_i}} B(e). \tag{3}$$

En contraposición, el coste operativo refleja el consumo real de recursos. Dado que el consumo de CPU es intrínsecamente constante e independiente de la asignación efectuada, la variabilidad del coste emana exclusivamente de la longitud de los caminos físicos empleados para el enrutamiento:

$$\texttt{Coste}(f^i) = \sum_{v \in N_{V_i}} C(v) + \sum_{e \in L_{V_i}} B(e) \cdot \left| f_L^i(e) \right|, \tag{4}$$

donde la cardinalidad $\left| f_L^i(e) \right|$ denota el número de saltos del camino físico (es decir, la longitud de dicho camino). Consecuentemente, minimizar el coste equivale analíticamente a favorecer asignaciones topológicamente compactas. El rendimiento global del sistema se rige por la maximización jerárquica de la tasa de aceptación (Ecuación 5) y la eficiencia del embebido (Ecuación 6).

$$\texttt{máx} \ \frac{|\mathcal{A}|}{|\mathcal{V}|} \tag{5}$$

$$\texttt{máx} \ \frac{\sum_{i \in \mathcal{A}} \texttt{Beneficio}(G_{V_i})}{\sum_{i \in \mathcal{A}} \texttt{Coste}(f^i)} \tag{6}$$

La dinámica secuencial de estas formulaciones se ilustra sobre el escenario de la Fig. 1. El procesamiento se inicia en el instante $t = 5$ con la llegada de la solicitud G_{V_1}. Su mapeo de nodos se establece como $f_N^1(x) = C$, $f_N^1(y) = A$ y $f_N^1(z) = D$, satisfaciendo las capacidades físicas (Ecuación 1). Por su parte, el enrutamiento se resuelve mediante los caminos $f_L^1((x, y)) = ((C, B), (B, A))$ y $f_L^1((y, z)) = ((A, D))$. Según la Ecuación 3, los recursos demandados otorgan un beneficio neto de 59 unidades. Sin embargo, dado que el enlace (x, y) carece de adyacencia directa en el sustrato y requiere dos saltos físicos, la Ecuación 4 penaliza esta distancia topológica, elevando el coste operativo del embebido a 61 unidades.

Posteriormente, en el instante $t = 10$ y previo a la expiración de la primera petición, el sistema recibe la solicitud G_{V_2}. Esta nueva red se embebe de manera análoga, definiendo sus respectivas funciones de asignación de nodos y enrutamiento sobre los recursos remanentes de la infraestructura. Si se analiza la red en $t = 15$ (punto de evaluación marcado en la cronología), se observa la coexistencia de ambos grafos virtuales activos simultáneamente ($\mathcal{A}_{15} = \{1, 2\}$). Este estado de ocupación acumulada se detalla en la Fig. 2. Concretamente, el mapeo de G_{V_2} se materializa mediante la asignación $f_N^2(p) = B$, $f_N^2(q) = D$ y $f_L^2((p, q)) = (B, D)$. Al requerir exactamente un único salto físico por enlace, este segundo embebido alcanza la máxima eficiencia teórica, equiparando su beneficio al coste ($\texttt{Beneficio}(G_{V_2}) = \texttt{Coste}(f^2) = 10$). Al concluir el flujo temporal en $t = 25$ tras la salida de ambas redes, el escenario registra una tasa de aceptación perfecta del 100% y un ratio agregado de eficiencia de $69/71 \approx 0.97$.

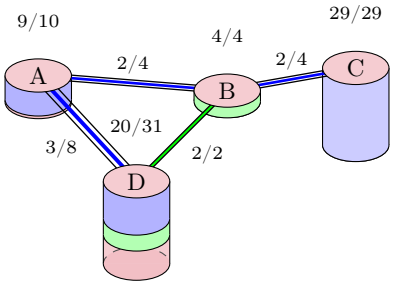


Fig. 2: Estado físico de ocupación y consumo remanente de la red sustrato tras la inserción de G_{V_1} y G_{V_2} en el instante $t = 15$.

3 Propuesta algorítmica

Resolver el VNEP de forma exacta resulta computacionalmente inviable para instancias de dimensiones realistas, una intratabilidad que se agrava severamente en la variante *online* debido a los requisitos de tiempo de respuesta. Para abordar este desafío, los autores de este trabajo proponen el marco heurístico constructivo detallado en el Algoritmo 1. Este algoritmo está concebido para generar soluciones factibles de manera ágil mediante dos fases secuenciales: el mapeo de nodos y el enrutamiento de enlaces.

El procedimiento recibe como parámetros de entrada la infraestructura física (G_S), la petición virtual a embeber (G_{V_i}) y el número máximo de iteraciones a ejecutar (M). Inicialmente, se establece la mejor solución encontrada (f^*) como vacía y su coste asociado (*MejorCoste*) a infinito (línea 1). Tras esto, el algoritmo opera de manera iterativa (línea 2). En cada iteración, la fase de nodos (líneas 4–11) selecciona un nodo virtual candidato (línea 6) que define el conjunto de nodos físicos candidatos ($\mathcal{C}_v \subseteq N_S$) que satisfacen estrictamente las restricciones de capacidad de procesamiento (Ecuación 1). A partir de dicho conjunto, el algoritmo asigna el nodo virtual a un candidato físico (línea 7). Posteriormente, la fase de enlaces determina los caminos físicos óptimos para interconectar dichos servidores (líneas 12–18), asegurando el cumplimiento de las restricciones de ancho de banda (Ecuación 2).

Para materializar las distintas operaciones de este marco, investigaciones previas centradas en el problema estático de una única petición han consolidado una configuración base altamente efectiva. Concretamente, la determinación de candidatos (Paso 1.1) restringe la búsqueda a los vértices lógicos adyacentes a los ya mapeados para preservar la topología virtual original. La selección del nodo virtual (Paso 1.2) y la ordenación de sus enlaces (Paso 2.1) se rigen por políticas voraces basadas en la máxima demanda requerida de CPU y ancho de banda, respectivamente. Por su parte, el enrutamiento de los enlaces (Paso 2.2) se resuelve de manera exacta mediante el algoritmo de Dijkstra. Bajo este esquema determinista, la selección del servidor físico anfitrión (Paso 1.3) incorpora com-

Algorithm 1 Marco Constructivo

Input: Red sustrato (G_S), solicitud virtual (G_{V_i}), número de iteraciones (M)

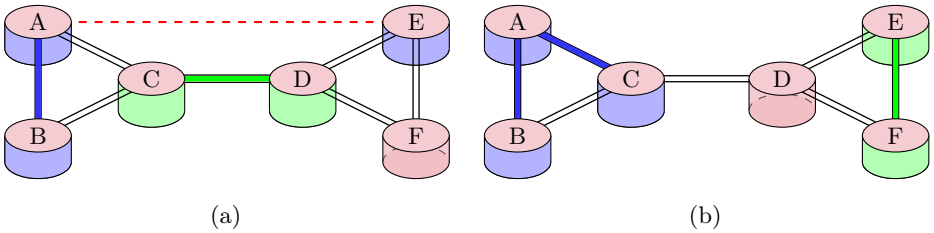
$f^* \leftarrow (\emptyset, \emptyset)$; $\quad$ *MejorCoste* $\leftarrow \infty$
for *iter* $\leftarrow 1$ to M **do**
  Inicializar mapeo actual $f^i = (f^i_N, f^i_L) \leftarrow (\emptyset, \emptyset)$
  while existan nodos virtuales sin mapear en G_{V_i} **do**
    $\mathcal{D} \leftarrow$ Determinar el conjunto de nodos virtuales candidatos ▷ Paso 1.1
    $v \leftarrow$ Seleccionar el mejor nodo virtual de $\mathcal{D}$ ▷ Paso 1.2
    $s \leftarrow$ Seleccionar nodo sustrato de C_v ▷ Paso 1.3
    if no existe un nodo s válido **then return** Fallo
    end if
    Mapear v sobre s ($f^i_N(v) \leftarrow s$)
  end while
  $\mathcal{L} \leftarrow$ Ordenar los enlaces virtuales ▷ Paso 2.1
  for cada enlace virtual $e = (u, v)$ en $\mathcal{L}$ **do**
    $p \leftarrow$ Encontrar un camino físico válido entre $f^i_N(u)$ y $f^i_N(v)$ ▷ Paso 2.2
    if no existe un camino p válido **then return** Fallo
    end if
    Mapear e sobre p ($f^i_L(e) \leftarrow p$)
  end for
  if f^i es factible y `Evaluar`$(f^i) <$ *MejorCoste* **then**
    $f^* \leftarrow f^i$; $\quad$ *MejorCoste* $\leftarrow$ `Evaluar`(f^i)
  end if
end for

Output: Mejor mapeo factible encontrado $f^* = (f^*_N, f^*_L)$

ponentes aleatorios para dotar al constructivo de la variabilidad probabilística necesaria para explorar el espacio de búsqueda durante las M iteraciones.

Aunque esta configuración garantiza embebidos locales muy compactos y arroja resultados competitivos para solicitudes aisladas, su aplicación directa en el MR-VNEP, en el que se centra esta investigación, induce ineficiencias severas. Un ejemplo de ineficacia se ilustra en el bloqueo de la Fig. 3. Si el sistema opera bajo el enfoque miope predeterminado (Fig. 3a), el embebido de una primera solicitud puede recaer arbitrariamente sobre recursos de interconexión central. Ante la llegada de una petición posterior que requiera transitar entre distintos extremos topológicos, el embebido fracasará debido a la congestión del puente (enlace (C, D)). Por el contrario, si se desplaza el embebido hacia la periferia (Fig. 3b), el núcleo del sustrato permanece intacto, garantizando la conectividad global y viabilizando la aceptación secuencial de ambas demandas.

Para resolver estas deficiencias y maximizar la tasa de aceptación a largo plazo, el presente trabajo centra su aportación en dotar al marco base de un comportamiento anticipatorio. Concretamente, se proponen dos líneas de intervención complementarias que explotan la información de la red física: el diseño de estrategias topológicas orientadas a controlar la asignación del nodo anclaje (redefiniendo la primera llamada al Paso 1.3) y la formulación de funciones de evaluación alternativas concebidas para romper empates minimizando la frag-



(a) (b)

Fig. 3: Asignación de solicitudes bajo un enfoque: (a) miope, (b) periférico.

mentación de recursos. En las siguientes subsecciones se presentan las estrategias diseñadas para abordar cada uno de estos mecanismos.

3.1 Estrategias de selección del nodo anclaje

El comportamiento miope del algoritmo base reside en la primera ejecución del Paso 1.3: al mapear el primer nodo virtual v de la solicitud, el algoritmo establece lo que en la literatura especializada se denomina el *nodo anclaje*. Dado que la filosofía del marco constructivo fuerza al resto de la red virtual a ubicarse espacialmente en torno a esta ubicación inicial, esta primera decisión condiciona el resto del embebido. Si se delega esta selección inicial a decisiones aleatorias o puramente locales, las redes virtuales tienden a consumir recurrentemente las regiones centrales de la infraestructura, aprovechando su alta densidad conectiva. Esta concentración desmedida agota prematuramente los servidores y enlaces críticos que actúan como puentes estructurales. Por el contrario, iniciar el proceso en regiones periféricas o de menor criticidad topológica preserva el núcleo del sustrato para demandas posteriores. Con el objetivo de dotar al marco de esta visión anticipatoria, se presentan cuatro estrategias para la selección del nodo de anclaje dentro del conjunto de servidores viables $\mathcal{C}_v \subseteq N_S$, cada una fundamentada en diferentes métricas topológicas:

- **Aleatoria**: selecciona el nodo anclaje de forma uniforme en $\mathcal{C}_v$,

$$s_a \sim \mathcal{U}(\mathcal{C}_v).$$

- **Grado**: prioriza nodos con menor conectividad local,

$$s_a \in \arg\min_{s \in \mathcal{C}_v} \deg(s).$$

- **Importancia**: favorece nodos en zonas de baja influencia global. Sea A la matriz de adyacencia del sustrato y $\mathbf{x}$ el autovector principal asociado al mayor autovalor λ_{max},

$$A\mathbf{x} = \lambda_{\max}\mathbf{x}, \qquad s_a \in \arg\min_{s \in \mathcal{C}_v} x_s.$$

  En la implementación, $\mathbf{x}$ se aproxima mediante el método iterativo de las potencias hasta alcanzar la convergencia:

$$\mathbf{x}_{k+1} \leftarrow \frac{A\mathbf{x}_k}{\|A\mathbf{x}_k\|}.$$

– **Intermediación**: evita nodos que actúan como cuellos de botella en rutas mínimas. Para un nodo $s \in \mathcal{C}_v$,

$$BC(s) = \sum_{\substack{u,w \in N_S \setminus \{s\} \\ u \neq w}} \frac{\sigma_{uw}(s)}{\sigma_{uw}}, \qquad s_a \in \arg\min_{s \in \mathcal{C}_v} BC(s),$$

donde σ_{uw} es el número de caminos mínimos entre u y w, y $\sigma_{uw}(s)$ los que atraviesan s.

3.2 Funciones alternativas para desempate

Debido a la naturaleza combinatoria del VNEP, el marco constructivo puede generar soluciones con idéntico coste; en un escenario *online*, pueden resultar ineficientes a largo plazo. Para discriminar entre estos embebidos, se proponen distintas estrategias de desempate que se pueden utilizar en la función `Evaluar` (líneas 19–21) como métrica secundaria para cuantificar la degradación estructural agregada.

Estas estrategias se basan en las mismas métricas de centralidad empleadas en la selección del nodo anclaje. En este caso, en lugar de evaluar únicamente el nodo inicial, las métricas se agregan sobre el conjunto de nodos físicos utilizados en la solución. De este modo, el embebido seleccionado es aquel que minimiza el valor agregado de la métrica correspondiente, es decir, aquel situado en una región menos crítica.

4 Resultados experimentales

La evaluación experimental emplea las instancias propuestas en [8], extendidas para conformar 72 escenarios dinámicos que garantizan la representatividad estadística. Esta batería de pruebas resulta de combinar 3 infraestructuras físicas de distinta capacidad, 6 conjuntos de 1000 solicitudes virtuales y 4 perfiles temporales de carga definidos por su tasa de llegada y tiempo de vida medio. Por otro lado, la implementación algorítmica se ha desarrollado en Java 17 y ejecutado sobre una máquina virtual con 16 procesadores, 32 GB de RAM y Ubuntu Server 20.04.

Para evaluar la calidad de los embebidos realizados, el análisis se fundamenta en las tres métricas:

– **Aceptadas ($\overline{\#\textbf{Aceptadas}}$):** promedio de solicitudes (es decir, peticiones de VNRs) admitidas. Son el indicador principal de rentabilidad (Ecuación 5).
– **Eficiencia ($\overline{\textbf{B/C}}$):** ratio global entre el beneficio obtenido y el coste operativo (Ecuación 6). Valores cercanos a 1 denotan mapeos compactos con una penalización mínima por enrutamiento físico.
– **Tiempo ($\overline{\textbf{Tiempo}}$):** tiempo computacional medio, en segundos, para procesar una simulación completa.

El primer experimento tiene como objetivo analizar las estrategias de anclaje presentadas en la Sección 3.1. Además, se estudia la evolución de las métricas al escalar el número de iteraciones, cuyos resultados se detallan en la Tabla 1. Los datos reflejan un comportamiento similar en todas las alternativas: la mayor mejora de rentabilidad se concentra en las primeras 100 iteraciones. Ampliar la búsqueda hasta 1000 iteraciones dispara el tiempo medio por encima de los 12 segundos por solicitud, un coste inviable para un entorno *online*, a cambio de un incremento inferior al 3% en las aceptaciones. En consecuencia, se establece un límite fijo de 100 iteraciones para el resto de los experimentos. Asimismo, resulta reseñable cómo la estrategia aleatoria, pese a su bajo rendimiento inicial, converge, razonablemente, hacia los mejores resultados al incrementar las iteraciones.

Tabla 1: Evolución del rendimiento en función del número de iteraciones para cada estrategia de anclaje.

Estrategia	**Iteraciones**	$\overline{\#\textbf{Aceptadas}}$	$\overline{\textbf{B/C}}$	$\overline{\textbf{Tiempo}}$
Aleatoria	1	356.24	0.667	0.090
	10	499.86	0.783	0.296
	100	548.48	0.842	1.686
	1000	564.31	0.868	12.315
Grado	1	375.25	0.680	0.089
	10	506.24	0.780	0.289
	100	550.07	0.837	1.680
	1000	563.58	0.861	12.400
Importancia	1	331.18	0.702	0.093
	10	471.75	0.761	0.301
	100	521.58	0.813	1.733
	1000	539.99	0.841	13.183
Intermediación	1	372.56	0.681	0.089
	10	508.69	0.787	0.291
	100	550.40	0.843	1.663
	1000	564.42	0.865	12.273

Fijado el límite de 100 iteraciones, la Tabla 2 evalúa el impacto de combinar las estrategias de anclaje con las distintas métricas de desempate. Los datos confirman una premisa fundamental: dotar al algoritmo de consciencia topológica mitiga la miopía estructural y supera de forma consistente al enfoque puramente aleatorio.

El mayor éxito operativo se logra al acoplar un anclaje por intermediación con un desempate por grado, alcanzando el máximo de 554.01 aceptaciones medias. Frente a la configuración base (Aleatorio-Aleatorio), esto supone alojar más de 5 redes virtuales adicionales por cada 1000 peticiones, utilizando exactamente la misma infraestructura física y requiriendo menos tiempo de cómputo. En el sector de las telecomunicaciones, exprimir este margen de mejora (cercano al

1%) sin inversión adicional en hardware se traduce en un aumento directo de la rentabilidad neta.

En términos de compacidad, el desempate por grado se revela como el mecanismo más eficaz para eliminar ineficiencias de enrutamiento. Al aplicarlo sobre un anclaje aleatorio, eleva la eficiencia global ($\overline{\text{B/C}}$) a 0.852, marcando el mapeo más compacto y con menor desperdicio de ancho de banda de toda la experimentación.

La única excepción a esta tendencia alcista es la centralidad por vector propio (importancia). Utilizar esta métrica para fijar el nodo anclaje desploma el rendimiento del sistema (cayendo hasta las 476.86 aceptaciones). La experimentación demuestra que forzar a las redes a esquivar los nodos sistémicamente más influyentes no protege el núcleo de la infraestructura, sino que aísla y fragmenta los recursos periféricos de forma tan agresiva que bloquea el enrutamiento de futuras peticiones.

Tabla 2: Resultados medios para las distintas combinaciones de estrategia de anclaje y función alternativa de desempate a 100 iteraciones.

Anclaje	**Desempate**	$\overline{\textbf{\#Aceptadas}}$	$\overline{\textbf{B/C}}$	$\overline{\textbf{Tiempo}}$
Aleatorio	Aleatorio	548.76	0.842	2.082
	Grado	552.68	**0.852**	2.072
	Importancia	552.53	0.848	2.061
	Intermediación	552.90	0.851	2.083
Grado	Aleatorio	547.22	0.829	1.974
	Grado	550.68	0.836	**1.967**
	Importancia	551.03	0.836	1.974
	Intermediación	550.89	0.834	1.975
Importancia	Aleatorio	476.86	0.797	2.074
	Grado	491.29	0.802	2.069
	Importancia	489.24	0.802	2.057
	Intermediación	482.90	0.800	2.076
Intermediación	Aleatorio	550.04	0.842	2.074
	Grado	**554.01**	0.849	2.054
	Importancia	553.83	0.846	2.046
	Intermediación	553.08	0.847	2.048

5 Conclusiones y trabajos futuros

Este trabajo propone la integración de métricas topológicas en un marco constructivo para resolver el MR-VNEP. Al incorporar la consciencia estructural tanto en la selección del nodo anclaje como en las funciones de desempate, se logra mitigar la miopía inherente a las heurísticas voraces tradicionales.

Los resultados confirman que guiar la búsqueda con información de la red supera de forma consistente al enfoque puramente aleatorio. En concreto, el

desempate por grado físico destaca como el mecanismo más efectivo, logrando el mayor nivel de compacidad y eficiencia global. Por el contrario, emplear la centralidad de vector propio para anclar las peticiones resulta contraproducente; forzar a las redes a esquivar los nodos de mayor influencia global fragmenta agresivamente los recursos y penaliza la tasa de aceptación a largo plazo.

Dado que la evaluación experimental comparte una misma topología sustrato subyacente, la principal línea de trabajo futuro consistirá en probar estas estrategias sobre arquitecturas heterogéneas (mallas, árboles o redes libres de escala) para validar la robustez de los hallazgos. Adicionalmente, se explorará el diseño de mecanismos de anclaje adaptativos, capaces de ajustar su comportamiento dinámicamente en función de la degradación y el estado de carga de la red en tiempo real.

Agradecimientos. Esta investigación ha sido parcialmente financiada por los proyectos con ref. ID: CIRMA-CM Ref. TEC-2024/COM-404; PID2024-160226OB-C22; PID2021-125709OA-C22; TSI-100930-2023-3 (MCA07), y MCA06.

Referencias

1. Amaldi, E., Coniglio, S., Koster, A.M., Tieves, M.: On the computational complexity of the virtual network embedding problem. Electronic Notes in Discrete Mathematics **52**, 213–220 (2016)
2. Chowdhury, M., Rahman, M.R., Boutaba, R.: Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. IEEE/ACM Transactions on networking **20**(1), 206–219 (2011)
3. Fan, J., Ammar, M.H.: Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies. In: Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications. pp. 1–12 (2006)
4. Fischer, A., Botero, J.F., Beck, M.T., Meer, H.d., Hesselbach, X.: Virtual Network Embedding: A Survey. IEEE Communications Surveys & Tutorials **15**(4) (2013)
5. Guerzoni, R., Trivisonno, R., Vaishnavi, I., Despotovic, Z., Hecker, A., Beker, S., Soldani, D.: A novel approach to virtual networks embedding for SDN management and orchestration. In: 2014 IEEE Network Operations and Management Symposium (NOMS). pp. 1–7 (2014)
6. Lischka, J., Karl, H.: A virtual network mapping algorithm based on subgraph isomorphism detection. In: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures. pp. 81–88. VISA '09, Association for Computing Machinery, New York, NY, USA (2009)
7. Moura, L.F., Gaspary, L.P., Buriol, L.S.: A branch-and-price algorithm for the single-path virtual network embedding problem. Networks **71**(3), 188–208 (2018)
8. Rubio-Loyola, J., Aguilar-Fuster, C.: Novel Initialization Functions for Metaheuristic-Based Online Virtual Network Embedding. Journal of Network and Systems Management **32**(3), 46 (2024)
9. Satpathy, A., Sahoo, M.N., Swain, C., Bellavista, P., Guizani, M., Muhammad, K., Bakshi, S.: Virtual network embedding: Literature assessment, recent advancements, opportunities, and challenges. IEEE Communications Surveys & Tutorials **27**(6), 3861–3914 (2025)