Minimização da Latência no Posicionamento de Funções em Cloud RANs

Jean Lucas de Lima, Rodrigo S. Couto *

[1]Universidade do Estado do Rio de Janeiro - PEL - DETEL/FEN

jeanlucaslima@hotmail.com, rodrigo.couto@uerj.br

Abstract. *The concept of Cloud Radio Access Network (C-RAN) is to perform radiobase functions in a cloud infrastructure, which can be centralized or composed of several hierarchical levels. Thus, the stations act only as signal receivers, which are later processed in the cloud. Given the distance between the cloud and the stations, latency is a critical factor in C-RAN. In this work, we formulate a mixed integer linear programming problem to choose the placement of the radio functions in a C-RAN, to minimize the latency in a cloud with different levels of hierarchy. To solve the problem, this work proposes two heuristics and shows situations in which they reach the optimal result.*

Resumo. *O conceito de C-RAN (Cloud Radio Access Network) consiste em executar funções de estações rádio base em uma infraestrutura de nuvem, que pode ser centralizada ou composta por diversos níveis de hierarquia. Assim, as estações atuam apenas como receptores de sinais, que são posteriormente processados na nuvem. Dada a distância entre a nuvem e as estações, a latência é um fator crítico em C-RAN. Neste trabalho formula-se um problema de programação linear inteira mista para escolher o posicionamento das funções de rádio em uma C-RAN, de forma a minimizar a latência em uma nuvem com diferentes níveis de hierarquia. Para solução do problema, este trabalho propõe duas heurísticas e mostra situações nas quais essas alcançam o resultado ótimo.*

1. Introdução

A rede de acesso a radio (*Radio Access Network* - RAN) é uma infraestrutura de telecomunicações projetada para prover conectividade a dispositivos móveis em um sistema de rede celular. Assim, em uma RAN, o usuário conecta seu dispositivo a uma estação base (*Base Station* - BS) que, por sua vez, encaminha o tráfego para o núcleo da rede. Com o crescente consumo de dados de usuários finais, as RANs têm necessitado de crescente aumento de capacidade. Um dos caminhos para expandir uma RAN é aumentar o número de BSs, criando uma estrutura de rede de pequenas células, ou aumentar a capacidade das BSs já instaladas. Entretanto, esse aumento de capacidade pode reduzir a eficiência do uso de recursos, já que uma determinada BS pode se manter ociosa por longos períodos. Dados indicam que somente 15% a 20% das BSs operam a mais de 50% da sua capacidade total [Alyafawi et al., 2015]. Por exemplo, BSs instaladas nas proximidades de grandes estádios de futebol recebem uma alta quantidades de usuários em dias de evento. Entretanto, podem estar ociosas nos demais períodos. Uma alternativa é centralizar os

*Este trabalho foi realizado com recursos da FAPERJ, CNPq, CAPES e dos processos nº 15/24494-8 e nº 15/24490-2, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

recursos, que podem ser compartilhados por diversas BSs, semelhante ao que ocorre em Computação em Nuvem. Com isso, uma BS pode solicitar à infraestrutura centralizada apenas a quantidade de recursos necessária em um determinado momento. Essa solução é denominada Cloud RAN, ou C-RAN, e visa oferecer escalabilidade e flexibilidade em um sistema de rede celular [Checko et al., 2015].

Para oferecer serviços de rede celular, uma BS possui unidades de rádio (*Remote Radio Head* - RRH) e unidades de banda base (*Band Base Unit* - BBU) . A RRH realiza o processamento digital de sinais, a conversão analógico/digital e digital/analógico, e implementa interfaces com os meios de transmissão [Checko et al., 2015]. A BBU realiza tarefas de mais alto nível, como controle de acesso ao meio e correção de erros [Bartelt et al., 2015]. Em redes celulares tradicionais, a RRH e a BBU estão na mesma BS, integradas em um mesmo equipamento ou conectadas por fibra óptica. Em uma C-RAN, as RRH são instaladas nas BSs, que estão espalhadas geograficamente, enquanto as unidades de banda base são centralizadas e podem atender diversas BSs.

Uma desvantagem da C-RAN é a latência inserida entre a BBU e a RRH. Isso é crítico visto que essas unidades se comunicam por enlaces de poucos metros e, na C-RAN, utilizam enlaces que podem ter comprimentos da ordem de quilômetros [mar,]. Para solucionar esse problema, utiliza-se o conceito denominado *Dynamic* C-RAN [Dalla-Costa et al., 2017a]. Esse conceito é similar ao de Computação em Névoa [Coutinho et al., 2016]. Assim, nesse tipo de arquitetura, o processamento das funções de rádio é realizado de forma dinâmica e é distribuído em um conjunto de nuvens, que podem ser organizadas de forma hierárquica. Por exemplo, um conjunto de nuvens de borda, com pouca capacidade computacional, pode ser instalado em locais próximos a algumas BSs. Caso essas BSs necessitem de maior poder computacional, é possível utilizar nuvens mais centrais, com maior capacidade. Mais uma vez, recorre-se ao exemplo do estádio. Em dias comuns, os usuários que estão próximos ao estádio utilizam nuvens locais, que são suficientes para suprir as demandas. Em dias de evento, utilizam-se nuvens centrais para processar o tráfego dos usuários excedentes.

Uma forma de implementar uma *Dynamic* C-RAN é utilizar o conceito de Virtualização de Funções de Rede (*Network Functions Virtualization* - NFV) [Mijumbi et al., 2015]. Assim, cada função da BBU pode ser implementada como uma função de rede virtualizada (*Virtualized Network Function* - VNF). As VNFs são então distribuídas pela hierarquia de nuvens de acordo com a demanda. Essa distribuição deve ser realizada por algoritmos de posicionamento de VNFs. No caso específico de *Dynamic* C-RAN, [Dalla-Costa et al., 2017b] propõem um modelo de programação linear inteira mista (*Mixed Integer Linear Programming* - MILP) para escolher o posicionamento de VNFs de forma a minimizar o uso de banda na rede.

Este trabalho formula um problema MILP baseado em [Dalla-Costa et al., 2017b], para posicionar funções em uma *Dynamic* C-RAN. Esse problema minimiza a latência média da rede e considera restrições de capacidade das nuvens. Além disso, dada a complexidade de tempo do MILP, este trabalho propõe duas heurísticas para a solução do problema, mostrando situações nas quais essas alcançam o resultado ótimo. Uma das heurísticas escolhe o posicionamento baseado na ordenação dos valores de latência, enquanto a outra desconsidera essa ordem. A heurística sem ordenação possui complexidade $O(n)$, enquanto a com ordenação possui complexidade $O(n \log n)$. Os resultados deste

trabalho mostram que, no cenário considerado, não há ganhos significativos em ordenar a latência.

O trabalho está estruturado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados, enquanto a Seção 3 apresenta uma visão geral sobre C-RAN. O problema de otimização é formulado na Seção 4. As Seções 5 e 6 descrevem as heurísticas propostas e apresentam resultados. Finalmente, a Seção 7 conclui o trabalho e aponta direções futuras.

2. Trabalhos Relacionados

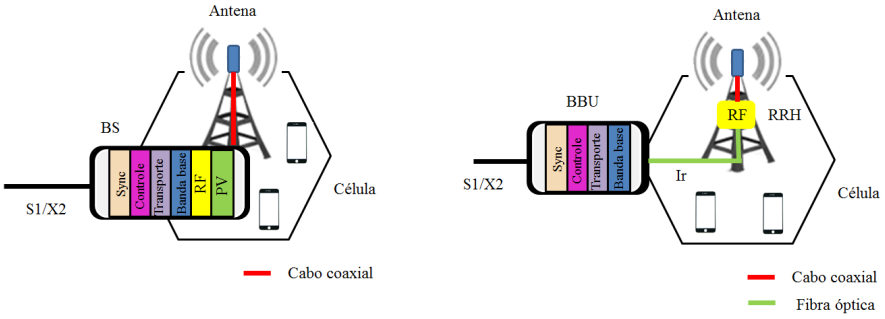
A proposta de C-RAN visa centralizar a unidade de banda base (BBU), separando-a da unidade de rádio (RRH). A BBU, por sua vez, pode oferecer os serviços como um só bloco ou divididos em diversos blocos. Nessa última opção, as funcionalidades da BBU são decompostas em diversas funções, que podem ser VNFs de uma infraestrutura NFV. Diversos trabalhos mostram os benefícios de dividir as funções da BBU [Wubben et al., 2014, Bartelt et al., 2015]. Por exemplo, [Bartelt et al., 2015] mostram que a divisão em funções possibilita reduzir a taxa de dadas necessária na rede.

Quando as funções da BBU são divididas, um algoritmo deve determinar o posicionamento de cada VNF na rede. O posicionamento de VNFs é um assunto bastante estudado na área de NFV [Queiroz et al., 2017, Luizelli et al., 2015]. Entretanto, poucos trabalhos consideram o posicionamento em NFV para o cenário específico de C-RAN [Herrera e Botero, 2016]. Nessa linha, os trabalhos [Dalla-Costa et al., 2017a, Dalla-Costa et al., 2017b] formulam um problema de otimização para minimizar o uso de banda nos enlaces entre as nuvens e privilegiar as nuvens de borda, que são as nuvens mais próximas das BSs. Apesar de privilegiar a nuvem de borda reduzir a latência da rede, essa solução é restritiva, uma vez que ignora a baixa latência de nuvens regionais, isto é, que se localizam entre as centrais e as de borda. Assim, este trabalho se baseia em [Dalla-Costa et al., 2017b] para formular um novo problema que considera a latência na função objetivo e trata a banda como uma restrição. Com isso, privilegiam-se também nuvens intermediárias. Tratar a banda como restrição é justificado pelo fato de as operadoras de celular possuírem redes bem provisionadas. Além disso, este trabalho propôs heurísticas para solucionar o problema formulado, diferente do trabalho da literatura, no qual apenas a solução por meio de um otimizador MILP é apresentada.

3. Evolução da Arquitetura de Rede RAN

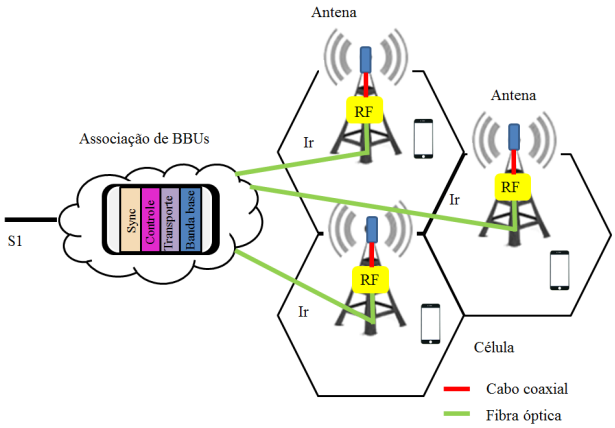
Na arquitetura RAN tradicional, ilustrada na Figura 1(a), a RRH e a BBU estão integradas em um único elemento físico. A interface lógica opcional X2 é definida entre estações rádio base, enquanto a interface S1 conecta os dispositivos móveis à rede celular. Uma evolução da RAN tradicional é separar a unidade de rádio da unidade de banda base. Essa arquitetura, indicada na na Figura 1(b), contém um unidade de rádio remota (*Remote Radio Head* - RRH) separada de uma unidade de banda base (*Band Base Unit* - BBU). As RRHs podem ser colocadas em postes ou telhados, aproveitando refrigeração eficiente e, consequentemente, proporcionando economia com ar condicionado nos locais das BBUs. A C-RAN, mostrada na Figura 1(c), evolui essa arquitetura, centralizando as funções das BBUs em uma nuvem para tornar eficiente a utilização de recursos. Para utilizar as unidades de processamento de banda base de modo a proporcionar seu melhor aproveitamento em questão de capacidade, faz-se necessária sua associação com outras

BBUs para que várias RRHs utilizem de uma mesma nuvem com BBUs. Assim, evita-se o desperdício de processamento, como ocorre com a arquitetura RAN tradicional. As RRHs são conectadas às associações de BBUs por meio de fibras ópticas, sendo que as RRHs podem compartilhar as BBUs dinamicamente. Assim, usuários de diferentes células podem usar o serviço fornecido por uma BBU, melhorando a utilização de recurso de banda base [Wang et al., 2014]. A Associação de BBUs pode ser centralizada ou, como no caso da Dynamic *C-RAN*, pode ser distribuída em diversas nuvens.



(a) Arquitetura RAN tradicional.

(b) Arquitetura RAN com RRHs.



(c) Arquitetura C-RAN com RRHs.

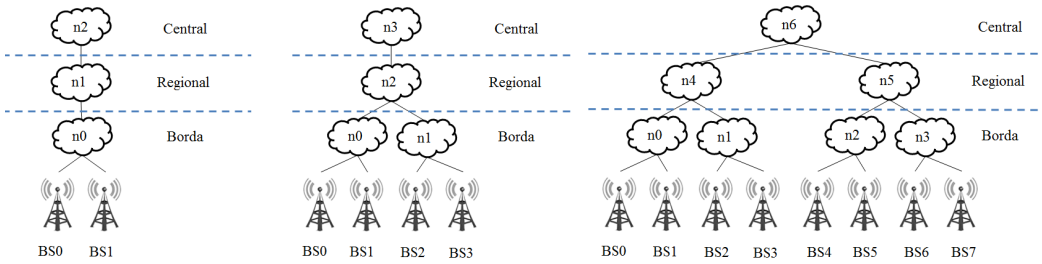
Figura 1. Evolução da arquitetura RAN. Figura adaptada de [Checko et al., 2015].

4. Modelagem do Problema

No problema proposto considera-se que, para o funcionamento de uma BS, todas suas funções $f \in \mathcal{F}$ devem ser posicionadas na infraestrutura C-RAN. Cada função pode ser implementada como uma VNF em uma infraestrutura NFV [Mijumbi et al., 2015]. Cada VNF consome, na nuvem que está associada, uma quantidade de banda e uma fatia de seus recursos destinados à hospedagem de VNFs, como memória e processamento.

O problema formulado tem como objetivo minimizar a latência percebida por uma BS. Cada uma das funções de uma BS é executada em uma determinada nuvem, em um determinado nível da hierarquia. A hierarquia das nuvens é uma árvore, como mostram os exemplos da Figura 2. Nesses exemplos, existem três níveis de nuvem. Quanto mais baixo o nível da nuvem, mais baixa é sua latência para a BS. Além disso, espera-se que, quanto mais baixo o nível, menor a capacidade de processamento e banda da nuvem. Devido à

estrutura em árvore, cada BS só pode ser atendida por uma nuvem em cada nível. Por exemplo, na Figura 2(c) a BS4 pode ser atendida pelas nuvens n2, n5 e n6.



(a) Topologia 1. (b) Topologia 2. (c) Topologia 3.

Figura 2. Topologias Utilizadas.

Define-se, neste trabalho, a latência l_a percebida por uma BS a como a latência dessa BS até a função mais distante possível. Por exemplo, Na Figura 2(a) se a BS0 possuir uma função em n0, uma em n1 e outra em n2, a latência l_a percebida por essa BS é a sua latência do seu caminho até a nuvem $n2$. Se a BS só possuísse funções em n0 e n1, sua latência l_a seria a latência até nuvem $n1$. Assim, o problema formulado visa minimizar a média entre as latências l_a de todas as BSs da infraestrutura. A Tabela 1 lista as variáveis, parâmetros e conjuntos utilizados na formulação do problema, bem como as notações utilizadas e suas descrições. A formulação do MILP é apresentada a seguir.

$$\text{minimizar:} \quad \frac{1}{|\mathcal{A}|} \sum_{\forall a \in \mathcal{A}} l_a. \tag{1}$$

$$\text{sujeito a:} \quad \sum_{n \in \mathcal{D}_a} d_{f,n,a} = 1 \qquad \forall f \in \mathcal{F}, \forall a \in \mathcal{A}; \tag{2}$$

$$\sum_{f \in \mathcal{F}, a \in \mathcal{A} | n \in \mathcal{D}_a} d_{f,n,a} \cdot D_V(f,n,a) \leq C_V^n \quad \forall n \in \mathcal{N}; \tag{3}$$

$$\sum_{f \in \mathcal{F}, a \in \mathcal{A} | n \in \mathcal{D}_a} d_{f,n,a} \cdot D_B(f,n,a) \leq C_B^n \quad \forall n \in \mathcal{N}; \tag{4}$$

$$l_a \geq d_{f,n,a} \cdot D_L(n,a) \quad \forall f \in \mathcal{F}, a \in \mathcal{A}, n \in \mathcal{D}_a; \tag{5}$$

$$l_a \leq l_{max} \quad \forall a \in \mathcal{A}; \tag{6}$$

$$l_a \in \mathbb{R}^+ \quad ; \quad d_{f,n,a} \in \{0,1\}. \tag{7}$$

A Equação 1 é a função objetivo, que consiste em minimizar a latência média, considerando todas a latência l_a de todas as BSs da infraestrutura. Note que $|\mathcal{A}|$ é o número total de BSs da infraestrutura. A Equação 2 define que cada função f de uma BS a só pode ser posicionada em uma única nuvem n. Note que $\mathcal{D}_a$ é o conjunto das nuvens que podem atender a BS a. Por exemplo, na Figura 2(c), o $\mathcal{D}_1$ correspondente à BS1 (isto é, $a = 1$), possui as nuvens n0, n4 e n6. A variável de decisão $d_{f,n,a}$ é a saída do problema, que indica se a função f, da BS a, está posicionada na nuvem n.

As Equações 3 e 4 asseguram que o posicionamento de funções respeita, respectivamente, as capacidades de hospedagem de funções e banda de cada nuvem. Neste trabalho, seguindo a mesma nomenclatura de [Dalla-Costa et al., 2017b], a capacidade de

Tabela 1. Notações utilizadas no problema.

Notação	Descrição	Tipo
$\mathcal{N}$	Nuvens existentes na infraestrutura	Conjunto
$\mathcal{A}$	BSs existentes na infraestrutura	Conjunto
$\mathcal{F}$	Tipos de funções de rede	Conjunto
$\mathcal{D}_a$	Nuvens que podem receber funções da BS a	Conjunto
C_V^n	Capacidade em VDUs da nuvem n	Parâmetro
C_B^n	Capacidade de banda do enlace de saída da nuvem n	Parâmetro
$D_V(f,n,a)$	Demanda de VDUs da função f, da BS a, na nuvem n	Parâmetro
$D_B(f,n,a)$	Demanda de banda da função f, da BS a, na nuvem n	Parâmetro
$D_L(n,a)$	Latência entre a nuvem n e a BS a	Parâmetro
l_{max}	Latência máxima permitida na C-RAN	Parâmetro
$d_{f,n,a}$	Variável binária que indica se a função f, da BS a, deve ser posicionada na nuvem n	variável
l_a	Latência percebida pela BS a	Variável

hospedagem de uma nuvem n é medida em número de VDUs (*Virtual Data Units*), dado por C_V^n. Uma VDU é uma fatia indivisível da nuvem, considerando suas capacidades de memória e processamento. Uma função de rede ocupa então um determinado número de VDUs. Assim, o parâmetro $D_V\left(f,n,a\right)$ consiste na quantidade de VDUs necessárias para hospedar a função f, da BS a, na nuvem n. Em relação à banda, considera-se como C_B^n a capacidade do enlace de saída da nuvem n para a nuvem hierarquicamente superior. No caso da nuvem central, essa capacidade pode ser considerada como infinita ou mesmo igual à capacidade do enlace de conexão da RAN com outras redes, como o núcleo da rede celular. O parâmetro $D_B\left(f,n,a\right)$ é a quantidade de banda que a função f, da BS a, consome no enlace de saída da nuvem n. Por simplicidade, é possível considerar que os parâmetros $D_V\left(f,n,a\right)$ e $D_B\left(f,n,a\right)$ são independentes de n e a, como realizado mais adiante neste trabalho.

A Equação 5 calcula a variável l_a para cada BS a, considerando os valores de latência $D_L\left(n,a\right)$ entre a nuvem n e a BS a. Note que essa equação força l_a a ser maior ou igual à maior latência entre a BS a e uma nuvem n que hospeda alguma de suas funções. Entretanto, na solução ótima, l_a será exatamente igual a essa latência mencionada, visto que a função objetivo da Equação 1 minimiza os valores de l_a, atribuindo-a o menor valor possível. A Equação 6 impede o posicionamento de funções em nuvens que possuam uma latência com valor maior do que o máximo permitido, dado por L_{max}. Por fim, a Equação 7 descreve o domínio das variáveis.

Por se tratar de um problema com variáveis binárias, a solução do problema por meio de um otimizador MILP não escala com o número de nuvens e BSs. Além disso, o problema deve ser executado periodicamente para agir de acordo com o aumento da demanda das BSs. Consequentemente, sua solução deve ser realizada o mais rápido possível. Para tal, propõem-se duas heurísticas para a solução do problema, descritas a seguir.

5. Heurística sem Ordenação de Latência

A ideia da heurística proposta é, para cada BS, tentar posicionar todas as suas funções na sua nuvem de borda. Caso não seja possível por falta de capacidade de banda ou de VDU, tenta-se posicionar as funções restantes na nuvem regional. Caso ainda não seja possível, posicionam-se as funções na nuvem central. A cada posicionamento realizado, subtraem-se as demandas $D_V(f,n,a)$ e $D_B(f,n,a)$ dos parâmetros de capacidade C_V^n e C_B^n da nuvem escolhida. Quando todas as funções de uma BS são alocadas, atende-se

uma próxima BS, posicionando suas funções nas nuvens da infraestrutura. Essa heurística considera que a rede é homogênea, ou seja, cada nuvem possui os mesmos valores de latência para todas as BSs que ela pode atender. Nesse caso, o parâmetro $D_L(n, a)$ só depende do nível da nuvem n na hierarquia e não da BS utilizada. Essa consideração permite escolher arbitrariamente a ordem na qual as BSs são atendidas pela heurística, sem priorizar BSs com menores latência. Mais adiante, na Seção 6, propõe-se neste trabalho uma heurística que posiciona as BSs de acordo com a latência dos enlaces. A seguir descreve-se formalmente o algoritmo da heurística proposta para redes homogêneas.

5.1. Descrição do Algoritmo

O Algoritmo 1 detalha a heurística para redes homogêneas. A linhas 1 itera para todas as BS, enquanto a linha 2 itera para todos os tipos de função. Na linha 3 itera-se entre as nuvens possíveis para a BS a. Por definição, o conjunto $\mathcal{D}_a$ é ordenado da nuvem de menor hierarquia até a nuvem de maior hierarquia. Assim, para três níveis de hierarquia, o algoritmo tenta posicionar a função inicialmente na nuvem de borda, em seguida a regional e finalmente a central. A linha 4 verifica se a função f já foi posicionada em alguma nuvem. Para tal, utiliza-se a variável auxiliar $p_{f,a}$, que indica se a função f da BS a já foi posicionada. Se a função não estiver posicionada, a linha 5 verifica se há VDUs e banda disponíveis na nuvem n e se a latência para essa nuvem é menor ou igual ao maior valor de latência permitido l_{max}. Caso isso seja afirmativo, as linhas 6 e 7 atualizam as variáveis $p_{f,a}$ e $d_{f,n,a}$. Finalmente, nas linhas 8 e 9 são decrementadas da capacidade daquela nuvem a demanda de VDUs e de banda da função posicionada.

Algoritmo 1: Heurística para C-RANs Homogêneas

Entrada: $\mathcal{A}$, $\mathcal{F}$, $\mathcal{D}_a$, C_V^n, C_B^n, $D_V(f,n,a)$, $D_B(f,n,a)$, $D_L(n,a)$, l_{max}
Saída: $d_{f,n,a}$

```
1  para a ∈ A faça
2      para f ∈ F faça
3          para n ∈ D_a faça
4              se p_{f,a} = 0 então
5                  se D_V(f,n,a) ≤ C_V^n e D_B(f,n,a) ≤ C_B^n e D_L(n,a) < l_max então
6                      p_{f,a} ← 1;
7                      d_{f,n,a} ← 1;
8                      C_V^n ← C_V^n − D_V(f,n,a);
9                      C_B^n ← C_B^n − D_B(f,n,a);
10                 fim
11             fim
12         fim
13     fim
14 fim
```

Utilizando as variáveis calculadas pelo Algoritmo 1, é possível obter a latência média (isto é, a função objetivo da Equação 1) fazendo:

$$l_{med} = \frac{1}{|\mathcal{A}|} \sum_{\forall a \in \mathcal{A}} l_a = \frac{1}{|\mathcal{A}|} \sum_{\forall a \in \mathcal{A}} \max_{\forall f \in \mathcal{F}, n \in \mathcal{D}_a} (d_{f,n,a} \cdot D_L(n,a)) \tag{8}$$

O algoritmo apresentado anteriormente possui número de passos proporcional ao número de BSs, dado por $|\mathcal{A}|$, número de tipos de função, dado por $|\mathcal{F}|$, e número de

nuvens que atendem uma BS, dado por $|\mathcal{D}_a|$. Para $|\mathcal{D}_a|$ considera-se que todas as BSs podem ser atendidas pelo mesmo número de nuvens, o que sempre ocorre em topologias em árvore, consideradas neste trabalho. Assim, utilizando a notação O para análise de pior caso, tem-se que o algoritmo possui complexidade de tempo $O(|\mathcal{A}||\mathcal{F}||\mathcal{D}_a|)$. Entretanto, o número de tipos de função $|\mathcal{F}|$ é constante para uma determinada tecnologia de comunicação celular, sendo cinco para LTE na abordagem de [Wubben et al., 2014]. Da mesma forma, $|\mathcal{D}_a|$ é constante para uma determinada arquitetura C-RAN, sendo igual a três nos trabalhos da literatura [Dalla-Costa et al., 2017b, Dalla-Costa et al., 2017a]. Assim, utilizando as propriedades da notação O, removem-se as constantes da expressão de complexidade, concluindo que a heurística proposta possui complexidade $O(|\mathcal{A}|)$.

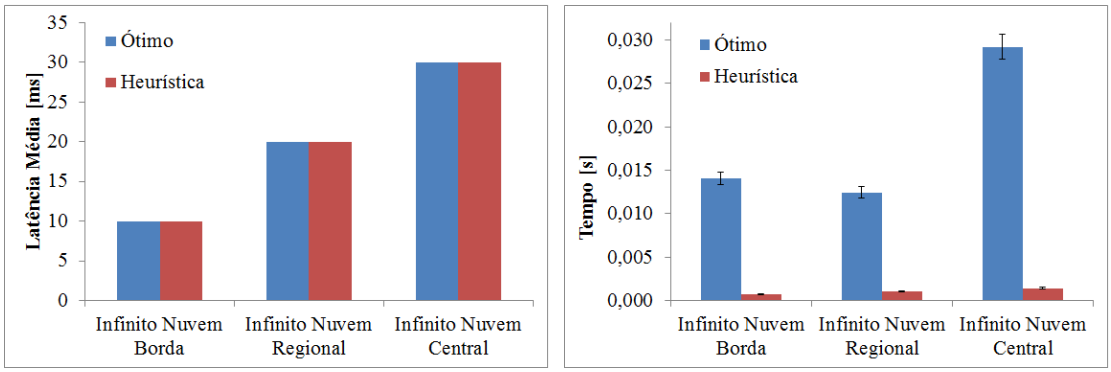
5.2. Resultados

Esta seção avalia a heurística proposta em comparação com a solução ótima. Para a solução ótima, executa-se o problema da Seção 4 utilizando o GLPK [GNU, 2017], enquanto o Algoritmo 1 é implementado em Python. Os resultados são obtidos em uma máquina virtual Ubuntu 16.04 com dois núcleos de CPU e 1GB de RAM. Essa máquina virtual executa em uma máquina com CPU Intel® Core™ i5-2450M e 4G de RAM. Para cada topologia, são consideradas diferentes configurações dos recursos de banda C_V^n e de VDUs C_B^n de cada nuvem n. Para simplificar a análise, os recursos de banda são mantidos como ilimitados. Os recursos de VDUs são distribuídos, em cada topologia, de acordo com três configurações. Na primeira, denominada Infinito Nuvem de Borda, todas as nuvens possuem capacidade infinita. Na segunda, denominada Infinito Nuvem Regional, as nuvens de borda são limitadas, mas as demais possuem capacidade infinita. Na configuração Infinito Nuvem Central apenas a nuvem central possui capacidade infinita. A Tabela 2 mostra a capacidade em VDUs de cada nuvem utilizada e seus valores de latência para as BSs. Note, pelos valores de latência da Tabela 2, que os enlaces de todas as topologias consideradas possuem 10 ms de latência. Esses valores são arbitrários, podendo existir C-RANs com requisitos estritos de latência, com enlaces da ordem de microssegundos [mar,]. Entretanto, a ordem de grandeza desses valores não afeta a análise apresentada neste trabalho.

Tabela 2. Parâmetros utilizados na avaliação.

Topologia	Configuração	Capacidade C_V^n (VDUs)			Latência $D_L(n,a)$ (ms)		
		Borda	Regional	Central	Borda	Regional	Central
1	Infinito Nuvem Borda	∞	∞	∞	10	20	30
	Infinito Nuvem Regional	2	∞	∞	10	20	30
	Infinito Nuvem Central	2	2	∞	10	20	30
2	Infinito Nuvem Borda	∞	∞	∞	10	20	30
	Infinito Nuvem Regional	2	∞	∞	10	20	30
	Infinito Nuvem Central	2	2	∞	10	20	30
3	Infinito Nuvem Borda	∞	∞	∞	10	20	30
	Infinito Nuvem Regional	5	∞	∞	10	20	30
	Infinito Nuvem Central	5	5	∞	10	20	30

Para a Topologia 1, a Figura 3(a) mostra o valor da função objetivo (isto é, latência média), definida na Equação 8. Os resultados mostram que o valor da função objetivo da



(a) Latência média. (b) Tempo de execução.

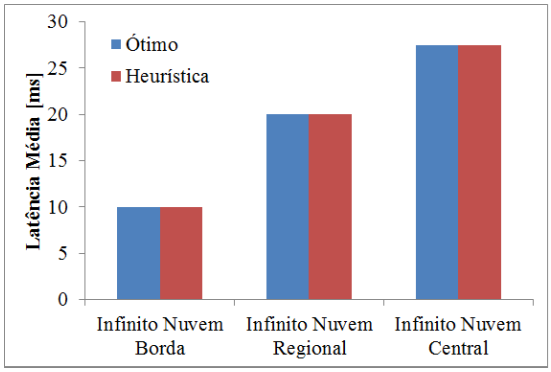
Figura 3. Resultados para a Topologia 1 homogênea.

solução ótima é igual ao do valor obtido pela heurística. Além disso, é possível observar, como esperado, que a latência aumenta à medida que diminuem-se os recursos das nuvens de menor hierarquia. A Figura 3(b) mostra o tempo de execução da solução ótima e da heurística, representado com nível de confiança de 95% após a coleta de 10 rodadas. É possível notar que, apesar de os tempos serem pequenos para os dois casos, a heurística possui tempos de execução consideravelmente inferiores. É esperado também que, para o caso ótimo, ao reduzirem-se os recursos das nuvens de menor hierarquia, mais possibilidades são acrescentadas ao problema, aumentando o tempo de solução. Por exemplo, para o cenário Infinito Nuvem Borda, apenas a nuvem de borda já satisfaz a solução do problema. Entretanto, a Figura 3(b) não mostra esse comportamento entre os cenários Infinito Nuvem Borda e Infinito Nuvem Regional. Isso ocorre devido aos pequenos tempos de execução, que podem acarretar falta de precisão na medição. Esse comportamento está mais evidente nos resultados das Topologias 2 e 3, apresentados a seguir.

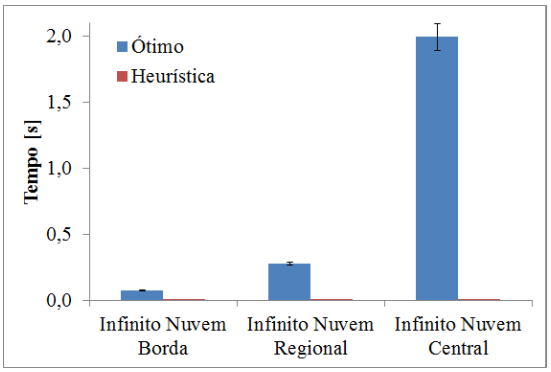
As Figuras 4 e 5 apresentam os resultados para as Topologias 2 e 3 respectivamente. Note que, da mesma forma que a Topologia 1, as Figuras 4(a) e 5(a) mostram que a heurística possui a mesma latência média da solução ótima. Nos tempos de execução, apresentados nas Figuras 4(b) e 5(b), é possível observar melhor que limitar os níveis mais baixos da hierarquia aumenta o espaço de solução, aumentando o tempo de execução. Os resultados do tempo de execução também mostram que a heurística obtém a solução de forma significativamente mais rápida, sendo imperceptíveis nos gráficos traçados. Por fim, comparando os tempos obtidos nas Figuras 3(b), 4(b) e 5(b), mostra-se que o tempo de execução da solução ótima aumenta com o número de BSs na topologia.

6. Heurística com Ordenação de Latência

Apesar de os resultados da heurística da Seção 5 serem iguais aos da solução ótima, isso pode não ocorrer em redes heterogêneas, nas quais os enlaces da rede possuem valores diferentes de latência (p.ex, na Figura 2(c) a nuvem n4 possuir latência de 60 ms com a BS0, mas 20 ms com a BS3). Isso ocorre pois, no Algoritmo 1, as BSs são alocadas sem uma ordem pré-definida. Assim, o algoritmo pode posicionar funções de uma BS que possui alta latência com uma nuvem n e esgotar seus recursos antes de as funções de uma BS com baixa latência serem posicionadas nessa mesma nuvem. Para verificar o caso de redes heterogêneas, propõe-se neste trabalho uma segunda heurística que alocas as funções

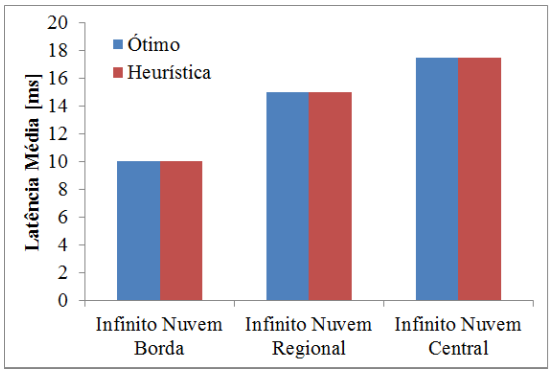


(a) Latência média.

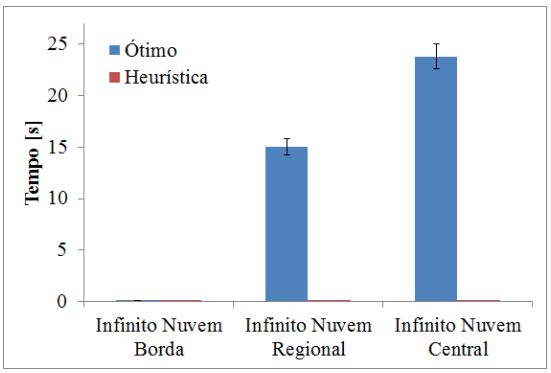


(b) Tempo de execução.

Figura 4. Resultados para a Topologia 2 homogênea.



(a) Latência média.



(b) Tempo de execução.

Figura 5. Resultados para a Topologia 3 homogênea.

em ordem crescente da latência entre uma BS e sua respectiva nuvem. Para tal, ordenam-se todos os possíveis pares (n, a), onde n é uma nuvem e a é uma BS. Após isso, alocam-se as funções na ordem desses pares, privilegiando assim as BSs que possuem menores latências com suas nuvens. A seguir descreve-se o algoritmo da heurística proposta.

6.1. Descrição do Algoritmo

O Algoritmo 2 detalha a heurística para redes heterogêneas. A linha 1 recebe todas as BSs e nuvens e, utilizando os parâmetros $D_L(n, a)$ para cada par (n,a), constrói a lista ordenada. Assim, a linha 2 itera para todos os pares (n,a) na ordem crescente de sua latência. A partir da linha 4, o Algoritmo 2 executa os mesmos passos do Algoritmo 1.

A complexidade do conjunto de passos da linha 2 a 13 é a mesma do Algoritmo 1, ou seja, $O(\mathcal{A})$. Entretanto, é necessário considerar também a complexidade da ordenação na linha 1. Considerando um algoritmo simples de ordenação como o *Bubble Sort*, sua complexidade de pior caso é $O(n^2)$ [Szwarcfiter e Markenzon, 2013]. Assim, como a lista possui $|\mathcal{A}| \cdot |\mathcal{D}_a|$ elementos e $|\mathcal{D}_a|$ é considerado constante, a complexidade do *Bubble Sort* é $O(|\mathcal{A}|^2)$. Consequentemente, a complexidade de todo o Algoritmo 2 é $O(|\mathcal{A}|^2 + |\mathcal{A}|)$. Utilizando as propriedades da notação O, considera-se apenas o termo de maior complexidade. Assim, a complexidade do Algoritmo 2 utilizando *Bubble Sort* é $O(|\mathcal{A}|^2)$.

Apesar do exposto anteriormente, é possível utilizar algoritmos de busca mais

Algoritmo 2: Heurística Dynamic C-RANs Heterogêneas

Entrada: $\mathcal{A}$, $\mathcal{F}$, $\mathcal{D}_a$, $\mathcal{N}$, C_V^n, C_B^n, $D_V(f,n,a)$, $D_B(f,n,a)$, $D_L(n,a)$, l_{max}
Saída: $d_{f,n,a}$

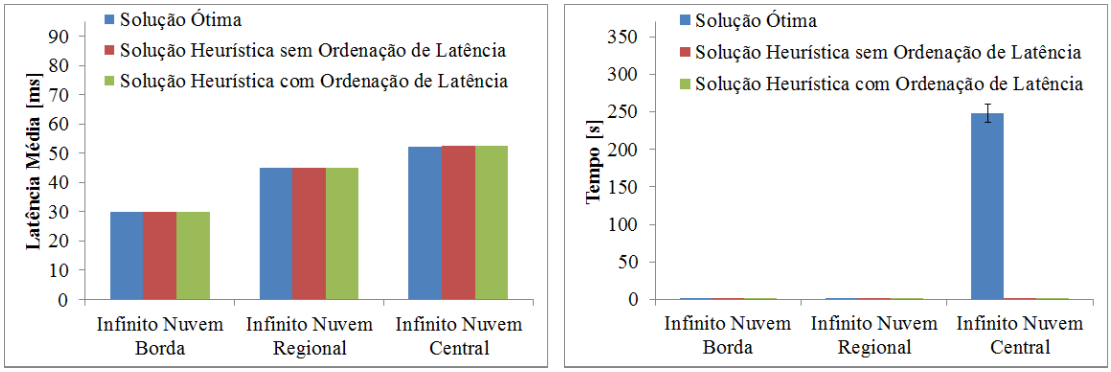
```
listaOrdenada = constroiListaParesOrdenados (A,N);
para (n, a) ∈ listaOrdenada faça
    para f ∈ F faça
        se p_{f,a} = 0 então
            se D_V(f,n,a) ≤ C_V^n e D_B(f,n,a) ≤ C_B^n e D_L(n,a) < l_max então
                p_{f,a} ← 1;
                d_{f,n,a} ← 1;
                C_V^n ← C_V^n − D_V(f,n,a);
                C_B^n ← C_B^n − D_B(f,n,a);
            fim
        fim
    fim
fim
```

sofisticados, como o *Heap Sort*. Esse algoritmo possui complexidade de pior caso $O(n \log n)$ [Szwarcfiter e Markenzon, 2013]. Assim, a complexidade do Algoritmo 2 se torna $O(|\mathcal{A}| \log(|\mathcal{A}|) + |\mathcal{A}|)$. Utilizando as propriedades da notação O, tem-se que Algoritmo 2 com *Heap Sort* possui complexidade de pior caso $O(|\mathcal{A}| \log(|\mathcal{A}|))$.

6.2. Resultados

Esta seção analisa os resultados obtidos para a solução ótima, para o Algoritmo 1 e para o Algoritmo 2. Esses resultados são obtidos para a Topologia 3, da Figura 2(c), com os mesmos parâmetros da Tabela 2. Entretanto, todos os valores de latência do enlace à esquerda de uma nuvem são alterados para 50 ms, caracterizando uma nuvem heterogênea. Em relação às capacidades, as três configurações analisadas anteriormente são utilizadas. O resultados, presentes na Figura 6, mostram que heurística com ordenação (isto é, o Algoritmo 2) possui latência média igual à solução ótima em todos os casos avaliados. No entanto, ao contrário do esperado inicialmente, constatou-se também que a heurística sem ordenação de latências oferece o mesmo resultado do valor ótimo. Para esses cenários avaliados, a ordenação de latências não interferiu na latência média da topologia. Por fim, é possível notar que os tempos de execução da solução ótima podem ser consideravelmente maiores quando comparados aos tempos de execução das heurísticas.

Para avaliar se existe uma configuração de latência na qual o desempenho dos algoritmos diferem entre si, utiliza-se a Topologia 3 com as diferentes configurações de capacidade da Tabela 2. Em relação aos valores de latência, para cada enlace da topologia, multiplica-se uma constante de 30 ms por um valor obtido a partir de uma variável aleatória log-normal. Utiliza-se essa distribuição por sua fácil parametrização e por sempre retornar valores positivos. A log-normal consiste em uma distribuição na qual o logaritmo da variável aleatória segue uma distribuição normal com média μ e desvio padrão σ. Para uma baixa variação da latência entre os enlaces (isto é, baixa heterogeneidade), utiliza-se $\sigma = 0.125$. Já para uma alta variação (isto é, alta heterogeneidade), utiliza-se $\sigma = 1$. São realizadas 100 amostras do experimento para cada valor de σ. Os resultados obtidos, omitidos por questões de espaço e concisão, mostram que em todas as amostras o resultado dos dois algoritmos propostos é igual ao ótimo. Assim, essa é uma evidência



(a) Latência média. (b) Tempo de execução.

Figura 6. Resultados para a Topologia 3 heterogênea no Cenário 3.

experimental de que a ordenação de latências pode não causar impacto em uma rede com latência heterogênea e capacidades uniformes. Entretanto, é necessário, em um trabalho futuro, validar essa hipótese de forma teórica.

Apesar dos resultados anteriores, se as capacidades das nuvens também forem heterogêneas, existem casos particulares nos quais o valor de latência obtido pelas heurísticas é superior ao da solução ótima. Além disso, há casos nos quais a heurística com ordenação possui valores de latência inferiores ao da heurística sem ordenação e há casos nos quais essa situação se inverte. Para mostrar esse comportamento, realiza-se um experimento com a mesma variação de latência utilizada anteriormente, mas escolhendo de forma aleatória as capacidades C_V das nuvens regionais e das nuvens de borda. A capacidade da nuvem central é fixada em 40, ou seja, a nuvem central possui capacidade para, no pior dos casos, hospedar todas as cinco funções de cada uma das oito BSs. A capacidade de cada nuvem de borda é escolhida por uma distribuição uniforme entre 0 e $40M_{\text{bor}}$. A capacidade de cada nuvem regional é escolhida também por por uma distribuição uniforme entre 0 e $40M_{\text{reg}}$. Ou seja, M_{bor} e M_{reg} são parâmetros que regulam o tamanho de cada nuvem em relação à capacidade da nuvem central. São realizados experimentos para diferentes combinações de M_{bor} e M_{reg}, sendo 100 amostras para cada combinação. Para cada amostra, calcula-se, para cada algoritmo, o desvio relativo percentual em relação ao ótimo, dado por:

$$D_{\text{relativo}} = \frac{z^* - z}{z^*} \times 100\%, \tag{9}$$

onde z é o valor da função objetivo da solução ótima e z^* é o valor da função objetivo para o algoritmo avaliado.

As Tabelas 3 e 4 mostram, respectivamente, os resultados obtidos para variações de latência de $\sigma = 0.125$ e $\sigma = 1$, com diferentes valores de M_{bor} e M_{reg}. Os desvios relativos são apresentados com suas médias e intervalos de confiança com nível de 95%. Os resultados mostram que há um baixo desvio em relação ao ótimo e, considerando o intervalo de confiança, há pouca ou nenhuma diferença entre as duas heurísticas propostas neste trabalho. Assim, essa é uma evidência experimental de que a ordenação por latência não oferece melhora significativa na heurística proposta.

Tabela 3. Desvio relativo percentual para uma variação de latência $\sigma = 0.125$

M_{bor} \ M_{reg}	0.04		0.08		0.16	
	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
0.02	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
0.04	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.31 ± 0.22	0.5 ± 0.27
0.08	0.0 ± 0.0	0.0 ± 0.0	0.73 ± 0.31	1.87 ± 0.49	1.8 ± 0.46	2.33 ± 0.51
0.16	0.5 ± 0.31	0.65 ± 0.37	1.32 ± 0.51	2.26 ± 0.64	1.82 ± 0.47	2.99 ± 0.65

Tabela 4. Desvio relativo percentual para uma variação de latência $\sigma = 1$

M_{bor} \ M_{reg}	0.04		0.08		0.16	
	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
0.02	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
0.04	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.32 ± 0.21	0.49 ± 0.31
0.08	0.0 ± 0.0	0.0 ± 0.0	0.83 ± 0.4	1.54 ± 0.58	1.15 ± 0.47	2.3 ± 0.61
0.16	0.64 ± 0.37	0.69 ± 0.31	1.43 ± 0.56	2.2 ± 0.74	1.81 ± 0.66	2.9 ± 0.78

7. Conclusões e Trabalhos Futuros

A preocupação com a latência é um fator importante em uma infraestrutura C-RAN. Assim, é possível utilizar esquemas do tipo *Dynamic* C-RAN para aproximar a nuvem dos usuários e decidir, dinamicamente, a posição das funções da rede. Esse esquema possui nuvens organizadas de forma hierárquica, que são responsáveis por executar funções da C-RAN, como controle de acesso ao meio e correção de erros. Por exemplo, é possível organizar a infraestrutura nos níveis de borda, regional e central. Nesse tipo de infraestrutura, é importante desenvolver algoritmos de otimização que posicionam as funções da rede nas diversas nuvens da hierarquia.

Este trabalho complementou um modelo de otimização presente na literatura, propondo uma formulação que possui o objetivo de minimizar a latência média na rede. Além disso, este trabalho propôs heurísticas para solucionar o problema formulado. Os resultados mostraram que quando a capacidade das nuvens é homogênea, mesmo com valores heterogêneos de latência, as duas heurísticas alcançam o valor ótimo. Entretanto, quando tanto as capacidades em VDUs quanto os valores de latência são heterogêneos, as heurísticas podem se distanciar da solução ótima. Nesses casos, os resultados mostram que o desvio relativo em relação ao ótimo é próximo para as duas heurísticas. Assim, esse é um indício de que, no cenário analisado, a ordenação de latência não possui vantagens significativas.

Apesar de mostrar situações nas quais as heurísticas propostas alcançam o valor ótimo, este trabalho é baseado apenas em evidências experimentais. Assim, como trabalhos futuros pretende-se desenvolver formulações que mostram como a função objetivo das heurísticas se distancia do ótimo em função dos parâmetros da rede. Além disso, pretende-se considerar em um trabalho futuro a dinamicidade da rede. Ou seja, o problema deverá ser executado periodicamente em função da mudança nas demandas.

Referências

Alyafawi, I., Schiller, E., Braun, T., Dimitrova, D., Gomes, A. e Nikaein, N. (2015). Critical issues of centralized and cloudified LTE-FDD radio access networks. Em *IEEE ICC*, p. 5523–5528.

Bartelt, J., Rost, P., Wubben, D., Lessmann, J., Melis, B. e Fettweis, G. (2015). Fronthaul and backhaul requirements of flexibly centralized radio access networks. *IEEE Wireless Communications*, 22(5):105–111.

Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S. e Dittmann, L. (2015). Cloud RAN for mobile networks—a technology overview. *IEEE Communications surveys & tutorials*, 17(1):405–426.

Coutinho, A. A. T. R., Carneiro, E. O. e Greve, F. G. P. (2016). Computação em névoa: Conceitos, aplicações e desafios. Em *Minicursos do XXXIV SBRC*, p. 266–315.

Dalla-Costa, A. G., Bondan, L., Wickboldt, J. A., Both, C. B. e Granville, L. Z. (2017a). Maestro: An NFV orchestrator for wireless environments aware of VNF internal compositions. Em *IEEE AINA*, p. 484–491.

Dalla-Costa, A. G., Schimuneck, M. A., Wickboldt, J. A., Both, C. B., Gaspary, L. P. e Granville, L. Z. (2017b). NFV em redes 5G: Avaliando o desempenho de composição de funções virtualizadas via Maestro. Em *XXXV SBRC*, p. 1–14.

GNU (2017). Glpk (GNU linear programming kit). https://www.gnu.org/software/glpk/ - Acessado em dezembro de 2017.

Herrera, J. G. e Botero, J.-F. (2016). Resource allocation in NFV: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532.

Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P. e Gaspary, L. P. (2015). Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. Em *IFIP/IEEE IM*, p. 98–106.

Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F. e Boutaba, R. (2015). Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, 18(1):236–262.

Queiroz, G. F. C., Couto, R. S. e Sztajnberg, A. (2017). TRELIS: Posicionamento de funções virtuais de rede com economia de energia e resiliência. Em 16^o *WPERFORMANCE*, p. 1656–1669.

Szwarcfiter, J. L. e Markenzon, L. (2013). *Estruturas de Dados e seus Algoritmos*. Livros Técnicos e Científicos, 3 edição.

Wang, K., Zhao, M. e Zhou, W. (2014). Traffic-aware graph-based dynamic frequency reuse for heterogeneous cloud-ran. Em *IEEE GLOBECOM*, p. 2308–2313.

Wubben, D., Rost, P., Bartelt, J. S., Lalam, M., Savin, V., Gorgoglione, M., Dekorsy, A. e Fettweis, G. (2014). Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN. *IEEE signal processing magazine*, 31(6):35–44.