

**MODELOS E ALGORITMOS PARA O PROBLEMA  
DE MAPEAMENTO DE REDES VIRTUAIS**

BRÁULIO ANTÔNIO MESQUITA SOUZA

**MODELOS E ALGORITMOS PARA O PROBLEMA  
DE MAPEAMENTO DE REDES VIRTUAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais – Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: GERALDO ROBSON MATEUS  
COORIENTADORA: FERNANDA SUMIKA HOJO DE SOUZA

Belo Horizonte

Agosto de 2016

© 2016, Bráulio Antônio Mesquita Souza.  
. Todos os direitos reservados

Souza, Bráulio Antônio Mesquita.

S729m Modelos e algoritmos para o problema de mapeamento de redes virtuais [manuscrito] / Bráulio Antônio Mesquita Souza — 2016.  
xiv, 45 f. il.

Orientador: Geraldo Robson Mateus.  
Coorientadora: Fernanda Sumika Hojo de Souza  
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação  
Referências: f.42-45

1. Computação – Teses. 2. Virtualização de redes – Teses. 3. Programação linear inteira – Teses. 4. Geração de colunas – Teses. I. Mateus, Geraldo Robson. II. Souza, Fernanda Sumika Hojo de. III. Título.

CDU 519.6\*61 (043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa  
CRB 6ª Região nº 1510



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Modelos e algoritmos para o problema de mapeamento de redes virtuais

**BRÁULIO ANTÔNIO MESQUITA SOUZA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. GERALDO ROBSON MATEUS - Orientador  
Departamento de Ciência da Computação - UFMG

PROFA. FERNANDA SUMIKA HOJO DE SOUZA - Coorientadora  
Departamento de Ciência da Computação - UFSJ

PROF. ALEXANDRE SALLES DA CUNHA  
Departamento de Ciência da Computação - UFMG

PROF. FLÁVIO VINÍCIUS CRUZEIRO MARTINS  
Departamento de Computação - CEFETMG

Belo Horizonte, 08 de agosto de 2016.

# Agradecimentos

A Deus, que sempre me deu forças para nunca desistir.

Ao meu orientador, professor Geraldo Robson Mateus, pela confiança, pela oportunidade de poder trabalhar junto a ele, pela dedicação e empenho no decorrer do mestrado.

À minha orientadora da graduação e atual coorientadora, Fernanda Sumika Hojo de Souza, que me incentivou a continuar meus estudos, agradeço por toda a paciência e compreensão.

Aos professores do Departamento de Ciência da Computação da Universidade Federal de São João del-Rei (DCOMP-UFSJ), que me forneceram uma forte base de conhecimento, permitindo que eu chegasse até aqui.

Aos meus familiares, em especial meus pais, Paulo e Maria Luiza, minha madrinha, Terezinha, e minha tia Salete, que sempre acreditaram e torceram por mim.

Aos colegas do LaPO – Laboratório de Pesquisa Operacional e Otimização, que me ajudaram inúmeras vezes com conversas e ideias.

Aos meus amigos, agradeço imensamente pelos momentos de descanso e divertimento, sem eles essa trajetória teria sido bem mais difícil.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Universidade Federal de Minas Gerais (UFMG) e ao Departamento de Ciência da Computação (DCC), pelo apoio financeiro e pela infraestrutura de estudo e pesquisa oferecida ao longo do mestrado.

A todos, o meu muito obrigado!

*“Pensar é o trabalho mais difícil que existe.  
Talvez por isso tão poucos se dediquem a ele.”*  
(Henry Ford)

# Resumo

Virtualização de Redes tem sido utilizada para permitir que várias redes virtuais possam coexistir sobre uma mesma rede física compartilhando seus recursos. Neste cenário, surge o problema de mapear eficientemente as redes virtuais no substrato físico, atendendo requisitos de capacidade e qualidade. Neste trabalho, é apresentado um modelo compacto baseada em Programação Linear Inteira capaz de fornecer soluções ótimas para instâncias de pequeno porte. Dois modelos estendidos são propostos, apresentando limites duais mais fortes do que o modelo compacto. Com base nestes modelos, são propostos algoritmos *branch-and-price*. Resultados mostram que, para instâncias onde se considera restrições de localização geográfica dos nós, os algoritmos propostos obtiveram, dentro do limite de tempo estipulado, soluções melhores que o modelo compacto para instâncias com maior número de redes virtuais. Além disso, um estudo do impacto de diferentes funções objetivo (custo de mapeamento, atraso e balanceamento de carga) e restrições de qualidade de serviço (localização geográfica, atraso máximo e resiliência) é realizado nas soluções obtidas.

**Palavras-chave:** Mapeamento de Redes Virtuais, Programação Linear Inteira, Geração de Colunas.

# Abstract

Network Virtualization has been used to allow that many virtual networks can coexist over a substrate network, sharing its resources. In this scenario, the problem of mapping the virtual networks onto the physical substrate, meeting capacity and quality requirements, emerges. In this work, a compact model based on integer linear programming, which provides optimal solutions, is presented. Two extended models are proposed, with stronger dual bounds than the compact model. Based on these models, we also propose branch-and-price algorithms. Results show that, for instances where geographical location is considered for the nodes, both algorithms achieved, within the time limit, better solutions than the compact model for a greater number of virtual network requests. Moreover, it is performed a study on the impact of different objective functions (mapping cost, delay, and load balancing) and quality of service constraints (geographic location, maximum delay, and resilience) on the solutions

**Keywords:** Virtual Network Embedding, Combinatorial Optimization, Column Generation.



# Lista de Figuras

3.1	Duas VNs mapeadas no substrato físico . . . . .	11
3.2	Mapeamentos de VNs sem e com balanceamento de carga . . . . .	15
5.1	Comparação de resultados entre o modelo compacto e algoritmos <i>branch-and-price</i> sem o uso de restrições de localização geográfica . . . . .	36
5.2	Comparação de resultados entre o modelo compacto e algoritmos <i>branch-and-price</i> considerando restrições de localização geográfica . . . . .	38

# Lista de Tabelas

5.1	Resultados para o objetivo VNE_PLI_COST (Minimizar Custo de Mapeamento) . . . . .	31
5.2	Resultados para o objetivo VNE_PLI_LB (Balanceamento de Carga) . . .	32
5.3	Resultados para o objetivo VNE_PLI_DL (Minimizar Atraso) . . . . .	33
5.4	Comparação de resultados entre o modelo compacto e os algoritmos de geração de colunas, sem uso da restrição de localização geográfica . . . . .	35
5.5	Comparação de resultados entre o modelo compacto e os algoritmos de geração de colunas, com restrições de localização geográfica . . . . .	35
5.6	Resultados sumarizados para o modelo compacto e algoritmos <i>branch-and-price</i> sem uso de restrições de localização geográfica . . . . .	37
5.7	Resultados sumarizados dos testes com o modelo compacto e algoritmos <i>branch-and-price</i> com restrições de localização geográfica . . . . .	38

# List of Algorithms

1	Pseudocódigo do algoritmo de Geração de Configurações . . . . .	24
2	Pseudocódigo do algoritmo de Geração de Caminhos . . . . .	24
3	Pseudocódigo da heurística para a solução inicial . . . . .	27

# Lista de Siglas

<b>FI</b>	<i>Future Internet</i>
<b>PLI</b>	Programação Linear Inteira
<b>PLIM</b>	Programação Linear Inteira Mista
<b>QoS</b>	<i>Quality of Service</i>
<b>VN</b>	<i>Virtual Network</i>
<b>VNE</b>	<i>Virtual Network Embedding</i>

# Sumário

Agradecimentos	v
Resumo	vii
Abstract	viii
Lista de Figuras	ix
Lista de Tabelas	x
Lista de Siglas	xii
<b>1 Introdução</b>	<b>1</b>
1.1 Justificativa . . . . .	3
1.2 Objetivos . . . . .	3
1.2.1 Geral . . . . .	3
1.2.2 Específicos . . . . .	3
1.3 Contribuições . . . . .	4
1.4 Organização . . . . .	4
<b>2 Trabalhos Relacionados</b>	<b>5</b>
<b>3 Definição do Problema e Modelos</b>	<b>10</b>
3.1 Definição do Problema . . . . .	10
3.2 Modelos Compactos . . . . .	12
3.2.1 Modelos VNE_PLI . . . . .	12
3.3 Modelos Estendidos (VNE_GC) . . . . .	17
3.3.1 Modelo Estendido baseado na geração de configurações (VNE_GC_CONFIG) . . . . .	17

3.3.2	Modelo Estendido baseado na geração de caminhos (VNE_GC_PATH) . . . . .	20
<b>4</b>	<b>Algoritmos</b>	<b>23</b>
4.1	Algoritmos de Geração de Colunas . . . . .	23
4.1.1	Geração de Configurações . . . . .	23
4.1.2	Geração de Caminhos . . . . .	23
4.2	Algoritmos <i>Branch-and-Price</i> . . . . .	24
4.2.1	Estratégia de <i>branching</i> . . . . .	25
4.2.2	Solução Inicial . . . . .	26
4.2.3	Exploração da Árvore de <i>Branch-and-Price</i> . . . . .	27
4.2.4	Limites superiores . . . . .	27
<b>5</b>	<b>Resultados Computacionais</b>	<b>29</b>
5.1	Instâncias . . . . .	29
5.2	Impacto de diferentes funções objetivo e restrições . . . . .	30
5.3	Modelos baseados em Geração de Colunas . . . . .	34
5.4	Algoritmos Branch-and-Price . . . . .	36
<b>6</b>	<b>Considerações Finais</b>	<b>40</b>
6.1	Conclusões . . . . .	40
6.2	Trabalhos Futuros . . . . .	41
	<b>Referências Bibliográficas</b>	<b>42</b>

# Capítulo 1

## Introdução

A importância das redes de telecomunicações cresceu drasticamente nos últimos anos, assim como o acesso à internet. No geral, as tecnologias oferecem uma variedade de serviços como transmissão de áudio, vídeo e dados, em adição ao processamento e armazenamento remotos, entre outros. Além disso, a crescente demanda dos usuários da rede (pessoas, veículos, organizações, instituições, etc.) e a preocupação com a confiabilidade dos serviços para evitar a insatisfação desses clientes requerem graus de organização na rede. Assim, provedores de infraestrutura e serviços de redes começaram a oferecer serviços de alto nível, levando em conta atributos, incluindo capacidades, atraso, capacidade de sobrevivência (proteção contra falhas na conexão), e custos, de acordo com contratos com as entidades de usuários da rede. Para atender as demandas e garantias, todos esses atributos devem ser processados com mais acurácia.

Em sua concepção, a Internet, maior rede de comunicação de escala global, foi proposta com base no modelo TCP/IP, cuja simplicidade permitiu um rápido crescimento e grande evolução das aplicações. Porém, essa simplicidade é responsável pelas limitações que se tornaram cada vez mais evidentes e comprometem a expansão e a inclusão de novos serviços que não haviam sido previstos no projeto original. Algumas barreiras são devidas ao atual uso do protocolo IPv4, altos custos de expansão, dificuldade de avanço no desenvolvimento da Internet das Coisas [Giusto et al., 2010], entre outras. Esse fenômeno é conhecido como “Ossificação da Internet”, representando um problema para a Internet do Futuro (FI), onde arquiteturas alternativas mais flexíveis são críticas para a criação de uma rede mais segura e eficiente.

Atualmente, virtualização de redes tem sido utilizada como uma alternativa para corrigir a ossificação da internet [Turner & Taylor, 2005], permitindo a implementação e execução de novas funcionalidades [Fischer et al., 2013]. Embora não seja um conceito novo, virtualização é uma técnica para superar a resistência da internet atual às

mudanças fundamentais.

Redes virtuais são conjuntos de nós virtuais, ligados por enlaces virtuais. Através da virtualização dos recursos físicos é possível criar uma visão lógica dos componentes de hardware de forma que várias redes virtuais, com diferentes topologias, arquiteturas e características possam coexistir em um mesmo substrato e compartilhar recursos físicos [Chowdhury & Boutaba, 2010]. Durante a última década, cada vez mais, virtualização vem sendo adotada para melhorar eficiência e agilidade de processamento e armazenamento dos recursos baseados em software. Nesse contexto, surge o problema da alocação eficiente de recursos de infraestrutura para atender as demandas virtuais. Esse problema é conhecido como Mapeamento de Redes Virtuais (ou VNE - *Virtual Network Embedding*) [Fischer et al., 2013; Zhu & Ammar, 2006], cujo objetivo é alocar um conjunto de requisições de redes a um mesmo substrato físico. Algumas das principais vantagens da virtualização incluem flexibilidade, escalabilidade, isolamento e redução dos custos.

Diferentes parâmetros e critérios de otimização podem ser considerados ao realizar o mapeamento das redes virtuais na rede física [Stezenbach et al., 2012]. Os provedores podem aplicar custos de utilização dos enlaces (enlaces de melhor qualidade são mais caros, por exemplo) e buscar um mapeamento de menor custo. Outros objetivos que podem ser utilizados incluem redução na utilização dos enlaces, a fim de evitar sobrecargas na comunicação, minimização do atraso na comunicação entre os roteadores virtuais, dentre outros.

Além disso, os provedores podem incluir restrições adicionais ao problema para prover níveis de Qualidade de Serviço (QoS). A imposição de um atraso máximo entre os roteadores virtuais ajuda a aumentar o desempenho das aplicações que são executadas na rede, uma vez que a comunicação entre os roteadores virtuais será mais eficiente por utilizar um número menor de enlaces e enlaces que não acarretam atrasos muito grandes na comunicação.

Resiliência em redes é uma outra questão fundamental que não pode ser desconsiderada em cenários de virtualização. Em condições de falha na rede física, mecanismos de proteção e reconfiguração permitem que a rede mantenha sua máxima conectividade e QoS. Resiliência pode ser tratada em nível topológico, como por exemplo uma rede biconexa, que é robusta contra falhas em enlace único. Outros exemplos incluem o uso de protocolos de rede para realocar as demandas dinamicamente quando uma falha ocorre. Mecanismos de proteção alocam recursos extras para evitar a perda dos serviços em casos de falhas.

Nesse contexto, o VNE é um problema de otimização pertencente à classe de problemas NP-difíceis, provado a partir da redução do *Multi-way Separator Problem*



[Andersen, 2002]. Em [Amaldi et al., 2016], os autores também apresentam a prova de complexidade do problema, a partir da redução do *Maximum Stable Set Problem*, mostrando que o VNE é um problema fortemente NP-difícil. Além disso, o mapeamento deve ser o mais realista possível, incorporando aos modelos e algoritmos, características mais próximas do problema real. Técnicas e soluções de otimização representam um papel fundamental nesta tarefa, buscando a eficiência, escalabilidade e segurança dessas redes, além de atender as demandas dos usuários a um baixo custo.

No presente trabalho são propostos dois modelos estendidos para o mapeamento de várias redes virtuais simultaneamente e algoritmos de *branch-and-price* para sua resolução. Além disso, este trabalho apresenta uma análise do impacto de diferentes restrições de QoS, como restrições de localização geográfica, atraso máximo permitido na comunicação e resiliência contra falhas em enlaces.

## 1.1 Justificativa

Redes Virtuais têm sido utilizadas para permitir a incorporação das atuais e de novas tecnologias ao atual modelo de internet, contornando o problema da ossificação da internet. Com o avanço tecnológico, principalmente na área da Internet das Coisas, o uso das Redes Virtuais mostra-se de grande importância, onde tecnologias heterogêneas coexistem e a estrutura da rede tem papel fundamental para o funcionamento dos componentes. A maioria dos trabalhos existentes resolve o VNE através da utilização de heurísticas, o que não garante a otimalidade das soluções. Além disso, alguns trabalhos que propõem métodos exatos para o VNE consideram uma única rede virtual a ser mapeada no substrato físico.

## 1.2 Objetivos

### 1.2.1 Geral

Desenvolver modelos e algoritmos para a resolução do VNE.

### 1.2.2 Específicos

- Propor e comparar diferentes modelos para a resolução do VNE, através da utilização de diferentes funções objetivo e restrições.

- Desenvolver e avaliar modelos baseadas em geração de colunas para obtenção de bons limites de relaxação linear para o VNE considerando múltiplas redes virtuais simultaneamente.
- Desenvolver e avaliar algoritmos de *branch-and-price* para obtenção de soluções para o VNE em menor tempo computacional.

### 1.3 Contribuições

As contribuições deste trabalho incluem novos modelos e algoritmos exatos para o VNE, ainda pouco explorados na literatura. Diferente dos trabalhos presentes na literatura, os modelos propostos são capazes de realizar o mapeamento de múltiplas requisições de redes virtuais simultaneamente.

O artigo intitulado “*On the Impact of Resilience, Delay and Geographical Location constraints in the Virtual Network Embedding Problem*” é derivado da dissertação e foi publicado no XLVIII Simpósio Brasileiro de Pesquisa Operacional, 27 a 30 de setembro de 2016 em Vitória – ES.

### 1.4 Organização

O presente trabalho está organizado como segue. No capítulo 2 são apresentados trabalhos sobre o VNE e algumas variantes do problema. O capítulo 3 apresenta a definição formal do problema e os modelos matemáticos propostos. Os algoritmos de geração de colunas, bem como os algoritmos *branch-and-price* desenvolvidos, são apresentados no capítulo 4. Os resultados dos experimentos são apresentados no capítulo 5. Por fim, o capítulo 6 apresenta a conclusão e algumas possibilidades para trabalhos futuros.

## Capítulo 2

# Trabalhos Relacionados

Preocupados com a internet do futuro, muitas pesquisas têm sido desenvolvidas sobre o problema de Mapeamento de Redes Virtuais [Zhu & Ammar, 2006; Yu et al., 2008; Inführ & Raidl, 2011; Chowdhury et al., 2009]. De acordo com Fischer e outros [Fischer et al., 2013], o VNE pode ser classificado de acordo com a reconfiguração das VNs (estática ou dinâmica), a estratégia do algoritmo (centralizada ou distribuída) e a tolerância à falhas (conciso ou redundante). Essa taxonomia categoriza vários trabalhos presentes na literatura. Um *survey* completo que sumariza vários trabalhos sobre o VNE pode ser encontrado em [Fischer et al., 2013].

O Problema de Mapeamento de Redes Virtuais é de natureza *online*, uma vez que as requisições chegam no provedor de serviços dinamicamente e as demandas não são conhecidas *a priori*. O problema pode ser resolvido considerando as abordagens *online* ou *offline*, permitindo nesse segundo caso, que o algoritmo aguarde a chegada de um conjunto de requisições virtuais e as trate simultaneamente. É importante destacar que mesmo a abordagem *offline* é NP-Difícil.

Diferentes objetivos podem ser considerados como parâmetros de otimização, tais como benefício econômico, métricas de qualidade de serviço, utilização dos recursos físicos, entre outros. Houidi e outros [Houidi et al., 2011] utilizam um modelo de Programação Linear Inteira para encontrar soluções que minimizem o custo de mapeamento, ao mesmo tempo que maximizam a taxa de aceitação das redes virtuais. Em [Nonde et al., 2015], é utilizado um modelo que objetiva a utilização de um menor conjunto de recursos do substrato físico de forma a economizar energia, desligando-os quando ociosos.

No cenário *online*, as redes virtuais podem ter tempo de vida, isto é, após serem alocadas, permanecem na rede física por um período. Após esse tempo, a rede é desalocada, liberando seus recursos. Conforme as requisições entram e saem da rede,

os recursos físicos começam a fragmentar-se. Com o passar do tempo, os recursos físicos podem não ser suficientes para alocar novas redes devido à atual configuração.

Os modelos dinâmico e estático permitem que as VNs sejam realocadas ou não, respectivamente, ou seja, uma vez alocadas, elas podem ter suas posições alteradas ou serem mantidas no mesmo local durante seu tempo de vida na rede. No modelo dinâmico, o custo de reconfiguração das VNs deve ser considerado, porém com o benefício de melhorar a fragmentação do substrato [Fajjari et al., 2011; Sun et al., 2013]. Em [Tran et al., 2012], o modelo dinâmico é resolvido em etapas, onde a alocação de cada nova VN é resolvida através de um modelo de programação linear inteira. Sempre que uma VN não pode ser atendida devido à falta de recursos físicos, realoca-se o mínimo de VNs possível para atender a nova requisição.

O algoritmo de mapeamento pode operar através de uma estratégia centralizada ou distribuída. Enquanto uma única entidade é responsável pelo mapeamento no modelo centralizado, várias entidades da rede podem compartilhar a tarefa de mapeamento em um modo distribuído. A visão geral global provida pelo modelo centralizado permite que melhores soluções sejam geradas, a um maior custo de processamento. Por outro lado, um modelo distribuído possui uma sobrecarga na troca de mensagens entre seus componentes, mas fornece escalabilidade em cenários de larga escala [Houidi et al., 2008].

Em [Marquezan et al., 2010], os autores apresentam um algoritmo distribuído onde os roteadores físicos são capazes de gerenciar as demandas virtuais com base em informação local, descartando a necessidade de uma entidade centralizada para gerenciar toda a informação da rede. Nesse trabalho, os autores admitem que as demandas virtuais de cada rede sofram alterações durante seu tempo de operação, surgindo a necessidade de redistribuir os roteadores virtuais pela rede de forma a diminuir a sobrecarga na comunicação.

Um algoritmo heurístico baseado no princípio da proximidade é apresentado em [Liu et al., 2011], onde é considerada, além dos fatores de capacidade, a distância durante o mapeamento dos nós. O algoritmo proposto mostra-se eficiente tanto em termos de utilização da rede física quanto em tempo de execução.

Na prática, falhas ocorrem, e requerem que as redes provejam resiliência quando um nó ou enlace falha enquanto estão em operação [Rahman & Boutaba, 2013]. Em um modelo conciso, os recursos alocados do substrato são exatamente aqueles requisitados pelas VNs, isto é, nenhum recurso adicional é reservado. Modelos redundantes podem reservar recursos extras para lidar com falhas. Esses modelos geralmente aumentam a rejeição de novas demandas dependendo da reserva de recursos. Em [Herker et al., 2013], é apresentado um *survey* sobre o problema de sobrevivência à falhas no contexto

de virtualização e discutidas estratégias de solução do problema.

Outra característica importante no modo de resolução do problema refere-se à separação das tarefas de alocação dos nós e dos enlaces. Os algoritmos podem realizar o mapeamento dos nós e dos enlaces em uma única etapa ou encontrar um mapeamento eficiente para os nós para depois encontrar caminhos físicos para atender os enlaces virtuais. Em [Lischka & Karl, 2009], é apresentado um algoritmo que mapeia nós e enlaces em uma única etapa, através de uma abordagem baseada na detecção de isomorfismo de subgrafos.

Heurísticas para o VNE são propostas em [Zhu & Ammar, 2006], com o objetivo de encontrar soluções que, ao minimizar a utilização dos nós e enlaces, sejam capazes de prover balanceamento de carga dos recursos utilizados. Os autores comparam abordagens com e sem reconfiguração, mostrando que o algoritmo proposto supera algoritmos existentes na literatura e que permite a reconfiguração de um pequeno subconjunto de VNs críticas, obtendo vários benefícios da reconfiguração dinâmica a um baixo custo computacional.

Duas considerações podem ser realizadas para tornar a abordagem mais simples e utilizar os recursos da rede de forma mais eficiente. Em [Yu et al., 2008], os enlaces virtuais podem ser divididos entre vários caminhos físicos; e periodicamente os caminhos já atribuídos são migrados, visando reotimizar a utilização da rede física. Ambas considerações permitem uma maior aceitação de redes virtuais, comparadas às abordagens clássicas.

Um algoritmo adaptativo é proposto em [Fajjari et al., 2016]. Baseado em algoritmos aproximativos, trata o problema *online* e realiza o mapeamento das redes virtuais evitando a sobrecarga dos recursos, de forma a evitar gargalos na rede e favorecer a utilização dos recursos com pouca carga. Dessa forma, obtém-se uma maior taxa de aceitação de requisições, mas pode perder requisições significativas.

Wang e seus colaboradores [Wang et al., 2012] propõem uma heurística de duas fases, que utiliza informações sobre a topologia das redes através de métricas de centralidade dos nós para melhorar a utilização da rede e diminuir a complexidade do mapeamento. Através da utilização de duas métricas de centralidade, o algoritmo proposto é capaz de aumentar a taxa de aceitação de requisições, enquanto diminui os custos referentes à rede física. Em contrapartida à utilização de apenas uma informação dos nós, em [Shuiqing et al., 2016], os autores utilizam vários atributos como métrica para realizar o *ranking* dos nós, a fim de obter maiores taxa de aceitação e receita gerada pelas redes virtuais.

Em [Chowdhury et al., 2009], dois algoritmos são propostos: D-ViNE (*Deterministic VN Embedding*) e R-ViNE (*Randomized VN Embedding*). Os algoritmos são ba-

seados em um procedimento de dois estágios no qual o mapeamento dos nós é resolvido através de programação linear sem levar em consideração restrições de integralidade, seguido de um estágio de arredondamento. O mapeamento dos enlaces é realizado em sequência através de um dos procedimentos: (a) caminho mínimo quando um único caminho é considerado para cada enlace virtual, ou (b) fluxo de multi-produto, quando é permitido que o enlace virtual seja dividido em vários caminhos físicos. Resultados computacionais mostram que dividir o fluxo aumenta a taxa de aceitação das requisições pois evita gargalos; porém, na prática, a maioria das aplicações reais não permite essa flexibilidade.

Alguns trabalhos apresentam modelos baseados em geração de colunas e algoritmos *branch-and-price* para o VNE [Moura, 2015; Wang et al., 2016]. Os modelos apresentados baseiam-se em geração de caminhos e tratam o problema com uma única rede virtual a ser mapeada.

Devido à complexidade do problema, muitos autores optam pela utilização de meta-heurísticas para resolver o VNE, a fim de obter resultados bons, porém sem garantia de otimalidade. Uma meta-heurística de colônia de formigas é proposta em [Zhu & Wang, 2016], baseada em decomposição de grafos. Tal abordagem decompõe cada VN em combinações de topologias.

Restrições adicionais podem ser consideradas para tornar o problema mais próximo à aplicações reais. Em [Inführ & Raidl, 2011], os autores consideram a localização geográfica dos nós, um atraso máximo permitido na comunicação entre dois nós virtuais e capacidade de roteamento dos nós físicos. Restrições de localização podem, por exemplo, garantir que um serviço de jogo em tempo real possua nós virtuais em cidades estratégicas. Restrições de atraso garantem a qualidade de serviço em termos de atraso máximo, enquanto restrições de capacidade de roteamento impedem que os roteadores fiquem sobrecarregados com o fluxo de comunicação.

A maioria dos trabalhos presentes na literatura considera que os provedores de infraestrutura operam o tempo todo. Em [Rahman & Boutaba, 2013], os autores estudam a capacidade de sobrevivência da rede quando ocorrem falhas nos enlaces da rede física. Os autores propõem um modelo baseado em programação linear inteira mista e heurísticas eficientes para o problema de mapeamento de redes virtuais resilientes. A principal contribuição se refere à etapa de mapeamento dos enlaces, sendo empregada uma heurística da literatura para a etapa de mapeamento dos nós. As heurísticas trabalham em duas estratégias: proativa, na qual recursos extras são alocados como mecanismos de proteção para evitar falhas em aplicações onde o atraso é crítico; e estratégia reativa, a qual evita o desperdício dos recursos adicionais reservados tratando as falhas assim que ocorrem, levando a um pequeno atraso no mecanismo de resiliência.

A existência de recursos heterogêneos nas redes virtuais e física é abordada em [Nogueira et al., 2011]. Uma heurística é apresentada para resolver o problema levando em consideração heterogeneidade tanto nos nós, quanto nos enlaces.

A abordagem considerada nesse trabalho é classificada como estática para a reconfiguração das VNs, centralizada para a estratégia do algoritmo, concisa e redundante para a tolerância à falhas (os dois casos são avaliados), *offline* para o conhecimento das requisições e permite o mapeamento de um enlace virtual em apenas um caminho físico.

Este trabalho apresenta dois modelos estendidos de geração de colunas e, com base nesses modelos, algoritmos *branch-and-price* para o problema. Contrário a alguns trabalhos presentes na literatura [Moura, 2015; Wang et al., 2016], os modelos neste trabalho realizam o mapeamento de várias requisições de redes virtuais simultaneamente sobre uma mesma infraestrutura física. Além disso, é realizada a análise do impacto de diferentes funções-objetivo (custo do mapeamento, atraso e balanceamento de carga) e restrições de QoS (resiliência contra falha em enlaces, localização geográfica e atraso máximo) sobre o problema.

# Capítulo 3

## Definição do Problema e Modelos

### 3.1 Definição do Problema

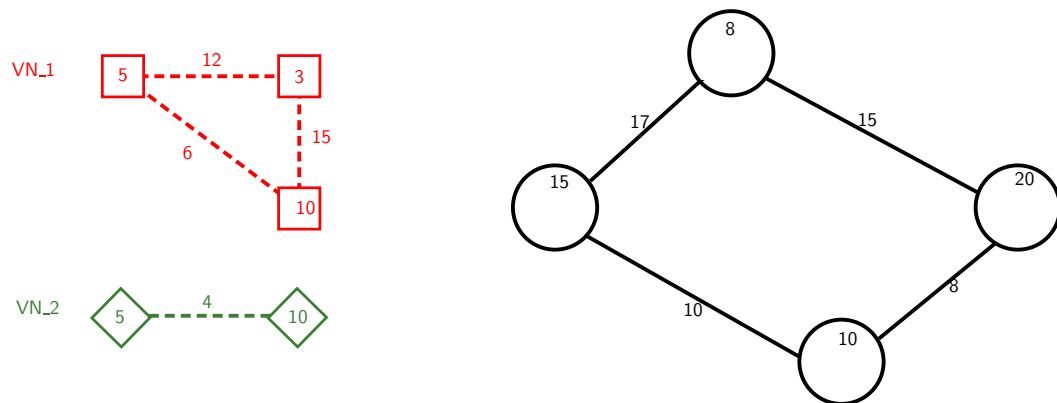
O problema de Mapeamento de Redes Virtuais pode ser modelado como segue. Seja  $G^s = (N^s, E^s)$  um grafo não direcionado com capacidades nas arestas e nos vértices representando o substrato físico, no qual  $N^s$  e  $E^s$  são os conjuntos de nós e enlaces físicos, respectivamente. De forma análoga, uma rede virtual  $v \in V$  é modelada como um grafo não direcionado  $G^v = (N^v, E^v)$ , composto por um conjunto de nós ( $N^v$ ) e arestas ( $E^v$ ) virtuais, associados à requisitos de CPU,  $c_k^v$ , para cada nó virtual  $k \in N^v$  e de largura de banda,  $b_{kl}^v$ , para cada requisição  $(k, l) \in E^v$ , respectivamente.

As redes virtuais demandam recursos da rede física. Portanto, a rede física deve ser capaz de prover um mapeamento que respeita tais requisitos. Um enlace virtual com demanda de 50 Mbps não pode ser mapeado em enlaces cujas larguras de banda são de 25 Mbps, por exemplo. Dessa forma, cada nó do substrato  $i \in N^s$  possui uma capacidade de CPU,  $C_i$ , e cada enlace  $(i, j) \in E^s$  do substrato possui uma capacidade de largura de banda,  $B_{ij}$ .

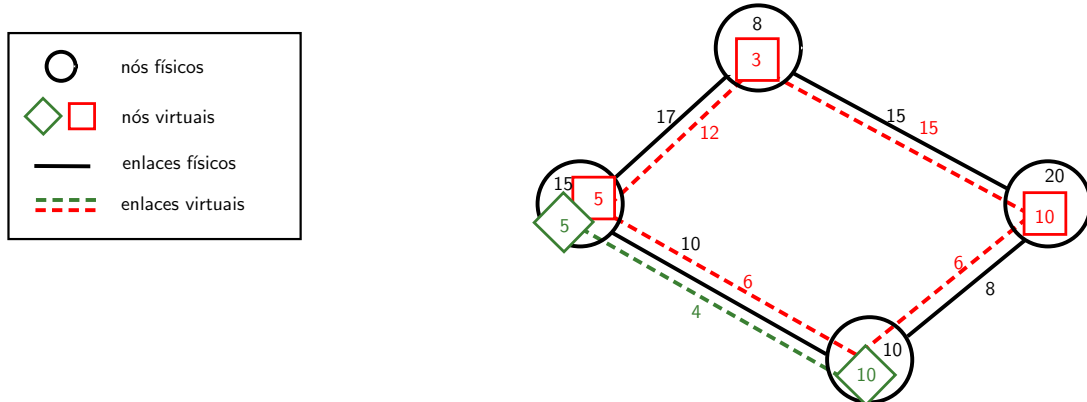
Formalmente, uma solução para o problema consiste em encontrar um mapeamento  $f : G^v \rightarrow G^s$ , no qual  $\forall k \in N^v, \exists i \in N^s : c_k^v \leq C_i$  e  $\forall (k, l) \in E^v, \exists \{(i, j)^1, \dots, (i, j)^m\} : b_{kl}^v \leq B_{ij}^1, \dots, b_{kl}^v \leq B_{ij}^m$ . Se for possível encontrar tal mapeamento, a rede virtual é aceita; caso contrário, rejeitada. Portanto, uma rede virtual é atendida se todas os requisitos de CPU dos roteadores e as demandas de largura de banda forem atendidas, não excedendo as capacidades da rede física.

A Figura 3.1 (a) apresenta duas redes virtuais ( $VN^1$  e  $VN^2$ ) mapeadas no mesmo substrato físico. Os valores internos a cada nó representado por losangos ou quadrados indicam os requisitos de CPU dos roteadores das redes virtuais, enquanto os valores sobre as linhas tracejadas indicam requisitos de largura de banda. O substrato físico é





(a) Redes virtuais e rede física com seus respectivos requisitos e/ou capacidades



(b) Um possível mapeamento das redes virtuais sobre a rede física

Figura 3.1: Duas VNs mapeadas no substrato físico

formado por quatro nós e quatro enlaces com suas capacidades associadas (os valores internos aos círculos indicam capacidade de CPU dos roteadores físicos, enquanto valores abaixo de cada enlace físico indicam capacidade de largura de banda dos enlaces). A rede virtual  $VN^1$  é composta por três nós e três enlaces, enquanto  $VN^2$  é composta por dois nós e um enlace. Uma solução para o mapeamento das redes é apresentada na Figura 3.1 (b), onde observa-se que as duas redes virtuais puderam ser mapeadas no mesmo substrato, respeitando todas as capacidades, e um nó físico hospeda mais de um nó virtual (de redes virtuais diferentes). Além disso, um dos enlaces virtuais é mapeado em um caminho físico com mais de um enlace.

Para o VNE, são apresentados os modelos a seguir: VNE\_PLI,

VNE\_GC\_CONFIG e VNE\_GC\_PATH.

Para os modelos que seguem, são considerados os seguintes conjuntos e dados para o problema:

$V$	conjunto de redes virtuais.
$N^v$	conjunto de nós virtuais da rede $v \in V$ .
$E^v$	conjunto de enlaces virtuais da rede $v \in V$ .
$N^s$	conjunto de nós do substrato físico.
$E^s$	conjunto de enlaces do substrato físico.

Com os dados de entrada:

$b_{kl}^v$	demanda por largura de banda entre os nós $k \in N^v$ e $l \in N^v$ da rede virtual $v \in V$ .
$c_k^v$	requisito de CPU do nó virtual $k \in N^v$ da rede virtual $v \in V$ .
$B_{ij}$	capacidade de largura de banda do enlace físico $(i, j) \in E^s$ .
$C_i$	capacidade de CPU do nó físico $i \in N^s$ .
$U_{ij}$	custo por unidade utilizada de largura de banda no enlace $(i, j) \in E^s$ .

## 3.2 Modelos Compactos

### 3.2.1 Modelos VNE\_PLI

O Problema de Mapeamento de Redes Virtuais (VNE) pode ser modelado como um Problema de Programação Linear Inteira (PLI). Vale destacar que os modelos compactos apresentados não são novas, mas baseados em modelos já existentes na literatura. A seguir são apresentadas três modelos VNE\_PLI: VNE\_PLI\_COST, VNE\_PLI\_LB e VNE\_PLI\_DL, que se diferem pela função objetivo. São utilizados três conjuntos de variáveis de decisão:

$y^v$	variável que indica se a rede virtual $v \in V$ é aceita ( $y^v = 1$ ) ou não ( $y^v = 0$ )
$z_{ki}^v$	variável que indica se o nó virtual $k \in N^v$ da rede virtual $v \in V$ é alocado ao nó físico $i \in N^s$ ( $z_{ki}^v = 1$ ) ou não ( $z_{ki}^v = 0$ )
$x_{ij}^{vkl}$	variável que indica se o enlace virtual $(k, l) \in E^v$ da rede virtual $v \in V$ utiliza o enlace físico $(i, j) \in E^s$ ao longo do caminho físico que conecta os nós físicos que hospedam os nós virtuais $k, l \in N^v$ ( $x_{ij}^{vkl} = 1$ ) ou não ( $x_{ij}^{vkl} = 0$ ).

Define-se  $P$  como a penalidade por não atender uma rede virtual.

Dessa forma, o problema pode ser modelado como segue.

**VNE\_PLI\_COST:**

Seja a função objetivo:

$$\min \left[ \sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} U_{ij} b_{kl}^v x_{ij}^{vkl} + P \cdot \sum_{v \in V} (1 - y^v) \right] \quad (3.1)$$

Sujeito ao seguinte conjunto de restrições:

$$\sum_{i \in N^s} z_{ki}^v = y^v \quad \forall k \in N^v, \forall v \in V \quad (3.2)$$

$$\sum_{k \in N^v} z_{ki}^v \leq 1 \quad \forall i \in N^s, \forall v \in V \quad (3.3)$$

$$\sum_{v \in V} \sum_{k \in N^v} c_k^v z_{ki}^v \leq C_i \quad \forall i \in N^s \quad (3.4)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} b_{kl}^v x_{ij}^{vkl} \leq B_{ij} \quad \forall (i,j) \in E^s \quad (3.5)$$

$$\sum_{(i,j) \in E^s} x_{ij}^{vkl} - \sum_{(h,i) \in E^s} x_{hi}^{vkl} = z_{ki}^v - z_{li}^v \quad \forall i \in N^s, \forall (k,l) \in E^v, \forall v \in V \quad (3.6)$$

$$y^v \in \{0, 1\} \quad \forall v \in V \quad (3.7)$$

$$z_{ki}^v \in \{0, 1\} \quad \forall k \in N^v, \forall i \in N^s, \forall v \in V \quad (3.8)$$

$$x_{ij}^{vkl} \in \{0, 1\} \quad \forall (k,l) \in E^v, \forall (i,j) \in E^s, \forall v \in V \quad (3.9)$$

A função objetivo (3.1) minimiza o custo de alocação das redes virtuais na rede física, adicionando uma penalidade  $P$  para cada rede virtual não-atendida, o que faz com que o modelo atenda o maior número de redes virtuais possível.

As restrições (3.2) garantem que todos os nós de uma rede virtual serão alocados a algum nó da rede física, caso a rede virtual seja aceita. As restrições (3.3) garantem que nós de uma mesma rede virtual não serão alocados em um mesmo nó físico. A garantia que as capacidades de CPU dos nós físicos não serão excedidas pelos requisitos de CPU demandados pelos nós virtuais é dada pelas restrições (3.4). Da mesma forma, as restrições (3.5) garantem a capacidade de largura de banda para os enlaces físicos. Nas restrições (3.6), garante-se a conectividade entre os nós virtuais na rede física através da associação das requisições virtuais em caminhos físicos. As restrições (3.7)-(3.9) indicam o domínio das variáveis.

Uma função objetivo alternativa, que provê balanceamento de carga nas soluções, é dada por:

**VNE\_PLI\_LB:**

$$\min \left[ \alpha + P \cdot \sum_{v \in V} (1 - y^v) \right] \quad (3.10)$$

Com a adição de um novo conjunto de restrições:

$$\alpha \geq \frac{\sum_{v \in V} \sum_{(k,l) \in E^v} b_{kl}^v x_{ij}^{vkl}}{B_{ij}} \quad \forall (i, j) \in E^s \quad (3.11)$$

Seja  $\alpha$  a maior taxa de utilização entre todos os enlaces físicos. A função objetivo (3.10), juntamente com as restrições (3.11) minimiza a maior utilização dos enlaces, o que tende a distribuir as requisições pela rede física, evitando a sobrecarga de alguns enlaces. A penalidade também é aplicada sempre que uma requisição é rejeitada.

A Figura 3.2 apresenta um exemplo de mapeamento de duas VNs em uma rede física. Os valores internos a cada nó representado por losangos ou quadrados indicam os requisitos de CPU dos roteadores das redes virtuais, enquanto os valores sobre as linhas tracejadas indicam requisitos de largura de banda. O substrato físico é formado por quatro nós e cinco enlaces com suas capacidades associadas (os valores internos aos círculos indicam capacidade de CPU dos roteadores físicos, enquanto valores ao lado de cada enlace físico indicam capacidade de largura de banda dos enlaces). A rede virtual VN<sup>1</sup> é composta por três nós e dois enlaces, enquanto VN<sup>2</sup> é composta por dois nós e um enlace. À esquerda, o mapeamento utiliza a função objetivo de minimização do custo, considerando que todas as arestas possuem mesmo custo. Observa-se que existe uma aresta que está sobrecarregada, isto é, as demandas que passam por ela ocupam toda sua capacidade (100). À direita, uma solução onde o mapeamento está balanceado pois, embora a rede física esteja sendo mais utilizada, nenhuma aresta possui utilização maior que 62,5%.

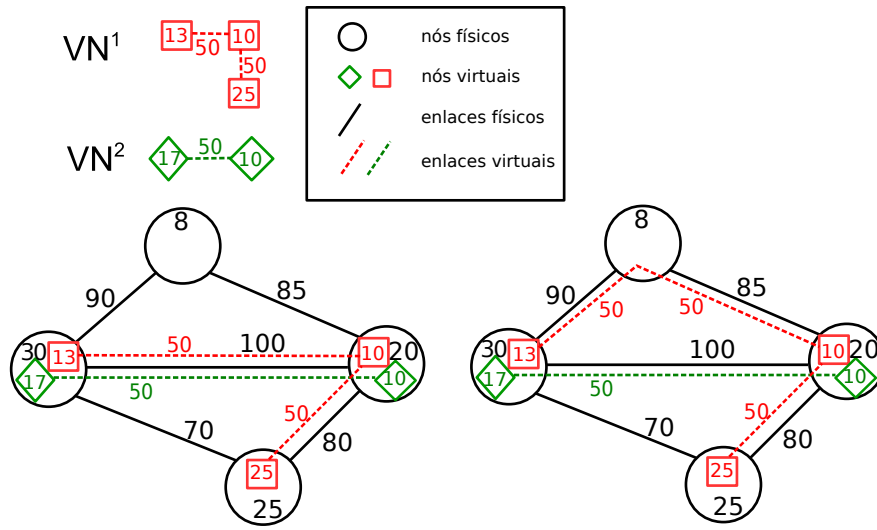


Figura 3.2: Mapeamentos de VNs sem e com balanceamento de carga

Uma terceira função objetivo (VNE\_PLI\_DL) pode ser proposta buscando minimizar o atraso na comunicação entre os componentes das requisições virtuais. Seja o parâmetro  $d_{ij}$  o atraso médio observado em cada enlace físico  $(i, j) \in E^s$ .

Para isso, a função objetivo pode ser expressa como:

**VNE\_PLI\_DL:**

$$\min \left[ \sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} d_{ij} x_{ij}^{vkl} + P \cdot \sum_{v \in V} (1 - y^v) \right] \quad (3.12)$$

A função objetivo (3.12) minimiza o atraso total de todas as requisições virtuais. Esse objetivo tende a associar os nós virtuais de uma mesma requisição próximos uns aos outros, para reduzir o atraso entre eles. A penalidade sobre requisições rejeitadas também é considerada neste cenário.

Objetivando criar cenários mais próximos às aplicações reais, restrições adicionais podem ser utilizadas para prover níveis de QoS. A seguir, são apresentados três conjuntos de restrições que podem ser adicionados aos modelos para tentar abranger mais particularidades do contexto de virtualização de redes.

### 1. Restrição de Resiliência (R)

As falhas são comuns nos cenários de rede, e resiliência é um parâmetro fundamental quando deseja prover QoS. Estudos mostram que falhas em enlace único representam 70% das falhas em redes [Iannaccone et al., 2002]. Embora contornar

falhas em nós também seja importante, antes de resolvê-las, uma solução para falhas em enlaces deve ser proposta, uma vez que tolerância à falhas em nós depende da tolerância à falhas nos enlaces adjacentes. A fim de garantir resiliência contra falha em enlace único na rede física, dois caminhos físicos disjuntos em enlaces devem existir para cada demanda virtual. Assim, as restrições (3.6) são alteradas para:

$$\sum_{(i,j) \in E^s} x_{ij}^{vkl} - \sum_{(h,i) \in E^s} x_{hi}^{vkl} = 2(z_{ki}^v - z_{li}^v) \quad \forall i \in N^s, \forall (k,l) \in E^v, \forall v \in V \quad (3.13)$$

## 2. Restrição de Localização Geográfica (L)

Restrições de localização geográfica podem ser de extrema importância dependendo dos requisitos dos usuários. Por exemplo, quando segurança e confidencialidade são prioridade, o usuário pode não querer que seus dados sejam hospedados em nós físicos específicos. Por outro lado, um usuário pode desejar que sua aplicação seja mais espalhada pela rede, aumentando a disponibilidade devido à diversidade geográfica. Para isso, os provedores de serviço podem incluir essa restrição adicional como um parâmetro de acordo de nível de serviço com seus clientes, permitindo que eles decidam onde seus dados poderão ser hospedados.

Considere a localização geográfica dos nós físicos e a localização geográfica preferencial dos nós virtuais. Assuma  $R^v$  como um raio máximo para cada rede  $v \in V$ , que define a maior distância entre o nó físico em que um nó virtual pode ser alocado e a sua localização preferencial. Então, as restrições (3.14) garantem que cada nó virtual é atribuído apenas a nós físicos que pertencem ao conjunto  $I^s \subseteq N^s$  tal que  $i \in I^s$  se e somente se  $d_{ki}^v \leq R^v$ . Dessa forma, a restrição (3.2) pode ser substituída por:

$$\sum_{i \in I^s} z_{ki}^v = y^v \quad \forall k \in N^v, \forall v \in V \quad (3.14)$$

## 3. Restrição de atraso (D)

Restrições de atraso estão diretamente relacionadas à qualidade de comunicação esperada entre um par de nós. Em aplicações críticas, esse parâmetro deve ser de alta prioridade. Portanto, considere parâmetros  $d_{ij}$  como o atraso médio observado em cada enlace físico  $(i,j) \in E^s$  e  $D_{kl}^v$  como o atraso máximo tolerado pelo enlace virtual  $(k,l) \in N^v$ . As restrições (3.15) garantem que a soma dos

atrasos de todos os enlaces físicos pelos quais um enlace virtual passa não excederá  $D_{kl}^v$ .

$$\sum_{(i,j) \in E^s} d_{ij} x_{ij}^{vkl} \leq D_{kl}^v \quad \forall (k, l) \in E^v, \forall v \in V \quad (3.15)$$

É importante citar que, quando as restrições de atraso são consideradas juntamente com as restrições de resiliência, o lado direito da equação (3.15) deve ser multiplicado por 2. Consequentemente, a restrição de atraso passa a impor um limite no atraso médio entre os caminhos que atendem uma requisição.

Os modelos compactos apresentados (VNE\_PLI\_COST, VNE\_PLI\_LB e VNE\_PLI\_DL) podem ser resolvidos através de um algoritmo de *branch-and-bound*, devido ao seu número polinomial de variáveis e restrições. Contudo, é esperado que apenas instâncias de pequeno porte tenham a otimalidade comprovada em tempo factível. Assim, são propostos modelos estendidos baseados em geração de colunas, com a finalidade de alcançar soluções ótimas para instâncias de maior porte.

### 3.3 Modelos Estendidos (VNE\_GC)

Entre as técnicas de PLI, a decomposição de Dantzig-Wolfe [George B. Dantzig, 1960] é conhecida por possibilitar a reformulação do problema através de um problema mestre restrito que utiliza um pequeno conjunto de variáveis iniciais; e um subproblema que, com base nos valores duais obtidos no problema mestre para gerar novas colunas (variáveis) que sejam interessantes para o problema mestre restrito. A principal vantagem dessa reformulação é a obtenção de modelos mais fortes, isto é, com melhores limites de relaxação linear para o problema.

No presente trabalho, são propostos dois modelos estendidos para o problema: VNE\_GC\_CONFIG, modelo baseado em configurações de redes, e VNE\_GC\_PATH, baseado em caminhos.

#### 3.3.1 Modelo Estendido baseado na geração de configurações (VNE\_GC\_CONFIG)

No primeiro modelo proposto, o subproblema consiste em gerar mapeamentos para as VNs, isto é, alocar os nós virtuais a nós físicos e encontrar caminhos físicos válidos para cada requisição virtual. O problema mestre minimiza os custos de largura de banda

na rede física e garante que as quantidades de CPU dos nós e de largura de banda dos enlaces alocadas às redes virtuais não excedem as capacidades da rede física.

### 3.3.1.1 Problema Mestre

Define-se como  $M^v$  o conjunto de mapeamentos possíveis para a rede virtual  $v \in V$  no substrato físico. Denota-se por  $C_m^v$  o custo do mapeamento  $m \in M^v$  definido como a largura de banda utilizada para atender tal rede. Sejam as seguintes variáveis de decisão:

- $\lambda_m^v$  indica se a rede virtual  $v \in V$  é mapeada no substrato físico através do mapeamento  $m \in M^v$  ( $\lambda_m^v = 1$ )
- $y^v$  indica se a rede virtual  $v \in V$  é atendida ( $y^v$ )

E os coeficientes:

- $a_{ki}^{mv}$  indica se, no mapeamento  $m \in M^v$ , o nó  $k \in N^v$  da rede virtual  $v \in V$  é mapeado no nó físico  $i \in N^s$  ( $a_{ki}^{mv} = 1$ )
- $e_{ij}^{mvkl}$  indica se, no mapeamento  $m \in M^v$ , o enlace virtual  $(k, l) \in E^v$  da rede  $v \in V$  utiliza o enlace físico  $(i, j) \in E^s$  ( $e_{ij}^{mvkl} = 1$ )

Função objetivo:

$$\min \left[ \sum_{v \in V} \sum_{m \in M^v} C_m^v \lambda_m^v + P \cdot \sum_{v \in V} (1 - y^v) \right] \quad (3.16)$$

Sujeito ao seguinte conjunto de restrições:

$$\sum_{m \in M^v} \lambda_m^v = y^v \quad \forall v \in V \quad (3.17)$$

$$\sum_{v \in V} \sum_{k \in N^v} \sum_{m \in M^v} c_v^k a_{ki}^{mv} \lambda_m^v \leq C_i \quad \forall i \in N^s \quad (3.18)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{m \in M^v} b_{kl}^v e_{ij}^{mvkl} \lambda_m^v \leq B_{ij} \quad \forall (i, j) \in E^s \quad (3.19)$$

$$\lambda_m^v \in [0, 1] \quad \forall v \in V, \forall m \in M^v \quad (3.20)$$

$$y^v \in [0, 1] \quad \forall v \in V \quad (3.21)$$

A função objetivo (3.16) minimiza o custo de largura de banda utilizada pelos mapeamentos, com uma penalidade  $P$  para cada rede virtual não atendida.

As restrições (3.17) garantem que para cada rede virtual aceita será designado um mapeamento. Os conjuntos de restrições (3.18) e (3.19) garantem que as capacidades



de CPU e largura de banda dos nós físicos não serão excedidas, respectivamente. O domínio das variáveis é definido pelas restrições (3.20) e (3.21).

### 3.3.1.2 Subproblema

O subproblema pode ser definido com as seguintes variáveis de decisão:

$z_{ki}^v$  indica se o nó virtual  $k \in N^v$  é mapeado no nó físico  $i \in N^s$  ( $z_{ki}^v = 1$ )  
 $x_{ij}^{vkl}$  indica se a requisição virtual  $(k, l) \in E^v$  passa pelo enlace físico  $(i, j) \in E^s$  pelo caminho que conecta os nós físicos que hospedam  $k, l \in N^v$  ( $x_{ij}^{vkl} = 1$ )

Sejam as variáveis duais  $\pi_v$ ,  $\alpha_i$  e  $\beta_{ij}$ , referentes às restrições (3.17), (3.18) e (3.19) do problema mestre, respectivamente.

O custo reduzido das novas colunas  $m$  para cada rede virtual  $v \in V$  é dado por:

$$\bar{C}_m^v = C_m^v + \pi_v + \sum_{k \in N^v} \sum_{i \in N^s} c_k^v \alpha_i z_{ki}^v + \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} b_{kl}^v \beta_{ij} x_{ij}^{vkl} \quad (3.22)$$

O custo de cada coluna representa o custo de mapeamento da rede  $v \in V$ :

$$C_m^v = \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} U_{ij} b_{kl}^v x_{ij}^{vkl} \quad (3.23)$$

Assim, o subproblema de precificação para cada rede virtual  $v \in V$  é dado por:

$$\min \left[ \pi_v + \sum_{k \in N^v} \sum_{i \in N^s} c_k^v \alpha_i z_{ki}^v + \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} b_{kl}^v (U_{ij} + \beta_{ij}) x_{ij}^{vkl} \right] \quad (3.24)$$

Sujeito ao conjunto de restrições:

$$\sum_{i \in N^s} z_{ki}^v = 1 \quad \forall k \in N^v \quad (3.25)$$

$$\sum_{k \in N^v} z_{ki}^v \leq 1 \quad \forall i \in N^s \quad (3.26)$$

$$\sum_{(i,j) \in E^s} x_{ij}^{vkl} - \sum_{(h,i) \in E^s} x_{hi}^{vkl} = z_{ki}^v - z_{li}^v \quad \forall i \in N^s, \forall (k, l) \in E^v \quad (3.27)$$

$$z_{ki}^v \in \{0, 1\} \quad \forall k \in N^v, \forall i \in N^s \quad (3.28)$$

$$x_{ij}^{vkl} \in \{0, 1\} \quad \forall (k, l) \in E^v, \forall (i, j) \in E^s \quad (3.29)$$

A função objetivo (3.24) minimiza o custo reduzido da coluna encontrada.

As restrições (3.25) garantem que todos os nós virtuais serão mapeados em algum nó físico, enquanto as restrições (3.26) impedem que nós virtuais da mesma rede sejam mapeados no mesmo nó físico. A existência de caminhos válidos entre os nós virtuais é garantida pelas restrições (3.27). As restrições (3.28) e (3.29) determinam o domínio das variáveis  $z$  e  $x$ , respectivamente.

### 3.3.2 Modelo Estendido baseado na geração de caminhos (VNE\_GC\_PATH)

O segundo modelo proposto, baseia-se na geração de caminhos físicos para as demandas virtuais. Neste modelo, o problema mestre aloca os roteadores virtuais aos roteadores físicos, garantindo capacidades de CPU, e escolhe os caminhos, respeitando as capacidades de largura de banda dos enlaces. O subproblema é responsável por gerar caminhos físicos para as demandas virtuais.

#### 3.3.2.1 Problema Mestre

Variáveis

- $\lambda_m^{vkl}$  indica se o link virtual  $(k, l) \in E^v$  da requisição virtual  $v \in V$  usa o caminho  $p$  ( $\lambda_m^{vkl} = 1$ )
- $z_i^{vk}$  indica se o nó virtual  $k \in N^v$  da requisição virtual  $v \in V$  é mapeado no nó físico  $i \in N^s$  ( $z_i^{vk} = 1$ )
- $y^v$  indica se a requisição virtual  $v \in V$  é aceita ou não ( $y^v = 1$ )

Coefficientes:

- $a_{ki}^{mv}$  indica se o nó  $k \in N^v$  da requisição virtual  $v \in V$  está hospedado no nó  $i \in N^s$  no caminho  $m$  ( $a_{ki}^{mv} = 1$ )
- $e_{ij}^m$  indica se o link físico  $(i, j) \in E^s$  faz parte do caminho  $m$  ( $e_{ij}^m = 1$ )
- $M_{kl}^v$  conjunto de caminhos físicos que atendem o link virtual  $(k, l) \in E^v$  de  $v \in V$
- $P$  penalidade por não atender uma requisição

Função objetivo:

$$\min \left[ \sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{m \in M_{kl}^v} c_m^{vkl} \lambda_m^{vkl} + P \sum_{v \in V} (1 - y^v) \right] \quad (3.30)$$

Sujeito ao conjunto de restrições:

$$\sum_{i \in N^s} z_i^{vk} = y^v \quad \forall k \in N^v, \forall v \in V \quad (3.31)$$

$$\sum_{k \in N^v} z_{ki}^v \leq 1 \quad \forall i \in N^s, \forall v \in V \quad (3.32)$$

$$\sum_{v \in V} \sum_{k \in N^v} c_v^k z_{ki}^v \leq C_i \quad \forall i \in N^s \quad (3.33)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{m \in M_{kl}^v} b_{kl}^v e_{ij}^m \lambda_m^{vkl} \leq B_{ij} \quad \forall (i,j) \in E^s \quad (3.34)$$

$$\sum_{m \in M_{kl}^v} \lambda_m^{vkl} = y^v \quad \forall (k,l) \in E^v, \forall v \in V \quad (3.35)$$

$$\sum_{m \in M_{kl}^v} a_{ki}^{pv} \lambda_m^{vkl} \leq z_{ki}^v \quad \forall i \in N^s, \forall (k,l) \in E^v, \forall v \in V \quad (3.36)$$

$$\sum_{m \in M_{kl}^v} a_{li}^{pv} \lambda_m^{vkl} \leq z_{li}^v \quad \forall i \in N^s, \forall (k,l) \in E^v, \forall v \in V \quad (3.37)$$

$$y^v \in [0, 1] \quad \forall v \in V \quad (3.38)$$

$$z_{ki}^v \in [0, 1] \quad \forall k \in N^v, \forall i \in N^s, \forall v \in V \quad (3.39)$$

$$\lambda_m^{vkl} \in [0, 1] \quad \forall m \in M_{kl}^v, \forall (k,l) \in E^v, \forall v \in V \quad (3.40)$$

A função objetivo (3.30) minimiza o custo de alocação dos caminhos utilizados, adicionando uma penalidade  $P$  para cada rede virtual não-atendida, tentando atender o maior número de redes virtuais possível.

As restrições (3.31) garantem que todos os nós de uma rede virtual serão alocados a algum nó da rede física, se esta rede virtual for aceita. As restrições (3.32) garantem que nós de uma mesma rede virtual não serão alocados em um mesmo nó físico. A garantia que as capacidades de CPU dos nós físicos não serão excedidas pelos requisitos de CPU demandados pelos nós virtuais é dada pelas restrições (3.33). Da mesma forma, as restrições (3.34) garantem que a capacidade de largura de banda para os enlaces físicos não serão excedidas pelos caminhos. As restrições (3.36) e (3.37) garantem que as alocações dos nós virtuais serão compatíveis com os nós iniciais e finais dos caminhos físicos. Por fim, as restrições (3.7)–(3.9) indicam o domínio das variáveis.

### 3.3.2.2 Subproblema

O subproblema pode ser definido com as seguintes variáveis de decisão:

- $o_i^{vkl}$  indica se o caminho físico que atende a demanda  $(k,l) \in E^v$  da rede virtual  $v \in V$  tem origem no nó  $i \in N^s$  ( $o_i^{vkl} = 1$ )
- $d_i^{vkl}$  indica se o caminho físico que atende a demanda  $(k,l) \in E^v$  da rede virtual  $v \in V$  tem como final o nó  $i \in N^s$  ( $d_i^{vkl} = 1$ )
- $x_{ij}^{vkl}$  indica se o enlace físico  $(i,j) \in E^s$  faz parte do caminho que atende a demanda  $(k,l) \in E^v$  da rede virtual  $v \in V$  ( $x_{ij}^{vkl} = 1$ )

O custo reduzido das novas colunas  $m$  para cada demanda  $(k, l) \in E^s$  da rede virtual  $v \in V$  é dado por:

$$\bar{C}_m^{vkl} = C_m^{vkl} + \alpha^{vkl} + \sum_{i \in N^s} \gamma_i^{vkl} o_i^{vkl} + \sum_{i \in N^s} \pi_i^{vkl} d_i^{vkl} + \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} b_{kl}^v \beta_{ij} x_{ij}^{vkl} \quad (3.41)$$

Onde  $\beta^{ij}$ ,  $\alpha^{vkl}$ ,  $\gamma_i^{vkl}$  e  $\pi_i^{vkl}$  são as variáveis duais referentes às restrições (3.34)–(3.37) do problema mestre, nessa ordem.

O custo de cada coluna representa o custo de mapeamento da rede  $v \in V$ :

$$C_m^{vkl} = \sum_{(i,j) \in E^s} U_{ij} b_{kl}^v x_{ij}^{vkl} \quad (3.42)$$

Assim, o subproblema de precificação para cada rede virtual  $v \in V$  é dado por:

$$\min \left[ \alpha^{vkl} + \sum_{i \in N^s} \gamma_i^{vkl} o_i^{vkl} + \sum_{i \in N^s} \pi_i^{vkl} d_i^{vkl} + \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^s} (U_{ij} - \beta_{ij}) b_{kl}^v x_{ij}^{vkl} \right] \quad (3.43)$$

Sujeito ao conjunto de restrições:

$$\sum_{i \in N^s} o_i^{vkl} = 1 \quad (3.44)$$

$$\sum_{i \in N^s} d_i^{vkl} = 1 \quad (3.45)$$

$$o_i^{vkl} + d_i^{vkl} \leq 1 \quad \forall i \in N^s \quad (3.46)$$

$$\sum_{(i,j) \in E^s} x_{ij}^{vkl} - \sum_{(h,i) \in E^s} x_{hi}^{vkl} = o_i^{vkl} - d_i^{vkl} \quad \forall i \in N^s, \forall (k, l) \in E^v \quad (3.47)$$

$$o_i^{vkl} \in \{0, 1\} \quad \forall i \in N^s \quad (3.48)$$

$$d_i^{vkl} \in \{0, 1\} \quad \forall i \in N^s \quad (3.49)$$

$$x_{ij}^{vkl} \in \{0, 1\} \quad \forall (k, l) \in E^v, \forall (i, j) \in E^s \quad (3.50)$$

A função objetivo (3.43) minimiza o custo reduzido da nova coluna.

As restrições (3.44) e (3.45) garantem a alocação dos nós físicos inicial e final do caminho físico, respectivamente. Pelas restrições (3.46), evita-se a escolha de um mesmo nó físico para início e fim do caminho. As restrições (3.47) garantem a conectividade do caminho. Os domínios das variáveis são determinados pelas restrições (3.48)–(3.50).

# Capítulo 4

## Algoritmos

Os modelos VNE\_GC permitem a utilização de algoritmos de geração de colunas, onde uma pequena parcela das variáveis do problema mestre são geradas pelo subproblema, embora o número de variáveis desses modelos seja exponencial em termos teóricos.

### 4.1 Algoritmos de Geração de Colunas

#### 4.1.1 Geração de Configurações

O Algoritmo 1 apresenta um pseudocódigo do algoritmo de geração de colunas proposto. Inicialmente, é gerado um conjunto inicial de colunas para o problema (linha 2). Em seguida, a cada iteração, resolve-se o problema mestre restrito, obtendo os valores das variáveis duais (linhas 4 e 5). Para cada rede virtual  $v \in V$ , resolve-se o subproblema para gerar uma nova configuração, obtendo uma nova coluna (linhas 6 a 11). Esta coluna é adicionada ao problema mestre restrito, caso apresente um custo reduzido atrativo (custo reduzido  $< 0$ ) (linha 4). O processo termina quando nenhuma coluna atrativa é gerada (loop entre as linhas 3 e 12).

#### 4.1.2 Geração de Caminhos

No Algoritmo 2, é apresentado o pseudocódigo do algoritmo de geração de colunas baseado em caminhos. No começo do algoritmo, é gerado um conjunto inicial de colunas para o problema (linha 2). Em seguida, a cada iteração, resolve-se o problema mestre restrito, obtendo os valores das variáveis duais (linha 5). Para cada enlace  $(k, l) \in E^v$  de cada rede virtual  $v \in V$ , resolve-se o subproblema, obtendo um caminho físico (coluna) (linha 8 do *loop* das linhas 6 a 13). O caminho é adicionado ao problema

---

**Algoritmo 1** Pseudocódigo do algoritmo de Geração de Configurações

---

```

1: função GERAÇÃO DE COLUNAS
2:   Gerar conjunto inicial de colunas
3:   enquanto Existir coluna atrativa faça
4:     Adicionar colunas ao Problema Mestre Restrito
5:     Resolver Problema Mestre Restrito
6:     para cada rede virtual  $v \in V$  faça
7:       Resolver Subproblema para  $v$ 
8:       se Custo reduzido da nova coluna  $< 0$  então
9:         Adicionar nova coluna a um pool
10:      fim se
11:    fim para
12:  fim enquanto
13:  retorna Solução Atual do Problema Mestre
14: fim função

```

---

mestre restrito, caso seu custo reduzido seja atrativo (custo reduzido  $< 0$ ) (linhas 4 e 10). O processo de geração de colunas é repetido até que nenhum caminho atrativo seja encontrado (*loop* entre as linhas 3 e 14).

---

**Algoritmo 2** Pseudocódigo do algoritmo de Geração de Caminhos

---

```

1: função GERAÇÃO DE COLUNAS
2:   Gerar conjunto inicial de colunas
3:   enquanto Existir coluna atrativa faça
4:     Adicionar colunas ao Problema Mestre Restrito
5:     Resolver Problema Mestre Restrito
6:     para cada rede virtual  $v \in V$  faça
7:       para cada enlace virtual  $(k, l) \in E^v$  faça
8:         Encontrar caminho físico para  $(k, l)$ 
9:         se Custo reduzido do caminho  $< 0$  então
10:          Adicionar nova coluna a um pool
11:        fim se
12:      fim para
13:    fim para
14:  fim enquanto
15:  retorna Solução Atual do Problema Mestre
16: fim função

```

---

## 4.2 Algoritmos *Branch-and-Price*

*Branch-and-Price* [Barnhart et al., 1998] é um método de resolução de Problemas de Programação Linear Inteira (PLI) e Problemas de Programação Linear Inteira Mista

(PLIM). O método se baseia em um algoritmo de *Branch-and-Bound* [Doig et al., 1960] com a possibilidade de geração de novas colunas em cada nó da árvore.

São desenvolvidos dois algoritmos *branch-and-price*: VNE\_BP\_CONFIG, que utiliza o modelo VNE\_GC\_CONFIG, e VNE\_BP\_PATH, baseada no modelo VNE\_GC\_PATH.

### 4.2.1 Estratégia de *branching*

A regra de *branching* “padrão” escolhe uma variável fracionária  $\lambda$  e então, deriva dois novos problemas, onde  $\lambda$  é fixado respectivamente em 0 e 1. Essa abordagem, no entanto, implica em algumas dificuldades, como o custo de impedir que as colunas removidas sejam geradas novamente, além de provocar desbalanceamento na árvore de busca [Vanderbeck, 2011].

Com base nisso, a estratégia de *branching* adotada nesse trabalho é realizada em três níveis:

#### 4.2.1.1 Branching em aceitação/rejeição de rede virtual

São derivados novos problemas:

- No primeiro problema, a rede virtual  $v \in V$  é rejeitada, isto é,  $y^v = 0$
- No segundo, a rede virtual  $v \in V$  é aceita, isto é,  $y^v = 1$

Os dois modelos (VNE\_GC\_CONFIG e VNE\_GC\_PATH) possuem a variável  $y^v$  no problema mestre. Assim, caso exista, escolhe-se a variável  $y^v$  fracionária mais próxima de 0,5.

#### 4.2.1.2 Branching em alocação dos nós

São derivados dois novos problemas:

- No primeiro, o nó virtual  $k \in N^v$  da rede virtual  $v \in V$  não pode ser alocado ao nó físico  $i \in N^s$ , isto é,  $z_{ki}^v = 0$
- No segundo problema, o nó virtual  $k \in N^v$  da rede virtual  $v \in V$  deve ser alocado ao nó físico  $i \in N^s$ , isto é,  $z_{ki}^v = 1$

A escolha da variável a ser derivada é feita inicialmente nas variáveis  $z_{ki}^v$ , como segue:

Para VNE\_BP\_CONFIG, escolhe-se o mapeamento de um nó virtual  $k \in N^v$  de uma rede virtual  $v \in V$  a um nó físico  $i \in N^s$ , tal que  $0 < u_{ki}^v < 1$ , onde  $u_{ki}^v = \sum_{m \in M^v} a_{ki}^{mv} \lambda_m^v$ . Opta-se pelo  $u_{ki}^v$  mais próximo de 0,5.

Como o modelo VNE\_GC\_PATH possui variáveis  $z_{ki}^v$  no problema mestre, para derivar os novos problemas, basta escolher aquela variável fracionária mais próxima de 0,5.

#### 4.2.1.3 Branching em alocação dos enlaces

Também são derivados dois novos problemas:

- Um problema, onde o caminho que atende a demanda virtual  $(k, l) \in E^v$  da rede virtual  $v \in V$  não pode conter o enlace físico  $(i, j) \in E^s$ , isto é,  $x_{ij}^{vkl} = 0$
- Outro, onde o caminho que atende a demanda virtual  $(k, l) \in E^v$  da rede virtual  $v \in V$  deve passar pelo enlace físico  $(i, j) \in E^s$ , isto é,  $x_{ij}^{vkl} = 1$

É escolhido o mapeamento de uma demanda virtual  $(k, l) \in E^v$  de uma rede virtual  $v \in V$  em um enlace físico  $(i, j) \in E^s$ , tal que  $0 < u_{ij}^{vkl} < 1$ , onde  $u_{ij}^{vkl} = \sum_{m \in M^v} e_{ij}^{mvkl} \lambda_m^v$  para VNE\_GC\_CONFIG, e  $u_{ij}^{vkl} = \sum_{m \in M^{vkl}} e_{ij}^{mv} \lambda_m^v$  para VNE\_GC\_PATH.

#### 4.2.2 Solução Inicial

Para gerar um conjunto inicial de colunas para o problema mestre restrito, é utilizada a heurística construtiva proposta por [Araújo & Souza, 2015]. A heurística busca um mapeamento onde, em cada etapa, são priorizados os nós físicos menos saturados. Esse conjunto de colunas garante uma solução básica viável inicial.

O Algoritmo 3 apresenta a heurística utilizada para gerar a solução inicial dos algoritmos *branch-and-price*. Para cada rede virtual, o algoritmo busca realizar o mapeamento em duas etapas: mapeamento dos nós (linha 3 a 11) e mapeamento dos enlaces (linhas 12 a 21). Para o mapeamento dos nós, cria-se uma lista restrita de candidatos (LRC), que considera o grau de saturação dos nós (linha 3). Após o mapeamento dos nós virtuais da rede, para cada enlace virtual, utiliza-se o algoritmo Dijkstra para encontrar um caminho mínimo entre os nós físicos aos quais os nós virtuais do enlace foram mapeados (linhas 13 a 15).

O grau de saturação dos nós é dada pela fórmula:

$$Sat(n) = R(n) \cdot \sum_{e \in E^s} R(e)$$



Onde  $R(n)$  é a capacidade residual do nó físico  $n$  e  $R(e)$  é a capacidade residual de cada enlace  $e$  conectado ao nó  $n$ .

---

**Algoritmo 3** Pseudocódigo da heurística para a solução inicial

---

```

1: função HEURÍSTICA CONSTRUTIVA
2:   para cada rede virtual  $v \in V$  faça
3:     LRC  $\leftarrow$  CalculaLRC( $v$ )
4:     para cada nó virtual  $n \in N^v$  faça
5:       ret  $\leftarrow$  Mapeia( $n$ , LRC)
6:       se ret = false então
7:         ERRO no mapeamento dos nós
8:       senão
9:         Adiciona mapeamento à Solução
10:      fim se
11:    fim para
12:    para cada enlace virtual  $e \in E^v$  faça
13:       $n_1 \leftarrow$  1º nó físico do enlace  $e^V$ 
14:       $n_2 \leftarrow$  2º nó físico do enlace  $e^V$ 
15:      ret  $\leftarrow$  Dijkstra( $n_1$ ,  $n_2$ )
16:      se ret = false então
17:        ERRO no mapeamento dos enlaces
18:      senão
19:        Adiciona caminho na Solução
20:    fim se
21:  fim para
22:  fim para
23:  retorna Solução
24: fim função

```

---

### 4.2.3 Exploração da *Árvore de Branch-and-Price*

Quando novos problemas (ou nós da árvore) são derivados, eles contêm algumas informações provindas do nó “pai”, tais como limites, colunas geradas, decisões de *branching*, etc. Ao derivar novos problemas e fixar variáveis, as colunas do nó pai são mantidas, removendo aquelas que não satisfazem a nova restrição.

Utiliza-se uma estratégia *best-bound* para a exploração dos nós, onde a cada iteração, escolhe-se para ser explorado, o nó cujo pai possui menor limite inferior.

### 4.2.4 Limites superiores

Em cada nó da árvore, para gerar soluções inteiras, utiliza-se uma heurística primal. Após resolver a relaxação da geração de colunas, o problema é resolvido com o conjunto

atual de colunas de forma inteira. Foi determinado um tempo de resolução de 30 segundos para essa etapa, pois o tempo pode ser muito grande conforme o número de colunas geradas.

# Capítulo 5

## Resultados Computacionais

Nesta seção, são descritas as características das instâncias utilizadas. Em seguida, apresentados os resultados utilizando os modelos compactos com as diferentes funções objetivo e restrições adicionais, resolvidas através de um algoritmo *branch-and-bound*, utilizando os modelos VNE\_PLI. Por fim são mostrados os resultados com os algoritmos de geração de colunas e *branch-and-price*.

Para avaliar o desempenho dos algoritmos e o impacto das funções objetivo e restrições adicionais, os modelos foram implementados em C++, utilizando o pacote de otimização CPLEX versão 12.6.1 [ILOG, 2013] (incluindo o problema mestre restrito quanto os subproblemas de ambos os modelos *branch-and-price*). Os testes foram conduzidos em um computador com processador Intel Core i7-4790K 4.00GHz, com 16GB de memória RAM, sobre sistema operacional Ubuntu 16.04.

### 5.1 Instâncias

Para os experimentos com os modelos apresentados, foram utilizados os mesmos conjuntos de instâncias utilizados em [Chowdhury et al., 2009]<sup>1</sup>. Através da ferramenta GT-ITM, os autores geraram uma rede do substrato aleatoriamente com 50 vértices, posicionados em um *grid* ( $25 \times 25$ ). Cada par de vértices do substrato está conectado com uma probabilidade de 50%. Os recursos de CPU e de largura de banda dos nós e dos enlaces do substrato correspondem a números reais distribuídos uniformemente entre 50 e 100.

Nós virtuais também encontram-se posicionados em um *grid* ( $25 \times 25$ ). Os requisitos de CPU dos nós virtuais são distribuídos uniformemente entre 0 e 20 e os

---

<sup>1</sup><http://www.mosharaf.com/ViNE-Yard.tar.gz>

requisitos de largura de banda dos enlaces virtuais entre 0 e 50. O grau médio de conectividade das VNs foi fixado em 50%.

Devido ao grande número de VNs, nos experimentos, a quantidade de VNs foi reduzida a fim de ser possível resolver de forma exata, em uma abordagem estática, onde essas requisições são conhecidas *a priori*.

## 5.2 Impacto de diferentes funções objetivo e restrições

Inicialmente, foram executados testes com algoritmos *branch-and-bound* baseados nos modelos VNE\_PLI\_COST, VNE\_PLI\_LB e VNE\_PLI\_DL, juntamente com as restrições de QoS apresentadas na Seção 3.2.1. Foram utilizadas as instâncias apresentadas na Seção 5.1, restringindo o número de redes virtuais em 5, 10, 15 e 20.

Os testes foram realizados para as funções objetivo VNE\_PLI\_COST, VNE\_PLI\_LB e VNE\_PLI\_DL, com as restrições adicionais citadas na Seção 3.2.1, que foram adicionadas uma-a-uma, e combinadas. As Tabelas 5.1, 5.2 e 5.3 resumizam os resultados para todas as instâncias e as três funções objetivo, respectivamente. A primeira coluna indica a instância de acordo com o tamanho do conjunto de redes virtuais ( $|V| \in \{5, 10, 15, 20\}$ ). A segunda coluna indica quais restrições adicionais são consideradas: Resiliência (R), Localização Geográfica (L) ou Atraso (D), e suas combinações. O caractere ‘-’ nesta coluna indica que foi utilizado o modelo sem o uso de quaisquer restrições adicionais. A coluna ‘Objetivo’ refere-se ao custo da solução de acordo com a função objetivo utilizada. É importante notar que as penalidades não são consideradas nos valores apresentados nas tabelas. A coluna ‘Taxa de Aceit. (%)’ indica a porcentagem de redes virtuais que tiveram suas demandas atendidas. As duas colunas seguintes mostram a porcentagem dos enlaces utilizada na solução e a média de largura de banda utilizada de tais enlaces, respectivamente. As mesmas métricas são apresentadas em relação à utilização dos nós. Por fim, ‘Tempo (seg.)’ apresenta os valores em segundos do tempo gasto pelo algoritmo. Valores maiores ou iguais a 3600 indicam que o algoritmo foi interrompido por atingir o limite de tempo estipulado em 1 hora, retornando a melhor solução encontrada.

A Tabela 5.1 mostra que instâncias maiores (aumentando o número de VNs) apresentam um maior custo na função objetivo, considerando os casos em que todas as requisições são aceitas. Além disso, em todos os casos, pode-se observar que a taxa de aceitação tende a diminuir quando o tamanho das instâncias aumenta. Considerando o modelo original, i.e., sem a adição de outras restrições, todas as instâncias apresentadas

Tabela 5.1: Resultados para o objetivo VNE\_PLI\_COST (Minimizar Custo de Mapeamento)

$ V $	Adic.	Objetivo	Taxa de Aceitação	Enlaces Utilizados (%)	Média de Utilização dos Enlaces (%)	Nós Utilizados (%)	Média de Utilização dos Nós (%)	Temp (seg.)
5	D	664,63	100,00	5,52	61,76	28,00	16,85	12,47
	D R	3581,11	100,00	16,57	62,58	34,00	12,96	3614,68
	R	3599,35	100,00	16,57	62,34	36,00	12,66	3600,42
	L	3257,67	100,00	8,84	41,03	28,00	18,49	0,06
	L D	2030,20	80,00	6,63	38,83	26,00	15,69	0,10
	L D R	1293,50	40,00	6,63	46,57	10,00	7,53	0,20
	L R	8860,26	100,00	19,34	56,82	28,00	19,00	0,61
	-	664,63	100,00	5,52	61,76	28,00	16,75	15,70
10	D	1643,72	100,00	12,15	57,07	46,00	22,96	72,65
	D R	9809,56	100,00	30,39	68,18	58,00	18,57	3601,66
	R	9842,69	100,00	33,15	63,56	56,00	20,94	3601,33
	L	7873,80	100,00	14,92	57,94	48,00	22,81	0,26
	L D	4898,60	80,00	9,39	61,29	34,00	23,12	0,26
	L D R	3944,40	40,00	13,26	49,25	16,00	15,82	1,75
	L R	22769,90	100,00	38,67	65,97	52,00	21,71	14,27
	-	1643,72	100,00	12,15	57,30	46,00	22,51	354,20
15	D	2554,86	100,00	14,36	66,99	56,00	28,25	2585,31
	D R	10194,75	86,67	32,60	64,95	58,00	18,63	3601,82
	R	20993,70	100,00	50,83	60,95	60,00	27,19	3600,02
	L	12347,10	100,00	22,10	59,46	58,00	28,71	0,86
	L D	5846,15	73,33	12,71	53,57	40,00	24,79	0,88
	L D R	7282,60	40,00	16,57	59,32	22,00	15,60	4,10
	L R	36742,40	100,00	56,35	66,48	68,00	25,08	3601,41
	-	2606,34	100,00	16,02	58,65	58,00	28,63	3602,42
20	D	4100,83	100,00	20,44	62,15	72,00	32,65	3602,87
	D R	9749,70	70,00	32,04	60,99	60,00	17,66	3600,03
	R	23585,80	90,00	58,01	59,97	80,00	23,00	3600,04
	L	15535,70	100,00	28,18	60,20	62,00	36,54	1,47
	L D	8630,80	80,00	19,34	57,92	52,00	31,23	1,47
	L D R	7235,00	30,00	16,57	58,09	22,00	16,39	18,78
	L R	55631,30	100,00	72,93	74,16	76,00	30,52	3601,56
	-	3993,11	100,00	19,89	65,02	68,00	34,43	3600,03

tiveram suas requisições completamente atendidas pelo algoritmo. Porém, a adição de outras restrições faz com que a taxa de aceitação diminua. Observa-se também, que quando as restrições são adicionadas individualmente, a taxa de aceitação permanece inalterada, exceto para a instância com 20 redes virtuais. Quando as restrições são adicionadas combinadas, a taxa de aceitação diminui significativamente, indicando a dificuldade em atendê-las simultaneamente. Note que a combinação “L+D+R” leva à menores taxas de aceitação para todas as instâncias. O tempo de execução mostra que o problema pode ser resolvido na otimalidade em poucos segundos quando os nós são restringidos à localidades geográficas específicas (não considerando outros parâmetros adicionais), diminuindo o número de possibilidades na alocação. Quando a restrição de localização geográfica não é considerada, o algoritmo atinge o limite de tempo estipulado, retornando a melhor solução viável encontrada. Considerando a porcentagem de

Tabela 5.2: Resultados para o objetivo VNE\_PLI\_LB (Balanceamento de Carga)

$ V $	Adic.	Objetivo	Taxa de Aceitação	Enlaces Utilizados (%)	Média de Utilização dos Enlaces (%)	Nós Utilizados (%)	Média de Utilização dos Nós (%)	Temp (seg.)
5	D	49,24	100	11,05	37,03	30	16,19	0,79
	D R	69,08	100	29,28	44,11	32	15,42	56,94
	R	51,96	100	69,06	35,73	32	14,03	25,13
	A	52,8	100	41,44	29,39	30	16,96	0,76
	A D	49,3	80	8,29	36,83	28	14,01	0,22
	A D R	70,7	40	8,84	44,18	10	9,56	6,23
	A R	55,26	100	61,88	40,81	34	15,42	11,71
	-	49,24	100	11,05	37,57	34	14,42	1,37
10	D	49,99	100	22,1	34,32	56	20,32	15,79
	D R	70,43	100	56,35	45,45	56	19,01	3600,04
	R	57,88	100	97,24	45,48	54	21,11	3620,91
	A	56,1	100	37,57	34,96	58	19,31	4,75
	A D	90,5	80	14,92	44,38	38	19,17	0,61
	A D R	79	40	19,34	38,93	22	12,67	41,98
	A R	73,79	100	92,82	49,03	60	18,41	408,65
	-	49,99	100	49,17	27,54	58	19,33	17,41
15	D	49,99	100	32,6	34,45	72	23,17	25,98
	D R	78,65	93,33	62,43	50,64	62	22,3	3600,04
	R	89,97	100	92,82	56,05	76	20,92	3600,03
	A	58,96	100	67,4	38,83	76	21,67	26,28
	A D	63,25	73,33	19,34	39,77	44	21,27	1,54
	A D R	79	40	29,83	38,48	30	11,45	172,3
	A R	87,57	100	96,13	62,15	70	23,76	3600,02
	-	49,99	100	39,23	29,39	74	22,2	47,84
20	D	49,99	100	39,23	34,58	86	27,67	178,87
	D R	88	75	62,43	50,45	52	21,94	3600,03
	R	99,02	100	100	84,22	90	26,35	3600,02
	A	58,96	100	87,85	39,47	78	29,53	520,26
	A D	70,6	80	38,67	38,18	54	29,18	28,57
	A D R	79	30	29,28	39,68	28	12,09	495,51
	A R	89,7	90	98,9	69,94	70	25,65	3600,03
	-	49,99	100	59,67	29,98	86	27,16	609,88

nós utilizados, nota-se que os valores são similares para todas as variações de restrições adicionais em uma mesma instância, exceto nos casos em que a taxa de aceitação é menor, pois nestes casos, são utilizados menos nós físicos. Em relação à utilização dos enlaces, é importante destacar que todas as versões que incluem resiliência apresentam maiores valores. Uma vez que prover sobrevivência contra falhas consome mais recursos para alocar largura de banda extra, a utilização de enlaces é maior nesses casos.

De acordo com a Tabela 5.2, o problema se torna mais difícil quando balanceamento de carga é considerado como função objetivo. Pode ser visto que a taxa de aceitação diminui na maioria dos casos, comparado às soluções obtidas com VNE\_PLI\_COST. Em geral, o número de enlaces e nós utilizados nas soluções é superior aquele obtido com a função objetivo VNE\_PLI\_COST. Porém, a média de utilização dos recursos nas soluções de VNE\_PLI\_LB é menor, mostrando que os enlaces virtuais encontram-se mais espalhados pela rede física. O número de casos

Tabela 5.3: Resultados para o objetivo VNE\_PLI\_DL (Minimizar Atraso)

$ V $	Adic.	Objetivo	Taxa de Aceitação	Enlaces Utilizados (%)	Média de Utilização dos Enlaces (%)	Nós Utilizados (%)	Média de Utilização dos Nós (%)	Temp (seg.)
5	D	21,32	100,00	4,97	64,83	22,00	20,24	41,16
	D R	109,52	100,00	16,57	62,58	36,00	13,12	3610,79
	R	108,89	100,00	14,92	69,54	36,00	13,09	3600,37
	L	114,56	100,00	8,29	44,49	28,00	19,37	0,06
	L D	71,40	80,00	7,18	37,07	26,00	15,69	0,10
	L D R	40,40	40,00	6,63	46,57	10,00	7,53	0,27
	L R	289,75	100,00	17,68	61,55	30,00	17,75	1,03
	-	21,32	100,00	4,97	64,74	22,00	20,76	69,58
10	D	59,25	100,00	11,05	62,89	46,00	22,48	234,80
	D R	345,67	100,00	34,25	62,10	52,00	20,39	3600,02
	R	334,18	100,00	30,94	68,24	60,00	17,71	3600,02
	L	292,29	100,00	14,92	58,41	46,00	24,13	0,18
	L D	172,00	80,00	9,94	58,40	32,00	24,33	0,23
	L D R	169,50	40,00	13,81	46,94	20,00	11,42	2,10
	L R	809,92	100,00	38,67	63,75	52,00	21,70	136,51
	-	59,25	100,00	11,05	62,47	46,00	22,72	420,81
15	D	94,47	100,00	13,26	70,73	52,00	31,50	3601,95
	D R	472,25	86,67	39,23	63,63	60,00	19,86	3601,76
	R	668,83	100,00	47,51	66,17	68,00	24,05	3600,02
	L	466,92	100,00	21,55	58,87	56,00	29,17	0,78
	L D	222,95	73,33	11,60	62,14	36,00	25,57	0,84
	L D R	287,80	40,00	17,13	56,84	22,00	16,75	5,41
	L R	1309,23	100,00	56,91	64,17	64,00	26,39	3289,46
	-	92,61	100,00	14,36	65,26	52,00	30,39	1959,91
20	D	147,32	100,00	19,89	63,64	68,00	33,96	3603,69
	D R	397,50	75,00	35,91	62,80	58,00	19,97	3600,03
	R	1447,90	95,00	74,59	61,16	84,00	24,94	3600,05
	L	612,90	100,00	27,07	63,54	62,00	36,77	3,05
	L D	352,20	80,00	18,78	60,36	50,00	31,79	2,21
	L D R	288,00	30,00	17,13	55,65	22,00	16,50	14,50
	L R	1984,23	100,00	73,48	71,55	72,00	32,25	3601,61
	-	148,57	100,00	18,23	71,00	58,00	39,23	3601,62

resolvidos dentro do tempo limite é superior ao objetivo VNE\_PLI\_COST. É possível notar que os casos em que o limite de tempo foi atingido são aqueles que envolvem a restrição de resiliência.

Os resultados sumarizados para o VNE\_PLI\_DL são mostrados na Tabela 5.3. É importante notar que a adição da restrição de atraso afeta a solução mesmo quando a função objetivo é a minimização do atraso. A função objetivo VNE\_PLI\_DL minimiza o atraso total de todas as requisições virtuais, enquanto as restrições de atraso impõem limites de atraso aos caminhos individualmente. VNE\_PLI\_COST e VNE\_PLI\_DL têm, na maioria dos casos, taxas de aceitação semelhantes, embora VNE\_PLI\_DL obtenha maiores taxas de aceitação em alguns casos com a instância de 20 VNs. Outra observação interessante está relacionada à utilização dos componentes físicos. O percentual de nós e links utilizados é menor que VNE\_PLI\_COST e VNE\_PLI\_LB, enquanto a média de utilização é maior. Esse comportamento indica que os compo-

mentes virtuais são alocados próximos, sobrecarregando um conjunto de nós e enlaces físicos para atender a minimização do atraso. Novamente, é possível notar que o número de casos resolvidos com prova de otimalidade é maior que VNE\_PLI\_COST e os casos em que o limite de tempo foi atingido envolvem a restrição de resiliência.

Observa-se que as restrições de localização geográfica tornam o problema mais fácil de ser resolvido, por restringir as possibilidades de mapeamento dos nós. Conforme permite-se o mapeamento dos nós virtuais em um maior número de nós físicos, a taxa de aceitação aumenta, assim como a dificuldade do problema. Quando a resiliência contra falha em enlace único é considerada, o problema se torna mais complexo, com menores taxas de aceitação, devido ao alto uso dos recursos físicos. Restrições de atraso não mostraram impacto significativo nas soluções. Além disso, quando as restrições são adicionadas em combinação, a taxa de aceitação geralmente diminui, indicando que é difícil atendê-las simultaneamente. A combinação “L+D+R” levou a menores taxas de aceitação em todas as instâncias. Em relação aos diferentes objetivos, o principal impacto ocorre no uso dos recursos da rede. As soluções com balanceamento de carga espalham as alocações pela rede, levando a um número maior de componentes em uso, mas menores taxas de utilização. Soluções que minimizam o atraso concentram as alocações em recursos próximos, o que implica em uma maior taxa de utilização. Por fim, as soluções com minimização do custo de mapeamento parecem ter uma utilização intermediária entre as outras duas.

### 5.3 Modelos baseados em Geração de Colunas

Para os testes com os modelos estendidos de geração de colunas, variou-se o número de redes virtuais<sup>1</sup>. Além disso, considera-se a inclusão da restrição de localização geográfica em alguns casos.

As Tabelas 5.4 e 5.5 apresentam os resultados do modelo compacto (VNE\_PLI\_COST) resolvido por um algoritmo *branch-and-bound* e dos algoritmos de geração de colunas para os modelos VNE\_GC\_CONFIG e VNE\_GC\_PATH. Para o modelo compacto, são apresentados o custo e o tempo de resolução para sua relaxação linear e para o problema inteiro. Para os algoritmos de geração de colunas, são apresentados o custo da função objetivo, *gap*, número de colunas geradas (Cols) e os tempos gastos no problema mestre (PM), subproblema (SP) e tempo total. As colunas ‘GAP (%)’ representam o *gap* entre a melhor solução inteira encontrada pelo algoritmo *branch-and-bound* no limite de tempo e o valor da relaxação linear.

<sup>1</sup> $|V| \in \{5, 10, 15, 20, 25, 30, 35\}$  para testes sem restrições de localização geográfica; e  $|V| \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 225, 250\}$ , caso contrário.



V	MODELO COMPACTO					VNE_GC_CONFIG						VNE_GC_PATH					
	Melhor Inteira		Relaxação Linear			Objetivo	Cols	GAP (%)	Tempo (seg.)			Objetivo	Cols	GAP (%)	Tempo (seg.)		
	Objetivo	Tempo (seg.)	Objetivo	GAP (%)	Tempo (seg.)				PM	SP	Total				PM	SP	Total
5	664,64	15,36	0,00	100,00	0,01	636,03	20	4,30	0,01	1,99	2,00	468,26	90	29,55	0,00	0,63	0,63
10	1643,72	354,75	0,00	100,00	0,03	1541,10	58	6,24	0,01	32,89	32,90	1441,81	389	12,28	0,10	4,03	4,13
15	2606,34	3602,31	0,00	100,00	0,03	2439,77	125	6,39	0,01	57,65	57,66	2359,52	797	9,47	0,16	6,38	6,54
20	3995,50	3600,03	0,00	100,00	0,05	3819,26	213	4,41	0,01	111,74	111,75	3747,27	1517	6,21	0,53	10,78	11,31
25	4686,58	3600,48	0,00	100,00	0,07	4500,49	343	3,97	0,04	166,02	166,06	4485,90	1726	4,28	0,54	12,04	12,58
30	7068,34	3604,64	0,00	100,00	0,08	6552,40	471	7,30	0,02	240,24	240,26	6504,33	2833	7,98	1,99	22,65	24,64
35	8919,20	3600,06	0,00	100,00	0,11	7720,47	575	13,44	0,02	296,73	296,75	7713,71	3321	13,52	2,92	29,95	32,87

Tabela 5.4: Comparação de resultados entre o modelo compacto e os algoritmos de geração de colunas, sem uso da restrição de localização geográfica

V	MODELO COMPACTO					VNE_GC_CONFIG						VNE_GC_PATH					
	Melhor Inteira		Relaxação Linear			Objetivo	Cols	GAP (%)	Tempo (seg.)			Objetivo	Cols	GAP (%)	Tempo (seg.)		
	Objetivo	Tempo (seg.)	Objetivo	GAP (%)	Tempo (seg.)				PM	SP	Total				PM	SP	Total
10	7873,80	0,20	7354,00	6,60	0,02	7801,32	36	0,92	0,01	0,22	0,23	7801,31	128	0,92	0,00	0,42	0,42
20	15535,74	1,35	13959,67	10,14	0,05	15164,85	104	2,39	0,01	0,68	0,69	15164,84	384	2,39	0,02	1,08	1,10
30	25763,06	21,64	23114,84	10,28	0,11	25069,90	224	2,69	0,01	1,41	1,42	25069,90	723	2,69	0,07	1,69	1,76
40	40394,87	3600,05	36393,08	9,91	0,39	39165,15	350	3,04	0,03	3,86	3,89	39122,90	1402	3,15	0,28	3,06	3,34
50	78069,87	3606,38	59862,98	23,32	1,71	64898,07	539	16,87	0,08	5,47	5,55	64857,11	2147	16,92	0,92	4,18	5,10
60	119019,47	3600,10	80940,20	31,99	4,46	87219,75	725	26,72	0,15	8,16	8,31	87194,74	2928	26,74	1,41	5,49	6,90
70	171017,21	3600,06	130361,78	23,77	4,30	136411,86	946	20,24	0,17	9,85	10,02	136353,83	3532	20,27	1,74	6,56	8,30
80	245449,99	3600,06	182965,31	25,46	5,34	189429,01	1261	22,82	0,25	9,94	10,19	189345,36	4294	22,86	2,33	8,03	10,36
90	295889,11	3600,07	227833,34	23,00	5,72	233804,54	1585	20,98	0,28	11,35	11,63	233692,96	4824	21,02	2,80	8,88	11,68
100	346772,78	3600,08	290828,95	16,13	6,90	296065,98	1968	14,62	0,24	12,02	12,26	295937,35	5056	14,66	2,94	9,08	12,02

Tabela 5.5: Comparação de resultados entre o modelo compacto e os algoritmos de geração de colunas, com restrições de localização geográfica

Em ambas as tabelas, verifica-se que os modelos estendidos apresentam limites melhores que a relaxação linear do modelo compacto. Neste ponto, deve-se dar uma maior atenção à relaxação linear quando desconsidera-se restrição de localização geográfica (Tabela 5.4). O fato de todas instâncias apresentarem custo 0 (zero) se dá pela restrição de conservação de fluxo. Quando dois nós virtuais podem ser mapeados nos mesmos nós físicos, na relaxação linear do problema, o lado direito da equação (3.6) se torna 0 (zero), não obrigando a existência de caminhos físicos para tal demanda. Isso não ocorre nos modelos estendidos, pois nesses modelos, essa restrição é considerada no subproblema, logo, é resolvida de forma inteira. Deve-se notar que o algoritmo *branch-and-bound* atinge o limite de tempo em para várias instâncias, não garantindo a otimalidade das soluções. Além disso, observa-se que os limites do modelo VNE\_GC\_CONFIG são superiores que aqueles do modelo VNE\_GC\_PATH.

É possível observar também, que os tempos gastos para resolver as relaxações lineares dos modelos através dos algoritmos de geração de colunas, concentram-se no subproblema, mostrando uma grande dificuldade em resolvê-lo. Os tempos na Tabela (5.4) são maiores devido à restrição localização não ser considerada. É importante notar que o modelo VNE\_GC\_CONFIG apresenta tempos de resolução elevados.

## 5.4 Algoritmos Branch-and-Price

A seguir são apresentados os resultados obtidos através dos algoritmos *branch-and-price* propostos. Tais resultados são comparados com um algoritmo *branch-and-bound* que resolve o modelo VNE\_PLI\_COST.

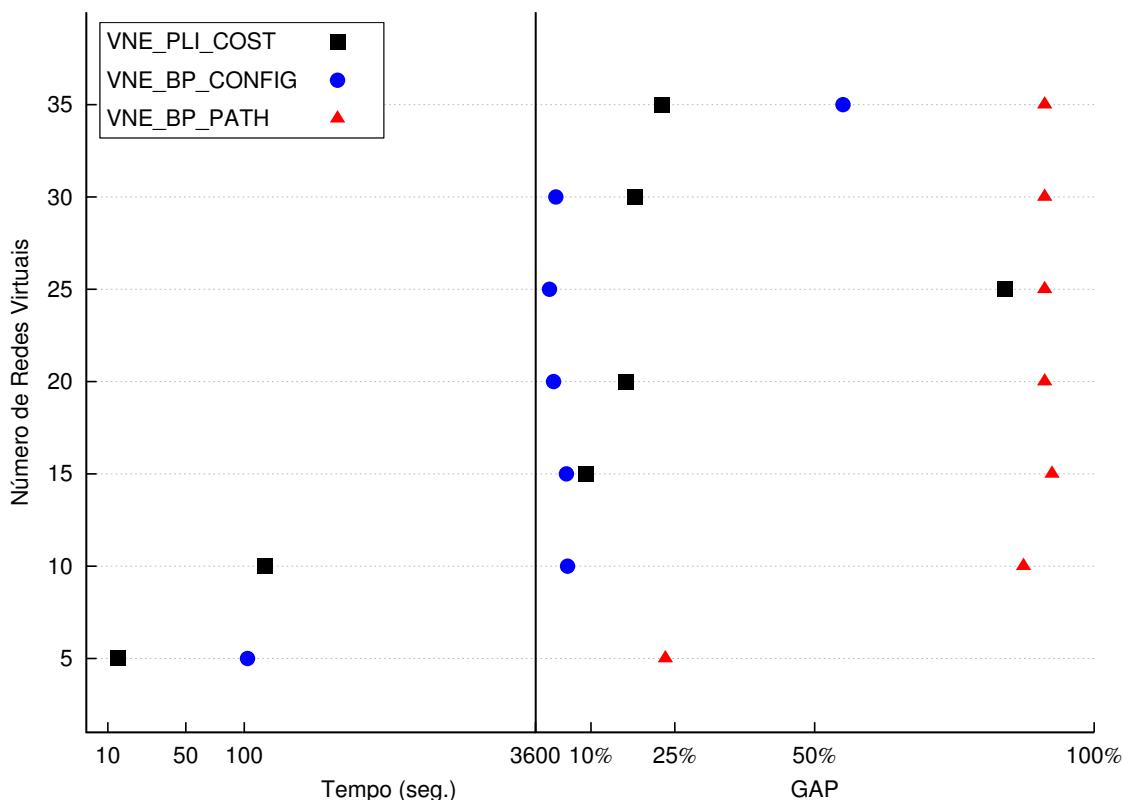


Figura 5.1: Comparação de resultados entre o modelo compacto e algoritmos *branch-and-price* sem o uso de restrições de localização geográfica

Figuras 5.1 e 5.2 e Tabelas 5.6 e 5.7 apresentam os resultados dos testes utilizando o modelo compacto e os algoritmos *branch-and-price* propostos. Nas figuras, ‘VNE\_PLI\_COST’ se refere ao modelo VNE\_PLI\_COST resolvido por *branch-and-bound*, enquanto ‘VNE\_BP\_CONFIG’ e ‘VNE\_BP\_PATH’ se referem aos algoritmos *branch-and-price* propostos de geração de configurações e de caminhos, respectivamente. Os valores sublinhados na coluna ‘GAP (%)’ indicam o método que obteve a solução com menor *gap* dentro do limite de tempo.

V	MODELO COMPACTO			VNE_BP_CONFIG				VNE_BP_PATH			
	Objetivo	Tempo (seg.)	GAP (%)	Melhor Limite	Melhor Inteira	Tempo (seg.)	GAP (%)	Melhor Limite	Melhor Inteira	Tempo (seg.)	GAP (%)
5	664,64	15,36	0,00	664,64	664,64	139,52	0,00	511,00	666,29	3601,69	23,30
10	1643,72	354,75	0,00	1561,02	1657,07	3602,62	5,80	1418,47	11395,70	3601,73	87,55
15	2606,34	3600,00	9,10	2447,00	2592,08	3600,86	5,60	2359,52	31354,22	3607,43	92,47
20	3995,50	3600,00	16,34	3823,33	3953,99	3612,23	3,30	3747,27	42315,14	3610,46	91,14
25	4686,58	3600,00	84,09	4500,50	4619,37	3638,61	2,57	4485,90	42810,45	3600,66	89,52
30	7068,34	3600,00	17,86	6552,50	6803,94	3610,47	3,70	6504,33	63656,67	3622,15	89,78
35	8919,20	3600,00	22,78	7720,47	17183,87	3628,57	55,07	7713,71	74006,43	3613,38	89,57

Tabela 5.6: Resultados sumarizados para o modelo compacto e algoritmos *branch-and-price* sem uso de restrições de localização geográfica

A Figura 5.1 e a Tabela 5.6 apresentam os resultados para os testes desconsiderando restrições de localização geográfica, isto é, os nós virtuais podem ser alocados a quaisquer nós da rede física. O gráfico apresenta o tempo que os algoritmos demandaram para resolver as instâncias com diferentes números de redes virtuais. Para instâncias onde o tempo limite é atingido, é apresentado o *gap* de otimalidade. Observa-se que o algoritmo VNE\_PLI\_COST é capaz de obter a solução ótima apenas para as instâncias com 5 e 10 redes virtuais, enquanto o algoritmo VNE\_BP\_CONFIG resolve apenas a instância com 5 redes virtuais na otimalidade. Contudo, para instâncias com 15 a 30 redes virtuais, o algoritmo VNE\_PLI\_CONFIG obteve soluções melhores, isto é, apresentam menores *gaps*. Já o algoritmo VNE\_BP\_PATH não foi capaz de resolver nenhuma das instâncias sem restrições de localização dentro do tempo limite, apresentando soluções com *gap* de otimalidade muito altos.

Os resultados dos testes com restrição de localização geográfica são apresentados na Tabela 5.7 e na Figura 5.2. Observa-se no gráfico da Figura 5.2, que o modelo VNE\_PLI\_COST, resolvido através de um algoritmo *branch-and-bound*, consegue obter soluções ótimas para as menores instâncias em tempo hábil. Porém, conforme o número de redes virtuais cresce, o problema se torna mais difícil e o algoritmo atinge o tempo limite, retornando a melhor solução encontrada. Isso pode ser visto nas instâncias a partir de 40 redes virtuais para o algoritmo *branch-and-bound* e 20 redes virtuais para os algoritmos VNE\_BP\_CONFIG e VNE\_BP\_PATH, onde as soluções obtidas apresentam *gaps*. Nas instâncias com até 70 redes virtuais, o modelo compacto, mesmo atingindo o tempo limite, consegue alcançar soluções melhores que os algoritmos VNE\_BP. A partir desse valor, os algoritmos VNE\_BP possuem desempenhos melhores que o algoritmo *branch-and-bound*. Os algoritmos VNE\_BP apresentam valores próximos, embora o algoritmo VNE\_BP\_CONFIG, para a maioria das instâncias, valores ligeiramente melhores. Esses resultados encontram-se sumarizados na Tabela 5.7.

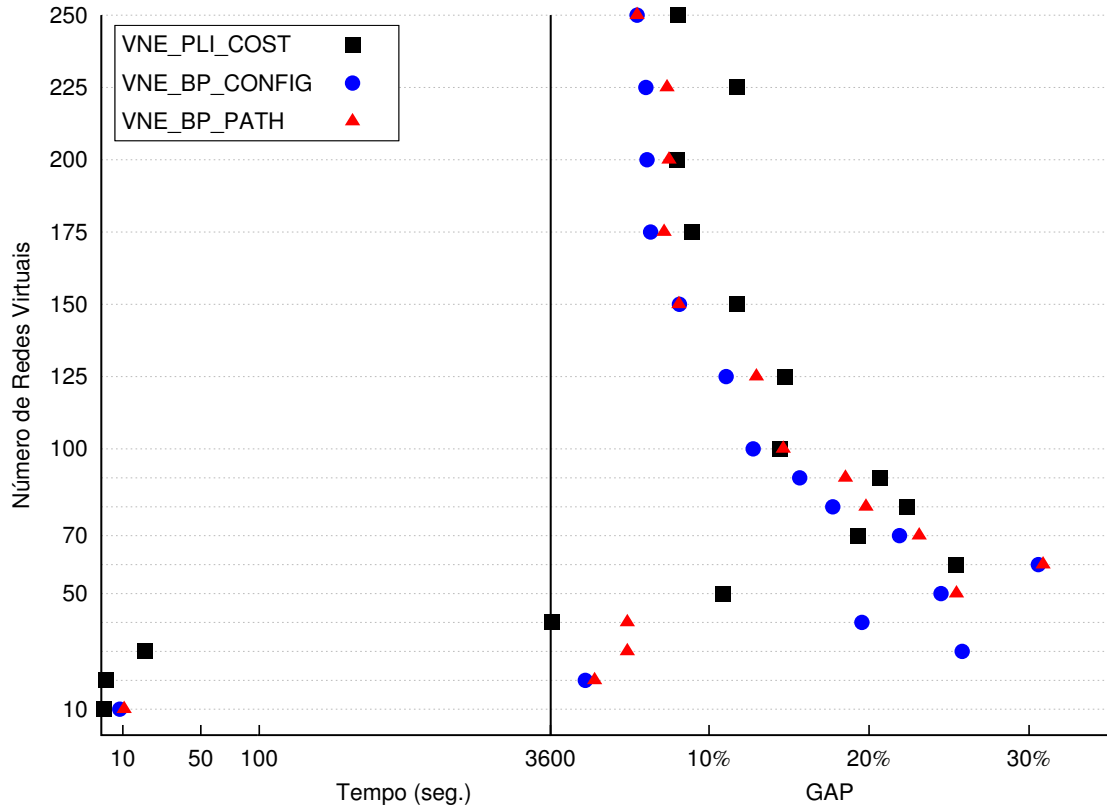


Figura 5.2: Comparação de resultados entre o modelo compacto e algoritmos *branch-and-price* considerando restrições de localização geográfica

V	MODELO COMPACTO			VNE_BP_CONFIG				VNE_BP_PATH			
	Objetivo	Tempo (seg.)	GAP (%)	Melhor Limite	Melhor Inteira	Tempo (seg.)	GAP (%)	Melhor Limite	Melhor Inteira	Tempo (seg.)	GAP (%)
10	7873,80	0,20	0,00	7873,80	7873,80	8,46	0,00	7873,80	7873,80	12,85	0,00
20	15535,74	1,35	0,00	15270,00	15625,34	3600,02	2,27	15212,59	15660,13	3600,39	2,85
30	25763,06	21,64	0,00	25099,08	33831,93	3600,64	25,81	25082,27	26373,66	3602,03	4,89
40	40394,87	3600,05	0,17	39182,93	48705,18	3622,40	19,55	39173,18	41510,85	3625,61	5,63
50	78069,87	3606,38	10,88	69148,02	91578,03	3628,86	24,49	68705,67	92176,93	3613,38	25,46
60	119019,47	3600,10	25,40	87540,29	126104,83	3611,95	30,58	87518,05	126606,25	3603,39	30,87
70	171017,21	3600,06	19,31	138128,94	176860,26	3600,80	21,90	137261,02	178583,94	3620,08	23,13
80	245449,99	3600,06	22,35	190300,68	231315,93	3615,48	17,73	190106,63	237062,98	3613,81	19,80
90	295889,11	3600,07	20,71	234266,59	277755,24	3603,43	15,66	234625,04	287980,47	3605,61	18,52
100	346772,78	3600,08	14,42	296674,46	340016,11	3606,94	12,75	296623,11	347525,80	3607,07	14,64
125	526079,00	3600,00	14,74	447519,59	503250,51	3622,08	11,07	447579,74	514255,88	3615,28	12,96
150	742594,00	3600,00	11,74	653955,81	711990,50	3613,21	8,15	654253,13	712053,47	3602,30	8,12
175	925306,00	3600,00	8,93	841671,19	898741,95	3615,07	6,35	841848,64	907160,78	3602,57	7,19
200	1122780,00	3600,00	8,01	1032680,33	1099998,11	3600,12	6,12	1033090,61	1116797,71	3629,70	7,49
225	1405770,00	3600,10	11,72	1240813,65	1320706,58	3618,15	6,05	1241190,33	1340066,00	3621,91	7,37
250	1585730,00	3600,00	8,07	1457621,35	1542440,57	3604,27	5,50	1457804,45	1542672,11	3605,09	5,50

Tabela 5.7: Resultados sumarizados dos testes com o modelo compacto e algoritmos *branch-and-price* com restrições de localização geográfica

Para a maioria das instâncias, observa-se que os algoritmos atingem o limite de tempo estabelecido. Em geral, o algoritmo VNE\_BP\_CONFIG apresentou resultados superiores aos demais, especialmente para instâncias com maior número de redes virtuais. Quando não se considera localização geográfica, o algoritmo VNE\_BP\_PATH,

embora apresente limites próximos do algoritmo VNE\_BP\_CONFIG não é capaz de encontrar soluções inteiras boas.

# Capítulo 6

## Considerações Finais

### 6.1 Conclusões

Neste trabalho, o problema do Mapeamento de Redes Virtuais no Substrato Físico foi estudado em sua abordagem *offline*.

Foram propostos dois modelos estendidos de geração de colunas, capazes de prover limites duais bem mais fortes que o modelo compacto apresentado. Embora os algoritmos *branch-and-price* tenham atingido o limite de tempo imposto para quase todas as instâncias, os dois algoritmos foram capazes de obter soluções melhores, em termos de gap, que o modelo compacto para instâncias maiores do problema quando a restrição de localização geográfica era considerada. A utilização da heurística primal insere limitações no método, tanto em termos de tempo computacional, pois o número de nós da árvore que serão explorados é reduzido, quanto na qualidade das soluções geradas.

Foi avaliado o impacto de restrições adicionais ao VNE sobre diferentes objetivos através de um modelo de Programação Linear Inteira. Restrições de resiliência, localização geográfica e de atraso foram testadas juntamente com três objetivos. VNE\_PLI\_COST minimiza o custo total da alocação, VNE\_PLI\_LB minimiza a utilização máxima dos enlaces, gerando soluções com balanceamento de carga, enquanto VNE\_PLI\_DL minimiza o atraso máximo na comunicação das demandas. As restrições extras foram adicionadas ao modelo individualmente e em combinações, caracterizando como as soluções são afetadas em termos de taxa de aceitação, uso dos componentes físicos e tempo computacional.

Resultados computacionais mostraram que restrições de localização geográfica tornam o problema mais fácil de ser resolvido, enquanto restrições de resiliência contra falha em enlace único, o tornam mais complexo, com menores taxas de aceitação.

Restrições de atraso não mostraram impacto significativo nas soluções. Em relação aos diferentes objetivos, o principal impacto ocorre no uso dos recursos da rede. As soluções com balanceamento de carga espalham as alocações pela rede, levando a um número maior de componentes em uso, mas menores taxas de utilização. Soluções que minimizam o atraso concentram as alocações em recursos próximos, o que implica em uma maior taxa de utilização.

## 6.2 Trabalhos Futuros

Como trabalhos futuros, a versão *online* do problema pode ser considerada, tratando as requisições assim que chegam. Além disso, a reconfiguração das VNs já alocadas deve ser considerada de forma a prover menor fragmentação no substrato. A identificação de componentes críticas pode permitir um redimensionamento da rede em lugares estratégicos com o objetivo de aumentar as taxas de aceitação.

A utilização de meta-heurísticas para obter uma solução inicial pode acelerar os algoritmos VNE\_BP\_PATH e VNE\_BP\_CONFIG, iniciando-os a partir de soluções melhores. É possível desenvolver heurísticas para geração de limites primais; porém, estas devem ser capazes de utilizar as decisões de *branching* para resolver cada nó da árvore de *branch-and-price* e prover melhores limites superiores em um pequeno tempo, permitindo a exploração de mais nós. Devido ao tempo gasto pelo algoritmo de geração de colunas estar concentrado no subproblema, o que indica a dificuldade em resolvê-lo, a utilização de outras estratégias para resolver o subproblema pode acelerar o processo de convergência do método. Para o modelo VNE\_BP\_PATH, é possível utilizar o algoritmo de caminho mínimo Floyd-Warshall no subproblema. Novamente, deve-se garantir que tais soluções sejam capazes de respeitar as decisões de *branching* tomadas até o momento.

Nos modelos utilizados no presente trabalho, foram utilizadas penalidades para que seja atendido o maior número de redes virtuais possível. Uma possibilidade de trabalho futuro inclui tratar tal penalidade como restrição. Outra abordagem a ser estudada engloba a utilização de modelos multi-objetivo, uma vez que ao mesmo tempo que busca-se pelo atendimento do maior número de redes virtuais, é desejado que o mapeamento das mesmas seja feito com o menor custo.

# Referências Bibliográficas

- Amaldi, E.; Coniglio, S.; Koster, A. M. & Tieves, M. (2016). On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213 – 220. ISSN 1571-0653. INOC 2015 – 7th International Network Optimization Conference.
- Andersen, D. G. (2002). Theoretical approaches to node assignment. Available at: <http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps>. Unpublished Manuscript.
- Araújo, S. M. A. & Souza, F. S. H. d. (2015). Uma heurística grasp-rvns para o problema de mapeamento de redes virtuais. *Anais do XLVII SBPO*, pp. 2138–2149.
- Barnhart, C.; Johnson, E. L.; Nemhauser, G. L.; Savelsbergh, M. W. P. & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316--329. ISSN 0030-364X.
- Chowdhury, N. M. K. & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 54(5):862--876. ISSN 1389-1286.
- Chowdhury, N. M. M. K.; Rahman, M. R. & Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. Em *INFOCOM*, pp. 783–791. IEEE.
- Doig, A. G.; Land, B. H. & Doig, A. G. (1960). An automatic method for solving discrete programming problems. *Econometrica*, pp. 497--520.
- Fajjari, I.; Aitsaadi, N.; Dab, B. & Pujolle, G. (2016). Novel adaptive virtual network embedding algorithm for cloud's private backbone network. *Computer Communications*, 84(C):12--24. ISSN 0140-3664.



- Fajjari, I.; Aitsaadi, N.; Pujolle, G. & Zimmermann, H. (2011). VNR algorithm: A greedy approach for virtual networks reconfigurations. Em *GLOBECOM*, pp. 1–6. IEEE.
- Fischer, A.; Botero, J.; Till Beck, M.; de Meer, H. & Hesselbach, X. (2013). Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906. ISSN 1553-877X.
- George B. Dantzig, P. W. (1960). Decomposition principle for linear programs. *Operations Research*, 8(1):101–111. ISSN 0030364X, 15265463.
- Giusto, D.; Iera, A.; Morabito, G. & Atzori, L. (2010). *The internet of things: 20th Tyrrhenian workshop on digital communications*. Springer Science & Business Media.
- Herker, S.; Khan, A. & An, X. (2013). Survey on survivable virtual network embedding problem and solutions. Em *ICNS 2013, The Ninth International Conference on Networking and Services, ICNS*, pp. 99--104.
- Houidi, I.; Louati, W.; Ameer, W. B. & Zeghlache, D. (2011). Virtual network provisioning across multiple substrate networks. *Computer Networks*, 55(4):1011 – 1023. ISSN 1389-1286. Special Issue on Architectures and Protocols for the Future Internet.
- Houidi, I.; Louati, W. & Zeghlache, D. (2008). A distributed virtual network mapping algorithm. Em *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 5634–5640.
- Iannaccone, G.; Chuah, C.-n.; Mortier, R.; Bhattacharyya, S. & Diot, C. (2002). Analysis of link failures in an IP backbone. Em *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, IMW '02*, pp. 237--242, New York, NY, USA. ACM.
- ILOG, I. (2013). CPLEX optimizer. [Online; acessado em 30-Mar-2013].
- Inführ, J. & Raidl, G. R. (2011). Introducing the virtual network mapping problem with delay, routing and location constraints. Em *Proceedings of the 5th International Conference on Network Optimization, INOC'11*, pp. 105--117, Berlin, Heidelberg. Springer-Verlag.
- Lischka, J. & Karl, H. (2009). A virtual network mapping algorithm based on subgraph isomorphism detection. Em *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, VISA '09*, pp. 81--88, New York, NY, USA. ACM.

- Liu, J.; Huang, T.; Chen, J.-y. & Liu, Y.-j. (2011). A new algorithm based on the proximity principle for the virtual network embedding problem. *Journal of Zhejiang University – SCIENCE C*, 12(11):910–918. ISSN 1869-1951.
- Marquezan, C. C.; Granville, L. Z.; Nunzi, G. & Brunner, M. (2010). Distributed autonomic resource management for network virtualization. Em *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pp. 463–470. ISSN 1542-1201.
- Moura, L. F. d. S. (2015). Branch & price for the virtual network embedding problem. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.
- Nogueira, J.; Melo, M.; Carapinha, J. & Sargento, S. (2011). Virtual network mapping into heterogeneous substrate networks. Em *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pp. 438–444. ISSN 1530-1346.
- Nonde, L.; El-Gorashi, T. & Elmirghani, J. (2015). Energy efficient virtual network embedding for cloud networks. *Lightwave Technology, Journal of*, 33(9):1828–1849. ISSN 0733-8724.
- Rahman, M. R. & Boutaba, R. (2013). SVNE: Survivable virtual network embedding algorithms for network virtualization. *IEEE Transactions on Network and Service Management*, 10(2):105–118.
- Shuiqing, G.; Jing, C.; Siyi, Z. & Qingchao, Z. (2016). Virtual network embedding with multi-attribute node ranking based on TOPSIS. *KSII Transactions on Internet & Information Systems*, 10(2):522 – 541. ISSN 19767277.
- Stezenbach, D.; Hartmann, M. & Tutschku, K. (2012). Parameters and challenges for virtual network embedding in the future internet. Em *2012 IEEE Network Operations and Management Symposium*, pp. 1272–1278. ISSN 1542-1201.
- Sun, G.; Yu, H.; Anand, V. & Li, L. (2013). A cost efficient framework and algorithm for embedding dynamic virtual network requests. *Future Generation Computer Systems*, 29(5):1265--1277. ISSN 0167-739X.
- Tran, P. N.; Casucci, L. & Timm-Giel, A. (2012). Optimal mapping of virtual networks considering reactive reconfiguration. Em *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pp. 35–40.

- Turner, J. S. & Taylor, D. E. (2005). Diversifying the internet. Em *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 2, pp. 6 pp.–760. ISSN 1930-529X.
- Vanderbeck, F. (2011). Branching in branch-and-price: A generic scheme. *Math. Program.*, 130(2):249–294. ISSN 0025-5610.
- Wang, Y.; Hu, Q. & Cao, X. (2016). A branch-and-price framework for optimal virtual network embedding. *Computer Networks*, 94(C):318–326. ISSN 1389-1286.
- Wang, Z.; Han, Y.; Lin, T.; Tang, H. & Ci, S. (2012). Virtual network embedding by exploiting topological information. Em *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2603–2608. ISSN 1930-529X.
- Yu, M.; Yi, Y.; Rexford, J. & Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *Computer Communication Review*, 38(2):17–29.
- Zhu, F. & Wang, H. (2016). A modified ACO algorithm for virtual network embedding based on graph decomposition. *Computer Communications*, 80:1 – 15. ISSN 0140-3664.
- Zhu, Y. & Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, pp. 1–12.