

**ABORDAGENS EXATA E HEURÍSTICA PARA O
MAPEAMENTO DE REDES VIRTUAIS**

SAMUEL MOREIRA ABREU ARAÚJO

**ABORDAGENS EXATA E HEURÍSTICA PARA O
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 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
Janeiro de 2018

© 2018, Samuel Moreira Abreu Araújo.
Todos os direitos reservados.

Araújo, Samuel Moreira Abreu

A663a Abordagens exata e heurística para o mapeamento de redes virtuais / Samuel Moreira Abreu Araújo - Belo Horizonte, 2018.
xxiv, 78 f. : il. ; 29cm

Dissertação (mestrado) - Universidade Federal de Minas Gerais - Departamento de Ciência da Computação.

Orientador: Geraldo Robson Mateus
Coorientadora: Fernanda Sumika Hojo de Souza

1. Computação — Teses. 2. Otimização combinatória. 3. Redes de Computadores.
I. Orientador. II. Coorientadora. III. Título.

CDU 519.6*61(043)



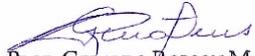
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Abordagens exata e heurística para o mapeamento de redes virtuais

SAMUEL MOREIRA ABREU ARAÚJO

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. DANIEL FERNANDES MACEDO
Departamento de Ciência da Computação - UFMG


PROF. HAROLD GAMBINI SANTOS
Departamento de Ciência da Computação - UFOP

Belo Horizonte, 27 de Fevereiro de 2018.

Dedico este trabalho aos meus familiares e em especial ao meu pai, mãe, irmão e namorada. Pessoas que sempre me deram apoio financeiro e emocional durante todo meu o percurso acadêmico. Aos meus amigos pelos momentos de alegria e que sem sombra de dúvida, possuem grande parcela de tudo que conquistei. Um sincero obrigado a todos!

Agradecimentos

Agradeço à minha mãe Dulce Abreu e meu pai Valdinês Moreira, heróis que me deram apoio nas horas de desânimo e cansaço.

Obrigado a meu irmão David Moreira, que sempre se esforçou em me fazer entender que o futuro é feito a partir da constante dedicação no presente.

Agradeço a minha namorada Joyce Ribeiro, que de forma especial me deu força e coragem nos momentos de dificuldade.

Meus agradecimentos aos amigos, primos, tios e tias e todos os irmãos na amizade que fazem parte da minha formação social e humana, e que vão continuar presentes em minha vida.

A UFMG, pelo ambiente criativo e amigável proporcionado aos alunos. Ao corpo docente, direção e administração que propiciaram a janela através da qual posso hoje vislumbrar um horizonte superior. A UFSJ, que me proporcionou o embasamento teórico para iniciar o mestrado. Ao Colégio Crispim Jaques Bias fortes pela excelente formação ao nível de ensino médio.

Em especial ao meu orientador, Prof. Geraldo Robson Mateus; e a minha coorientadora de mestrado e orientadora de graduação Prof. Fernanda Sumika; agradeço pelos ensinamentos, paciência e principalmente pela confiança no meu esforço e capacidade.

Por fim, agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por propiciar o apoio financeiro, sem o qual seria impossível concluir o mestrado. Agradeço ainda a UFMG por custear as participações em congressos, que, sem dúvida, acrescentaram muito em minha formação intelectual.

“UBUNTU (origem Bantu) - Sou o que sou pelo que nós somos. Não significa que uma pessoa não se preocupe com o seu progresso pessoal. A questão é: o meu progresso pessoal está ao serviço do progresso da minha comunidade? Isso é o mais importante na vida. E se uma pessoa conseguir viver assim, terá atingido algo muito importante e admirável.”

(Nelson Mandela)

Resumo

A virtualização de redes é uma técnica que permite a emulação de várias redes virtuais, de forma que as diferentes redes coexistam simultaneamente sobre a mesma estrutura física de rede. Assim, roteadores e enlaces físicos podem hospedar um conjunto de redes virtuais, desde que não tenham suas capacidades de processamento e banda extrapolados. Tal ação é benéfica por permitir a liberdade da rigidez dos protocolos de rede, facilitando o desenvolvimento de novas tecnologias para a *web*. Nesse propósito, definir qual conjunto de equipamentos físicos irão hospedar uma ou mais redes virtuais é uma tarefa complexa, problema denominado Mapeamento de Redes Virtuais (VNE). O VNE é pertencente à classe dos problemas NP-difíceis, logo, sua resolução exata pode ser inviável para cenários de grande porte. A maioria dos trabalhos existentes na literatura efetua o mapeamento somente sobre um único domínio de rede, este trabalho se distingue por apresentar abordagens de programação linear inteira e heurística para a resolução do VNE em ambientes compostos por um ou mais domínios de rede. Os modelos tratados focam alguns desafios atuais de virtualização de redes e diferenciam-se nos objetivos propostos, sendo: maximizar o balanceamento de carga, minimizar o uso de energia elétrica, maximizar o lucro dos provedores de serviço e minimizar a troca de mensagens entre diferentes domínios. Ao final, são feitas simulações para as abordagens exata e heurística. Resultados experimentais mostram o desempenho de ambas abordagens em cenários *online*, periódico e *offline*, sendo usado o software CPLEX 12.6 no modelo exato. Neste trabalho ainda é proposta a aplicação da abordagem heurística em um ambiente multidomínio e *online*, através do controle de um orquestrador. Um estudo comparativo de três modelos de orquestração distintos é apresentado, com: conhecimento total, parcial e sem conhecimento da infraestrutura de rede física interna aos domínios. Os testes de orquestração abordam diferentes aspectos como consumo de energia, geração de lucro e troca de mensagens.

Palavras-chave: Otimização, Virtualização de rede, Redes multidomínio.

Abstract

Network virtualization is a technique that allows the emulation of multiple virtual networks, so that different networks coexist simultaneously on the same physical network structure. Thereby, routers and physical links can host a set of virtual networks as long as they do not have their processing and bandwidth capabilities extrapolated. Such action is beneficial by allowing the freedom upon of network protocols hardness, facilitating the development of new web technologies. In this purpose, define which physical equipment set will host one or more virtual networks is a complex task, problem as known Virtual Network Embedding (VNE). The VNE belongs to the NP-hard class of problems, so its exact resolution may not be feasible for large scenarios. Most of literature existing works presents the virtualization only in a single network domain, this work is distinguished by presenting an integer linear programming model and a heuristic for the resolution of the VNE in environments managed by one or more network domains. The models deal with present network virtualization challenges and are differentiated in the proposed objectives: maximize the load balancing, minimize the use of energy, maximize the profit of the service providers and minimize the exchange of messages between different domains. At the end of this work, simulations are done for both the exact and heuristic approaches. Experimental results show the performance of these approaches in online, periodic and offline scenarios, using CPLEX 12.6 software in the exact model. In this work it is still proposed the application of the heuristic approach in a multidomain and online environment, through the control of an orchestrator. A comparative study regarding the implementation of three different orchestration models is presented: total knowledge, partial and without knowledge of the internal physical network infrastructure of the domains. Experimental results show the performance of the orchestrator along with the heuristic, reporting energy consumption, profit generation and message exchange metrics for the different approaches.

Keywords: Optimization, Network Virtualization, Multi-domain networks.

Lista de Figuras

1.1	Representação de um ambiente com múltiplos domínios	5
3.1	Exemplo da modelagem do SN com três domínios e uma VN	27
3.2	Representação do mapeamento de duas VNs sobre um domínio	29
3.3	Exemplo de chegada de requisições	33
4.1	Exemplos de ótimo local e global	37
4.2	Representação de vizinhança N^1	42
4.3	Representação de vizinhança N^2	43
4.4	Orquestrador com conhecimento total do SN	44
4.5	Orquestrador com conhecimento parcial do SN	45
4.6	Orquestrador sem conhecimento do SN	46
5.1	Número de requisições aceitas nos cenários SN_4R2 e SN_4R3	56
5.2	Tempo total de execução nos cenários SN_4R2 e SN_4R3	57
5.3	Número de requisições aceitas no cenário $SN1R1$	58
5.4	Número de requisições aceitas no cenário $SN2R1$	59
5.5	Número de requisições aceitas no cenário $SN3R1$	59
5.6	Lucro total gerado no cenário $SN1R1$	60
5.7	Lucro total gerado no cenário $SN2R1$	60
5.8	Lucro total gerado no cenário $SN3R1$	61
5.9	Atraso médio de atendimento no cenário $SN1R1$	62
5.10	Atraso médio de atendimento no cenário $SN2R1$	62
5.11	Atraso médio de atendimento no cenário $SN3R1$	63
5.12	Tempo total de execução no cenário $SN1R1$	64
5.13	Tempo total de execução no cenário $SN2R1$	64
5.14	Tempo total de execução no cenário $SN3R1$	65
5.15	Número de requisições aceitas nos cenários $SN1R2$ e $SN1R3$	66
5.16	Lucro final acumulado nos cenários $SN1R2$ e $SN1R3$	66

5.17	Gasto de energia total nos cenários <i>SN1R2</i> e <i>SN1R3</i>	67
5.18	Total de mensagens trocadas nos cenários <i>SN1R2</i> e <i>SN1R3</i>	67

Lista de Tabelas

2.1	Classificação dos trabalhos relacionados	23
4.1	Tabela de mensagens de coordenação	46
5.1	Parâmetros para geração da rede física	51
5.2	Parâmetros para geração das requisições	52
5.3	Parâmetros inter-domínio	53
5.4	Parâmetros intra-domínio	53
5.5	Quantidade de restrições e variáveis inteiras para o cenário <i>SN1R1</i>	64

Lista de Algoritmos

1	GRASPF-RVNS ($V[\]$, α)	38
2	Construtivo (G^v , α)	39
3	RVNS(s_0)	41

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
1 Introdução	1
1.1 Contextualização	1
1.2 Virtualização de Redes	5
1.2.1 Taxonomia	6
1.2.2 Aplicações	8
1.2.3 Complexidade	10
1.3 Motivação	10
1.4 Objetivos	12
1.4.1 Geral	12
1.4.2 Específicos	12
1.5 Contribuições	13
1.6 Organização	14
2 Trabalhos Relacionados	15
2.1 Abordagens <i>Offline</i> , Periódica e <i>Online</i>	15
2.2 Abordagens Estáticas e Dinâmicas	16
2.3 Abordagens Concisas e Redundantes	18
2.4 Abordagens Centralizadas e Distribuídas	18
2.5 Abordagens com diferentes objetivos e outros aspectos	21

3	Mapeamento de Redes Virtuais e Formulações	25
3.1	Definição Formal do Problema	25
3.2	Formulações Propostas	27
3.2.1	Maximizar o balanceamento de carga	29
3.2.2	Minimizar consumo de energia elétrica	30
3.2.3	Maximizar o lucro dos domínios	32
3.2.4	Minimizar a comunicação entre domínios	32
3.3	Abordagens <i>Online</i> , Periódica e <i>Offline</i>	33
4	Abordagem Heurística e Orquestração de Redes	35
4.1	Heurística GRASPF-RVNS	35
4.1.1	Heurística Construtiva	38
4.1.2	Heurística de Refinamento	40
4.1.3	Estruturas de Vizinhança	41
4.2	Orquestração para o VNEMD utilizando a abordagem heurística	43
4.2.1	Orquestrador com Conhecimento Total do <i>SN</i>	43
4.2.2	Orquestrador com Conhecimento Parcial do <i>SN</i>	44
4.2.3	Orquestrador sem Conhecimento do <i>SN</i>	45
4.2.4	Troca de Mensagens de Coordenação	46
5	Experimentos Computacionais	49
5.1	Métricas de Avaliação	49
5.2	O simulador	50
5.3	As instâncias	51
5.4	Abordagens Avaliadas	53
5.5	Análise de Desempenho	55
5.5.1	Análises preliminares em cenário SD e <i>online</i>	55
5.5.2	Análises <i>offline</i> , <i>online</i> e Periódica	57
5.5.3	Análise do Orquestrador	65
6	Considerações Finais	69
6.1	Conclusão	69
6.2	Trabalhos Futuros	71
	Referências Bibliográficas	73

Capítulo 1

Introdução

Neste capítulo, inicialmente é feita uma breve introdução ao mapeamento de redes virtuais, onde é apresentada uma contextualização do tema junto aos objetivos abordados, bem como a contribuição para o desenvolvimento da área. Após são discutidos os principais conceitos e definições sobre a virtualização de redes, tal como um entendimento de sua aplicação prática.

1.1 Contextualização

A evolução da computação é fruto da necessidade do ser humano em facilitar tarefas cotidianas aliada a uma nova demanda de serviços [Farah, 2004]. Em uma visão proposta por Lule [2016], a internet evoluiu muito desde sua concepção, não só em relação a infraestrutura, mas também quanto a sua utilidade. Nesse aspecto, a internet hoje é vista como um pilar da sociedade moderna, capaz de centralizar e prover facilmente ao usuário informações sobre diferentes questões como cultura, sociedade, política etc.

A corporação *Global e-Sustainability Initiative* (GeSI) atua com a colaboração de empresas e organizações de tecnologias da informação e comunicação (TICs) de todo o mundo, de forma a gerar informações sobre recursos e práticas que buscam alcançar a sustentabilidade social e ambiental integrada através das TICs. Em um paralelo junto a evolução e aceitação da internet, um painel traçado por GeSI destaca o grande aumento da disponibilidade de banda larga global. Em 2002, existiam no mundo cerca de 67 milhões de roteadores; estima-se que tal número deva subir para 898 milhões em 2020. Ainda em 2002, existiam 1,1 bilhões de contas virtuais de aplicações móveis; tal panorama tende a crescer para um número de 4,8 bilhões em 2020 [The Climate Group, 2008].

O crescimento da internet é fortemente impulsionado por uma nova gama de aplicações e dados portáteis, que podem ser acessados de qualquer ambiente de maneira *online*. Estas aplicações estão em um processo de migração do modelo chamado *on-premise* para o modelo chamado de computação em nuvem (*cloud computing*) [Pham et al., 2011]. No modelo *on-premise* armazena-se os arquivos e dados em um único lugar o que deixa o acesso restrito a um único ambiente. Diferentemente, o modelo de computação em nuvem usa servidores compartilhados com a internet, o que permite o acesso remoto aos arquivos e dados com liberdade e flexibilidade de acesso ao usuário.

Existe ainda à consolidação da chamada internet das coisas (*Internet of Things* - IoT), que busca conectar aparelhos eletrônicos do dia a dia à internet e trazer mais dinâmica e integração ao cotidiano das pessoas [Bojanova & Voas, 2017]. Nesse segmento, muitos aplicativos, arquivos e dados não precisam necessariamente estar armazenados no computador do usuário, e sim em um ambiente chamado nuvem de dados. Entretanto essa liberdade tem um preço, a dependência intrínseca de uma internet rápida e estável.

O notado crescimento da internet esbarra em fatores importantes e alarmantes inerentes ao fato da internet ser desenvolvida sobre protocolos rígidos e limitados. A internet foi criada em 1969 e em meados dos anos 80 já adotava o protocolo *IPv4*. Atualmente a internet ainda é fundamentada no protocolo *IPv4*, que é seguro e confiável, mas, em contrapartida, ao ser desenvolvido não previa a grande quantidade de conexões existentes nos tempos atuais. O campo de endereçamento do *IPv4* torna-se pequeno dado a grande quantidade de endereçamentos que são necessários, visto que cada computador precisa um endereçamento único para estabelecer uma conexão. Outro problema é que existe um grande número de diferentes aplicações operando sobre a internet, onde requerimentos como segurança, escalabilidade, flexibilidade, mobilidade e isolamento, podem ser demandados por determinada aplicação e não foram previstos na criação do protocolo. Dessa forma, um grande desafio começa a se desenhar: a demanda por *links* com segurança e disponibilidade, e que atenda os diferentes requisitos demandados pelas aplicações. Tal problema, misturado à rigidez dos protocolos que não permitem alterações em seu núcleo, é chamado de "ossificação da internet" [Turner & Taylor, 2005; Zhu & Ammar, 2006].

Uma alternativa inicial para o problema de falta de endereços físicos é a adoção do protocolo *IPv6* que aumenta o campo de endereçamento, antes de *32bits* no *IPv4*, para *128bits*. Tal adoção não é simples e implica a troca de *hardware* que inicialmente oferecia compatibilidade somente com o *IPv4* por um *hardware* híbrido que ofereça suporte ao *IPv6*. Tal migração possui um custo financeiro potencialmente alto, pois implica na substituição de vários milhões de dispositivos legados ao *IPv4*. A implan-

tação global do *IPv6* está sendo feita desde 1999 de forma lenta e gradual. Em 2017, segundo dados da *Asia-Pacific Network Information Centre* (APNIC)¹, países como os Estados Unidos possuem uma taxa de tráfego de dados de 44% sobre o protocolo *IPv6*, mas por outro lado a maioria dos países da Ásia, Europa, África possuem menos de 2% de adoção do protocolo *IPv6*.

Uma alternativa rápida e de baixo custo para o problema da ossificação da internet, é a aplicação de mecanismos de virtualização de redes, que permitem a implementação de novas funcionalidades ao se criar uma visão lógica do *hardware*, de forma que múltiplas redes virtuais com características particulares possam compartilhar o mesmo substrato físico. Neste contexto, surge o problema do mapeamento de redes virtuais (ou em inglês *Virtual Network Embedding* - VNE). O VNE consiste em determinar o mapeamento de requisições de redes virtuais (VNs), compostas por roteadores e enlaces virtuais aninhados sobre roteadores e enlaces físicos que compõem a rede física (ou substrato de rede - SN)[Chowdhury et al., 2009; Fischer et al., 2013].

O problema de se mapear uma ou mais VNs, com demandas de processamento e banda não nulas sobre um SN com capacidades limitadas, é tido como pertencente à classe dos problemas NP-difíceis mesmo no cenário *offline*, no qual todas as VNs são conhecidas *a priori* [Zhu & Ammar, 2006; Houidi et al., 2008; Amaldi et al., 2016]. No cenário *online*, as demandas de uma VN e seu tempo de vida somente passam a ser conhecidos no momento de sua chegada ao servidor, o que dificulta uma melhor atuação do algoritmo de mapeamento por desconhecer demandas futuras.

De acordo com Chowdhury & Boutaba [2009], no VNE existem diferentes entidades envolvidas que possuem ações distintas: os provedores de infraestrutura (ISPs), responsáveis por gerir os componentes do SN ou domínios; os provedores de serviço (SPs), responsáveis por instanciar e gerir as VNs, além de contratar os serviços dos ISPs; e os usuários finais, que consomem os serviços prestados pelos SPs. Em nosso trabalho, um domínio de rede é definido como um conjunto de equipamentos de infraestrutura de redes gerenciados por um ISP, podendo ser composto por *hubs*, *switches*, roteadores, fibra ótica, etc. Cada ISP envolvido pode gerir um ou mais domínios de rede e cada domínio pode possuir políticas de gerência de recursos independentes.

Outra questão relativa ao VNE consiste na composição do SN que pode ser formado por um único domínio (SD) ou pela união de vários domínios diferentes (multi-domínios - MD). Em ambientes MD se torna estratégica a questão do mapeamento de redes virtuais de forma descentralizada, o que gera uma maior autonomia nas tomadas de decisão de cada domínio. Com o mapeamento descentralizado de VNs, cada domí-

¹Dados retirados de <https://stats.labs.apnic.net/ipv6>, acessado em: 28/02/2018.

nio pode aplicar um algoritmo de mapeamento próprio de forma independente a fim de maximizar seus interesses. Para tanto, cada domínio tem independência total de otimização local. Outro ponto é que os mapeamentos em multidomínios podem ser executados em paralelo, de forma distribuída, o que caracteriza a migração do paradigma de mapeamento centralizado para o descentralizado.

Cada domínio de rede é administrado por uma gerência com princípios administrativos particulares. Em ambientes reais é comum que uma unidade gestora de um domínio não divulgue dados relativos à sua administração, pois são informações pertencentes à inteligência dos processos de negócios da empresa. Dentre essas informações estão topologias, taxa de ocupação, tarifação, consumo de energia, etc. Contudo, esse compartilhamento de informações pode ser de interesse em determinados casos, fazendo com que a empresa a empresa mudar de postura perante argumentos convincentes como o aumento de receita. A premissa é que o compartilhamento de informações entre os domínios envolvidos pode gerar um mapeamento mais inteligente, com um custo menor e com um lucro maior para os domínios envolvidos.

Diversos trabalhos desenvolvidos abordam e simulam a questão do mapeamento SD [Chowdhury et al., 2009; Nguyen et al., 2017], no qual os recursos de rede são regidos por políticas intrínsecas a um único ISP. Em uma abordagem MD, podem estar envolvidos vários ISPs, onde cada ISP administra seus recursos de rede de acordo com políticas internas [Houidi et al., 2011; Samuel et al., 2013]. A Figura 1.1 mostra um conjunto de VNs a serem mapeadas sobre o cenário MD. Um dos grandes desafios existentes é gerar bons mapeamentos tanto em ambientes SD quanto MD. Na Figura 1.1 cada domínio é controlado por um ISP diferente. Um dos desafios inerentes a essa situação é estipular políticas de comunicação entre domínios que respeitem as políticas locais de cada ISP.

Dado o desafio de se prover o mapeamento MD, existe uma série de fatores que devem ser abordados, tais como a troca de mensagens, que se feita de forma descomedida pode gerar um *overhead*² de comunicação, causar gargalos no SN e ser prejudicial em futuros mapeamentos; o consumo de energia, aspecto de importante destaque, que segue uma linha de consciência ambiental e o lucro dos ISPs. Outro ponto importante é a necessidade de um mapeamento rápido, neste caso abordagens exatas e/ou baseadas em programação linear tendem a ser lentas, devido à complexidade do problema junto ao grande volume de componentes presentes em cada domínio.

²Em virtualização, *overhead* refere-se a um consumo extra de banda em um enlace e potenciais atrasos gerados pela troca de mensagens entre domínios diferentes para se prover um mapeamento.

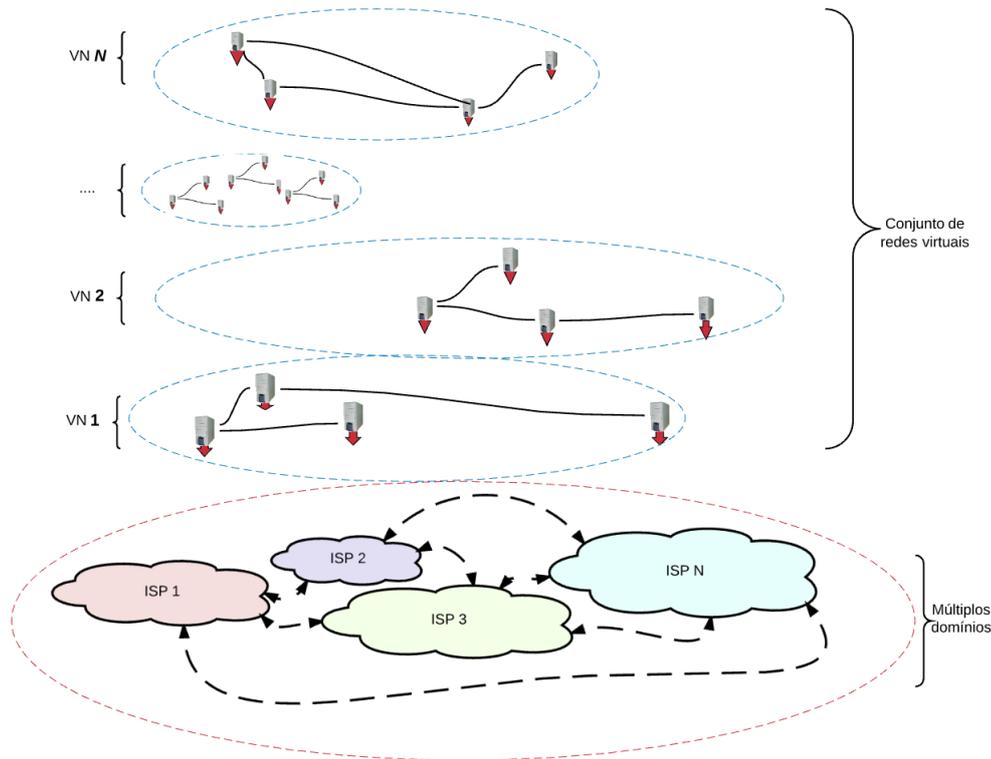


Figura 1.1: Representação do mapeamento de redes virtuais MD. Composta por um conjunto de redes virtuais sobrepostas aninhadas sobre um conjunto de diferentes domínios.

1.2 Virtualização de Redes

Em um paradigma antepassado, se um programa de computador foi criado baseado em uma arquitetura de computador χ , de uma empresa 1, dificilmente ele funcionará em uma máquina τ , de arquitetura diferente, em uma empresa 2. Nesse caso, ou a empresa 2 adquire uma máquina de arquitetura χ ou se deve criar outro programa. Dado esse problema, iniciou-se a décadas a necessidade de simular ambientes virtuais para as aplicações serem portáteis [Goldberg, 1974]. Formalmente à virtualização se deu pela necessidade de emular ambientes isolados e capazes de rodar diferentes *softwares* de maneira independente, mas dentro do mesmo equipamento físico.

Nos dias atuais, a definição de virtualização apresentada por Goldberg [1974] pode ser estendida a um conceito mais amplo, e pode ser aplicada as infraestruturas de redes de computadores. O ponto é que a virtualização de redes permite a separação de fluxos de dados em redes virtuais distintas, e pode criar a ilusão da disponibilidade de recursos físicos específicos a serem explorados pelas aplicações. Este trabalho foca na virtualização de redes de computadores, onde são emulados componentes de

redes virtuais sobre a rede física de maneira que as aplicações possuam a liberdade de implementar as suas funcionalidades de maneira independente.

1.2.1 Taxonomia

O VNE é um problema segmentado em diferentes frentes de estudo e por conseguinte abordado em diversos trabalhos distintos. A fim de utilizar um modelo mais próximo da realidade, alguns aspectos devem ser respeitados como: heterogeneidades de recursos dos substratos físicos, tempo de duração das requisições e o uso de topologias diversificadamente genéricas. De maneira a unificar as várias classificações e terminologias existentes para o VNE, a taxonomia adotada segue o entendimento de Fischer et al. [2013], e pode ser dividida em diferentes abordagens.

- **Estático ou dinâmico**, fator que concerne em manter ou não o mapeamento de uma VN no mesmo local. No mapeamento estático cada VN é mapeada sobre o SN de maneira fixa, neste caso o mapeamento é mantido usando os mesmos recursos até o término do tempo de vida da requisição. Em mapeamentos dinâmicos, mesmo que uma VN já esteja mapeada, o SN pode sofrer reconfigurações e realocar recursos em prol de algum benefício.
- **Centralizado ou distribuído**, característica referente à entidade de mapeamento das VNs. Em trabalhos centralizados tal entidade conhece todo o SN, e faz o mapeamento de forma global. Em trabalhos distribuídos podem existir múltiplas unidades de mapeamento. Conforme Fischer et al. [2013], o modelo centralizado tende a ter um melhor desempenho em termos de otimização da função objetivo que o distribuído. O fato ocorre por, em ambientes distribuídos, ser aplicado o mapeamento sem a informação das unidades de mapeamento adjacentes, o que pode acarretar em perdas no desempenho da rede ou nos valores da função objetivo.
- **Conciso ou redundante**, aspecto relacionado ao provimento de resiliência contra as falhas. Em abordagens redundantes o algoritmo de mapeamento deve prover reservas de recursos e mecanismos de migração para o caso de possíveis falhas nos componentes físicos do SN. Em mapeamentos concisos não se emprega tal de redundância. Pelo entendimento de Oliveira et al. [2015], os mapeamentos redundantes tendem a ter menores taxas de aceitação, onde o ato de se prover resiliência acarreta muitas vezes em recursos ociosos que poderiam ser usados por outras VNs.

- **Online, offline ou periódico**, propriedade que aborda a taxa de chegada das VNs a serem processadas. Cenários *online* são caracterizados pela chegada das VNs de maneira aleatória e desconhecida, onde não se conhece nenhum aspecto sobre a topologia, tempo de vida e requisitos demandados. Em cenários *offline* se aplica o mapeamento em todo o conjunto de VNs existentes de uma vez, neste caso o algoritmo beneficia-se do fato de conhecer integralmente as topologias, tempos de vida e requisitos. Existe ainda o cenário periódico, que se posiciona entre o modelo *online* e *offline*. Em tal modelo as VNs chegam de forma similar ao cenário *online*, mas são agrupadas e tratadas em lotes. Em determinados momentos de tempo esses lotes são submetidos ao mapeamento, como em um cenário *offline*.

O VNE ainda pode ser diferenciado pela sua forma de resolução. Alguns trabalhos fazem uso de programação matemática exata ou relaxada, ou ainda uso de técnicas heurísticas. Outra vertente sugere a resolução por meio de abordagens chamadas *matheuristics*, tal linha mistura os benefícios das metaheurísticas com o da programação matemática.

Ao se desenvolver abordagens para o VNE, Fischer et al. [2013] distingue as técnicas adotadas para ao mapeamento dos componentes da requisição virtual em coordenado e não coordenado. No modelo coordenado o mapeamento de um enlace virtual é feito sempre a cada par de nós virtuais adjacentes com uma visão completa da rede física. Esse tipo de resolução tende a ter um desempenho em termos de função objetivo melhor, pois o mapeamento de nós leva em conta a contribuição que o mapeamento de enlaces virtuais têm na função objetivo. Por outro lado abordagens não coordenadas tendem a ser mais simples de se implementar, efetuando o mapeamento de todos os nós virtuais em uma primeira etapa, seguido de outra etapa para o mapeamento de todos os enlaces virtuais.

Em trabalhos recentes, o VNE é tratado em diferentes ambientes quanto à composição do SN. Em ambientes SD, o SN é composto por um único domínio de rede onde não existe variação de políticas de preços, consumo, comunicação e mapeamento. Em ambientes MD cada ISP envolvido pode gerir um ou mais domínios de rede e cada domínio pode possuir políticas de gerência de recursos independentes.

De acordo com Fischer et al. [2013] as abordagens para o VNE também se distinguem por assumir objetivos distintos, como exemplo:

- Qualidade de serviço (QoS): aborda a dimensão das acoplagens, nível de *stress* de um substrato, taxa de utilização, custo, *delay*, taxas de aceitação e rejeição e quantidade de recursos usados.

- Resiliência: analisa probabilidade de falhas e recuperação caso ocorram, necessidade e uso de *backups*, migrações de redes em caso de quedas e etc.
- Tempo de execução do algoritmo, tempo gasto efetivo para realizar todo o mapeamento.
- Número de mensagens trocadas entre os domínios e outras entidades.
- Número de reconfigurações feitas, no caso de uma abordagem dinâmica.
- Tempo gerado por paralisações feitas durante a execução de reconfigurações.
- Quantidade de energia elétrica consumida.
- Custo financeiro para se prover um mapeamento.
- Lucro gerado por cada mapeamento efetivado.

1.2.2 Aplicações

A virtualização de redes aflora benefícios inerentes à liberdade das aplicações quanto ao uso de IPs físicos e da virtualização de recursos inexistentes na arquitetura. O problema relacionado aos IPs físicos, é fundamentado no fato do campo de endereçamento de IPs presente no protocolo *IPv4* possuir 2^{32} *bits*, sendo escrito com quatro octetos de *bits* representados em número decimal que podem variar de 0 a 255, o que gera 4.294.967.296 endereços físicos diferentes e singulares. Dado o fato dos endereços de IPs físicos serem únicos por conexão em todo mundo e na prática nem todos os endereços estarem disponíveis, pois existem classes especiais para diversas classificações de rede, números reservados para máscaras de rede e *broadcasts*, depara-se com o seguinte problema: o número de aplicações que requerem conexão com a internet estão superando o número de IPs disponíveis [Beeharry & Nowbutsing, 2016]. Com a virtualização é possível criar redes virtuais com endereços IPs virtuais independentes dos endereços de IPs físicos presentes nos roteadores hospedeiros [Zhu & Ammar, 2006].

Um exemplo da aplicação de redes virtuais é o provimento de ambientes de games, como *Hamachi*, *Tunnngle*, etc. Nesse caso a virtualização de redes faz com que jogadores espalhados por diferentes lugares do mundo se conectem a outros jogadores como se estivessem em uma mesma rede privada local. Dessa forma os *softwares* envolvidos têm a impressão de se comunicar com os outros *softwares* por meio de um IP físico, mas na verdade trata-se de um IP virtual.

Quando são tratados aspectos do futuro da internet, as principais preocupações são a flexibilidade e a capacidade de possíveis alterações, para não se criar "ossificações"

futuras. Nesse sentido, para Anderson et al. [2005] e Chowdhury & Boutaba [2008], existem duas linhas de pesquisas bem definidas, os puristas e os pluralistas. Os puristas afirmam que a virtualização de redes serve apenas como um meio de avaliação de novas arquiteturas, e não como um aspecto fundamental para o futuro da internet. Por outro lado os pluralistas defendem que a virtualização é um atributo fundamental da arquitetura de redes, de forma permitir que a rede suporte concomitantemente múltiplas pilhas de protocolos, garantindo dinamicidade e facilitando a evolução da arquitetura.

A retrocompatibilidade com versões antigas é uma questão de grande preocupação ao se implantar qualquer nova tecnologia. Segundo Chowdhury & Boutaba [2008] a virtualização de redes pode ser empregada de maneira a facilitar a migração incremental do protocolo *IPv4* para o *IPv6*. Tal ação pode ser feita de forma que determinada aplicação execute sobre redes virtuais independente do protocolo usado na camada de rede. Nesse caso, o domínio ou conjunto de domínios envolvidos na prestação do serviço devem ser encarregar de viabilizar a integração entre as estruturas que fazem uso do *IPv4* ou *IPv6*. Tal ação viabiliza a continuidade do funcionamento dos aplicativos, serviços e tecnologias já existentes.

Partindo de uma visão pluralista, a virtualização de redes abre um novo leque para o desenvolvimento das mais diversas aplicações para a internet, viabilizando ao desenvolvedor a liberdade de criar/implementar funcionalidades inexistentes nos protocolos TPC/IP originais. Como exemplo de aplicações que fazem uso da virtualização de redes, estão: o provimento de *streaming*, o compartilhamento de dados via P2P (*Peer-to-peer*), telefonia sobre IP (Voip), etc. Assim, a pesquisa e desenvolvimento de projetos para o VNE atrai a atenção de empresas de tecnologia, serviços e aplicações sobre a internet.

Segundo Fischer [2016], mais recentemente as aplicações do VNE são expandidas e encontradas em diferentes áreas, como: *data centers*, que fazem uso de técnicas de virtualização para prover intercomunicação entre diferentes Máquinas Virtuais (VMs); em redes industriais, que usam a virtualização a fim de garantir a comunicação com níveis de segurança e entrega confiável de pacotes de dados entre processos críticos; nas Redes Definidas por Software (SND), de maneira a facilitar a implantação, distribuição e gerência de aplicativos; na Virtualização de Funções de Rede (NFV), que possibilita a virtualização de diversos tipos de funções dos nós da rede, como servidores, bancos de dados, etc.

Outra vantagem da aplicação de virtualização de redes está na viabilização da chamada computação nas nuvens. Neste caso as empresas não necessitam mais comprar grandes *mainframes* para prover seus sistemas. Através da virtualização de redes,

elas podem contratar servidores de dados distribuídos, acessíveis de qualquer lugar no mundo, de fácil escalabilidade, e que aprovisionam por meio de emulação um ambiente computacional completo, o que engloba sistemas operacionais, bibliotecas, sistemas de arquivos, capacidade de processamento, capacidade de armazenamento etc.

1.2.3 Complexidade

O problema de definir o melhor posicionamento para uma VN sobre o SN, mesmo que feito de maneira *offline* e sendo uma VN apenas a ser processada, é tido como pertencente à classe NP-difícil. Um problema é pertencente à classe NP-difícil se e somente se ele é tão difícil quanto os problemas mais difíceis em NP. Para se demonstrar que um problema é NP-difícil, deve-se mostrar que pode se reduzir em tempo polinomial e resolver um problema já conhecido pertencente à classe NP-difícil para o problema em questão.

Os autores Andersen [2002], Yu et al. [2008] e Amaldi et al. [2016] apresentam diferentes tipos de prova. Andersen demonstra que o problema já conhecido como pertencente à classe NP-difícil, *Multiway Separator Problem* pode ser reduzido em tempo polinomial para o problema de atribuição de nós em uma rede *ethernet* e resolvido. Yu et al. demonstram que ao menos alguns casos particulares do VNE, onde os caminhos não podem ser divididos, são pertencentes à classe NP-difícil, pois o problema já conhecido como pertencente à classe NP-difícil, *unsplittable multicommodity flow problem*, pode ser reduzido em tempo polinomial para esses casos e resolvido. Amaldi et al. propõem uma abordagem diferente, onde eles demonstram que o problema já conhecido como pertencente à classe NP-difícil, *maximum stable set problem*, pode ser reduzido em tempo polinomial para o VNE e resolvido.

Assim, o VNE mesmo para sua versão mais simples é um problema de complexidade combinatorial, classificado como NP-difícil. Neste caso a cada novo nó ou enlace adicionado de forma linear ao SN ou a VN, faz o esforço computacional para a resolução crescer exponencialmente.

1.3 Motivação

As principais motivações desta dissertação estão no fato que progressos na área de virtualização de redes, não trazem benefícios somente para a área acadêmica, mas promovem um amparo tecnológico ao desenvolvimento e o aprimoramento de diversas tecnologias. Dessa maneira, a pesquisa e desenvolvimento de projetos junto à virtuali-

zação de redes, é de atenção também de empresas de tecnologia, serviços e aplicações sobre a internet.

Segundo dados fornecidos pelo portal *Statista*³, o acesso de pessoas à internet quase dobrou na última década. Estima-se que em 2017, no mundo, 3,58 bilhões de pessoas tenham se conectado à grande rede, o que corresponde a 47% da população mundial, estimada em 7,6 bilhões. Tal aumento é amparado pela consolidação de tecnologias baseadas na computação nas nuvens.

Em conjunto com a notada expansão da internet, é gerada uma lista de impactos ambientais inerentes à produção e consumo de dispositivos eletrônicos de suporte às TICs. Em 2020 as TICs serão responsáveis pela emissão de aproximadamente 7.8 $GtCO_2$ (bilhões de toneladas de dióxido de carbono) por ano [The Climate Group, 2008]. Tal panorama de crescente demanda, acarreta em um alto consumo de recursos energéticos para fabricação e manutenção de estruturas físicas, o que leva uma vertente de pesquisadores a trabalhar com os desafios de sustentabilidade ambiental para o setor. Dentre esses desafios se destaca o de prover o consumo de energia de maneira consciente e que minimize os impactos ambientais como a produção de CO_2 .

De acordo com Addis et al. [2016], ISPs de grande porte como a *AT&T* e a *China Mobile* alcançaram juntas em 2010 o gasto energético de 11TWh (*Terawatt-hora*, equivale a $10^{12}Wh$), enquanto ISPs de médio porte como a *TelecomItalia* e a *GRNET* gastaram aproximadamente 400GWh (*Gigawatt-hora*, equivale a 10^9Wh) no ano de 2015. Além disso, Addis et al. mostra que 20% do consumo de energia é gasto pelas redes de ligações centrais do sistema (*backbone*) e 75% do consumo é dispendido nos enlances de acesso e o restante em outros pontos. Tais dados motivam uma crescente preocupação com o gasto consciente de energia dos ISPs, onde é visível o alto grau de consumo energético e geração de CO_2 para implantação e manutenção dos componentes de suporte às TICs.

De maneira oposta às redes de computadores tradicionais, à virtualização de redes pode prover ganhos inerentes ao compartilhamento de recursos virtualizados, como redução de gasto com energia elétrica, menores custos de manutenção e o aumento de produtividade dos roteadores e enlances físicos. Em um ambiente de internet, existe um custo financeiro atrelado ao gasto energético dos aparelhos de suporte às TICs, o que pode representar o dispêndio de altas cifras monetárias para os ISPs. Em ambientes reais, o fato de um roteador estar ligado ou desligado, contribui essencialmente no consumo de energia, onde o mesmo acontece para os cabeamentos físicos. O interessante

³O *Statista* é uma portal de estatísticas, perspectivas de mercado, inteligência de negócios, relatórios da indústria etc. Disponível em <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>

é que a economia de recursos elétricos pode caminhar junto com a redução dos custos de implantação e operação. A premissa é que ao se utilizar menos energia, potencialmente os gastos de comprar energia elétrica vão ser menores, o que iminentemente pode gerar um lucro maior aos provedores de infraestrutura e um impacto ambiental menor. Por esse motivo uma das motivações deste trabalho é minimizar o custo monetário gasto com energia pelos ISPs.

Diante de tantos desafios, é notável a necessidade de propor novas abordagens que propiciem uma melhora na qualidade das redes virtualizadas, gerando uma conexão melhor, com mais lucro aos ISPs e um menor gasto de energia. Levando-se em conta o crescimento de usuários conectados à internet e as inúmeras aplicações sobre ela, torna-se cada vez mais importante o desenvolvimento de novos algoritmos e abordagens para o problema.

1.4 Objetivos

1.4.1 Geral

Desenvolver diferentes abordagens para o VNE em um único domínio (VNESD) e para o VNE em múltiplos domínios (VNEMD), que atendam de maneira *online*, periódica ou *offline* as demandas de VNs, e avaliar suas respectivas vantagens e desvantagens.

1.4.2 Específicos

- Desenvolver um modelo matemático para resolução exata do VNE estático, centralizado, SD e MD, para os cenários *online*, periódico e *offline*.
- Propor uma abordagem rápida baseada em adaptações de metaheurísticas para o VNE estático, centralizado, SD e MD, para os cenários *online*, periódico e *offline*.
- Definir um conjunto de políticas de mapeamento para maximizar o lucro e minimizar o gasto de energia elétrica, a troca de mensagens e o balanceamento de carga para as resoluções exata e heurística do VNE estático, centralizado, SD e MD, para os cenários *offline*, periódico e *offline*.
- Analisar o desempenho da abordagem heurística desenvolvida com base nos resultados do modelo matemático proposto em cenários *online*, periódico e *offline*, e em ambientes SD e MD.

- Analisar o desempenho da abordagem heurística desenvolvida junto a variações de um orquestrador com conhecimento total, parcial e sem conhecimento do SN em cenários *online* e MD.

1.5 Contribuições

Este trabalho além de resolver o VNE estático, *offline*, centralizado, SD e MD, de maneira exata com um modelo matemático baseado em Programação Linear Inteira (PLI), apresenta uma heurística chamada *Greedy Randomized Adaptive Search Procedure with filter* onde a busca local feita com a heurística *Reduced Variable neighborhood search* (GRASPF-RVNS), que executa o mapeamento total de uma VN em no máximo 300ms (milissegundos) por requisição. Ao final essa dissertação ainda apresenta um estudo da heurística GRASPF-RVNS atuando junto a um orquestrador com conhecimento total, parcial e nenhum do SN. Em meio ao desenvolvimento desta dissertação se desdobraram quatro trabalhos científicos completos:

- Em 2018, foi publicado o artigo *Managing Virtual Network Embedding through Reconfiguration and Expansion*, no journal *SIMULATION: Transactions of the Society for Modeling and Simulation International* (Araújo et al. [2018]). No artigo o VNE é abordado em um ambiente SD com uma heurística GRASPF-RVNS. O foco principal do artigo são aspectos de gerência e expansão da SN a fim de maximizar a taxa de aceitação. O trabalho propõe mecanismos de reconfiguração, que buscam otimizar o uso da SN; e a aplicação de técnicas de detecção de gargalos, que definem pontos de investimentos para a ampliação física da SN.
- Em 2018, foi publicado o artigo *Virtual Network Embedding in Multi-Domain Environments with Energy Efficiency Concepts*, no International Conference on Information Networking (ICOIN 2018), realizado de entre 10 e 12 de janeiro de 2018 em Chiang Mai, Thailand (Araújo et al. [2018]). No artigo é proposta uma abordagem heurística GRASPF-RVNS de rápida execução para o VNEMD. O trabalho emprega diferentes objetivos, como maximizar o lucro, minimizar a energia, maximizar o balanceamento de carga e uma versão de otimização híbrida que trabalha com os três objetivos aninhados.
- Em 2017 foi publicado o trabalho intitulado *Modelagens para o Problema de Mapeamento de Redes Virtuais em ambientes Multidomínios*, no XLIX Simpósio Brasileiro de Pesquisa Operacional (SBPO 2017), realizado em Blumenau - SC, no

período de 27 a 30 de agosto de 2017 (Araújo et al. [2017]). No trabalho é proposta uma modelagem em PLI com funções diferentes de otimização: minimizar a energia, maximizar o lucro, minimizar o *overhead* de comunicação e maximizar o balanceamento de carga. Ao final do artigo são comparados os pontos positivos e negativos de cada objetivo.

1.6 Organização

Este trabalho foi estruturado em seis capítulos, como apresentado a seguir: No capítulo 2 são apresentados trabalhos relacionados. No capítulo 3 apresenta-se a definição do problema e as formulações propostas. No capítulo 4 é apresentada a abordagem heurística, junto aos modelos de orquestrador desenvolvidos. No capítulo 5 são apresentados os experimentos computacionais realizados. Por fim o capítulo 6 apresenta as considerações finais e a conclusão do trabalho.

Capítulo 2

Trabalhos Relacionados

Alguns trabalhos relacionados são apresentados nesse capítulo. São confrontadas abordagens que transitam pelas definições entre: estáticas ou dinâmicas, centralizadas ou distribuídas, concisas ou redundantes, exatas ou heurísticas e *online* ou *offline*. Toda taxonomia usada é proposta por Fischer et al. [2013] e previamente explicada na seção 1.2.1.

2.1 Abordagens *Offline*, Periódica e *Online*

Os trabalhos existentes podem ser classificados em *online*, periódico ou *offline*. No cenário *online*, não se conhecem as características das VNs que chegam, modelo que melhor caracteriza um ambiente real de internet, adotado em Zhu & Ammar [2006]; Chowdhury et al. [2009]; Houidi et al. [2011]; Yu et al. [2011]; Alkmim et al. [2011]; Cheng et al. [2011]; Stasi et al. [2013]; Melo et al. [2013]; Ghazisaeedi & Huang [2015]; Oliveira et al. [2015]; Liu et al. [2016]; Nguyen et al. [2017]; Beikta et al. [2017].

De maneira similar a modelagem proposta nesta dissertação, Melo et al. [2013] abordam aspectos de localização geográfica, capacidade de banda e processamento para o mapeamento dos nós e enlaces virtuais. Melo et al. diferenciam-se de outras abordagens em PLI por propor o mapeamento final com uso de formulações baseadas em restrições de *multi-commodity flow*. O trabalho proposto por Melo et al. trata o VNE de maneira exata, centralizada e *online*.

O modelo *offline* trata o mapeamento de todas as VNs simultaneamente, tal tratamento pode ser encontrado em Alkmim et al. [2011] que abordam o VNE com Programação Inteira Binária (PIB) em cenários *offline* e *online*, e Coniglio et al. [2015] que propõem a resolução do VNE através de Programação Inteira Mista. Demais abordagens *offline* são encontradas em Leivadeas et al. [2013] e Eramo et al. [2016]. O

trabalho de Alkmim et al. aplica uma abordagem exata, que gera o resultado ótimo, e outra relaxada. Os autores concluem que mesmo a abordagem relaxada, em instâncias grandes, pode levar um tempo consideravelmente alto para efetuar um mapeamento.

O modelo periódico vem em uma linha de mapeamento intermediária entre o modelo *offline* e o *online*. Houidi et al. [2011] além de trabalhar o modelo *online*, apresenta uma abordagem periódica baseada em janelas de tempo. Ao final do artigo, Houidi et al. concluem que a abordagem periódica consegue melhorar a taxa de aceitação e reduzir o custo de mapeamento em comparação a abordagem *online*.

A abordagem de Beikta et al. [2017] propõe uma heurística de mapeamento *online* e periódica, sendo o modelo periódico processado a cada fatia de tempo decorrido. Ambas abordagens são não coordenadas e centralizadas, onde primeiro são mapeados todos os nós virtuais; caso se obtenha êxito, são mapeados todos os *links* virtuais. Beikta et al. adotam restrições de mapeamento para que determinadas VNs sejam mapeadas em locais específicos, ou sobre componentes físicos disjuntos de outras VNs, com o propósito de não gerar conflito entre determinados componentes. Beikta et al. buscam maximizar o lucro do ISP e garantir o não conflito entre VNs diferentes, além de propor um fluxo variável de banda.

2.2 Abordagens Estáticas e Dinâmicas

A classificação quanto a ser estático ou dinâmico, diz respeito ao mapeamento da VN ser fixo, ou poder ser alterado conforme as necessidades do ISP. Tal aspecto também é chamado de reconfiguração e foi abordado em Eramo et al. [2016], onde foi mostrado que a reconfiguração traz vantagens em termos de melhorias na taxa de aceitação e no lucro.

Um SN fragmentado possui um conjunto de componentes físicos com recursos disponíveis, mas que não podem ser utilizados, pois os componentes adjacentes à região fragmentada não possuem recursos disponíveis para se estabelecer o roteamento/mapeamento de uma VN sobre a região [Fajjari et al., 2011; Stasi et al., 2013; Cheng et al., 2011]. Segundo Stasi et al. [2013], a constante entrada e saída de VNs em um ambiente *online* pode ocasionar o problema de fragmentação do SN. Uma alternativa para este problema é justamente o ato de se prover a reconfiguração das VNs sobre o SN, a fim de reajustar o uso dos recursos e potencialmente evitar o isolamento de recursos físicos.

Eramo et al. [2016] apresentam uma abordagem ótima em PLI, *offline* e coordenada. Como a PLI é aplicada conhecendo as requisições de todas as VNs, a solução

ótima é obtida no mapeamento inicial. No problema apresentado por Eramo et al. o fluxo de dados é variável, o que justifica o uso de reconfigurações para se recalcularem o mapeamento e minimizar o gasto de energia. Para isso duas políticas de reconfiguração são avaliadas, uma baseada no consumo imediato de banda, e outra baseada no consumo de banda definido com base em um processo de *Markov*. O trabalho conclui que a reconfiguração baseada no modelo *Markov* gerou uma redução no custo de mapeamento de 35% em relação ao outro modelo.

Em um modelo de mapeamento dinâmico, Zhu & Ammar [2006] apresenta duas abordagens *online* para comparar as melhorias geradas pelo uso da reconfiguração, chamadas de VNA I, sem reconfiguração e coordenada; e VNA II com reconfiguração e não coordenada. As abordagens de Zhu & Ammar apoiam minimizar as regiões críticas de componentes mapeados sobre o SN. Neste caso somente as requisições virtuais mapeadas sobre os componentes físicos mais estressados são reconfiguradas. Para Zhu & Ammar o conceito de *stress* se dá pelo número de mapeamentos que determinado componente físico recebeu. Dentre outros fatos, Zhu & Ammar concluem que a reconfiguração gera um uso mais igualitário de recursos do SN.

Em Ghazisaeedi & Huang [2015] é proposto um algoritmo *online* e não coordenado, com reconfiguração de enlaces e nós utilizados, a fim de minimizar o consumo de energia elétrica usado para manter ativa a infraestrutura física. No algoritmo heurístico, chamado *ONL-LLns-F*, é feita uma aprendizagem dos momentos de picos de consumo alto e baixo de energia, baseado em mapeamentos feitos em momentos anteriores. Com base no aprendizado do algoritmo, em momentos de poucos mapeamentos as redes são reconfiguradas de maneira a consumir menos energia, enquanto que em momentos de alta requisição o mapeamento original é retomado, com uso de todos os recursos de todo SN.

Ao se aplicar reconfigurações em ambientes reais de internet, necessita-se de interromper os mapeamentos e fluxos de dados, remover as conexões ativas e reconfigurá-las, o que pode gastar muito tempo, ser custoso computacionalmente e causar problemas na qualidade do serviço prestado. Alguns trabalhos que não efetuam reconfiguração podem ser observados como: Chowdhury et al. [2009]; Houidi et al. [2011]; Yu et al. [2011]; Alkmim et al. [2011]; Cheng et al. [2011]; Chiaraviglio et al. [2012]; Melo et al. [2013]; Oliveira et al. [2015]; Liu et al. [2016]; Nguyen et al. [2017]; Beikta et al. [2017].

2.3 Abordagens Concisas e Redundantes

Abordagens taxadas como redundantes, trabalham o mapeamento provendo características de resiliência a falhas. Em Oliveira et al. [2015] é apresentado uma heurística de mapeamento com duas estratégias chamadas de proativa e reativa. O mapeamento proativo é feito com a metaheurística *simulated annealing*, onde cada enlace virtual é mapeado em mais de um enlace físico. A estratégia reativa atua em caso de ataques e quedas de recursos, fazendo a migração dos mapeamentos originais para os componentes redundantes através de uma estratégia de fluxo máximo. A abordagem proposta foca justamente em minimizar o consumo energético e consequentemente maximizar o lucro. Outras abordagens redundantes podem ser observadas em Yu et al. [2011] e Zhang et al. [2012]. Yu et al. [2011] abordam o VNE com uma heurística coordenada e Zhang et al. [2012] abordam o VNE de maneiras exata e heurística, provendo recuperação de falhas de enlaces através de componentes disjuntos.

Tratar aspectos de resiliência são características importantes no provimento de conexões a internet, mas, em contrapartida, o ato de se aprovisionar uma camada extra de recursos para eventuais falhas deixa muitos recursos ociosos, o que pode afetar a taxa de aceitação, lucro e consumo de energia. Grande parte de trabalhos existentes na literatura abordam o VNE de maneira concisa. Liu et al. [2016] apresentam um algoritmo heurístico, conciso, não coordenado e inspirado em inteligência coletiva chamado de VNE-ABC (*virtual network embedding with artificial bee colony*). No trabalho de Liu et al. é proposto um conjunto de regras atribuídas às abelhas sendo; abelhas *employed* e *onlooker* responsáveis por refinar e melhorar as soluções e as abelhas *scout* que são responsáveis por fugir de ótimos locais.

Em uma abordagem concisa, *online* e baseada em computação evolucionária, Zhang et al. [2013] propõem um algoritmo de enxame de partículas (ou em inglês *particle swarm optimization* - PSO) no qual cada partícula representa uma possível solução e melhora sua posição iterativamente através de modificações no mapeamento de nós e enlaces. Outras abordagens concisas são trabalhadas em Stasi et al. [2013]; Ghazisaeedi & Huang [2015]; Liu et al. [2016]; Nguyen et al. [2017]; Beikta et al. [2017].

2.4 Abordagens Centralizadas e Distribuídas

Algoritmos centralizados possuem uma visão total do SN, podendo aplicar uma otimização global ao mapeamento, modelo adotado em Chowdhury et al. [2009]; Cheng et al. [2011]; Alkmim et al. [2011]; Zhang et al. [2013]; Liu et al. [2016]; Nguyen et al. [2017]; Beikta et al. [2017]. Apesar de muito difundido, tal modelo tende a ser não apli-

cável. Em ambientes reais de larga escala, centralizar todas as ações sobre uma única entidade afeta a escalabilidade, segurança e pode acarretar em uma fila de requisições esperando atendimento. Em contrapartida, a qualidade do mapeamento tende a ser superior do que em mapeamentos descentralizados.

Em Chowdhury et al. [2009], uma família de algoritmos centralizados formulados com PL são apresentados para se resolver o VNE. Entre eles são propostas duas variações: o D-ViNE (determinístico) e R-ViNE (randômico). Ambas abordagens são baseadas em uma estratégia de mapeamento coordenado e *online*, que objetivam aumentar a taxa de aceitação da rede. O algoritmo D-ViNE, aborda o problema de forma determinística para o mapeamento dos nós, onde a escolha dos nós físicos a serem mapeados é feita de acordo com a solução gerada por um modelo relaxado denominado VNE_RELAX. Já R-ViNE usa um mapeamento de nós aleatório. Para o mapeamento dos enlaces são usadas abordagens diferentes que são testadas nos dois algoritmos. A primeira abordagem utiliza um mapeamento de um caminho mínimo, mas não leva em conta o nível de saturação quanto ao fluxo presente. Em outra abordagem o fluxo pode ser partido em mais de um caminho mínimo, mas que se unem ao mesmo destino, chamado *multicommodity flow* (MCF). O trabalho de Chowdhury et al. [2009] possui as mesmas restrições de banda, CPU e localização geográfica presentes nesta dissertação. Por esse motivo a abordagem determinística que maximiza o balanceamento de carga, proposta por Chowdhury et al., chamada de D-ViNE-LB é usada como comparativo na seção de testes preliminares *online* e SD (seção 5.5.1).

Cheng et al. [2011] abordam o VNE em um ambiente *online* e centralizado por meio de duas heurísticas: o *RW-Maxmatch* e *RW-BFS*. No caso ambos modelos fazem um *rank* de todos os nós por meio de um modelo *Markov Random Walks*. *RW-Maxmatch* mapeia os nós virtuais de acordo com o *rank* gerado e os enlaces virtuais através de *multi-commodity flow*. Diferentemente, *RW-BFS* usa uma estratégia de *backtracking* junto a uma busca em largura (*Breadth-First Search - BFS*), onde à medida que os nós são descobertos eles são mapeados de forma coordenada.

Abordagens distribuídas são encontradas em menor número na literatura. Neste caso o mais trabalhado são *frameworks* de como se aplicar algoritmos centralizados em ambientes descentralizados, de maneira a deixar aberta a escolha da abordagem de resolução a ser usada. Beck et al. [2015] apresentam um *framework* chamado DPVNE, de como se aplicar algoritmos centralizados em ambientes descentralizados, tornando-os escaláveis. Beck et al. apresentam algumas técnicas de particionamento do SN, de forma que vários SN coexistam lateralmente em frações do SN que podem ser gerenciadas separadamente. No trabalho ainda é afirmado que abordagens centralizadas possuem uma baixa escalabilidade e um forte *delay* para se mapear requisições que chegam

simultaneamente.

Em Samuel et al. [2013], um *framework* para o mapeamento descentralizado em MD é apresentado. A formulação proposta é baseada no particionamento da VN em k diferentes subgrafos a serem mapeados junto a um subconjunto de domínios envolvidos. Na proposta de Samuel et al., cada domínio é livre para aplicar seu próprio algoritmo de mapeamento de forma independente e, caso seja interessante, repassar parte da requisição para outro domínio. Para tanto, o próprio SP deve conhecer o(s) domínio(s) para requisitar o mapeamento, ressaltando que no caso do SP conhecer apenas um domínio, pode haver um tipo de monopólio onde somente o domínio conhecido será favorecido financeiramente.

O conceito de orquestração de redes virtuais, proposto nos trabalhos de Guerzoni et al. [2014] e Dalla-Costa et al. [2017] atribui a uma entidade centralizada, chamada de orquestrador, a função de coordenar o mapeamento de uma VN sobre o SN composto por MD. No trabalho Samuel et al. [2013] a entidade responsável pelo particionamento de uma VN em subgrafos é classificada como um orquestrador de mapeamentos. Diferentemente de Samuel et al. [2013], nessa dissertação é proposto um orquestrador para ambientes MD, mas com três variações: com conhecimento total, parcial e sem conhecimento do SN.

Leivadeas et al. [2013] apresentam um *framework* junto a modelos de resolução para o mapeamento distribuído de requisições, *offline* e em nuvens de dados. Tal resolução é feita de duas formas: com uma heurística e com um modelo exato. A abordagem heurística é baseada no uso da metaheurística *Iterated Local Search* (ILS) como busca local. No caso, a solução inicial é construída de maneira gulosa e submetida a uma sequência alternada de busca local e perturbações, sendo tanto a busca local quanto a perturbações retiradas da literatura [Tao et al., 1992]. Ao final, a abordagem heurística é confrontada com um modelo exato, formulado com base em uma PLI extraída de um modelo de programação quadrática. Para o mapeamento em mais de um domínio, cada VN é repartida em sub-VNs, e cada sub-VN é repassada a um orquestrador que faz consultas sobre qual domínio é mais barato para o mapeamento. A premissa é que o mapeamento feito sempre nos servidores mais baratos implica em minimizar o custo de maneira global. Leivadeas et al. concluem que sem uma alocação de recursos adequada, algumas VNs podem ser direcionadas para um único servidor, o que afeta o balanceamento de carga e gera congestionamentos.

A abordagem trabalhada por Houidi et al. [2011] atua em um ambiente de múltiplas estruturas de rede. Antes de se efetuar um mapeamento, caso seja necessário, a abordagem proposta divide cada VN entrante, sendo que para cada roteador virtual é atribuído um domínio para ser mapeado. O particionamento é feito com base em

algoritmos de *max-flow/min-cut* e técnicas de programação linear. Em uma segunda fase o mapeamento é feito de forma exata e heurística. Houdi et al. ainda apresentam uma prova, com base no problema MAX-2-SAT, que o particionamento ótimo de uma VN sobre determinado grafo é um problema pertencente a classe NP-difícil.

2.5 Abordagens com diferentes objetivos e outros aspectos

Quanto aos objetivos dos mapeamentos, algumas variações são encontradas na literatura: Houdi et al. [2011] objetivam minimizar o custo de mapeamento; Samuel et al. [2013] têm como objetivo maximizar o lucro de cada ISP, minimizar o número de recursos usados e a comunicação inter-domínios; Addis et al. [2016] objetivam melhorar a eficiência energética; Cheng et al. [2011], Chiaraviglio et al. [2012], Oliveira et al. [2015], Liu et al. [2016] e Beikta et al. [2017] objetivam melhorar o lucro gerado pelo ISP; Melo et al. [2013] comparam vários objetivos diferentes, dentre eles maximizar o balanceamento de carga, minimizar a distância dos caminhos físicos e maximizar a banda residual dos enlaces.

Para Chiaraviglio et al. [2012] dois fortes aspectos estão relacionados ao lucro de um ISP: o número de clientes ativos e o consumo de energia gasto para manter uma VN ativa. Assim, quando o objetivo é aumentar o lucro, é substancial que o ISP proveja bons mapeamentos de rede. Na estratégia heurística, coordenada e *online* proposta por Chiaraviglio et al. os componentes virtuais devem ser mapeados sempre com o menor número de recursos físicos disponíveis, conseqüentemente os recursos não usados podem ser desligados, o que gera economia de energia.

Addis et al. [2016] propõem um modelo chamado de *Energy-aware Network Management (EANM)* que recomenda um conjunto de técnicas verdes que objetivam melhorar a eficiência energética das abordagens. A meta do EANM é adaptar o consumo das redes aos níveis de tráfego percebidos. O modelo proposto por Addis et al. realiza uma análise do comportamento dos mapeamentos e o perfil energético dos componentes de redes, protocolos e requisitos de QoS de cada VN, para ao final reduzir o uso energético dos componentes físicos.

Minimizar o consumo de energia e maximizar o balanceamento de carga são objetivos diferentes que podem afetar diretamente a qualidade de serviço dos mapeamentos efetivados. Enquanto Chiaraviglio et al. [2012] propõem usar o mínimo de recursos físicos possíveis, Ghazisaeedi & Huang [2015] propõem uma reconfiguração a fim de minimizar o consumo de energia. Segundo Srikantiah et al. [2008], deve se levar em

conta que o balanceamento de carga é benéfico do ponto de vista do desempenho dos *data centers* que podem estar envolvidos na rede mapeada. Para Srikantaiah et al. [2008], um nível de processamento acima de 70% da capacidade de CPU de um *data center* já causa uma degradação do serviço prestado.

Consoante ao trabalho de Srikantaiah et al. [2008], Zheng & Veeravalli [2012] estipulam faixas de carga de trabalho ideais para os *mega data centers*. Segundo Zheng & Veeravalli um *data center* com uma carga de trabalho inferior a 30% pode ser prejudicial do ponto de vista de eficiência energética e impactar negativamente na relação custo/lucro gerado pelo processamento de dados. Por outro lado, uma carga de trabalho superior a 70% pode causar problemas de processamento, vazão e latência, o que é prejudicial ao usuário que está conectado.

As redes de computadores tradicionais utilizam intensivamente equipamentos específicos para prover funções de bancos de dados, tradução de endereços, *firewall*, *proxy*, *honeypots*, etc. Tais equipamentos são chamados de *middleboxes* e em um modelo de rede tradicional não são virtualizáveis. Em uma arquitetura de rede diferenciada, o VNE pode requerer o uso de funções específicas de redes, de forma que os *middleboxes* são substituídos por máquinas virtuais (VM) de propósito geral. Essa variação do problema é chamada na literatura de Virtualização de Funções de Rede (NFV). Nesse caso as funções de rede podem ser instanciadas em qualquer lugar do SN, onde o fluxo de dados das conexões deve obrigatoriamente passar por essas funções de redes seguindo uma ordem estabelecida na definição do acesso. O trabalho apresentado por Luizelli et al. [2015] traz uma modelagem em Programação Linear Inteira (PLI) e uma heurística para realizar a orquestração das requisições com funções de rede, sendo em ambas abordagens aplicada a função objetivo de reduzir o número de funções de rede instanciadas.

Dada a taxonomia apresentada na seção 1.2.1, alguns dos trabalhos relacionados utilizados no referencial teórico são sumarizados e mostrados na Tabela 2.1. Ainda seguindo a taxonomia de Fischer et al. [2013], as abordagens implementadas nessa dissertação se caracterizam por serem:

- Estáticas, onde caso a VN seja mapeada com sucesso, seu mapeamento é mantido na mesma posição até que seu tempo de vida seja expirado e caso a VN seja rejeitada, esta é descartada sem novas tentativas de mapeamento.;
- Concisas, onde não são providos aspectos de resiliência de recursos;
- Centralizadas, onde o mapeamento pode ser centralizado a uma única entidade ou coordenado por uma única entidade;

- *Online*, periódicas e *offline*, onde os três casos são tratados separadamente e comparados.

Tabela 2.1: Classificação dos trabalhos relacionados

Trabalho	Taxonomia
Zhu & Ammar [2006]	estático e dinâmico, conciso, centralizado, heurístico, minimizar os componentes estressados
Chowdhury et al. [2009]	<i>online</i> , estático, conciso, centralizado, heurístico, maximizar o balanceamento de carga
Alkmim et al. [2011]	<i>offline</i> e <i>online</i> , estático, conciso, centralizado, exato (ótimo e relaxado), minimizar a largura de banda
Cheng et al. [2011]	<i>online</i> , estático, conciso, centralizado, heurístico, maximizar o lucro
Yu et al. [2011]	<i>online</i> , estático, redundante, centralizado, heurístico, minimizar o custo de reconfiguração
Houidi et al. [2011]	<i>online</i> e periódico, estático, conciso, centralizado, exato (ótimo) e heurístico, minimizar o custo
Zhang et al. [2012]	<i>online</i> , estático, redundante, centralizado, exato (ótimo) e heurístico, minimizar o <i>delay</i>
Zhang et al. [2013]	<i>online</i> , estático, conciso, centralizado heurístico, maximizar o lucro
Stasi et al. [2013]	<i>online</i> , dinâmico, conciso, centralizado, heurístico, maximizar o lucro
Leivadeas et al. [2013]	<i>offline</i> , estático, conciso, distribuído, heurístico e exato (ótimo), minimizar o custo de mapeamento
Melo et al. [2013]	<i>online</i> , estático, conciso, centralizado, exato (ótimo), maximizar o balanceamento de carga
Ghazisaeedi & Huang [2015]	<i>online</i> , dinâmico, conciso, centralizado, heurística, minimizar o gasto de energia
Oliveira et al. [2015]	<i>online</i> , estático, redundante, centralizado, heurístico, maximizar a banda residual
Coniglio et al. [2015]	<i>offline</i> , estático, conciso, centralizado, exato (ótimo), maximizar o lucro
Liu et al. [2016]	<i>online</i> , estático, conciso, centralizado, heurístico, maximizar o lucro
Eramo et al. [2016]	<i>offline</i> , dinâmico, conciso, centralizado, exato (ótimo), minimizar o custo de energia e de reconfiguração
Nguyen et al. [2017]	<i>online</i> , estático, conciso, centralizado, heurístico, maximizar o lucro
Beikta et al. [2017]	<i>online</i> e periódico, estático, conciso, centralizado, heurístico, maximizar o lucro

Capítulo 3

Mapeamento de Redes Virtuais e Formulações

A resolução do VNE por meio de uma abordagem exata busca encontrar uma solução de boa qualidade ou ótima. Devido ao número elevado de combinações para o problema, definir a solução que traz o melhor retorno é uma tarefa computacionalmente árdua. O principal objetivo deste capítulo é definir e apresentar uma formulação matemática para o VNE, bem como um conjunto de objetivos que abordam os desafios inerentes à atualidade.

3.1 Definição Formal do Problema

Nessa dissertação, o modelo arquitetural da interação entre usuário final, SPs e ISPs, segue um modelo similar ao proposto por Chowdhury & Boutaba [2008], como mostrado nas etapas:

1. Um usuário final da rede gera uma requisição por meio de uma aplicação qualquer;
2. A requisição é repassada a um SP que oferece suporte à aplicação;
3. O SP entra em contato com o(s) ISP(s) que podem potencialmente realizar o mapeamento da VN;
4. O ISP escolhido para hospedar a VN realiza o mapeamento e o devolve ao SP;
5. De posse da VN mapeada, o SP comunica aos pares usuários finais da aplicação que a conexão foi estabelecida;

O modelo aqui proposto ainda contém uma unidade de controle chamada de orquestrador de rede. O orquestrador possui a função de intermediar as comunicações entre os SPs e os ISPs. Em nosso modelo, a cada requisição de VN feita por um usuário final, o SP contrata os serviços do ISP com garantia de conexão por um determinado tempo de vida, ou seja, os recursos já são contratados por um intervalo de tempo estabelecido.

Em trabalhos da literatura como Chowdhury et al. [2009]; Nonde et al. [2015] o VNE é abordado em um domínio, sendo chamado de monodomínio (VNESD). A abordagem aqui proposta expande esse arquétipo para um mundo composto por vários domínios que disputam mercado (VNEMD). No VNEMD cada domínio detém uma gama de dispositivos físicos de rede que são usados para a comunicação entre os pares de roteadores virtuais. A definição do problema bem como a modelagem apresentada é então expandida para um tipo genérico, que pode ser utilizada tanto em cenários SD, quanto em MD.

Deste modo, tanto o VNESD como o VNEMD podem ser definidos através de um grafo não direcionado $G = (N, E)$ que representa o SN. Seja $N = \{N^1 \cup N^2 \dots \cup N^{|D|}\}$, a união dos subconjuntos de roteadores físicos, onde cada $N^d \subset N$ é um conjunto de roteadores físicos pertencente ao domínio $d \in D$ e $D = \{1, 2, 3, \dots, |D|\}$ o conjunto de domínios existentes. Similarmente $E = \{E^1 \cup E^2 \dots \cup E^{|D|}\}$ representa a união dos subconjuntos de enlaces físicos; cada $E^d \subset E$ é um conjunto de enlaces físicos internos ao domínio $d \in D$ (enlaces intra-D). O conjunto E^0 representa os enlaces inter-domínio, que são aqueles que fazem fronteira e possuem gerência acordada entre os domínios (enlaces inter-D). Cada nó $i \in N$ possui uma capacidade máxima de CPU C_i associada e um posicionamento geográfico (x_i, y_i) fixo. Cada enlace $(i, j) \in E \cup E^0$ possui uma capacidade de banda máxima B_{ij} . Um exemplo da representação do grafo G é mostrado na Figura 3.1, que possui três domínios diferentes e cada domínio possui um acordo de ligação com os demais domínios, através de um enlace inter-domínio.

Seja V o conjunto de VNs. Cada requisição $v \in V$ possui um tempo de entrada t_e^v , um tempo de vida t^v e é representada como um grafo não direcionado $G^v = (N^v, E^v)$, onde N^v representa o conjunto de roteadores virtuais e E^v o conjunto de enlaces virtuais. Cada nó virtual $k \in N^v$ possui uma demanda de CPU diferente c_k^v e uma posição geográfica (x_k, y_k) que é utilizada para definir uma região de mapeamento com base em um raio de localização d_{max}^v . Para cada nó virtual $k \in N^v$ a viabilidade de posicionamento geográfico sobre um nó físico $i \in N$ é calculada com o uso de uma função euclidiana $d((x_i, y_i), (x_k, y_k))$ e d_{max}^v . Se $d((x_i, y_i), (x_k, y_k)) \leq d_{max}^v$ o roteador físico i pode hospedar o nó virtual k ; caso contrário, este mapeamento não é válido. Cada enlace virtual $(k, l) \in E^v$ possui uma demanda de banda b_{kl}^v requerida para o mapeamento. Um exemplo de VN composta por 3 roteadores físicos e suas respectivas regiões

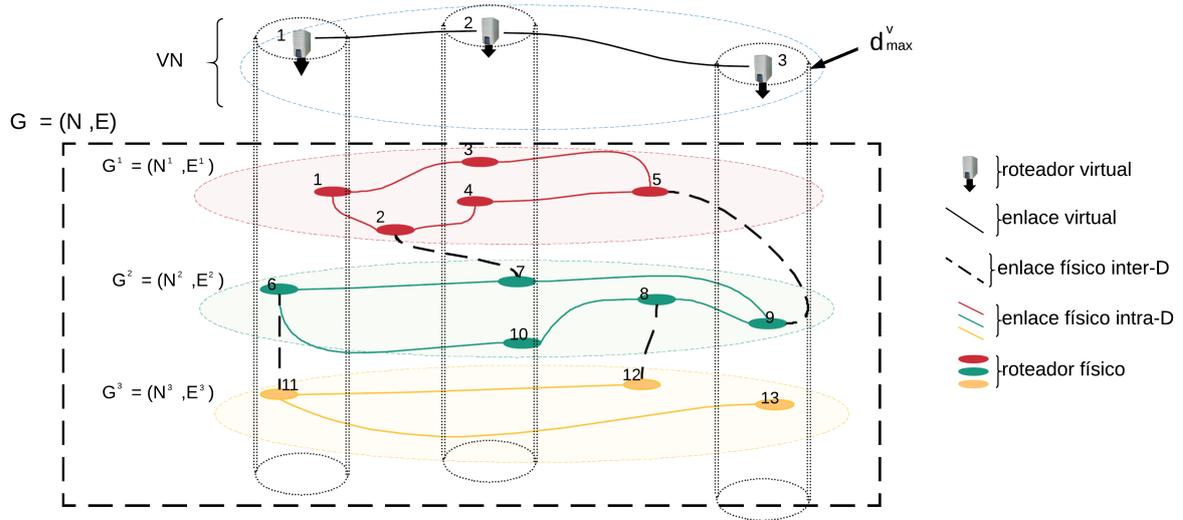


Figura 3.1: Exemplo da modelagem do SN com três domínios e uma VN .

de mapeamento pode ser visto na Figura 3.1.

Uma solução para o mapeamento de uma VN $v \in V$ no VNEMD consiste em determinar com sucesso o mapeamento $f : G^v \rightarrow G$. Assim, o mapeamento de nós¹ é dado por: $\forall k \in N^v, \exists i \in N : c_k^v \leq \bar{C}_i$, onde \bar{C}_i representa o CPU residual do nó físico $i \in N$. O mapeamento de enlaces² é dado por: $\forall (k, l) \in E^v, \exists \{(i, j)^1, \dots, (i, j)^m\} : b_{kl}^v \leq \bar{B}_{ij}^1, \dots, b_{kl}^v \leq \bar{B}_{ij}^m$, onde \bar{B}_{ij} representa a largura de banda residual dos enlaces $(i, j) \in E \cup E^0$ que compõem o caminho físico que irá hospedar o enlace virtual. Caso as duas etapas sejam atendidas com sucesso, a VN pode ser mapeada; caso contrário o mapeamento não é possível nos recursos residuais existentes.

3.2 Formulações Propostas

Tanto o VNESD quanto o VNEMD podem ser modelados com uso de PLI, através de um conjunto de variáveis e restrições a serem respeitadas. De modo análogo à definição do problema (seção 3.1), a formulação matemática denotada pode ser aplicada em ambas variações do VNE sem alteração. Para tal, admite-se V como conjunto de requisições a serem mapeadas, e considera-se $|V| = 1$ no cenário *online* e $|V| > 1$ no cenário *offline* ou periódico. Para tal modelagem toma-se as seguintes variáveis de decisão:

- $y^v \in \{0, 1\}$ - Se igual a 1 indica que a VN $v \in V$ foi mapeada com sucesso, 0 caso

¹Cada nó virtual de uma mesma VN deve ser mapeado em um nó físico diferente

²Cada enlace virtual pode ser mapeado em um enlace físico único ou em um caminho constituído por mais de um enlace físico

contrário.

- $z_{ki}^v \in \{0, 1\}$ - Se igual a 1 indica que o nó virtual $k \in N^v$ da VN $v \in V$ foi mapeado sobre o nó físico $i \in N$, 0 caso contrário.
- $x_{ij}^{vkl} \in \{0, 1\}$ - Se igual a 1 indica que o enlace virtual $(k, l) \in E^v$ da VN $v \in V$ foi mapeado sobre o enlace físico $(i, j) \in E \cup E^0$, 0 caso contrário.

O mapeamento de uma VN depende da topologia e das condições do SN. Cada ISP possui uma visão local do SN, e conhece somente as condições da estrutura física de seu domínio. Neste caso, ao se efetuar o mapeamento em mais de um ISP, gera-se um tráfego a mais resultante da comunicação entre os domínios envolvidos, chamado de troca de mensagens, representado pelo parâmetro:

- b_{ij} - consiste no consumo extra de banda pelo enlace físico $(i, j) \in E^0$ para a troca de mensagens entre um par de roteadores i e j que estabelecem a comunicação inter-domínio.

A formulação utilizada para a resolução exata por meio de PLI faz uso das seguintes restrições:

$$\sum_{i \in N} a_{ki}^v z_{ki}^v \geq y^v \quad \forall k \in N^v, \forall v \in V \quad (3.1)$$

$$\sum_{k \in N^v} a_{ki}^v z_{ki}^v \leq 1 \quad \forall i \in N, \forall v \in V \quad (3.2)$$

$$\sum_{v \in V} \sum_{k \in N^v} c_k^v a_{ki}^v z_{ki}^v \leq \bar{C}_i \quad \forall i \in N \quad (3.3)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} b_{kl}^v x_{ij}^{vkl} \leq \bar{B}_{ij} \quad \forall (i, j) \in E \quad (3.4)$$

$$\sum_{v \in V} \sum_{(k,l) \in E^v} (b_{kl}^v + b_{ij}) x_{ij}^{vkl} \leq \bar{B}_{ij} \quad \forall (i, j) \in E^0 \quad (3.5)$$

$$\sum_{(i,j) \in E \cup E^0} x_{ij}^{vkl} - \sum_{(h,i) \in E \cup E^0} x_{hi}^{vkl} = z_{ki}^v - z_{li}^v \quad \forall i \in N, \forall (k, l) \in E^v, \forall v \in V \quad (3.6)$$

As restrições 3.1 asseguram que todos os nós virtuais $k \in N^v$ são mapeados para a VN ser aceita, enquanto as restrições 3.2 asseguram que todos os nós virtuais $k \in N^v$ são mapeados em nós físicos $i \in N$ diferentes. As restrições 3.3 asseguram que o mapeamento não irá extrapolar a capacidade de CPU residual \bar{C}_i de cada nó físico $i \in N$, similarmente as restrições 3.4 asseguram que o mapeamento não irá extrapolar a capacidade de banda residual \bar{B}_{ij} de cada enlace físico intra-domínio $(i, j) \in E$. As restrições 3.5 asseguram que o mapeamento não irá extrapolar a capacidade de banda

residual \bar{B}_{ij} de cada enlace físico inter-domínio $(i, j) \in E^0$. Por fim, as restrições de fluxo 3.6 asseguram os mapeamentos dos enlaces virtuais sobre caminhos de enlaces físicos.

Um exemplo de mapeamento *offline* viável é mostrado na Figura 3.2, onde duas VNs são mapeadas simultaneamente sobre o mesmo SN (neste caso $|V| = 2$), de maneira a compartilhar os recursos físicos disponíveis concomitantemente (sendo $|D| = 1$). Repare que os recursos de CPU e banda requeridos por ambas VNs devem ser suportados pelos disponíveis no SN no momento do mapeamento, bem como o raio de mapeamento de cada roteador virtual deve ser respeitado.

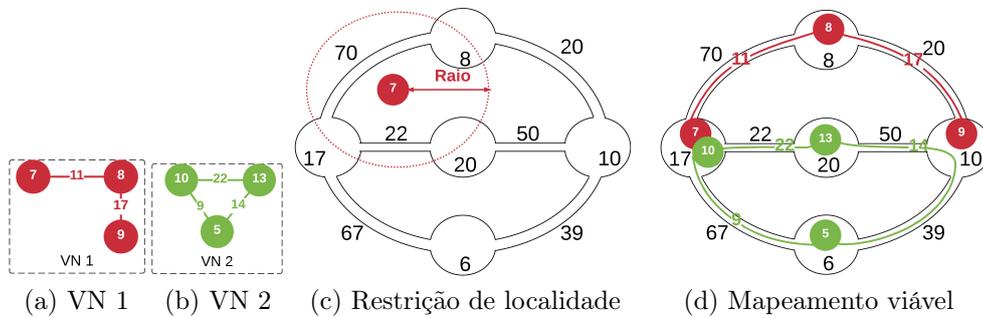


Figura 3.2: Representação do mapeamento viável de duas VNs sobre um domínio

3.2.1 Maximizar o balanceamento de carga

Com o intuito de maximizar o balanceamento de carga e potencialmente gerar menos gargalos sobre o SN, a função objetivo pode ser moldada de maneira a distribuir melhor os mapeamentos virtuais sobre os componentes físicos com mais banda residual. Para executar esse método calcula-se a porcentagem de uso dos enlaces físicos, tal estimação é feita através da razão de uso pelo residual dos enlaces. Assim VNESD e o VNEMD podem então ter sua função objetivo formulada a fim de melhorar o balanceamento de carga. Tal função objetivo é mostrada na equação 3.7:

$$\text{Minimizar} \sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{(i,j) \in E} \frac{b_{kl}^v}{B_{ij}} x_{ij}^{vkl} + \sum_{v \in V} \sum_{(k,l) \in E^v} \sum_{(i,j) \in E^0} \frac{(b_{kl}^v + b_{ij})}{B_{ij}} x_{ij}^{vkl} + M \cdot \sum_{v \in V} (1 - y^v) \quad (3.7)$$

3.2.2 Minimizar consumo de energia elétrica

Dentre as características inerentes aos componentes de um ISP, cada roteador e cada enlace podem assumir três estados de atividade distintos: *Idle*, quando o roteador ou enlace está inativo, neste caso não existe o consumo de energia elétrica; *stand-by*, quando o roteador ou enlace está em estado de espera para efetuar algum processamento, neste o consumo de energia elétrica é o mínimo; e *working*, quando o roteador ou enlace já estão de fato ligados, processando/transmitindo alguma informação, neste caso existe o consumo de energia em *stand-by* acrescido da energia para efetuar-se o processamento/transmissão. Assim são dados os seguintes parâmetros (aferidos em *watts*):

- Γ^d - consumo de energia elétrica de um roteador em *stand-by* pertencente ao domínio d . Consiste no gasto de energia mínimo de manter-se ligado a um roteador;
- γ^d - consumo adicional de energia elétrica de um roteador pertencente ao domínio d a cada 1 *Mbps* de CPU processado. Consiste no gasto adicional de energia demandado por um roteador;
- Δ^d - consumo de energia elétrica de um enlace intra-domínio em *stand-by* a cada *Km* de enlace pertencente ao domínio d . Quanto mais distante geograficamente os roteadores físicos do enlace $(i, j) \in E$ estiverem, maior o gasto de energia. Consiste no gasto de energia mínimo para manter ligado um enlace inter-domínio;
- δ^d - consumo adicional de energia elétrica de um enlace intra-domínio pertencente ao domínio $d \in E$, a cada 1 *Mbps* de banda consumida por *Km* de conexão; Consiste no gasto adicional de energia demandado por um enlace intra-domínio;
- Ω^{de} - consumo de energia elétrica de um enlace inter-domínio em *stand-by* a cada *Km* de enlace pertencente às fronteiras dos domínios $d, e \in D$. Quanto mais distante geograficamente os roteadores do enlace $(i, j) \in E^0$ estiverem, maior o consumo de energia. Consiste no gasto de energia mínimo para manter ligado um enlace de fronteira;
- ω^{de} - consumo adicional de energia elétrica adicional de um enlace inter-domínio pertencente às fronteiras dos domínios $d, e \in D$, a cada 1 *Mbps* de banda consumida por *Km* de conexão. Consiste no gasto de energia mínimo para manter ligado um enlace intra-domínio;
- χ , custo de energia elétrica a cada *MW* consumido.

Ante a filosofia do gasto consciente de energia elétrica, o problema é formulado com uso de medidas sustentáveis, onde se tenta adequar as tecnologias de mapeamento virtual para o uso racional de energia elétrica. Assim os componentes desligados podem ser maximizados, de modo a permanecerem no estado *idle*, para reduzir o consumo de energia. Com base nesse princípio algumas fórmulas são incluídas a fim de delinear a função objetivo.

$$C1^v = t^v \cdot \left(\sum_{i \in N} \Gamma^i \cdot \sum_{k \in N^v} \gamma^i c_k^v z_{ki}^v \right) \quad (3.8)$$

$$C2^v = t^v \cdot \left(\sum_{(i,j) \in E} \Delta^{ij} \cdot \sum_{(k,l) \in E^v} \delta^{ij} \cdot b_{kl}^v \cdot x_{ij}^{vkl} \cdot d((x_i, y_i), (x_j, y_j)) \right) \quad (3.9)$$

$$C3^v = t^v \cdot \left(\sum_{(i,j) \in E^0} \Omega^{ij} \cdot \sum_{(k,l) \in E^v} \omega^{ij} \cdot b_{kl}^v \cdot x_{ij}^{vkl} \cdot d((x_i, y_i), (x_j, y_j)) \right) \quad (3.10)$$

O consumo de energia gasta por determinada VN $v \in V$ ativa sobre o SN, é dado pelo gasto dos componentes físicos em *stand-by* acrescido pela quantidade de recursos utilizados durante todo o período de atividade da VN, dado por t^v . Desta forma, $C1^v$ representa o consumo de energia de todos os roteadores; $C2^v$ o consumo de energia de todos enlaces intra-D; e $C3^v$ o consumo de energia de todos enlaces inter-D, usados pela VN $v \in V$. Neste caso a função objetivo de minimizar o consumo de energia elétrica é dada por:

$$\text{Minimizar} \quad \sum_{v \in V} (c1^v + c2^v + c3^v) + M \cdot \sum_{v \in V} (1 - y^v) \quad (3.11)$$

O modelo de energia empregado nessa dissertação é simplificado e de fácil assimilação. Em aplicações específicas tal modelo pode ser alterado de maneira a melhor se ajustar as necessidades do ambiente tratado. O IEEE 802.3az³ é um exemplo de padrão de gasto de energia que pode ser adotado, e que trata do consumo e taxas de dados suportadas por cabos de par trançado não blindado (ou em inglês *Unshielded Twisted Pair* - UTP).

³Dados retirados de <http://grouper.ieee.org/groups/802/3/az/>, acessado em: 03/03/2018.

3.2.3 Maximizar o lucro dos domínios

Os ISPs são constituídos por organizações empresariais que fornecem acesso à internet, onde o mesmo não é feito de forma gratuita. Para cada ISP pode ser interessante mapear toda VN em seu domínio para obter maior ganho; ou ainda mapear a VN com uso de recursos acordados com um domínio adjacente, para não rejeitar a VN. Para modelar esta tarifação alguns parâmetros são definidos, sendo tarifados em unidades monetárias (\$):

- κ , receita recebida pelo ISP, pelo mapeamento de um roteador virtual, calculado por *Mbps* de CPU processado.
- φ , receita recebida pelo ISP, pelo mapeamento de um enlace virtual, cobrado por banda consumida a cada *Km* de distância entre os roteadores virtuais.

Usar princípios de gasto consciente de energia elétrica para se formular o VNE, não propriamente é usar o mapeamento que gera menor ou maior ônus para o cliente ou lucro para o ISP. Neste caso, diferentes ISPs podem acarretar em diferentes tarifas de uso de roteadores e enlaces físicos, pois cada componente pode possuir um diferente consumo de energia elétrica. O VNESD e o VNEMD podem então ter sua função objetivo definida a fim de maximizar o lucro dos ISPs, para tal, algumas equações são definidas.

A receita cobrada pelo mapeamento de uma VN $v \in V$ é dada em função da quantidade de recursos virtuais demandados pela própria VN durante todo seu tempo de vida, expressa por:

$$R^v = t^v \cdot \left(\sum_{k \in N^v} c_k^v \cdot \kappa + \sum_{(k,l) \in E^v} b_{kl}^v \cdot \varphi \cdot d((x_k, y_k), (x_l, y_l)) \right) \quad (3.12)$$

A função objetivo que busca maximizar o lucro dos domínios é formulada pela diferença entre a receita (equação 3.12) e o custo financeiro de energia (equações 3.8, 3.9 e 3.10), expressa por:

$$\text{Maximizar} \quad \sum_{v \in V} (R^v \cdot y^v - (C1^v + C2^v + C3^v) \cdot \chi) \quad (3.13)$$

3.2.4 Minimizar a comunicação entre domínios

A troca de mensagens entre os domínios diferentes pode acarretar em gargalos no SN devido ao uso extra de banda para a comunicação entre os domínios adjacentes. Tal

aspecto pode em certos momentos afetar o desempenho dos mapeamentos. O VNE pode então ter sua função moldada a fim de minimizar a comunicação entre domínios, onde a função objetivo pode ser definida como:

$$\text{Minimizar} \quad \sum_{v \in V} \sum_{(i,j) \in E^0} \sum_{(k,l) \in E^v} t^v \cdot (b_{ij} \cdot x_{ij}^{vkl}) + M \cdot \sum_{v \in V} (1 - y^v) \quad (3.14)$$

Na formulação das funções 3.7, 3.11 e 3.14 foi aplicado uma constante M de valor muito alto que age como uma penalidade para cada VN não atendida. Essa constante atua de maneira a forçar a variável y^v de cada VN entrante $v \in V$ a assumir o valor 1, garantindo o mapeamento de todos os nós virtuais (restrições 3.1). Caso o valor da função objetivo seja muito alto em cenários de minimização, tem-se a garantia que determinada VN não foi atendida, onde o valor da variável y^v em questão deve assumir o valor 0.

3.3 Abordagens *Online*, Periódica e *Offline*

A modelagem exposta na seção 3.2 pode atuar tanto em um ambiente *online*, quanto periódico ou *offline*. A diferença entre as abordagens está na quantidade de VNs que são processadas por vez. Em tal tratamento, as VNs são agrupadas em lotes para serem processadas em grupos. Neste caso o tamanho do lote pode variar, onde ele depende do número de VNs requisitantes em determinada janela de tempo t .

No modelo *online*, o tamanho da janela de tempo do conjunto de VNs a ser processado é sempre 1, ou seja, cada VN é processada em separado de acordo com o seu tempo de chegada. Neste caso a VN não sofre atraso de mapeamento, visto que a mesma é tratada de imediato. No exemplo da Figura 3.3 as VNs são processadas na ordem $V = \{VN1\}, \{VN2\}, \{VN3\}, \{VN4\}, \{VN5\}$. Neste caso são feitas 5 execuções do modelo matemático, uma para cada VN.

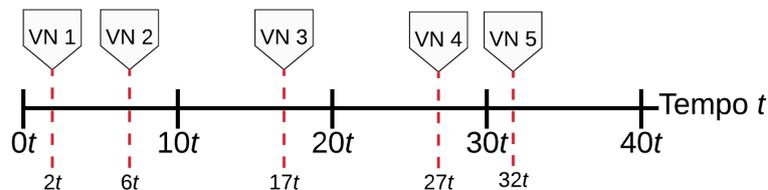


Figura 3.3: Exemplo do tempo de chegada das requisições.

Pelos trabalhos da literatura, conceitualmente o modelo *offline* não possui princí-

pios de tempo de entrada, saída ou duração. Na abordagem o tamanho do lote de VNs a ser processado é sempre a quantidade total de VNs existentes. No caso do exemplo da Figura 3.3 o lote de requisições é de tamanho 5. Neste contexto todas as VNs são mapeadas juntas em um só lote composto por $V = \{VN1, VN2, VN3, VN4, VN5\}$. Nessa abordagem é feita apenas 1 execução do modelo matemático.

O modelo periódico pode ser concebido como uma abordagem proveniente das abordagens *online* e *offline*. Em tal modelo a janela de processamento pode ter sua duração alterada e os lotes de mapeamento um número variado de VNs. No exemplo da Figura 3.3, suponha que a duração da janela de processamento seja $10t$, neste caso o modelo matemático é executado por 4 vezes, sendo para $V = \{VN1, VN2\}, \{VN3\}, \{VN4\}, \{VN5\}$. Diferentemente do modelo *online*, este modelo gera atrasos, pois a $VN1$ e a $VN2$ devem esperar até a janela de tempo fechar para serem processadas, e assim sucessivamente. Ao final, no exemplo da Figura 3.3 o atraso total é de $26t$.

Capítulo 4

Abordagem Heurística e Orquestração de Redes

Dado a complexidade do problema, resolver o VNE de maneira exata mesmo que por meio de *solvers* consagrados como o CPLEX pode ser computacionalmente impraticável. O objetivo deste capítulo é apresentar uma heurística computacionalmente eficiente que resolva tanto o VNESD quanto o VNEMD, em ambientes *online*, *offline* e periódico. Este capítulo ainda apresenta diferentes abordagens de orquestradores para o VNEMD, que possuem a função de coordenar os mapeamentos de maneira descentralizada.

4.1 Heurística GRASPF-RVNS

Devido ao VNE ser pertencente à classe NP-difícil, como visto na seção 1.2.3, existe uma inviabilidade computacional de sua resolução exata para instâncias de grande porte e abordagens online. Deste modo é proposta uma heurística para resolvê-lo. Heurísticas são técnicas que procuram boas soluções e em tempo computacional aceitável, mas que por outro lado abrem mão da garantia de otimalidade. A heurística proposta é chamada de GRASPF-RVNS, e possui princípios da metaheurística GRASP, apresentada por Feo & Resende [1995] e da metaheurística VNS concebida por Mladenović & Hansen [1997].

A metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) é um algoritmo simples e amplamente aplicado em problemas de otimização. Proposta por Feo & Resende [1995], trata-se de uma abordagem iterativa de multipartida, onde a cada iteração é gerada uma nova solução inicial viável, seguida de uma etapa de busca local.

A adaptação heurística apresentada nessa seção, chamada de GRASPF-RVNS, visa prover soluções de mapeamentos rápidas, com um tempo final máximo de t_c unidades de tempo por execução centralizada. No caso de execuções descentralizadas, o tempo de um mapeamento centralizado t_c é dividido aritmeticamente entre os domínios envolvidos, sendo expresso por $t_d = \frac{t_c}{|D|}$, onde t_d representa o tempo de execução máximo de cada domínio.

Na abordagem proposta nessa dissertação, o VNE é empregado no mapeamento de requisições geradas por aplicações que precisam de um tempo de resposta rápido, onde atrasos podem ser prejudiciais para o usuário final. Visando satisfazer a necessidade do baixo tempo de mapeamento, t_c foi definido em $300ms$. Mas no caso da virtualização ser aplicada em cenários onde as VNs não demandem um tempo de resposta baixo, esse tempo pode ser redefinido para valores maiores.

Na abordagem heurística proposta nessa seção, o algoritmo GRASPF é usado como um filtro que gera um conjunto de soluções durante t_1 ms , sendo sempre a melhor solução armazenada. Em uma etapa posterior, a melhor solução encontrada com o GRASPF é então submetida a uma busca local por mais t_2 ms . Caso nenhuma solução viável seja encontrada, a VN é rejeitada.

Definindo mais especificamente os tempos de execução de cada fase da heurística GRASPF-RVNS, t_1 possui um tempo de execução mais baixo que t_2 . Essa escolha parte da premissa que melhores soluções podem ser geradas ao se destinar mais tempo a busca local do que a geração da solução inicial. Tais tempos são definidos de acordo com a abordagem avaliada, sendo:

- Abordagem centralizada, $t_1 = \frac{t_c}{100} \cdot 20$ e $t_2 = \frac{t_c}{100} \cdot 80$;
- Abordagem descentralizada, $t_1 = \frac{t_d}{100} \cdot 20$ e $t_2 = \frac{t_d}{100} \cdot 80$.

Tal adaptação deixa o algoritmo mais rápido que a abordagem clássica GRASP, o que se encaixa melhor em propostas que necessitam de uma resposta rápida.

A busca local empregada, atua como um refinamento para a melhor solução encontrada por GRASPF. Para isso uma variação da metaheurística *Variable Neighborhood Search* (VNS) proposta por Mladenović & Hansen [1997] é utilizada. Tal variação é chamada de RVSN (*Reduced Variable Neighborhood Search*) e tem como característica não explorar todo espaço de soluções, e sim promover saltos nesse espaço, o que propicia em uma maior diversificação das vizinhanças analisadas. O uso de tal variação é motivada por reduzir significativamente o tempo de processamento em relação ao VNS tradicional, visto a grande quantidade de potenciais soluções vizinhas que o VNE possui a serem investigadas.

Uma das grandes dificuldades das técnicas heurísticas é fugir dos chamados ótimos locais. Em problemas de maximização, quando qualquer movimento de vizinhança feito decrementa o valor da função objetivo, este ponto é chamado de ótimo local. Um ótimo global corresponde ao maior valor que a função objetivo pode alcançar dentre todos os ótimos locais existentes em todo espaço de soluções. Uma exemplificação de ótimos globais, locais e vizinhança podem ser abstraídos e observados na Figura 4.1

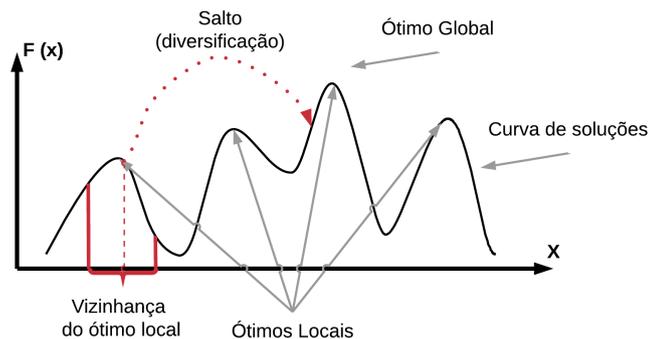


Figura 4.1: Exemplos de ótimo local, global, vizinhança e curva de soluções.

No VNE, amostrar todos os pontos sobre o espaço de soluções para definir todos os ótimos locais é custoso computacionalmente e pode requerer um número exponencial de movimentos. A abordagem GRASPF-RVNS utiliza de vizinhanças com características de alta diversificação, a fim de amostrar um número maior de diferentes soluções e tentar comparar o máximo de diferentes ótimos locais. Tal diversificação de vizinhança é explicada na seção 4.1.3.

A abordagem GRASPF-RVNS (Algoritmo 1) visa a geração de soluções de qualidade e rápidas. De maneira simplificada, tal abordagem é baseada em três etapas:

- Ordena-se o conjunto de VNs entrantes de acordo com a demanda total de recursos de enlaces virtuais de cada VN em ordem crescente.
- Para cada VN entrante, encontrar melhor solução através da GRASPF.
- Submeter a melhor solução encontrada a uma busca local.

Caso a abordagem seja *online*, o método começa a processar uma VN por vez, e a etapa de ordenação é nula (linha 1, Algoritmo 1). Caso a abordagem seja *offline* ou periódica, o método começa a processar o lote de VNs pelas menores, ordenadas na etapa anterior. Tal escolha é motivada pelo fato de que o mapeamento de VNs que consomem menos recursos implicarem em uma maior sobra de recursos residuais para os mapeamentos a *posteriori*.

Algoritmo 1 GRASPF-RVNS ($V[]$, α)

```

1: ordena( $V$ ) ,  $i = 0$ 
2: enquanto  $i < |V|$  faça
3:    $f^* = +\infty$ ;
4:   enquanto tempo  $< t_1$  faça
5:      $s' \leftarrow$  Construtivo ( $V[i]$ ,  $\alpha$ )   {fase de construção}
6:     se  $f(s') < f^*$  então
7:        $s^*$ ,  $f^* \leftarrow s'$ ,  $f(s')$ 
8:     fim se
9:   fim enquanto
10:  se  $f^* = +\infty$  então
11:     $V[i] = \text{Null}$  , falso   {GRASP falha de mapeamento}
12:  fim se
13:   $V[i] = \text{RVNS}(s^*)$  , verdadeiro   {busca local}
14:   $i++$ 
15: fim enquanto
16: devolve  $V[ ]$ 

```

Inicialmente, entre as linhas 4 e 9 do Algoritmo 1, a fase de construção é chamada para cada VN, onde é gerada uma sequência de soluções iniciais a serem comparadas, sendo sempre a melhor armazenada (linha 7, Algoritmo 1). Este modelo de iteração é efetuado por t_1 ms. Caso ao menos uma solução consiga ser gerada com sucesso nesta fase, a solução então é submetida a uma busca local (linha 13, Algoritmo 1). Caso nenhuma solução inicial consiga ser gerada, o algoritmo então rejeita a VN.

4.1.1 Heurística Construtiva

Conceitualmente o espaço de todas as soluções viáveis possíveis pode ser definido como espaço de busca do VNE (Figura 4.1). Em um algoritmo centralizado, todo o espaço de busca pode ser potencialmente analisado; em algoritmos descentralizados é como se esse espaço de busca fosse segmentado, onde cada domínio apenas enxerga os fragmentos de soluções limitados por suas fronteiras, o que reduz o espaço de busca.

Independente do caso avaliado, genericamente, o algoritmo proposto irá trabalhar nesse espaço de busca fragmentado ou não, a fim de procurar por uma solução viável e de boa qualidade, que pode esporadicamente atingir um ponto extremo de mínimo ou máximo global de solução.

A heurística construtiva apresentada é de natureza gulosa (Algoritmo 2), e dificilmente irá gerar um ótimo global, mas possui um tempo computacional viável, e por ser executada várias vezes (linha 3, Algoritmo 1), tende a amostrar um bom número de soluções a serem comparadas, o que potencialmente pode gerar boas soluções a serem

submetidas a busca local.

Algoritmo 2 Construtivo (G^v, α)

```

1: para cada  $n^v \in G^V$  faça
2:   se não mapeia( $n^v$ , random_elemento(LRC( $n^v$ ,  $\alpha$ ))) então
3:     devolve falso   {falha de mapeamento de nós}
4:   fim se
5: fim para
6: para cada  $e^v \in G^V$  faça
7:    $n_1^P, n_2^P \leftarrow 1^\circ, 2^\circ$  nós físicos do enlace  $e^v$ 
8:   se não BFS( $n_1^P, n_2^P$ ) então
9:     devolve falso   {Falha mapeamento de enlaces}
10:  fim se
11: fim para
12: devolve verdadeiro

```

O Algoritmo 2 apresenta a função construtiva do Algoritmo GRASPF (Algoritmo 1). Para cada VN entrante, é executada uma sequência de ações para se efetuar o mapeamento. Inicialmente é feito o mapeamento de nós (linha 2, Algoritmo 2), nesta etapa deve-se encontrar um conjunto de nós físicos que possuam capacidade de CPU e localidade disponíveis para o mapeamento. Para cada nó virtual deve-se fazer o mapeamento sobre um nó físico, sendo que um nó físico nunca deve hospedar mais de um nó virtual da mesma VN. Tal ação é feita com base em um parâmetro α .

O mapeamento de nós virtuais é realizado iterando-se por todos os nós virtuais da requisição (linhas 1 a 5 do Algoritmo 2). Sendo que para cada nó virtual, é gerada uma lista de nós físicos candidatos (LC) a mapeamento que respeitem a localização geográfica e a capacidade de CPU residual. A partir desta lista e do parâmetro recebido α , é criada a LRC (linha 2 do Algoritmo 2) com elementos que atendam os critérios da equação 4.1:

$$\{n \in LRC \mid h(n) \geq h_{max} - \alpha(h_{max} - h_{min})\} \quad (4.1)$$

$$h(n) = R(n) \sum_{e \in E \cup E^0} R(e) \quad (4.2)$$

Na equação 4.2, $R(n)$ representa a capacidade residual de CPU do nó físico n e $R(e)$ a capacidade residual de cada enlace e ligado ao nó n . A função $h(n)$ prioriza o mapeamento em nós físicos menos saturados em termos da sua capacidade de CPU e largura de banda nos enlaces incidentes. A partir da LRC formada, um nó físico é escolhido aleatoriamente para hospedar o nó virtual em questão (linha 2, Algoritmo

2). Caso a LRC esteja vazia, é porque não há posições de mapeamento viáveis, nesse caso retorna-se falso (linha 3, Algoritmo 2).

Caso o mapeamento de nós falhe, o algoritmo então retorna a falha (linha 3 do Algoritmo 2), o que pode ocasionar em uma completa rejeição da VN (linha 3, Algoritmo 2). Caso a etapa ocorra com sucesso, uma segunda etapa é iniciada, sendo responsável pelo mapeamento de todos os enlaces virtuais (linhas 7 a 10, Algoritmo 2). Neste caso, dado um mapeamento de nós válido, para cada enlace virtual da VN deve-se encontrar um caminho no SN que liga os nós físicos onde os nós do enlace virtual foram mapeados. Tal ação é feita através de uma busca em largura (BFS) (linha 8, Algoritmo 2). Caso algum enlace não consiga ser mapeado, o algoritmo retorna então uma falha de mapeamento de enlaces (linha 9, Algoritmo 2).

Na abordagem apresentada no algoritmo 2, não é necessário definir o tamanho da LRC, mas sim quais elementos fazem parte dela, para tal, é utilizado o parâmetro $\alpha \in [0, 1]$. Quanto mais próximo de 0 for o valor de α , menos elementos candidatos a LRC irá conter, e quanto mais próximo de 1 for o valor de α mais elementos a LRC irá conter.

4.1.2 Heurística de Refinamento

A heurística de refinamento (busca local) trabalha com o princípio de promover trocas sistemáticas sobre a solução inicial de maneira que a nova solução continue viável, mas em uma melhor posição no espaço de soluções permitido. Dessa maneira, geram-se mudanças no uso de recursos do SN. A busca local somente é executada caso o Algoritmo construtivo (Algoritmo 2) tenha conseguido chegar a um mapeamento inicial válido.

A busca local implementada usa os conceitos da heurística RVNS. Tal abordagem possui uma característica de variação da solução amostrada por meio de movimentos com certo grau de aleatoriedade. O RVNS tenta explorar a partir da solução inicial uma sequência de diferentes soluções vizinhas a fim de melhorar a solução corrente. Uma das características do RVNS é nunca gerar uma solução inviável, ou seja, mesmo que ela não obtenha sucesso em termos de melhoria da solução, na pior das hipóteses a mesma solução inicial é retornada. Tal abordagem é apresentada no Algoritmo 3.

A busca local proposta, trabalha com dois movimentos de trocas em vizinhança descritos na seção 4.1.3. A abordagem RVNS (Algoritmo 3) submete a solução corrente s à vizinhança N^1 . Se a solução vizinha achada for melhor que a corrente (linha 6, Algoritmo 3), a solução vizinha passa a ser a corrente e a vizinhança N^1 continua sendo

Algoritmo 3 RVNS(s_0)

```

1:  $s \leftarrow s_0$ ; {solução inicial}
2: enquanto  $tempo < t_2$  faça
3:    $k \leftarrow 1$ ;
4:   enquanto  $k \leq 2$  faça
5:      $s' \leftarrow \text{random\_vizinhança}N^k(s)$ ;
6:     se  $f(s') < f(s)$  então
7:        $s, k \leftarrow s', 1$ ;
8:     senão
9:        $k \leftarrow k + 1$ ;
10:    fim se
11:  fim enquanto
12: fim enquanto
13: devolve  $s$ 

```

explorada em outra iteração. Caso N^1 não gere melhora (linha 6, Algoritmo 3) k é incrementado (linha 9, Algoritmo 3) e começa a ser explorada a vizinhança N^2 . Caso N^2 gere melhora, a solução achada passa a ser a solução corrente e volta-se a explorar a vizinhança N^1 . Essa sequência é executada até que o tempo limite t_2 tenha decorrido. Ao final, a abordagem retorna a melhor solução encontrada (s), ou caso não tenha ocorrido nenhuma melhora retorna-se a solução inicial (linha 13, Algoritmo 3). Na abordagem proposta, N^1 tende a ser executada mais vezes, tal opção é fundamentada no fato de ser uma vizinhança de execução mais leve, o que implica em uma amostragem maior de soluções, visto que o método possui uma limitação de tempo.

4.1.3 Estruturas de Vizinhança

Como definição de vizinhança adotamos que dada uma solução qualquer, uma vizinhança é gerada em função dos diferentes movimentos que podem ser feitos para modificar a solução avaliada. Alguns movimentos podem analisar soluções mais próximas, e outras soluções mais distantes da encontrada até o momento.

Na literatura existem diferentes tipos de vizinhanças que variam entre as diferentes abordagens existentes para o VNE. Para a aplicação do RVNS, foram desenvolvidas duas estruturas de vizinhanças diferentes: N^1 que promove somente a troca de um enlace físico inicialmente usado; e N^2 que promove a troca de um roteador físico pertencente à solução examinada, e conseqüentemente a troca de todos enlaces antes incidentes sobre o roteador alterado.

Mais especificamente, o movimento de vizinhança N^1 ocorre com a remoção do mapeamento de um enlace virtual qualquer $(k, l) \in E^v$ sobre o respectivo enlace físico

$(i, j) \in E \cup E^0$ usado na solução corrente (s') de determinada VN. Tal ação proíbe momentaneamente que o enlace (i, j) seja usado na nova vizinhança investigada, e força sua possível troca por outro enlace $(i', j') \in E \cup E^0$ que respeite as restrições 3.4, 3.5 e 3.6. Repare que para a formação da solução vizinha (s'') a troca do enlace $(i, j) \in E \cup E^0$ pode implicar na troca de demais enlaces que compõem o caminho, ou pode ser necessário a formação de um caminho totalmente diferente do inicial. Tal movimento pode gerar várias soluções vizinhas, um exemplo de vizinhança pode ser visualizada na Figura 4.2.

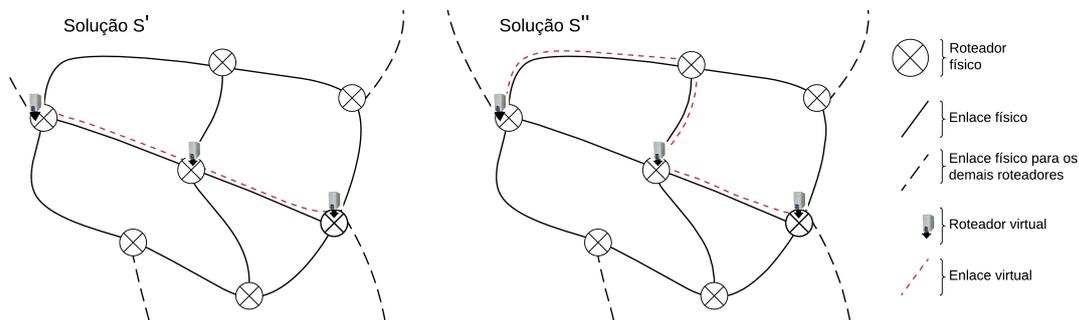


Figura 4.2: Representação de vizinhança N^1 , a esquerda solução s' , a direita solução s'' .

Dada sua simplicidade de movimento, a vizinhança N^1 pode potencialmente gerar melhoras em termos de função objetivo, mas devido a baixa diversificação aplicada no movimento de vizinhança, a solução gerada pode eventualmente ficar presa em uma região de ótimo local. Por esse motivo a vizinhança N^2 é proposta com um movimento mais denso de vizinhança.

O movimento de vizinhança N^2 é mais complexo e computacionalmente custoso que N^1 . Tal movimento implica na troca de um roteador físico $i \in N$ que faz parte da solução corrente (s') de determinada VN por outro não usado e que respeite além do raio de mapeamento as restrições 3.1, 3.2 e 3.3. Essa ação pode acarretar na necessidade de um novo roteamento de todos os enlaces que incidiam inicialmente sobre i , sendo que este novo roteamento deve respeitar as restrições 3.4, 3.5 e 3.6. Similarmente a N^1 , tal movimento pode gerar inúmeras soluções vizinhas, um exemplo de vizinhança pode ser visualizada na Figura 4.3.

Dada sua complexidade de movimento, a vizinhança N^2 pode potencialmente não gerar melhoras em termos de função objetivo, ou pior, não gerar soluções viáveis ao problema, se for o caso de um SN com pouca oferta de CPU nos roteadores físicos. Nesse aspecto, outras vizinhanças alternativas intermediárias entre N^1 e N^2 podem ser cogitadas, implementadas e testadas em trabalhos futuros.

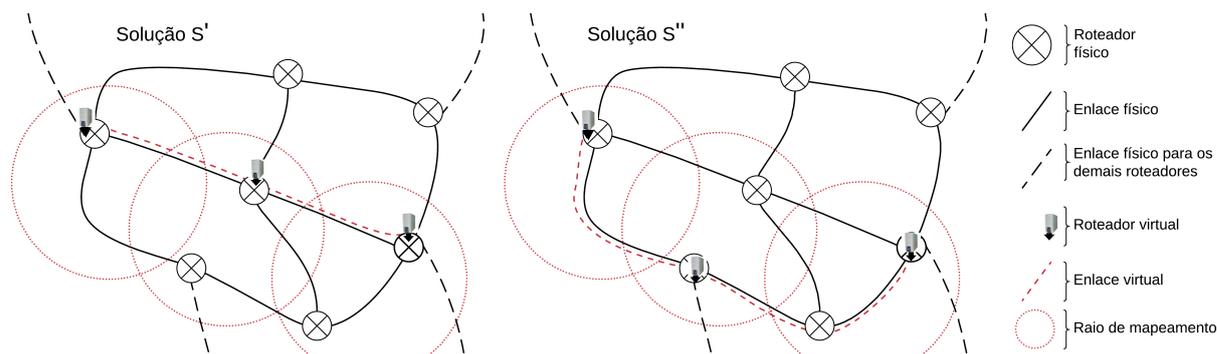


Figura 4.3: Vizinhança N^2 , a esquerda solução s' , a direita solução s'' .

4.2 Orquestração para o VNEMD utilizando a abordagem heurística

Dado o desafio de atuar em um ambiente MD, esta seção apresenta uma abstração de unidade de controle chamada de orquestrador, que trabalha junto à heurística GRASPF-RVNS. Em nossa abordagem, orquestrador assume um papel de intermediação na contratação dos serviços dos ISPs pelos SPs em ambientes MD. São propostos três modelos de orquestração distintos: com conhecimento total (seção 4.2.1), parcial (seção 4.2.2) e sem conhecimento da infraestrutura de rede física interna aos domínios (seção 4.2.3). O desenvolvimento do orquestrador utilizando a heurística se dá justamente por em ambientes de redes reais a resolução do VNE de maneira descentralizada ser uma necessidade, e a escolha de não usar um modelo de resolução exato e sim a heurística é motivada pelo fato de abordagens exatas serem ineficientes computacionalmente para instâncias de grande porte. No capítulo 2 o conceito de coordenação era relacionado com a estruturação interna do algoritmo de mapeamento. Diferentemente, nesta seção coordenação se refere à maneira de interação do orquestrador com os domínios de rede.

4.2.1 Abordagem com conhecimento total

Neste modelo, o orquestrador faz uma consulta inicial a cada domínio existente e recebe os dados de toda topologia pertencente ao SN (Tabela 4.1, mensagens 10 e 11), o que inclui informações de localização de todos os roteadores físicos pertencentes aos domínios, seus respectivos enlaces, capacidades e consumo de energia. Em cenários reais de redes tradicionais, cada domínio tende a esconder o máximo de informações sobre sua topologia interna; logo, esta abordagem é aplicada apenas com intuito de comparação de resultados.

Nesta abordagem, o orquestrador possui autonomia global para a otimização, onde o algoritmo GRASPF-RVNS é aplicado em todo SN de maneira coordenada e centralizada. Como o orquestrador conhece o estado de todo o SN, as trocas de mensagens são geradas somente para a efetivação de um mapeamento (Tabela 4.1, mensagem 7). Como o orquestrador é quem define o mapeamento, o próprio atualiza internamente o estado *a posteriori* do SN sem a necessidade de demais consultas. Esse modelo é apresentado na Figura 4.4.

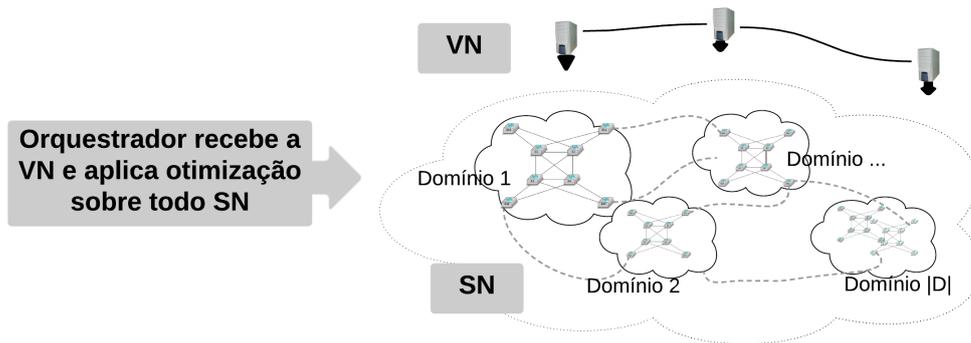


Figura 4.4: Orquestrador com autonomia de otimização global, com conhecimento total do SN.

4.2.2 Abordagem com conhecimento parcial do SN

Este modelo se caracteriza pelo compartilhamento do nível de saturação dos domínios existentes com o orquestrador (Tabela 4.1, mensagem 9). Deste modo, o orquestrador possui a informação de qual domínio possui as capacidades mais próximas do esgotamento e quais estão mais aptos a efetivar um mapeamento. Tal nível de saturação é mensurado pela equação 4.3:

$$\text{Saturação do domínio} = \frac{\sum_{i \in N^d} \bar{C}_i \cdot \sum_{(i,j) \in E^d \cup E^0} \bar{B}_{ij}}{|N^d|} \quad (4.3)$$

Inicialmente o orquestrador envia uma mensagem de consulta sobre o nível de saturação de cada domínio (Tabela 4.1, mensagem 9). Após, o orquestrador fica responsável por repassar o mapeamento da VN para o domínio que possui menor nível de saturação de seus componentes (Tabela 4.1, mensagem 7). Caso o domínio envolvido não tenha capacidade para mapear algum componente requerido pela VN, ele pode se comunicar com os domínios adjacentes através da comunicação inter-domínio para requerer o mapeamento do componente faltante (Tabela 4.1, mensagens 4, 5 e 6). Somente se o domínio escolhido responder pela completa rejeição do mapeamento (Tabela 4.1, mensagem 3), o orquestrador tenta realizar o mapeamento com o segundo domínio

com menor nível de saturação. Assim ocorre sucessivamente até a VN ser mapeada ou até que não haja mais domínios a serem consultados. Ao final, caso algum domínio tenha recebido o mapeamento de algum componente, este deve atualizar junto ao orquestrador o novo nível de saturação do domínio envolvido (Tabela 4.1, mensagem 9). Neste caso a coordenação das informações para o mapeamento é feita pelo orquestrador, e o mapeamento é feito no domínio. Cada domínio aplica a otimização dentro de suas fronteiras, de maneira local. Este modelo pode ser exemplificado na Figura 4.5.

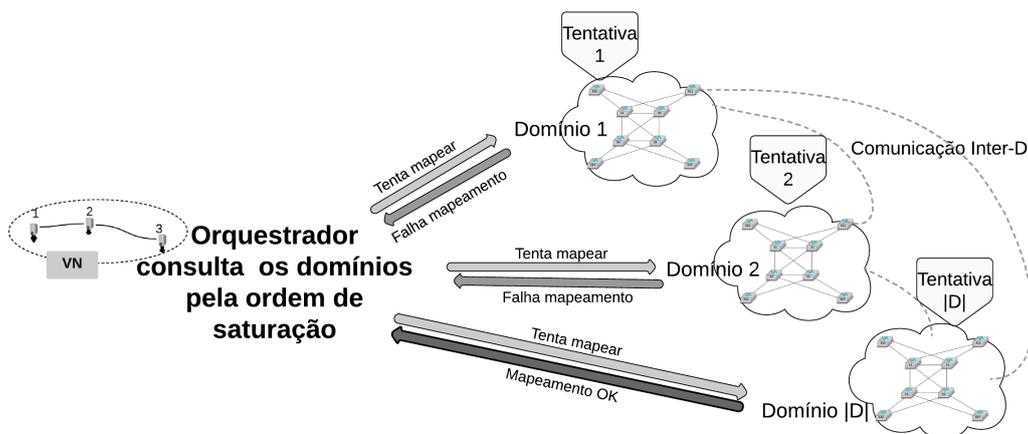


Figura 4.5: Orquestrador tenta efetuar o mapeamento em ordem crescente de saturação dos domínios. Neste exemplo, o mapeamento somente é aceito na última tentativa.

4.2.3 Abordagem sem conhecimento do SN

Diferentemente dos modelos anteriores, neste caso o orquestrador não possui nenhuma informação interna aos domínios envolvidos. É de responsabilidade do orquestrador consultar cada domínio se o mesmo consegue prover o mapeamento e qual é o valor da função objetivo envolvida (Tabela 4.1, mensagens 1, 2 e 3). Ao final contrata-se o serviço do domínio mais vantajoso (Tabela 4.1, mensagens 7 e 8).

Deste modo cada domínio é responsável por mapear e otimizar localmente a VN que lhe foi requerida. Caso por alguma inviabilidade local algum componente da VN não possa ser mapeado, o domínio pode requisitar a um domínio adjacente que mapeie o componente faltante (Tabela 4.1, mensagens 4, 5 e 6). Nesse caso se configura uma troca de mensagens entre os dois domínios envolvidos. Tal modelo é representado pela Figura 4.6.

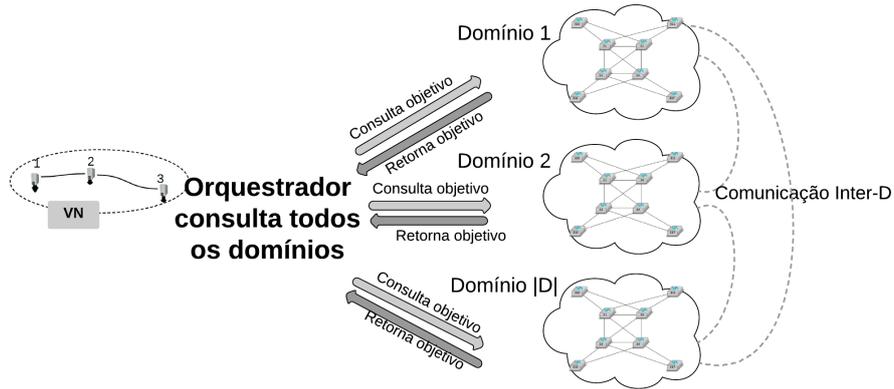


Figura 4.6: O orquestrador consulta todos os domínios existentes e escolhe o que oferecer as melhores condições de mapeamento.

4.2.4 Mensagens de Coordenação

Sempre que o orquestrador se comunica com algum domínio ou algum domínio se comunica com outro domínio é gerada uma mensagem de pergunta, resposta ou ordem. Tais mensagens são exemplificadas na Tabela 4.1.

Sejam as seguintes abordagens:

- $VNEO_{ct}$, abordagem heurística junto ao orquestrador com conhecimento total do SN (seção 4.2.1).
- $VNEO_{cp}$, abordagem heurística junto ao orquestrador com conhecimento parcial do SN (seção 4.2.2).
- $VNEO_{sc}$, abordagem heurística junto ao orquestrador sem conhecimento do SN (seção 4.2.3).

Tabela 4.1: Tabela de mensagens

ID	Tipo	Emissor	Receptor	Conteúdo	Abordagem
1	Pergunta	orquestrador	domínio d	VN	$VNEO_{sc}$
2	Resposta	domínio d	orquestrador	OK, valor objetivo	$VNEO_{sc}$
3	Resposta	domínio d	orquestrador	falha	$VNEO_{cp}, VNEO_{sc}$
4	Pergunta	domínio d'	domínio d''	roteador u , roteador v	$VNEO_{cp}, VNEO_{sc}$
5	Resposta	domínio d''	domínio d'	OK, caminho, valor objetivo	$VNEO_{cp}, VNEO_{sc}$
6	Resposta	domínio d''	domínio d'	falha	$VNEO_{cp}, VNEO_{sc}$
7	Ordem	orquestrador	domínio d	OK, VN	$VNEO_{ct}, VNEO_{cp}, VNEO_{sc}$
8	Ordem	domínio d'	domínio d''	OK, u, v , caminho	$VNEO_{cp}, VNEO_{sc}$
9	resposta	domínio d'	orquestrador	$h(x)$	$VNEO_{cp}$
10	Pergunta	orquestrador	domínio d'	$G^d(N^d, E^d)$	$VNEO_{ct}$
11	resposta	domínio d'	orquestrador	$G^d(N^d, E^d)$	$VNEO_{ct}$

1. Mensagem enviada do orquestrador para o domínio $d \in D$, onde é requisitado o valor objetivo para atender uma VN.

2. Mensagem enviada do domínio d ao orquestrador, onde é devolvido o valor objetivo envolvido de atender a VN com sucesso.
3. Mensagem enviada do domínio d ao orquestrador, onde é reportado que não consegue atender a VN.
4. Mensagem enviada do domínio d' para o domínio adjacente d'' , onde é requerido um roteador físico que atenda às restrições do roteador virtual u e que possua um caminho com banda mínima b_{uv} até o roteador de fronteira v .
5. Mensagem enviada do domínio d'' para o domínio d' , onde é respondida com sucesso a solicitação de busca de roteador físico e devolvendo a rota estabelecida entre origem de fronteira v e roteador físico u e o respectivo valor envolvido.
6. Mensagem enviada do domínio d'' para o domínio d' , onde é respondida a falha no atendimento do roteador físico requisitado.
7. Mensagem enviada do orquestrador para o domínio $d \in D$, onde é requisitado que o mesmo efetive o mapeamento definido *a priori*.
8. Mensagem enviada do domínio d' para o domínio d'' , para que o mesmo efetive o mapeamento do componente u , no caminho previamente consultado.
9. Mensagem enviada do domínio $d \in D$ para o orquestrador, onde é reportada a saturação interna dos componentes do domínio d (equação 4.3).
10. Mensagem enviada do orquestrador para o domínio $d \in D$, onde é consultada a estrutura topológica interna ao domínio d .
11. Mensagem enviada do domínio $d \in D$ para o orquestrador, onde é reportada a estrutura topológica do domínio d .

Capítulo 5

Experimentos Computacionais

Neste capítulo, inicialmente são apresentadas as métricas de avaliação usadas bem como as características de implementação do simulador. Posteriormente são apresentadas as particularidades das instâncias geradas e as abordagens avaliadas. Ao final é apresentada uma análise de desempenho das abordagens avaliadas nos diferentes cenários de teste.

5.1 Métricas de Avaliação

De maneira a confrontar as abordagens propostas, é necessário estabelecer um conjunto de medidas quantitativas para a avaliação dos resultados obtidos. Essa seção formaliza as métricas utilizadas para a comparação das abordagens testadas.

A primeira métrica avaliada é a aceitação total (AT) das VNs. Esta métrica é dada pelo número de VNs aceitas durante toda execução da abordagem. Seja V o conjunto de VNs que chegaram e y^v a variável binária que indica se a requisição v foi atendida ou não. Tem-se que:

$$AT = \sum_{v \in V} y^v \quad (5.1)$$

Quando uma VN é mapeada no modelo periódico, existe um atraso do momento que a VN é requisitada até o tempo em que ela é atendida. Seja t_e^v o tempo de chegada e t_m^v o tempo de conclusão do mapeamento de determinada VN $v \in V$, têm-se o atraso médio (AM) de mapeamento definido por:

$$AM = \frac{\sum_{v \in V} y^v \cdot (t_m^v - t_e^v)}{\sum_{v \in V} y^v} \quad (5.2)$$

O tempo real de processamento é um aspecto importante quando lidamos com problemas que exigem uma resposta rápida. Sendo o simulador implementado em C++, para aferição do tempo de processamento de cada VN foi utilizada a biblioteca padrão da linguagem de programação C (`time.h`), onde a contagem de tempo foi feita sempre antes e depois de se iniciar o método proposto. No caso das abordagens que usam CPLEX, a mesma estratégia foi usada, sendo o CPLEX alterado para não usar paralelismo.

Uma métrica importante tanto do ponto de vista dos domínios quanto do meio ambiente é o consumo de energia elétrica. Quanto menor o consumo de energia maior é o lucro dos domínios, pois terá uma despesa menor com o gasto de energia elétrica. Pelo lado ambiental tal métrica é importante para promover a chamada consciência verde, que prega o uso racional de energia elétrica. Cada VN mapeada com sucesso gera um consumo de energia elétrica enquanto permanece ativa, nesse caso esse consumo é aferido pela equação 3.11, já apresentada na seção 3.2. Dado o consumo de energia elétrica gerado por cada VN e a receita (equação 3.12), é possível estimar o lucro dos domínios com base na receita menos o custo de energia elétrica, nesse caso o lucro é aferido pela equação 3.13, apresentada na seção 3.2.

Sempre que uma VN é mapeada em mais de um domínio de rede é gerada uma carga de mensagens de coordenação e comunicação. A redução da geração dessas mensagens é importante, pois existe um consumo de banda resultante acarretado pela comunicação inter-domínio. Esse *overhead* de mensagens é contabilizado então em dois momentos:

- Na etapa de definição do mapeamento, existe o *overhead* de coordenação gerado pelas mensagens mostradas na Tabela 4.1
- Durante o tempo de vida de determinada VN (t^v), se o mapeamento fizer uso de enlaces inter-D, esse possui um consumo de banda extra, reservado para a comunicação entre os domínios adjacentes, definidos na equação 3.14.

5.2 O simulador

Segundo Calheiros et al. [2011], a realização de testes de virtualização em ambientes de redes reais é inviável mediante aos altos custos financeiros envolvidos, as diferentes configurações necessárias e a dificuldade de se controlar as condições dos experimentos realizados. Nesses tipos de teste é indicado então o uso de simuladores que podem em uma única máquina realizar experimentos controlados, simulando vários roteadores e enlaces, ajudando a validar as abordagens e as técnicas de orquestração.

Para a realização dos testes foi desenvolvido um simulador para testar as abordagens propostas. O simulador não gera as instâncias, para tal são implementados modelos de geradores da literatura (seção 5.3). Assim, cada conjunto de instâncias construído é submetido para a execução junto as diferentes abordagens pelo simulador.

A linguagem definida para implementação do simulador foi *C++* por ser uma linguagem flexível, que permite que se utilize a programação procedural estruturada com bom desempenho em termos de tempo de compilação e execução. As abordagens exatas são modeladas e executadas com uso do solver CPLEX e *C++*.

5.3 As instâncias

Para os testes da seção 5.5, os dados de entrada do simulador são gerados usando o modelo de Waxman [1988] e alguns princípios do artigo de Fischer & de Meer [2016].

Para a formação do SN, cada domínio de rede físico é criado com base no modelo de Waxman [1988]. Na bateria de cenários mostrada a seguir cada domínio possui entre 3 e 6 enlaces de comunicação inter-domínio com cada um dos outros domínios adjacentes. Logo não existem domínios isolados. Os demais parâmetros são mostrados na Tabela 5.1.

Tabela 5.1: Parâmetros para geração do *SN*

Parâmetro	SN
Número de domínios	3
Número de nós ($ N $)	25 por domínio
<i>Grid</i> coordenadas (x, y)	50×50
Localização nós (x_i, y_i)	randômica
α (parâmetro Waxman)	0.5
β (parâmetro Waxman)	0.3
CPU (C_i)	Uniformemente distribuído entre 50 e 100
Banda (B_{ij})	Uniformemente distribuído entre 50 e 100

As simulações consideram 10.000 e 50.000 unidades de tempo (t) respectivamente para os testes das seções 5.5.2 e 5.5.3. Com base nos SN criados, são geradas requisições seguindo os conceitos de Fischer & de Meer [2016], onde garante-se que se as VNs mapeadas em uma posição específica, elas são 100% mapeáveis, consumindo 100% dos recursos disponíveis (cenário *SN1*). Com base nesse aspecto são criadas duas outras variações de cenários, no primeiro existe uma sobra garantida de 25% de recursos físicos em cada componente do SN (cenário *SN2*), e no outro cenário existe a falta de recursos garantida de 25% para cada componente (cenário *SN3*). Ainda é gerado um quarto cenário (*SN4*) usando os parâmetros da Tabela 5.1, mas com apenas um domínio de

rede contendo 100 nós e considerando $50.000t$. o $SN4$ é usado para alguns testes preliminares centralizados e *online* (seção 5.5.1).

A partir do conjunto de estruturas físicas, são gerados três cenários de requisições para os testes, chamados de $R1$, $R2$ e $R3$.

- Em $R1$ as VNs possuem uma taxa de chegada de 50 VNs ao longo das $10.000t$, este cenário é aplicado para os testes da seção 5.5.2, onde são analisadas uma transição entre os modelos *online*, periódico e *offline*. $R1$ se caracteriza por as VNs possuírem um tempo de vida maior que $R2$ e $R3$, onde em algum momento, todas as VNs entrantes compartilham a mesma fatia de tempo.
- Em $R2$ as VNs possuem uma taxa de chegada de 250 VNs ao longo das $50.000t$, este cenário é aplicado nos testes das seções 5.5.1 e 5.5.3, onde as VNs são analisadas no modelo *online*. $R2$ se caracteriza por as VN possuírem um tempo de vida maior que $R3$, o que acarreta em uma menor taxa de chegada de requisições.
- Em $R3$ as VNs possuem uma taxa de chegada de 2.000 VNs ao longo das $50.000t$, este cenário é aplicado nos testes das seções 5.5.1 e 5.5.3, onde as VNs são analisadas no modelo *online*. $R3$ se caracteriza por as VN possuírem um tempo de vida menor que $R2$, o que acarreta em uma maior taxa de chegada de requisições.

Alguns trabalhos da literatura como Zhu & Ammar [2006] e Yu et al. [2011] atribuem a uma VN um tempo de vida baixo, com um valor seguindo uma distribuição exponencial de $\mu = 10t$. Outros trabalhos atribuem um tempo de vida maior, como Chowdhury et al. [2009] onde o tempo de vida segue uma distribuição exponencial com $\mu = 1000t$. Tal valor de duração de uma VN depende da aplicação que requer o mapeamento. Nessa dissertação adotamos uma linha de tempos de duração altos similares à adoção de Chowdhury et al. [2009]. De modo geral, os parâmetros usados nas gerações das requisições foram extraídos de Chowdhury et al. [2009] e Fischer [2016], sendo mostrados na Tabela 5.2.

Tabela 5.2: Parâmetros para geração das VNs

Parâmetro	VNs
Número de VNs	50 ($R1$) / 250 ($R2$) / 2.000 ($R3$)
Número de nós de cada VN	2 – 12
Grid coordenadas (x_k, y_k)	50×50
Raio (d_{max}^v)	distribuição Poisson com $\mu = 15$
Taxa de chegada	distribuição Poisson $\mu = 200t$ ($R1$) / $\mu = 200t$ ($R2$) / $\mu = 25t$ ($R3$)
Tempo de vida (t^v)	distribuição exponencial $\mu = 15.000$ ($R1$) / $\mu = 10.000$ ($R2$) / $\mu = 1.000t$ ($R3$)
CPU (c_k^v)	Uniformemente distribuído entre 0.001 e 20
Banda (b_{kl}^v)	Uniformemente distribuído entre 0.001 e 50

Existem diversos fatores envolvidos no consumo de energia elétrica de cada componente pertencente à rede física, o que implica em uma variação dos custos de cada domínio, como: resfriamento de componentes, instalação de módulos, amplificadores, número de portas, tipo de cabeamento, roteadores etc. Abstraindo dessa grande variação de custos, neste trabalho consideramos um modelo simplificado de consumo de energia, com valores embasados em equipamentos de mercado. As tarifações aplicadas pelos ISPs também são baseadas em valores reais. O custo de energia aplicado foi de $R\$0.79 / kWh$, os valores cobrados para o roteamento são $\kappa = R\$ 0.047$ e $\varphi = R\$ 0.022$, o *overhead* de comunicação é estipulado em 10% da banda passante nos respectivos enlaces inter-domínio. Os demais parâmetros adotados nos componentes intra-domínio e inter-domínio são mostrados nas Tabelas 5.3 e 5.4 respectivamente.

Tabela 5.3: Parâmetros inter-domínio

G_d^P	Γ^d	γ^d	Δ^d	δ^d
G_1^P	0.50	0.043	0.03	0.0030
G_2^P	0.90	0.044	0.04	0.0045
G_3^P	0.53	0.056	0.05	0.0050

Tabela 5.4: Parâmetros intra-domínio

E_{de}^P	Ω^{de}	ω^{de}
$E_{1,2}^P$	0.45	0.045
$E_{1,3}^P$	0.50	0.065
$E_{2,3}^P$	0.30	0.021

5.4 Abordagens Avaliadas

Os experimentos computacionais realizados são feitos com 3 variações da modelagem de PLI (seção 3.2), 3 variações da heurística (seção 4.1) e 3 variações do orquestrador junto a heurística (seção 4.2) e 1 abordagem da literatura, sendo:

- **Abordagens em PLI**, possuem tempo de execução máximo de 1800s (segundos), independente do número de chamadas do solver. Usam o CPLEX com o paralelismo de dados desabilitado, pode variar em:
 - *PLI_lb* -Maximiza o balanceamento de carga (equação 3.7).
 - *PLI_lr* - Maximiza o lucro (equação 3.13).
 - *PLI_ms* - Minimiza o *overhead* de comunicação (equação 3.14).
- **Abordagens GRASPF-RVNS**, possuem tempo de execução total máximo de 300ms por requisição, sendo $t_1 = 60ms$ e $t_2 = 240ms$, pode variar em:
 - *GR_RVNS_lb* -Maximiza o balanceamento de carga (equação 3.7).
 - *GR_RVNS_lr* - Maximiza o lucro (equação 3.13).

- *GR_RVNS_ms* - Minimiza o *overhead* de comunicação (equação 3.14).
- ***D-ViNE-LB*** - Algoritmo da literatura [Chowdhury et al., 2009]. Prioriza o balanceamento de carga em uma variação que não permite separação em múltiplos caminhos. Usa o *GNU Linear Programming Kit* (GLPK) sem paralelismo de dados.
- **Orquestrador com GRASPF-RVNS**, utiliza a heurística GRASPF-RVNS que maximiza o lucro (equação 3.13) junto ao orquestrador e com tempo de execução total máximo de $300ms$ por requisição, pode variar em:
 - *VNEO_ct* - Orquestração com conhecimento total do SN (seção 4.2.1). Os tempos das fases de execução são fracionados em $t_1 = 60ms$ e $t_2 = 240ms$.
 - *VNEO_cp* - Orquestração com conhecimento parcial do SN (seção 4.2.2). Como os testes são realizados em 3 domínios, os tempos das fases de execução são fracionados em $100ms$ por domínio para a otimização local, onde $t_1 = 20ms$ e $t_2 = 80ms$.
 - *VNEO_sc* - Orquestração sem conhecimento do SN (seção 4.2.3). Como os testes são realizados em 3 domínios, os tempos das fases de execução são fracionados em $100ms$ por domínio para a otimização local, onde $t_1 = 20ms$ e $t_2 = 80ms$.

A métrica escolhida para ser otimizada junto ao orquestrador foi o lucro dos ISPs (equação 3.13). Tal fato é motivado justamente por esta métrica trabalhar com a minimização de energia conjuntamente a maximização do lucro, aspectos pertinentes as preocupações sócioambientais atuais e que têm sido alvo de um grande número de pesquisas, como visto nos capítulos 1 e 2.

O tempo de execução das heurísticas foi limitado em $300ms$ pois nessa dissertação são tratadas aplicações com a necessidade de conexões rápidas, como acontece frequentemente em aplicações envolvendo um consumidor final de internet. Mas caso não seja necessário um tempo de resposta baixo, esse número pode ser alterado para valores maiores. Outro aspecto relacionado ao valor de execução máximo ser definido em $300ms$ para as heurísticas, está no fato de ser similar ao tempo gasto pela abordagem exata em instâncias pequenas e em um tratamento *online*. A premissa é que aumentando o número de restrições e variáveis do problema o tempo gasto pela abordagem exata vai aumentar consideravelmente, dado a característica exponencial do problema, enquanto a heurística vai continuar com um tempo de execução baixo.

O trabalho de Alkmim et al. [2011] atribui um tempo máximo de execução de 60s nos modelos exatos e relaxados para cenários com 25 nós no SN. A abordagem exata aplicada nessa dissertação opta por um tempo de execução mais alto em relação a Alkmim et al. [2011], sendo de 1800s. Tal tempo de 1800s é baseado no fato das instâncias aqui trabalhadas serem maiores que em Alkmim et al. [2011], e por um tempo maior de execução tender a gerar melhores resultados para serem usados como limites comparativos junto à heurística.

5.5 Análise de Desempenho

Nesta seção são mostrados os resultados computacionais obtidos através das abordagens propostas, bem como os cenários investigados. Os testes foram realizados em um computador *Pentium* G870 3.30GHz, com 16GB de RAM DDR3 1333MHz, utilizando o sistema operacional Ubuntu 14.04.2 LTS. As abordagens não determinísticas foram testadas 20 vezes e os dados mostrados com um intervalo de confiança de 95%.

Baseado em alguns testes iniciais com o objetivo de maximizar o balanceamento de carga em cenários SD, o valor de α (equação 4.1) foi definido em 0.6 por prover um número bom de soluções a serem analisadas. Uma das características inerentes ao valor atribuído a α , é que tal escolha pode deixar o algoritmo completamente guloso, quando α se aproxima de 0 ou completamente aleatório, quando α se aproxima de 1. Se $\alpha = 0$ a LRC vai conter somente 1 elemento (h_{max}) e se $\alpha = 1$ a LRC vai conter todos os elementos da LC. Em testes iniciais a escolha de $\alpha = 0.6$ gerou uma maior aceitação final em relação a outros valores, o que motiva a manter esse mesmo valor de parâmetro para os testes apresentados nas seções 5.5.1, 5.5.2 e 5.5.3.

5.5.1 Análises preliminares em cenário SD e *online*

Esta seção faz uma comparação preliminar das abordagens propostas junto ao algoritmo da literatura *D-ViNE-LB*. As abordagens escolhidas para a comparação são a *PLI_lb* e a *GR_RVNS_lb* por ambas proporem maximizar o balanceamento de carga assim como o *D-ViNE-LB*. Os testes preliminares são aplicados no cenário *online* e SD, justamente por *D-ViNE-LB* não atuar em ambientes MD e em abordagens periódicas ou *offline*. Nesta seção, a limitação de tempo de 1800ms para a abordagem *PLI_lb* não foi aplicada, justamente para se ter conhecimento do gasto de tempo total até o resultado ótimo de cada mapeamento.

Os gráficos mostrados nas Figuras 5.1(a) e 5.1(b) mostram a aceitação total final das abordagens *PLI_lb*, *GR_RVNS_lb* e *D-ViNE-LB* nos cenários de baixa (*R2*) e

alta ($R3$) rotatividade de VNs. O SN_4 foi criado com uma disponibilidade de recursos exatos para um mapeamento perfeito, não havendo recursos físicos em abundância. Como percebido, tal fato gera uma aceitação total reduzida, o que indica que todas as abordagens tiveram o desempenho agravado por se tratar de um ambiente *online*.

Em ambos testes, Figuras 5.1(a) e 5.1(b), a abordagem exata consegue o melhor desempenho em termos de aceitação, mas ainda fica longe do mapeamento ótimo de 250 e 2000VNs respectivamente, que provavelmente só consegue ser achado com um tratamento *offline*. Tal fato deixa em evidência a dificuldade enfrentada pelas abordagens *online*. A baixa dispersão dos dados com intervalo de 95% de confiança demonstra a robustez da abordagem GR_RVNS_lb .

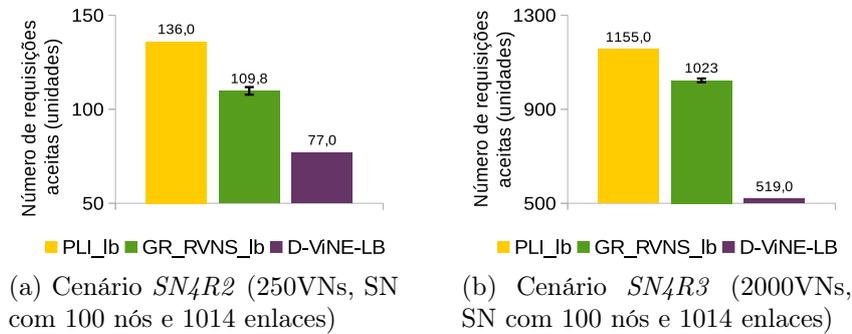


Figura 5.1: Número de requisições aceitas (equação 5.1)

Para os testes em ambientes SD a heurística GR_RVNS_lb conseguiu um bom desempenho em termos de aceitação, tendo números relativamente próximos a abordagem exata PLI_lb , mas tal proximidade não se repete nos testes em ambientes MD, conforme mostrado na seção 5.5.2. O que indica que a abordagem GR_RVNS_lb deve ser melhorada para ambientes MD.

Outro ponto que deve ser ressaltado nas Figuras 5.1(a) e 5.1(b) é a baixa aceitação atingida pela abordagem $D-ViNE-LB$. Fato sugere que o uso do modelo relaxado proposto por Chowdhury et al. não possui um bom desempenho quando o algoritmo é aplicado em um cenário de exata disponibilidade de recursos. As abordagens PLI_lb e GR_RVNS_lb conseguiram uma taxa de mapeamento constante, tanto no cenário de alta rotatividade de VNs (SN_4R2 , Figura 5.1(b)) quanto no de baixa (SN_4R2 , Figura 5.1(a)). Diferentemente, a abordagem $D-ViNE-LB$ possui uma queda de rendimento no cenário de alta rotatividade de VNs (SN_4R2 , Figura 5.1(b)) em relação ao de baixa (SN_4R1 , Figura 5.1(a)).

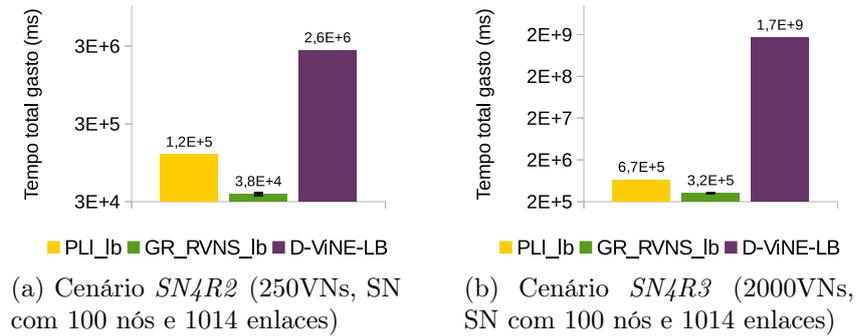


Figura 5.2: Tempo total de execução

Os testes mostrados nas Figuras 5.2(a) e 5.2(b) apresentam o tempo total gasto pelas abordagens para a execução dos cenários *SN4R2* e *SN4R3*. O tempo de execução da abordagem *D-ViNE-LB* foi muito alto, ultrapassando fortemente os tempos das outras duas abordagens em ambos casos, o que junto à baixa aceitação apresentada nas Figuras 5.1(a) e 5.1(b), deixa a abordagem impraticável para ambientes reais. A abordagem *GR_RVNS_lb* foi a que melhor se saiu neste quesito, executando em ambos casos sempre abaixo do tempo máximo de $300ms$ por requisição. Tal fato ocorre devido às VNs rejeitadas não efetuarem a etapa de busca local. A abordagem *PLI_lb* possui um tempo de execução alto, fator que motiva os próximos testes (seção 5.5.2) a serem executados com um limite de tempo e em cenários relativamente menores.

5.5.2 Análises *offline*, *online* e periódica

Devido à complexidade exponencial do problema (seção 1.2.3), algumas execuções do modelo PLI não conseguiram chegar ao resultado ótimo dentro do tempo máximo de execução estabelecido ($1800s$). Mesmo assim os resultados são mostrados a fim de fornecer dados para comparação. As execuções que não chegaram ao ótimo são destacadas com um x e podem ser visualizadas nas Figuras 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13 e 5.14.

Os testes das Figuras 5.3, 5.4 e 5.5 demonstram a evolução da aceitação final das abordagens, junto à variação do tamanho dos lotes de VNs analisados ($|V|$), definidos com base na abertura e fechamento das janelas de tempo. Para os casos onde a janela de tempo $t = 1$, tem-se $|V| = 1$ e o modelo executa a abordagem *online*. Para os casos *offline*, tem-se $|V| = 50$ e todas as VNs são processadas juntas. Por fim, os casos intermediários caracterizam as abordagens periódicas.

Quando o abordado é a aceitação total, Figuras 5.3, 5.4 e 5.5, percebe-se que para as abordagens baseadas em *PLI* e GRASPF-RVNS, quanto maior o tamanho da janela

de tempo, maior é a aceitação total final das VNs. Determinado fato ocorre devido ao processamento das VNs em conjunto trazer um ganho de informação quanto aos recursos demandados, onde sendo conhecidas previamente, as demandas de recursos são supridas pelas ofertas de maneira a atender o maior número de VNs possível.

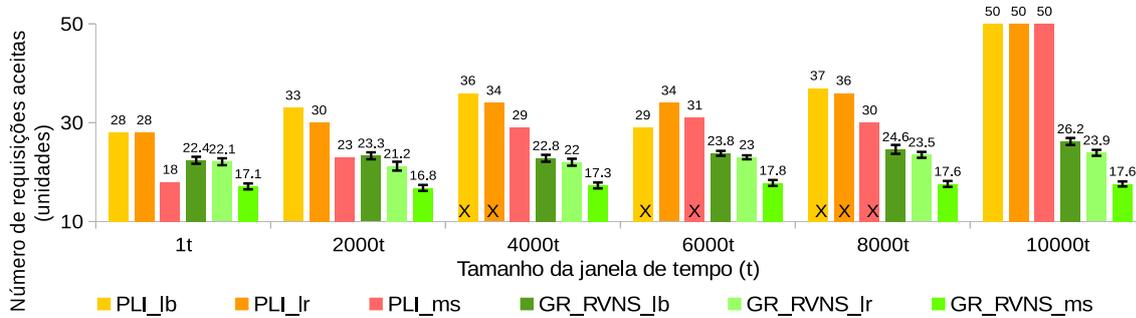


Figura 5.3: Número de requisições aceitas no cenário *SN1R1* (aceitação total, métrica 5.1).

Ainda na Figura 5.3, percebe-se que independente da função objetivo escolhida, a abordagem *offline* consegue mapear todas as VN requisitantes pelo CPLEX, o que demonstra o funcionamento da penalidade M inserida nas funções objetivos usadas. Quando o examinado são as abordagens que maximizam o lucro de maneira periódica, percebe-se que tanto o *PLI_lr* quanto *GR_RVNS_lr*, possuem uma aceitação total mais baixa que as abordagens *PLI_lb* quanto *GR_RVNS_lb* que maximizam o balanceamento. Tal fato indica que quando se procura maximizar a aceitação, a função de balanceamento de carga é a melhor opção.

A função objetivo que reduz o uso de enlaces inter-domínios, aplicada nas abordagens *PLI_ms* e *GR_RVNS_ms*, causa um impacto negativo grande em relação à aceitação total aferida nos cenários das Figuras 5.3, 5.4 e 5.5. Tal aspecto indica que o mapeamento se feito de forma a evitar o uso de enlaces inter-domínios, tende a gerar uma rejeição maior, ficando evidente que mesmo pagando o preço de uma reserva de banda para comunicação entre domínios adjacentes, o mapeamento em mais de um domínio é necessário para se ter uma aceitação alta.

Como constatado nos testes preliminares na seção 5.5.1, as abordagens *GR_RVNS* no geral têm uma aceitação pior que as abordagens baseadas em *PLI*, mas em contrapartida conseguem ser executadas em tempo satisfatório mesmo em cenários densos como o *offline* (Figuras 5.12, 5.13 e 5.14). No cenário com escassez e sobra de recursos (Figuras 5.4 e 5.5) as abordagem com *PLI* tem seu pior resultado em relação ao tempo gasto, onde devido a limitação de tempo de 18000s, a solução ótima é encontrada apenas nos cenários de janelas de tempo de 1t e 2000t.

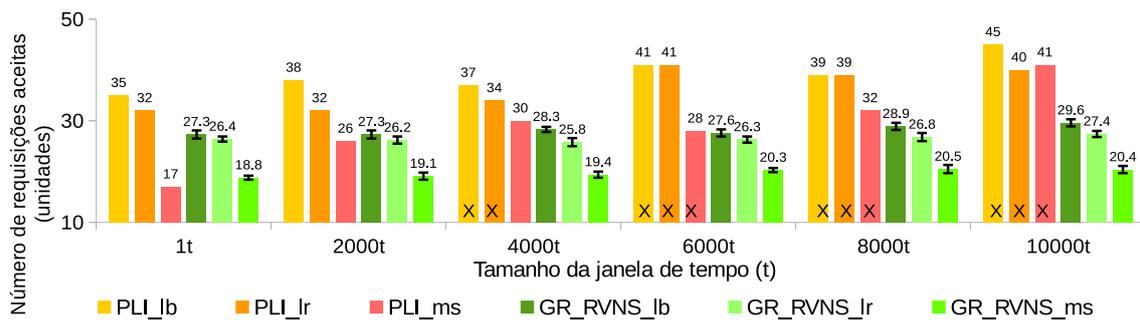


Figura 5.4: Número de requisições aceitas no cenário *SN2R1* (sobra de 25% de recursos, aceitação total, métrica 5.1).

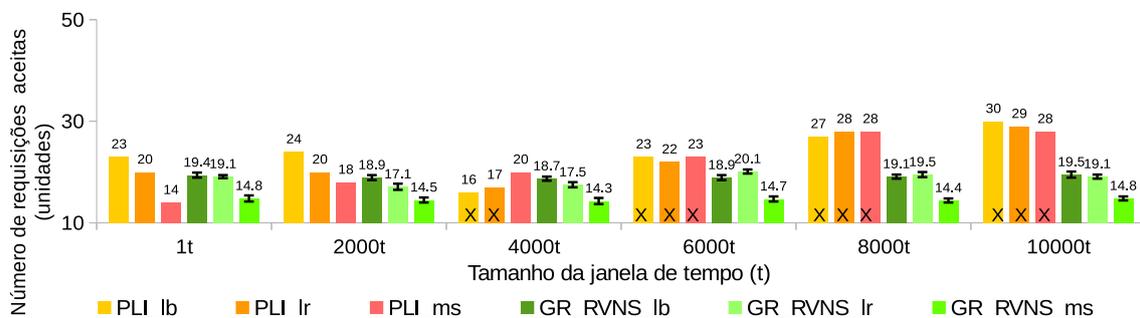


Figura 5.5: Número de requisições aceitas no cenário *SN3R1* (falta de 25% de recursos, aceitação total, métrica 5.1).

Ainda pode ser observado na Figura 5.3 que o CPLEX consegue achar o ótimo para a abordagem *offline*, mas não para o modelo periódico com janela de tempo de $8000t$. Neste caso, são executadas duas chamadas do modelo matemático sendo uma para a própria janela de $8000t$ e outra para a janela residual de $2000t$. Tal fato indica que é mais rápido uma execução *offline* de todo o problema com um uso total de recursos (cenário *SN1*), do que duas execuções com uso de recursos alterados. Neste caso a execução de $8000t$ ocorre em um ambiente com sobra de recursos, pois o SN consegue comportar todas as VNs juntas mas somente entram um subconjunto de VNs que chegam até $8000t$, o que ocasiona em um SN residual fragmentado para a execução da janela de tempo restante de $2000t$.

O consumo de energia e o lucro são duas métricas que andam juntas, dessa forma um gasto de energia baixo tende a ser uma condição necessária para um lucro final mais alto. As Figuras 5.6, 5.7 e 5.8 representam o lucro gerado pelas abordagens propostas nas variações de cenários *SN1*, *SN2* e *SN3*. Quando o analisado é a abordagem *online* (Figura 5.3), a *PLI_lr* e *PLI_lb* conseguem mapear a mesma quantidade de VNs, mas *PLI_lr* consegue um lucro maior que *PLI_lb* (Figura 5.6), fato ocorre por *PLI_lr*

fazer um uso mais conciso dos recursos de rede, o que minimiza o gasto de energia e maximiza o lucro.

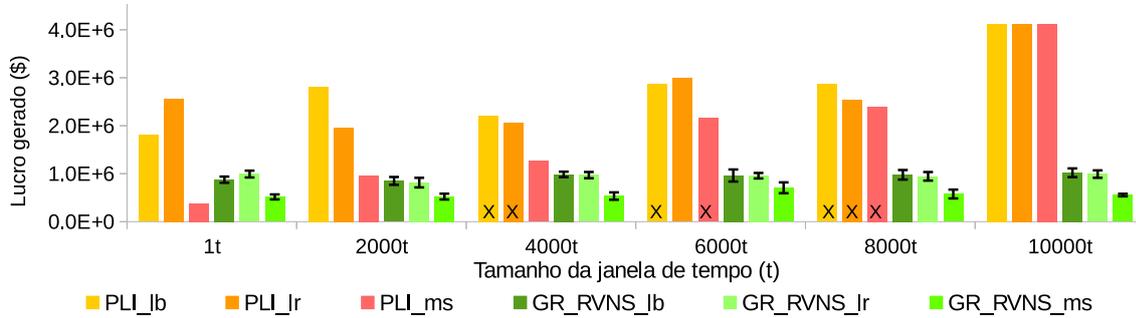


Figura 5.6: Lucro total gerado no cenário *SN1R1* (métrica 3.13).

Os testes *offline* apresentados nas figuras 5.3 e 5.6 chegaram ao ótimo global. Sendo os cenários de testes construídos com os princípios de Fischer & de Meer [2016], que garantem o consumo de recursos exato do SN por um mapeamento perfeito das VNs, pode se garantir que tais valores dos testes *offline* das figuras 5.3 e 5.6 são um limite superior à aceitação e ao lucro respectivamente.

Nos cenários onde é aplicada a modelagem exata (PLI), o ótimo global somente é alcançado na abordagem *offline*. Nesse sentido as abordagens *online* e periódica pecam por não terem um conhecimento total das VNs. Um exemplo acontece na Figura 5.6 com a janela de tempo de tamanho $2000t$, onde mesmo usando a função objetivo de maximizar o lucro a abordagem *PLI_lr* gera um lucro menor que *PLI_lb*. Tal fato sucede por *PLI_lb* possuir uma aceitação total maior que *PLI_lr* (Figura 5.3), o que pode implicar em um lucro maior.

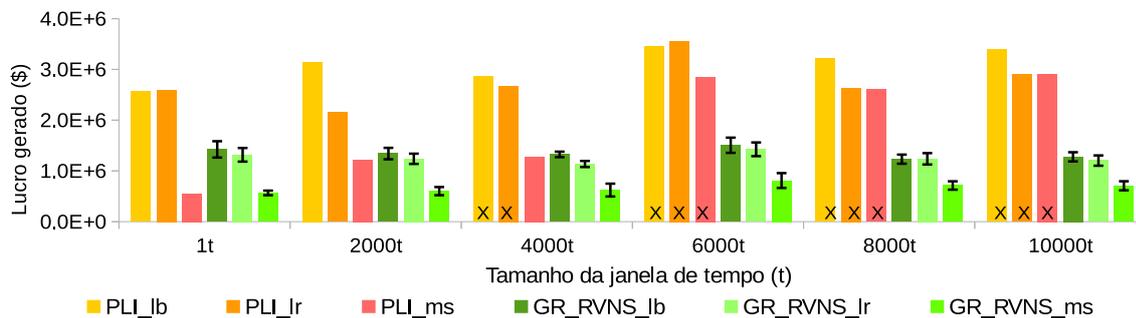


Figura 5.7: Lucro total gerado no cenário *SN2R1* (sobra de 25% de recursos, métrica 3.13).

Mais uma vez é percebido o bom funcionamento da penalidade M , onde todas as VNs são mapeadas gerando o maior lucro para as variações *offline* (Figura 5.6).

Diferentemente dos testes preliminares (seção 5.5.1) que são efetuados em um ambiente SD, as heurísticas *GR_RVNS* possuem um comportamento ruim se comparadas aos modelos de *PLI* em ambientes MD. Nesse caso a baixa aceitação final percebida nas Figuras 5.3, 5.4 e 5.5 têm como consequência um baixo lucro.

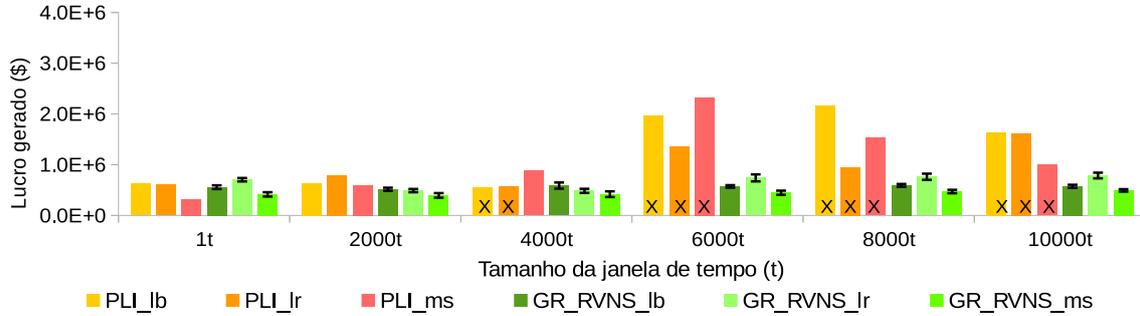


Figura 5.8: Lucro total gerado no cenário *SN3R1* (falta de 25% de recursos, métrica 3.13).

Ao se analisar o cenário com a sobra de recursos de 25% (Figura 5.7), o lucro gerado supera os valores percebidos na Figura 5.6. Nos modelos *online* e periódico, onde existem rejeições (Figuras 5.3, 5.4 e 5.5), tal fato indica que para se aumentar o lucro e a aceitação total uma das alternativas é aumentar a disponibilidade de recursos existentes. Para o cenário *offline* ótimo, onde não existe rejeição, para se aumentar o lucro deve se partir para outras alternativas, como troca de equipamentos do SN por outros que consumam menos energia, utilizar energia mais barata etc.

Nos cenários de recursos diminuídos, Figuras 5.5 e 5.8, as abordagens heurísticas *GR_RVNS* alcançam um desempenho mais próximo do modelo em *PLI* do que nos cenários 5.3 e 5.4. Substratos de rede como o SN2, que simulam a falta de recursos, muitas vezes acontecem em ambientes reais, nesses casos as abordagens *GR_RVNS* conseguem um bom lucro, em alguns momentos se aproximando dos modelos em *PLI*. Na Figura 5.8, a abordagem *GR_RVNS_lr* consegue no cenário *offline* lucrar mais que a abordagem *GR_RVNS_lb*, justamente por *GR_RVNS_lr* usar recursos que consumam menos energia e de forma mais concisa.

Quando uma VN é mapeada no modelo *online* ($|V| = 1$) não são gerados atrasos de mapeamento pois a VN é processada de imediato. Quando o analisado é um cenário periódico, atrasos potencialmente vão ocorrer (seção 4.3). As Figuras 5.9, 5.10 e 5.11 demonstram os atrasos gerados pelas VN aceitas.

Observando as Figuras 5.9, 5.10 e 5.11, de modo a analisar globalmente as janelas de tempo, percebe-se que o atraso médio de mapeamento cresce de acordo com o aumento da janela de tempo. Onde em janelas de tempo maiores, as VNs entrantes

do começo da janela devem esperar mais para serem processadas que em janelas de tempos menores, mas, em contrapartida, quanto maior o tamanho da janela de tempo, maior é também o lucro (Figuras 5.6, 5.7 e 5.8).

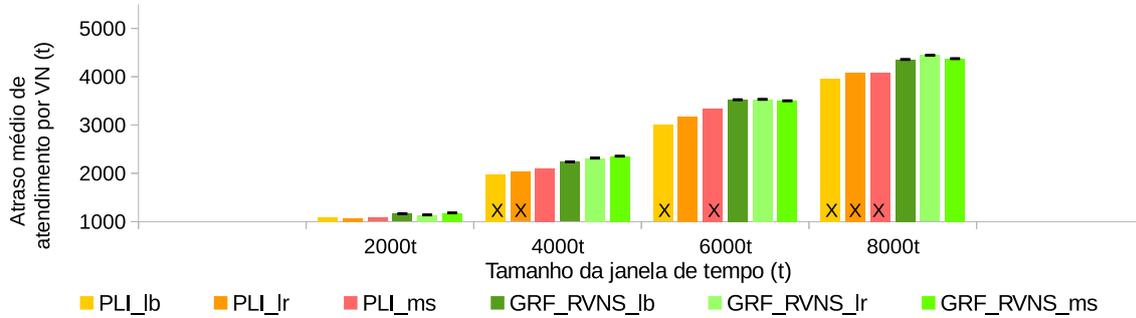


Figura 5.9: Atraso médio de atendimento no cenário *SN1R1* (métrica 5.2).

Comparando as Figuras 5.3, 5.4 e 5.5 com as Figuras 5.9, 5.10 e 5.11, de modo a observar cada janela de tempo em separado, percebe-se que as variações que possuem uma aceitação final maior tendem a gerar um atraso médio mais balanceado, se aproximando do valor médio da janela de tempo, como na Figura 5.10 na janela de 8000t, percebe-se que o atraso médio de cada VN processada pelas abordagens *PLI* ficam aproximados a 4000t. Esse comportamento ocorre devido ao fato que ao se mapear mais VNs o atraso médio fica mais balanceado entre as VNs, ou seja, tanto as VNs que estão no começo da janela de tempo vão ser mapeadas com um atraso grande, quanto as que estão ao final vão ser mapeadas com um atraso pequeno, gerando um atraso médio mais balanceado. O mesmo não ocorre quando existe uma baixa aceitação, como na Figura 5.10 na janela de 4000t, onde a abordagem *GRF_RVNS_ms* possui um baixo atraso médio, esse acontecimento se dá justamente pela baixa aceitação total da abordagem percebida na Figura 5.4, onde as poucas VNs mapeadas por *GRF_RVNS_ms* estavam localizadas ao final da janela de tempo.

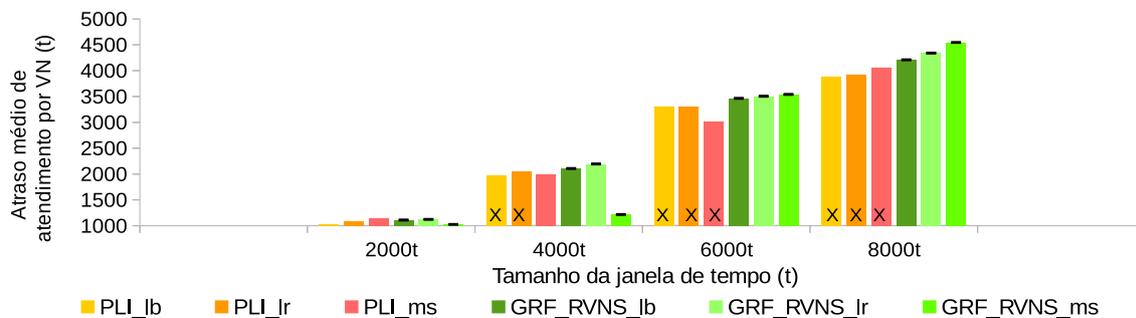


Figura 5.10: Atraso médio de atendimento no cenário *SN2R1* (sobra de 25% de recursos, métrica 5.2).

O mapeamento se feito com janelas de tempo grandes, além de afetar negativamente o cliente que pode ter que esperar para ter sua conexão atendida (Figuras 5.9, 5.10 e 5.11), pode causar uma sobrecarga repentina no SN, que se muito alta pode causar problemas de processamento, latência e vazão (Zheng & Veeravalli [2012]). Como exemplo, em um momento $t1$ o SN pode estar com poucas VNs ativas, o que acarreta em um consumo baixo de energia e continuar assim até um momento posterior. Em outro momento, por exemplo, em $t10$ pode-se ser necessário mapear outro lote de VNs, o que pode causar um aumento abrupto e significativo no uso dos recursos do SN e em um conseqüente aumento no consumo de energia. Como o lote mapeado em $t10$ foi ativado junto as VNs tendem a ter o tempo de vida expirados de maneira relativamente próximas, o que pode acarretar em um novo período de ociosidade do SN. Essa alternância de ciclos pode ser prejudicial para o SN ocasionando picos de consumo de energia, mapeamentos com muitos saltos (*hops*) e baixo balanceamento de carga, mas, em contrapartida pode ser benéfico se analisado o consumo final de energia. Neste caso os roteadores e enlaces somente são ligados para um processamento alto de requisições, amortizando o custo de ligar determinado componente entre as suas várias VNs ativas mapeadas.

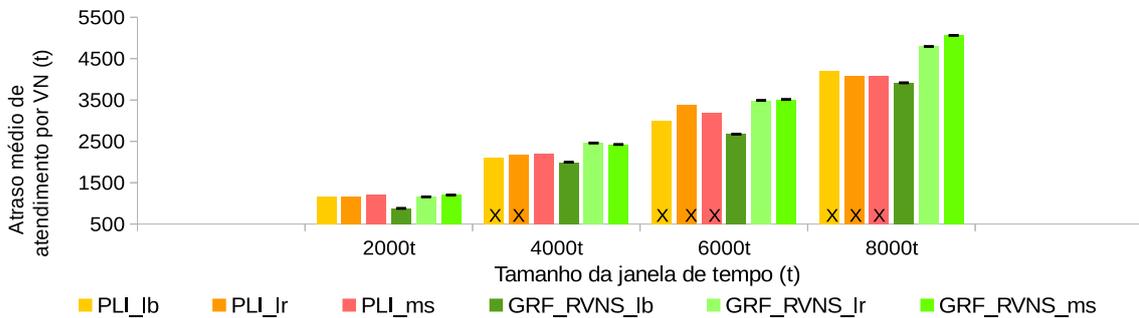


Figura 5.11: Atraso médio de atendimento no cenário *SN3R1* (falta de 25% de recursos, métrica 5.2).

As Figuras 5.12, 5.13 e 5.14 mostram o gasto de tempo final para o processamento de todas requisições. Neste caso o tempo é computado tanto para as VN aceitas, quanto para as rejeitadas. O tempo gasto pelas abordagens *GR_RVNS* é no máximo de $300ms$ por requisição, independente do cenário ser *online*, periódico ou *offline*. Quanto a abordagem em *PLI*, esta possui um tempo máximo de processamento de $1800s$, justificado por ser uma abordagem exata, onde no pior caso todas as soluções devem ser investigadas.

Em cenários *online*, como nas Figuras 5.12, 5.13 e 5.14, a abordagem *PLI* (junto ao CPLEX) consegue convergir rapidamente para a solução, sendo mais rápido que

as heurísticas GR_RVNS . Tal fato ocorre pelo número de restrições e variáveis serem menores que nos modelos periódicos (Tabela 5.5), e a abordagem GR_RVNS possuir um tempo de busca local fixado em $240ms$, onde em alguns casos o algoritmo pode ficar executando sem gerar melhoras na função objetivo. Mas tal fato não se repete para outros cenários. Quando a janela de tempo é aumentada, aumenta-se o tamanho do lote de VNs processadas, o que implica diretamente no aumento do número de variáveis. Nesse caso o modelo PLI passa a consumir um tempo maior de execução, já superando o tempo das abordagens GR_RVNS . Na modelagem proposta (seção 4.2) quanto maior o tamanho do lote analisado, maior o número de variáveis inteiras e restrições que o modelo vai possuir (Tabela 5.5).

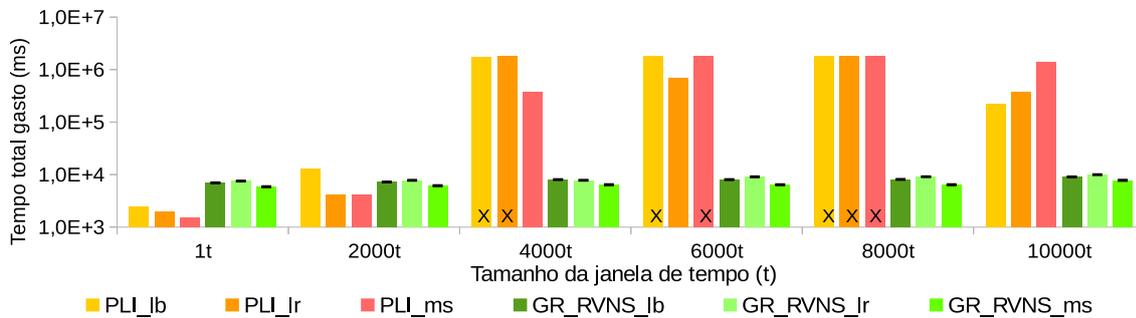


Figura 5.12: Tempo total de execução no cenário $SN1R1$.

Tabela 5.5: Quantidade de restrições e variáveis inteiras para o cenário $SN1R1$.

Tamanho do lote de VNs (unidades)	1	10	20	30	40	50
Número de linhas (restrições)	281	3219	7949	12312	16843	22801
Número de colunas (variáveis)	334	11337	29730	46599	64106	87242

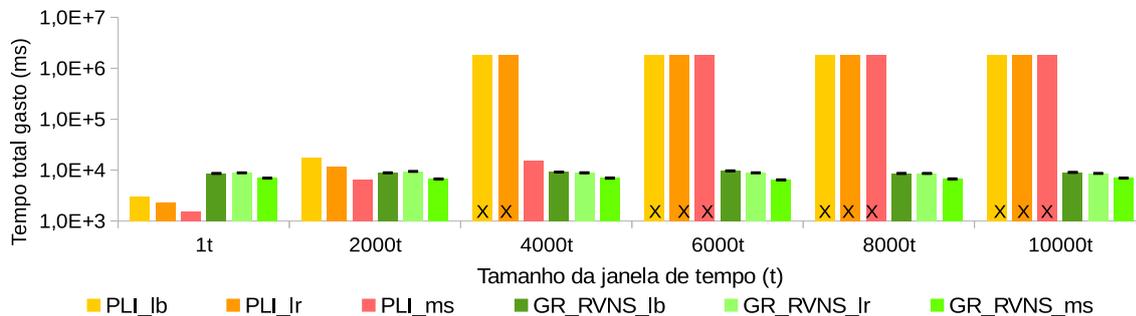


Figura 5.13: Tempo total de execução no cenário $SN2R1$ (sobra de 25% de recursos).

Outro ponto que eleva o tempo de processamento do modelo PLI é o aumento da quantidade de recursos ofertados pelo SN (Figura 5.13). Neste caso ao se aumentar as

capacidades de oferta aumenta-se também as opções para o mapeamento, o que gera mais soluções viáveis a serem investigadas. A redução dos recursos ofertados pelo SN também gera um aumento no tempo de processamento para o modelo em PLI (Figura 5.14), tal fato ocorre por nesses casos o modelo ter que analisar quais VNs devem ser rejeitadas ou aceitas, sendo que para cada VN rejeitada/aceita existe um residual diferente de recursos físicos a ser investigado para o mapeamento das outras VNs.

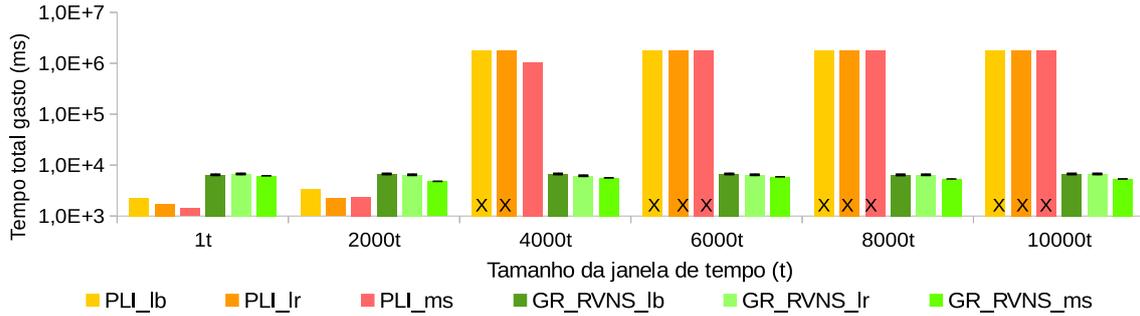


Figura 5.14: Tempo total de execução no cenário *SN3R1* (falta de 25% de recursos).

Ainda nas Figuras 5.12, 5.13 e 5.14 destaca-se que dentre as abordagens PLI que conseguiram chegar ao resultado ótimo, as que possuem o menor tempo de execução são as que buscam minimizar o uso de enlaces inter-domínios (*PLI_ms*). Este acontecimento é baseado no fato que a função objetivo que minimiza o uso de enlaces inter-domínios (equação 3.14) age junto a um conjunto de componentes físicos bem menor que as funções que maximizam o lucro (equação 3.13) e que maximizam o balanceamento de carga (equação 3.7)

5.5.3 Análise do Orquestrador

Nesta seção são mostrados os resultados computacionais obtidos através das simulações realizadas. Foram avaliadas três abordagens: *VNEO_ct*, *VNEO_cp* e *VNEO_sc*. Todas as abordagens utilizaram o algoritmo de mapeamento GRASPF-RVNS combinado ao seu modelo de orquestrador.

Os experimentos mostrados na Figura 5.15 abordam a aceitação final e na Figura 5.16 o lucro acumulado final. Percebe-se que em ambos cenários, de baixa densidade (*SN1R2*) e de alta densidade (*SN1R3*), a aceitação da abordagem *VNEO_ct* apresenta-se ligeiramente superior a *VNEO_cp* e *VNEO_sc*; fato esperado por *VNEO_ct* aplicar a otimização globalmente buscando maximizar o lucro, no caso para um lucro alto deve haver uma aceitação final alta. Quando são observados *VNEO_cp* e *VNEO_sc*, percebe-se que *VNEO_sc* possui uma aceitação final ligeiramente mais

alta que $VNEO_cp$. Tal fato ocorre por $VNEO_sc$ consultar todos os domínios e escolher o que lucra mais, e que conseqüentemente gasta menos energia. Um gasto reduzido de energia implica no menor uso de recursos físicos, o que afeta diretamente no residual dos componentes e na disponibilidade de recursos para mapeamentos futuros.

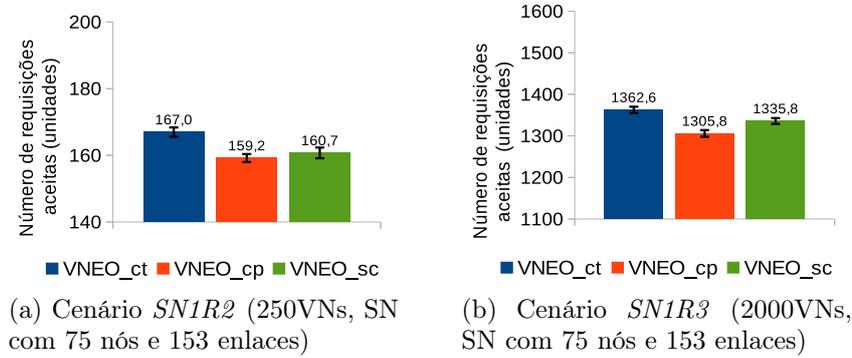


Figura 5.15: Número de requisições aceitas (equação 5.1)

Em relação ao lucro final, apresentado nas Figuras 5.16(a) e 5.16(b), $VNEO_sc$ acaba lucrando mais do que $VNEO_cp$ em ambos cenários, refletindo as consultas que o orquestrador faz a fim de contratar os serviços do domínio de maior lucro (menor gasto de energia). Além disso, percebe-se que o cenário $SN1R2$ acaba gerando um lucro maior que o cenário $SN1R3$. Este fato acontece pelas VNs de $SN1R2$ possuem um tempo de vida maior, o que gera uma taxa de lucro mais constante. No cenário $SN1R3$, as requisições possuem um tempo de vida menor e uma taxa de rotatividade maior, logo, as entradas e saídas acabam gerando baixas no SN por determinados períodos de tempo, o que afeta o lucro.

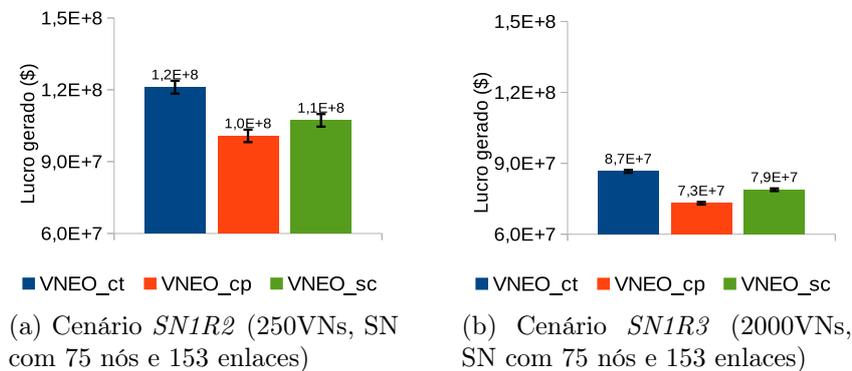


Figura 5.16: Lucro final acumulado (equação 3.13)

Se comparamos as Figuras 5.16(a) com 5.17(a) e 5.16(b) com 5.17(b), percebemos que o consumo de energia tem um comportamento complementar ao lucro dos domínios,

refletindo os aspectos da modelagem, onde um lucro maior é gerado com base em um gasto menor. É possível notar ainda que, em ambos cenários, *VNEO_cp* sempre está gastando mais energia e lucrando menos que as outras abordagens. Este fato ocorre por esta abordagem se preocupar com a saturação dos enlaces (equação 4.3) e baixa comunicação entre domínios (Figuras 5.18(a) e 5.18(b)), abrindo mão de minimizar o consumo de energia (Figuras 5.17(a) e 5.17(b)).

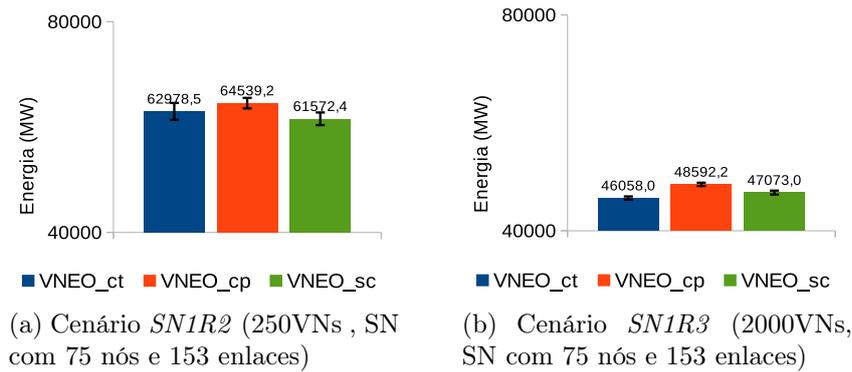


Figura 5.17: Gasto de energia total (equação 3.11)

Mesmo que uma VN seja rejeitada, as mensagens de coordenação trocadas durante a tentativa de seu mapeamento são computadas. A diferença da quantidade de mensagens trocadas (Figuras 5.18(a) e 5.18(b)) é fruto do fato de *SN1R2* possuir VNs com tempo de vida maior, havendo pouca rotatividade; ao passo que em *SN1R3* existe um número muito grande de VNs entrantes e saíntes com um tempo de vida menor, o que eleva a quantidade de mensagens.

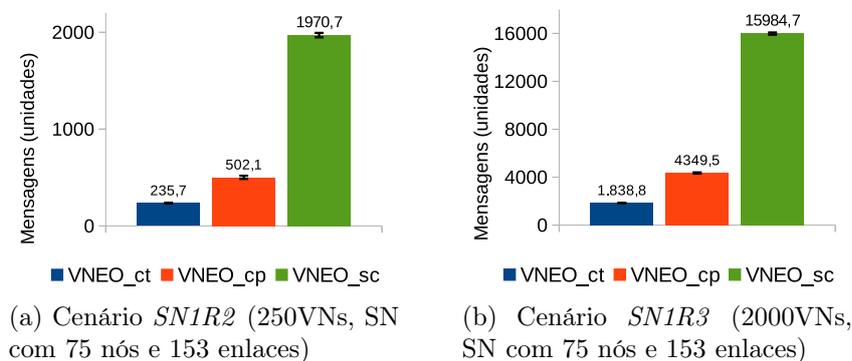


Figura 5.18: Total de mensagens trocadas (Tabela 5.1)

Percebe-se que o *overhead* de coordenação causado pela comunicação interdomínio não é tão impactante em termos de aceitação, mas se levado em conta for a

quantidade de mensagens enviadas/recebidas, mostradas nas Figuras 5.18(a) e 5.18(b), pode-se haver problemas com atrasos no tempo de mapeamento até que sua coordenação seja efetivada entre vários domínios. Analisando o comportamento das diferentes abordagens percebe-se que em ambos cenários a abordagem *VNEO_sc* gera uma carga mais alta de mensagens trocadas em relação a abordagem *VNEO_cp*, pois em *VNEO_cp*, o orquestrador já possui uma certa informação que implica na consulta sendo realizada em poucos domínios. Por outro lado, *VNEO_sc* é capaz de prover soluções de melhor qualidade para o mapeamento. Dessa forma, observa-se que a maior troca de mensagens permite uma melhor coordenação e otimização dos recursos.

Capítulo 6

Considerações Finais

Este trabalho discorreu sobre diferentes abordagens para o mapeamento de redes virtuais de forma *online*, periódica e *offline*. Ao final, ainda foram analisados aspectos de troca de mensagens de coordenação e sincronização geradas por diferentes implementações. Foi apresentado um breve referencial teórico sobre o assunto, onde foram elucidados alguns benefícios e desafios gerados por prover o mapeamento de redes virtuais.

6.1 Conclusões

A virtualização de redes possui grande importância para o futuro da internet, sendo fundamental o estudo e aprimoramento das teorias e técnicas existentes para a adequação à realidade ambiental e social que o mundo vive. Percebe-se que existem muitos desafios a serem superados como a de prover conexões de redes rápidas, estáveis, de custo baixo e energeticamente sustentáveis.

As análises mostraram que em determinados cenários o número de rejeições de VNs pode ser elevado, mesmo com folgas nos recursos do SN e com uso de técnicas de otimização adequadas. Por se tratar de um problema de natureza *online*, as VNs que tem seu tempo de vida finalizado, deixam o SN suscetível a uma possível fragmentação. Por conseguinte, mesmo que existam recursos disponíveis, tais recursos podem se tornar inacessíveis (isolados) para um mapeamento futuro.

Em tratamentos *online* e periódicos do problema, soluções ótimas globais dificilmente serão encontradas com as técnicas aplicadas. O ótimo global será alcançado com a aplicação de mecanismos de reconfiguração junto à abordagem exata, de modo que a cada nova VN entrante o mapeamento de todas as VNs ativas sobre o SN é reconfigurado. Dessa maneira, a nova VN entrante irá fazer parte da nova solução ótima gerada.

Porém, deve se levar em conta que o número de variáveis analisadas pelo modelo irá aumentar, o que demanda mais tempo de execução para se gerar o resultado ótimo.

Apesar da queda da aceitação final e conseqüente fragmentação constatados nos cenários *online* e periódicos, estes cenários são os que mais se aproximam de aspectos reais, e na prática são os que recebem mais atenção da comunidade científica. Em cenários *offline* existe um aumento significativo do número de variáveis a serem analisadas, o que deixa a resolução extremamente complexa e custosa. Constatou-se que o modelo PLI não se mostra adequado do ponto de vista de tempo de execução, já que o tempo de execução foi alto em vários casos e na prática aplicações para este tipo de problema necessitam de uma solução rápida. A abordagem heurística portou-se de maneira satisfatória, onde apesar do baixo desempenho em relação à abordagem PLI, possui a vantagem de trabalhar em um tempo viável.

Por se tratar de um ambiente dinâmico quanto à chegada das requisições, as abordagens *online* e periódica podem ter um comportamento final deturpado em relação ao objetivo buscado. Fato se dá pelo mapeamento feito de forma imediata, sem a informação de futuros mapeamentos e sem aplicação de técnicas de reconfiguração para se mitigar esse problema. Constatou-se que maximizar o balanceamento de carga mostrou-se promissor quanto a maximizar a aceitação total. Nesse caso quando procura-se maximizar o lucro, a melhor alternativa também pode ser maximizar o balanceamento de carga, pois uma aceitação maior terá impactos em uma receita maior, e em uma despesa relativamente menor, pois haverá várias VNs ativas para fracionar o custo de energia base de equipamentos em *stand-by* de rede.

Os experimentos computacionais mostraram que as modelagens possuem um bom comportamento em caráter de simulação, conseguindo alcançar bons resultados, principalmente quanto à economia de recursos financeiros e energéticos. Não se pode afirmar qual modelagem é melhor, mas é possível analisar qual seria mais indicada em função do cenário e objetivo da aplicação. Por ser um estudo ímpar, torna-se de grande importância para referências teóricas futuras.

Os testes mostraram que o uso de técnicas adequadas podem aumentar significativamente o lucro de um domínio e minimizar o consumo, mas que podem implicar em uma troca maior de mensagens de coordenação e de comunicação. Os testes mostraram que a troca de mensagens de coordenação não tem um impacto muito alto, em termos de redução do número de VNs aceitas. No entanto, pode acarretar algum atraso no tempo de mapeamento até que sua coordenação seja efetivada entre vários domínios. Outro ponto é que a troca de mensagens de coordenação gerada pelo orquestrador é executada somente no momento de estudo e posicionamento do mapeamento da VN. Por outro lado, as mensagens de comunicação, continuam ativas durante o

tempo de vida da VN, o que gera um consumo extra de banda e gasto de energia, mas, mesmo assim se percebe que esse consumo de banda não se torna um problema para se maximizar a aceitação total.

Os resultados obtidos pelas simulações ainda indicam que o aumento do gasto de energia e da troca de mensagens de coordenação, junto à diminuição do lucro gerado por modelos orquestrados, ainda são desafios a serem superados. A heurística de mapeamento GRASPF-RVNS se comportou bem junto ao orquestrador, mesmo perante a dificuldade do problema e ao baixo tempo computacional exigido.

6.2 Trabalhos Futuros

De posse dos resultados observados na seção 5, percebe-se que alguns pontos podem ser melhorados. A abordagem GRASPF-RVNS, apesar de rápida pode ser melhor elaborada, principalmente em cenários periódicos. A escolha de se ordenar as VNs para serem processadas em lotes, mostra-se rápida, mas em contrapartida não gerou bons resultados. Alternativas de um mapeamento em conjunto para determinado lote de VNs pode ser implantado nesse caso.

Os conceitos de vizinhança podem ser mais bem trabalhados, principalmente no que diz respeito à geração de vizinhanças no mesmo domínio ou em domínios diferentes. Outras linhas de heurísticas podem ser aplicadas na tentativa de gerar maiores ganhos em termos das funções objetivo.

As métricas de troca de mensagens tanto de coordenação quanto de comunicação não foram trabalhadas quanto ao *delay* causado, próximos trabalhos podem levar em conta tal análise.

O modelo matemático por ter-se mostrado lento em instâncias de maior porte pode ser tratado de forma diferenciada, como por exemplo, com uso de geração de colunas, ou com abordagens híbridas, como *matheuristics*.

Apesar dos testes feitos englobarem um grande número de instâncias diferentes, com uma geração das topologias utilizando técnicas fortemente usadas na literatura, outros cenários ainda devem ser investigados, principalmente no que se refere à vulnerabilidade de recursos e atrasos de mapeamentos em cenários de porte realmente volumosos.

Em trabalhos futuros pretendemos estender e avaliar experimentalmente os modelos propostos para o VNE adicionando conceitos de funções de rede e SDN, propondo uma demanda elástica de requisições e a variação do consumo de banda e CPU dos componentes físicos. Pretendemos ainda aprimorar o modelo de energia, aplicar e

considerar outros fatores como variação da tarifação de energia elétrica, ganhos com energias limpas e crédito de carbono além de despesas de operação e expansão.

Referências Bibliográficas

- Addis, B.; Capone, A.; Carello, G.; Gianoli, L. & Sansò, B. (2016). Energy management in communication networks: a journey through modeling and optimization glasses. *Computer Communications*, 91-92:76 – 94. ISSN 0140-3664.
- Alkmim, G. P.; Batista, D. M. & Fonseca, N. L. S. d. (2011). Mapeamento de redes virtuais em substratos de rede. *Anais do XXIX SBRC*.
- 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(Supplement C):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.
- Anderson, T.; Peterson, L.; Shenker, S. & Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41. ISSN 0018-9162.
- Araújo, S.; Souza, F. S. H. & Mateus, G. R. (2018). Virtual network embedding in Multi-Domain environments with energy efficiency concepts. Em *2018 International Conference on Information Networking (ICOIN 2018)*, Chiang Ma, Thailand.
- Araújo, S. M. A.; Guidoni, D. L.; de Souza, F. S. H. & Mateus, G. R. (2018). Managing virtual network embedding through reconfiguration and expansion. *SIMULATION*. ISSN 0037-5497.
- Araújo, S. M. A.; Souza, F. S. H. & Mateus, R. G. (2017). Modelagens para o problema de mapeamento de redes virtuais em ambientes multi-domínio. *Anais do XLIX SBPO*.

- Beck, M. T.; Fischer, A.; Botero, J. F.; Linnhoff-Popien, C. & de Meer, H. (2015). Distributed and scalable embedding of virtual networks. *Journal of Network and Computer Applications*, 56:124 – 136. ISSN 1084-8045.
- Beeharry, J. & Nowbutsing, B. (2016). Forecasting ipv4 exhaustion and ipv6 migration. Em *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, pp. 336–340.
- Beikta, C.; Gzpek, D.; Ula, A. & Lokman, E. (2017). Secure virtual network embedding with flexible bandwidth-based revenue maximization. *Comput. Netw.*, 121(C):89–99. ISSN 1389-1286.
- Bojanova, I. & Voas, J. (2017). Trusting the internet of things. *IT Professional*, 19(5):16–19. ISSN 1520-9202.
- Calheiros, R. N.; Ranjan, R.; Beloglazov, A.; De Rose, C. A. F. & Buyya, R. (2011). Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software—Practice & Experience*, 41(1):23--50. ISSN 0038-0644.
- Cheng, X.; Su, S.; Zhang, Z.; Wang, H.; Yang, F.; Luo, Y. & Wang, J. (2011). Virtual network embedding through topology-aware node ranking. *SIGCOMM Comput. Commun. Rev.*, 41(2):38--47. ISSN 0146-4833.
- Chiaraviglio, L.; Mellia, M. & Neri, F. (2012). Minimizing isp network energy cost: Formulation and solutions. *IEEE/ACM Transactions on Networking*, 20(2):463–476. ISSN 1063-6692.
- Chowdhury, N. M. K. & Boutaba, R. (2008). A survey of network virtualization. Relatório técnico CS-2008-25, University of Waterloo, David R. Cheriton School of Computer Science, Canada.
- Chowdhury, N. M. M. K. & Boutaba, R. (2009). Network virtualization: State of the art and research challenges. *IEEE Communications Magazine*, 47(7):20–26. ISSN 0163-6804.
- Chowdhury, N. M. M. K.; Rahman, M. R. & Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. Em *IEEE INFOCOM 2009*, pp. 783–791. ISSN 0743-166X.
- Coniglio, S.; Grimm, B.; Koster, A. M. C. A.; Tieves, M. & Werner, A. (2015). Optimal offline virtual network embedding with rent-at-bulk aspects. *CoRR*, abs/1501.07887.

- Dalla-Costa, A. G.; Schimuneck, M. A. K.; Wickboldt, J. A.; Both, C. B.; Gaspary, L. P. & Granville, L. Z. (2017). NFV em redes 5G: Avaliando o desempenho de composição de funções virtualizadas via maestro. *Anais do XXXV SBRC*.
- Eramo, V.; Miucci, E. & Ammar, M. (2016). Study of reconfiguration cost and energy aware VNE policies in cycle-stationary traffic scenarios. *IEEE Journal on Selected Areas in Communications*, 34(5):1281–1297. ISSN 0733-8716.
- Fajjari, I.; Aitsaadi, N.; Pujolle, G. & Zimmermann, H. (2011). Vnr algorithm: A greedy approach for virtual networks reconfigurations. *Globecom 2011 - IEEE Global Communications Conference*, pp. 1–6.
- Farah, R. (2004). *Psicologia e Informática: O ser humano diante de novas tecnologias*. Oficina do Livro, São Paulo. ISBN 9788588698246.
- Feo, T. A. & Resende, M. G. (1995). Greedy randomized adaptive search procedure. *Journal of Global Optimization*, 6:109–133.
- Fischer, A. (2016). *An Evaluation Methodology for Virtual Network Embedding*. PhD thesis, Queensland University of Technology.
- Fischer, A.; Botero, J. F.; Beck, M. T.; de Meer, H. & Hesselbach, X. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906. ISSN 1553-877X.
- Fischer, A. & de Meer, H. (2016). Generating virtual network embedding problems with guaranteed solutions. *IEEE Transactions on Network and Service Management*, 13(3):504–517. ISSN 1932-4537.
- Ghazisaeedi, E. & Huang, C. (2015). Energy-aware node and link reconfiguration for virtualized network environments. *Computer Networks*, 93:460 – 479. ISSN 1389-1286. Cloud Networking and Communications II.
- Goldberg, R. P. (1974). Survey of virtual machine research. *Computer*, 7(6):34–45. ISSN 0018-9162.
- Guerzoni, R.; Trivisonno, R.; Vaishnavi, I.; Despotovic, Z.; Hecker, A.; Beker, S. & Soldani, D. (2014). A novel approach to virtual networks embedding for sdn management and orchestration. Em *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–7. ISSN 1542-1201.

- 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 and autonomic virtual network mapping framework. pp. 241–247. ISSN 2168-1864.
- Leivadeas, A.; Papagianni, C. & Papavassiliou, S. (2013). Efficient resource mapping framework over networked clouds via iterated local search-based request partitioning. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1077–1086. ISSN 1045-9219.
- Liu, X.; Zhang, Z.; Li, X. & Su, S. (2016). Optimal virtual network embedding based on artificial bee colony. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):273. ISSN 1687-1499.
- Luizelli, M. C.; Bays, L. R.; Buriol, L. S.; Barcellos, M. P. & Gasparly, L. P. (2015). Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. Em *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 98–106. ISSN 1573-0077.
- Lule, J. (2016). *Understanding Media and Culture: An Introduction to Mass Communication*. University of Minnesota Libraries Publishing. ISBN 13:978-1-9461352-6-1.
- Melo, M.; Sargento, S.; Killat, U.; Timm-Giel, A. & Carapinha, J. (2013). Optimal virtual network embedding: Node-link formulation. *IEEE Transactions on Network and Service Management*, 10(4):356–368. ISSN 1932-4537.
- Mladenović, N. & Hansen, P. (1997). Variable neighborhood search. *Computer and Operations Research*, 24(11):1097–1100. ISSN 0305-0548.
- Nguyen, L. D.; Kim, N.; Kim, S. & Kim, C.-K. (2017). Rt-vne: A real-time strategy for virtual network embedding towards resource efficiency. Em *2017 International Conference on Information Networking (ICOIN)*, pp. 185–190.
- Nonde, L.; El-Gorashi, T. E. H. & Elmirghani, J. M. H. (2015). Energy efficient virtual network embedding for cloud networks. *Journal of Lightwave Technology*, 33(9):1828–1849. ISSN 0733-8724.
- Oliveira, R. R.; Marcon, D. S.; Bays, L. R.; Neves, M. C.; Gasparly, L. P.; Medhi, D. & Barcellos, M. P. (2015). Opportunistic resilience embedding (ore): Toward

- cost-efficient resilient virtual networks. *Computer Networks*, 89(Supplement C):59 – 77. ISSN 1389-1286.
- Pham, D. P.; Lin, C. F.; Yuan, S. M. & Jou, E. (2011). Database backed by cloud data store for on-premise applications. pp. 708–713.
- Samuel, F.; Chowdhury, M. & Boutaba, R. (2013). Polyvine: policy-based virtual network embedding across multiple domains. *Journal of Internet Services and Applications*, 4(1):6. ISSN 1869-0238.
- Srikantaiah, S.; Kansal, A. & Zhao, F. (2008). Energy aware consolidation for cloud computing. Em *Proceedings of the 2008 Conference on Power Aware Computing and Systems*, HotPower'08, pp. 10--10, Berkeley, CA, USA. USENIX Association.
- Stasi, G. D.; Avallone, S. & Canonico, R. (2013). Virtual network embedding in wireless mesh networks through reconfiguration of channels. *IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 537–544.
- Tao, L.; C. Zhao, Y.; Thulasiraman, K. & Swamy, M. (1992). Simulated annealing and tabu search algorithms for multiway graph partition. 02.
- The Climate Group (2008). SMART 2020: Enabling the low carbon economy in the information age. Global eSustainability Initiative (GeSI).
- 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.
- Waxman, B. M. (1988). Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622. ISSN 0733-8716.
- Yu, H.; Anand, V.; Qiao, C. & Sun, G. (2011). Cost efficient design of survivable virtual infrastructure to recover from facility node failures. Em *2011 IEEE International Conference on Communications (ICC)*, pp. 1–6. ISSN 1550-3607.
- 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.
- Zhang, X.; Chen, X. & Phillips, C. (2012). Achieving effective resilience for qos-aware application mapping. *Computer Networks*, 56(14):3179 – 3191. ISSN 1389-1286.

- Zhang, Z.; Cheng, X.; Su, S.; Wang, Y.; Shuang, K. & Luo, Y. (2013). A unified enhanced particle swarm optimization-based virtual network embedding algorithm. *International Journal of Communication Systems*, 26(8):1054--1073. ISSN 1099-1131.
- Zheng, Q. & Veeravalli, B. (2012). Utilization-based pricing for power management and profit optimization in data centers. *Journal of Parallel and Distributed Computing*, 72(1):27 – 34. ISSN 0743-7315.
- Zhu, Y. & Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. pp. 1–12. ISSN 0743-166X.