

# Uma Heurística Híbrida para o Gerenciamento de Recursos na Virtualização de Funções de Rede

Lucas Rezende Bruno  
Lucas de Freitas Vidigal

Departamento de Ciência da Computação  
Universidade Federal de São João del-Rei  
São João del-Rei – MG – Brazil

Email: lucasbruno@outlook.com, lucasvidigal3@gmail.com

Daniel Ludovico Guidoni

Fernanda Sumika Hojo de Souza  
Departamento de Ciência da Computação  
Universidade Federal de São João del-Rei  
São João del-Rei – MG – Brazil

Email: {guidoni,fsumika}@ufsj.edu.br

**Resumo**—A virtualização de funções de rede surgiu a partir da ideia de separar as funções de rede dos dispositivos de *hardware* específicos, devido ao alto custo dos *middleboxes*. Assim, são criadas instâncias virtuais chamadas de Funções Virtuais de Rede que proveem as mesmas funcionalidades de suas contrapartidas físicas. Exemplos de funções de rede incluem *firewalls*, DHCP, NAT, etc. Neste contexto, surge o problema conhecido como “Posicionamento e Encadeamento de Funções Virtuais de Rede”, pertencente à classe NP-difícil, que consiste em decidir onde serão posicionadas as funções na infraestrutura de forma adequada a atender um conjunto de requisições de serviço e garantir o encadeamento de seus fluxos. O objetivo deste trabalho foi desenvolver uma heurística híbrida baseada em métricas de centralidade, visando a alocação eficiente dos recursos físicos a serem compartilhados por um conjunto de requisições. Os resultados computacionais obtidos demonstraram que a heurística proposta é capaz de obter resultados próximos aos do solver CPLEX, e com tempo computacional inferior.

**Index Terms**—Virtualização, Heurística, Funções de Rede, Gerenciamento.

## I. INTRODUÇÃO

A Internet, maior agregação de redes de comunicações com escala global, foi criada com base no modelo TCP/IP cuja simplicidade permitiu um rápido crescimento e grande evolução de aplicações. No entanto, essa mesma simplicidade é responsável por limitações que se tornam cada vez mais evidentes e comprometem sua expansão, bem como a inclusão de novas funcionalidades que não haviam sido previstas no projeto original. Algumas limitações estão relacionadas ao atual uso do IPv4, custos elevados para expansão, dificuldade para avançar o desenvolvimento da chamada “Internet das Coisas” [1], entre outros. Este fenômeno denominado “ossificação da Internet” [2] dificulta a evolução da Internet do Futuro, na qual a busca por arquiteturas alternativas e mais flexíveis para criar uma rede mais eficiente e segura é fundamental.

Atualmente, a virtualização de redes [3], [4] vem sendo utilizada como uma alternativa para contornar o problema da ossificação da Internet, por ser possível experimentar e implementar novas funcionalidades. Durante a última década, tecnologias de virtualização vêm sendo adotadas a fim de usufruir de eficiência e agilidade de recursos de computação e armazenagem baseados em software. Através da virtualização, é possível criar uma visão lógica do *hardware*, de maneira que

múltiplas redes virtuais possam operar simultaneamente sobre os mesmos recursos físicos, também chamado de substrato de rede [5]. Alguns dos principais benefícios do uso da virtualização incluem flexibilidade, escalabilidade, isolamento, redução de custos e segurança.

Impulsionada por essa ideia, surgiu uma proposta que agrega às redes virtuais preceitos de funções virtuais de rede (*Virtual Network Function* - VNF) [6]. A Virtualização de Funções de Rede (*Network Function Virtualization* - NFV) [7], [8] consegue corrigir problemas como o uso de *middleboxes*, i.e., equipamentos de *hardware* dedicados/proprietários de alto custo e com localidades fixas que fazem o provisionamento de serviços em redes de computadores. O ciclo de serviços com o uso de *middleboxes* é bastante complexo e, reduzir seu uso passou a ser fundamental a fim de prover escalabilidade. Além disso, em caso de expansão dos serviços, faz-se necessário o aumento da infraestrutura física, ocasionando custos conhecidos como CAPital EXpediture (CAPEX) e OPERATION EXpenses (OPEX) [9].

No contexto de NFV, são criadas instâncias VNFs que proveem as mesmas funcionalidades de suas contrapartidas físicas (*Physical Network Functions* - PNF). Exemplos de funções de rede incluem roteamento, *firewall*, *Dynamic Host Configuration Protocol* (DHCP), *Network Address Translation* (NAT), *proxy*, etc. A infraestrutura NFVI (*NFV Infrastructure*) provê ainda recursos como armazenamento, processamento, entre outros [8]. Na prática, serviços tendem a demandar mais de um tipo de função de rede, levando à ideia de encadeamento de múltiplas funções de rede. Por exemplo, uma aplicação de tráfego de vídeo requer um serviço de otimização de vídeo que é provido pelo uso de duas funções de rede, um otimizador de vídeo e um *firewall*, nesta ordem. As funções precisam ser atravessadas em uma sequência formando uma cadeia de serviços, que provê funções de redes encadeadas para fluxos de tráfego específicos [10].

Um dos desafios enfrentados pela abordagem da NFV é o problema de Posicionamento e Encadeamento de Funções Virtuais de Rede (VNF-PC - *Virtual Network Function Placement and Chaining Problem*). Este problema consiste na interconexão de um conjunto de VNFs através da rede, para garantir o atendimento de requisições de fluxo descritas através

de uma cadeia de serviço de rede (*Service Function Chaining* - SFC). Os fluxos correspondem a comunicações fim-a-fim, passando pelas funções que sejam necessárias. Segundo [6], este problema se divide em três fases: posicionamento, atribuição e encadeamento. A fase do posicionamento consiste na determinação da quantidade de instâncias de funções virtuais que serão necessárias para uma determinada demanda de requisições e onde serão mapeadas na infraestrutura física (os nós da infraestrutura são também referidos como N-PoPs, ou pontos de presença). A fase de atribuição determina quais funções virtuais instanciadas e mapeadas estarão destinadas a quais fluxos. A terceira e última fase destina-se a coordenar o encadeamento, criando caminhos que interconectam as funções virtuais para cada fluxo. A Figura 1 ilustra o problema VNF-PC. As maiores dificuldades encontradas para resolver o VNF-PC são ligadas à sua natureza combinatória, pertencendo à classe NP-difícil [11].

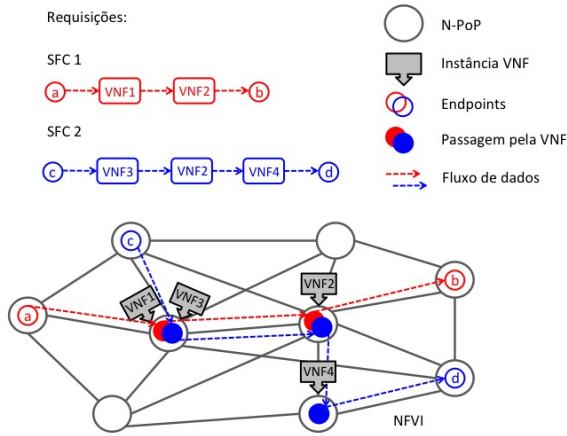


Figura 1. Posicionamento e Encadeamento de Funções Virtuais de Rede. As requisições SFC1 e SFC2 compartilham a infraestrutura (NFVI) utilizando a mesma função virtual VNF2. Todas as funções de rede foram instanciadas em N-PoPs e encadeadas para atender os fluxos das SFC1 e SFC2 corretamente.

Neste trabalho, uma heurística híbrida baseada em métricas de centralidade é proposta, visando a alocação eficiente dos recursos da infraestrutura física e das instâncias virtuais a serem compartilhados por um conjunto de SFCs. Buscando se beneficiar das qualidades de métodos exatos e heurísticos, a heurística proposta combina programação matemática e heurísticas de busca local para a resolução do problema. Os resultados computacionais obtidos demonstraram que a heurística proposta é capaz de obter resultados próximos aos do solver CPLEX, e com tempo computacional inferior.

A organização deste trabalho é dada como segue. Na seção 2, são apresentados trabalhos relacionados enquanto na seção 3, o problema VNF-PC é descrito formalmente através de sua modelagem matemática. Na seção 4, a heurística híbrida proposta é apresentada. Na seção 5, são mostrados os resultados computacionais obtidos pela heurística e pelo modelo matemático; e na seção 6, o trabalho é concluído, sendo apresentadas possibilidades de trabalhos futuros.

## II. TRABALHOS RELACIONADOS

O problema de posicionamento e encadeamento de funções virtuais tem sido bastante estudado na literatura. Em [12], é abordado o problema de posicionamento de funções de rede através de abordagens heurísticas baseadas em arredondamento que visam reduzir OPEX. Um estudo teórico acerca do problema de posicionamento de VNFs é conduzido e sua aplicação prática é demonstrada através da avaliação de desempenho em diversos cenários de rede típicos da aplicação.

Abordagens exatas e heurísticas para o problema de posicionamento e encadeamento de funções virtuais foram propostas por [6], buscando minimizar o número de VNFs implementadas na rede. Os resultados obtidos demonstraram uma redução de até 25% nos atrasos fim-a-fim, quando comparados com uma infraestrutura tradicional de rede. A heurística proposta pelos autores foi capaz de obter resultados próximos à otimalidade e com tempos computacionais factíveis.

Em [13], foram desenvolvidas uma formulação PLI e uma heurística baseada em programação dinâmica para resolver instâncias de maior porte. A heurística em questão provê soluções que estão até 1.3 do valor ótimo. Além disso, os autores demonstram que o uso de funções virtualizadas é capaz de reduzir em até 4× o valor de OPEX.

Em [14] é apresentado o NFV-PEAR, que é um framework para o posicionamento e encadeamento adaptativo das VNFs. O NFV-PEAR visa reorganizar periodicamente os posicionamentos e encadeamentos de VNFs previamente determinados, a fim de se manter um desempenho fim-a-fim aceitável quando ocorrem flutuações nos cursos de processamento e requisitos de fluxo. A partir dos resultados, foi observado que esta proposta tem um grande potencial para reduzir o número de mudanças na rede necessárias para assegurar o desempenho, dando estabilidade para os serviços.

Diferente dos trabalhos acima relacionados, a proposta de solução deste artigo é baseada em uma abordagem híbrida entre programação matemática e heurísticas, além de utilizar características topológicas (métricas de centralidade) para guiar o mapeamento das VNFs na infraestrutura física.

## III. MODELO MATEMÁTICO

O problema de posicionamento e encadeamento de funções virtuais pode ser modelado através de Programação Linear Inteira (PLI). A infraestrutura física é modelada através de um grafo não direcionado capacitado. Por outro lado, as requisições SFC são modeladas por grafos direcionados, contendo o conjunto de VNFs necessárias ao serviço, com restrições de precedência e demandas de recursos. O problema consiste em decidir onde serão posicionadas as VNFs na infraestrutura de forma adequada a atender o conjunto de requisições de serviço (SFCs), além de realizar a alocação e o encadeamento das funções para cada requisição de fluxo. O modelo matemático a seguir é baseado em [13].

### Parâmetros de Entrada:

- $G = (N, E)$  - Grafo do substrato físico
- $N$  - Conjunto de N-PoPs

- $E$  - Conjunto dos enlaces físicos
- $V$  - Conjunto de requisições SFC
- $G^v = (N^v, E^v)$  - Grafo da SFC  $v \in V$
- $N^v = N^{end} \cup N^{vnf}$  - Conjunto de nós da SFC
- $N^{end}$  - Conjunto de endpoints da SFC
- $N^{vnf}$  - Conjunto de VNFs da SFC
- $E^v$  - Conjunto dos enlaces virtuais da SFC
- $F$  - Conjunto das funções de rede
- $U_i^m$  - número de instâncias do tipo  $m \in F$  que podem ser instanciadas no N-PoP  $i \in N$  (devido a restrições físicas)

#### Atributos do substrato físico:

- $C_i$  - Capacidade de CPU do N-PoP  $i \in N$
- $B_{ij}$  - Capacidade de largura de banda do enlace físico  $(i,j) \in E$
- $d_{ij}$  - Atraso relativo ao enlace físico  $(i,j) \in E$

#### Atributos das funções de rede:

- $c_m$  - Demanda de CPU da função  $m \in F$
- $p_m$  - Capacidade de processamento da função  $m \in F$
- $d_m$  - Atraso gerado pela função  $m \in F$

#### Atributos das requisições SFC:

- $D^v$  - Atraso máx. permitido pela requisição SFC  $v \in V$
- $b_{kl}^v$  - Demanda de largura de banda do enlace virtual  $(k,l) \in E^v$ ,  $v \in V$

#### Variáveis de decisão:

- $y_i^{vk} \in \{0,1\}$  - indica se o nó virtual  $k \in N^{end}$  da requisição SFC  $v \in V$  é mapeado no N-PoP  $i \in N$ , assumindo valor 1 e 0 caso contrário.
- $z_{in}^{vk} \in \{0,1\}$  - indica se o nó virtual  $k \in N^{vnf}$  da requisição SFC  $v \in V$  é mapeado na instância virtual  $n \in U_i^m$  (onde,  $m = type(k)$ ) do N-PoP  $i \in N$ , assumindo valor 1 e 0 caso contrário.
- $w_{in}^m \in \{0,1\}$  - indica se a instância  $n \in U_i^m$  da função de rede  $m \in M$  foi criada no N-PoP  $i \in N$ .
- $x_{ij}^{vkl} \in \{0,1\}$  - indica se o enlace virtual  $(k,l) \in E^v$  da requisição SFC  $v \in V$  utiliza o enlace físico  $(i,j) \in E$  no caminho que conecta os N-PoPs que hospedam os nós virtuais  $k, l \in N^v$ .

$$\min \sum_{i \in N} \sum_{m \in M} \sum_{n \in U_i^m} w_{in}^m \quad (1)$$

$$y_i^{vk} = 1 \quad \forall v \in V, \quad \forall k \in N^{end} | i = pos(k) \quad (2)$$

$$\sum_{i \in N} \sum_{n \in U_i^m} z_{in}^{vk} = 1 \quad \forall v \in V, \quad \forall k \in N^v | m = type(k) \quad (3)$$

$$\sum_{k \in N^{vnf}} \sum_{n \in U_i^m} z_{in}^{vk} + \sum_{k \in N^{end}} y_i^{vk} \leq 1 \quad \forall v \in V, \quad \forall i \in N \quad (4)$$

$$\sum_{m \in F} \sum_{n \in U_i^m} c^m w_{in}^m \leq C_i \quad \forall i \in N \quad (5)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} b_{kl}^v x_{ij}^{vkl} \leq B_{ij} \quad \forall (i,j) \in E \quad (6)$$

$$\sum_{v \in V} \sum_{k \in N^v | m = type(k)} b_{lk}^v z_{in}^{vk} \leq p^m w_{in}^m \quad (7)$$

$$\forall m \in F, \quad \forall i \in N, n \in U_i^m$$

$$\sum_{(i,j) \in E} x_{ij}^{vkl} - \sum_{(h,i) \in E} x_{hi}^{vkl} = \sum_{n \in U_i^m} z_{in}^{vk} + y_i^{vk} - \sum_{n \in U_i^m} z_{in}^{vl} - y_i^{vl} \quad \forall i \in N, \quad \forall (k,l) \in E^v, \quad \forall v \in V \quad (8)$$

$$\sum_{(i,j) \in E} \sum_{(k,l) \in E^v} d_{ij} x_{ij}^{vkl} + \sum_{i \in N} \sum_{k \in N^{vnf}} \sum_{n \in U_i^m} d^m z_{in}^{vk} \leq D^v \quad \forall v \in V \quad (9)$$

A função objetivo (1) busca minimizar o número de instâncias de funções virtuais de rede mapeadas na infraestrutura. As restrições (2) realizam o mapeamento dos endpoints da SFC na infraestrutura. As restrições (3) realizam o mapeamento das VNFs da SFC na infraestrutura. As restrições (4) garantem que cada N-PoP  $i \in N$  hospedará apenas um nó virtual  $k \in N^v$  da requisição SFC  $v \in V$ . As restrições (5) asseguram que a disponibilidade de CPU do N-PoP  $i \in N$  não será ultrapassada pelas demandas das instâncias VNF mapeadas no mesmo. As restrições (6) garantem que a disponibilidade de largura de banda do enlace físico  $(i,j) \in E$  não será ultrapassada pelas demandas dos enlaces virtuais mapeados no mesmo. As restrições (7) garantem que a capacidade de processamento de cada instância VNF  $n \in U_i^m$  do tipo  $m \in F$  mapeada no N-PoP  $i \in N$ , será respeitada pelo tráfego  $b_{kl}^v$  das SFCs  $v \in V$  que a compartilham. As restrições (8) asseguram a conservação de fluxo nos N-PoPs  $i \in N$  para cada enlace virtual  $(k,l) \in E^v$  de uma SFC  $v \in V$ . As restrições (9) garantem que o atraso máximo  $D^v$  exigido pela SFC  $v \in V$  seja respeitado, de acordo com o atraso gerado na rota física e na passagem pelas funções de rede.

O modelo matemático (1)-(9) pode ser resolvido através de um *solver* de otimização que implemente o método *Branch-and-Bound* (BB). Contudo, é esperado que apenas instâncias de pequeno porte sejam passíveis de solução à otimalidade, devido à complexidade exponencial do algoritmo de solução. Para contornar esta limitação, uma heurística híbrida é proposta na próxima seção.

#### IV. HEURÍSTICA HÍBRIDA BASEADA EM MÉTRICAS DE CENTRALIDADE

Heurísticas são procedimentos que buscam boas soluções a um custo computacional aceitável sem, no entanto, fornecer qualquer garantia de otimalidade ou mesmo de proximidade da solução ótima [15]. Uma heurística construtiva guia a construção de uma solução inicial para o problema, elemento a elemento e heurísticas de busca local visam melhorar uma

solução inicial a partir de alterações nos seus elementos, utilizando o conceito de vizinhança. Nesta seção, é proposta uma heurística construtiva baseada em métricas de centralidade e quatro heurísticas de busca local, baseadas na remoção de VNFs instanciadas que possam estar em excesso.

#### A. Heurística Construtiva

A heurística construtiva proposta é iniciada realizando a ordenação de forma decrescente dos N-PoPs do substrato a partir de uma determinada característica topológica. Para esta implementação, foram utilizadas três métricas de centralidade: grau, *betweenness* e *closeness*. Tais métricas são definidas como segue:

- **Grau:** número de links (arestas) incidentes a um nó do grafo, ou seja, quantidade de nós do grafo que estão conectados ao nó em questão;
- **Betweenness:** conhecida como medida de intermediação. É a proporção de caminhos entre um nó qualquer A e um nó qualquer B, que passam por um nó V. Ou seja, a quantidade de caminhos que um determinado nó V é intermediário.
- **Closeness:** conhecida como média de proximidade. Mede a menor distância média entre um nó em relação aos outros nós de um grafo. Neste caso, quanto maior a centralidade maior a distância dos outros nós em relação a um determinado nó.

Após a ordenação do conjunto de N-PoPs de acordo com uma dessas métricas, a heurística seleciona iterativamente o primeiro nó do conjunto, ou seja, o N-PoP com o maior valor da métrica dada. Verifica-se se o N-PoP tem capacidade de processamento o suficiente para hospedar uma função virtual presente na lista de requisições. Caso tenha, a função é alocada ao ponto de presença e a capacidade deste N-PoP é atualizada. Caso o nó não tenha capacidade, é verificado se ele pode hospedar algum outro tipo de função virtual, visto que cada função consome recursos distintos. Se o nó não comporta nenhum outro tipo de função, ele não poderá mais ser usado no algoritmo e é retirado do conjunto; caso contrário, ele continua no conjunto para que possa ser analisado se haverá uma possibilidade de uma nova alocação de uma VNF distinta, e na próxima iteração será selecionado um novo nó. Este processo repete-se até que tenham sido instanciadas todas as VNFs necessárias para o conjunto de requisições. Este número de funções virtuais é calculado como um limite superior, antes do início da heurística construtiva, sendo para cada tipo diferente de função virtual (proxy, NAT, firewall, IDS), a soma requerida por todas as requisições juntas.

Este procedimento consiste na etapa de posicionamento das VNFs na NFVI. As etapas de atribuição e encadeamento são resolvidas através do solver de otimização como um problema de viabilidade, i.e., um modelo consistindo apenas das restrições (2),(4),(6)-(9) e uma função objetivo vazia. Após este procedimento, as VNFs que não estejam sendo utilizadas por nenhuma requisição são removidas, visto que algumas podem não ser necessárias uma vez que foram posicionadas

em função de um limite superior. O pseudocódigo a seguir representa a heurística construtiva proposta.

**Input:** Substrato  $G$ , limite superior  $l_s$ , conjunto de requisições  $V$ , característica topológica  $ct$

**Output:** Solução inicial do problema  $sol$

$ln$  = conjunto de N-PoPs do Substrato  $G$

Ordena conjunto  $ln$  de forma decrescente a partir da característica  $ct$

**while**  $l_s > 0$  **do**

$v$  = VNF do conjunto  $V$

$n$  = primeira N-PoP do conjunto  $ln$

**if**  $n$  tem capacidade para alocar  $v$  **then**

$l_s = l_s - 1$

Aloca  $v$  em  $n$

Decrementa a capacidade de  $n$  pelo consumo de recursos de  $v$

**else if**  $n$  não tem capacidade para alocar nenhuma outra VNF **then**

Retira  $n$  do conjunto  $ln$

**end if**

**end while**

▷ *Resultado*

Resolve modelo de viabilidade a partir do posicionamento realizado e guarda em  $sol$

**return**  $sol$

#### B. Heurísticas de Busca Local

A partir da solução inicial gerada pela heurística construtiva, é executado o algoritmo de busca local, que visa melhorar a solução construída utilizando técnicas de remoção. Foram implementados diferentes estratégias, dentre elas, uma busca local primeiro aprimorante, busca local randômica, busca local baseada na heurística de “VNF menos usada” e busca local baseada na heurística de “VNF mais usada”. Estas estratégias serão explicadas a seguir.

- 1) **Primeiro Aprimorante (BL1):** A busca local primeiro aprimorante procura todos os nós em que foram alocadas VNFs e tenta remover cada uma delas realocando as SFCs que a utilizavam para outras VNFs do mesmo tipo. Caso a remoção gere inviabilidade, a VNF removida é realocada. Senão, mantém-se a mudança, com a função objetivo diminuindo seu valor em uma unidade. A busca termina quando não houver mais a possibilidade de remover nenhuma VNF, onde todos os nós tenham sido analisados.
- 2) **Randômica (BL2):** A busca local randômica escolhe um nó aleatório e tenta remover uma VNF alocada no mesmo. Verifica-se se a remoção conserva a viabilidade da solução através do remanejamento das SFCs, desfazendo-a caso isto não aconteça. A busca termina depois de um número determinado de iterações.

- 3) **VNF menos usada (BL3):** A busca local que utiliza essa estratégia de remoção procura as funções que foram menos utilizadas dentro do substrato a partir de um valor denominado *slack* (valor de “folga”). Esse valor é obtido através da restrição (7) do PLI, o qual determina o quanto de processamento de uma determinada VNF está ocupado. A partir do *slack* é feita uma ordenação das instâncias mais utilizadas em ordem crescente, e aplica-se a estratégia de remoção enquanto for possível encontrar uma solução viável.
- 4) **VNF mais usada (BL4):** Essa busca mantém os mesmos princípios da busca local “VNF menos usada”; contudo, a ordenação das VNFs a partir do valor do *slack* é realizado de forma decrescente, obtendo-se assim as funções virtuais mais utilizadas.

## V. RESULTADOS COMPUTACIONAIS

Para a realização dos testes, foram utilizadas duas topologias do repositório “The Internet Topology Zoo”<sup>1</sup>. As SFCs foram geradas de forma similar a [13]. Cada SFC tem dois *endpoints*, onde o fluxo inicia e termina, escolhidos de forma aleatória a partir dos nós do substrato; e deve atravessar três funções de rede sequencialmente definidas de forma aleatória (proxy, NAT, firewall, IDS). A largura de banda do fluxo é definido aleatoriamente, sendo um valor entre [1000,320000] Mbps e o atraso máximo suportado pela SFC é um valor aleatório entre [100,200] segundos. Os resultados computacionais da heurística híbrida proposta foram comparados com aqueles obtidos através do **ILOG CPLEX**. A execução do CPLEX foi limitada a uma hora. Os testes foram realizados em um INTEL(R) Core(TM) i7-5820k, com 16GB de RAM e um sistema operacional Ubuntu LTS 16.04.

A topologia *Geant2012* consiste em um grafo com 40 nós e 61 arestas, uma densidade de aproximadamente 0.078 e um diâmetro de tamanho 8. A topologia *UUNET* possui com 49 nós e 84 arestas, densidade aproximada de 0.071 e diâmetro 8 de tamanho. As Tabelas I e II apresentam os valores das soluções (minimização do número de funções virtuais) encontradas em cada uma das variações da heurística híbrida. CPLEX trata-se da solução obtida após uma hora de execução, seguida do GAP que representa a distância estimada para o valor ótimo. *Sol\_Inicial* é a solução encontrada pela heurística construtiva.

É possível perceber que todas as buscas conseguiram alcançar um resultado muito próximo do valor obtido pelo CPLEX nas duas topologias e números de requisições diferentes; em muitas delas obtendo o mesmo resultado (em negrito). Não houve impacto significativo na qualidade da solução final, entre a escolha das três diferentes características topológicas para a seleção dos nós na heurística construtiva. Entre as quatro buscas locais, também não houve uma diferença significativa, mas BL1 e BL4 tiveram mais soluções com a função objetivo igual à do CPLEX. Ressalta-se que cada uma das buscas locais são distintas, então não apresentam os mesmos resultados. O GAP apresentado na coluna de solução

do CPLEX indica a dificuldade de resolução do problema, pois mesmo após uma hora de execução, a otimalidade só foi comprovada em uma instância.

Comparando-se com outros trabalhos da literatura, a heurística baseada em programação dinâmica proposta em [13] garante soluções até 1.3 vezes o valor ótimo, ou seja, até 30% inferiores. Analisando os valores de GAP e os resultados da heurística híbrida aqui proposta, é possível perceber que tais limites não foram atingidos nos cenários testados, demonstrando a qualidade das soluções obtidas.

Na Figura 2, um gráfico comparando o tempo médio gasto para cada uma das buscas em função do número de SFCs é apresentado. A busca randômica foi a mais rápida, seguida das heurísticas de VNF mais e menos usadas. A busca de primeiro aprimorante é a mais custosa. Considerando o custo/benefício, pode-se concluir que as heurísticas de VNF mais e menos usadas apresentaram resultados mais interessantes na maioria dos testes, obtendo um valor de FO próximo ao CPLEX por um custo de tempo ainda menor que o da busca local primeiro aprimorante. Comparando com o tempo do CPLEX, que executou no tempo limite de uma hora, as variações da heurística híbrida obtiveram um tempo de execução inferior e factível.

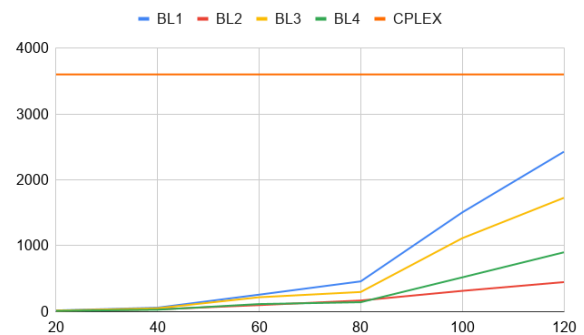


Figura 2. Tempo médio das Buscas Locais de acordo com o nº de requisições

As Tabelas III e IV apresentam os valores intermediários obtidos pelo CPLEX nos tempos 1, 10, 30 e 60 minutos de execução, quando o limite de tempo é atingido. O objetivo da apresentação destes resultados é demonstrar que a solução entregue ao final de uma hora de execução, nem sempre foi encontrada rapidamente pelo *solver*. Para os conjuntos de SFCs menores, com 20 e com 40 SFCs, a solução de fato foi encontrada no primeiro minuto de resolução, mas sua otimalidade não foi comprovada após uma hora, não sendo possível afirmar que se trata da solução ótima. Para conjuntos de SFCs maiores, com 40 a 80 requisições, a solução retornada foi encontrada antes dos 10 min de execução. Porém, para conjuntos de maior porte, com 100 ou 120 requisições, as soluções retornadas foram encontradas com até 30 min de execução, sem comprovação de otimalidade após uma hora. Isso demonstra que a heurística foi capaz de atingir resultados semelhantes em tempos inferiores aos do CPLEX.

<sup>1</sup>www.topology-zoo.org

Tabela I  
TOPOLOGIA GEANT 2012

N° SFCs	Grau				Closeness				Betwensness				CPLEX			
	Sol_Inicial	BL1	BL2	BL3	BL4	Sol_Inicial	BL1	BL2	BL3	BL4	Sol_Inicial	BL1		BL2	BL3	BL4
20	40	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b>	43	<b>14</b>	15	<b>14</b>	<b>14</b>	39	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b> (0,0%)
40	76	<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b>	78	<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b>	79	<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b> (7,7%)
60	109	<b>38</b>	<b>38</b>	<b>38</b>	<b>38</b>	107	<b>38</b>	<b>38</b>	39	<b>38</b>	114	<b>38</b>	<b>38</b>	<b>38</b>	<b>38</b>	<b>38</b> (8,3%)
80	130	<b>49</b>	50	52	<b>49</b>	124	<b>49</b>	<b>49</b>	<b>49</b>	<b>49</b>	124	<b>49</b>	50	<b>49</b>	<b>49</b>	<b>49</b> (3,7%)
100	139	<b>63</b>	<b>63</b>	64	<b>63</b>	130	<b>63</b>	64	64	<b>63</b>	138	<b>63</b>	64	<b>63</b>	<b>63</b>	<b>63</b> (5,2%)
120	127	75	81	77	75	140	<b>74</b>	81	87	75	144	76	78	82	75	<b>74</b> (3,5%)

Tabela II  
TOPOLOGIA UNNET

N° SFCs	Degree				Closeness				Betweensness				CPLEX			
	Sol_Inicial	BL1	BL2	BL3	BL4	Sol_Inicial	BL1	BL2	BL3	BL4	Sol_Inicial	BL1		BL2	BL3	BL4
20	39	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b>	35	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b>	38	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b>	<b>14</b> (18,6%)
40	79	<b>26</b>	<b>26</b>	<b>26</b>	<b>26</b>	75	<b>26</b>	<b>26</b>	<b>26</b>	<b>26</b>	77	<b>26</b>	<b>26</b>	<b>26</b>	<b>26</b>	<b>26</b> (5,9%)
60	109	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>	112	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>	117	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b> (7,0%)
80	133	<b>50</b>	51	<b>50</b>	<b>50</b>	120	<b>50</b>	51	<b>50</b>	<b>50</b>	115	<b>50</b>	51	<b>50</b>	<b>50</b>	<b>50</b> (6,0%)
100	139	65	66	67	<b>64</b>	143	<b>64</b>	73	<b>64</b>	<b>64</b>	132	<b>64</b>	66	65	<b>64</b>	<b>64</b> (5,0%)
120	155	78	81	88	<b>77</b>	148	<b>77</b>	83	81	78	161	<b>77</b>	82	82	78	<b>77</b> (3,9%)

Tabela III  
TOPOLOGIA GEANT 2012 - CPLEX

SFCs	1(min)	10(min)	30(min)	60(min)
20	<b>14.0</b>	14.0	14.0	14.0
40	<b>25.0</b>	25.0	25.0	25.0
60	44.0	44.0	<b>38.0</b>	38.0
80	-	<b>49.0</b>	49.0	49.0
100	-	72.0	<b>63.0</b>	63.0
120	-	95.0	<b>74.0</b>	74.0

Tabela IV  
TOPOLOGIA UNNET - CPLEX

N° SFCs	1(min)	10(min)	30(min)	60(min)
20	<b>14.0</b>	14.0	14.0	14.0
40	28.0	<b>26.0</b>	26.0	26.0
60	65.0	<b>39.0</b>	39.0	39.0
80	131.0	<b>50.0</b>	50.0	50.0
100	-	66.0	<b>64.0</b>	64.0
120	90.0	90.0	<b>77.0</b>	77.0

## VI. CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi proposta uma heurística híbrida baseada em métricas de centralidade para a resolução do problema de Posicionamento e Encadeamento de Funções Virtuais de Rede. Diferentes abordagens para as buscas locais e para a heurística construtiva foram avaliadas, utilizando topologias reais e conjuntos de até 120 requisições SFC. De forma geral, todas as abordagens obtiveram boas soluções, próximas ou iguais ao resultado do CPLEX com limite de tempo de uma hora de execução, convergindo à solução final em um tempo significativamente menor. Para trabalhos futuros, considera-se a utilização de metaheurísticas, a realização de testes em outras instâncias e o tratamento do problema de maneira online.

## REFERÊNCIAS

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010. [Online]. Available: <https://doi.org/10.1016/j.comnet.2010.05.010>
- [2] Y. Zhu and M. H. Ammar, "Algorithms for assigning substrate network resources to virtual network components." in *INFOCOM*. IEEE, 2006.
- [3] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2009.10.017>
- [4] A. Fischer, J. Botero, M. Till Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys Tutorials*, *IEEE*, vol. 15, no. 4, pp. 1888–1906, Fourth 2013.
- [5] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping." in *INFOCOM*. IEEE, 2009, pp. 783–791.
- [6] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 98–106.
- [7] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.
- [8] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Computer Networks*, vol. 133, pp. 212 – 262, 2018.
- [9] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [10] A. Gupta, M. F. Habib, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, "On service-chaining strategies using virtual network functions in operator networks," *Computer Networks*, vol. 133, pp. 1 – 16, 2018. [Online]. Available: <https://doi.org/10.1016/j.comnet.2018.01.028>
- [11] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sept 2016.
- [12] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1346–1354.
- [13] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM)*, ser. CNSM '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 50–56. [Online]. Available: <http://dx.doi.org/10.1109/CNSM.2015.7367338>
- [14] G. Miotto, M. C. Luizelli, W. L. d. C. Cordeiro, and L. P. Gaspary, "Adaptive placement & chaining of virtual network functions with nfv-pear," *Journal of Internet Services and Applications*, vol. 10, no. 1, p. 3, Feb 2019.
- [15] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics*, 2nd ed. Springer Publishing Company, Incorporated, 2010.