

**Diego Luchi**

***MODELO DE MINIMIZAÇÃO DO TRÁFEGO  
RETRANSMITIDO COM DISTRIBUIÇÃO DE  
TRÁFEGO EFICIENTE PARA O PROJETO DA  
TOPOLOGIA VIRTUAL***

Vitória (ES), Brasil

Maio de 2012

**Diego Luchi**

***MODELO DE MINIMIZAÇÃO DO TRÁFEGO  
RETRANSMITIDO COM DISTRIBUIÇÃO DE  
TRÁFEGO EFICIENTE PARA O PROJETO DA  
TOPOLOGIA VIRTUAL***

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do título de Mestre em Informática.

Orientadora:  
Prof<sup>ª</sup>. Dr<sup>ª</sup>. Maria Cristina Rangel

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória (ES), Brasil

Maio de 2012

***MODELO DE MINIMIZAÇÃO DO TRÁFEGO  
RETRANSMITIDO COM DISTRIBUIÇÃO DE TRÁFEGO  
EFICIENTE PARA O PROJETO DA TOPOLOGIA VIRTUAL***

DIEGO LUCHI

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do título de Mestre em Informática.

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Maria Cristina Rangel  
Orientadora  
PPGI/UFES

---

Prof. Dr. Elias Silva de Oliveira  
PPGI/UFES

---

Prof. Dr. Marcelo Eduardo Vieira Segatto  
PPGEE/UFES

# *Dedicatória*

Dedico este trabalho a minha família. Em especial a minha mãe, por ter se dedicado ao máximo para que esse objetivo fosse atingido.

A minha orientadora pela disponibilidade e paciência.

A FAPES pelo apoio financeiro que foi essencial à dedicação exclusiva ao Programa de Pós-Graduação em Informática.

# *Resumo*

Com a evolução da Internet há uma demanda crescente por taxas de transmissões maiores nas redes de comunicações de longas distâncias (WAN), e os provedores de tais serviços sempre se esforçaram para otimizar os recursos disponíveis para atender essa demanda. O amadurecimento da tecnologia e dispositivos de comunicação utilizando fibra óptica possibilitou remodelar a topologia da rede, livre da topologia física, denominada topologia virtual. Estabelecer essa topologia pode ser modelado como um problema de otimização inteira mista. Podemos citar o modelo de minimização do tráfego retransmitido que busca uma topologia virtual que minimize a necessidade de processamento total dos roteadores, que constitui o maior gargalo atualmente das redes de comunicações, aumentando assim a vazão da rede. Nesse trabalho apresentamos uma função de avaliação para esse modelo com complexidade  $O(N^2)$ , que nos permite avaliar tanto a conexidade quanto seu valor de função objetivo podendo ser utilizada em conjunto com qualquer meta-heurística. Também realizamos uma pequena modificação na função objetivo do modelo de minimização do tráfego retransmitido encontrado na literatura, pois, apesar de minimizar o tráfego total processado na rede, não o distribui da melhor forma, acarretando a sobrecarga de alguns nós da rede física enquanto outros permanecem ociosos. Essa modificação, com finalidade de distribuir melhor o tráfego, foi realizada inserindo um termo na função objetivo relativo ao nó com maior carga de processamento. Uma bateria de testes computacionais é realizada sobre um conjunto de instâncias geradas aleatoriamente para este fim. Os resultados obtidos são comparados ao do modelo original apresentando, como esperado, o total de tráfego roteado praticamente inalterado porém sem sobrecarga de nós na rede física. Além disso, duas versões da meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedures*) foram propostas para a resolução do problema estudado. A comparação da qualidade das soluções das versões do GRASP são comparadas entre si baseada em um limite inferior proposto neste trabalho.

**PALAVRAS CHAVE.** Otimização. Redes ópticas. Projeto da topologia virtual.

# *Abstract*

The evolution of Internet was followed by a growing demand for higher transmission rates in Wide-area networks, and the providers of such services, even today, make great efforts to optimize the resources available to meet this demand. The advancement of technology and communication devices using fiber optics makes possible to reconfigure the network topology, free from physical topology, which is called virtual topology. Establishing this topology can be modeled as a mixed integer optimization problem. We can cite the model of minimization of forwarded traffic network which seeks a virtual topology that minimizes the total processing load on routers, which is currently the biggest bottleneck of communication networks, thus increasing the flow of network. We present an evaluation function for this model of complexity  $O(N^2)$ , which allows us to evaluate both the connectedness and its value of objective function, and can be used together with any meta-heuristics. We also do a small modification in the objective function of the original model, because in spite of minimizing the total traffic processed on the network, not the best way of distributing, causing an overload of some nodes of the physical network while others remain idle. A battery of tests is performed on a set of randomly generated instances for this purpose. The results obtained are compared to the original model showing, as expected, the total amount of traffic routed essentially unchanged but without overloading nodes in the physical network. In addition, two versions of the meta-heuristic GRASP (Greedy Randomized Adaptive Search Procedures) have been proposed to solve the problem studied. A comparison of the quality of the solutions of the versions of GRASP are compared to each other based on a lower bound proposed in this paper.

**KEYWORDS.** Optimization. Optical networks. Virtual Topology Design.

# *Conteúdo*

## **Lista de Figuras**

**Lista de Algoritmos** . . . . . p. 10

**1 Introdução** . . . . . p. 11

1.1 Projeto da topologia virtual . . . . . p. 13

1.2 Problema de roteamento de tráfego . . . . . p. 14

1.3 Problema de roteamento do comprimento de onda . . . . . p. 14

1.4 Problema de designação de comprimento de onda . . . . . p. 15

1.5 Redes ópticas . . . . . p. 15

1.6 Trabalhos anteriores . . . . . p. 20

1.7 Objetivos . . . . . p. 22

1.8 Organização da dissertação . . . . . p. 22

**2 Modelos de otimização para o projeto da topologia virtual e roteamento de tráfego** . . . . . p. 24

2.1 Definições . . . . . p. 24

2.2 Parâmetros . . . . . p. 25

2.3 Problema de minimização do congestionamento . . . . . p. 26

2.3.1 Modelo agregado de minimização do congestionamento . . . . . p. 29

2.4 Minimização do tráfego retransmitido . . . . . p. 30

2.4.1 Modelo agregado de minimização do tráfego retransmitido . . . . . p. 32

2.5 Limites inferiores . . . . . p. 33

<b>3</b>	<b>Modelo de minimização do tráfego retransmitido com distribuição de tráfego eficiente</b>	p. 37
3.1	Definições . . . . .	p. 38
3.2	Formulação . . . . .	p. 39
3.3	Limites para o $FT_{Max}$ . . . . .	p. 42
<b>4</b>	<b>GRASP - Greedy Randomized Adaptive Search Procedures</b>	p. 45
4.1	Construção da solução inicial . . . . .	p. 46
4.2	Refinamento da solução . . . . .	p. 48
4.3	Avaliação da solução . . . . .	p. 50
<b>5</b>	<b>Resultados computacionais</b>	p. 53
5.1	Instâncias . . . . .	p. 54
5.1.1	Número de nós . . . . .	p. 54
5.1.2	Grau lógico . . . . .	p. 54
5.1.3	Matriz de tráfego . . . . .	p. 54
5.2	Resultados computacionais exatos para os modelos de original e modificado .	p. 55
5.3	Resultados para o GRASP . . . . .	p. 61
<b>6</b>	<b>Conclusão</b>	p. 67
<b>7</b>	<b>Trabalhos futuros</b>	p. 69
	<b>Bibliografia</b>	p. 70



# *Lista de Figuras*

1.1	Taxa do crescimento do tráfego e de largura de banda da Internet entre os anos 2007 e 2011. Fonte: TeleGeography. . . . .	p. 12
1.2	Esquema de transmissão TDM. . . . .	p. 16
1.3	Esquema de transmissão FDM. . . . .	p. 17
1.4	<i>OADM (Optical add-drop multiplexer)</i> . . . . .	p. 18
1.5	Topologia Física e Topologia Virtual. . . . .	p. 19
2.1	Número de retransmissões das demandas com origem em 1. . . . .	p. 35
3.1	Cenários (a) e (b) de roteamento de tráfego para $FT_{Net} = 10$ . . . . .	p. 37
3.2	Tráfego retransmitido para cada nó em uma rede de 10 nós utilizando o modelo original, $\Delta_l = 4$ , $FT_{Net} = 134,5996$ . . . . .	p. 38
3.3	Tráfego retransmitido para cada nó em uma rede de 10 nós utilizando o modelo modificado, $\Delta_l = 4$ , $FT_{Net} = 134,5996$ . . . . .	p. 41
3.4	Exemplo de roteamento em que $FT_{Net} = FT_{Max}$ . . . . .	p. 43
4.1	Permuta dos enlaces $(s_1, d_1)$ e $(s_2, d_2)$ por $(s_1, d_2)$ e $(s_2, d_1)$ . . . . .	p. 49
4.2	Topologia desconexa causada pela permuta dos enlaces. . . . .	p. 50
4.3	Solução ótima para o roteamento de tráfego na otimalidade ( $FT_{Net} = 5$ ). . . . .	p. 51
4.4	Solução viável para o roteamento de tráfego ( $FT_{Net} = 15$ ). . . . .	p. 51
5.1	Tráfego retransmitido total da rede para cada tipo de matriz de tráfego: 0%, 25%, 50% e 75%. . . . .	p. 57
5.2	Tráfego retransmitido do nó mais sobrecarregado para cada tipo de matriz de tráfego: 0%, 25%, 50% e 75%. . . . .	p. 57
5.3	Tráfego retransmitido do nó mais sobrecarregado. . . . .	p. 59
5.4	Enlace mais congestionado da rede. . . . .	p. 60

5.5	Tempo de execução até encontrar a solução ótima. . . . .	p. 60
5.6	Tempo de execução até encontrar a solução ótima. . . . .	p. 61
5.7	Avaliações realizadas em um segundo para instâncias de dimensão 20. . . . .	p. 62
5.8	Avaliações realizadas em um segundo para instâncias de dimensão 30. . . . .	p. 63
5.9	Média do tempo computacional de construção da solução inicial. . . . .	p. 64
5.10	Qualidade da solução inicial e final utilizando as duas abordagens do GRASP.	p. 65
5.11	Qualidade da solução inicial e final utilizando as duas abordagens do GRASP.	p. 65

## *Lista de Algoritmos*

1	<i>Pseudo-código do loop principal do GRASP.</i>	p. 46
2	<i>Construção da solução inicial pelo Método 1.</i>	p. 47
3	<i>Construção da solução inicial pelo Método 2.</i>	p. 48
4	<i>Pseudo-código da Busca Local.</i>	p. 49
5	<i>Função de avaliação que realiza o cálculo do <math>FT_{Net}</math>.</i>	p. 52

# 1 *Introdução*

Nas últimas décadas houve um enorme avanço na tecnologia de redes de comunicações motivado principalmente pela sua disseminação global, que deu origem à Internet, o maior fenômeno da área de telecomunicações desde o advento do sistema telefônico. Hoje em dia, até mesmo as redes telefônicas convencionais e móveis trabalham sobre o suporte de redes digitais de comunicações que interligam cidades, países e continentes, permitindo a formação de redes corporativas privadas (sistemas bancários, industriais e de outras instituições privadas) e redes acadêmicas, como a Rede Nacional de Pesquisa no Brasil (RNP2). Outras aplicações integradas a redes de comunicações são bastante utilizadas atualmente, como a telemetria e o controle remoto de processos na área industrial, a telemedicina na área de saúde, e as teleconferências para comunicação pessoal e educação à distância, entre outras possibilidades.

O desenvolvimento e a interconexão global das redes de comunicações foram motivados pelo o aumento exponencial do número de usuários, cuja demanda por velocidade de transmissão de dados está exigindo hoje das operadoras de telecomunicações um enorme esforço no sentido de otimizar a utilização dos recursos disponíveis em suas redes. A figura 1.1 mostra o crescimento do tráfego e da largura de banda da Internet em relação ao ano anterior, considerando os anos 2007/2008, 2008/2009, 2009/2010 e 2010/2011.

Para suprir esta demanda, a transmissão de dados por fibras ópticas é hoje a melhor alternativa para implementação de enlaces de comunicação de alta velocidade, capazes de transmitir dados em taxas que chegam a trilhões de *bits* por segundo em um meio físico extremamente leve, compacto, confiável e imune a ruído, além de cobrir longas distâncias com baixa atenuação de sinal (AGRAWAL, 2000). Essas características tornou a fibra óptica a substituta ideal, e largamente difundida, dos cabos metálicos existentes em redes metropolitanas (*MAN - Metropolitan Area Network*) e de longas distâncias (*WAN - Wide Area Network*).

A velocidade de transmissão nas redes de comunicação era limitada pela banda de transmissão dos enlaces. Atualmente é limitada pela capacidade de processamento dos roteadores de tráfego. Esses elementos são responsáveis pelo processamento eletrônico dos pacotes de

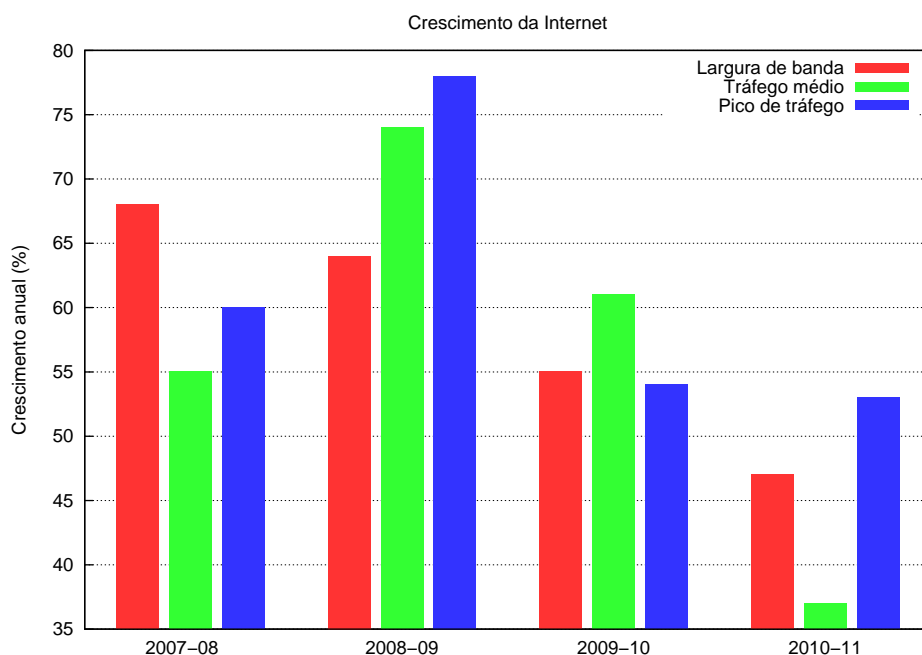


Figura 1.1: Taxa do crescimento do tráfego e de largura de banda da Internet entre os anos 2007 e 2011. Fonte: TeleGeography.

dados para seu redirecionamento pelos enlaces da rede até que alcance o destinatário. O modo que esse redirecionamento é realizado depende do protocolo de roteamento adotado, que pode variar de uma rede para outra.

No princípio, as tecnologias de transmissão via fibra óptica obrigavam que a cada nó houvesse conversão de sinal óptico para elétrico, para ser processado pelo roteador. Uma vez processado, o sinal era novamente convertido de elétrico para óptico e encaminhado para um enlace de saída. Neste cenário, todo o conteúdo da fibra sofre conversão OEO (óptico-eletróptico) e é processado, exigindo que o roteador seja capaz de processar um grande volume de pacotes em um pequeno intervalo de tempo. Contudo, como a banda de transmissão da fibra óptica é muito grande, o volume de pacotes é normalmente superior à capacidade de processamento dos roteadores, e essa falta de poder computacional é o atual gargalo da vazão das redes, acarretando na subutilização dos enlaces.

O desenvolvimento da tecnologia de multiplexação de comprimento de onda e de dispositivos ópticos, como emissores e receptores sintonizáveis, filtros, multiplexadores, demultiplexadores e chaves ópticas possibilitam que uma parte do trabalho de roteamento de tráfego seja transferida para a camada óptica da rede de comunicações, sendo possível, com um projeto de rede apropriado, poupar capacidade de processamento dos roteadores eletrônicos (RAMSWAMI; SIVARAJAN; SASAKI, 2009) e conseqüentemente aumentar a vazão da rede. Com esses dispositivos, que apresentaremos o funcionamento com mais detalhes na seção 1.5,

pode-se determinar uma nova topologia funcional para a rede, redefinindo a adjacência dos elementos em termos de roteamento. Essa nova topologia é denominada topologia virtual.

Com a possibilidade de reconfiguração da topologia da rede, determinar a topologia ótima e como aplicá-la pode ser modelado como um problema de otimização inteira mista, em que se deve levar em conta os limites impostos pela capacidade e número de dispositivos ópticos e eletrônicos para definir uma configuração de rede que privilegie o desempenho da transmissão de dados.

Por se tratar de um problema de complexidade computacional muito alta, *NP*-Completo (RAMASWAMI; SIVARAJAN, 1996), devido a parte inteira do problema, o projeto de redes ópticas é foco de atenção de muitos pesquisadores, no sentido de definir critérios e métodos que permitam obter a melhor solução possível para a configuração da rede (RAMASWAMI; SIVARAJAN; SASAKI, 2009; BIENSTOCK; GUENLUEK, 1995; RAMASWAMI; SIVARAJAN, 1996; MUKHERJEE et al., 1996; ALMEIDA et al., 2006; LIMA et al., 2010). A busca da solução ótima deste problema por métodos exatos ou enumerativos se mostra inviável justificando a utilização de heurísticas e meta-heurísticas (DUTTA; ROUSKAS, 2000; LEONARDI; MELLIA; MARSAN, 2000).

O projeto de redes ópticas é composto de quatro subproblemas distintos que serão listados e descritos sucintamente nas seções a seguir para que os objetivos apresentados na seção 1.7 se tornem mais claros.

## 1.1 Projeto da topologia virtual

O projeto da topologia virtual define qual será a configuração dos enlaces lógicos que interligam um número determinado de pares de nós da rede sem que haja processamento eletrônico de tráfego em nós intermediários, na topologia física, para efeitos de roteamento. Em outras palavras, resolver esse subproblema é definir qual será a topologia virtual da rede para um cenário de tráfego específico, esse normalmente representado na forma de uma matriz.

Dado um cenário de tráfego, isto é, uma matriz de tráfego, é possível afirmar que determinadas topologias são melhores que outras e, para essa avaliação, é utilizado o subproblema de roteamento de tráfego que será descrito na seção 1.2. Buscar a topologia ótima é um problema de otimização combinatória com  $N^2 - N$  variáveis binárias, onde  $N$  representa o número de nós da rede e cada variável representa o enlace que interconecta cada par de nós. Caso uma variável possua valor igual a 1, significa que o existe o enlace entre o par de nós, caso contrário (valor igual a 0), não há um enlace interligando-os. O grafo que representa esta topologia virtual é

direcionado, isto é, a existência de um enlace do nó  $i$  para  $j$  não implica na existência de um de  $j$  para  $i$ .

## 1.2 Problema de roteamento de tráfego

O problema de roteamento de tráfego normalmente é resolvido em conjunto com o projeto da topologia virtual para avaliar a qualidade das topologias, ou de modo independente, quando utilizado em conjunto com métodos aproximados ou heurísticos. Esse subproblema define como as demandas de tráfego, normalmente existentes entre todos os pares de nós da rede, serão distribuídas pelos enlaces lógicos de topologia pré-determinada. Ao final desse roteamento é possível apontar qual topologia apresenta melhor resultado dependendo do critério de avaliação utilizado, por exemplo, quantidade de tráfego retransmitido, nível de ocupação dos enlaces, etc.

Em um cenário real, essa tarefa seria realizada por algoritmo de roteamento, contudo, tais algoritmos muitas vezes não transmitem o tráfego da melhor maneira possível, o que poderia levar ao julgamento incorreto de uma topologia em questão. Para eliminar esse fator, normalmente utiliza-se modelos de programação linear, semelhante ao problema de fluxo de custo mínimo em grafos (AHUJA, 1988), para avaliar a topologia em um cenário ótimo de roteamento de tráfego. Por constituir a parte linear do problema estudado, o roteamento de tráfego é resolvido por algoritmos exatos devido ao baixo custo computacional.

## 1.3 Problema de roteamento do comprimento de onda

Uma vez estabelecida qual é a topologia ótima deve-se aplicá-la na topologia física existente. Se na topologia virtual há dois elementos adjacentes que não são adjacentes na topologia física deve-se definir por qual caminho essa ligação será realizada, isto é, quais elementos o enlace lógico atravessará até chegar ao destino. De modo mais formal, esse subproblema define o percurso físico dos comprimentos de onda pelos enlaces de fibra óptica.

O objetivo do roteamento de comprimentos de onda é alocar o caminho físico com menor atraso de propagação ou degradação de sinal, ou então buscando minimizar o número de comprimentos de onda necessários para implementação da rede, entre outras métricas. Geralmente é considerado um limite em relação ao número de entradas e saídas nas chaves ópticas situadas em cada nó da rede, denominado grau lógico.

## 1.4 Problema de designação de comprimento de onda

Esse subproblema tem como objetivo definir qual será o comprimento de onda associado a cada enlace lógico. Este subproblema possui relação direta com o problema de coloração de grafos, onde cada cor representa um comprimento de onda. O objetivo é minimizar a quantidade de cores necessárias na rede, obedecendo à restrição de não haver duas cores iguais no mesmo enlace físico (ZANG; JUE; MUKHERJEE, 2000).

Do mesmo modo que o roteamento de tráfego serve para a avaliação da topologia virtual, esse subproblema é utilizado para avaliar o cenário de roteamento de comprimento de onda. Diferente do primeiro, que é um modelo de programação linear, esse possui uma complexidade computacional elevada.

Para resolver esse subproblema como um problema de coloração de grafos deve-se assumir que cada enlace lógico da topologia virtual é um vértice do grafo, e todos os enlaces lógicos que compartilham algum enlace físico (resultado obtido após o roteamento do comprimento de onda) tornam-se adjacentes.

Com exceção dos trabalhos realizados em (KRISHNASWAMY; SIVARAJAN, 2001; ASSIS; GIOZZA; WALDMAN, 2005; LIMA et al., 2010) que buscam resolver esses subproblemas através dos modelos de otimização unificado, isto é, que aborda os quatro subproblemas em um único modelo de otimização, tais subproblemas normalmente são agrupados de dois a dois. O projeto da topologia virtual depende diretamente do roteamento de tráfego, sendo estes subproblemas resolvidos conjuntamente (*Virtual Topology Design*) em (RAMASWAMI; SIVARAJAN, 1996). Para uma melhor alocação de recursos da camada óptica da rede, resolve-se o roteamento e designação de comprimentos de onda (*Routing and Wavelength Assignment*) também de maneira integrada, como em (MUKHERJEE et al., 1996). Nesta dissertação são abordados somente os dois primeiros subproblemas.

Apresentamos a seguir algumas tecnologias que permitem que se defina uma topologia virtual de caminhos ópticos sobre a topologia física. Em seguida, os modelos de otimização elaborados ao *VTD (Virtual Topology Design)*, contribuições mais recentes ao tema, os objetivos e a organização dessa dissertação.

## 1.5 Redes ópticas

Nas redes ópticas o sinal transmitido na fibra óptica é uma onda eletromagnética, situada na faixa infravermelha do espectro luminoso, que se propaga através de sucessivas reflexões totais



nas paredes da fibra. Assim como outros meios de transmissão, guiados ou não, houve a necessidade da criação de métodos de multiplexação e demultiplexação de sinal. O objetivo principal de qualquer método de multiplexação é possibilitar que diversos sinais possam compartilhar o mesmo meio de transmissão sem que haja interferência de um sinal em outro, mantendo assim sua integridade. No destino ocorre o processo inverso, a demultiplexação, e os sinais são separados.

Podemos classificar, de modo grosseiro, os métodos existentes de multiplexação em duas modalidades: um que compartilham o meio físico utilizando janelas de tempo e em cada janela somente um sinal é transmitido, chamado *TDM* (*Time-division multiplexing*) (figura 1.2); e outro, que cada sinal é transmitido simultaneamente no tempo, porém em frequências diferentes, chamado *FDM* (*Frequency division multiplexing*) (figura 1.3). O método *FDM* é considerado uma técnica melhor que o *TDM*, uma vez que o último só viabiliza que vários sinais trafeguem no meio físico sem que ocorra interferência entre os sinais, contudo, não ocorre nenhum aumento em relação à banda de transmissão, isto é, o valor da capacidade permanecerá o mesmo e será dividido entre os vários sinais que desejam compartilhar o meio. Enquanto o *FDM* permite que mais de um sinal seja enviado pelo meio físico operando em diferentes frequências ao mesmo tempo, obtendo assim maiores velocidades de transmissão através.<sup>1</sup>

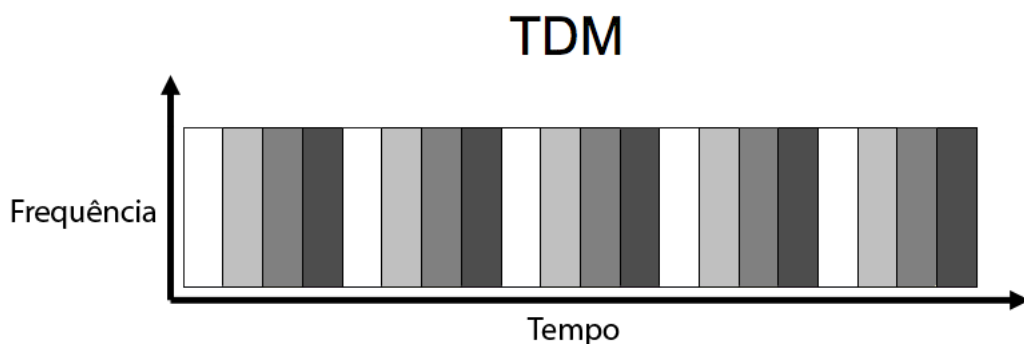


Figura 1.2: Esquema de transmissão TDM.

Nas redes ópticas a multiplexação de sinal é o *WDM* (*Wavelength-division multiplexing*) que é um caso específico do *FDM*. Afinal, a luz é uma onda eletromagnética assim como as utilizadas para realizar o *FDM*. No entanto, quando se fala em *FDM*, geralmente estamos tratando de frequências de rádio, e quando nos referimos à luz, tende-se a falar em comprimento de onda.

A principal vantagem desse método de multiplexação é devido ao fenômeno da difração presente nas ondas eletromagnéticas, permitindo que diversos comprimentos de onda de um

<sup>1</sup>Considerando que haja largura de banda suficiente no meio, que é o caso da fibra.

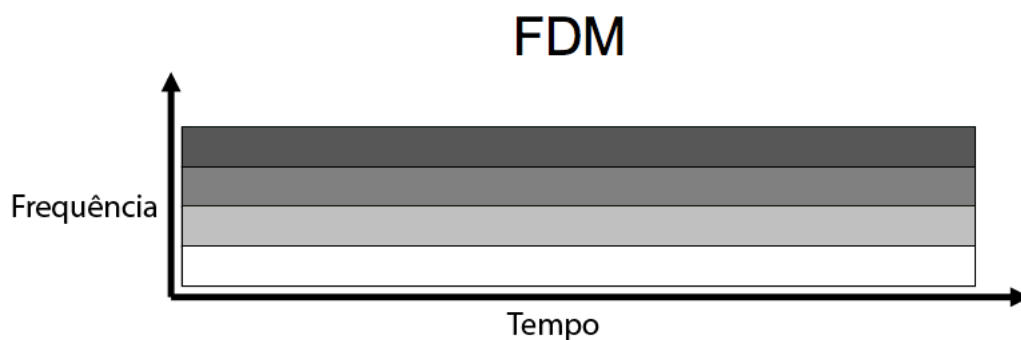


Figura 1.3: Esquema de transmissão FDM.

mesmo feixe luminoso pudessem ser separados para serem manipulados separadamente. Esse fato permitiu multiplexar os sinais a serem enviados através da fibra óptica sem qualquer tipo de processamento eletrônico e, de modo análogo, demultiplexá-lo.

Desse princípio surgiram outras tecnologias de divisão por comprimento de onda, como o *DWDM* (*Dense Wavelength-division multiplexing*), e o *CWDM* (*Coarse Wavelength-division multiplexing*), sendo que a diferença entre elas são basicamente a densidade de comprimentos de onda e a largura de banda de canal. Os sistemas *DWDM* são utilizados principalmente em redes *WAN* de alta capacidade, enquanto que os sistemas *CWDM* são utilizados tipicamente em redes *MAN*, por serem mais simples e baratos.

As primeiras implementações de sistemas ópticos de alta capacidade de transmissão que utilizavam o *WDM* foram ponto-a-ponto, com o objetivo de aumentar a capacidade de transmissão por fibra. Com enlaces comportando altas taxas de transmissão, os roteadores tornaram-se um fator limitante para a vazão das redes ópticas pela falta de poder computacional suficiente para processar a quantidade de tráfego que um enlace óptico pode transmitir.

Com o desenvolvimento tecnológico dos dispositivos utilizados em redes ópticas, logo surgiram comutadores ópticos, como *OADMs* (*Optical Add and Drop Multiplexers*) e *OXC* (*Optical Cross-Connectors*), permitindo roteamento de tráfego na camada óptica. Nesse ponto as redes ópticas deixaram de ser uma tecnologia de enlace de alta velocidade para serem uma tecnologia de rede de alta velocidade (BASTOS, 2006). Atualmente o *WDM* visa tornar o meio óptico mais eficiente, possibilitando não apenas o aumento da banda sem a alteração da infraestrutura, mas também possibilita uma maior flexibilidade aos sistemas de fibras ópticas como discutiremos adiante.

O *OADM* é um dispositivo óptico capaz de inserir (*add*) um ou mais comprimentos de onda e retirar (*drop*) os mesmos de uma fibra, deixando os outros comprimentos de ondas livres para atravessarem sem que haja uma conversão óptico-elétrica (OE). Por outro lado, nos

sistemas *WDM* ponto-a-ponto convencionais, todos os comprimentos de onda tinham terminações no fim de cada fibra, o que fazia a topologia virtual coincidir com a topologia física em termos dos nós fonte e destino de cada enlace. Pode-se ver na figura 1.4 o funcionamento de um *OADM* retirando e adicionando um comprimento de onda, simbolizado na cor vermelha, da fibra óptica. Os dados recebidos podem, através do roteador eletrônico do nó, ser retirados da rede definitivamente (*drop*), ou então retransmitidos (*forward*) de volta junto com outros dados no comprimento de onda adicionado (*add*). A conversão é realizada somente em alguns canais preestabelecidos, no caso da figura 1.4, o que possui a cor vermelha, enquanto os demais comprimentos de onda não são processados eletronicamente pelo nó, que retirou apenas o comprimento de onda desejado.

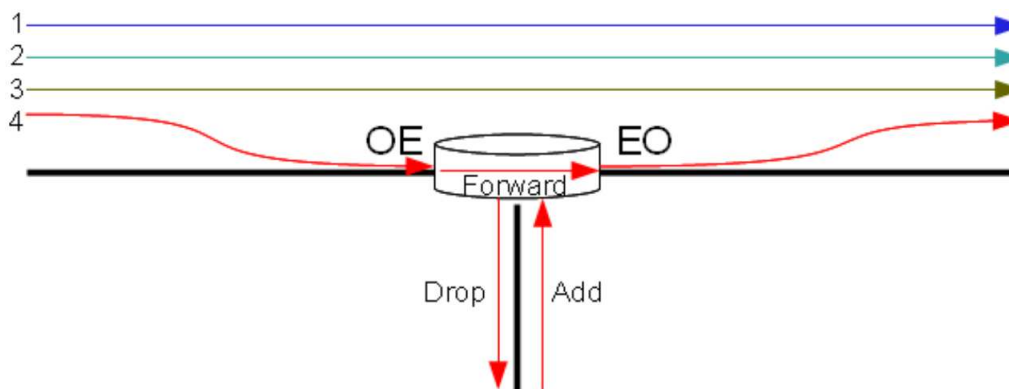


Figura 1.4: *OADM* (*Optical add-drop multiplexer*).

Outro elemento fundamental usado na arquitetura de uma rede óptica é o *OXC*. Esse dispositivo pode conectar sinais ópticos a portas de entrada e saída quaisquer. Normalmente através do fenômeno de reflexão, não necessitando de realizar conversão OE do sinal óptico para realizar comutação de comprimentos de onda. Consequentemente, também não há nenhum processamento eletrônico de tráfego envolvido nesse processo. A junção desses dispositivos e tecnologias citados permite que seja estabelecida uma topologia virtual de forma independente da topologia física, como se pode ver na figura 1.5.

Embora a topologia física apresentada na figura 1.5 tenha configuração de anel, a topologia virtual tem configuração de malha, totalmente conectada, onde cada nó da rede possui comprimentos de onda interligando-o diretamente com todos os outros nós da rede. A cada um desses comprimentos de onda damos o nome de enlace lógico e a quantidade de enlaces lógicos que chegam e saem de cada nó é o seu grau lógico. Ao contrário do encontrado na bibliografia de teoria dos grafos, no que tange o grau de dígrafos, o grau de entrada e saída é contabilizado de modo independente e, quando referencia-se ao grau de determinado vértice o valor é a soma de ambos – será mantido a terminologia encontrada em (RAMASWAMI; SIVARAJAN, 1996), e

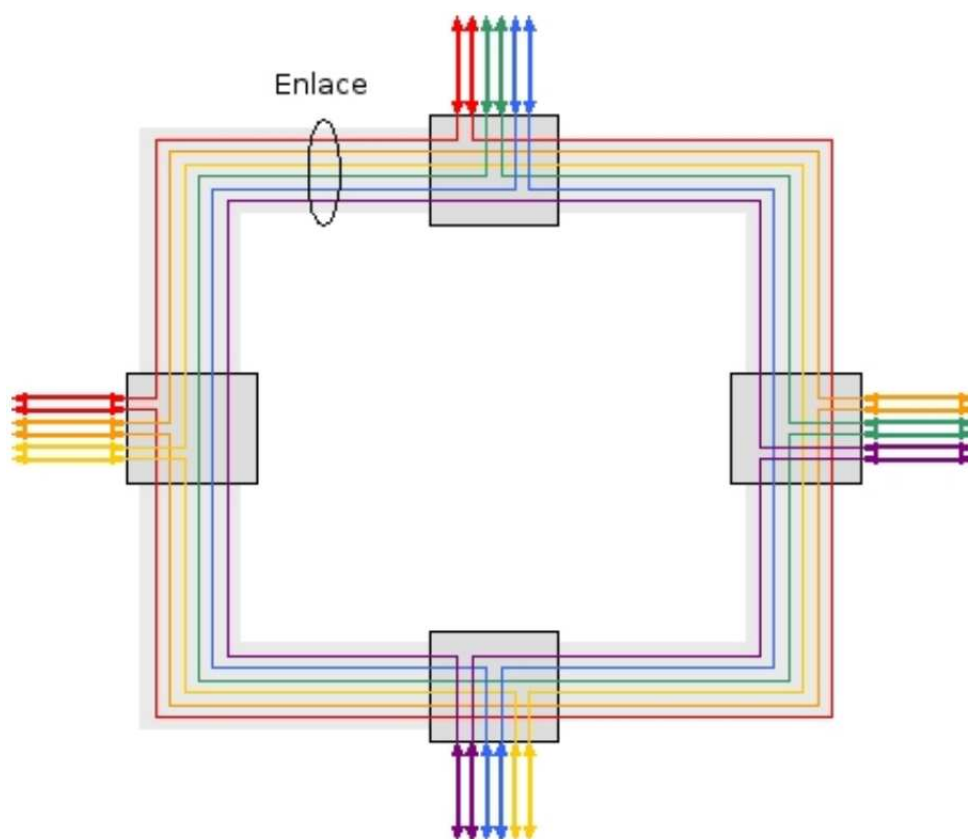


Figura 1.5: Topologia Física e Topologia Virtual.

seguido por muitos outros pesquisadores, que utiliza o termo grau como o maior entre o grau de entrada e saída do vértice. No exemplo 1.4, o grau lógico é unitário, enquanto que na figura 1.5, o grau lógico é 3 (três).

A esse tipo de rede apresentado na figura 1.5 dá-se o nome de rede totalmente transparente, uma vez que não ocorre nenhuma conversão OEO. Em contrapartida, uma rede que sofre conversão OEO em todos os enlaces é denominada opaca. Redes completamente transparentes são ideais, tanto devido ao custo de processamento de eletrônicos de pacotes, quanto ao custo associado ao resolver os subproblemas de projeto da topologia virtual e roteamento de tráfego. Ambos, nesse caso, são resolvidos trivialmente pois a solução sempre é um dígrafo completo. Contudo, redes totalmente transparentes nem sempre são uma alternativa viável, ora limitado pelo número de comprimentos de onda disponíveis – por exemplo, em uma fibra que utiliza *DWDM* permite no máximo 80 comprimentos de onda – ora devido à limitação do número de transceptores presentes nos *OADMs* da rede. Esse trabalho, assim como vários outros disponíveis na literatura, aborda redes semitransparentes, o grau de transparência depende do grau lógico, que é um parâmetro limitador ao projeto da topologia virtual como veremos na seção seguinte.

A proposição de topologias lógicas em redes ópticas originou um novo campo de estudo, a busca da topologia virtual ótima dadas as condições de tráfego de uma rede. Iniciaram também discussões sobre topologias virtuais dinâmicas, pois as demandas de tráfegos não são constantes (GENCATA; MUKHERJEE, 2002). Dois grandes desafios nesse campo são a elaboração de métodos computacionalmente eficientes que consigam determinar uma topologia em tempo hábil, isto é, antes que o cenário de tráfego se altere, e a adaptação da topologia virtual em uma nova topologia com o mínimo de reconfigurações necessárias (BANERJEE; MUKHERJEE, 2000).

## 1.6 Trabalhos anteriores

O problema de redes ópticas pode ser dividido em dois subproblemas, o *VTD* (*Virtual Topology Design*), que consiste no projeto da topologia virtual e roteamento de tráfego, e *RWA* (*Routing and Wavelength Assignment*) que é composto pelo roteamento do comprimento de onda e designação dos comprimentos de onda. Ambos já foram amplamente estudados em trabalhos anteriores e, nessa seção, descrevemos algumas abordagens utilizadas.

Em (MUKHERJEE et al., 1996) apresentaram uma formulação para o *VTD* como um problema de otimização com função objetivo não linear composta pelo congestionamento dos enlaces e o atraso de transmissão. Esse trabalho foi uma importante contribuição para o estudo desse problema pois alguns elementos da terminologia são utilizados até hoje, além da padronização da divisão do problema nos quatro subproblemas descritos nas seções 1.1, 1.2, 1.3 e 1.4. Além das contribuições citadas, os autores propuseram, nesse mesmo trabalho, a meta-heurística *Simulated Annealing* para resolver o problema do projeto da topologia virtual. No mesmo ano, (RAMASWAMI; SIVARAJAN, 1996), desenvolveu uma formulação inteira mista para o *VTD* (descrita na seção 2.3) no qual pode-se agregar as demandas de tráfego, e propôs o método de *lower-bound*, o *iLB* (*iterative Lower Bound*), que gera resultados mais próximos da solução ótima que qualquer outro método encontrado na literatura para o modelo proposto. Contudo, é um método caro computacionalmente. Nesse mesmo trabalho, foram apresentadas também quatro heurísticas para o problema, comparando os resultados destas com um método de geração aleatória. Dentre as heurísticas apresentadas podemos destacar a heurística *HLDA* pela sua simplicidade de implementação, velocidade de execução, além de obter resultados de qualidade satisfatória nos casos em que as demandas de tráfegos da rede não são homogêneas, verificado em (LIMA et al., 2007).

O número de formulações que buscam resolver tanto o *VTD* quanto o *RWA* de forma in-

tegrada vem crescendo entre os pesquisadores. O artigo (ZANG; JUE; MUKHERJEE, 2000), apresenta em detalhes o *RWA*, propõe formulações para esse problema com e sem restrições de continuidade de comprimentos de onda, isto é, se ao atravessar um nó, há a obrigatoriedade, ou não, do enlace lógico permanecer com o mesmo comprimento de onda. Além disso, o autor realiza uma revisão das abordagens utilizadas em trabalhos anteriores, sejam exatos ou heurísticos, para resolver esse problema.

De modo semelhante, em (LEONARDI; MELLIA; MARSAN, 2000), os autores revisam diferentes métodos de solução do *VTD*, apresentando alguns resultados comparativos entre diferentes heurísticas. Uma outra excelente revisão bibliográfica do projeto de redes ópticas pode ser encontrada em (DUTTA; ROUSKAS, 2000).

Em (ALMEIDA et al., 2006), os autores apresentaram um modelo de otimização inteira mista para o *VTD* que minimiza o tráfego retransmitido da rede. É importante ressaltar que esse é o maior limitador à vazão das redes atualmente. Devido as características da formulação, nenhum método de *lower bound* proposto anteriormente ao *VTD* se adapta a esse modelo. Somente em (LIMA et al., 2010), um método adequado foi proposto para esse modelo de otimização. Além disso, (LIMA et al., 2010) propõe um modelo de otimização que aborda os quatro subproblemas de forma integrada seguindo a linha da abordagem encontrada em (KRISHNASWAMY; SIVARAJAN, 2001).

Há também na bibliografia um vasto estudo utilizando modelagens de topologia dinâmicas, isto é, que se alteram dados a medida que as demandas de tráfego também o fazem. Em (BANERJEE; MUKHERJEE, 2000) são apresentadas formulações matemáticas que conseguem minimizar o número de reconfigurações necessárias para transformar uma topologia virtual em outra. Vale ressaltar que isso pode ser aplicado desde que seja possível transformar uma topologia na outra, e portanto, aplicável a qualquer modelo de otimização para o *VTD*. Em (RAMAMURTHY; RAMAKRISHNAN, 2000), os autores propõe uma formulação de reconfiguração que tem como critérios tanto o atraso de propagação dos pacotes quanto o mínimo de reconfigurações. Em (GENCATA; MUKHERJEE, 2002) é apresentado um algoritmo de reconfiguração da topologia baseado no conhecimento dos padrões de tráfego em redes de *backbones* ao longo do dia, além de reagir em tempo hábil no caso de eventuais desvios, obtendo resultados bastante satisfatórios.

## 1.7 Objetivos

O objetivo principal desse trabalho é a proposição de um modelo de otimização para o projeto da topologia virtual que minimize a carga de processamento eletrônico da rede, como o modelo em (ALMEIDA et al., 2006), contudo, o modelo apresentado neste trabalho não sobrecarrega nenhum elemento da rede. Como o modelo proposto é basicamente o modelo de minimização do tráfego retransmitido (ALMEIDA et al., 2006) com algumas adições em sua formulação, o chamaremos de modelo modificado, e o modelo de minimização do tráfego retransmitido de (ALMEIDA et al., 2006) será chamado de modelo original.

Como objetivos secundários têm-se a adaptação do método de *lower bound*, encontrado em (LIMA et al., 2010), para o modelo modificado, e a elaboração de uma função de avaliação de baixo custo computacional para calcular o total de tráfego retransmitido na topologia virtual. A elaboração desta função de avaliação elimina a necessidade de resolvidores lineares, estes são responsáveis por grande parte do tempo de execução das meta-heurísticas e, devido ao custo computacional reduzido do método proposto, é possível uma maior exploração do espaço de busca.

Além disso, duas versões da meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedures*) foram propostas para a resolução do problema estudado. A comparação da qualidade das soluções das versões do GRASP são comparadas entre si baseada *lower bound* proposto neste trabalho.

## 1.8 Organização da dissertação

A dissertação foi organizada de modo a facilitar o entendimento das contribuições. O capítulo 2 introduz os modelos de otimização encontrado na literatura relevantes a este trabalho, são eles, o modelo de minimização do congestionamento (RAMASWAMI; SIVARAJAN, 1996) e o modelo de minimização do tráfego retransmitido (ALMEIDA et al., 2006). Ao final do capítulo 2 é descrito o procedimento de obtenção de limites inferiores apresentado em (LIMA et al., 2010) para o modelo de minimização do tráfego retransmitido (ALMEIDA et al., 2006).

No capítulo 3 é apresentado o modelo modificado para o projeto da topologia virtual, que busca uma divisão de tráfego que não sobrecarrega nenhum elemento da rede. Nesse mesmo capítulo propomos uma adaptação do método de geração de *lower bound*, proposto em (LIMA et al., 2010), para utilizá-lo em conjunto com o modelo modificado.

Nesse trabalho também foi implementado a meta-heurística GRASP, utilizando dois mé-

todos para a construção da solução inicial, e na etapa de avaliação foi utilizada a função de avaliação proposta neste trabalho para tal tarefa. O capítulo 4 é dedicado ao *GRASP* e à função de avaliação, apresentando a ideia geral da meta-heurística, os dois procedimentos de construção da solução inicial - caracterizando as duas versões do *GRASP* - e os detalhes da função de avaliação, ressaltando seu baixo custo computacional.

No capítulo 5 são exibidos os procedimentos utilizados na criação das instâncias testadas, os resultados computacionais para os modelos de otimização - modelo original e modificado - e para o *GRASP*. Devido a complexidade dos problemas abordados, os testes para os modelos exatos são realizados somente para pequenos casos.

Nos capítulos 6 e 7, respectivamente, são apresentadas as conclusões obtidas da dissertação e discutidas propostas de trabalhos futuros.



## 2 *Modelos de otimização para o projeto da topologia virtual e roteamento de tráfego*

Estabelecer a topologia virtual pode ser modelado como um problema de otimização inteira mista, ou seja, existe uma parte inteira, o projeto da topologia virtual, e uma parte linear, o roteamento de tráfego. Os modelos de otimização, quando executados para resolver as duas partes em conjunto, tem um custo computacional que cresce rapidamente com o aumento do número de nós da rede devido à parte inteira, sendo que a segunda parte possui um custo computacional reduzido. Um grande número de trabalhos na literatura utilizam somente a parte linear em conjunto com heurísticas e meta-heurísticas para avaliar a qualidade das topologias obtidas (DUTTA; ROUSKAS, 2000).

Nesta seção serão abordados dois modelos de otimização para o *VTD* (*Virtual Topology Design*), o primeiro, encontrado em (RAMASWAMI; SIVARAJAN, 1996) que, além da importância histórica – pois é a primeira formulação do projeto da topologia virtual como um problema de otimização inteira mista – também há contribuições diretas aos modelos formulados posteriormente pois compartilham várias restrições deste, e o segundo, proposto em (ALMEIDA et al., 2006) é o modelo no qual esse trabalho se fundamenta.

Essa seção está dividida em cinco partes: a terminologia utilizada para descrever o problema, descrição dos parâmetros, uma seção para cada um dos modelos acima citados e a abordagem do método de *lower bound* proposto em (LIMA et al., 2010).

### 2.1 Definições

- $n$  representa o número de nós da rede.
- $b_{ij} \in \{0, 1\}$  representa o enlace que parte de  $i$  e tem como destino  $j$  na topologia virtual. Se  $b_{ij} = 1$  existe um enlace lógico, ou caminho óptico, de  $i$  para  $j$ , caso contrário,  $b_{ij} = 0$ ,

tal enlace não está presente na topologia. Como normalmente pode haver enlaces entre quaisquer pares de nós, a exceção de quando  $i = j$ , essas variáveis normalmente são representada na forma de uma matriz de adjacência com diagonal nula.

- $d_{ij} \in \mathbb{R}^+$  representa o atraso de propagação de sinal no enlace lógico do nó  $i$  ao  $j$ .
- $d_{max}$  é definido como o maior  $d_{ij}$ , ou seja, valor de maior atraso de propagação de sinal entre quaisquer elementos da topologia física.
- $\alpha \in \mathbb{R}^+$  parâmetro que funciona como um fator para tornar inviáveis as soluções para o roteamento de tráfego que possuam um atraso de propagação acima desse corte.
- $\lambda_{ij}^{sd} \in \mathbb{R}^+$  representa a parcela de tráfego que sai de  $s$  e tem como destino final  $d$  presente no enlace  $(i, j)$ .
- $\lambda_{ij} = \sum_{sd} \lambda_{ij}^{sd}$ , ou seja,  $\lambda_{ij}$  é definido como o total de tráfego que atravessa o enlace  $(i, j)$  independente de fonte e destino.
- $\lambda^{sd} = \sum_{ij} \lambda_{ij}^{sd}$ , de modo análogo,  $\lambda^{sd}$  representa todo o tráfego que tem  $s$  como origem e  $d$  como destino.
- $\lambda_{max}$  é definido como o maior  $\lambda_{ij}^{sd}$ .<sup>1</sup>
- $\Delta_l \in \mathbb{N}^+$  é o parâmetro que representa o grau lógico, isto é, a conectividade máxima que um determinado nó pode ter, isto é, o número máximo de enlaces lógicos que podem chegar e partir de cada nó.
- $FT_j$  representa todo o tráfego que chega ao nó  $j$  e não tem ele como destino final, necessitando ser processado para propósitos de roteamento.<sup>2</sup>
- $FT_{Net}$  é definido como o custo total de processamento devido a retransmissão de tráfego, em outras palavras, o somatório de todos os  $FT_j$ .<sup>2</sup>

## 2.2 Parâmetros

Nessa subseção descreveremos os parâmetros utilizados nos modelos de otimização, são eles: a matriz de tráfego da rede, o grau lógico, a matriz de atraso de propagação da topologia física e o parâmetro  $\alpha$ . Nesse trabalho, assim como em (RAMASWAMI; SIVARAJAN, 1996)

<sup>1</sup>Definição utilizada no modelo proposto em (RAMASWAMI; SIVARAJAN, 1996).

<sup>2</sup>Utilizada no modelo encontrado em (ALMEIDA et al., 2006).

e (ALMEIDA et al., 2006), a restrição de atraso de propagação foi relaxada,  $\alpha = \infty$ , tornando qualquer solução válida para o roteamento de tráfego em termos de atraso de propagação. Portanto, não há necessidade de utilizar parâmetros de atraso de propagação.

Um dos motivos dessa relaxação, além de tornar mais simples o processo de roteamento de tráfego, é a dificuldade de estabelecer os valores para cada  $d_{ij}$  que se referem ao atraso de propagação dos caminhos ópticos. Para obter esses valores é necessário resolver o roteamento do comprimento de onda e determinar o percurso na topologia física que cada caminho óptico percorrerá, sendo então  $d_{ij}$  igual a soma dos atrasos de propagação de cada enlace físico que o caminho óptico atravessa.

O grau lógico é necessário pois age como um limitador e impede que os modelos sempre gerem uma topologia totalmente transparente, sendo esta, a melhor topologia possível em termos de retransmissão. Por exemplo, quando o valor do grau lógico é 1, os algoritmos procuram a melhor topologia que possua somente um enlace partindo e chegando em cada nó. Esta solução será uma topologia em anel, semelhante ao Problema do Caixeiro Viajante (FLOOD, 1956).

A matriz de tráfego é o parâmetro que estabelece as demandas de tráfego entre cada par  $s$  e  $d$  da rede. Em outras palavras, contém os valores de todos os  $\lambda^{sd}$ . Podemos ver um exemplo de uma matriz de tráfego para uma rede com 5 nós a seguir.

$$\Lambda = \begin{bmatrix} 0 & \lambda^{12} & \lambda^{13} & \lambda^{14} & \lambda^{15} \\ \lambda^{21} & 0 & \lambda^{23} & \lambda^{24} & \lambda^{25} \\ \lambda^{31} & \lambda^{32} & 0 & \lambda^{34} & \lambda^{35} \\ \lambda^{41} & \lambda^{42} & \lambda^{43} & 0 & \lambda^{45} \\ \lambda^{51} & \lambda^{52} & \lambda^{53} & \lambda^{54} & 0 \end{bmatrix}$$

## 2.3 Problema de minimização do congestionamento

O critério de otimização mais utilizado nos trabalhos previamente publicados é a minimização do congestionamento (RAMASWAMI; SIVARAJAN, 1996). Os modelos que utilizam este critério minimizam a soma das demandas que passam pelo enlace mais ocupado da rede, dividindo-as por outros enlaces até atingir um valor mínimo de congestionamento para o enlace mais ocupado. É importante ressaltar que na época que esse modelo foi proposto, o congestionamento dos enlaces era um gargalo significativo à vazão da rede, contudo, hoje não mais representa um fator limitante tão grande quanto o processamento eletrônico de pacotes.

### Função objetivo

A função objetivo especifica o critério de otimalidade da solução. A qualidade da solução do modelo de minimização do congestionamento está relacionada com a carga do enlace mais congestionado da rede. A solução ótima será aquela que possuir a ocupação mínima do enlace mais congestionado da rede.

$$\text{Minimize } \lambda_{max} \quad (2.3.1)$$

### Restrições

Conservação de tráfego

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, & s = i \\ -\lambda^{sd}, & d = i \\ 0, & s \neq i, \quad d \neq i \end{cases}, \forall s, d, i = 1, \dots, n \quad (2.3.2)$$

Fluxo total dos enlaces lógicos

$$\lambda_{ij} = \sum_{s,d} \lambda_{ij}^{sd}, \forall i, j = 1, \dots, n \quad (2.3.3)$$

$$\lambda_{ij} \leq \lambda_{max}, \forall i, j = 1, \dots, n \quad (2.3.4)$$

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}, \forall i, j, s, d = 1, \dots, n \quad (2.3.5)$$

Restrição de atraso médio da demanda

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d_{max} \quad (2.3.6)$$

Restrições de grau lógico

$$\sum_i b_{ij} = \Delta_l, \forall j = 1, \dots, n \quad (2.3.7)$$

$$\sum_j b_{ij} = \Delta_i, \forall i = 1, \dots, n \quad (2.3.8)$$

Restrições de não negatividade e integralidade

$$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0, \forall i, j, s, d = 1, \dots, n \quad (2.3.9)$$

$$b_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (2.3.10)$$

A equação 2.3.2 define a restrição de conservação de tráfego e impõem que a quantidade de tráfego associada às demandas entre cada par de nós permaneça inalterada durante todo o processo de roteamento, garantindo que a diferença entre o que chega e o que sai de cada nó assuma os valores que conservem o fluxo das demandas de tráfego. Todos os nós intermediários de uma demanda arbitrária terão a diferença de zero, enquanto que a fonte e o destino de cada demanda terão valores em módulo igual a  $\lambda^{sd}$ .

A equação 2.3.3 é a restrição que define o tráfego total de cada enlace lógico, isto é, o congestionamento, como a soma de todas as componentes que atravessam um enlace no sentido (i,j) não levando em consideração a origem ou destino daquele tráfego, enquanto a equação 2.3.4, significa que a soma de todas as componentes de tráfego de um enlace não deve exceder o valor de  $\lambda_{max}$ . Como a formulação busca minimizar o valor de  $\lambda_{max}$ , esse sempre irá assumir valor igual ao do enlace mais congestionado.

A equação 2.3.5 tem duas implicações: a primeira é permitir que seja transmitido uma demanda pelo enlace lógico (i,j) se somente se existir tal enlace,  $b_{ij} = 1$ , pois quando esse for nulo para algum (i,j) todos os  $\lambda_{ij}^{sd}$ , para o mesmo (i,j), também serão; a segunda implicação garante que todas as componentes da demanda tenham seu valor menor ou igual a demanda que a originou, sendo esta subdividida ou não durante o roteamento.

As restrições de atraso, formuladas na equação 2.3.6, impedem que o tráfego seja roteado por um caminho que tenha o atraso maior que  $\alpha d_{max}$ . Relaxando a restrição, fazendo  $\alpha = \infty$ , todas as componentes de tráfego  $\lambda_{ij}^{sd}$  podem ser roteadas por qualquer percurso.

As equações 2.3.8 e 2.3.7 são as restrições de grau lógico que limitam o número de enlaces que chegam e partem de cada nó da rede.

As equações 2.3.9 e 2.3.10 são, respectivamente, restrições de não-negatividade para  $\lambda_{ij}$  e de integralidade 0-1 da variável  $b_{ij}$ .

### 2.3.1 Modelo agregado de minimização do congestionamento

Além do modelo proposto em (RAMASWAMI; SIVARAJAN, 1996) descrito em detalhes na seção anterior, os pesquisadores propuseram no mesmo trabalho um modelo agregado de minimização do congestionamento para melhorar o desempenho computacional. Este modelo diminui o número de variáveis e restrições do problema. O modelo original identifica cada demanda por um par de índices (s,d), contudo, é possível agrupar essas demandas somente pela sua origem. Essas modificações podem ser aplicadas também ao modelo de otimização descrito em (ALMEIDA et al., 2006), e também ao modelo que será proposto no capítulo 3. Sejam  $\lambda^s = \sum_d \lambda^{sd}$  o tráfego total que parte de  $s$  e  $\lambda_{ij}^s$  o tráfego que tem origem em  $s$  que passa pelo enlace (i,j).

#### Função objetivo

$$\text{Minimize } \lambda_{max} \quad (2.3.1.1)$$

#### Restrições

Conservação de tráfego

$$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s, & s = i \\ -\lambda^{si}, & s \neq i \end{cases} \quad \forall s, i = 1, \dots, n \quad (2.3.1.2)$$

Fluxo total dos enlaces lógicos

$$\lambda_{ij} = \sum_s \lambda_{ij}^s, \quad \forall i, j = 1, \dots, n \quad (2.3.1.3)$$

$$\lambda_{ij} \leq \lambda_{max}, \quad \forall i, j = 1, \dots, n \quad (2.3.1.4)$$

$$\lambda_{ij}^s \leq b_{ij} \lambda^s, \quad \forall i, j, s = 1, \dots, n \quad (2.3.1.5)$$

Restrição de atraso médio da demanda

$$\sum_{i,j} \lambda_{ij}^s d_{ij} \leq \lambda^s \alpha d_{max} \quad (2.3.1.6)$$

Restrições de grau lógico

$$\sum_i b_{ij} = \Delta_l, \forall j = 1, \dots, n \quad (2.3.1.7)$$

$$\sum_j b_{ij} = \Delta_l, \forall i = 1, \dots, n \quad (2.3.1.8)$$

Restrições de não negatividade e integralidade

$$\lambda_{ij}^s, \lambda_{ij}, \lambda_{max} \geq 0, \forall i, j, s = 1, \dots, n \quad (2.3.1.9)$$

$$b_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (2.3.1.10)$$

As descrições das equações 2.3.1.1 a 2.3.1.10 são análogas ao modelo original.

## 2.4 Minimização do tráfego retransmitido

Em (ALMEIDA et al., 2006) os autores propõem o modelo de minimização do tráfego retransmitido, que minimiza o processamento de tráfego total da rede. Cada nó deve processar a demanda que o tenha como destino e, mesmo em uma rede totalmente conectada, esse processamento é inevitável. Contudo, em redes que possuem algum roteamento na camada eletrônica, o que é o caso mais comum, deve-se processar também os pacotes que chegam ao nó e não o tem como destino, para que sejam retransmitidos. Esse modelo tem como objetivo minimizar a quantidade de tráfego retransmitido eletronicamente na rede, sendo tal quantidade denominada  $FT_{Net}$  (*Network Forwarded Traffic*).

**Função objetivo**

$$\text{Minimize } FT_{Net} \quad (2.4.1)$$

**Restrições**

Conservação de tráfego

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, & s = i \\ -\lambda^{sd}, & d = i \\ 0, & s \neq i, d \neq i \end{cases}, \forall s, d, i = 1, \dots, n \quad (2.4.2)$$

Limitação de fluxo

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}, \forall i, j, s, d = 1, \dots, n \quad (2.4.3)$$

Restrições de grau lógico

$$\sum_i b_{ij} = \Delta_l, \forall j = 1, \dots, n \quad (2.4.4)$$

$$\sum_j b_{ij} = \Delta_l, \forall i = 1, \dots, n \quad (2.4.5)$$

Definição de tráfego multi-salto

$$FT_j = \sum_{s,d} \sum_i \lambda_{ij}^{sd}, d \neq j, \forall j = 1, \dots, n \quad (2.4.6)$$

Definição de tráfego retransmitido

$$FT_{Net} = \sum_j FT_j \quad (2.4.7)$$

Restrições de não negatividade e integralidade

$$\lambda_{ij}^{sd}, FT_j \geq 0, \forall i, j, s, d = 1, \dots, n \quad (2.4.8)$$

$$b_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (2.4.9)$$

Todas restrições, exceto as equações 2.4.6 e 2.4.7 são iguais as do modelo original proposto em (RAMASWAMI; SIVARAJAN, 1996). A definição de tráfego multi-salto (equação 1.6.4.6) denota o tráfego retransmitido, que é todo tráfego que chega ao nó e não o tem como destino,  $d \neq j$ , devendo ser processado e retransmitido. A definição do tráfego retransmitido (equação



1.6.4.7), denominado de  $FT_{Net}$ , é o somatório de todos os tráfegos multi-salto. As mesmas modificações aplicadas ao modelo de otimização do congestionamento para torná-lo agregado também podem ser aplicadas a esse.

### 2.4.1 Modelo agregado de minimização do tráfego retransmitido

Embora o modelo agregado se assemelhe bastante com o original, é necessário uma modificação na definição do tráfego multi-salto (equação 2.4.6) para o funcionamento correto. Tal modificação é bem intuitiva pois, ao agregar um modelo perde-se o índice  $d$ , relativo ao destino de cada demanda. Com isso, não se pode mais afirmar se o tráfego que chegou a um determinado nó não o tem como destino para ser totalizado no  $FT_{Net}$ . Há duas formas de contornar esse problema: a primeira é calcular todo o tráfego que chega ao nó no  $FT_j$ . Dessa forma o  $FT_j$  será igual a  $\sum_{s,d} \sum_i \lambda_{ij}^{sd}$ , que pode ser dividido em duas parcelas:  $\sum_{s,d} \sum_i \lambda_{ij}^{sd} = A + B$  onde  $A = \sum_{s,d} \sum_i \lambda_{ij}^{sd}, d = j$  e  $B = \sum_{s,d} \sum_i \lambda_{ij}^{sd}, d \neq j$ . O termo  $B$  é a definição original para o  $FT_{Net}$ , enquanto o termo  $A$  é relativo a todo o tráfego que tem aquele nó como destino. Como mencionado anteriormente, é um processamento inevitável e ocorrerá mesmo em uma rede completamente transparente. Pode-se observar que todo o trabalho de otimização ocorre sobre o termo  $B$ , em termos práticos, a solução obtida será a mesma, porém, a função objetivo será incrementada de  $A$ . A segunda forma, um pouco mais elegante, é a redefinição do tráfego multi-salto: toda demanda que parte de um nó e não teve esse como origem é fruto de uma retransmissão. A diferença da segunda abordagem é que o modelo agregado contém toda informação necessária, índices  $s$  e  $i$ , para realizar a distinção se uma demanda foi ou não originada naquele nó.

#### Função objetivo

$$\text{Minimize } FT_{Net} \quad (2.4.1.1)$$

#### Restrições

Conservação de tráfego

$$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s, & s = i \\ -\lambda^{si}, & s \neq i \end{cases}, \forall s, i = 1, \dots, n \quad (2.4.1.2)$$

Limitação de fluxo

$$\lambda_{ij}^s \leq b_{ij} \lambda^s, \forall i, j, s = 1, \dots, n \quad (2.4.1.3)$$

Restrições de grau lógico

$$\sum_i b_{ij} = \Delta_l, \forall j = 1, \dots, n \quad (2.4.1.4)$$

$$\sum_j b_{ij} = \Delta_l, \forall i = 1, \dots, n \quad (2.4.1.5)$$

Definição de tráfego multi-salto

$$FT_i = \sum_s \sum_j \lambda_{ij}^s, s \neq i, s \neq j, \forall i = 1, \dots, n \quad (2.4.1.6)$$

Definição de tráfego retransmitido

$$FT_{Net} = \sum_j FT_j \quad (2.4.1.7)$$

Restrições de não negatividade e integralidade

$$\lambda_{ij}^{sd}, FT_j \geq 0, \forall i, j, s, d = 1, \dots, n \quad (2.4.1.8)$$

$$b_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (2.4.1.9)$$

As simplificações realizadas no modelo de minimização do tráfego retransmitido para torná-lo agregado são análogas às do modelo de minimização do congestionamento, com exceção da definição do tráfego retransmitido.

## 2.5 Limites inferiores

Ao trabalhar com problemas de otimização combinatória ocasionalmente nos deparamos com problemas mais complexos, com maior quantidade de variáveis, de modo que se torna inviável encontrar uma solução exata dentro de uma janela de tempo aceitável. Para contornar isso, uma alternativa é utilizar métodos heurísticos que resolvem o problema em tempo hábil

em detrimento da solução exata.

Uma dificuldade ao se trabalhar com métodos heurísticos é avaliar a qualidade da solução encontrada, pois não se conhece *a priori* a solução ótima para realizar uma comparação e determinar o quão próximos são os resultados. Duas abordagens são encontrada na literatura, (a) comparando os resultados com outros métodos heurísticos, mostrando qual método tem um melhor desempenho, embora não se pode afirmar nada quanto a proximidade da solução ótima; (b) gerando limites inferiores, ou *lower bounds*, relaxando o problema através da relaxação lagrangeana (FISHER, 1981), da relaxação das variáveis inteiras do problema tornando-o linear, ou aplicando algoritmos específicos para o problema em questão (LIMA et al., 2010). Esses métodos de obtenção de limites inferiores geram um valor, não uma solução, que serve como um ótimo teórico, sempre melhor ou igual ao real, o que possibilita aferir a qualidade de métodos aproximados.

Em (RAMASWAMI; SIVARAJAN, 1996), junto com o modelo de minimização do congestionamento, foi proposto um método iterativo para a obtenção do *lower bound* através da relaxação das variáveis inteiras do problema, chamado *iLB*. Embora haja uma grande similaridade entre os modelos de (RAMASWAMI; SIVARAJAN, 1996) e (ALMEIDA et al., 2006), os métodos desenvolvidos para obtenção de limites inferiores para um modelo não se aplica ao outro. De fato, ao relaxar as variáveis  $b_{ij}$  no modelo de minimização do tráfego retransmitido, o valor do  $FT_{Net}$  será nulo pois, sempre será possível atribuir um valor pequeno o suficiente para cada  $b_{ij}$  de modo a criar uma topologia sem a necessidade de retransmissão. Somente em (LIMA et al., 2010), foi proposto um algoritmo específico para o modelo de minimização do tráfego retransmitido, (ALMEIDA et al., 2006), que será apresentado nessa seção. Uma prova mais detalhada pode ser encontrada em (LIMA et al., 2010).

Quando se conhece a topologia em que deve rotear o tráfego, o procedimento de cálculo do tráfego retransmitido torna-se mais simples, para isso, basta determinar a menor distância entre cada par de nós  $(s, d)$  e contabilizar a quantidade de vértices que a demanda atravessou partindo de  $s$  até chegar em  $d$ . Por fim, multiplica-se o valor da demanda  $\lambda^{sd}$  pela quantidade de retransmissões. Ao realizar esse procedimento para todas as demandas, obtém-se o valor total do tráfego retransmitido para uma dada topologia. Contudo, para obter o *lower bound* é necessário conhecer o menor valor do tráfego retransmitido para qualquer topologia viável, isto é, dentro das restrições de grau lógico. Como tal topologia não é conhecida faz-se necessário o uso de algumas considerações para chegar a esse resultado.

O procedimento proposto por (LIMA et al., 2010) consiste de duas etapas, cada uma ignorando uma das restrições de grau lógico e obtendo a solução ótima nessas condições. Ao

ignorar uma das restrições, o problema se torna mais simples. Pode-se afirmar também que todas as soluções viáveis anteriormente assim permanecem, pois ao remover uma restrição do problema as soluções continuam a atender todas restrições do problema (além da excluída). As soluções geradas podem ter valores menores que a solução ótima do problema original. Como dito, o método então consiste em encontrar a solução ótima para cada etapa, cada um ignorando uma restrição de grau lógico, e como explicado, ambas soluções terão valores de função objetivo menores, ou no pior caso, iguais a solução ótima do problema original. E como o objetivo é encontrar o valor que mais se aproxima do ótimo real, o valor retornado é o pior dentre os encontrados.

Também é intuitivo que pode-se rotear diretamente, isto é, sem que haja processamento, uma quantidade de demandas igual ao valor do grau lógico. A figura 2.1 ilustra uma topologia de grau lógico 1 (anel) onde se pode observar os números de retransmissões das demandas com origem em 1. De 1 para 2, o número de retransmissões é 0, de 1 para 3, o número é 2 e assim por diante até observar que de 1 para 6 houve a necessidade de 4 retransmissões.

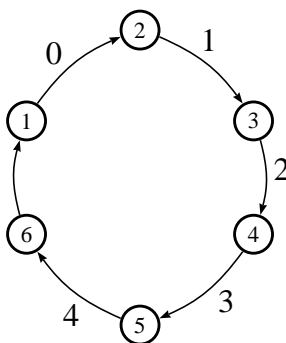


Figura 2.1: Número de retransmissões das demandas com origem em 1.

Para  $\Delta_l \geq 1$ , o raciocínio é análogo. Tomando novamente um nó qualquer como exemplo, do qual parte  $\Delta_l$  enlaces, pode-se concluir que  $\Delta_l$  demandas não irão sofrer roteamento. É importante observar que a partir de cada nó saem  $\Delta_l$  enlaces resultando em  $\Delta_l$  elementos adjacentes a cada elemento da rede, consequentemente,  $\Delta_l$  demandas livres de retransmissões,  $\Delta_l^2$  demandas com uma retransmissão,  $\Delta_l^3$  com duas, etc. De modo genérico, teremos  $\Delta_l^{r+1}$  demandas que sofrem  $r$  retransmissões.

Dito isso, para calcular o *lower bound* proposto em (LIMA et al., 2010) as maiores demandas devem sofrer menos retransmissões, pois caso contrário impactaria em uma maior soma. Para isso, deve-se ordenar, em ordem decrescente, as demandas que partem de cada nó, e após essa etapa, as  $\Delta_l$  maiores demandas de cada nó serão transmitidas diretamente da origem até

o destino, ou seja, sem retransmissão. Das demandas restantes, as  $\Delta_l^2$  maiores sofrerão uma retransmissão e serão somadas ao  $FT_{Net}$  uma única vez; o processo é repetido, considerando um número de retransmissões cada vez maiores até que todas as demandas tenha sido roteadas. Pode observar que essa topologia viola as restrições do grau lógico de entrada, pois não há verificação se o nó destino pode ou não receber mais enlaces. O mesmo processo é feito para o tráfego que chega em cada nó, porém desta vez ignorando as restrições de grau lógico de saída e, ao final os resultados são comparados e o maior deles é retornado, pois, como ambos os resultados são um *lower bound* o maior deles se aproximará mais do ótimo. No capítulo 3 é apresentada uma versão modificada desse algoritmo adaptada para o modelo proposto nessa dissertação.

### 3 *Modelo de minimização do tráfego retransmitido com distribuição de tráfego eficiente*

O principal objetivo do modelo de minimização do tráfego retransmitido é a capacidade de poupar a carga de processamento na rede como um todo, contudo, na função objetivo da formulação (equação 2.4.1) não há distinção de como esse valor total de processamento é dividido pelos elementos da rede. Na figura 3.1 temos dois cenários para o roteamento de tráfego. No cenário (a) dois elementos possuem  $FT_j = 5$ , enquanto que no cenário (b) um possui  $FT_j = 0$  e outro  $FT_j = 10$ . O  $FT_{Net}$  será igual a 10 e ambas as soluções serão vistas como equivalentes em termos de qualidade, como na figura 3.1 (a) e (b) respectivamente. Como não há ganho em relação à função objetivo no cenário (a), a distribuição do valor total de processamento de forma mais igualitária não ocorre.

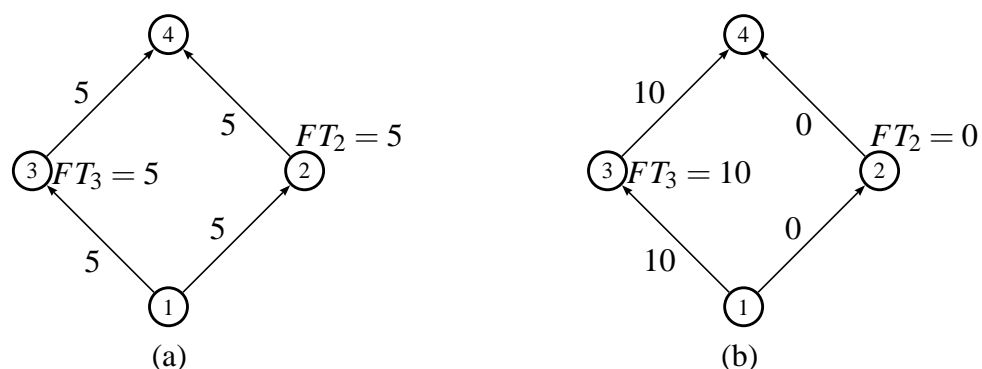


Figura 3.1: Cenários (a) e (b) de roteamento de tráfego para  $FT_{Net} = 10$ .

Em um cenário real o comportamento do cenário (b) é indesejado, cada roteador tem uma capacidade máxima de processamento e a sobrecarga de alguns elementos acabará por limitar a vazão, pois, quando estes atingirem o limite de sua capacidade de processamento impedirá que mais tráfego seja roteado deixando os demais nós da rede subutilizados. Na figura 3.2 pode-se ver que essa divisão não é realizada de modo igualitária para o modelo de minimização

do tráfego retransmitido (ALMEIDA et al., 2006).

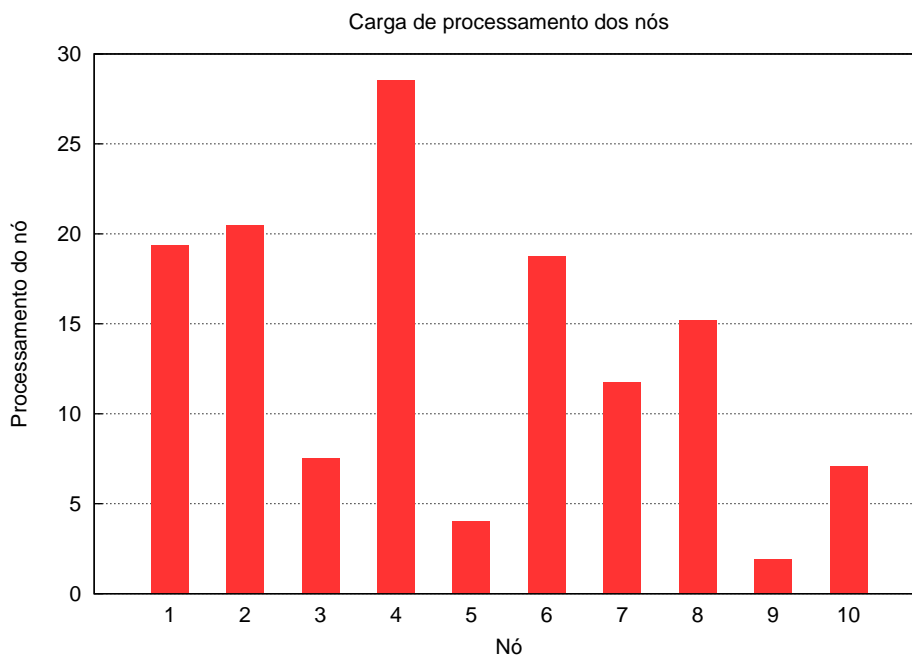


Figura 3.2: Tráfego retransmitido para cada nó em uma rede de 10 nós utilizando o modelo original,  $\Delta_l = 4$ ,  $FT_{Net} = 134,5996$ .

De modo geral, o modelo proposto nesta dissertação, chamado de modelo modificado, tem como objetivo eliminar esse comportamento existente no modelo de minimização do tráfego retransmitido original fazendo com que as demandas sejam distribuídas de modo a não sobrecarregar nenhum nó da rede. Para conseguir esse resultado, é necessário realizar algumas adições no modelo original (ALMEIDA et al., 2006), contudo, essas modificações tornam o modelo mais caro computacionalmente. Nas seções seguintes, são apresentadas as definições complementares necessárias para a formulação do modelo modificado, a formulação e os limites inferiores e superiores para a variável  $FT_{Max}$  (definida na seção 3.1).

### 3.1 Definições

- $FT_{Max}$  é definido como o maior  $FT_j$  da rede.
- $FT_{Min}$  é definido como o menor  $FT_j$  da rede. Essa variável não existe na formulação do modelo modificado, contudo, será utilizada na demonstração do limite inferior de  $FT_{Max}$  (seção 3.2).
- $\alpha \in [0, 1]$  é parâmetro que multiplica, na função objetivo, a contribuição do  $FT_{Net}$ .
- $\beta \in [0, 1]$  é parâmetro que multiplica, na função objetivo, a contribuição do  $FT_{Max}$ .

## 3.2 Formulação

### Função objetivo

$$\text{Minimize } \alpha \cdot FT_{Net} + \beta \cdot FT_{Max} \quad (3.2.1)$$

### Restrições

Conservação de tráfego

$$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd}, & s = i \\ -\lambda^{sd}, & d = i \\ 0, & s \neq i, d \neq i \end{cases}, \forall s, d, i = 1, \dots, n \quad (3.2.2)$$

Limitação de fluxo

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}, \forall i, j, s, d = 1, \dots, n \quad (3.2.3)$$

Restrições de grau lógico

$$\sum_i b_{ij} = \Delta_l, \forall j = 1, \dots, n \quad (3.2.4)$$

$$\sum_j b_{ij} = \Delta_l, \forall i = 1, \dots, n \quad (3.2.5)$$

Definição de tráfego multi-salto

$$FT_j = \sum_{s,d} \sum_i \lambda_{ij}^{sd}, \forall j = 1, \dots, n \quad (3.2.6)$$

Definição de tráfego retransmitido

$$FT_{Net} = \sum_j FT_j \quad (3.2.7)$$

Definição do tráfego máximo



$$FT_{Max} \geq FT_j, \forall j = 1, \dots, n \quad (3.2.8)$$

Restrições de não negatividade e integralidade

$$\lambda_{ij}^{sd}, FT_j \geq 0, \alpha \in [0, 1], \beta \in [0, 1] \forall i, j, s, d = 1, \dots, n \quad (3.2.9)$$

$$b_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (3.2.10)$$

$$\alpha + \beta = 1 \quad (3.2.11)$$

Quando  $\beta = 0$ , o modelo executa exatamente o modelo de minimização do tráfego retransmitido original, pois o segundo termo da função objetivo (equação 3.2.1) é nulo, enquanto, que para valores maiores, para  $\beta$ , o  $FT_{Max}$  impactará na qualidade da solução e será minimizado.

Ao analisar a figura 3.1 novamente, considerando  $\beta > 0$ , pode-se concluir que a primeira configuração resultará em um valor menor de função objetivo, pois em ambos os casos o  $FT_{Net} = 10$ , contudo, o  $FT_{Max} = 5$  no cenário (a) é menor que do segundo, o que caracteriza uma melhor solução para a formulação apresentada. Qualquer valor maior que zero para  $\beta$ , já produz esse efeito, porém caso  $\beta$  seja próximo de zero, o tráfego será distribuído de modo que o  $FT_{Net}$  fique praticamente inalterado em relação ao modelo original, podendo impossibilitar uma distribuição eficiente. Enquanto que para valores maiores, especialmente quando  $\beta = 1$ , a prioridade torna-se a divisão igualitária, mesmo que o  $FT_{Net}$  resultante seja maior. A figura 3.3 apresenta a solução do exemplo da figura 3.2 utilizando o modelo modificado. Pode-se ver como a distribuição da carga de processamento foi dividida de forma homogênea entre os nós da rede.

A formulação não permite a possibilidade de  $\beta < 0$  (equação 3.2.9), pois ocorrerá um efeito oposto ao pretendido, ou seja, o  $FT_{Max}$  será maximizado, e como não há na formulação uma restrição que limite superiormente o valor do  $FT_{Max}$ , tem-se que  $FT_{Max} \rightarrow +\infty$ , levando o valor de função objetivo para  $-\infty$ . Ainda que o  $FT_{Max}$  fosse limitado superiormente pelo limite teórico, apresentado na seção seguinte, impedindo que a função objetivo fosse levada para  $-\infty$ , o  $FT_{Max}$  ainda seria maximizado. Basicamente há duas maneiras de se maximizar, ou minimizar o  $FT_{Max}$ , a primeira delas se dá ao subdividir uma demanda por vários caminhos alternativos até atingir seu destino final, enquanto a segunda ocorre pela atribuição de diferentes demandas às diferentes rotas. Quando minimizamos, ou maximizamos, o  $FT_{Max}$  ambos os efeitos ocor-

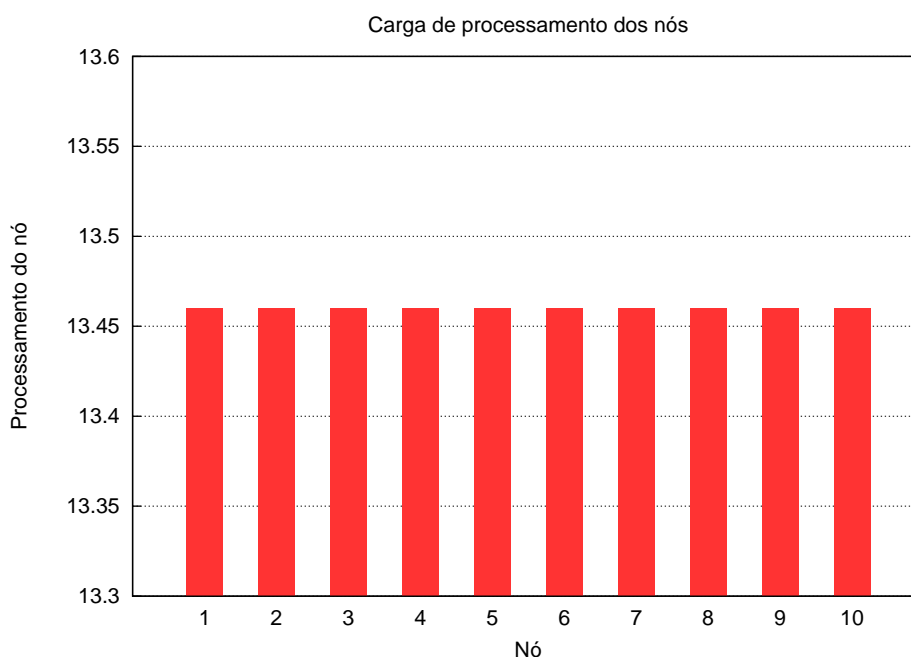


Figura 3.3: Tráfego retransmitido para cada nó em uma rede de 10 nós utilizando o modelo modificado,  $\Delta_l = 4$ ,  $FT_{Net} = 134,5996$ .

rem, isto é, ao minimizar,  $\beta \geq 0$ , as demandas tanto serão realocadas por diferentes rotas e cada demanda poderá se subdividir entre as várias rotas para sanar o problema de sobrecarga; e ao maximizar,  $\beta < 0$ , as demanda tenderão a não se bifurcarem por diversos caminhos, o que é desejável para determinados tipos de aplicações, contudo, o fator também ocorre na finalidade de maximizar o  $FT_{Max}$ , ou seja, várias demandas não bifurcadas serão alocadas para atravessar um mesmo nó para aumentar ao máximo o valor do  $FT_j$  daquele nó. Por propósitos de simplificação neste trabalho, esses casos não foram considerados, para isso,  $\alpha$  e  $\beta$  foram definidos com uma combinação linear convexa (equação 3.2.11) equilibrando a contribuição de cada termo da função objetivo.

Uma outra consideração importante em relação ao modelo proposto é em relação à definição de tráfego multi-salto. Quando essa definição é igual ao do modelo original, considerando  $\beta > 0$ , o modelo balanceará a carga de modo a minimizar todos os tráfegos retransmitidos, porém neste caso não se leva em consideração a carga de recebimento dos nós. É sabido que o processamento desse recebimento é inevitável, por isso, o modelo original o desconsidera. Contudo, quando é analisado do ponto de vista de balanceamento da carga de processamento dos nós, provavelmente cada nó possui sua própria carga de processamento inevitável diferente e a distribuição perfeitamente homogênea do tráfego retransmitido irá manter essas diferenças. Por exemplo, considere dois nós A e B que possuem uma carga de processamento devido ao recebimento de pacotes de respectivamente 4 e 10 (cargas de processamento inevitáveis), e todo

o processamento devido a retransmissão seja dividido de forma homogênea com o valor de 6. Os custos de processamento finais de cada nó seriam 10 e 16, ou seja, balancear somente considerando as cargas de retransmissão mantém as diferenças do processamento inevitável. Uma abordagem muito mais eficiente para a distribuição se dá quando não é feita essa distinção entre o tráfego recebido e o retransmitido, distribuindo menos carga para os nós que já estão inevitavelmente mais sobrecarregados que outros, e mais cargas para os demais a fim de diminuir essas diferenças.

A formulação do modelo modificado com as demandas de tráfego agregada se difere muito pouco do modelo de minimização do tráfego retransmitido em sua versão original agregada, bastando substituir a função objetivo do problema pela equação 3.2.1, adicionar a restrição definida na equação 3.2.8 ao problema, e adicionar os parâmetros  $\alpha$  e  $\beta$  nas restrições de não negatividade. Devido a essa similaridade será omitida a versão agregada do modelo modificado.

### 3.3 Limites para o $FT_{Max}$

Na presente formulação, a função objetivo é alterada em relação ao modelo original. Nesta sessão é mostrado que o  $FT_{Max}$  não é maior que o  $FT_{Net}$  e, portanto o valor de função objetivo do modelo modificado será sempre menor ao do modelo original. Neste contexto, existe a necessidade de um novo método de cálculo de limite inferior dedicado ao modelo modificado proposto nessa dissertação.

O método de geração de *lower bound* proposto em (LIMA et al., 2010) para o modelo de minimização do tráfego retransmitido (ALMEIDA et al., 2006) quando aplicado no modelo modificado não se mostra tão eficiente, principalmente nos casos que  $\beta$  assume um valor alto. Nesses casos o valor retornado pelo método citado pode superar o valor do ótimo.

Uma solução possível é adotarmos o método apresentado em (LIMA et al., 2010) e obter o limite inferior para o  $FT_{Net}$ , então multiplicamos este valor por  $\alpha$  utilizado no modelo modificado. Dessa forma é adquirido um novo limite inferior mas ainda inadequado ao modelo modificado. De fato, como  $\beta \cdot FT_{Max} \geq 0$ , logo  $\alpha \cdot FT_{Net} \leq \alpha \cdot FT_{Net} + \beta \cdot FT_{Max}$ . Observe que a medida que  $\beta$  aumenta, a importância da segunda parcela torna-se mais expressiva, acarretando em limites mais distantes do valor ótimo do modelo modificado. Isto indica a necessidade de adequar o método proposto em (LIMA et al., 2010) para o modelo modificado.

Como dito no capítulo 2, o método de *lower bound* formulado em (LIMA et al., 2010) obtém o limite inferior para o  $FT_{Net}$ . Nesta seção será demonstrado os limites inferiores e superiores para  $FT_{Max}$  em função do  $FT_{Net}$ .

As definições dos limites superiores e inferiores para  $FT_{Max}$  são apresentadas a seguir. Entende-se por  $FT_{Net}$  a soma do tráfego retransmitido de todos os nós da rede e  $FT_{Max}$ , a quantidade de tráfego retransmitido do elemento que mais retransmitiu tráfego. Pela (equação 3.2.7) temos:

$$FT_{Net} = \sum_j FT_j$$

Como o  $FT_{Max}$  por definição é o maior dos  $FT_j$ , podemos separá-lo do somatório.

$$FT_{Net} = \sum_{j \neq Max} FT_j + FT_{Max}$$

Subtraindo  $\sum_{j \neq Max} FT_j$  de ambos os lados da equação, temos:

$$FT_{Net} - \sum_{j \neq Max} FT_j = FT_{Max}$$

A restrição de não negatividade impõe que  $FT_j \geq 0$ , logo o  $FT_{Max}$  será máximo quando  $\sum_{j \neq Max} FT_j = 0$ , e nesse caso  $FT_{Max} = FT_{Net}$ . Portanto, o limite superior para o  $FT_{Max}$  é o próprio  $FT_{Net}$  e na figura 3.4, pode-se ver um exemplo em que todo o tráfego retransmitido na rede é realizado pelo nó 4.

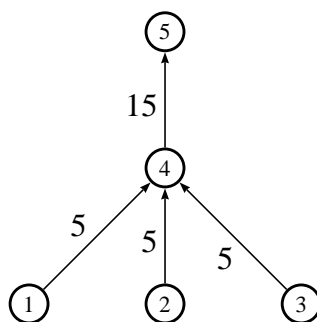


Figura 3.4: Exemplo de roteamento em que  $FT_{Net} = FT_{Max}$ .

Com relação ao limite inferior, temos por definição que  $FT_{Max}$  é o maior dentre todos os  $FT_j$  e de modo análogo,  $FT_{Min}$  representa o menor  $FT_j$ . Tem-se  $FT_{Max} \geq FT_j \geq FT_{Min}, \forall j = 1, \dots, n$ , e portanto, o menor valor que  $FT_{Max}$  pode assumir é o próprio  $FT_{Min}$ .

Quando todos os  $n$  elementos da rede possuírem o mesmo  $FT_j$ , é fácil ver que

$$FT_{Max} = FT_j = FT_{Min}, \forall j = 1, \dots, n. \quad (3.4.1)$$

Neste caso o tráfego total retransmitido na rede será igual a

$$FT_{Net} = \sum_j FT_j = FT_j \cdot n.$$

Isolando o valor de  $FT_j$  temos

$$FT_j = FT_{Net}/n, \forall j = 1, \dots, n. \quad (3.4.2)$$

Pela equação 3.4.1,  $FT_{Max} = FT_j, \forall j = 1, \dots, n$ , e podemos substituir na equação 3.4.2 obtendo

$$FT_{Max} = FT_{Net}/n \quad (3.4.3)$$

concluindo que o  $FT_{Max}$  é limitado inferiormente por  $FT_{Net}/n$ . Multiplicando  $\beta \geq 0$  ambos os lados da equação 3.4.3 tem-se o segundo termo da função objetivo (equação 3.2.1). Portanto

$$\beta \cdot FT_{Max} = \beta \cdot FT_{Net}/n. \quad (3.4.4)$$

Como o limite inferior de  $FT_{Max}$  está relacionado com  $FT_{Net}$ , é possível estimar o valor do  $FT_{Net}$  utilizando o método definido em (LIMA et al., 2010). Seja  $F\bar{T}_{Net}$  tal limite, sendo assim, o valor do limite inferior do modelo modificado é determinado por  $\alpha \cdot F\bar{T}_{Net} + \beta \cdot F\bar{T}_{Net}/n$ .

## 4 *GRASP - Greedy Randomized Adaptive Search Procedures*

As meta-heurísticas são uma boa alternativa para a solução de problemas de otimização combinatória que possuem um grande número de variáveis e restrições. Neste trabalho foi escolhida a meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) (FEO; RESENDE, 1995) para resolver instâncias de dimensões maiores, apresentados no capítulo seguinte.

Inicialmente o GRASP foi proposto para a solução do problema de cobertura de conjuntos (FEO; RESENDE, 1989) e desde então tem sido adaptada para a solução de diversos problemas, entre eles: Problema do Caixeiro Viajante (MOTTA; OCHI; MARTINHON, 2001), satisfabilidade (RESENDE; THOMAS, 1996), coloração de grafos (LAGUNA; MARTÍ, 2001), dentre outros, comprovando sua eficácia em diferentes problemas.

Uma das características que diferencia o GRASP de outras meta-heurísticas encontradas na literatura é o fato deste primar na construção inicial assegurando que tenha uma boa qualidade, utilizando a etapa de refinamento da solução para obter alguma melhoria, em contraste com outros métodos, como o *simulated annealing* (LAARHOVEN; AARTS, 1987), algoritmos genéticos (HAUPT; HAUPT; WILEY, 2004), dentre outros, que normalmente iniciam a busca com uma solução qualquer e concentram todos os esforços na etapa de refinamento. Abaixo, no Algoritmo 1, podemos ver o pseudo-código do *loop* principal da meta-heurística GRASP.

Com essa filosofia em mente, este capítulo é dividido em três partes condicionada aos passos do algoritmo 1: construção da solução (topologia) inicial, método de refinamento e procedimento utilizado na avaliação das soluções encontradas.

**Algoritmo 1:** Pseudo-código do loop principal do GRASP.

```

1 Enquanto condição de parada não for satisfeita faça
2   Passo Solução inicial
3   | solução ← const_aleatoria_gulosa_adaptativa();
4   Passo Refinamento
5   | solução ← busca local(solução);
6   Passo Atualização
7   | Se solução é a melhor solução até então conhecida então
8   |   atualize(solução);
9 Retorne a solução encontrada

```

## 4.1 Construção da solução inicial

Na construção da solução inicial, o GRASP busca gerar uma solução que possua uma melhor qualidade e, no caso dos modelos de otimização abordados neste trabalho, significa um menor tráfego retransmitido.

O algoritmo de construção da solução inicial, como o próprio nome da heurística sugere, é construtivo, guloso, adaptativo e aleatório. O fator de aleatoriedade é controlado por um dos parâmetros fornecidos ao algoritmo, nomeado de  $\alpha$  (RESENDE; THOMAS, 1996). De modo geral, a construção da solução inicia sem nenhum elemento presente na solução e, a cada passo, um elemento da Lista Restrita de Candidatos (LRC) é sorteado e inserido na solução. A LRC é uma lista que contém todos os elementos que podem ser inseridos na solução de modo que não torne a solução inviável. É importante ressaltar que a lista é mantida ordenada, com o elemento mais promissor no primeiro índice e o menos promissor no final da lista. Para o problema de projeto da topologia virtual cada elemento, neste caso, representa um enlace lógico,  $b_{ij}$ , responsável por interconectar um par de nós na topologia virtual.

O sorteio de qual elemento deve ser inserido na solução é regulado pelo parâmetro  $\alpha$ , que limita quais elementos participam do sorteio. Por exemplo, se  $\alpha = 1$ , todos os elementos da LRC podem ser escolhidos, enquanto  $\alpha = 0.1$  permite somente que os elementos presente nos primeiros 10% da lista sejam selecionados.

Ao adotar um  $\alpha$  pequeno, normalmente a solução inicial fica situada próxima a um mínimo local, e as soluções geradas entre diferentes iterações do *loop* principal tendem a serem próximas no espaço de busca. O oposto ocorre quando  $\alpha$  aumenta, pois as soluções passam a ficar com um alto grau de aleatoriedade tendendo a se espalharem ao longo do espaço de busca e mais distantes de um mínimo local. Nestes casos, os elementos menos promissores da LRC

terão mais chances de estarem na solução.

O ideal é que  $\alpha$  seja pequeno o suficiente para gerar soluções iniciais com qualidade satisfatória e grande o suficiente a ponto de permitir uma maior exploração do espaço de busca. Neste trabalho, dois métodos para a construção da solução inicial são propostos. O Algoritmo 2 mostra como esta etapa é realizada pelo primeiro método.

**Algoritmo 2:** *Construção da solução inicial pelo Método 1.*

**Dados:**  $\alpha$ , matriz de tráfego  $\Lambda$ ,  $\Delta_l$ ,  $n$  (dimensão da instância)

- 1 **Passo Inicializa a solução**
- 2   | Solução  $\leftarrow \emptyset$
- 3 **Enquanto Solução não estiver completa faça**
- 4   | Sortear um elemento  $i$ , ( $i \in \{1, \dots, n\}$ ).
- 5   | **Passo Construção da LRC**
- 6   |   | LRC  $\leftarrow$  determinar todos os  $b_{ij}$  que podem ser inseridos na solução sem violar as restrições de grau lógico;
- 7   |   | Ordenar a LRC em ordem decrescente utilizando o valor de  $\lambda^{ij}$ .
- 8   | **Passo Sorteio do elemento**
- 9   |   | Sortear um elemento da LRC pertencente ao intervalo  $[1, \text{Tamanho da LRC} \cdot \alpha]$ ;
- 10   |   | Adicionar o elemento sorteado na solução.
- 11 **Passo Retornar**
- 12   | Retornar a solução encontrada.

O Algoritmo 2 se assemelha muito com o critério utilizado pela heurística HLDA, apresentada em (RAMASWAMI; SIVARAJAN, 1996), pois o critério utilizado para definir qual elemento é melhor (linha 7) é o valor existente na matriz de tráfego. Priorizar a criação de enlaces entre os pares de nós que transmitem as maiores quantidades de tráfego faz com que essas demandas não necessitem mais de roteamento, portanto, tende a gerar valores pequenos para o  $FT_{Net}$ . Este critério, como já verificado em (LIMA et al., 2007) com a heurística HLDA, funciona bem nos casos em que há uma diferença grande entre os valores da matriz de tráfego, pois os valores não escolhidos terão pouco impacto para propósitos de roteamento por serem demandas pequenas. Para matrizes de tráfego com relativa uniformidade esse critério gera soluções piores, uma vez que não é analisado o impacto de outras demandas no roteamento.

Para comparar com o Método 1 mostrado no Algoritmo 2, um outro método de construção da solução inicial também é apresentada neste trabalho. Esta segunda proposta, chamada de Método 2, avalia o impacto de todas as demandas entre todos os pares de nós (fator de adaptatividade do GRASP) e, conseqüentemente, possui um custo computacional mais elevado que a primeira abordagem. Neste caso, a busca inicia com todos os elementos presente na solução e, a cada iteração, algum elemento é sorteado e removido até uma solução viável ser gerada.



Pode-se observar que desse modo a busca inicia, provavelmente, violando as restrições de grau lógico, uma vez que é assumida uma topologia completamente transparente. Os sucessivos sorteios na LRC, excluindo os enlaces, busca diminuir o valor do grau lógico de cada nó até que todos possuam um grau viável e, então, a solução encontrada é retornada. Esse método pode ser analisado no Algoritmo 3.

**Algoritmo 3:** *Construção da solução inicial pelo Método 2.*

**Dados:**  $\alpha$ , matriz de tráfego  $\Lambda$ ,  $\Delta_l$ ,  $n$  (dimensão da instância)

- 1 **Passo Inicializa a solução**
- 2      $Solução \leftarrow b_{ij}, \forall i, j = 1 \dots n$
- 3 **Enquanto** *Solução violar as restrições de grau lógico* **faça**
- 4     Sortear um elemento  $i$ , ( $i \in \{1, \dots, n\}$ ).
- 5     **Passo Construção da LRC**
- 6          $LRC \leftarrow$  determinar todos os  $b_{ij}$  que devem ser removidos da solução.
- 7         Avaliar cada uma das topologias possíveis de serem geradas.
- 8         Ordenar a LRC em ordem crescente utilizando o valor do  $FT_{Net}$  encontrado na avaliação de cada topologia.
- 9     **Passo Sorteio do elemento**
- 10         Sortear um elemento da LRC pertencente ao intervalo  $[1, Tamanho\ da\ LRC \cdot \alpha]$ ;
- 11         Remover o elemento sorteado na solução;
- 11 **Passo Retornar**
- 12     Retornar a solução encontrada.

Por fim, é necessário definir uma estratégia para classificar os elementos na LRC. Esse critério foi definido pela função adaptativa que mede a influência que cada enlace exerce no  $FT_{Net}$ . Ao simular a remoção de cada elemento da LRC que parte de um determinado nó  $e$ , avaliando a topologia resultante, para cada caso (linha 7), pode-se afirmar através de uma simples comparação, quais arestas são “melhores” ou “piores” para aquele cenário de tráfego e ordená-las na LRC (linha 8). Por “melhores” entende-se as arestas que, ao serem removidas, causam uma alteração mínima no valor do  $FT_{Net}$ , enquanto as “piores” são aquelas que resultam em uma maior alteração no valor do tráfego retransmitido.

## 4.2 Refinamento da solução

Após a obtenção da solução inicial, o processo de refinamento desta solução é iniciado. Esta etapa é essencial para obter uma solução com boa qualidade, pois apesar do GRASP investir esforços para iniciar a busca com uma solução de boa qualidade, normalmente há um potencial grande de melhora. Devido a intenção de explorar de modo abrangente o espaço de busca, escolhemos um valor um pouco mais alto para o  $\alpha$ , o que implica também em soluções mais

distantes de um mínimo local. O objetivo desta etapa é convergir para um mínimo local através de buscas sucessivas em uma estrutura de vizinhança definida. Por vizinhança entende-se todas as soluções que podem ser geradas a partir de uma solução conhecida seguindo algum critério.

Na literatura, um dos métodos mais utilizado para geração de vizinhos é o 2-opt (CROES, 1958). O refinamento ou busca local é apresentado no Algoritmo 4 a seguir. A estratégia utilizada no 2-opt para avaliar duas topologias (linha 10) é descrito na seção seguinte.

**Algoritmo 4:** Pseudo-código da Busca Local.

**Dados:** topologia virtual obtida como solução inicial para o problema

- 1 **Enquanto** *condição de parada não for satisfeita* **faça**
- 2     **Passo Sorteio dos elementos de origem**
- 3     |     sortear dois elementos  $s_1$  e  $s_2$ .
- 4     **Passo Sorteio dos elementos de destino**
- 5     |     Sortear elementos vizinhos  $d_1$  e  $d_2$  de  $s_1$  e  $s_2$ , respectivamente.
- 6     **Passo Permuta**
- 7     |     Remover os enlaces  $(s_1, d_1)$  e  $(s_2, d_2)$ .
- 8     |     Adicionar os enlaces  $(s_1, d_2)$  e  $(s_2, d_1)$ .
- 9     **Passo Atualização da solução**
- 10    |    **Se** a solução gerada for melhor que a corrente **então**
- 11    |    Solução corrente  $\leftarrow$  a solução gerada.
- 12    |    **Senão**
- 13    |    Descartar a solução gerada.

Neste trabalho a estratégia do 2-opt foi formulada de modo que as soluções geradas através da permuta sempre sejam válidas em relação às restrições de grau lógico. Isto é garantido pois ao final da permuta, os graus dos nós permanecem inalterados. A figura 4.1 ilustra esse fato.

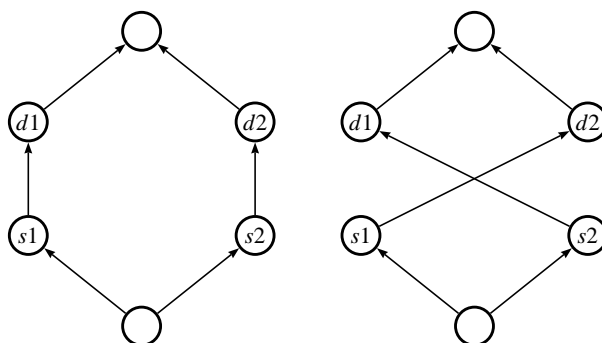


Figura 4.1: Permuta dos enlaces  $(s_1, d_1)$  e  $(s_2, d_2)$  por  $(s_1, d_2)$  e  $(s_2, d_1)$ .

O processo de permuta é iniciado com o sorteio de dois de origem,  $s_1$  e  $s_2$ . No caso de haver mais de um enlace com origem em  $s_1$ , um deles é sorteado. O mesmo processo ocorre

para  $s_2$ . O destino dos enlaces sorteados com origem em  $s_1$  e  $s_2$  são, respectivamente,  $d_1$  e  $d_2$ . Ao substituir os enlaces  $(s_1, d_1)$  e  $(s_2, d_2)$  por  $(s_1, d_2)$  e  $(s_2, d_1)$  a permuta ocorre de acordo com a linha 8 do Algoritmo 4.

Pode-se observar que existe o mesmo número de enlaces com origem em  $s_1$  e  $s_2$  com destino em  $d_1$  e  $d_2$ . É válido ressaltar que embora esse procedimento não altere o valor do grau lógico para nenhum elemento da rede, não há garantias quanto à conectividade da nova solução obtida. Contudo, para grau lógico maior, a desconexão da topologia em duas componentes conexas torna-se improvável devido ao aumento da conectividade da rede. A figura 4.2 ilustra um exemplo que a permuta gerou uma topologia desconexa. Esse problema relatado não oferece perigo para o algoritmo, pois na etapa de avaliação da solução (seção seguinte), é explicado como esta topologia desconexa é descartada.

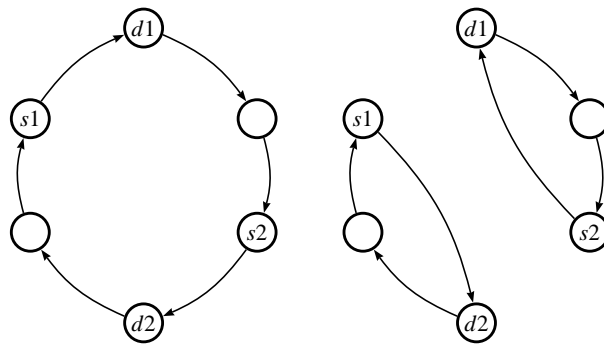


Figura 4.2: Topologia desconexa causada pela permuta dos enlaces.

### 4.3 Avaliação da solução

A etapa de avaliação da qualidade da solução é crucial para o refinamento e, neste trabalho, também para a geração da solução inicial pelo Método 2, ilustrado no Algoritmo 3. Em diversos trabalhos que abordam o projeto da topologia virtual, podemos citar (LIMA et al., 2007), é utilizado um resolvidor linear para esta etapa destinado a resolver o roteamento de tráfego e, devido o alto custo computacional para encontrar este modelo, o número de comparações a serem realizadas é limitado. Nesta seção é apresentada uma proposta de função de avaliação para o modelo de minimização do tráfego retransmitido (ALMEIDA et al., 2006) e uma simples modificação para adaptá-la ao modelo modificado proposto neste trabalho.

A contribuição de uma demanda de tráfego para o  $FT_{Net}$  é exatamente o seu valor,  $\lambda^{sd}$ , multiplicada pelo número de retransmissões que sofreu até chegar em seu destino, considerando

que a demanda não será subdividida por múltiplos caminhos. Na otimalidade, o roteamento de tráfego é realizado de modo que o número de retransmissões seja mínimo para todas as demandas de tráfego. Nas figuras 4.3 e 4.4 pode-se ver duas soluções para o roteamento de tráfego de uma mesma topologia, sendo uma ótima e outra não.

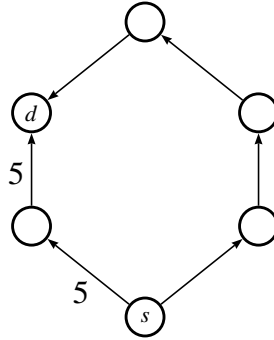


Figura 4.3: Solução ótima para o roteamento de tráfego na otimalidade ( $FT_{Net} = 5$ ).

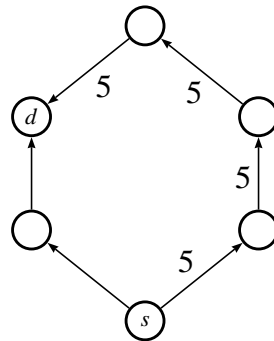


Figura 4.4: Solução viável para o roteamento de tráfego ( $FT_{Net} = 15$ ).

Portanto, para determinar o tráfego retransmitido em uma dada topologia é necessário descobrir o menor caminho entre todos os pares de nós da rede e multiplicar o valor da demanda pela quantidade mínima de saltos necessários para chegar ao destino. É importante ressaltar que devido a topologia virtual ser representada com um dígrafo tanto a conexidade quanto a distância não são necessariamente iguais em ambos os sentidos. Na figura 4.3 o nó  $s$  alcança todos os outros nós da rede, contudo, ao partir do nó  $d$  é impossível alcançar qualquer outro elemento. Portanto, é necessário determinar a distância mínima de todos os pares de nós.

Se essa tarefa fosse realizada utilizando o algoritmo do menor caminho Dijkstra (AHUJA et al., 1990) para todos os  $n$  nós da rede, a avaliação teria uma complexidade de  $O(n^2 \cdot \log(n))$ , contudo, devido as particularidades do problema as arestas possuem custos iguais. Para determinar a distância mínima casos, propomos utilizar o algoritmo exato de busca em largura que possui complexidade linear, calculando o menor caminho entre todos os pares de nós da rede com um custo na ordem de  $O(n^2)$ .

Como pode ser visto no Algoritmo 5 as distâncias são iniciadas com  $\infty$  e atualizadas com a distância mínima quando o nó é encontrado. Caso a topologia seja desconexa, o valor perma-

**Algoritmo 5:** Função de avaliação que realiza o cálculo do  $FT_{Net}$ .

**Dados:** Topologia virtual

- 1  $FT_{Net} = 0;$
- 2 **para** cada nó da rede ( $1 \dots n$ ) **faça**
- 3     **Passo Inicializa a solução**
- 4     | distancia[i]  $\leftarrow +\infty, \forall i = 1..n.$
- 5     **Passo Determinar as distâncias**
- 6     | distancia[i]  $\leftarrow$  busca largura(i);
- 7     **para** cada nó da rede ( $1..n$ ) **faça**
- 8     |      $FT_{Net} \leftarrow FT_{Net} + distancia[i];$
- 9     |     **Se**  $FT_{Net} = +\infty$  **então**
- 10    |     retornar  $+\infty.$  //topologia desconexa
- 11 **Passo Retornar**
- 12     Retornar  $FT_{Net}$

necerá  $\infty$  e, neste caso, a solução terá o maior valor possível sendo descartada na comparação realizada na etapa de refinamento. Na etapa de construção da solução inicial do Método 2, essa mesma estratégia foi realizada, isto é, caso a topologia se torne desconexa será descartada e o procedimento recomeça.

Uma simples adaptação pode ser realizada no procedimento de avaliação para adequá-la ao modelo modificado. Esta adaptação tem o propósito de contemplar o segundo termo da função objetivo do modelo (equação 3.2.1) já que o procedimento descrito no Algoritmo 5 calcula o  $FT_{Net}$ . Na seção 3.3 foi mostrado que o valor de  $\beta \cdot FT_{Max}$  está limitado entre o  $\beta \cdot FT_{Net}/n$  e  $\beta \cdot FT_{Net}$ , que serão chamados, respectivamente, de LB (*Lower bound*) e UB (*Upper bound*). Caso o valor retornado pela avaliação de uma nova solução somada ao UB for menor que o resultado da avaliação da solução que teve a última melhoria somada ao LB, pode-se afirmar que a nova solução é melhor em termos do tráfego retransmitido e  $FT_{Max}$ . Vale ressaltar que tanto UB quanto LB se referem ao máximo e ao mínimo de  $\beta \cdot FT_{Max}$  para uma determinada topologia e matriz de tráfego. Esta modificação é realizada com base nestes dois valores e não no cálculo do  $FT_{Max}$ . À medida que  $\beta$  aumenta, a diferença entre entre LB e UB também aumenta e, conseqüentemente, torna-se mais difícil distinguir qual é o melhor resultado. Portanto, esta estimativa só é eficaz para  $\beta$  pequeno.

## 5 *Resultados computacionais*

Os testes computacionais apresentados neste capítulo foram realizados para comparar tanto os resultados do modelo original e modificado, descritos nos capítulos 2 e 3, quanto as duas abordagens, em especial a geração da solução inicial apresentadas no capítulo 4, para o projeto da topologia virtual. Além disto, também é verificado o desempenho da função de avaliação.

Os modelos de otimização inteira mista foram escritos na linguagem ZIMPL (KOCH, 2004) e resolvidos utilizando o resolvidor inteiro SCIP (*Solving Constraint Integer Programs*) (ACHTERBERG, 2009) em conjunto com o resolvidor linear SoPlex (WUNDERLING, 1996), enquanto a meta-heurística GRASP foi implementada na linguagem de programação JAVA.

Os testes foram realizados em um computador de processador Intel i5-2500 *Sandy Bridge* com 3.3GHz de *clock* e 8GBytes de memória, executando o sistema operacional Linux (kernel 3.2). A análise do tempo de processamento apresentada nos gráficos deste capítulo foi realizada de duas formas: para a meta-heurística GRASP, o valor foi obtido através da diferença de duas chamadas do método *System.nanoTime*, uma inserida antes e outra depois do trecho de código que queremos analisar; para os modelos de otimização, utilizamos o tempo de busca do SCIP para alcançar a solução ótima. Seguindo a linha do trabalho realizado em (RAMASWAMI; SIVARAJAN, 1996) e de outros trabalhos que se utilizam de heurísticas para resolver o projeto da topologia virtual, foram realizadas cinco execuções do GRASP para cada instância utilizando o critério do modelo proposto com  $\beta = 0.01$ , selecionando o melhor resultado tanto da qualidade da solução inicial quanto para o resultado final do GRASP.

A escolha do parâmetro  $\beta = 0.01$  foi baseado em testes preliminares utilizando o modelo exato e além disso, foi verificado nos resultados apresentados na seção 5.2 (diferença entre UB e LB) que qualquer valor acima de zero produz um impacto positivo significativo no  $FT_{Max}$ . Como o valor de  $\beta$  é pequeno, o procedimento de avaliação descrito no capítulo anterior possui uma maior confiabilidade se uma solução é melhor que outra em termos da distribuição de tráfego e e retransmissão de tráfego.

Na seção seguinte é descrito como foi gerado o banco de instâncias utilizadas nos testes

computacionais para os modelos de otimização e para o GRASP. Nas duas seções, 5.2 e 5.3, são apresentados os resultados computacionais dos modelos de otimização e do GRASP, respectivamente.

## 5.1 Instâncias

Nesta seção apresentamos o procedimento que gerou o banco de dados das instâncias utilizadas nos testes computacionais das seções seguintes. Cada instância é composta de três parâmetros: o número de nós, o grau lógico e a matriz de tráfego, analisados separadamente a seguir.

### 5.1.1 Número de nós

O número de nós determina a quantidade de variáveis na matriz que armazena a topologia virtual (dimensão do problema), e tem relação direta com a dificuldade de encontrar a solução ótima. Por esse motivo, as soluções exatas dos modelos de minimização do tráfego retransmitido e o modelo modificado são encontradas somente para números de nós iguais a 8. Para instâncias de dimensão 20, 30 e 40 são destinados aos testes utilizando a meta-heurística GRASP.

### 5.1.2 Grau lógico

Para cada dimensão de instância gerada para os testes computacionais foram utilizados os valores de grau lógico 4, 6, 8 e 10, exceto para as instâncias que possuem dimensão igual a 8. Nestes casos foram utilizados somente os graus lógicos 2, 4 e 6.

### 5.1.3 Matriz de tráfego

A matriz de tráfego contém todo o tráfego que será roteado entre todos os pares de nós da rede. Para simular diferentes cenários de tráfego foram criadas, de modo aleatório, duas grandezas de demandas: uma pequena, isto é, com valores no intervalo  $[0,10]$  e outra grande variando no intervalo  $[10,100]$ . Com esses dois tipos de demandas, pequena e grande, são criados quatro tipos de matrizes de tráfego definidos pela percentagem de cada tipo de demanda.

Os tipos de matrizes de tráfegos são:

- 0% (uniforme): todos os elementos da matriz de tráfego são demandas pequenas.
- 25%: 25% dos elementos são demandas grandes e 75% são demandas pequenas.
- 50%: metade dos elementos são demandas grandes e a outra metade, demandas pequenas.
- 75%: Similar do tipo 25%, contudo, a divisão é invertida.

Para cada dimensão, grau lógico e tipo de matriz de tráfego, foram geradas 10 instâncias, ou seja, com as mesmas características. Totalizando 600 instâncias divididas da seguinte forma: 120 instâncias de dimensão igual a 8 e 160 instâncias para cada uma das demais dimensões.

A seguir uma instância de dimensão 10 e grau lógico 4.

#Número de nós

10

#Grau lógico

4

#Matriz de tráfego

0.00	71.11	5.87	5.05	0.27	63.53	8.74	34.45	8.23	0.72
7.21	0.00	75.96	4.84	2.28	2.55	5.56	4.27	53.19	8.37
4.25	9.66	0.00	83.10	3.07	0.81	0.87	16.69	8.18	4.23
4.97	80.63	17.57	0.00	1.65	5.70	7.82	50.67	1.74	5.89
2.40	6.51	5.18	2.52	0.00	2.37	51.96	9.70	0.50	4.30
9.69	5.33	3.90	37.62	4.77	0.00	0.93	4.10	1.28	5.97
6.01	93.09	74.13	17.73	57.49	7.10	0.00	4.14	2.50	63.52
5.82	4.04	26.95	7.44	4.10	5.18	6.67	0.00	7.78	5.78
4.49	9.16	94.85	0.12	77.99	2.61	4.91	4.52	0.00	8.45
15.41	5.49	8.04	39.14	4.84	6.10	7.76	5.96	1.59	0.00

#End

## 5.2 Resultados computacionais exatos para os modelos de original e modificado

Os testes realizados neste trabalho têm por objetivo comparar o modelo modificado proposto nessa dissertação com o modelo original (ALMEIDA et al., 2006) e, para isso, foram executados três tipos de testes usando o resolvidor inteiro SCIP. O primeiro teste pretende



comparar os valores do tráfego retransmitido em ambos os modelos, no modelo original esse valor foi obtido da própria função objetivo do problema, já que este consiste no  $FT_{Net}$ , enquanto que no modelo modificado esse valor foi obtido através da soma dos valores das variáveis  $FT_j$  em cada nó da rede.

O segundo teste compara o valor do  $FT_{Max}$  da solução apresentada por cada um dos modelos e, novamente, por falta dessa definição no modelo original, todos os valores de  $FT_j$  foram comparados e escolhido o maior deles. Já no modelo modificado este valor foi obtido da variável  $FT_{Max}$ .

O terceiro e último teste compara o tempo de execução para cada um dos modelos. Neste teste, devido ao maior número de restrições existentes no modelo modificado modificado, é esperado é esperado que o tempo para encontrar a solução ótima seja maior. Lembrando que os tempos computacionais avaliados nesta seção são os tempos retornados pelo SCIP.

Como os parâmetros  $\alpha$  e  $\beta$  influenciam diretamente no valor da função objetivo, isto é, ( $Minimize \alpha \cdot FT_{Net} + \beta \cdot FT_{Max}$ ) como a retransmissão e distribuição tráfego impacta na qualidade da solução, foram utilizados cinco valores para  $\alpha$ , lembrando que  $\beta$  é complementar e a soma dos dois parâmetros será sempre igual a 1. Os valores utilizados foram:  $\alpha = 0$ ,  $\alpha = 0.1$ ,  $\alpha = 0.5$ ,  $\alpha = 0.9$ , quando temos  $\alpha = 1$  o segundo termo da função objetivo se anula e ambos os modelos, o original e o modificado, se tornam idênticos, portanto, não há necessidade de testar esse valor, uma vez que o modelo original será executado.

As instâncias de dimensão 8 foram utilizadas nestes testes para o grau lógico de valores 2, 4 e 6. Para cada grau lógico foram geradas 40 instâncias, 10 para cada tipo de matriz de tráfego (0

É sabido que o modelo original irá obter valores menores, ou iguais, para o tráfego total retransmitido, afinal, a solução retornada é sempre ótima para esse critério, portanto, o modelo modificado será no máximo igual ao modelo original nesse quesito. À medida que é realizado uma distribuição igualitária do tráfego, o valor do tráfego retransmitido tende a aumentar. Por isto, o modelo original foi escolhido como referência para o tráfego retransmitido,  $FT_{Net}$ , e os resultados são expressos pela diferença percentual em relação a este modelo. Analogamente, sabe-se que nenhum modelo irá realizar uma distribuição mais igualitária do que o modelo modificado utilizando  $\alpha = 0$ , neste caso será realizado uma distribuição de tráfego completamente homogênea, portanto, os resultados relativos à distribuição de tráfego,  $FT_{Max}$ , utilizam estes resultados para servir de referencial de comparação. Por fim, os gráficos que mostram o tempo de execução dos modelos medidos pelo SCIP, mostram a diferença percentual do modelo original, que possui uma execução sabidamente mais rápida.

Para evitar vários gráficos repetitivos, pois o comportamento dos resultados foram análogos, selecionamos alguns gráficos representativos para compor os resultados.

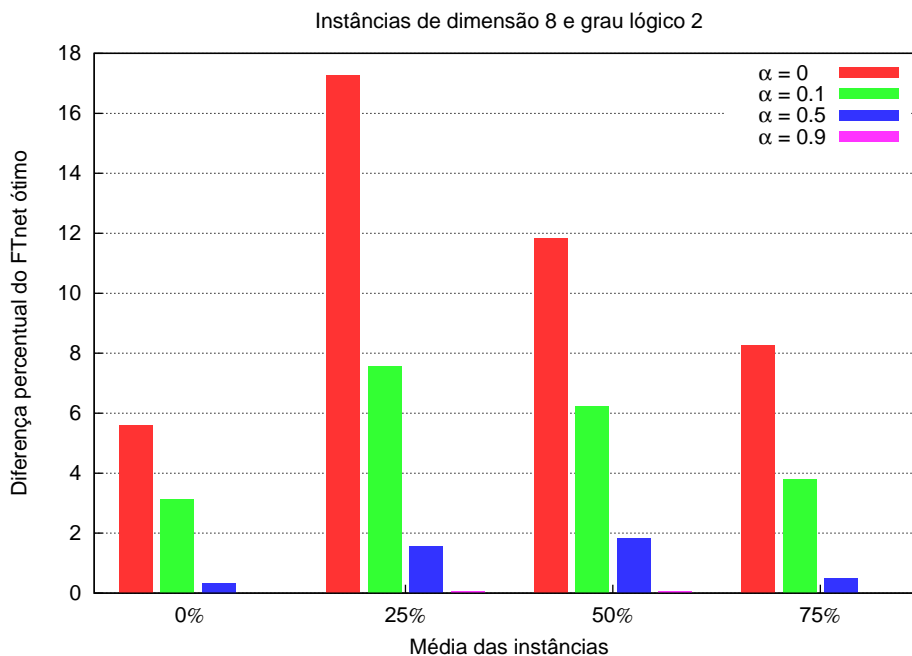


Figura 5.1: Tráfego retransmitido total da rede para cada tipo de matriz de tráfego: 0%, 25%, 50% e 75%.

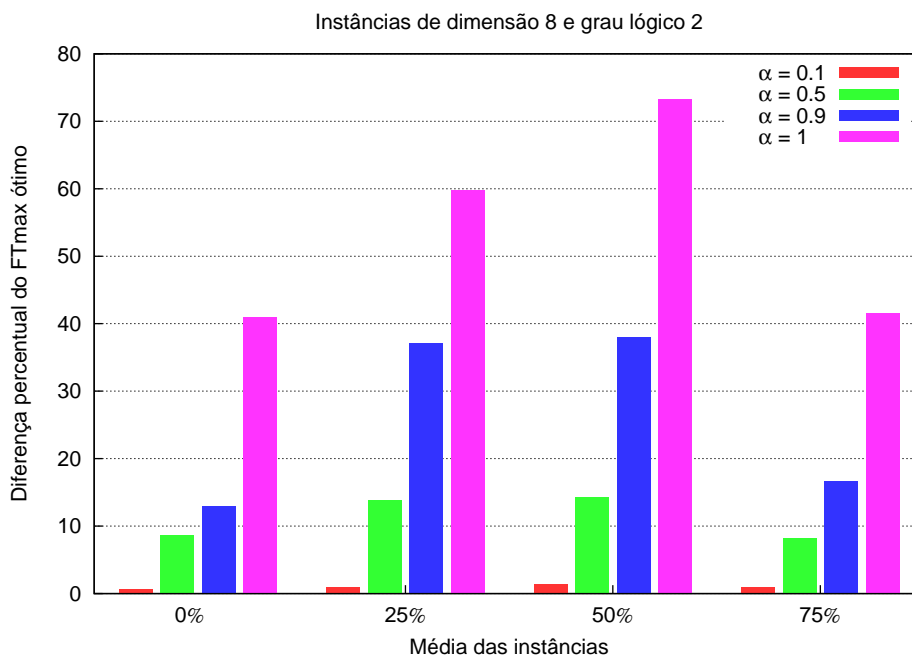


Figura 5.2: Tráfego retransmitido do nó mais sobrecarregado para cada tipo de matriz de tráfego: 0%, 25%, 50% e 75%.

Analisando a figura 5.1 vemos que o comportamento esperado ocorre, à medida que aumentamos a importância do tráfego retransmitido, em outras palavras, aumentamos o valor de

$\alpha$ , valor do tráfego retransmitido total,  $FT_{Net}$  diminui. Além disso, os resultados para  $\alpha = 0.9$  foram iguais ao modelo original, sendo verdadeiro esse comportamento para todos os tipos de matriz de tráfego. Na figura 5.2, analisamos o valor de  $FT_{Max}$ . Pode-se notar que o comportamento é análogo, isto é, diminui à medida que o valor de  $\beta$  aumenta. Para todos os tipos de matriz de tráfego esse comportamento foi verdadeiro.

Esse tipo de relação esperada se mostra verdadeira para todos os casos que o grau lógico é pequeno em relação ao número de nós da rede. Nesses casos há poucas rotas entre os pares de nós da rede, devido a baixa conectividade da rede, e para realizar uma distribuição mais igualitária quase sempre deve-se enviar a demanda por uma rota mais longa, isto é, com um maior número de saltos o que impacta em um maior processamento devido a retransmissão de pacotes. Porém, mesmo quando  $\alpha = 0.1$ , isto é, com uma importância pequena, o maior resultado foi somente 7,5% acima do  $FT_{Net}$  ótimo (figura 5.1).

À medida que a conectividade da rede aumenta e, conseqüentemente, mais rotas entre todos os pares de nós da rede, muitas destas com o mesmo número de saltos, torna-se possível realizar uma distribuição de processamento mais balanceada sem acarretar um aumento do processamento de pacotes. Esse comportamento foi verificado nas instâncias com grau lógico 4, ou superior, para uma rede de 8 nós. Nestes casos, independente do valor de  $\alpha$ , o valor para o  $FT_{Net}$  foi ótimo, ou muito próximo deste (menos de 0,04%). Esses resíduos numéricos são resultantes da baixa precisão das variáveis de ponto flutuante que acumulam algum erro pequeno ao longo da execução do algoritmo.

Analisando a carga do elemento mais sobrecarregado,  $FT_{Max}$ , para grau lógico igual a 4, temos um cenário parecido com o exemplo dado no capítulo 3, no qual para qualquer valor de  $\beta$  acima de zero irá produzir uma divisão igualitária sem o aumento de processamento de pacotes na rede, como pode ser visto na figura 5.3. Como o modelo original ( $\alpha = 1$ ) não se preocupa com esse aspecto, mesmo havendo caminhos alternativos com o mesmo número de saltos devido ao alto grau lógico, não é realizada uma distribuição do tráfego entre os demais nós, resultando numa carga de processamento do elemento mais sobrecarregado 125% a 200% acima do elemento mais sobrecarregado no modelo modificado.

Nenhum dos modelos analisados contém na função objetivo um termo relativo ao enlace mais congestionado da rede, contudo, devido ao modelo modificado realizar a distribuição da carga de processamento com o objetivo de não sobrecarregar nenhum elemento, foi verificado que há uma tendência de ocorrer também uma redução do congestionamento do enlace mais sobrecarregado em determinados tipos de matrizes de tráfego. Esse comportamento foi observado principalmente para as matrizes de tráfego do grupo 0% (uniforme) e 75%, obtendo uma

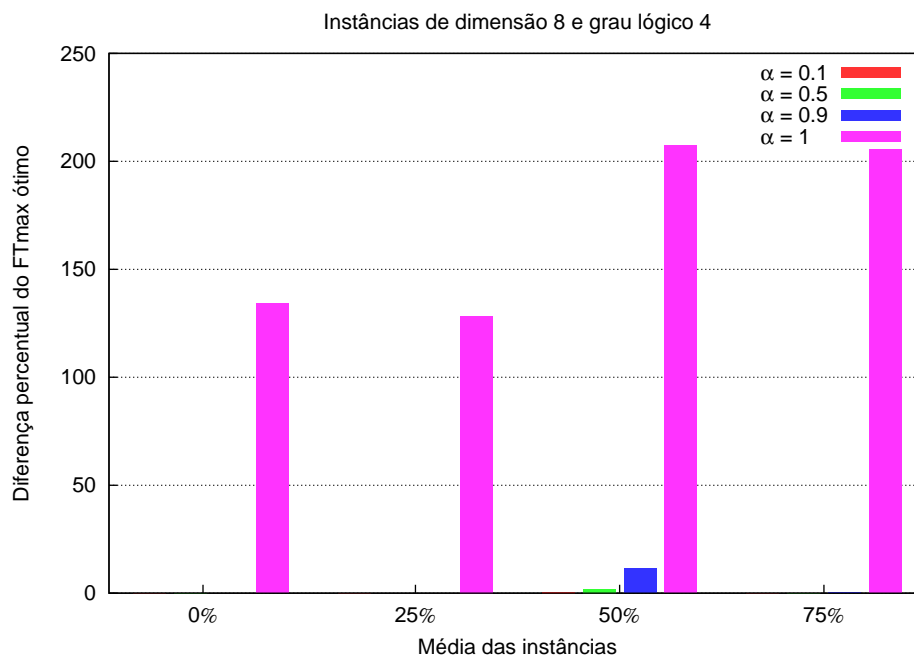


Figura 5.3: Tráfego retransmitido do nó mais sobrecarregado.

diferença média de cerca de 20% em relação ao modelo original ( $\alpha = 1$ ). Nesses dois grupos a maioria das demandas se comportam como demandas grandes e o modelo original sempre irá roteá-las pelas rotas mais curtas e, nesses casos, há uma tendência que alguns enlaces da rede sejam sobrecarregados. Já o modelo modificado evitara utilizar os mesmo enlaces pois evita aumentar muito o valor de processamento de um mesmo nó. No grupo 50%, esse comportamento também existe, porém é muito menos visível devido ao maior número de demandas pequenas e à diminuição de demandas grandes na rede e a diferença entre os modelos foi de aproximadamente 10%. O grupo 25% é um caso específico, as instâncias executadas possuem 8 nós e grau lógico 2 ou maior, ou seja, existem pelo menos dois enlaces partindo de cada nó para transportar as duas demandas grandes que se originará em média em cada nó da rede. A diferença entre todos os modelos do pior resultado para o melhor não chegou a 3% como pode ser visto na figura 5.4.

Em relação ao tempo de execução do algoritmo exato até atingir a solução ótima, como esperado, o modelo original executa mais rápido que o modelo modificado chegando a 225%, no pior caso, de diferença no tempo de execução em relação ao modelo original, como pode ser visto na figura 5.5. Analisando o impacto de  $\alpha$  no modelo modificado percebe-se que há uma influência no custo computacional, pois quanto mais próximo do modelo original, ou seja, maior  $\alpha$  mais rápida é a execução. Porém essa diferença só é visível para grau lógico baixo. Quando o grau lógico é maior, isto é, quando há rotas suficientes para que a distribuição seja realizada sem o aumento do  $FT_{Net}$ , a figura 5.6 mostra que não há qualquer relação entre o valor

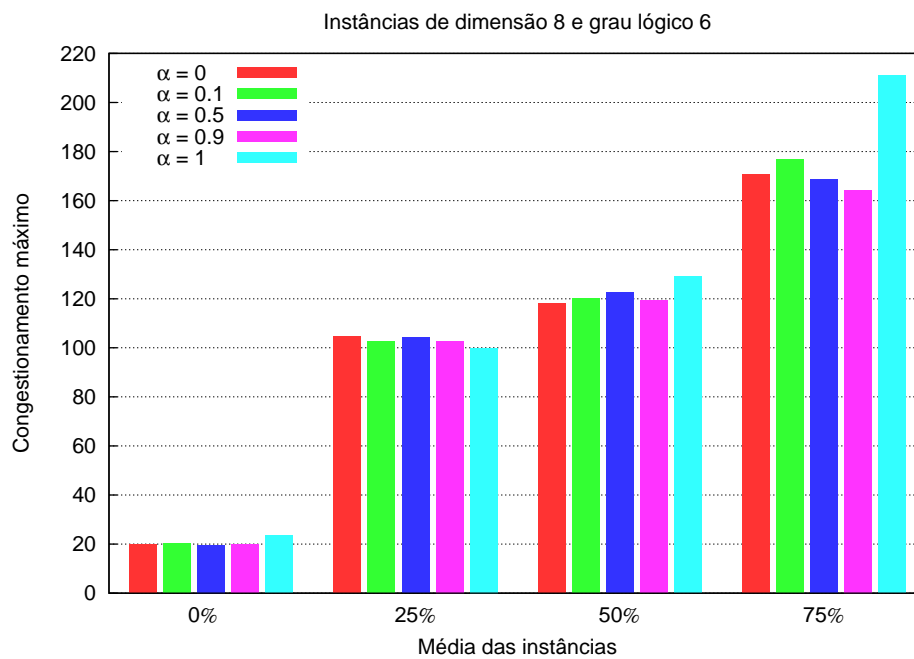


Figura 5.4: Enlace mais congestionado da rede.

de  $\alpha$  e o tempo de execução do modelo. O modelo modificado possui tempo de execução maior que o modelo original. A maior diferença do tempo de execução ficou em torno de 70%.

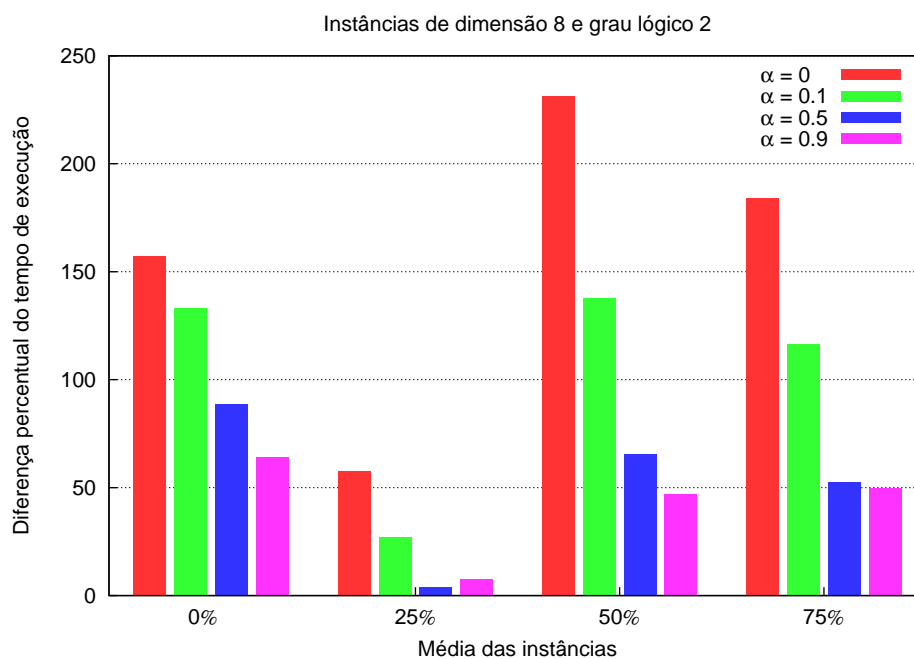


Figura 5.5: Tempo de execução até encontrar a solução ótima.

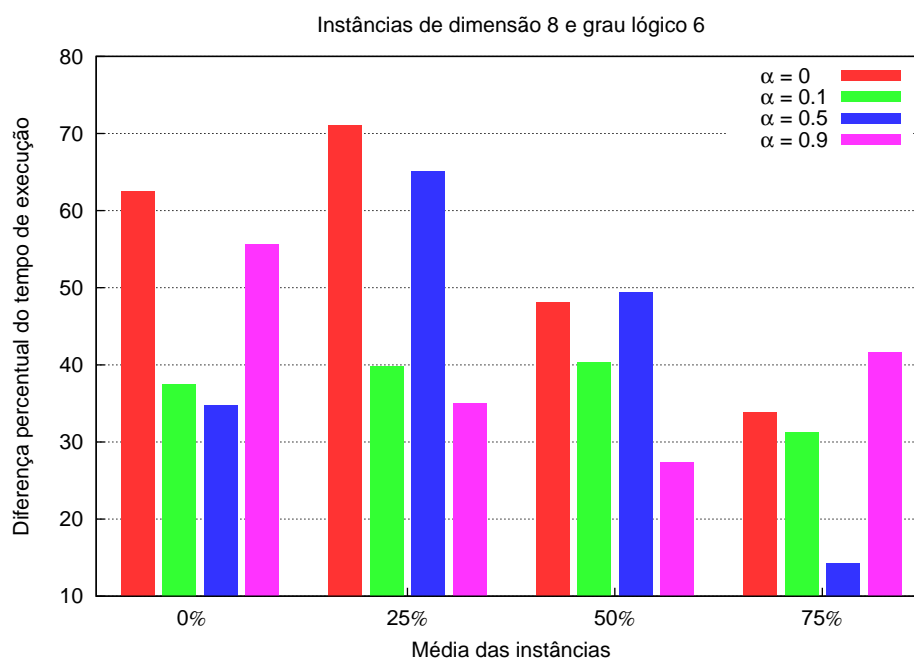


Figura 5.6: Tempo de execução até encontrar a solução ótima.

### 5.3 Resultados para o GRASP

As instâncias de dimensões maiores, de 20 até 40, foram resolvidas pela meta-heurística GRASP utilizando as duas abordagens, Método 1 e Método 2, apresentadas no capítulo capítulo 4, seção 4.1. Para cada dimensão, foram geradas 40 instâncias para cada grau lógico (4, 6, 8 e 10). Estas 40 instâncias são divididas do mesmo modo que as apresentadas na seção anterior, 10 para cada tipo de matriz de tráfego.

Testes preliminares em instâncias de dimensão 20 foram realizados para a sintonia dos parâmetros do algoritmo e, para esses casos,  $\alpha = 0.2$  se mostrou um valor adequado, isto é, pequeno o suficiente para gerar soluções viáveis de boa qualidade e, ao mesmo tempo, permitir uma exploração maior do espaço de soluções. Valores menores que 0.2 levam o algoritmo a algum mínimo local, enquanto valores maiores torna a etapa de refinamento mais dispendiosa.

Lembrando que nessa dissertação foram propostas duas versões para a etapa de construção da solução inicial para o GRASP, que chamamos de Método 1 e Método 2. O GRASP com Método 1 é denominado de GRASP1 e com o Método 2, GRASP2. Para comparação entre as versões, os critérios de parada do GRASP1 e GRASP2 foram os mesmos. Para o *loop* principal, o GRASP executou até que o número de iterações seja menor ou igual a  $n$  (dimensão da instância) e a etapa de refinamento é interrompida quando não fosse encontrada nenhuma solução melhor que a atual depois de  $n^2$  iterações.

Além do tempo de execução e qualidade da solução inicial, também foi medido a qualidade da solução obtida pelo GRASP ao final de todo o processo. O algoritmo de avaliação de uma topologia foi testado frente a variação da dimensão da instância e grau lógico para avaliar quantas topologias de cada tipo podem ser avaliadas em um segundo.

Devido ao método de avaliação impactar também na velocidade de obtenção da solução inicial do Método 2 utilizado no GRASP2, iniciamos a análise pelo método de avaliação. Os gráficos 5.7 e 5.8, mostram a quantidade de avaliações realizadas em um segundo. Devido a consistência do tempo de avaliação medido, os resultados de todas as 160 instâncias (para cada dimensão) é apresentada em um único gráfico contendo as médias dos resultados por grau lógico.

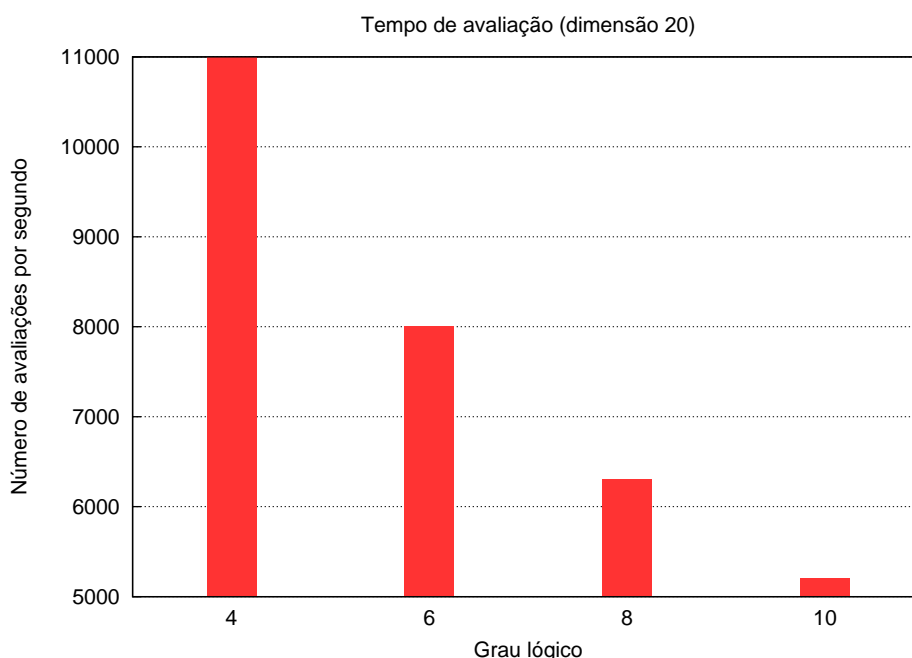


Figura 5.7: Avaliações realizadas em um segundo para instâncias de dimensão 20.

Podemos ver nos gráficos 5.7 e 5.8 que tanto o aumento do grau lógico quanto o aumento do número de nós torna a avaliação mais complexa. Esse fato era esperado, pois é um resultado conhecido na literatura de grafos que o desempenho da busca em largura é linear em relação ao número de vértices e arestas do grafo. Pode-se observar que ao dobrar o número de arestas de grau lógico 4 para 8, o número de avaliações realizadas por segundo cai praticamente à metade. Ao aumentar o número de nós a avaliação é mais custosa, o que não é uma contradição, já que o aumento de nós implica também no número de arestas. Além disso, como procedimento de avaliação realiza uma busca em largura iniciando em cada nó da rede para determinar o tráfego retransmitido total, o aumento do número de nós também implica em um número maior de chamadas ao procedimento de busca em largura, aumentando assim o custo da avaliação.

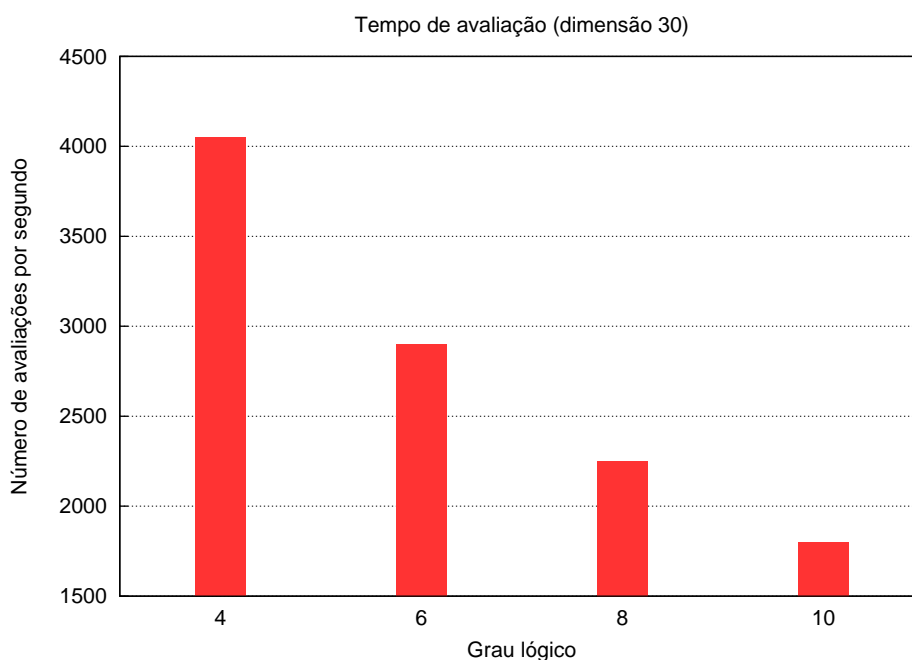


Figura 5.8: Avaliações realizadas em um segundo para instâncias de dimensão 30.

É importante ressaltar que o método de avaliação proposto se mostrou mais eficiente que o modelo de programação linear destinado à mesma finalidade e consequentemente, foi possível a elaboração do Método 2 de geração da solução inicial utilizado no GRASP2. Apesar da eficiência deste método de avaliação, ainda é custoso para as maiores instâncias testadas, avaliando cerca de 1000 topologias por segundo para instâncias de dimensão 40 e grau lógico 10. No SCIP, utilizando o modelo original para resolver somente o roteamento de tráfego, o tempo de computação foi da ordem de 0,2 segundos para a primeira avaliação, ou seja, cerca de 200 vezes mais lento. Para as avaliações seguintes é possível utilizar o *presolver* que melhora o desempenho das execuções seguintes, contudo, nos testes realizados essa melhoria no tempo computacional não teve um ganho tão alto a ponto de viabilizar o GRASP2.

Devido ao tempo computacional crescente do método de avaliação em relação à dimensão da topologia, é esperado que o método GRASP2, por utilizar o método de avaliação proposto para a construção da solução inicial, tenha um tempo computacional também crescente em relação à dimensão do problema, além do fato que o aumento da dimensão acarreta em um número muito maior de arestas que podem ser removidas do grafo a cada passo do algoritmo e portanto, o número de topologia que devem ser avaliadas também aumenta. Em suma, o aumento do dimensão do problema não só aumenta o custo de cada avaliação como também aumenta o número de avaliações necessárias para a construção da solução inicial pelo método utilizado no GRASP2. O gráfico 5.9 exibe a diferença de tempo de execução entre os dois métodos de construção da topologia inicial, os valores apresentados expressam as médias de



todas as instâncias de cada dimensão.

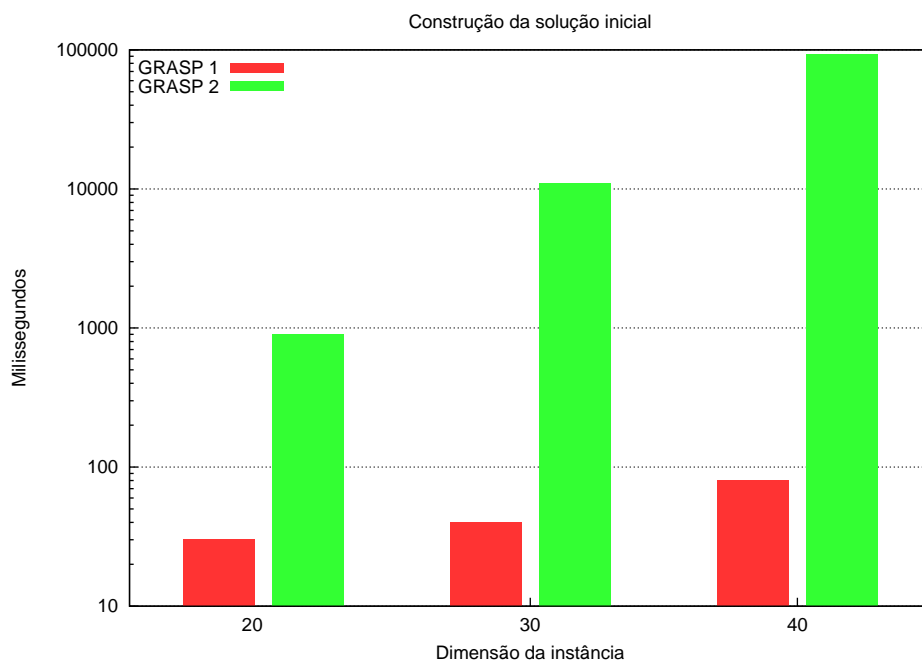


Figura 5.9: Média do tempo computacional de construção da solução inicial.

Apesar do GRASP2 utilizar um método para a geração da solução inicial que possui um custo computacional mais elevado que o Método 1 utilizado no GRASP1, pode-se ver claramente no gráfico da figura 5.10, a maior qualidade da solução inicial, principalmente nos casos onde as demandas de tráfego são semelhantes, os tipos 0% e 75%. Nestes casos, em várias instâncias, a média das soluções iniciais do Método 2 foi superior à média das soluções finais do GRASP1. O gráfico da figura 5.10 auxilia a observação desse comportamento pois apresenta quanto o valor, em média, do tráfego retransmitido para as soluções iniciais e finais de ambas abordagens distam, em porcentagem, do LB.

Essa diferença é evidente para grau lógico pequeno, como mostrado na figura 5.10 que são instâncias de dimensão  $n=10$  e grau lógico 6 (observe as colunas verde e azul). O gráfico ilustrado na figura 5.11 apresenta uma instância que caracteriza grau lógico mais elevando, com  $n=20$  e grau lógico 10, isto é, quando a conectividade é máxima para as instâncias geradas nessa dissertação. Nesses casos, o critério para a alocação dos enlaces se torna-se menos importante e a qualidade das soluções iniciais é compatível entre os Método 1 e Método 2 (observe as colunas vermelha e verde).

Com exceção dos casos de dimensão pequena e alta conectividade, é evidente a qualidade superior dos resultados obtido pelo método GRASP2, que obteve resultados iniciais mais próximos do LB tornando a segunda etapa do algoritmo, a melhoria, mais rápida e justificando o

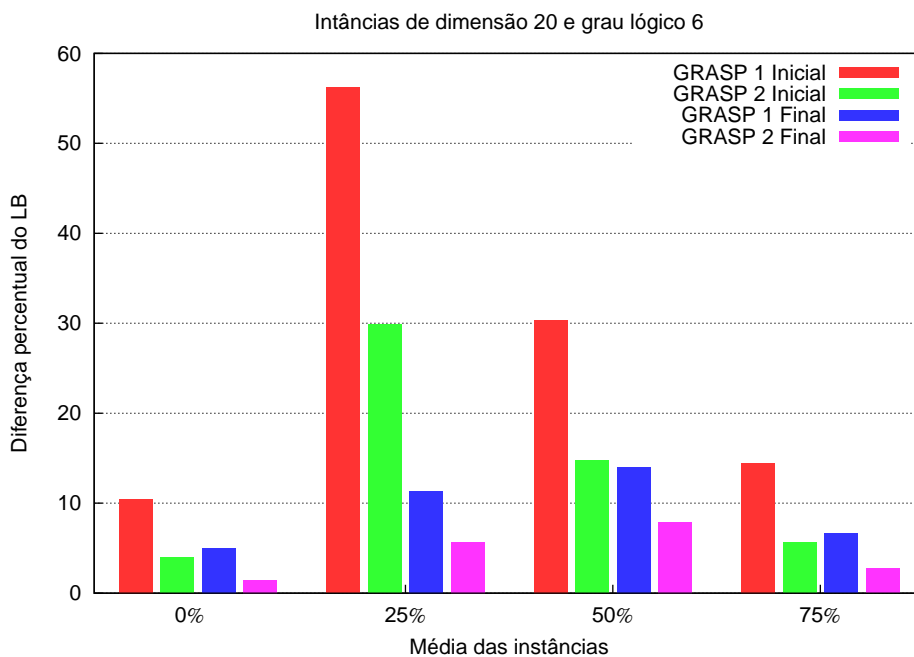


Figura 5.10: Qualidade da solução inicial e final utilizando as duas abordagens do GRASP.

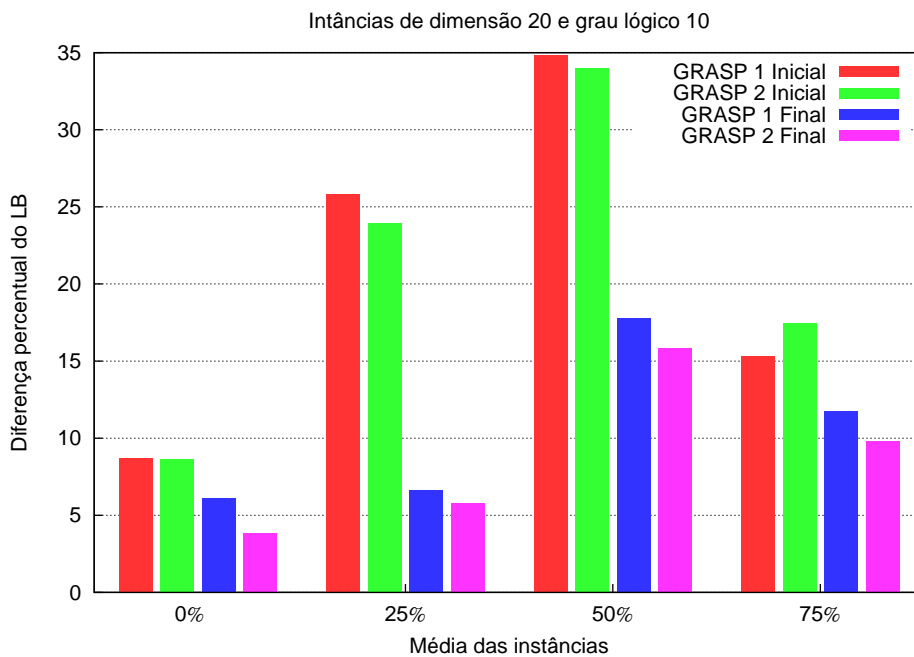


Figura 5.11: Qualidade da solução inicial e final utilizando as duas abordagens do GRASP.

tempo consumido na construção de uma solução inicial de melhor qualidade. Mesmo após a etapa de melhoria os resultados obtidos pelo GRASP2 se aproximaram, em média, cerca de 8% a mais do LB que o GRASP1.

Em relação ao tempo de execução da etapa de refinamento da solução, foi aferido utilizando o GRASP2 um tempo computacional menor para qualquer dimensão de instância, grau lógico

e tipo de matriz de tráfego, executando esta etapa de 2 a 4 vezes, dependendo da dimensão do problema, mais rápido que o GRASP1. Para ilustrar, o caso mais demorado foi quando  $n=40$  e, o tempo médio neste caso para o GRASP1 no refinamento da solução foi cerca de 12,15 minutos, enquanto o GRASP2 levou aproximadamente 3,1 minutos para concluir a mesma tarefa.

Ao considerar o tempo total da meta-heurística, isto é, construção da solução inicial e refinamento da solução o GRASP2 se mostrou mais demorado, especialmente em instâncias de grande dimensão, devido o grau de complexidade computacional da construção da solução inicial utilizado no GRASP2 que cresce rapidamente com o tamanho da instância. Vale lembrar que o procedimento de construção da solução inicial em uma instância de dimensão  $n=40$  ocorre 40 vezes ao longo da execução da meta-heurística, cada execução leva cerca de 100 segundos (figura 5.9), totalizando 4000 segundos (1 hora e 6 minutos). No GRASP1 essa mesma etapa executada 40 vezes ocorre em menos de 1 segundo. Basicamente o tempo computacional do GRASP1 é inteiramente o custo do refinamento da solução. Contudo, para instâncias pequenas o custo da solução inicial para o GRASP2 diminui drasticamente mas não o suficiente para que o tempo total do GRASP2 seja menor que o tempo total do GRASP1 para essas instâncias. Para instâncias de dimensão  $n=20$  o tempo computacional de ambas as abordagens foram similares, e o GRASP2 obteve qualidade superior nos resultados.

## 6 Conclusão

Nesta dissertação apresentamos uma formulação para o projeto da topologia virtual que contempla uma distribuição de tráfego de modo a não sobrecarregar nenhum elemento da rede. De modo complementar, foi provado os limites inferiores e superiores do novo termo adicionado à função objetivo.

Sobre a formulação apresentada neste trabalho, a distribuição de tráfego é realizada de modo homogênea, obtendo valores praticamente idênticos ao modelo original para o  $FT_{Net}$ . Para essa análise, instâncias de dimensões menores,  $n=8$  e graus lógicos 2, 4 e 6, foram executadas no resolvidor linear SCIP.

Para problemas de dimensões maiores,  $n=20, 30$  e  $40$  com graus lógicos 4, 6, 8 e 10, os testes foram executados com duas versões da meta-heurística GRASP (GRASP1 e GRASP2) utilizando dois métodos de construção da solução inicial. O primeiro método, denominado Método 1, gera uma solução inicial de forma rápida, alocando enlaces entre os elementos que possuem maiores valores para demandas existentes na matriz de tráfego, já o segundo método, denominado Método 2, possui um custo computacional maior que o primeiro, pois avalia o impacto de todas as arestas que partem de um determinado nó e após a avaliação, escolhe dentre os melhores elementos, repetindo o processo para todos os nós até que as restrições de grau lógico sejam satisfeitas. Devido ao grande número de avaliações realizadas na construção da solução inicial do Método 2, a função de avaliação proposta nesta dissertação, capaz de avaliar milhares de topologias por segundo, possibilitou que este método fosse executado em tempo hábil.

A função de avaliação influencia também a quantidade de comparações que podem ser realizadas pelo 2-opt na etapa de refinamento e, por possuir um desempenho superior aos resolvidores lineares que implementam o algoritmo Simplex, como o SCIP, possibilitou uma maior exploração do espaço de soluções.

O Método 2 se mostrou muito eficiente em termos da qualidade de solução na construção da solução inicial, apesar de possuir custo computacional maior que o Método 1. A solução

inicial obtida foi tão boa que este método pode funcionar como um método construtivo para os casos onde haja uma uniformidade nas demandas de tráfego, ignorando a etapa de refinamento da solução quando não há tempo hábil. Para o Método 1, a construção da solução inicial é realizada de forma rápida porém, possui uma qualidade inferior, conseqüentemente, necessita de um maior esforço na etapa de refinamento.

Mesmo utilizando a função de avaliação proposta neste trabalho, a etapa de refinamento ainda se mostrou custosa para ambos os métodos estudados (GRASP1 e GRASP2). O método GRASP2 se mostrou superior tanto em qualidade quanto em tempo computacional para instâncias de dimensão  $n=20$  devido a diminuição do tempo na etapa de refinamento, justificando o esforço na etapa de construção da solução inicial. Contudo, para instâncias maiores, embora a qualidade dos resultados se mantenham superior em relação ao GRASP1, o tempo investido para construir a solução inicial não se justifica totalmente pela rapidez alcançada na etapa de refinamento. Talvez seja necessário uma simplificação no método de construção da solução inicial a fim de tornar a construção mais ágil mesmo para instâncias de grandes dimensões.

## 7 *Trabalhos futuros*

A estrutura de vizinhança 2-opt utilizada nesse trabalho foi baseada em escolhas aleatórias para saber quais arestas seriam permutadas para gerar os vizinhos. Esta escolha pode ser melhorada se utilizando do conhecimento do problema. A função de avaliação proposta nesta dissertação pode ser utilizada para determinar quais elementos são mais promissores, ou quais enlaces não devem ser alterados pois gerariam soluções piores ou desconectadas.

A adaptação que pode ser realizada na função de avaliação, utilizada no GRASP, para adequá-la ao modelo proposto não calcula o  $FT_{Max}$  da solução de fato, ao invés disto, se baseia no conhecimento dos limites inferiores e superiores para determinar se uma solução é melhor que outra. Elaborar uma função de avaliação eficiente que contemple tanto o  $FT_{Net}$  quanto o  $FT_{Max}$  é sem dúvida um grande desafio a ser explorado em trabalhos futuros.

A etapa de construção da solução inicial do GRASP2 se mostrou custosa para as instâncias com dimensões maiores. Esse rápido aumento no tempo de execução é causado pela verificação de todas as arestas do grafo para determinar qual deve ser removida, esse processo é repetido até que a restrição de grau lógico seja satisfeita. E como o número de arestas aumenta quadraticamente em relação a dimensão da instância o tempo de execução aumenta rapidamente. Um possível modo de tornar essa etapa mais rápida é realizar o sorteio de um nó  $s$  e determinar qual é a melhor aresta que pode ser removida com origem em  $s$ , note que desse modo uma aresta é removida com  $n - 1$  avaliações ao invés de  $n^2 - n$ . Esse procedimento garante uma geração mais rápida, mesmo para instâncias maiores, porém ainda não foi avaliado o impacto na qualidade da solução em relação ao adotado no GRASP2.

## *Bibliografia*

- ACHTERBERG, T. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, v. 1, n. 1, p. 1–41, July 2009. [Http://mpc.zib.de/index.php/MPC/article/view/4](http://mpc.zib.de/index.php/MPC/article/view/4).
- AGRAWAL, G. Nonlinear fiber optics. *Nonlinear Science at the Dawn of the 21st Century*, Springer, p. 195–211, 2000.
- AHUJA, R. Network flows. *Technische Hochschule Darmstadt*, 1988.
- AHUJA, R. et al. Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*, ACM, v. 37, n. 2, p. 213–223, 1990.
- ALMEIDA, R. et al. Design of virtual topologies for large optical networks through an efficient milp formulation. *Optical Switching and Networking*, Elsevier, v. 3, n. 1, p. 2–10, 2006.
- ASSIS, K.; GIOZZA, W.; WALDMAN, H. Wdm optical networks: A complete design. *Journal of Communication and Information Systems*, v. 20, n. 3, p. 81–95, 2005.
- BANERJEE, D.; MUKHERJEE, B. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press, v. 8, n. 5, p. 598–607, 2000.
- BASTOS, C. Otimização de redes ópticas transparentes por colônia de formigas, proposta de dissertação (mestrado em engenharia da computação). *Coordenação de Pós-Graduação do Departamento de Sistemas Computacionais da Universidade de Pernambuco*, 2006.
- BIENSTOCK, D.; GUENLUEK, O. Computational experience with a difficult mixedinteger multicommodity flow problem. *Mathematical Programming*, Springer, v. 68, n. 1, p. 213–237, 1995.
- CROES, G. A method for solving traveling-salesman problems. *Operations Research*, JSTOR, p. 791–812, 1958.
- DUTTA, R.; ROUSKAS, G. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, Citeseer, v. 1, n. 1, p. 73–89, 2000.
- FEO, T.; RESENDE, M. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, Elsevier, v. 8, n. 2, p. 67–71, 1989.
- FEO, T.; RESENDE, M. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.
- FISHER, M. The lagrangian relaxation method for solving integer programming problems. *Management science*, JSTOR, p. 1–18, 1981.

- FLOOD, M. The traveling-salesman problem. *Operations Research*, JSTOR, v. 4, n. 1, p. 61–75, 1956.
- GENCATA, A.; MUKHERJEE, B. Virtual-topology adaptation for wdm mesh networks under dynamic traffic. *IEEE/ACM Transactions on Networking*, v. 11, p. 236–247, 2002.
- HAUPT, R.; HAUPT, S.; WILEY, J. Practical genetic algorithms. Wiley Online Library, 2004.
- KOCH, T. *Rapid mathematical prototyping*. Tese (Doutorado) — Technische Universität Berlin, 2004.
- KRISHNASWAMY, R.; SIVARAJAN, K. Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers. *Networking, IEEE/ACM Transactions on*, IEEE, v. 9, n. 2, p. 186–198, 2001.
- LAARHOVEN, P.; AARTS, E. Simulated annealing: theory and applications. Springer, v. 37, 1987.
- LAGUNA, M.; MARTÍ, R. A grasp for coloring sparse graphs. *Computational optimization and applications*, Springer, v. 19, n. 2, p. 165–178, 2001.
- LEONARDI, E.; MELLIA, M.; MARSAN, M. Algorithms for the logical topology design in wdm all-optical networks. *Optical Networks Magazine*, v. 1, n. 1, p. 35–46, 2000.
- LIMA, F. d. O. et al. Um estudo empírico da eficiência de heurísticas na otimização do congestionamento em redes ópticas. *Congresso de Pesquisa Operacional*, Anais do XXXIX Congresso de Pesquisa Operacional, 2007.
- LIMA, F. de O. et al. Projeto completo de redes ópticas. *Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo*, 2010.
- MOTTA, L.; OCHI, L.; MARTINHON, C. Grasp metaheuristics for the generalized covering tour problem. Citeseer, p. 387–391, 2001.
- MUKHERJEE, B. et al. Some principles for designing a wide-area wdm optical network. *Networking, IEEE/ACM Transactions on*, IEEE, v. 4, n. 5, p. 684–696, 1996.
- RAMAMURTHY, B.; RAMAKRISHNAN, A. Virtual topology reconfiguration of wavelength-routed optical wdm networks. *in Proceedings of IEEE GLOBECOM*, p. 1269–1275, 2000.
- RAMASWAMI, R.; SIVARAJAN, K. Design of logical topologies for wavelength-routed optical networks. *Selected Areas in Communications, IEEE Journal on*, IEEE, v. 14, n. 5, p. 840–851, 1996.
- RAMASWAMI, R.; SIVARAJAN, K.; SASAKI, G. Optical networks: a practical perspective. Morgan Kaufmann, 2009.
- RESENDE, M.; THOMAS, T. A grasp for satisfiability. Citeseer, 1996.
- WUNDERLING, R. *Paralleler und objektorientierter Simplex-Algorithmus*. Tese (Doutorado) — Technische Universität Berlin, 1996. [Http://www.zib.de/Publications/abstracts/TR-96-09/](http://www.zib.de/Publications/abstracts/TR-96-09/).



ZANG, H.; JUE, J.; MUKHERJEE, B. A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks. *Optical Networks Magazine*, v. 1, n. 1, p. 47–60, 2000.