

Luís Miguel Carvalho Raposo

DETERMINAÇÃO DE LIGAÇÕES PONTO-MULTIPONTO COM RESTRIÇÕES

Dissertação para obtenção do grau de Mestre em Engenharia
Electrotécnica e de Computadores, Área de Especialização em Telecomunicações

Setembro de 2015



UNIVERSIDADE DE COIMBRA



FCTUC **FACULDADE DE CIÊNCIAS
E TECNOLOGIA**
UNIVERSIDADE DE COIMBRA

Determinação de ligações ponto multi-ponto com restrições

Luís Miguel Carvalho Raposo

Dissertação para obtenção do Grau de Mestre em
**Engenharia Electrotécnica e de Computadores, Área de
Especialização em Telecomunicações**

Orientador: Professora Doutora Teresa Martinez dos Santos Gomes

Júri

Presidente: Professora Doutora Lúcia Maria dos Reis Albuquerque Martins
Orientador: Professora Doutora Teresa Martinez dos Santos Gomes
Vogal: Professora Doutora Rita Cristina Girão Coelho da Silva

Setembro de 2015

Agradecimentos

O primeiro agradecimento vai para a Professora Doutora Teresa Martinez, pela sua orientação, partilha de conhecimento e disponibilidade demonstrada durante o desenvolvimento desta dissertação.

À Professora Doutora Lúcia Martins, pela sua ajuda na compreensão do problema de construção das árvores de Steiner.

Aos meus pais, por permitirem, através de esforço e sacrifício, a minha formação e pelo carinho e atenção incessante durante todas as fases, especialmente as menos boas, do meu percurso académico.

Aos restantes familiares, que nunca me deixam esquecer o que é realmente importante.

Por último, mas não menos importante, a todos os amigos no verdadeiro sentido da palavra.

Muito Obrigado.

Abstract

The increase of transmission rates and the demand for large quantities of bandwidth to serve Multicast applications has led optical networks to become the first option in supporting high-speed connections between multiple users in a network. A Multicast session is established by the creation of a Multicast tree where information is simultaneously routed to a group of end-point users by a point-to-multipoint optical channel. When constructing a Multicast tree, the nodes of a network have to be equipped with optical-splitters to replicate the signal, which may lead to a costly network construction. To overcome this high cost, only a few nodes of the network may have this capacity. In order to reduce the cost of Multicast routing with the constraints mentioned, a new routing heuristic is proposed. This new heuristic, which uses an existing heuristic in a repetitive procedure, is intended to calculate Multicast trees in mixed-graph networks.

Due to the vast amount of information carried by an optical fiber, the occurrence of a failure in a Multicast tree may have a significant impact on the deliverance of traffic to the destinations of a Multicast session. In order to ensure survivability under single link failures, the predominant form of failures in optical networks, two trees with no common arcs, i.e. arc-disjoint trees, may be calculated. The first tree, referred to as primary tree, is used to transmit the information and the second one, secondary or protection tree, is reserved to transmit the information in the event of a single arc failure. The approaches used to derive this pair of trees, exploiting Multicast routing heuristics, aim to calculate a primary tree and, afterwards, calculate the secondary tree in a network where the arcs of the primary tree were removed. In this dissertation a new approach for the calculation of a pair of arc-disjoint trees is proposed. In this protection scheme, the primary tree is gradually constructed and checked, after the addition of each one of the destinations of the Multicast session, if it can be protected. In the event of an inadmissible protection tree, one of the arcs of the primary tree is detected and considered to be the reason for which the protection tree can not be calculated, and thus excluded from the derivation of a new primary tree.

The algorithms were evaluated through simulations and the analysis of the results has shown that the new approaches have successfully achieved their goal, namely increasing the number of requests that are established with protection and the average cost minimization of the Multicast trees calculated with the new routing heuristic.

Keywords: Multicast, Protection, Arc-Disjoint Trees, Multicast Routing, Optical Networks.

Resumo

O aumento das taxas de transmissão de informação e a procura de grandes quantidades de largura de banda para aplicações Multicast levou as redes ópticas¹ a tornarem-se o meio de transmissão preferencial no fornecimento de ligações de alta-velocidade entre múltiplos utilizadores de uma rede. Numa sessão Multicast, em vez de serem estabelecidas múltiplas ligações ponto-a-ponto, estabelece-se uma sessão ponto-a-multiponto suportada por uma árvore Multicast tirando partido, sempre que possível, da capacidade dos nós dividirem o sinal óptico. Devido ao seu elevado custo, apenas um número limitado de nós da rede possui esta capacidade de divisão. Com o propósito de diminuir o custo do encaminhamento Multicast com estas restrições, é proposta uma nova heurística de encaminhamento para este tipo de redes usando de forma repetitiva um algoritmo existente. Este algoritmo foi criado para o cálculo de árvores Multicast em redes mistas.

A enorme quantidade de informação transportada por uma fibra óptica, e o impacto em caso de falha da árvore Multicast, levou à necessidade de implementar técnicas de protecção Multicast eficientes de forma a garantir a capacidade de sobrevivência da rede em caso de falhas isoladas de arcos ou linhas. A sobrevivência de uma sessão Multicast suportada por uma árvore pode ser conseguida através de uma árvore disjunta nos arcos à primeira, ou seja duas árvores que não partilham nenhum arco entre si. As árvores calculadas são designadas, respectivamente, de árvore primária e árvore secundária (ou de protecção). A abordagem usada pelas heurísticas de encaminhamento Multicast na literatura visam, em primeiro lugar, o cálculo de uma árvore primária seguindo-se a construção de uma árvore secundária com a restrição de não ser possível usar nenhum arco pertencente à árvore primária. É aqui desenvolvida uma nova abordagem ao cálculo de um par de árvores disjuntas nos arcos, onde é verificado se se pode proteger uma árvore primária, gradualmente construída, após a adição de cada um dos seus nós destino. Na impossibilidade de protecção, o novo método identifica um arco que, possivelmente, impede a existência deste par de árvores e excluí este arco no novo cálculo de uma árvore primária.

A eficiência dos dois algoritmos propostos é testada através de simulações e os resultados obtidos são apresentados. A análise destes resultados simulacionais prova a superioridade da nova heurística, no cálculo de árvores Multicast com menor custo médio, e do novo esquema de protecção, no aumento do número de sessões Multicast estabelecidas com protecção.

Palavras-Chave: Multicast, Protecção, Árvores de Arcos Disjuntos, Encaminhamento Multicast, Redes Ópticas.

¹Esta dissertação não segue as regras do Novo Acordo Ortográfico.

Conteúdo

Lista de Figuras	iii
Lista de Tabelas	vii
Lista de Algoritmos	ix
Lista de Acrónimos	xi
1 Introdução	1
1.1 Objectivos	2
1.2 Organização da Dissertação	2
2 Conceitos Preliminares	5
2.1 Teoria dos Grafos	5
2.1.1 Notação e Definições	5
2.1.2 Representações da Rede	9
2.2 Redes Ópticas WDM	10
2.2.1 Componentes das Redes WDM	10
2.3 <i>Wavelength-Routed WDM Networks</i>	11
2.4 Multicast em Redes Ópticas WDM	13
2.4.1 <i>Light-Trees</i>	14
2.4.2 Arquitectura dos Nós	16
2.4.3 <i>Multicast Routing and Wavelength Assignment (MRWA)</i>	17
2.4.4 Encaminhamento Multicast em Redes Ópticas	18
2.4.5 Protecção de Sessões Multicast em Redes Ópticas	20
3 Heurísticas de Encaminhamento Multicast em Redes <i>Sparse-Splitting</i>	23
3.1 Algumas Considerações Importantes	23
3.2 Heurísticas Implementadas	27
3.2.1 <i>Multicast Using Splitters (MUS)</i>	28
3.2.2 <i>Sparse Splitting Multicast Routing Heuristic (SSMRH)</i>	29

Conteúdo

3.2.3	<i>Mixed-graph Sparse-splitting Heuristic (MSH)</i>	30
3.3	Heurística Proposta	35
3.3.1	<i>New Mixed-graph Sparse-splitting Heuristic (NMSH)</i>	35
4	Protecção de Sessões Multicast em Redes Sparse-Splitting	37
4.1	Protecção <i>Arc-Disjoint Trees (ADT)</i>	37
4.2	Protecção ADT Proposta	39
5	Análise de Desempenho	45
5.1	Descrição do Ambiente Experimental	45
5.2	Heurísticas de Encaminhamento Multicast	46
5.3	Protecção de Sessões Multicast	49
6	Conclusões e Trabalho Futuro	53
6.1	Contribuições da Dissertação	54
6.2	Trabalho Futuro	54
A	Arquitectura dos Nós de Redes Ópticas WDM	55
B	Resultados Obtidos	59
	Bibliografia	69

Lista de Figuras

2.1	Grafo (a) dirigido e (b) não dirigido.	6
2.2	Grafo misto.	6
2.3	Exemplo de um caminho.	7
2.4	Grafo (a) conexo e (b) desconexo.	7
2.5	Exemplo de um ciclo.	8
2.6	Exemplo de uma árvore dirigida.	8
2.7	Duas árvores abrangentes da rede dirigida da figura 2.1.	9
2.8	OADM de N canais. Imagem adaptada de [21].	11
2.9	Exemplo de uma rede <i>wavelength-routed</i> com quatro <i>light-paths</i> . Imagem adaptada de [21].	12
2.10	<i>Wavelength-continuity constraint</i> (a) sem conversor de comprimento de onda e (b) com conversão de comprimento de onda. Imagem adaptada de [21].	13
2.11	(a) Unicast vs (b) Multicast.	14
2.12	(a) <i>Light-forest</i> e (b) <i>light-trees</i> . Imagem adaptada de [31].	15
2.13	<i>Light-trees</i> construídas em configurações de rede diferentes (cada tipo de linha representa um comprimento de onda). (a) Todos os nós são MI DoC, (b) rede <i>sparse-splitting</i> , com nós MI DoC, (c) rede <i>sparse-splitting</i> , com nós MI DaC. Imagem adaptada de [6].	17
2.14	Exemplo de um problema MRWA em que os nós MI são DaC. Nos casos (a) e (b) existe um nó MC sem e com capacidade de conversão de comprimento de onda, respectivamente. Em (c) e (d) não existem nós MC, sendo o caso (c) sem conversão de comprimento de onda e o caso (d) com conversão de comprimento de onda. Imagem adaptada de [20].	18
2.15	Exemplo de duas árvores onde os seus arcos são disjuntos. (a) Árvore primária e (b) árvore secundária. Imagem adaptada de [11].	21
2.16	Esquemas de protecção Multicast. Imagem adaptada de [26].	22
3.1	Exemplo de uma árvore Multicast para a rede da figura 2.12.	26

Lista de Figuras

3.2	Exemplo da utilização do ANT. (a) Topologia da rede onde apenas estão assinalados os nós destino MC, (b) a respectiva rede MC e a árvore de Steiner calculada e (c) a <i>light-tree</i> resultante. Imagem adaptada de [6].	29
3.3	Exemplo da utilização do MSH. Em (a) é adicionado o destino d_1 , em (b) d_2 , em (c) d_3 e, finalmente, em (d) d_4 . Imagem adaptada de [11].	34
4.1	Exemplo de (a) uma rede <i>sparse-splitting</i> onde (b) representa a árvore primária obtida pela heurística MSH e (c) a rede resultante da remoção – arcos de custo incrementado de α – dos arcos pertencentes à árvore primária.	39
4.2	Vários passos da construção de duas árvores disjuntas nos arcos, para a rede da figura 4.1, com o novo método de protecção. Em (a) e (c) são representadas as árvores primárias parciais e em (b) e (d) as respectivas árvores secundárias.	40
4.3	(a) Rede obtida através do novo método onde é calculada a árvore primária. Árvores (b) primária e (c) secundária, obtidas com o novo método.	42
5.1	Custo médio da árvore por número de destinos para a rede “rede40-3” com 5 nós MC.	47
5.2	Custo médio da árvore por número de destinos para a rede “rnd40”.	49
5.3	(a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2” com 5 nós MC.	50
5.4	(a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2” com 5 nós MC.	51
5.5	(a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2”.	52
A.1	Arquitectura de um (a) SaD-OXC $N \times N$ baseado em (b) comutadores SaD. Imagem adaptada de [4].	56
A.2	Arquitectura de um MOSaD-OXC $N \times N$. Imagem adaptada de [4].	57
A.3	Exemplo de um módulo TCM 1x8. Imagem adaptada de [3].	57
B.1	Custo médio da árvore por número de destinos para a rede “rede40-0” com 5 nós MC.	60
B.2	Custo médio da árvore por número de destinos para a rede “rede40-1” com 5 nós MC.	61

B.3	Custo médio da árvore por número de destinos para a rede “rede40-2” com 5 nós MC.	62
B.4	Custo médio da árvore por número de destinos para a rede “rede50” com 5 nós MC.	63
B.5	(a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-0” com 5 nós MC.	64
B.6	(a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-1” com 5 nós MC.	65
B.7	(a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-3” com 5 nós MC.	66
B.8	(a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede50” com 5 nós MC.	67
B.9	(a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rnd40” com 5 nós MC.	68

Lista de Figuras

Lista de Tabelas

2.1	Algoritmo de Árvores Disjuntas nas Linhas (H-LDT)	21
2.2	Algoritmo de Árvores Disjuntas nos Arcos (H-ADT)	21
3.1	Tabela de predecessores da árvore Multicast representada na figura 3.1. . .	26
3.2	Tabela de distâncias mínimas da árvore Multicast representada na figura	
3.1.	26

Lista de Algoritmos

1	Algoritmo de preenchimento da tabela de predecessores e de cálculo das distâncias correspondentes (AFTP)	25
2	MUS	30
3	SSMRH	31
4	MSH	33
5	NMSH	36
6	H-NADT	44

Lista de Algoritmos

Lista de Acrónimos

- ADT** Árvores Disjuntas nos Arcos (*Arc-Disjoint Trees*)
- ANT** Transformação de Rede Auxiliar (*Auxiliary Network Transformation*)
- APTP** Algoritmo de Preenchimento da Tabela de Predecessores
- ASE** Emissão Espontânea Amplificada (*Amplified Spontaneous Emission*)
- DaC** *Drop-and-Continue*
- DoC** *Drop-or-Continue*
- LDT** Árvores Disjuntas nas Linhas (*Link-Disjoint Trees*)
- LR** Encaminhamento de *Light-path* (*Light-path Routing*)
- MC** *Multicast Capable*
- MI** *Multicast Incapable*
- MOSaD** *Multicast-Only and Splitter-and-Delivery*
- MRWA** Problema do Encaminhamento Multicast e Atribuição de Comprimentos de Onda (*Multicast Routing and Wavelength Assignment Problem*).
- MSH** *Mixed-graph Sparse-Splitting Heuristic*
- MUS** *Multicast Using Splitters*
- NADT** Novas Árvores Disjuntas nos Arcos (*New Arc-Disjoint Trees*)
- NMCF** *Nearest MC node First*
- NMSH** *New Mixed-graph Sparse-Splitting Heuristic*
- OADM** Multiplexadores Ópticos de Adição/Extracção (*Optical Add/Drop Multiplexer*)
- OTMCF** *On-Tree MC node First*

Lista de Acrónimos

- OXC** *Crossconnect Óptico (Optical Crossconnect)*
- RWA** Problema do Encaminhamento e Atribuição de Comprimentos de Onda (*Routing and Wavelength Assignment Problem*).
- SaD** *Splitter-and-Delivery*
- SPH** *Shortest Path Heuristic*
- SSMRH** *Sparse Splitting Multicast Routing Heuristic*
- TaC** *Tap-and-Continue*
- TCM** *Tap-and-Continue Modules*
- ToC** *Tap-or-Continue*
- WA** Atribuição de Comprimentos de Onda (*Wavelength Assignment*)
- WDM** Multiplexação por Divisão de Comprimento de Onda (*Wavelength Division Multiplexing*)

1

Introdução

Com o firme crescimento do débito dos serviços de comunicação e a necessidade de grandes quantidades de largura de banda para aplicações Multicast, como por exemplo a Televisão em Alta-Definição, videoconferência, video-on-demand [23], entre outras aplicações, a fibra óptica é o meio escolhido para fornecer ligações fiáveis e de alta velocidade. Para além da grande quantidade de largura de banda disponível, a fibra óptica possui simultaneamente uma taxa de bits errados relativamente baixa [8–12, 21]. Outra grande vantagem da utilização da fibra óptica é podermos multiplicar (por um factor elevado) a largura de banda efectiva de um sistema de comunicação deste tipo, através da utilização da técnica *Wavelength Division Multiplexing* (WDM) [21] onde cada fibra é constituída por diversos canais de comunicação com diferentes comprimentos de onda e ligações diferentes podem usar o mesmo comprimento de onda se não partilharem fibras em comum [13]. As redes ópticas WDM são assim as escolhidas para os serviços Multicast anteriormente referidos.

Numa rede óptica WDM, a informação é transportada em *light-paths* (canais ópticos ponto-a-ponto) [23] que ligam dois nós da rede (fonte e destino) e onde é atribuído um comprimento de onda a cada ligação. As primeiras implementações de ligações numa rede óptica foram conseguidas à custa de conjuntos de *light-paths* (um para cada destino) com origem no mesmo nó. Para as redes ópticas WDM suportarem a comunicação Multicast, e evitarem enviar uma cópia individual de uma mensagem para todos os destinos de uma sessão, foi introduzido em [24] o conceito de *light-tree* (canal óptico *point-to-multipoint*), que é uma extensão dos *light-paths*, e onde o nó fonte está equipado com um transmissor

1. Introdução

e os nós intermediários estão equipados com *optical splitters* [6].

A enorme quantidade de informação transportada pela fibra óptica e a possível perda em caso de falha numa *light-tree* levou à necessidade de implementar técnicas de protecção Multicast eficientes. Estas técnicas são desenvolvidas de forma a garantir capacidade de sobrevivência da rede em caso de falhas isoladas de arcos ou linhas, as mais comuns neste tipo de redes [8, 9, 11] e que ocorrem quando uma linha física de fibra da rede falha [21].

1.1 Objectivos

A realização desta dissertação teve especial foco em heurísticas de encaminhamento Multicast para redes ópticas WDM e na questão inerente da protecção de árvores em caso de falhas isoladas dos arcos em redes mistas, onde a protecção é conseguida através do cálculo de duas árvores disjuntas nos arcos: a árvore primária (usada em caso de funcionamento normal) e a árvore secundária (usada em caso de falha de um arco). Os objectivos inicialmente propostos nesta dissertação foram:

1. Implementação de duas heurísticas de determinação de árvores Multicast apresentadas nos artigos [7] e [11] com a respectiva árvore de protecção.
2. Melhorar o desempenho das heurísticas anteriores, nomeadamente através da proposta de um mecanismo de cálculo de uma árvore de protecção (disjunta nos arcos de uma árvore primária) embebido na determinação de uma árvore primária.
3. Realização de uma análise comparativa das várias heurísticas implementadas e propostas do ponto de vista do custo das soluções obtidas e da taxa de bloqueio apresentada.

1.2 Organização da Dissertação

Esta dissertação é composta por 6 capítulos. Após a Introdução, são apresentados no capítulo 2 os conceitos necessários à compreensão da comunicação Multicast em redes WDM, às heurísticas de encaminhamento e técnicas de protecção para sessões Multicast. No capítulo 3 são descritas de forma detalhada as heurísticas implementadas nesta dissertação e é proposta uma nova heurística de encaminhamento Multicast com o objectivo de reduzir o custo médio de árvores calculadas em relação às heurísticas implementadas. No capítulo 4 é descrita a associação da técnica de protecção de cálculo de um par de árvores disjuntas nos arcos com as heurísticas implementadas e são expostas as limitações deste método. Com vista à obtenção de melhores resultados para a protecção de sessões Multicast, ainda neste capítulo, é proposto um novo método de protecção que

calcula um par de árvores disjuntas nos arcos, mas que tem em conta a existência de uma árvore de protecção após a adição de cada um dos nós destino na árvore que vai sendo construída. Os resultados experimentais, para as heurísticas descritas no capítulo 3 e para as técnicas de protecção apresentadas no capítulo 4, são analisados no capítulo 5. Por fim, no capítulo 6 são apresentadas as principais conclusões extraídas desta dissertação, assim como sugestões para trabalho futuro.

1. Introdução

2

Conceitos Preliminares

Neste capítulo serão apresentados em detalhe os conceitos necessários à compreensão do trabalho desenvolvido ao longo da dissertação. Em primeiro lugar, na secção 2.1 serão introduzidas algumas definições importantes da teoria dos grafos, visto que são os grafos que representam as redes de telecomunicações usadas durante todo o desenvolvimento do trabalho. Na secção 2.2 são introduzidos conceitos sobre redes ópticas e em 2.3 conceitos sobre as redes ópticas WDM. Estas introduções iniciais servem de base para a abordagem do tema da comunicação Multicast em redes ópticas WDM, tópico apresentado na secção 2.4, onde têm aplicação os algoritmos de encaminhamento e a protecção de sessões Multicast, os temas centrais desta dissertação.

2.1 Teoria dos Grafos

2.1.1 Notação e Definições

Um grafo $G(N, E)$ é um sistema formado por um conjunto finito de elementos $N = \{1, 2, \dots, n\}$, de tamanho $|N| = n$, designados por *vértices* ou *nós*, e um conjunto $E = \{(i, j) \mid i, j \in N \text{ e } i \neq j\}$, de tamanho $|E| = m$, de pares não ordenados de vértices chamados *linhas* ou *arestas* [22]. Nestas condições, um grafo é considerado *não dirigido*. Se E for um conjunto de pares ordenados de vértices, então a designação dos elementos deste conjunto passa a ser *arco* (será usada a notação A , em alternativa a E , para distinguir estes dois tipos de conjuntos), e o grafo passar a ser *dirigido*. Na figura 2.1 são ilustrados dois exemplos, cada um relativo às duas classificações de um grafo. Pode ainda existir um

2. Conceitos Preliminares

terceiro tipo de classificação de grafos, os *mistos*, onde existem linhas com orientação (arcos) e linhas sem orientação (arestas).

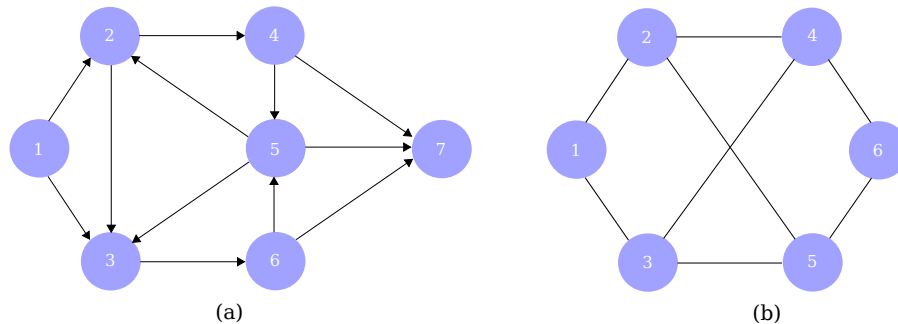


Figura 2.1: Grafo (a) dirigido e (b) não dirigido.

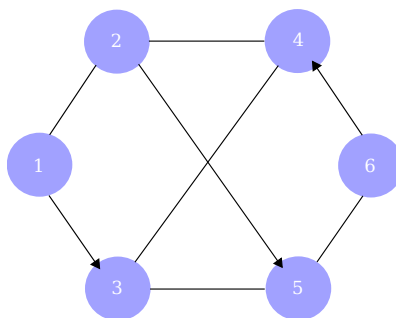


Figura 2.2: Grafo misto.

Nas redes dirigidas, os pares ordenados são então designados de arcos. Um arco $(i, j) \in A$, ou a_{ij} , tem dois nós extremos i e j , e designamos o nó i como *cauda* do arco a_{ij} e o nó j como a sua *cabeça*.

Um *caminho* P_k , num grafo com k nós, de um nó origem para um nó destino (diferente da origem), é definido como uma sequência de nós e arestas (a começar no nó origem e a acabar no nó destino) onde cada aresta é precedida e seguida pelos seus dois nós e cada nó é precedido e seguido por arestas a que pertence, com a restrição de não existirem nós nem arestas repetidos [22]. O *comprimento* de um caminho P_k é então o número de arestas que o constituem, ou seja $k - 1$. De modo semelhante, em redes dirigidas, um caminho P_k , num grafo com k nós, de um nó origem para um nó destino (diferente da origem), é definido como uma sequência de nós e arcos (a começar no nó origem e a acabar no nó destino) onde cada arco é precedido e seguido por uma cauda e uma cabeça e cada nó é precedido e seguido por um arco incidente e um arco emergente, com a restrição de não existirem nós nem arcos repetidos. A figura 2.3 mostra um exemplo de um caminho de comprimento quatro.

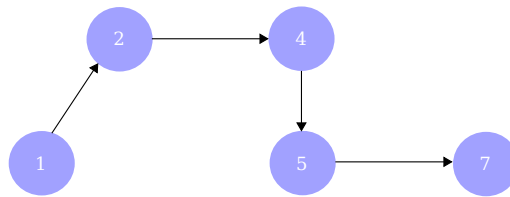


Figura 2.3: Exemplo de um caminho.

Com a atribuição de um número real, denominado por *peso*, às arestas de um grafo, passamos a designar este por *rede* ou grafo *valorado* [22]. Este peso, poderá representar, por exemplo, a capacidade do arco, o custo ou até a probabilidade de falha. Para o peso da aresta, ou arco, será usada a notação $c_{a_{ij}}$. O *custo* c_{ij} de um caminho entre dois vértices i e j é a soma do peso das arestas que fazem parte desse caminho (é considerada uma métrica aditiva para o custo do caminho), assumindo que a passagem nos vértices tem peso nulo. Podemos generalizar este conceito para o custo de uma árvore, que será a soma dos pesos das arestas que compõem a árvore.

O *grau* $d(i)$ de um vértice i é o número de arestas a que ele pertence. Num grafo dirigido, distingue-se o *semigrau de entrada*, que é o número de arcos que terminam (arcos incidentes) em i , do *semigrau de saída* que é o número de arcos que partem (arcos emergentes) de i , e a soma dos dois semigraus dá o grau. No exemplo da figura 2.1 (a) o grau do nó 3 é quatro, sendo o semigrau de entrada igual a três e o semigrau de saída igual a um.

Um grafo é *conexo* se existe um caminho entre cada par de vértices do grafo (caso contrário é *desconexo*) e *simplex* se não contiver *laços* - arcos em que o nó cauda é o mesmo que o nó cabeça - ou várias arestas com o mesmo par de vértices extremos. Um grafo dirigido pode ser *fortemente conexo* se existir um caminho entre cada par dos seus vértices ou *fracamente conexo* se substituirmos os arcos por arestas e o grafo contiver um caminho entre cada par dos seus vértices. Na figura 2.4 são ilustrados dois grafos. O grafo da figura 2.4 (a) que é conexo e o grafo da figura 2.4 (b) que é desconexo, sendo a sua componente definida pelo conjunto de nós $\{1, 2, 3, 4\}$ fortemente conexa.

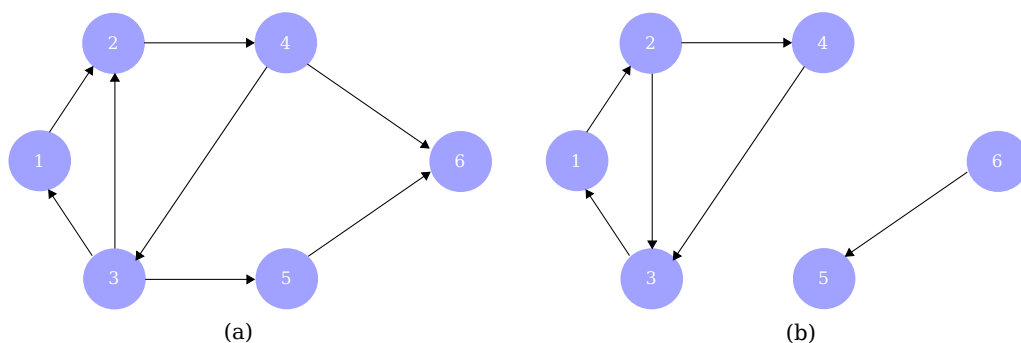


Figura 2.4: Grafo (a) conexo e (b) desconexo.

2. Conceitos Preliminares

Um *ciclo* C_n , de comprimento mínimo $n = 3$, é um caminho em que os vértices inicial e final são coincidentes, como é possível verificar no exemplo da figura 2.5.

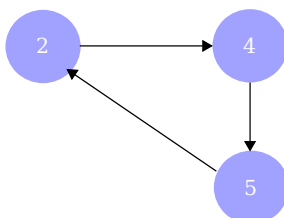


Figura 2.5: Exemplo de um ciclo.

Um grafo $G' = (N', A')$ é um *sub-grafo* de $G(N, A)$ se $N' \subseteq N$ e $A' \subseteq A$, com $A = \{(i, j) \mid i, j \in N' \text{ e } i \neq j\}$. Caso um sub-grafo G' seja acíclico e desconexo, designamos este sub-grafo de *floresta*. Uma *árvore* (exemplo da figura 2.6) é um sub-grafo conexo e acíclico de um grafo G , e que contém algumas propriedades importantes:

1. Uma árvore de n nós possui exactamente $n - 1$ arcos.
2. Uma árvore tem pelo menos dois nós com grau um.
3. Cada par de nós de uma árvore está ligado por um caminho único.

Um qualquer nó i da árvore com grau $d(i) > 2$ é designado por *nó de ramificação* (*branch node*). No exemplo da figura 2.6 os nós 3 e 5 são nós de ramificação.

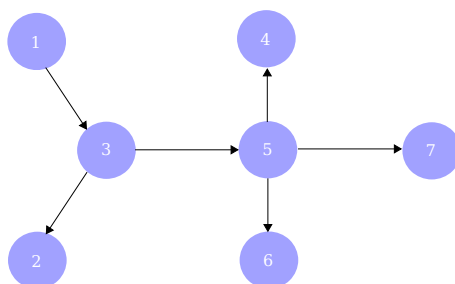


Figura 2.6: Exemplo de uma árvore dirigida.

Uma *árvore abrangente* ou *árvore geradora* é uma árvore de um grafo G que liga todos os vértices de G . De todas as árvores abrangentes possíveis, aquela que apresenta menor custo é a *árvore abrangente mínima* (podendo existir mais que uma árvore abrangente mínima). A figura 2.7 ilustra duas árvores abrangentes do grafo da figura 2.1. Uma *árvore de Steiner* é um sub-grafo de um grafo G que liga um determinado conjunto de vértices (com mais do que dois nós, menos que todos os nós do grafo, considerados como

terminais ou *destinos*). De todas as árvores de Steiner possíveis, aquela que apresenta o menor custo é a *árvore de Steiner mínima*.

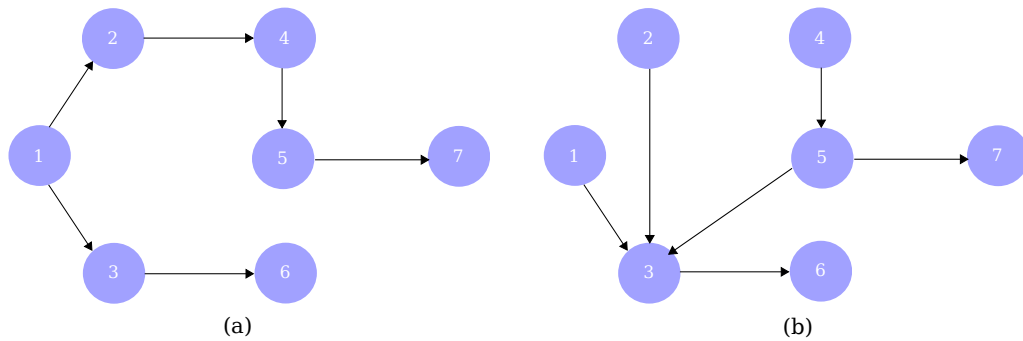


Figura 2.7: Duas árvores abrangentes da rede dirigida da figura 2.1.

2.1.2 Representações da Rede

A eficiência computacional de qualquer algoritmo de encaminhamento numa rede, para além de depender das suas características, depende ainda da forma como é computacionalmente representada a rede nas estruturas de dados usadas e dos esquemas de armazenamento criados para guardar e actualizar resultados calculados durante a execução do algoritmo [1]. Na representação de redes, são habitualmente necessários dois tipos de informação:

1. Topologia da rede, ou seja a estrutura dos nós e arcos da rede.
2. Informação sobre custo, capacidade e outros atributos associados com os nós e arcos da rede.

Uma das opções mais usadas para representar redes, e a escolhida para representar as redes usadas no decurso desta dissertação, é a *lista de adjacências* [1]. Nesta estrutura de representação de redes são armazenadas listas de adjacências de nós, de cada um dos nós da rede, como uma única lista ligada (coleção de células que contêm um ou mais campos). A lista de adjacências $A(i)$ de um nó i é o conjunto de nós adjacentes a esse mesmo nó, ou seja, $A(i) = \{j \in N : (i, j) \in A\}$. A lista de adjacências de um nó i será uma lista ligada contendo $|A(i)|$ células e cada uma destas células corresponderá a um arco $(i, j) \in A$, contendo os campos necessários à informação que queremos armazenar. Na lista de adjacências construída, os campos usados para cada célula são o nó j , o custo do arco $c_{a_{ij}}$ e uma cópia do custo do arco.

2.2 Redes Ópticas WDM

O crescimento da Internet, e o respectivo aumento do tráfego, levaram a que, hoje em dia, a largura de banda usada por cada utilizador seja um dos factores principais a ter em conta durante a construção de uma rede de telecomunicações, assim como ligações de alta-velocidade. As redes ópticas podem oferecer estas características com uma maior largura de banda em comparação com os cabos de cobre sendo menos susceptíveis a vários tipos de interferências electromagnéticas ou outros efeitos indesejados [23].

O aparecimento da tecnologia de multiplexação por divisão do comprimento de onda (*Wavelength Division Multiplexing* WDM) surgiu para explorar eficientemente o enorme potencial da capacidade de largura de banda da fibra óptica. Na multiplexação WDM, uma fibra tem a capacidade de transportar múltiplos canais de comunicação ópticos, onde cada um desses canais opera a um comprimento de onda diferente (que não se interferem entre eles desde que sejam mantidos suficientemente afastados [23]). A fibra que transporta o sinal pode ser vista como uma junção de várias “fibras virtuais”, cada uma transportando o seu próprio sinal.

A WDM tornou-se então a tecnologia escolhida para empregar nas redes ópticas e corresponder aos pedidos de largura de banda nas infraestruturas de telecomunicações.

2.2.1 Componentes das Redes WDM

Os componentes chave para o funcionamento de uma rede WDM são os *Optical Add/Drop Multiplexers* (OADM), os amplificadores ópticos e os *Optical Crossconnects* (OXC). Os OADMs são elementos de rede, cuja função passa por fornecer capacidade de adicionar e extrair tráfego na rede [30], e permitem que um número de canais de comprimento de onda sejam extraídos e adicionados reduzindo o número de conversões optoeletrónicas. Na sua constituição existe um desmultiplexador, seguido de um conjunto de comutadores 2x2 e de um multiplexador (figura 2.8). Se os comutadores estiverem no estado *bar* (comprimento de onda λ_i da figura 2.8), todos os comprimentos de onda passam através do OADM. Quando os comutadores são configurados para o estado *cross*, o sinal do comprimento de onda correspondente é extraído localmente, e esse comprimento de onda passa a estar disponível para ser atribuído a um novo sinal.

Os amplificadores ópticos são introduzidos na fibra e a sua principal função é compensar a atenuação que ocorre na propagação do sinal na fibra óptica.

Os OXCs comutam sinais ópticos de um porto de entrada para um porto de saída apropriado de uma fibra. Estes elementos estão localizados nos nós da rede e, como os OADM, podem extrair e adicionar tráfego na rede [30]. A grande diferença entre os OXCs e os OADMs é que os primeiros têm um grande número de portos, enquanto que

os últimos apenas têm dois portos [23]. Tantos os OXCs como os OADMs podem ter capacidade de conversão de comprimento de onda.

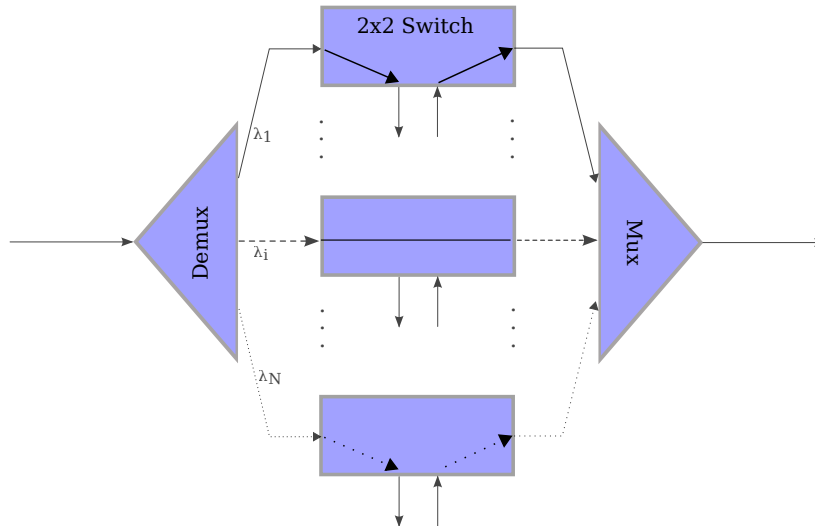


Figura 2.8: OADM de N canais. Imagem adaptada de [21].

2.3 Wavelength-Routed WDM Networks

Uma rede WDM que utiliza encaminhamento por comprimento de onda de forma a transferir tráfego é referida como uma *Wavelength-Routed WDM Network*. Dentro desta arquitectura, uma rede consiste numa colecção de nós que se associam a OXCs para estabelecer ligações de canais WDM entre pares de nós. Cada nó está equipado com um conjunto de transmissores, que envia mensagens para a rede, e receptores, que recebem mensagens da rede.

Numa rede *Wavelength-Routed* a comunicação entre utilizadores é feita através de um *light-path*. Um *light-path* é um canal óptico ponto-a-ponto que conecta dois nós numa rede, com um comprimento de onda atribuído a cada ligação estabelecida, e que pode abranger várias linhas de fibra. Numa rede óptica com N nós, se cada nó estiver equipado com $N - 1$ transmissores e receptores e se existirem comprimentos de onda suficientes em todas as linhas da fibra, então todos os pares de nós podem ser ligados por um *light-path* [21]. Apesar disto, as redes podem apresentar algumas limitações devido ao custo dos transceptores (transmissores e receptores) e às restrições tecnológicas que limitam o número de canais WDM que podem ser suportados numa fibra, entre outros factores. Por conseguinte, apenas um número limitado de *light-paths* pode ser realmente estabelecido numa rede.

Se considerarmos uma rede onde não existe conversão de comprimento de onda, podemos afirmar que um *light-path* opera sempre no mesmo comprimento de onda em todas

2. Conceitos Preliminares

as linhas de fibra que atravessa e, nesse caso, que a restrição da continuidade do comprimento de onda (*wavelength-continuity constraint*) é satisfeita. No caso de dois *light-paths* partilharem uma linha de fibra, dois comprimentos de onda diferentes deverão ser atribuídos a cada um dos caminhos ópticos (*wavelength clash constraint* [27]). A figura 2.9 ilustra um exemplo de uma rede (sem conversores de comprimento de onda) onde vários *light-paths*, com diferentes comprimentos de onda, são estabelecidos entre alguns pares de nós.

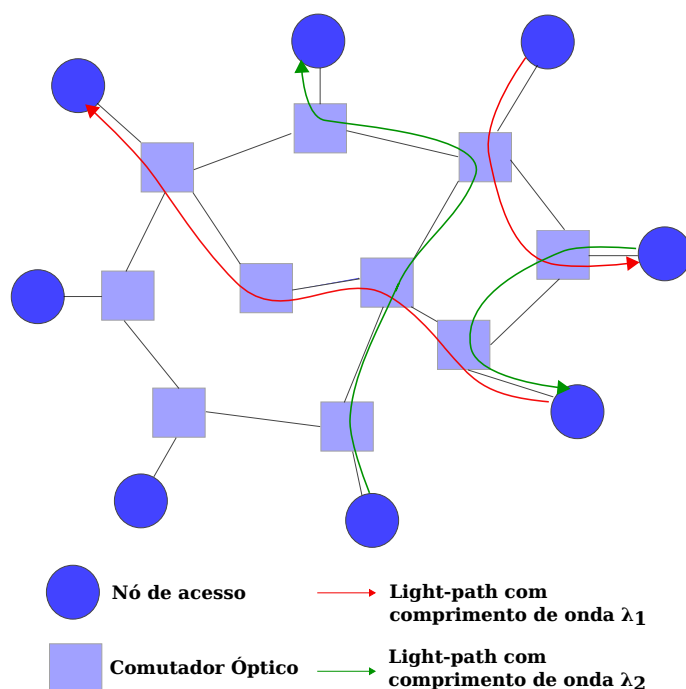


Figura 2.9: Exemplo de uma rede *wavelength-routed* com quatro *light-paths*. Imagem adaptada de [21].

Na figura 2.10 (a), são estabelecidos dois *light-paths* entre os nós A e B no comprimento de onda λ_1 e entre os nós B e C no comprimento de onda λ_2 . Se for pretendido estabelecer uma ligação entre os nós A e C, e não existirem mais comprimentos de onda disponíveis, então a restrição da continuidade do comprimento de onda impede a ligação destes dois nós. Uma forma de eliminar esta restrição do problema, e evitar perdas de largura de banda, passa por converter a mensagem recebida num comprimento de onda para um outro comprimento de onda num nó equipado para o efeito e enviar a mensagem para a próxima linha [33]. A solução será então um *light-path* onde coexistem diferentes comprimentos de onda desde a sua origem até ao destino. Na figura 2.10 (b), o nó B está equipado com um conversor de comprimento de onda cuja função é converter o comprimento de onda de um sinal que chega à entrada do nó por um comprimento de onda disponível no sistema.

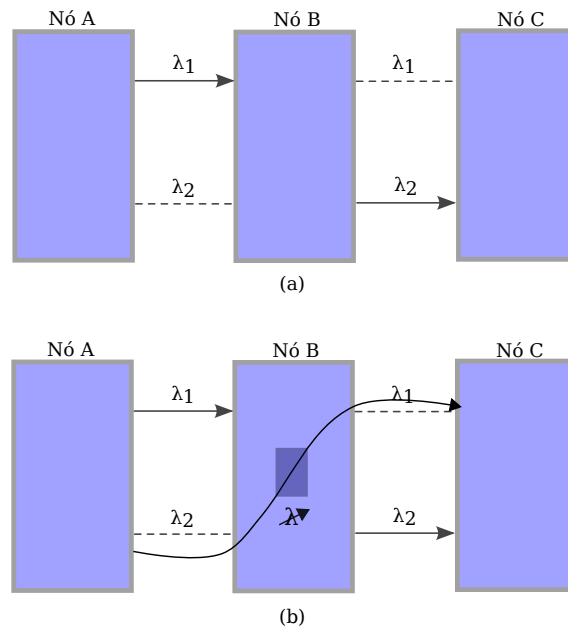


Figura 2.10: *Wavelength-continuity constraint* (a) sem conversor de comprimento de onda e (b) com conversão de comprimento de onda. Imagem adaptada de [21].

O problema de estabelecer *light-paths*, atribuir comprimentos de onda de forma a minimizar a quantidade de recursos de rede consumidos e assegurar que dois *light-paths* não compartilhem o mesmo comprimento de onda na mesma linha de fibra é designado de *Routing and Wavelength Assignment Problem (RWA)* e pode ser dividido em duas partes distintas, o *Light-path routing (LR)* cujo objectivo é encontrar caminhos para um conjunto de *light-paths* e o *Wavelength assignment (WA)*, onde se atribuem comprimentos de onda a esse conjunto de *light-paths* minimizando o número máximo de comprimentos de onda usados numa linha de fibra [23].

2.4 Multicast em Redes Ópticas WDM

As redes ópticas WDM são as mais adequadas para providenciar ligações para aplicações e serviços que estão disponíveis actualmente (televisão em alta-definição, videoconferência, aplicações do tipo streaming, entre outras). A maioria destas aplicações necessita de ligações ponto-a-multiponto. Daí advém a necessidade das redes poderem suportar os serviços fornecidos pela comunicação Multicast que providencia a capacidade de transmitir informação simultaneamente de uma fonte para múltiplos destinos. Na figura 2.11, é visível a eficiência do Multicast em relação ao Unicast. Neste exemplo, uma mensagem deve ser enviada de uma fonte para três destinos. Enquanto que no caso Unicast são necessários três comprimentos de onda diferentes para transmitir a mensagem, em Multicast apenas um é utilizado.

2. Conceitos Preliminares

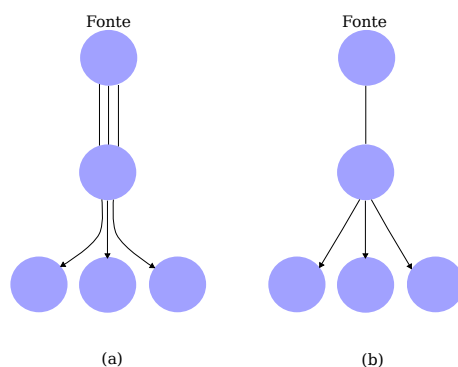


Figura 2.11: (a) Unicast vs (b) Multicast.

2.4.1 *Light-Trees*

O conceito de *light-tree* foi introduzido em [24] para garantir capacidade de suporte de sessões Multicast e diminuir o bloqueio nas rede ópticas, onde o bloqueio é definido como a incapacidade de obtenção de uma solução para uma sessão Multicast. Uma *light-tree* é uma extensão de um *light-path* com a introdução de capacidade Multicast ou seja, um canal óptico ponto-a-multiponto que pode abranger várias linhas de fibra e que utiliza um único comprimento de onda.

Uma implementação eficiente de sessões Multicast requer a existência de nós na rede que tenham a capacidade de dividir um sinal de entrada para múltiplos nós vizinhos, providenciando assim uma utilização eficiente das linhas de fibra. No caso da figura 2.11 (b), é facilmente verificável que pelo facto do nó intermédio ter essa capacidade de dividir o sinal, o uso de canais de comprimento de onda pode ser otimizado em relação a uma comunicação ponto-a-ponto. Assim, para que seja possível construir uma *light-tree* numa rede óptica, os nós de ramificação da *light-tree* (nós com grau superior a dois) devem ter, associados aos OXCs, *optical-splitters*, cuja função é replicar um sinal óptico de entrada em dois ou mais sinais de saída.

A implementação das *light-trees* deve ainda ter em conta alguns elementos que podem ser mais utilizados na rede em relação aos *light-paths*, como amplificadores ópticos e conversores de comprimento de onda. A maior utilização de amplificadores ópticos advém do facto de serem feitas n cópias do sinal óptico usando um ou mais *optical splitters*, criando cópias com $1/n$ da potência do sinal original. Para manter o sinal acima de um determinado limiar, mais amplificadores serão então necessários [21]. Já os conversores de comprimento de onda poderão ser mais utilizados nas *light-trees* devido à sua estrutura usar mais linhas de fibra que as usadas num *light-path*, sendo maior a probabilidade de ocorrer a utilização da mesma linha de fibra.

A introdução de *optical-splitters* em todos os nós da rede (rede *full-splitting*) tem uma desvantagem associada, uma vez que o custo da construção de comutadores com esta

capacidade é elevado [34] e inclui a utilização de amplificadores ópticos que introduzem ruído devido ao *Amplified Spontaneous Emission* (ASE) [23]. Para evitar estes problemas, apenas um determinado número de nós está equipado com *optical-splitters*. Este tipo de rede é designado por rede *sparse-splitting* [20]. Para além da restrição da continuidade do comprimento de onda, numa rede *sparse-splitting* (sem conversão de comprimentos de onda) o grau de um nó sem capacidade de replicar o sinal, numa *light-tree*, não pode ser maior que dois [36]. A esta restrição designamos de *sparse light-splitting constraint*. Assim, como consequência, várias *light-trees* em diferentes comprimentos de onda devem ser estabelecidas para uma sessão Multicast. As *light-forests* [34] consistem então em uma ou mais *light-trees* com raiz na fonte Multicast. Na figura 2.12 é distinguido o conceito entre uma árvore Multicast¹ composta por uma *light-forest* (2.12 (a)²) e uma árvore Multicast constituída por várias *light-trees* (2.12 (b)). Neste exemplo é visível que no caso da figura 2.12 (b), com a inclusão do conversor de comprimento de onda no nó equipado com *optical splitter*, uma única árvore pode cobrir todos os destinos usando apenas um transmissor no nó fonte e menos um canal de comprimento de onda que no caso da figura 2.12 (a). Assim, é facilmente verificável que uma das desvantagens em usar *light-forests* é o aumento do número de canais de comprimento de onda usados e maior número de transmissores no nó fonte da sessão Multicast.

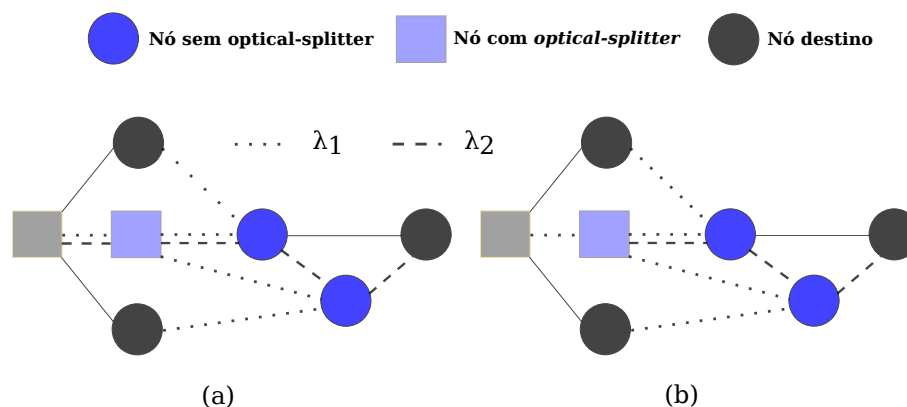


Figura 2.12: (a) *Light-forest* e (b) *light-trees*. Imagem adaptada de [31].

A eliminação da *sparse light-splitting constraint* durante a construção de uma árvore Multicast é obtida através da conversão de comprimento de onda introduzida nos nós da rede (*full wavelength conversion*). Porém, a colocação de conversores de comprimento de onda nos nós da rede também tem um impacto na sua construção, como no caso dos

¹O conjunto de *light-trees* obtidas que solucionam uma determinada sessão Multicast será designado daqui em diante como árvore Multicast.

²É de notar que o percurso físico de uma sessão Multicast não é uma árvore (no contexto da teoria dos grafos) [31], sendo necessário separar as *light-trees* por comprimentos de onda para obter de facto o conceito clássico de árvores.

2. Conceitos Preliminares

optical-splitters, devido ao elevado custo destes aparelhos. A solução para diminuir este custo passa por colocar este elemento em apenas alguns nós da rede (*sparse wavelength conversion*). Já foi demonstrado em [2] que as redes *sparse-splitting* podem alcançar os resultados das rede *full-splitting* a nível da probabilidade de bloqueio de uma sessão Multicast. Com a colocação de apenas uma fracção de conversores de comprimento de onda nas redes é possível alcançar os principais benefícios da colocação dos conversores em todos os nós da rede [28], se os *optical-splitters* e os conversores de comprimento de onda forem estrategicamente localizados. Nos artigos [5, 19] são estudadas formas de colocar um dado número de *optical splitters* em locais estratégicos de uma rede WDM para minimizar o número total de canais de comprimento de onda utilizados por uma ligação Multicast.

2.4.2 Arquitectura dos Nós

Numa rede *sparse-splitting* podem ser encontrados três tipos de nós:

1. *Multicast Capable (MC)* [24], nós equipados com *optical splitters* e, portanto, com capacidade de replicação do sinal.
2. *Multicast Incapable (MI)*, nós incapazes de replicar o sinal e que podem ser divididos em:
 - (a) *Drop-and-Continue (DaC)* ou *Tap-and-Continue (TaC)* [3], no qual é extraída uma pequena parte do sinal óptico no nó local e enviado o restante sinal para o nó sucessor deste.
 - (b) *Drop-or-Continue (DoC)* ou *Tap-or-Continue (ToC)*, se extrai uma pequena parte do sinal óptico no nó local ou se encaminha o sinal para o nó sucessor deste.

Na figura 2.13 são ilustradas as diferentes configurações de rede existentes e as respectivas *light-trees* para a sessão Multicast. Na árvore da figura 2.13 (a), são necessários quatro *light-paths* (logo, quatro comprimentos de onda diferentes) e onze canais de comprimento de onda. Com a inclusão de um nó MC (com conversão de comprimento de onda) na rede, apenas dois comprimentos de onda e oito canais de comprimento de onda são necessários, como é ilustrado na figura 2.13 (b). Se os nós MI passarem a ser DaC, como ilustrado na figura 2.13 (c), o número de comprimentos de onda mantêm-se em dois, mas o número de canais necessários decresce para sete. Analisando as diferentes configurações e respectivas *light-trees*, conclui-se que a inclusão de nós MC nas redes pode diminuir o número de comprimentos de onda necessários e logo o custo da rede.

A inclusão de nós MI que suportam DaC pode reduzir muito mais este custo. Porém, a maior parte das arquiteturas de comutadores de comprimento de onda ainda não suportam operações DaC [6]. Uma descrição resumida das arquiteturas dos nós MC e DaC pode ser vista no apêndice A.

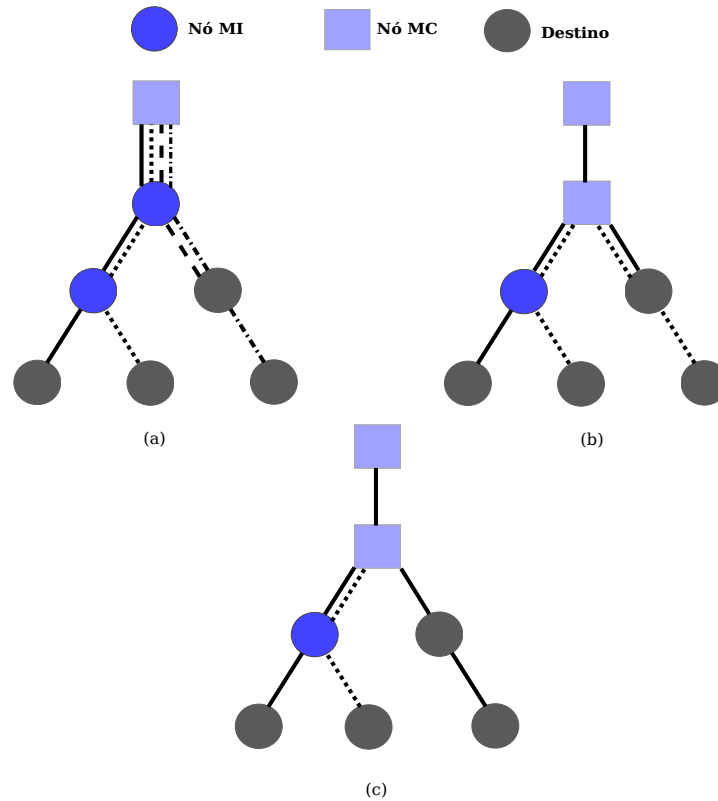


Figura 2.13: *Light-trees* construídas em configurações de rede diferentes (cada tipo de linha representa um comprimento de onda). (a) Todos os nós são MI DoC, (b) rede *sparse-splitting*, com nós MI DoC, (c) rede *sparse-splitting*, com nós MI DaC. Imagem adaptada de [6].

2.4.3 Multicast Routing and Wavelength Assignment (MRWA)

O problema de MRWA é uma extensão do RWA para o caso de tráfego Multicast, onde, como já foi visto, alguns nós têm a capacidade de fazer cópias de um sinal a ser enviado para múltiplos destinos. De forma semelhante ao problema RWA, os principais objectivos que se pretendem cumprir no MRWA são o encaminhamento Multicast e atribuição de comprimentos de onda às árvores estabelecidas, com o intuito de minimizar o número de comprimentos de onda requeridos ou maximizar o número de sessões estabelecidas (sem bloqueio).

Na figura 2.14 está ilustrado um problema de MRWA em diferentes configurações de rede. O nó inicial é o nó fonte e os restantes nós são nós destinos de uma sessão Multicast. É assumido que o nó fonte usa múltiplos transmissores e, logo, pode ter o número de nós

2. Conceitos Preliminares

filhos necessários para a criação de uma árvore, podendo transmitir para estes através de múltiplos comprimentos de onda. Os comprimentos de onda são representados por λ e por λ^* , sendo que este último representa qualquer λ disponível (e na mesma figura diferentes valores são permitidos em cada ocorrência de λ^*).

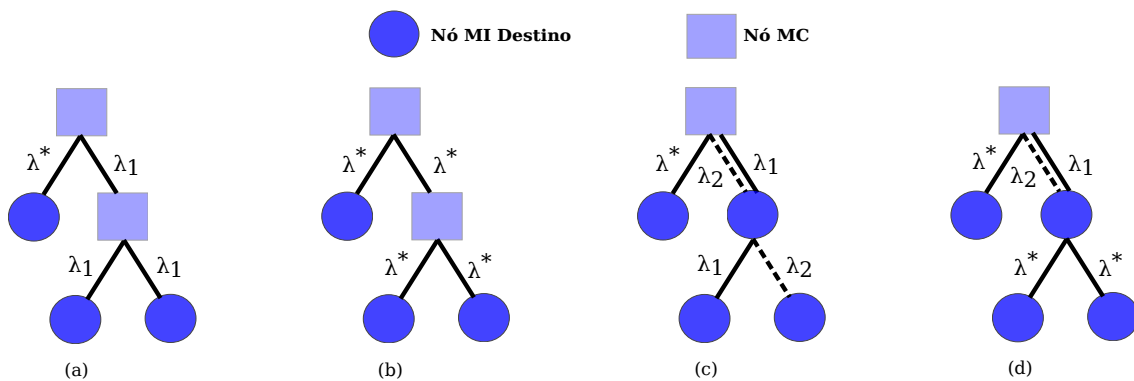


Figura 2.14: Exemplo de um problema MRWA em que os nós MI são DaC. Nos casos (a) e (b) existe um nó MC sem e com capacidade de conversão de comprimento de onda, respectivamente. Em (c) e (d) não existem nós MC, sendo o caso (c) sem conversão de comprimento de onda e o caso (d) com conversão de comprimento de onda. Imagem adaptada de [20].

Nos dois primeiros casos, figuras 2.14 (a) e 2.14 (b), é verificado que, embora o nó MC possa ter ou não conversor de comprimento de onda, a estrutura da árvore Multicast não é afectada. Porém, a forma como os comprimentos de onda são atribuídos às linhas é afectada. Nas árvores das figuras 2.14 (c) e 2.14 (d), são sempre necessários mais canais, em comparação com o exemplo anterior, para adicionar os nós pretendidos. Assim, mais largura de banda em termos do número de canais vai ser consumido.

A eficiência de um mecanismo de encaminhamento Multicast pode ser avaliada a partir do custo de cada sessão, que inclui o custo dos canais de comprimento de onda (número total de canais de comprimento de onda usados em todas as linhas da árvore) e o número máximo de diferentes comprimentos de onda usados por linha numa sessão. Nos artigos [6, 13, 17, 32, 34, 35] são desenvolvidas soluções para o problema MRWA para redes *sparse-splitting*, cujo objectivo passa por minimizar o número de comprimentos de onda usados por cada linha da rede.

2.4.4 Encaminhamento Multicast em Redes Ópticas

Nesta subsecção são apresentadas heurísticas criadas com o intuito de diminuir o custo das árvores Multicast calculadas para uma determinada sessão em redes com as seguintes características:

1. *Sparse-splitting*.
2. Com *sparse wavelength conversion*, onde a conversão dos comprimentos de onda é feita unicamente nos nós MC das redes.
3. Os nós MI das redes são DoC.
4. Não existem custos associados aos nós das redes.
5. O nó fonte da sessão Multicast pode ter múltiplos transmissores e transmitir com diferentes comprimentos de onda.
6. As linhas têm um número ilimitado de comprimentos de onda de forma a evitar bloqueio de uma sessão devido à falta deste recurso, ou seja para redes sem restrição de capacidade nos arcos.

Em [6], são propostos dois esquemas de encaminhamento Multicast para redes não dirigidas que constroem *light-trees* onde, em primeiro, é construída uma árvore que apenas liga os destinos MC da sessão, sucedendo-se a ligação dos restantes *light-paths* que ligam os destinos MI à árvore MC. Os autores apresentam então as heurísticas criadas, a *On-Tree MC node First (OTMCF)* e a *Nearest MC node First (NMCF)*.

A heurística OTMCF tem como principal objectivo minimizar o custo da árvore MC, calculando-a em primeiro lugar. De seguida, são ligados os nós destino MI aos nós MC mais próximos da árvore. A heurística NMCF tenta minimizar o custo da adição dos nós MI à árvore MC. Em primeiro a árvore MC é calculada considerando a união dos nós MC destino com os nós MC de maior proximidade aos nós MI destino e, por fim, os nós MI são adicionados. Em [6], conclui-se que as heurísticas OTMCF e NMCF apresentavam uma melhoria significativa em termos do custo total de canais de comprimento de onda, assim como no número de comprimentos de onda por linha, em comparação com outros algoritmos existentes. Entre as duas heurísticas, a NMCF foi aquela que apresentou melhor desempenho.

De forma a tentar minimizar quer o custo da árvore MC, quer o custo dos caminhos nos quais os nós MI são ligados à árvore MC, foi proposto em [7] o algoritmo *Multicast Using Splitters (MUS)*. Nesta heurística, à semelhança das heurísticas OTMCF e NMCF, é construída inicialmente uma árvore MC e, após esta construção, são ligados, um por um, os nós MI à árvore resultante através dos nós MC mais próximos. No caso de existir um nó MC no caminho adicionado, este passa a ser um dos possíveis nós através do qual o próximo nó destino pode ser adicionado à árvore. A comparação entre todas as heurísticas apresentadas até agora é feita em [7], onde é mostrado que a heurística MUS minimiza o custo total dos canais de comprimento de onda em relação às heurísticas anteriores. Esta

2. Conceitos Preliminares

diminuição provém do facto da heurística considerar os nós MC presentes na árvore MC calculada e os nós MC introduzidos na solução, para a ligação dos restantes destinos MI à árvore.

No artigo [10], os autores introduzem uma nova aproximação, baseada no algoritmo MUS, com a intenção de reduzir ainda mais o custo de uma árvore Multicast. Nesta heurística, é usado de forma repetitiva o algoritmo MUS para tentar diminuir o custo de uma solução previamente calculada com este algoritmo. A heurística criada é a *Sparse Splitting Multicast Routing Heuristic (SSMRH)*. Neste artigo, é feita uma comparação entre o algoritmo e as heurísticas existentes na literatura (NMCF, OTMCF e MUS) e, através de simulações, é concluído que o primeiro apresenta um melhor desempenho em termos de custo médio das árvores calculadas. Para além disso, se a percentagem de nós MC na rede for muito baixa, a redução de custo da árvore alcançada pelo SSMRH, em relação aos outros algoritmos, é mais significativa.

A pensar nas redes mistas, foi introduzido em [11] uma nova heurística de encaminhamento, para este tipo de redes, onde são ligados em primeiro lugar os destinos MC, a partir de um nó MC da árvore actual ou do nó fonte da sessão, por ordem crescente do custo do caminho que os liga à árvore actual. Após a ligação de todos os nós MC, os nós MI são adicionados de forma semelhante. Uma das características desta heurística é que, após a ligação de um destino, são removidos da árvore os destinos anteriormente ligados e é verificado se estes podem ser ligados com um menor custo, partindo dos nós MC que pertencem ao caminho do último nó destino adicionado. Esta heurística será descrita com mais detalhe no capítulo 3 uma vez que foi implementada no âmbito desta dissertação.

2.4.5 Protecção de Sessões Multicast em Redes Ópticas

A enorme quantidade de informação transportada por uma fibra óptica e a possível perda do conteúdo a enviar em caso de falha numa *light-tree*, que pode afectar o seu transporte para os vários destinos, levou à necessidade de implementar técnicas de protecção Multicast eficientes. Estas técnicas são desenvolvidas para providenciar capacidade de sobrevivência da rede em caso de falha única de uma das linhas, a mais comum neste tipo de redes [25], e assim evitar o bloqueio de uma sessão Multicast. Neste contexto, o bloqueio depende também (e principalmente) da capacidade do cálculo de uma árvore que consiga proteger a árvore obtida pelo algoritmo de encaminhamento Multicast utilizado.

Em [26] são propostos vários esquemas de protecção em redes ópticas *Mesh*. Uma forma directa de providenciar protecção a uma árvore Multicast (árvore primária), é estabelecer uma outra árvore (árvore de protecção ou secundária) disjunta nas linhas com a anterior através do algoritmo *Link-Disjoint Trees (LDT)*, descrito na tabela 2.1. Duas árvores são disjuntas nas linhas quando não partilham nenhuma linha da rede. Estas árvores são

usadas para fornecer protecção 1+1, onde ambas as árvores conduzem a informação a levar ao destino pretendido. Assim, quando ocorre uma falha numa linha, os nós destino afectados reconfiguram os seus comutadores para receber as mensagens ópticas a partir da árvore de protecção, ao invés da árvore primária.

-
- 1º Passo** - Criar uma árvore primária usando a heurística H.
 - 2º Passo** - Remover da rede as linhas da árvore primária.
 - 3º Passo** - Criar uma árvore secundária na rede modificada usando a heurística H.
-

Tabela 2.1: Algoritmo de Árvores Disjuntas nas Linhas (H-LDT)

O conceito de árvores disjuntas nas linhas pode ainda ser alargado para árvores disjuntas nos arcos com o algoritmo *Arc-Disjoint Trees* (ADT), descrito na tabela 2.2, onde as árvores primária e secundária podem partilhar linhas mas com direcções opostas. Este esquema de protecção é mais eficiente uma vez que as árvores fornecem protecção 1:1, onde apenas a árvore primária transmite o sinal óptico e a secundária é usada para transportar a informação quando ocorre a falha.

-
- 1º Passo** - Criar uma árvore primária usando a heurística H.
 - 2º Passo** - Remover da rede os arcos da árvore primária.
 - 3º Passo** - Criar uma árvore secundária na rede modificada usando a heurística H.
-

Tabela 2.2: Algoritmo de Árvores Disjuntas nos Arcos (H-ADT)

Um exemplo da utilização do esquema de protecção ADT pode ser visto na figura 2.15, onde são construídas duas árvores (primária e secundária) para realizar uma sessão Multicast, tendo como raiz o nó s e como destinos os nós d_1 e d_2 . Pela análise das árvores calculadas, existe uma linha usada pelas duas soluções, a linha composta pelos arcos (d_1, d_2) e (d_2, d_1) . No caso desta linha falhar, a informação pode ser enviada para o destino d_1 através da árvore primária e para d_2 através da árvore secundária [11].

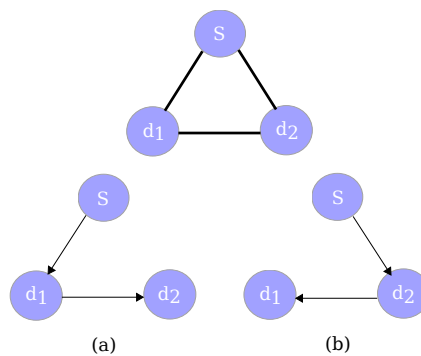


Figura 2.15: Exemplo de duas árvores onde os seus arcos são disjuntos. (a) Árvore primária e (b) árvore secundária. Imagem adaptada de [11].

2. Conceitos Preliminares

A introdução da técnica ADT em heurísticas existentes é feita em [11], onde a heurística MSH-ADT³ é comparada com outras heurísticas existentes combinadas com o esquema de protecção ADT, originando as técnicas NMCF-ADT, OTMCF-ADT e MUS-ADT.

No artigo [26], são ainda apresentados esquemas baseados em protecção aos segmentos e caminhos de uma árvore Multicast. Este tipo de protecção pode ser utilizado quando não é possível encontrar duas árvores disjuntas nos seus arcos. Na protecção por segmentos, a base de funcionamento é semelhante às protecções anteriormente descritas, sendo que agora o esquema está direccionado a proteger um segmento. Um segmento numa árvore é definido como uma sequência de arcos desde a fonte, ou de um nó de ramificação, para um nó folha, ou um nó de ramificação. Já na protecção de caminhos, a ideia é encontrar dois caminhos mais curtos entre cada par fonte e destino. Na figura 2.16 estão expostos todos os esquemas de protecção propostos em [26].

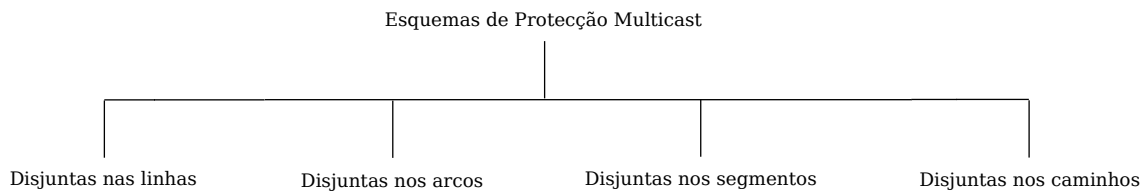


Figura 2.16: Esquemas de protecção Multicast. Imagem adaptada de [26].

³Na nomenclatura dos métodos que se seguem, o acrónimo da técnica de protecção é adicionado no acrónimo de cada heurística indicando que este é o mecanismo de protecção usado.

3

Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

A introdução de capacidade Multicast em redes ópticas WDM levou à necessidade de implementar formas eficientes de construir *light-trees* que tentassem tirar o máximo partido da capacidade de divisão do sinal dos nós da rede. Devido ao elevado custo deste tipo de nós, foi criado o conceito de redes *sparse-splitting*, introduzido no capítulo anterior, que aumenta a complexidade do cálculo de soluções de encaminhamento que levem a árvores com menor custo global. Neste capítulo, dividido em três secções, será feita a descrição detalhada da implementação dos algoritmos usados durante o decorrer deste trabalho para o estudo do encaminhamento Multicast neste tipo de redes. Na secção 3.1 são apresentadas algumas considerações relativas à implementação dos algoritmos, na secção 3.2 é descrita a implementação dos algoritmos e, por fim, em 3.3 é proposta uma nova heurística que pretende diminuir o custo da construção de árvores Multicast em relação às abordagens existentes.

3.1 Algumas Considerações Importantes

Com o intuito de implementar heurísticas que possam calcular soluções para o problema de minimização do custo de uma árvore Multicast, é necessário em primeiro lugar construir um algoritmo que possa determinar o caminho mais curto entre um par de nós de uma rede. Para a resolução deste problema, o algoritmo usado foi o amplamente co-

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

nhecido algoritmo de Dijkstra [14] que foi introduzido em 1959 por Edgar Dijkstra e que encontra o caminho mais curto de um dado nó fonte para todos os outros nós da rede, assumindo arcos com pesos não negativos. Na implementação deste algoritmo optou-se pela utilização de uma *binary heap*.

A representação de uma solução, obtida pelo algoritmo de Dijkstra, é feita através da construção de vectores de predecessores e de distâncias mínimas. Considera-se que num vector de predecessores o valor -1 significa que o nó nessa posição não pertence à árvore e o valor 0 indica que esse é o nó raiz da árvore representada nesse vector. Porém, como estamos a trabalhar em redes ópticas WDM, onde um nó pode ter vários predecessores (desde que estes pertençam a árvores com diferentes comprimentos de onda), uma nova forma de representar as soluções obtidas teve que ser criada. Desta maneira, a forma de representar estas soluções passa por criar uma tabela de vectores de predecessores onde cada um dos vectores da tabela representa uma *light-tree* da solução Multicast. O nó fonte da sessão Multicast é sempre a raiz do vector na primeira posição da tabela de predecessores (ver tabelas 3.1 e 3.2). A par da tabela de predecessores, existe uma tabela de distâncias mínimas, onde a distância de cada nó é calculada em relação ao nó raiz da árvore na correspondente posição da tabela de predecessores.

A descrição do preenchimento das tabelas que representam a árvore pode ser vista no algoritmo 1. Em primeiro lugar é preciso verificar se o vector com o novo caminho pode ser adicionado na posição i da tabela (linha 5 do algoritmo 1). Este caminho será adicionado se:

1. Não partilhar nenhum arco com a árvore da posição i ;
2. Não existe nenhum nó MI, partilhado (ou seja, em comum) com a árvore da posição i , com um predecessor diferente.
3. A sua fonte pertencer à árvore da posição i .

A introdução de um novo caminho (que pode não ter origem na raiz da árvore) nas tabelas referidas anteriormente, requer um conjunto de passos representados pelas operações \oplus e \odot no algoritmo 1. Após o cálculo da posição i da tabela onde vai ser introduzido o novo caminho, são copiados para o vector desta posição, com a operação \oplus , todos os predecessores diferentes de 0 e -1 que representam o novo caminho. Se a posição da tabela estiver vazia, o vector de predecessores do novo caminho é inserido nesta posição sem qualquer modificação. A actualização da tabela de distâncias é feita pela operação \odot . Sabendo qual é o nó raiz do vector da posição i desta tabela, esta operação substitui os valores de distância para cada um dos novos nós da árvore somando o valor da distância do nó raiz com o valor da distância dos novos nós presentes no vector de distâncias do

Input : Tabela de vectores de predecessores *TabPred*, tabela de vectores de distâncias *TabDist*, vector de predecessores *pred* que representa um caminho a adicionar à tabela e o respectivo vector *dist* que representa as distâncias dos nós do caminho à raiz local

Output: Tabela de vectores de predecessores *TabPred* e tabela de vectores de distâncias *TabDist*

```

1 begin
2   i = 0; // Posição inicial da tabela
3   do
4     if TabPred[i] existe then // Se existe a posição i em TabPred
5       if pred pode ser adicionado na posição i de TabPred then
6         // Novo ramo de TabPred
7         TabPred[i] = TabPred[i] ⊕ pred;
8         TabDist[i] = TabDist[i] ⊙ dist; // Novo ramo de TabDist
9         i = 0; // i=0 para terminar ciclo
10        else
11         i = i + 1; // Próxima posição de pesquisa
12        end
13      else // Se não existe a posição i em TabPred
14        TabPred[i] = pred; // pred na nova posição i em TabPred
15        TabDist[i] = dist; // dist na nova posição i em TabDist
16        i = 0; // i=0 para terminar ciclo
17      end
18    While i ≠ 0; // Quando i = 0 o pred já está na TabPred
19    É terminado o algoritmo;
20    return TabPred, TabDist
21 end

```

Algoritmo 1: Algoritmo de preenchimento da tabela de predecessores e de cálculo das distâncias correspondentes (AFTP)

caminho. No caso da posição *i* da tabela estar vazia, o vector de distâncias é inserido directamente nessa posição.

Na figura 3.1 é ilustrado um exemplo de uma árvore Multicast onde o nó 1 é fonte, os nós 3, 4 e 7 são os nós destino (nós a preto) e o nó 2 é MC. As tabelas 3.1 e 3.2 mostram a solução final da tabela de predecessores e de distâncias mínimas, respectivamente, para esta árvore Multicast.

É considerado que, em primeiro lugar, é adicionado o caminho mais curto desta solução, o caminho do nó 1 para o nó 3 (preenchendo as colunas 1, 2, 3 e 5 da posição 1 das tabelas 3.2 e 3.1). Depois de inserido o vector de predecessores na primeira posição da tabela, é calculado um novo caminho que tanto pode partir do nó fonte como do nó MC do caminho adicionado (o nó 2). Este caminho é, por exemplo, o caminho do nó 2 para

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

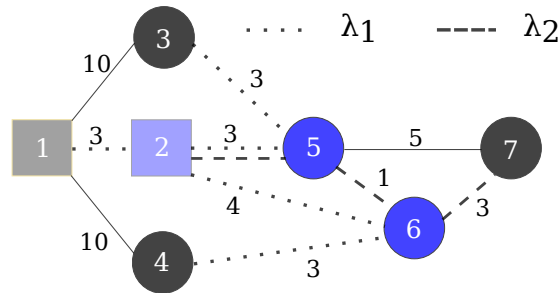


Figura 3.1: Exemplo de uma árvore Multicast para a rede da figura 2.12.

o nó 7, passando pelos nós 5 e 6 (ver figura 3.1). Segundo as regras de preenchimento da tabela, este caminho não poderá ser adicionado na primeira posição visto que viola a regra 1 (o arco (2,5) é partilhado pelo caminho e pela árvore desta posição). Como a tabela só contém um vector, o caminho é adicionado na seguinte posição da tabela, a segunda. É de notar que o nó fonte deste caminho é o nó 2, logo tanto o vector de predecessores como o vector de distâncias mínimas irá ter o valor 0 (nó fonte) e valor de distância 0 na segunda posição do vector. O próximo caminho a adicionar é o caminho do nó 2 para o nó 4. Seguindo os passos de preenchimento da tabela, este caminho não viola nenhum das regras quando verificamos se é adicionável na primeira posição da tabela. É então nesta posição que é adicionado o caminho (preenchendo as colunas 4 e 6 da posição 1 das tabelas 3.2 e 3.1). Se, por exemplo, este caminho não pudesse ser colocado na primeira posição da tabela, também não seria colocado na segunda devido a violar a regra 2, visto que o caminho e a árvore desta posição partilham o nó 6 como nó intermédio com predecessores diferentes.

	Nós da Rede						
Índice da Tabela	1	2	3	4	5	6	7
1	0	1	5	6	2	2	-1
2	-1	0	-1	-1	2	5	6

Tabela 3.1: Tabela de predecessores da árvore Multicast representada na figura 3.1.

	Nós da Rede						
Índice da Tabela	1	2	3	4	5	6	7
1	0	3	9	10	6	7	∞
2	∞	0	∞	∞	3	4	7

Tabela 3.2: Tabela de distâncias mínimas da árvore Multicast representada na figura 3.1.

Esta forma de construir a árvore Multicast permite identificar o número de *light-trees* presentes na solução, isto porque cada vector existente na tabela corresponde a uma *light-tree*. Assim, o número de *light-trees* presentes na árvore Multicast é igual ao tamanho da tabela de predecessores. Esta premissa pode ser verificada para a solução apresentada na figura 3.1, onde podemos ver que existem duas *light-trees* na solução e, consequentemente, o tamanho das tabelas que representam a solução (tabelas 3.1 e 3.2) é dois.

Se este trabalho se focasse em redes sem capacidade de conversão de comprimentos de onda, onde a solução seria uma *light-forest*, seria possível aferir o número máximo de comprimentos de onda por linha através, igualmente, do tamanho das tabelas. Em redes com conversão de comprimento de onda apenas em alguns nós, esta assunção já não é verdadeira visto que é possível existirem duas árvores diferentes, e que são adicionadas em posições diferentes da tabela, que podem partilhar o mesmo comprimento de onda.

3.2 Heurísticas Implementadas

Antes da descrição das heurísticas implementadas, é necessário introduzir a notação adicional que irá ser usada neste capítulo.

1. Uma sessão Multicast é representada por $r = (s, D)$, onde s é a raiz do problema e $D = \{d_1, d_2, \dots, d_p\}$ um conjunto finito de destinos.
2. Dentro do conjunto D existem dois sub-conjuntos, o conjunto de nós destino MC, D_{MC} , e o conjunto de nós destino MI, D_{MI} .
3. Se definirmos o custo de canal de comprimento de onda em cada arco a_{ij} como $c_{a_{ij}}$, e sendo $\eta_{a_{ij}}$ o número de canais de comprimento de onda atravessando o arco a_{ij} , o custo $C(T)$ de uma árvore Multicast T , obtida por uma qualquer heurística, pode ser vista como a soma do custo dos arcos pertencentes à árvore, ou seja $C(T) = \sum_{a_{ij} \in T} c_{a_{ij}} \eta_{a_{ij}}$.
4. N_{MC} representa o conjunto de nós MC da rede enquanto que T_{MC} representa o conjunto de nós MC presentes numa árvore Multicast (incluindo o nó raiz).
5. O conjunto de todos os nós presentes numa árvore T será designado por T_N .
6. O custo do caminho mais curto que liga os nós i e j é representado por c_{ij} .
7. O grafo induzido pelo caminho mais curto que liga os nós i e j é representado por P_{ij} .

3.2.1 *Multicast Using Splitters (MUS)*

A heurística de encaminhamento MUS é apresentada em [7] com o intuito de tentar minimizar o custo de uma árvore Multicast. Esta heurística tem por base a construção de uma árvore $T_{D_{MC}}$, que apenas liga os destinos MC da sessão, sucedendo-se a ligação dos restantes *light-paths* que ligam os destinos MI à árvore $T_{D_{MC}}$. O algoritmo de cálculo de uma árvore MC, usado nesta heurística, é proposto em [6]. Neste mesmo artigo, os autores provam que o cálculo da árvore MC com o menor custo é equivalente a encontrar uma árvore de Steiner na respectiva rede MC. O processo de encontrar uma árvore MC de custo mínimo é designado de *Auxiliary Network Transformation (ANT)* e pode ser executado da seguinte forma:

1. Construir uma rede MC, designada M_G , a partir da rede original G . A construção desta rede é guiada pelos seguintes passos:
 - (a) Todos os nós MC na rede G são incluídos num conjunto W .
 - (b) Se existe um caminho entre dois nós MC em G , estes dois nós são ligados em M_G .
 - (c) O custo do arco do nó i para o nó j em M_G é definido como o custo do caminho com menor peso que liga o nó i ao nó j em G , para todos os $i, j \in W$.
2. De seguida, uma heurística para o cálculo de uma árvore de Steiner é aplicada em M_G construindo uma árvore de custo mínimo $T_{r'}$ para a sessão Multicast $r' = (s, D_{MC})$.
3. A árvore final T é obtida substituindo cada arco (i, j) em $T_{r'}$ pelo correspondente caminho de custo mínimo do nó i para o nó j em G .

A figura 3.2 apresenta um exemplo da utilização do método ANT (as linhas mais grossas indicam os arcos na árvore). O nó fonte s está a sombreado e os nós d_1 e d_2 são os nós destino. Daqui em diante, os nós MC serão representados nas imagens como nós de forma rectangular enquanto que os MI de forma circular.

A primeira parte da implementação da heurística MUS recai então na construção desta árvore MC, e pode ser vista na linha 2 do pseudo-código no algoritmo 2. Devido à complexidade da transformação do grafo numa rede MC (um dos passos da técnica), os autores deste esquema propõem a construção de uma árvore MC de uma forma alternativa, através da observação de que o custo de um canal de comprimento de onda entre dois nós MC numa árvore MC de uma rede constituída apenas por nós MC é igual ao custo de um caminho de peso mínimo entre os mesmos dois nós na *light-tree* resultante na rede original. Esta premissa implica que se for aplicada uma heurística para cálculo de uma árvore

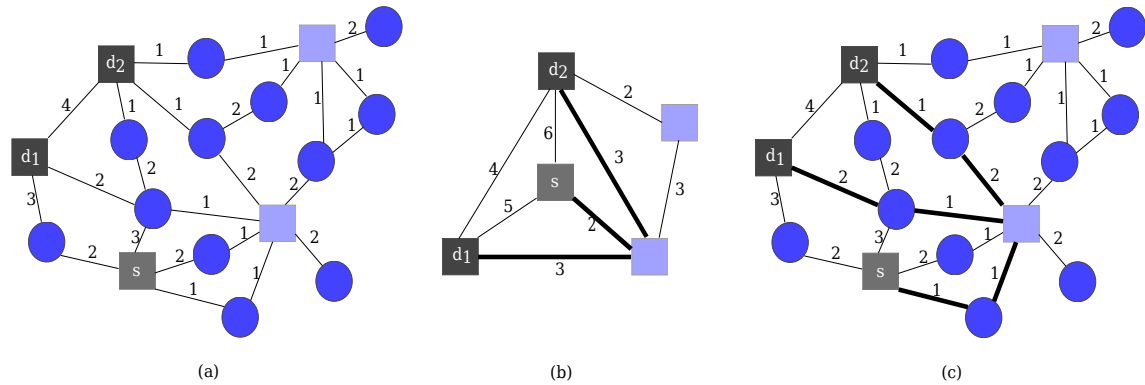


Figura 3.2: Exemplo da utilização do ANT. (a) Topologia da rede onde apenas estão assinalados os nós destino MC, (b) a respectiva rede MC e a árvore de Steiner calculada e (c) a *light-tree* resultante. Imagem adaptada de [6].

de Steiner à construção da árvore Multicast os mesmos resultados podem ser obtidos sem uma transformação explícita da rede, como a ANT. A heurística citada pelos autores é a *Shortest Path Heuristic (SPH)* [29], onde é inicializada a árvore Multicast com um nó arbitrário e completada com a adição progressiva do nó destino mais próximo, através do caminho mais curto entre o nó e a árvore, até se encontrarem todos os nós destinos na árvore Multicast. Devido à menor complexidade, este foi o caminho escolhido para substituir o método ANT e assim resolver o primeiro passo do MUS. Esta heurística foi construída e adaptada para a construção de uma árvore MC. Em primeiro lugar, nesta adaptação, a árvore é inicializada sempre com o nó raiz. Em segundo, apenas os nós MC, ou o nó raiz da sessão, podem ser utilizados como nós de pesquisa para a adição de nós destino. Por fim, é ainda necessário ter em conta as regras usadas para a construção de uma *light-tree*. No caso de não existirem nós MC no conjunto de nós destino, a árvore MC será apenas constituída pelo nó fonte.

No segundo passo da heurística MUS, são introduzidos os nós destino MI na árvore MC calculada. Para efectuar esta ligação, é utilizado o algoritmo de Dijkstra para o cálculo do caminho mais curto entre todos os nós MC da árvore (considerando igualmente o nó fonte) e todos os nós destino, e é escolhido o caminho de menor custo para ser adicionado. Este passo é executado até todos os destinos pertencerem à árvore actual. No algoritmo 2, a adição dos nós destino MI à árvore é representada a partir da linha 3.

3.2.2 *Sparse Splitting Multicast Routing Heuristic (SSMRH)*

A heurística *Sparse Splitting Multicast Routing Heuristic (SSMRH)* é apresentada em [10] com o intuito de calcular árvores Multicast que solucionem determinadas sessões, utilizando de forma repetitiva a heurística MUS. O seu pseudo-código é apresentado no algoritmo 3.

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

```
Input :  $G(V, E)$ ,  $N_{MC}$ ,  $r = (s, D)$  e função de custo  $c_e, e \in E$   
Output: Light-tree  $T$  para  $r$  em  $G$   
  
1 begin  
2    $T = \text{ANT}(G, N_{MC}, c_e, D_{MC});$  // Árvore MC obtida pelo ANT  
3   while  $D_{MI} \neq \emptyset$  do  
4      $p_{min} = \emptyset;$  // Caminho vazio  
5      $c(p_{min}) = \infty;$   
6     foreach  $y \in D_{MI}$  do  
7        $c_{xy} = \min_{i \in T_{MC}} c_{iy};$  //  $c_{xy}$  é o custo do caminho com menor custo  
8       if  $c_{xy} < c(p_{min})$  then // Actualiza  $n$   
9          $n = y;$  //  $n$  é o novo nó de menor distância a  $T$   
10         $p_{min} = p_{xy};$  // Arcos e nós do caminho que liga  $x$  e  $y$   
11      end  
12    end  
13    // Adição de  $p_{min}$  à árvore  $T$  com o APTP (ver algoritmo 1)  
14     $T = \text{APTP}(T, p_{min});$   
15     $D_{MI} = D_{MI} \setminus \{n\};$  // Nó  $n$  está na árvore, sai de  $D_{MI}$   
16  end  
17 return  $T$   
18 end
```

Algoritmo 2: MUS

Aproveitando a implementação da heurística MUS, a heurística SSMRH é facilmente construída. Primeiro é necessário calcular a árvore Multicast com o auxílio da heurística MUS. Após o cálculo da árvore Multicast original, identifica-se o conjunto de nós MC que não estão na árvore (linha 4 do algoritmo 3). Em seguida, cada um dos nós deste conjunto é adicionado individualmente no conjunto de destinos, e é calculada a solução para cada uma destas novas sessões (linha 9). Se o custo de uma das novas árvores for menor que o custo da solução original, o nó correspondente a essa árvore passa a estar definitivamente no conjunto de nós destino. De seguida, são novamente identificados os nós MC que não pertencem à árvore Multicast e repete-se o procedimento até não existir uma melhoria, em termos de custo, na árvore Multicast final. Uma das principais características e vantagens da heurística SSMRH, é que nunca calculará uma árvore com maior custo do que a melhor árvore calculada com o MUS, sendo possível assim diminuir o custo de uma árvore Multicast.

3.2.3 *Mixed-graph Sparse-splitting Heuristic (MSH)*

As heurísticas até agora introduzidas foram direccionadas para redes não dirigidas. Para introduzir um novo método de cálculo de árvores Multicast em redes mistas, é pro-

```

Input :  $G(V,E)$ ,  $N_{MC}$ ,  $r = (s,D)$  e função de custo  $c_e, e \in E$ 
Output: Light-tree  $T$  para  $r$  em  $G$ 

1 begin
2    $T = \text{MUS}(G, r);$  // Árvore inicial obtida pelo MUS
3    $c = C(T);$ 
4    $MC = N_{MC} \setminus T_{MC};$  // Nós MC não pertencentes à árvore inicial
5   while  $MC \neq \emptyset$  do
6     foreach  $MC_i \in MC$  do
7        $D_i = D \cup \{MC_i\};$  // Nó  $MC_i$  temporariamente adicionado a  $D$ 
8        $r_i = (s, D_i);$ 
9        $T_i = \text{MUS}(G, r_i);$  // Nova árvore obtida
10       $c_i = C(T_i);$  // Guarda custo de cada nova árvore obtida
11    end
12     $i^* = \arg \min_i c_i;$  // Identifica nó MC candidato a incluir em  $D$ 
13    if  $c_{i^*} < c$  then //  $c_{i^*}$  custo da árvore com menor custo calculada
14       $c = c_{i^*};$ 
15       $D = D_{i^*};$  // Nó  $MC_{i^*}$  é permanentemente adicionado a  $D$ 
16       $MC = N_{MC} \setminus T_{MC_{i^*}};$  // Nós MC que não estão na árvore
17    else // Não é possível alargar  $D$  e melhorar o custo de  $T$ 
18       $T = \text{MUS}(G, r);$  // Árvore com  $D$  alargado
19       $MC = \emptyset;$  // Para terminar
20    end
21  end
22  return  $T$ 
23 end

```

Algoritmo 3: SSMRH

posta em [11] uma nova heurística de encaminhamento, sendo ainda introduzida uma técnica de protecção para o caso de uma falha única nos arcos da rede¹. Parafraseando os autores, mesmo que as heurísticas sejam construídas para redes não dirigidas, na prática a rede deve ser modelada como uma rede mista pois:

1. Quando alguns pedidos retêm os recursos da rede, o grafo resultante é misto, logo o encaminhamento para pedidos subsequentes será calculado numa rede mista.
2. Quando, por exemplo, deve ser calculada uma árvore de protecção disjunta nos arcos da árvore primária, após a remoção dos arcos da árvore primária, a árvore secundária será calculada numa rede mista.
3. É possível que certos nós da rede possam receber informação a partir de outros nós,

¹Neste capítulo é feita apenas a análise à parte de encaminhamento realizada pela heurística proposta em [11].

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

mas que não a possam transmitir devido a limitações da camada física.

A heurística criada é designada de *Mixed-graph Sparse-splitting Heuristic (MSH)* e o seu pseudo-código é descrito no algoritmo 4. Neste algoritmo, os destinos são separados em dois tipos, destinos MC e MI. Os primeiros a serem inseridos na árvore, inicialmente constituída apenas pelo nó raiz, são os destinos MC devido à sua capacidade de replicação do sinal. Desta forma, é melhorada a possibilidade de se usar menos *light-trees*, e consequentemente menos comprimentos de onda na solução final. O primeiro passo é realizado através do algoritmo de Dijkstra, onde se calcula o menor caminho entre todos os nós MC da árvore e todos os nós destino MC (linha 5 do algoritmo 4) e se adiciona o caminho na árvore (linha 6 do algoritmo 4). O cálculo dos caminhos de menor custo entre os nós da rede é realizado previamente à execução do algoritmo, ou seja no algoritmo é apenas verificado qual o caminho de menor custo. Na escolha deste caminho, foi ainda considerada uma nova nuance não referida pelos autores de [11]. No caso de um empate no menor custo de dois, ou mais, caminhos possíveis de adição na árvore actual, o algoritmo irá adicionar à árvore aquele que apresentar pelo menos um nó MC. Esta nuance é adicionada para conceder à árvore mais opções de ligação a um próximo destino. Se vários caminhos tiverem o mesmo custo e pelo menos um nó MC no seu caminho, o algoritmo irá adicionar o caminho com menos arcos. Esta opção é feita para aumentar a probabilidade de protecção das árvores, tema que será introduzido no capítulo 4. Após a adição do caminho determinado à árvore actual, deve ser mantido nesta apenas o percurso que liga o nó fonte da sessão ao último nó destino adicionado, passo representado na linha 7. Em termos de representação da árvore, se o caminho calculado tiver como raiz o nó fonte então o caminho passa a ser a árvore actual. Caso contrário, um *novo procedimento* será executado que permite obter o caminho desde o nó raiz da sessão Multicast até ao último nó destino inserido. Este procedimento é equivalente à chamada da função APTP seguida da criação da representação do caminho desejado² (ver linhas 6 e 7 do algoritmo 4).

No próximo passo da heurística MSH (linhas 9 a 13 do algoritmo 4), são inseridos os nós destino anteriormente pertencentes à árvore (antes da inserção do novo nó). O procedimento é semelhante à ligação de um novo nó destino à árvore. A grande diferença está no facto de agora a árvore ficar intacta, ou seja não é mantido apenas um caminho. Em primeiro lugar são adicionados os destinos MC, pelas razões anteriormente explicadas, e de seguida os MI (linhas 15 a 19 do algoritmo 4). Os nós fonte destes caminhos são os nós MC que pertencem, ou que passam a pertencer após a ligação de um caminho,

²Opta-se por esta descrição para este processo por maior simplicidade. Em termos de código, o procedimento de encontrar o caminho entre o nó raiz da sessão e o nó destino adicionado requer a criação de uma tabela auxiliar. Nesta tabela é inserido o último caminho adicionado e é preenchido o restante percurso que liga o nó raiz da sessão ao nó raiz do caminho adicionado (sendo a procura deste percurso feita na tabela original).

```

Input :  $G(V,E)$ ,  $N_{MC}$ ,  $r = (s,D)$  e função de custo  $c_a, a \in A$ 
Output: Light-tree  $T$  para  $r$  em  $G$ 

1 begin
2    $T = G'(\{s\}, \emptyset), T_{MC} = G'(\{s\}, \emptyset), D'_{MC} = \emptyset, D'_{MI} = \emptyset;$ 
3   Repetir o próximo passo primeiro para  $\mathcal{A} = D_{MC}$  e depois para  $\mathcal{A} = D_{MI}$ ;
4   while  $\mathcal{A} \neq \emptyset$  do
      // Cálculo do caminho mais curto de  $x$  para  $y$ 
5      $c_{xy} = \min_{i \in T_{MC}, j \in \mathcal{A}} c_{ij};$  //  $c_{xy}$  é o custo do caminho com menor custo
      // Adição de  $p_{xy}$  à árvore  $T$  com o APTP (ver algoritmo 1)
6      $T = \text{APTP}(T, p_{xy});$ 
7      $T = T_{p_y};$  //  $T$  contem apenas o caminho de  $s$  para o nó  $y$ 
8      $D''_{MC} = D'_{MC};$ 
9     while  $D''_{MC} \neq \emptyset$  do
10       $c_{wz} = \min_{i \in T_{MC}, j \in D''_{MC}} c_{ij};$  // Custo mínimo do caminho de  $w$  para  $z$ 
11       $T = \text{APTP}(T, p_{wz});$  // Adição de  $p_{wz}$  à árvore  $T$  com o APTP
12       $D''_{MC} = D''_{MC} \setminus \{z\};$ 
13    end
14     $D''_{MI} = D'_{MI};$ 
15    while  $D''_{MI} \neq \emptyset$  do
16       $c_{wz} = \min_{i \in T_{MC}, j \in D''_{MI}} c_{ij};$  // Custo mínimo do caminho de  $w$  para  $z$ 
17       $T = \text{APTP}(T, p_{wz});$  // Adição de  $p_{wz}$  à árvore  $T$  com o APTP
18       $D''_{MI} = D''_{MI} \setminus \{z\};$ 
19    end
20    if  $y \in N_{MC}$  then // Se o nó destino  $y$  adicionado a  $T$  é MC
21       $D'_{MC} = D'_{MC} \cup \{y\};$  // Nós destino MC em  $T$ 
22    else
23       $D'_{MI} = D'_{MI} \cup \{y\};$  // Nós destino MI em  $T$ 
24    end
25     $\mathcal{A} = \mathcal{A} \setminus \{y\};$  // Remoção de  $\mathcal{A}$  do nó  $y$  adicionado
26  end
27  return  $T$ 
28 end
    
```

Algoritmo 4: MSH

à árvore. Os resultados obtidos pelo algoritmo de Dijkstra são usados, mais uma vez, para encontrar sucessivamente o caminho de menor custo. Neste passo, a adição dos nós destino é efectuada na tabela segundo as regras de preenchimento anteriormente referidas. Após a adição de todos os destinos MC, o algoritmo é novamente executado seguindo os mesmos passos para os destinos MI (linha 3 do algoritmo 4).

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

A figura 3.3 ilustra um exemplo [11] da utilização do algoritmo MSH numa rede mista, onde a fonte da sessão é o nó s e os nós destino (preenchidos a preto) são os nós d_1, d_2, d_3 e d_4 (os arcos que vão sendo considerados como parte da árvore estão a tracejado).

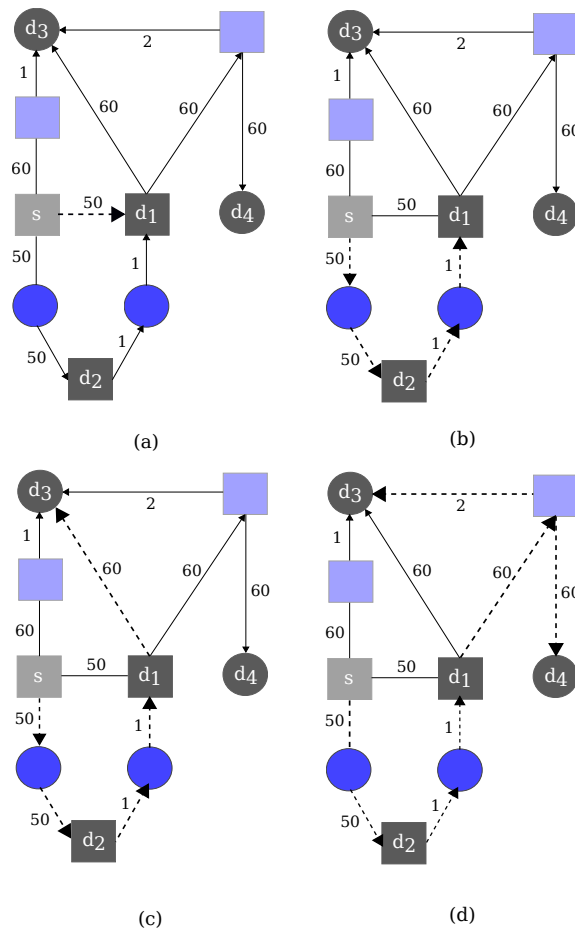


Figura 3.3: Exemplo da utilização do MSH. Em (a) é adicionado o destino d_1 , em (b) d_2 , em (c) d_3 e, finalmente, em (d) d_4 . Imagem adaptada de [11].

Seguindo os passos deste algoritmo, em primeiro lugar (com $\mathcal{A} = D_{MC}$) serão ligados à árvore os nós destino MC, d_1 e d_2 . Na figura 3.3 (a), a heurística começa por ligar o nó MC mais próximo da fonte, d_1 . De seguida, será ligado o nó d_2 ao nó MC, ou ao nó fonte (da árvore actual) mais próximo. Após esta ligação, é mantido na árvore apenas o caminho do nó fonte até d_2 (linha 7 com $y = d_2$). Na continuação do algoritmo, os destinos que já pertenciam à árvore (d_1) são de novo adicionados (ciclo das linhas 9 a 13). Na figura 3.3 (b) é representada a ligação dos nós destino MC. Na sequência do algoritmo, são adicionados os nós destino MI, d_3 e d_4 , seguindo o mesmo princípio, mas agora com $\mathcal{A} = D_{MI}$. Em primeiro é adicionado o nó d_3 , através do nó MC d_1 que pertence à árvore actual, e é mantido o caminho que acaba neste nó e tem início na fonte da sessão. Como os nós destino MC já fazem parte deste caminho, é ignorado o passo da adição destes nós

na árvore. A árvore Multicast obtida está ilustrada na figura 3.3 (c). Por fim, é inserido o último nó destino, d_4 , é mantido o caminho da raiz para este nó (caminho que liga o nó s a d_4 representado na figura 3.3 (d)) e são adicionados, respectivamente, os nós destino MC e MI que estavam na árvore anterior. O resultado final deste exemplo será a árvore Multicast ilustrada na figura 3.3 (d).

Em [11] são feitas comparações entre as heurísticas OTMCF, NMCF, MUS e MSH no contexto da sobrevivência de sessões Multicast, sendo apresentados resultados para o custo médio das árvores (primária e de protecção) calculadas e taxas de bloqueio. No contexto da minimização do custo de uma árvore Multicast, não são apresentados resultados, assim como não é feita esta análise para a heurística SSMRH. Serão então apresentados mais à frente, num capítulo próprio para o efeito, os resultados dos testes computacionais desenvolvidos durante este trabalho que permitem tirar algumas conclusões sobre a eficiência destes algoritmos, nomeadamente sobre a superioridade das heurísticas MSH e SSMRH em termos do custo médio das árvores Multicast calculadas.

3.3 Heurística Proposta

Considerando que a heurística SSMRH corresponde a uma melhoria da heurística MUS, é teoricamente possível a obtenção de melhores resultados, para o tipo de problemas abordados neste capítulo, caso a mesma ideia seja aplicada à heurística MSH. Com base no funcionamento do SSMRH, é proposta, de seguida, uma nova heurística criada para minimizar ainda mais o custo de uma árvore Multicast que soluciona o problema de encaminhamento numa rede Multicast *sparse-splitting* com *sparse wavelength conversion* para uma determinada sessão Multicast.

3.3.1 *New Mixed-graph Sparse-splitting Heuristic (NMSH)*

Na nova heurística, designada *New Mixed-graph Sparse-splitting Heuristic (NMSH)*, será usada de forma repetitiva a heurística MSH, previamente construída, de forma a encontrar a árvore Multicast de menor custo.

A heurística começa por calcular uma árvore para uma determinada sessão Multicast usando a técnica MSH. De seguida, os nós MC não pertencentes à árvore Multicast calculada são adicionados individualmente ao conjunto de destinos, e é calculada, com o auxílio do MSH, a respectiva árvore de cada um destes conjuntos. Se existir alguma árvore cujo custo seja menor que o da solução inicial, o destino previamente adicionado nesta sessão será adicionado permanentemente no conjunto de destinos. O algoritmo é repetido após o cálculo de cada árvore, até não ser possível melhorar o custo da árvore Multicast que satisfaz a sessão. O pseudo-código desta heurística pode ser visto no algoritmo 5.

3. Heurísticas de Encaminhamento Multicast em Redes *Sparse-Splitting*

```
Input :  $G(V,E)$ ,  $N_{MC}$ ,  $r = (s,D)$  e função de custo  $c_a, a \in A$   
Output: Light-tree  $T$  para  $r$  em  $G$   
  
1 begin  
2    $T = \text{MSH}(G,r)$ ; // Árvore inicial obtida pelo MSH  
3    $c = C(T)$ ;  
4    $MC = N_{MC} \setminus T_{MC}$ ; // Nós MC não pertencentes à árvore inicial  
5   while  $MC \neq \emptyset$  do  
6     foreach  $MC_i \in MC$  do  
7        $D_i = D \cup \{MC_i\}$ ; // Nó  $MC_i$  temporariamente adicionado a  $D$   
8        $r_i = (s,D_i)$ ;  
9        $T_i = \text{MSH}(G,r_i)$ ; // Nova árvore obtida  
10       $c_i = C(T_i)$ ; // Guarda custo de cada nova árvore obtida  
11    end  
12     $i^* = \underset{i}{\text{argmin}} c_i$ ; // Identifica nó MC candidato a incluir em  $D$   
13    if  $c_{i^*} < c$  then //  $c_{i^*}$  custo da árvore com menor custo calculada  
14       $c = c_{i^*}$ ;  
15       $D = D_{i^*}$ ; // Nó  $MC_i$  é permanentemente adicionado a  $D$   
16       $MC = N_{MC} \setminus T_{MC_{i^*}}$ ; // Nós MC que não estão na árvore  
17    else // Não é possível alargar  $D$  e melhorar o custo de  $T$   
18       $T = \text{MSH}(G,r)$ ; // Árvore com  $D$  alargado  
19       $MC = \emptyset$ ; // Para terminar  
20    end  
21  end  
22  return  $T$   
23 end
```

Algoritmo 5: NMSH

Computacionalmente, a implementação da heurística apenas requer um uso repetitivo do algoritmo MSH, sendo por isso de fácil construção. Ao usar esta forma de construir uma árvore, é garantido que a solução final nunca terá um custo maior que a melhor árvore calculada pelo algoritmo MSH, existindo assim uma vantagem óbvia, em termos de redução do custo de uma árvore, na utilização desta técnica.

4

Protecção de Sessões Multicast em Redes Sparse-Splitting

Devido à grande quantidade de informação transportada nas fibras de redes ópticas WDM, o problema da protecção de uma sessão Multicast nestas redes é fulcral em ordem a minimizar as perdas de tráfego em caso de falha única numa linha de fibra (*single-link*), a forma predominante de falhas nesta categoria de redes. Neste capítulo são introduzidos os esquemas de protecção desenvolvidos para resolver este tipo de falhas. Na secção 4.1 é descrita a implementação do esquema de protecção ADT nos métodos de encaminhamento Multicast utilizados nesta dissertação. Na secção 4.2 é proposta uma nova abordagem na protecção de sessões Multicast utilizando igualmente o conceito de árvores disjuntas nos seus arcos.

4.1 Protecção *Arc-Disjoint Trees (ADT)*

Assumindo que apenas ocorrem falhas únicas na linha de uma rede, foram apresentados em [26] dois métodos de protecção baseados na noção de arcos e linhas disjuntas, o *Arc-Disjoint Trees (ADT)* e *Link-Disjoint Trees (LDT)*. Ambas as técnicas protegem a rede da falha de uma das suas linhas mas como este trabalho incide na protecção em redes mistas, e visto que o esquema LDT ocupa mais recursos em comparação [11], é abordado apenas o método ADT.

4. Protecção de Sessões Multicast em Redes Sparse-Splitting

O procedimento para o cálculo de duas árvores disjuntas nos arcos segue então três passos:

1. Calcular a *light-tree* primária usando um algoritmo de encaminhamento Multicast.
2. Remover os arcos da árvore primária da rede original.
3. Calcular a *light-tree* secundária no grafo resultante usando um algoritmo de encaminhamento Multicast.

A selecção de um algoritmo de encaminhamento Multicast eficiente é importante na diminuição da probabilidade de bloqueio de uma sessão. O facto de um algoritmo poder calcular *light-trees* com menos arcos resultará num maior número de opções, ao nível do número de arcos, na construção de uma árvore secundária e assim no aumento da possibilidade de esta existir.

Analisando os passos da construção de duas árvores disjuntas nos arcos, é possível verificar que o foco da técnica ADT está na remoção, na rede original, dos arcos pertencentes à árvore calculada por uma heurística H , tornando possível o cálculo de uma nova árvore disjunta nos arcos à árvore primária, independentemente de ser usada uma nova heurística, ou não, no cálculo desta árvore. A implementação da técnica passa então por remover os arcos pertencentes à árvore primária calculada da estrutura usada para representar a rede, estrutura que neste trabalho é uma lista de adjacências. Na prática, esta remoção não é literal, ou seja, de forma a não ser perdida a informação sobre os arcos que fazem parte da rede, em vez de retirar da lista de adjacências os arcos da árvore primária são substituídos os seus pesos, adicionado a cada um deles uma constante igual ao número de arcos existentes na rede multiplicado pelo valor do maior peso de todos os arcos representados. Esta constante será designada de α . Utilizando esta modificação, a árvore secundária será calculada na nova representação da rede onde os arcos inexistentes são representados com um peso muito superior ao seu peso original. Assim, no cálculo de caminhos de custo mínimo, os arcos com peso muito elevado serão sempre preteridos em relação a outros arcos existentes na rede. Se a árvore secundária obtida contiver na tabela de vectores de distâncias nós cujo valor da distância é superior a α , então a árvore obtida contém arcos inexistentes e, logo, é inadmissível. No final do cálculo da árvore secundária, a lista de adjacências é restaurada, ou seja, os arcos removidos são repostos de novo na rede utilizando a cópia do peso original do arco, existente em cada uma das células das listas ligadas pertencentes à lista de adjacências.

No artigo [11] a técnica de protecção ADT é incorporada nas heurísticas de encaminhamento MUS e MSH, criando assim as heurísticas MSH-ADT e MUS-ADT. Neste mesmo artigo é feito o estudo da taxa de bloqueio resultante do uso desta protecção nas

heurísticas mencionadas. Embora as simulações apresentem, para as características das redes utilizadas nas simulações, um bloqueio significativamente baixo, existem limitações nesta técnica que podem ser superadas introduzindo no método capacidade de antever se é garantida a possibilidade de, para uma determinada árvore sob construção, ser calculada uma árvore secundária que permita a sobrevivência da sessão Multicast, mantendo a condição das árvores calculadas serem disjuntas nos seus arcos. Tendo em vista a diminuição da taxa de bloqueio de pedidos Multicast, é proposta uma nova aproximação do procedimento executado para o cálculo de um par de árvores disjuntas nos arcos.

4.2 Protecção ADT Proposta

Antes de apresentar a nova técnica de protecção ADT proposta, e de forma a expor as limitações da técnica de protecção ADT, é apresentado um exemplo da construção de duas árvores disjuntas nos arcos através da heurística MSH-ADT na rede ilustrada na figura 4.1 (a). Nesta rede, o nó fonte é o nó s e os nós destino são os nós d_1 , d_2 , e d_3 (nós a preto). Recordar-se que os nós MC da rede têm forma rectangular e os MI circular.

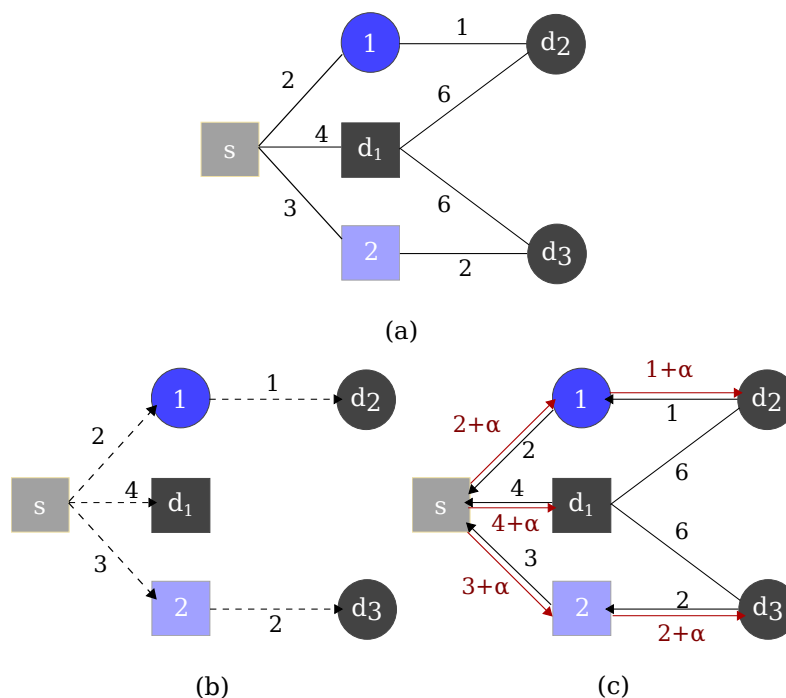


Figura 4.1: Exemplo de (a) uma rede *sparse-splitting* onde (b) representa a árvore primária obtida pela heurística MSH e (c) a rede resultante da remoção – arcos de custo incrementado de α – dos arcos pertencentes à árvore primária.

Utilizando a heurística MSH, a árvore Multicast obtida para esta sessão está ilustrada na figura 4.1 (b) e o seu custo total é igual a 12. Para construir uma segunda árvore,

4. Protecção de Sessões Multicast em Redes Sparse-Splitting

disjunta nos arcos à árvore primária, e seguindo a técnica de protecção ADT, são retirados da rede original os arcos pertencentes à árvore primária. A nova rede obtida é representada na figura 4.1 (c). Obviamente, a árvore primária calculada não pode ser protegida através do cálculo de uma árvore secundária na rede resultante. No entanto, existem caminhos suficientes para proteger uma outra árvore primária diferente da calculada inicialmente. Para resolver este problema, é apresentada uma nova abordagem que integra no mesmo algoritmo a determinação da árvore primária e secundária.

Considerando a rede da figura 4.1 (a), e utilizando a heurística MSH, serão construídas duas árvores disjuntas nos arcos segundo os passos desta nova abordagem. Na figura 4.2 pode ser acompanhada esta construção.

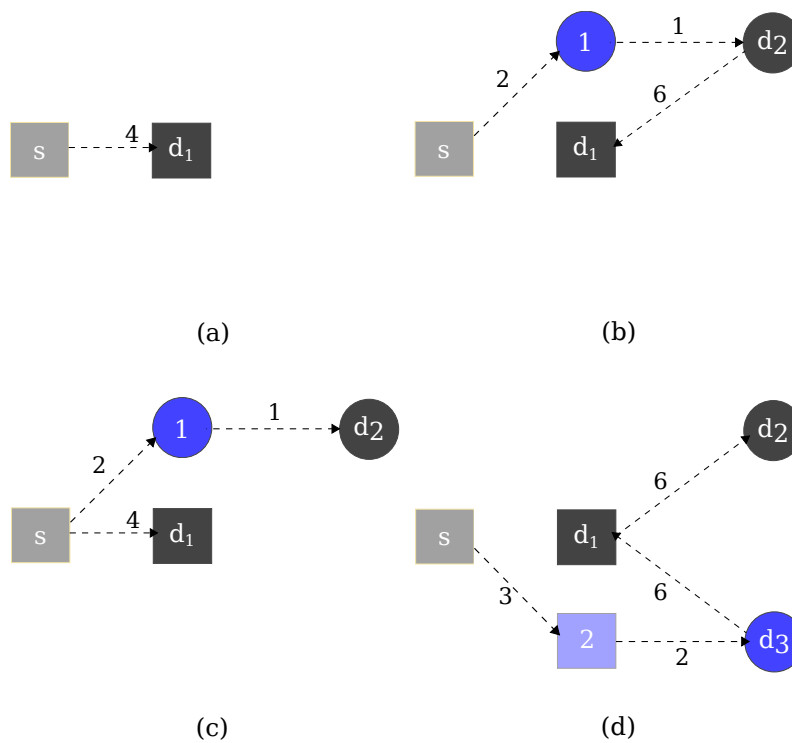


Figura 4.2: Vários passos da construção de duas árvores disjuntas nos arcos, para a rede da figura 4.1, com o novo método de protecção. Em (a) e (c) são representadas as árvores primárias parciais e em (b) e (d) as respectivas árvores secundárias.

A heurística MSH dita que os primeiros nós destino a ser ligados à árvore devem ser do tipo MC. Neste exemplo, o único nó destino MC é o nó d_1 e este será ligado à árvore através do único nó inicialmente pertencente a esta, o nó s . A árvore resultante é ilustrada em 4.2 (a). Após a adição do novo nó, são retirados da rede original os arcos que constituem a árvore primária parcial e é calculada a árvore disjunta nos arcos desta, representada na figura 4.2 (b). Note-se que o conjunto de nós destino da árvore secundária coincide com o conjunto de nós *correntemente* adicionados à árvore primária.

O cálculo de uma árvore secundária admissível dita a adição dos restantes destinos da sessão à árvore primária em construção, caso contrário esta adição é interrompida. Neste exemplo, como a árvore secundária é válida, continua-se o processo de ligação dos nós destino, sendo a rede restaurada para a sua constituição original. Os próximos nós a adicionar na árvore serão os nós destino MI. Após a adição de um novo nó à árvore (neste caso o nó d_2) e dos nós que pertenciam anteriormente à árvore (nó d_1), resultando na árvore representada em 4.2 (c), é calculada a árvore disjunta nos arcos da solução obtida, ilustrada em (d). Por último, falta adicionar o nó destino d_3 . Na figura 4.1 (b) é ilustrada a árvore primária obtida para a sessão Multicast pretendida e na figura 4.1 (c) a rede resultante da remoção dos arcos pertencentes à árvore primária calculada.

Como já foi referido anteriormente, na implementação usada para a construção de heurísticas, a remoção de determinados arcos da rede é feita através da adição de uma constante elevada (α) a cada um desses arcos. Desta forma, no cálculo de caminhos mais curtos entre dois nós, pode sempre existir a possibilidade de se usar estes arcos, embora tal signifique que a árvore obtida não é admissível, pois não é disjunta nos arcos com a árvore primária a proteger. É a partir desta observação que, no novo método, são identificados um ou mais arcos que impedem a obtenção de uma solução. A estes arcos, passamos a designar de arcos excluídos da árvore primária.

A heurística calcula todos os caminhos de custo mínimo que ligam os nós da árvore aos nós destino, pelo que se existir um ou vários caminhos cujo custo seja superior a α , é escolhido o caminho que apresenta maior custo e é identificado o primeiro arco deste caminho que faz parte da constituição da árvore primária. Este arco é identificado de forma a ser removido da rede original. Ao escolher o primeiro arco comum às duas árvores é, hipoteticamente, identificada a razão pela qual não é possível construir duas árvores disjuntas nos arcos e que torna a árvore secundária inadmissível. Assim, este arco é excluído do cálculo da árvore primária. Para que se possa fazer a distinção entre os arcos removidos para o cálculo de árvores secundárias e os arcos excluídos da árvore primária, é adicionada uma nova constante, designada de β , ao peso dos arcos excluídos cujo valor é igual a α multiplicado pelo número de arcos da rede¹. A razão pela qual é escolhido o caminho de custo mais elevado entre os caminhos inexistentes de menor custo prende-se com a tentativa de antecipar uma diminuição no custo da árvore primária a ser calculada. Após este procedimento, é reiniciado o processo do cálculo de duas árvores disjuntas nos arcos, onde é retirada a possibilidade da árvore primária utilizar o arco excluído devido ao seu elevado custo (valor igual a β somado com o seu peso). Este processo é repetido sempre que não seja possível obter uma árvore de protecção para uma

¹Esta distinção não é estritamente necessária. A sua aplicação teve em conta a execução do *debug* do método implementado.

4. Protecção de Sessões Multicast em Redes Sparse-Splitting

árvore primária em construção. Desta forma a árvore primária é calculada numa rede na qual foram sucessivamente removidos arcos que foram identificados como aqueles que, se presentes na árvore primária, impedem esta de ser protegida.

Retomando o exemplo anterior, em que se tinha finalizado a construção da árvore primária (representada na figura 4.1 (b)) com a adição do último nó destino, é verificado se existe, ou não, uma árvore de protecção para esta árvore. Seguindo o procedimento da heurística MSH-ADT na construção da árvore secundária (cujo primeiro passo é a remoção – modificação do custo – dos arcos da árvore primária), será calculado o menor caminho que liga um nó destino MC, neste caso d_1 , a um nó pertencente à árvore actual (o único nó existente inicialmente na árvore é o s). O resultado do cálculo do caminho mais curto será um valor superior a α e assim é detectada a impossibilidade de existência de uma árvore disjunta nos arcos da árvore primária criada. O maior, e neste caso único, caminho calculado é então o caminho que liga o nó s ao nó destino d_1 constituído apenas por um arco. Este arco é então o arco identificado para ser excluído – o seu custo é incrementado de β – durante o cálculo da árvore primária, cálculo este que será feito na nova rede representada na figura 4.3 (a). O processo é reiniciado e é construída a nova árvore primária (na rede sem os arcos excluídos) verificando sempre, após a adição de um novo nó, se existe para a árvore primária construída até então uma árvore secundária disjunta nos arcos com a anterior. As árvores primária e secundária calculadas, utilizando o novo método, são ilustradas, respectivamente, nas figuras 4.3 (b) e 4.3 (c).

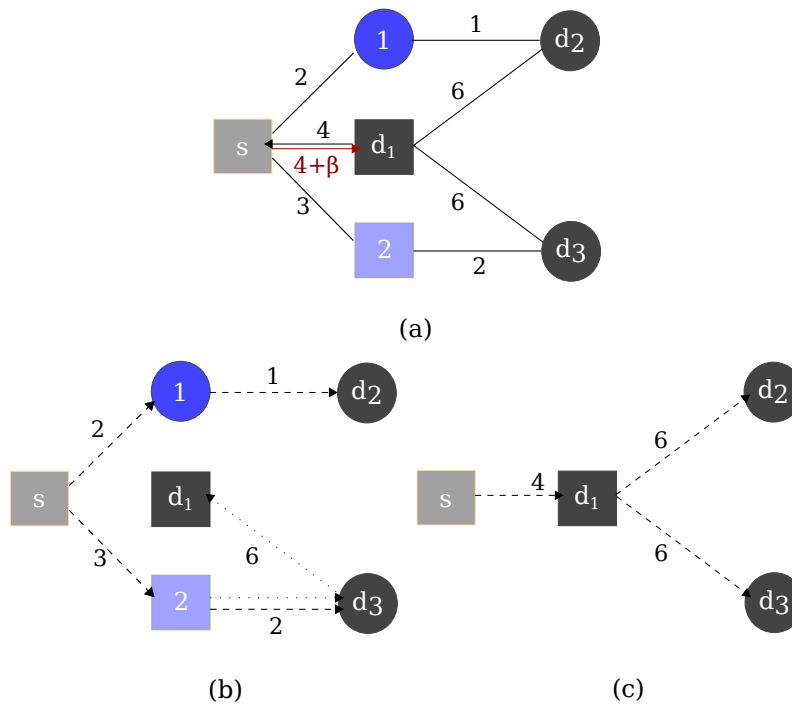


Figura 4.3: (a) Rede obtida através do novo método onde é calculada a árvore primária. Árvores (b) primária e (c) secundária, obtidas com o novo método.

Devido às características da rede ou a incapacidade do novo método, por vezes poderá não ser possível encontrar duas árvores disjuntas nos arcos que satisfaçam o pedido de uma determinada ligação Multicast. Nesta situação, o cálculo da árvore primária será interrompido assim que o novo método detectar que a heurística de cálculo da árvore primária requer a utilização de um ou mais arcos já excluídos, pelo que a solução obtida não é admissível. É então considerado que não existe uma árvore de protecção para qualquer árvore primária construída e, portanto, o pedido Multicast não é satisfeito.

O novo método de protecção, que será designado daqui em diante de *New Arc-Disjoint Trees* (NADT), foi implementado para as heurísticas MSH e MUS, mas pode ser usado em qualquer outra heurística com as devidas adaptações. O pseudo-código da técnica de protecção para uma heurística genérica H é exposto no algoritmo 6. Neste algoritmo, utiliza-se a notação $H_i(G, D, T)$ para representar uma iteração da heurística H na qual é adicionado um novo nó destino d_i à árvore T em construção (ou seja, T é parâmetro de entrada e de saída desta função).

Esta nova abordagem apenas requer uma nova versão de cada heurística que devolve a árvore em construção após a adição de cada nó destino. No caso da heurística MUS, esta requer a verificação, em primeiro lugar, da existência de uma árvore admissível após a adição de cada nó destino MC ligado à árvore primária a partir da heurística aplicada no cálculo da árvore de Steiner. De seguida, é realizado o mesmo processo mas desta vez para os destinos MI por adicionar.

4. Protecção de Sessões Multicast em Redes Sparse-Splitting

```
Input :  $G(V, E)$ ,  $N_{MC}$ ,  $r = (s, D)$  e função de custo  $c_a, a \in A$ ,  $H$  a heurística  
          escolhida  
Output: Light-tree primária  $T$  para  $r$  em  $G$ , light-tree secundária  $T_{prot}$  para  $r$  em  $G$   
1 begin  
2    $T = G'(\{s\}, \emptyset)$ ; // Inicia a criação da árvore  $T$   
3    $D_{prot} = \emptyset$ ;  
4    $A_{exc} = \emptyset$ ; // Arcos excluídos na construção da árvore primária  
5   foreach nó  $d_i \in D$  adicionado a  $T$  durante a execução de  $H_i(G, D, T)$  do  
6     Remover os arcos de  $A_{exc}$  da rede  $G$ ;  
7     if  $T$  é admissível then //  $T$  não contém nenhum arco excluído  
8       Remover os arcos de  $T$  da rede  $G$ ;  
9       Adicionar de novo os arcos de  $A_{exc}$  à rede  $G$ ;  
10       $D_{prot} = D_{prot} \cup \{d_i\}$ ; // Nós terminais na árvore actual  $T$   
11       $T_{prot} = H(G, D_{prot})$ ; // Árvore de protecção para a árvore  $T$   
12      if não é possível adicionar  $d, d \in D_{prot}$ , a  $T_{prot}$  then  
13        // Custo mínimo do caminho de  $x$  para  $y$   
14         $c_{xy} = \min_{i \in T_{prot_{MC}}, j \in D_{prot}} c_{ij}$ ;  
15        Identificar o primeiro arco  $a_{x'y'}$  de  $p_{xy}$  partilhado pelas duas árvores;  
16         $A_{exc} = A_{exc} \cup \{a_{x'y'}\}$ ;  
17         $T = G'(\{s\}, \emptyset)$ ; // Reinicia cálculo de  $T$   
18         $D_{prot} = \emptyset$ ; // Árvore actual sem nós terminais  
19      end  
20      Restaurar  $G$ , ou seja, reposição dos arcos de  $T$ ;  
21    else  
22      Restaurar  $G$ , ou seja, reposição dos arcos em  $A_{exc}$ ;  
23       $T_{prot} = \emptyset$ ; // É considerado que não existe protecção  
24       $T = H(G, D)$ ; // Árvore primária obtida na rede original  
25      É terminada a execução;  
26    end  
27  return  $T, T_{prot}$   
28 end
```

Algoritmo 6: H-NADT

5

Análise de Desempenho

Com o intuito de testar o desempenho das heurísticas criadas e realizar uma análise crítica do ponto de vista da eficiência em termos do custo médio das árvores Multicast obtidas e bloqueio de sessões Multicast, são apresentados neste capítulo os resultados experimentais obtidos através de simulações para as heurísticas implementadas em redes mistas *sparse-splitting* com *sparse wavelength conversion*. Em primeiro lugar, na secção 5.1, é descrito o ambiente experimental usado para as simulações. Na secções 5.2 e 5.3 são apresentados, respectivamente, os resultados relativos às heurísticas de encaminhamento Multicast e à protecção de sessões Multicast.

5.1 Descrição do Ambiente Experimental

A realização dos testes para os algoritmos desenvolvidos foi feita criando redes mistas, a partir de redes não dirigidas, de forma a que estas apresentassem características semelhantes às das redes utilizadas no artigo [11]. As redes obtidas contêm então as seguintes propriedades:

1. Uma rede “rnd40”, criada aleatoriamente¹, com 40 nós e 200 arestas e com custo dos arcos aleatório entre 1 e 200.
2. Uma rede “rede50”, com 50 nós e com 177 arestas cujo custo varia entre 1 e 100,

¹Gerador de redes gentilmente cedido pelo Professor José Luís Santos, Departamento de Matemática da Universidade de Coimbra

5. Análise de Desempenho

criada com o modelo Doar-Leslie [15] usando o software *GT-ITM Georgia Tech Internetwork Topology Models* (site do software.).

3. Quatro redes “rede40”, designadas de “rede40-0”, “rede40-1”, “rede40-2” e “rede40-3”, com 40 nós e com um número de ligações variando entre 194 e 226 com custo entre 1 e 100, criadas igualmente com o modelo anterior.

A obtenção de redes mistas foi conseguida através da remoção de um arco em 50% das ligações de cada uma das redes criadas. Para assegurar a possível protecção de todas as ligações aos nós constituintes da rede, os arcos candidatos a serem removidos foram aqueles que emergiam ou incidiam em nós cujo semigrau de entrada e de saída era maior que dois. A modelação da rede como uma rede *sparse-splitting*, para cada uma das redes, é conseguida através da escolha de alguns nós da rede para se tornarem nós MC. Esta selecção foi efectuada através da escolha dos nós que apresentavam maior grau. Na simulação de cada heurística, todos os nós da rede foram usados como nó fonte. Para cada nó fonte, o número de destinos $|D|$ varia entre 2 e 20 e foram criadas 500 sessões Multicast para cada $|D|$ (nunca repetindo a mesma sessão), nas redes de 40 nós, e 400 sessões nas redes de 50 nós, perfazendo o mesmo total de 380000 execuções por rede. Os resultados obtidos são expressos por tamanho $|D|$ da sessão Multicast, independentemente do nó fonte. O estudo das heurísticas de encaminhamento Multicast é feito através da obtenção do custo médio das árvores Multicast (ver secção 5.2). Para avaliar o desempenho dos algoritmos com protecção de sessões Multicast, foi calculado o custo médio das árvores primária e secundária obtidas (apenas quando a árvore de protecção existe) e a taxa de bloqueio (ver secção 5.3). Neste contexto, a taxa de bloqueio é definida como a razão entre o número de sessões Multicast não estabelecidas (devido à inexistência de uma árvore secundária disjunta nos arcos da árvore primária) e o número total de sessões Multicast. Os resultados finais foram obtidos executando 10 simulações para cada rede (escolhendo sementes diferentes para o gerador de números aleatórios utilizado para criar as sessões). Após o cálculo do custo médio das árvores e da taxa de bloqueio para cada simulação, são calculadas a média do custo médio das árvores e a média da taxa de bloqueio das 10 simulações. O desvio padrão é obtido usando as 10 amostras e cada uma das suas médias.

5.2 Heurísticas de Encaminhamento Multicast

Com o intuito de manter o texto auto-contido, são apenas apresentados neste capítulo os resultados para as redes “rede40-3” e “rnd40” que ilustram os resultados padrão das heurísticas. Os restantes resultados podem ser encontrados no apêndice B. É de notar

5.2 Heurísticas de Encaminhamento Multicast

que, para as redes testadas, as heurísticas encontram sempre uma árvore Multicast para qualquer sessão criada e o desvio padrão dos resultados é muito pequeno.

Na figura 5.1 são apresentados os resultados das simulações com as várias heurísticas implementadas, sendo estes resultados expressos no custo médio da árvore Multicast calculada por número de destinos. Nas simulações efectuadas, foram escolhidos 5 nós da rede para serem do tipo MC (tal como em [11]). Da análise dos resultados, em primeiro lugar, é possível confirmar, pela figura 5.1 (a), que a heurística MSH apresenta um menor custo médio, em termos de árvores Multicast calculadas, em relação à heurística MUS. Na figura 5.1 (b), é confirmada a superioridade, apresentada em [10], da heurística SSMRH em relação à heurística MUS, apresentando uma diferença relativa (neste cálculo, o denominador é o maior dos dois valores) de 6.3% para $|D| = 20$.

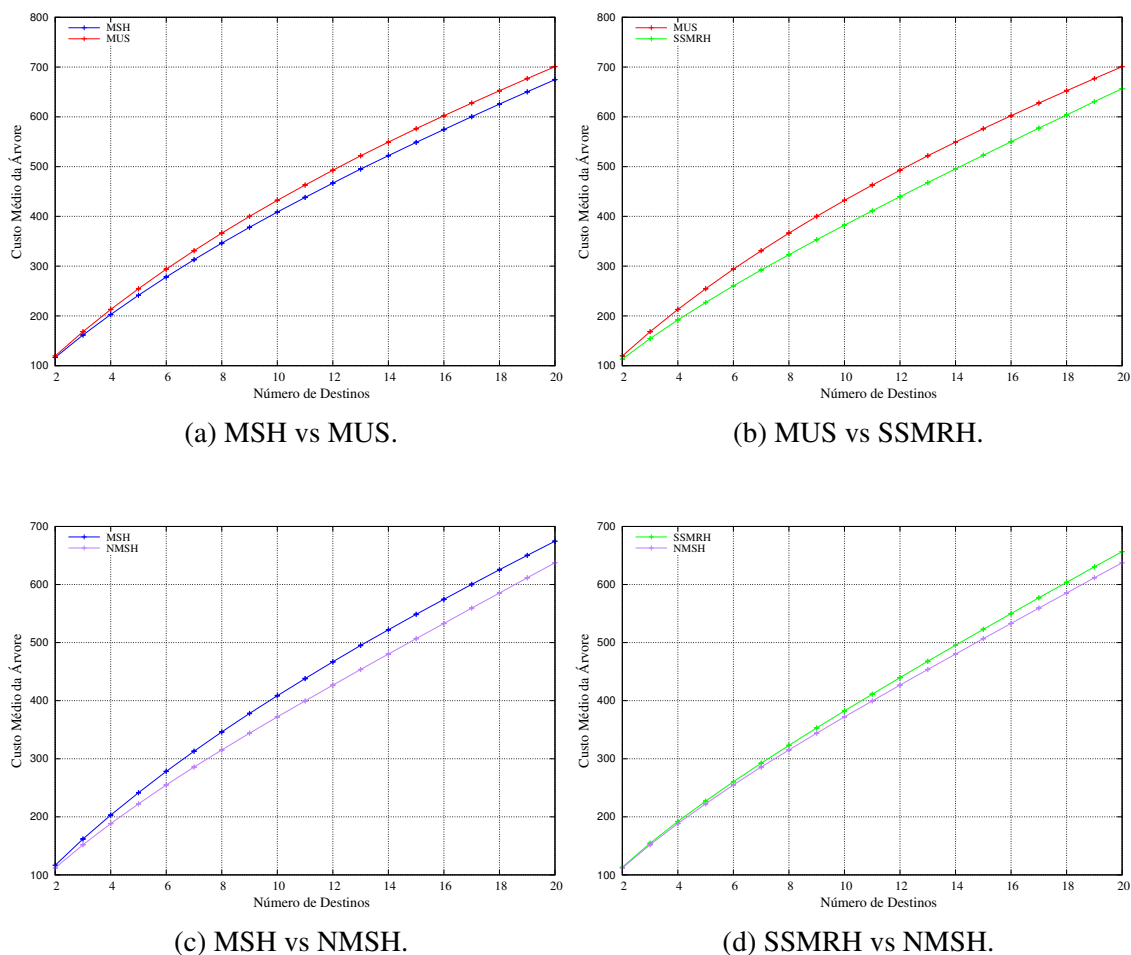


Figura 5.1: Custo médio da árvore por número de destinos para a rede “rede40-3” com 5 nós MC.

A comparação entre a nova heurística NMSH e a heurística MSH é feita na figura 5.1 (c). Como é possível observar, a heurística NMSH apresenta uma diminuição do custo médio das árvores Multicast calculadas, apresentando uma diferença relativa de cerca de

5. Análise de Desempenho

5.5% para $|D| = 20$. A inclusão dos nós MC no grupo de destinos da sessão Multicast, leva a heurística NMSH a calcular árvores Multicast, na pior das hipóteses, com custo igual ao das árvores calculadas com a heurística MSH, o que justifica a melhoria apresentada pelas simulações executadas. Na figura 5.1 (d) é comparada a eficiência das duas heurísticas, SSMRH e NMSH, que são construídas usando de forma repetitiva, respectivamente, as heurísticas MUS e MSH. Anteriormente, foi visto que a heurística MSH apresenta melhores resultados que a MUS. O resultado das simulações para as versões repetitivas destas heurísticas apresenta igualmente uma melhoria da nova heurística NMSH em relação à SSMRH, sendo a diferença relativa entre as duas heurísticas de cerca de 2.9% para $|D| = 20$. Deste modo, conclui-se que a forma iterativa de usar uma heurística criada para calcular uma árvore Multicast de custo mínimo parece ser mais eficaz quanto melhor for o desempenho da heurística base. A análise do gráfico da figura 5.1 (d) mostra que a heurística NMSH pode tornar-se ainda mais eficaz com o aumento do número de destinos da sessão Multicast.

O número de nós do tipo MC presentes nas redes também tem influência no custo médio das árvores Multicast calculadas. De modo a estudar o desempenho dos algoritmos em função desta característica das redes, foram feitas simulações para obtenção do custo médio das árvores Multicast, para a rede “rnd40”, usando as diferentes heurísticas implementadas. Os resultados são ilustrados na figura 5.2. A primeira conclusão que pode ser tirada da análise dos gráficos da figura 5.2 é que quanto maior for o número de nós MC na rede, menor será o custo médio de uma árvore Multicast calculada, independentemente da heurística usada para calcular a árvore. Este facto é acentuado com o aumento do número de nós destino presentes na sessão Multicast. Em termos da eficiência das heurísticas, é visível que, mais uma vez, a heurística NMSH leva a melhor a qualquer uma das outras heurísticas no que concerne à redução do custo médio das árvores calculadas. Esta redução, em relação às outras heurísticas, é reforçada para uma rede com uma baixa percentagem de nós MC, sendo a heurística mais eficaz em identificar a forma como os nós MC podem ser incluídos na árvore de maneira a reduzir o custo desta, como pode ser visto na figura 5.2 (a). Outro aspecto interessante, e que pode ser visto nas figuras 5.2 (b) e 5.2 (c), é que com o aumento de nós MC na rede e do número de nós destino nas sessões Multicast o custo médio das árvores calculadas com a heurística MSH tende a igualar (e até a reduzir, na figura 5.2 (c)) o custo médio das árvores calculadas com a heurística SSMRH. Uma vez que, nestes casos, a diferença entre as heurísticas MSH e NMSH é pequena, conclui-se que o MSH é uma boa opção para o cálculo de uma árvores Multicast para redes com uma grande percentagem de nós MC, ou para sessões com um elevado número de destinos, se for desejado evitar a utilização das heurísticas SSMRH e NMSH.

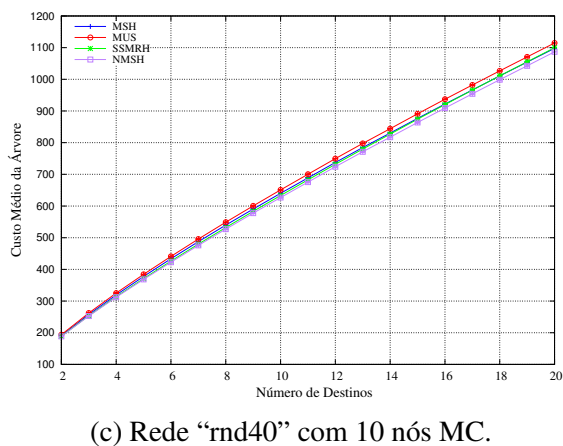
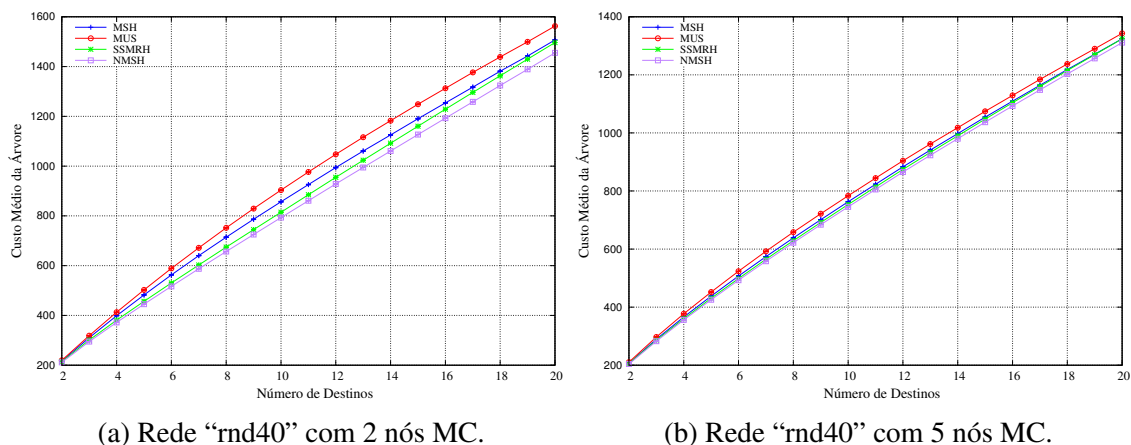


Figura 5.2: Custo médio da árvore por número de destinos para a rede "rnd40".

5.3 Protecção de Sessões Multicast

As simulações realizadas para avaliar o desempenho das heurísticas implementadas com os respectivos algoritmos de protecção foram feitas fixando o número de nós MC das redes em 5. Mais uma vez, para manter o texto auto-contido, serão apenas apresentados os resultados para a rede "rede40-2", encontrando-se os restantes resultados no apêndice B. Em geral, para todas as redes testadas, o desvio padrão apresenta valores pequenos o que sugere que não existe uma grande diferença entre a média de cada uma das amostras.

Na figura 5.3 (a) é apresentado o custo médio das árvores disjuntas nos arcos por número de destinos e na figura 5.3 (b) a taxa de bloqueio médio, também por número de destinos, resultante das simulações efectuadas com as heurísticas MUS-ADT, MSH-ADT, SSMRH-ADT e NMSH-ADT. Os resultados obtidos pela MSH-ADT (para todas as redes) foram confirmados pelos criadores desta heurística após envio de um email contendo em anexo os gráficos obtidos experimentalmente, as redes utilizadas e a descrição

5. Análise de Desempenho

do ambiente experimental. Os resultados obtidos pelas simulações efectuadas são semelhantes aos resultados ilustrados no artigo [11], onde os autores apresentam a heurística MSH-ADT. Para a rede “rede40-2” em particular, as maiores diferenças entre os resultados obtidos e os apresentados no artigo localizam-se no custo médio das árvores, explicado pelo facto da rede “rede40-2” ter arcos com peso entre 1 e 100 (enquanto que no artigo os pesos dos arcos da rede variam entre 1 e 200), e na taxa de bloqueio para os casos em que $|D| = 2$ e $|D| = 3$ (que os autores garantem ser igual a 0). Esta discrepância pode ser justificada pelo facto da rede aqui utilizada, possivelmente, conter um maior número de nós com semigrau de saída igual a 2, o que pode levar à ocupação dos dois arcos no cálculo de uma árvore primária e impedir, desde logo, a existência de uma árvore secundária disjunta nos arcos à árvore calculada.

A implementação da protecção ADT nas heurísticas SSMRH e NMSH foi executada de modo a estudar a variação da taxa de bloqueio através do uso destas heurísticas de encaminhamento no cálculo de duas árvores disjuntas nos arcos. Os resultados obtidos podem ser vistos na figura 5.3.

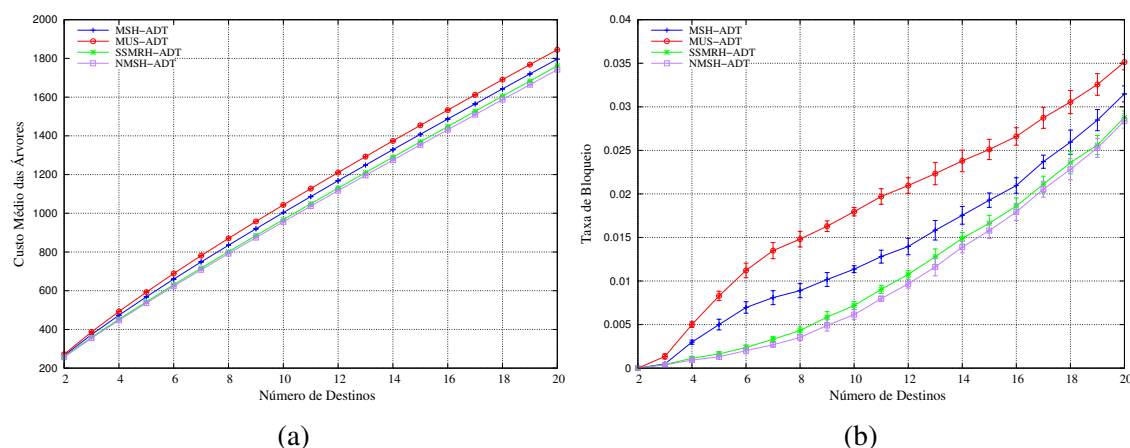


Figura 5.3: (a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2” com 5 nós MC.

Pela análise da figura 5.3 (b), é visível que as heurísticas SSMRH-ADT e NMSH-ADT apresentam melhores resultados, quer em termos de custo médio das árvores Multicast quer em termos da taxa de bloqueio, que as heurísticas MUS-ADT e MSH-ADT. A diminuição da taxa de bloqueio, atingida pelas heurísticas NMSH-ADT e SSMRH-ADT, pode ser explicada pela utilização de menos arcos no cálculo da árvore primária visto que são introduzidos mais nós MC durante a construção desta, o que leva a uma maior número de possibilidades no cálculo de uma árvore secundária disjunta nos arcos da árvore primária. Em termos globais, a melhor heurística é a NMSH-ADT que apresenta

a maior diminuição nos dois campos avaliados (seguido de muito próximo pela heurística SSMRH-ADT), embora nas simulações da rede “rede40-1” (figura B.6 do apêndice B) a taxa de bloqueio da heurística SSMRH-ADT seja a menor das duas. Analisando os restantes resultados, esta situação aparenta ser um caso pontual.

Os resultados da introdução do novo método de protecção NADT nas heurísticas MSH e MUS podem ser vistos na figura 5.4, onde estes são comparados com as heurísticas MSH-ADT e MUS-ADT. Desde logo, é possível analisar, a partir da figura 5.4 (a), que o custo médio das árvores calculadas pelas heurísticas com a protecção NADT é semelhante ao custo médio das árvores calculadas com a protecção ADT, sendo um pouco superior (embora não seja totalmente visível). Obviamente, este facto advém de ser calculado um número superior de árvores (neste caso, todas as desejadas) ao utilizar a protecção NADT.

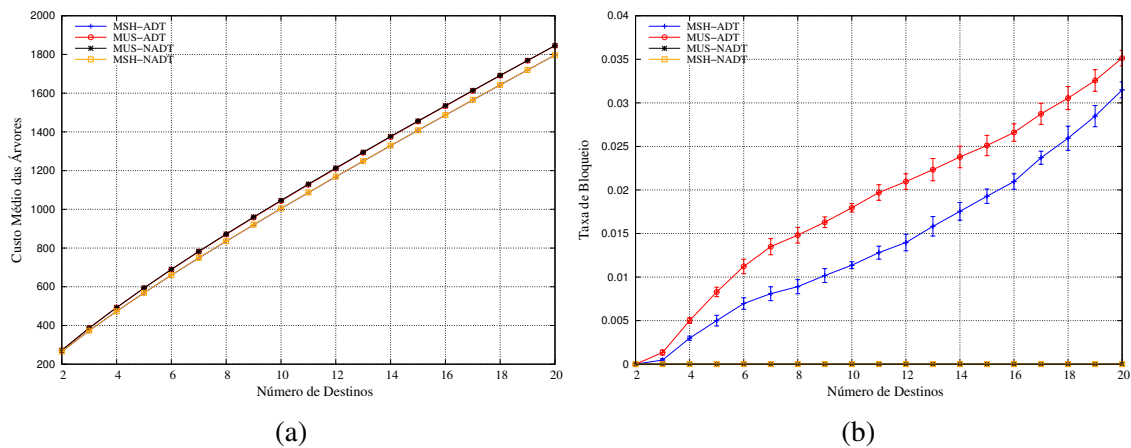


Figura 5.4: (a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2” com 5 nós MC.

Em termos do objectivo principal da criação deste novo método de protecção, a diminuição da taxa de bloqueio, este é conseguido de forma plenamente satisfatória visto que é possível encontrar sempre, para esta rede, duas árvores disjuntas nos arcos sem um aumento significativo do custo médio das árvores calculadas. A explicação para a taxa de bloqueio média ser nula advém do facto o semigrau de cada nó da rede ser maior ou igual que 2 e do novo método de protecção poder excluir, se necessário, pelo menos um destes arcos no cálculo da árvore primária, garantindo assim a possibilidade de existência de uma árvore de protecção. A heurística MSH-NADT é então aquela que apresenta melhores resultados visto que, embora a taxa de bloqueio seja igual à alcançada pela heurística MUS-NADT, apresenta menor custo médio das árvores Multicast calculadas.

Na figura 5.5 é feita a comparação entre as heurísticas que atingem os melhores resultados em termos da taxa de bloqueio média para uma sessão Multicast constituída por $|D|$

5. Análise de Desempenho

destinos. As heurísticas SSMRH-ADT e NMSH-ADT reduzem, como já visto, o custo médio das árvores calculadas, mas não conseguem acompanhar a significativa diminuição da taxa de bloqueio apresentada pelas heurísticas MSH-NADT e MUS-NADT. É concluído então que a protecção NADT é superior à ADT quando pretendido baixar a taxa de bloqueio de uma sessão Multicast ao criar duas árvores disjuntas nos arcos.

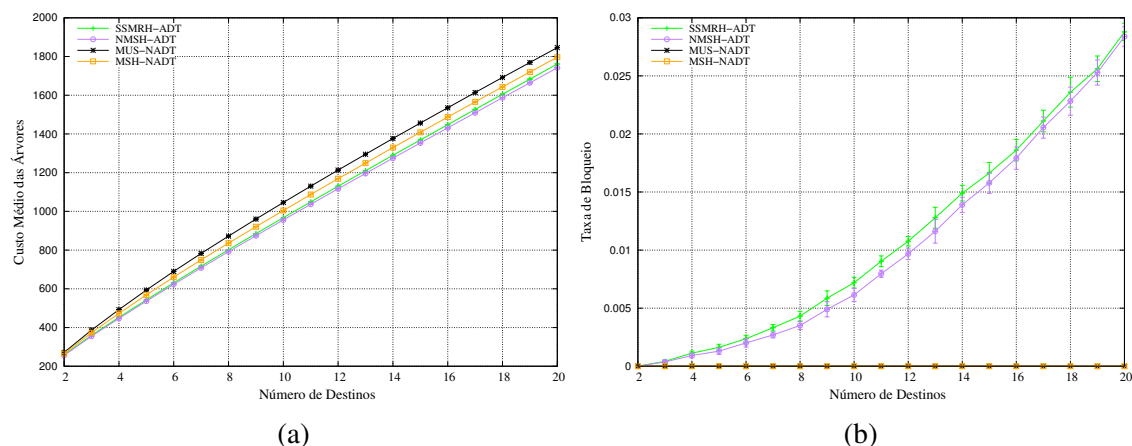


Figura 5.5: (a) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) taxa de bloqueio por número de destinos para a “rede40-2”.

A introdução da protecção NADT nas heurísticas SSMRH e NMSH poderia ter sido realizada e, teoricamente, teria sido atingida uma taxa de bloqueio muito pequena (ou mesmo nula) para o tipo de redes usadas, sem que fosse expectável qualquer degradação no custo médio das árvores Multicast calculadas em relação às heurísticas MSH-NADT e MUS-NADT. A opção de não introduzir a nova protecção nestas heurísticas recai no facto de estas já serem computacionalmente pesadas devido à execução repetitiva de uma heurística base no cálculo de soluções. Esta repetição aliada à protecção NADT tornaria a heurística extremamente pesada e, logo, com vantagem diminuta em relação às heurísticas criadas.

6

Conclusões e Trabalho Futuro

A protecção de sessões Multicast em redes ópticas WDM é um tema fundamental na garantia da sobrevivência de um pedido Multicast impedindo perdas de informação em casos de falhas isoladas nos arcos, ou linhas, de uma rede. Este foi o tema escolhido para o desenvolvimento desta dissertação, onde as ligações das redes estudadas continham algumas restrições, que foram apresentadas durante a descrição das redes ópticas WDM, introduzidas pela comunicação Multicast. Estas restrições traduziam-se em redes onde apenas alguns nós têm a capacidade de replicar o sinal (e os restantes nós são nós de passagem ou nós folha) e onde a capacidade de conversão de comprimento de onda está limitada a esses mesmos nós.

Foram apresentadas diversas heurísticas existentes de encaminhamento Multicast em redes *sparse-splitting*. Três heurísticas existentes foram implementadas e foi proposta uma nova abordagem ao cálculo de uma árvore que solucione uma sessão Multicast através do uso da heurística MSH, apresentada em [11], criada para redes mistas, a heurística NMSH. Através da repetida invocação da heurística MSH, variando o conjunto de destinos, esta heurística obtém árvores Multicast cujo custo é menor ou igual que o custo das árvores calculadas com a sua heurística base, como pode ser verificado nos resultados apresentados.

Nos esquemas de protecção existentes o cálculo da árvore de protecção é feito após a árvore primária estar completamente calculada, o que muitas vezes leva a que não seja possível obter uma árvore de protecção. De modo a reduzir a taxa de bloqueio no cálculo de um par de árvores e, assim, atingir o objectivo inicial desta dissertação, foi proposta

6. Conclusões e Trabalho Futuro

uma nova abordagem ao cálculo de uma árvore de protecção para sessões Multicast, mantendo a obrigatoriedade de esta ser disjunta nos arcos da árvore primária. Nesta abordagem, verifica-se, sequencialmente, se as árvores primárias obtidas após a adição de um nó destino podem ser protegidas. Se não é possível criar uma árvore disjunta nos arcos a uma destas árvores calculadas, o algoritmo tenta localizar o arco a excluir no cálculo de uma árvore primária e o processo é reiniciado. Este algoritmo de protecção foi designado de NADT e os resultados da sua aplicação nas heurísticas implementadas mostram que este novo meio de protecção pode realmente diminuir significativamente a taxa de bloqueio no cálculo de um par de árvores disjuntas nos arcos em relação à protecção ADT.

6.1 Contribuições da Dissertação

A principal contribuição desta dissertação foi a construção de um novo esquema de protecção que aplica uma nova abordagem ao cálculo de um par de árvores disjuntas nos arcos. Uma outra contribuição foi a criação e implementação de uma nova versão de uma heurística que diminui o custo médio do cálculo de uma árvore Multicast em relação a outras heurísticas existentes.

6.2 Trabalho Futuro

Na sequência do trabalho realizado, propõe-se, em primeiro lugar, a adaptação e estudo do comportamento das heurísticas implementadas em redes onde os nós MI são do tipo DaC, visto ser este tipo de capacidade que possivelmente estará mais presente nos nós das redes ópticas do futuro.

Em segundo lugar, seria interessante verificar se a técnica de protecção proposta, que condiciona e molda a construção de uma árvore primária à existência de uma árvore secundária, pode ser aplicada com sucesso noutros contextos, como por exemplo em redes não dirigidas. Mais interessante poderá ser a incorporação desta nova técnica de protecção em algoritmos de encaminhamento em redes onde os nós MI são do tipo DaC.

O problema do encaminhamento Multicast em redes ópticas WDM abordado nesta dissertação, para o cálculo de um par de árvores disjuntas nos arcos, teve sempre em conta a minimização do custo das soluções encontradas. Assim, finalmente, sugere-se a hipótese de adaptar estas heurísticas de forma a minimizar o número de comprimentos de onda necessários no estabelecimento de uma árvore Multicast.



Arquitectura dos Nós de Redes Ópticas WDM

Em [16] a arquitectura proposta designa-se *Splitter-and-Delivery (SaD) Cross-Connect* (figura A.1 (a)), onde um SaD-OXC de $N \times N$ consiste num conjunto de W comutadores SaD (figura A.1 (b)), um para cada comprimento de onda. O funcionamento de um comutador SaD consiste na criação de um número (igual ao grau do nó) de cópias do sinal original e comutar os sinais para a saída apropriada usando comutadores de 2×1 . Este nó providencia um serviço não-bloqueante, logo poderá servir várias sessões Multicast no mesmo comprimento de onda. Apesar disso, a arquitectura do nó não distingue entre diferentes tipos de tráfego, podendo existir perdas de potência desnecessárias no caso do tráfego ser Unicast.

Em [4] é apresentada o *Multicast-Only and Splitter-and-Delivery (MOSaD) (A.2)*, que é uma actualização da arquitectura SaD. Nesta arquitectura, apenas são usados *optical-splitters* para ligações Multicast, sendo que as ligações Unicast não são afectadas por este processo e assim não estão expostas às perdas de potência. Este tipo de arquitectura providencia um serviço não-bloqueante apenas para ligações Unicast. Como desvantagem apenas uma sessão Multicast em cada canal pode ser realizada ao mesmo tempo.

A. Arquitectura dos Nós de Redes Ópticas WDM

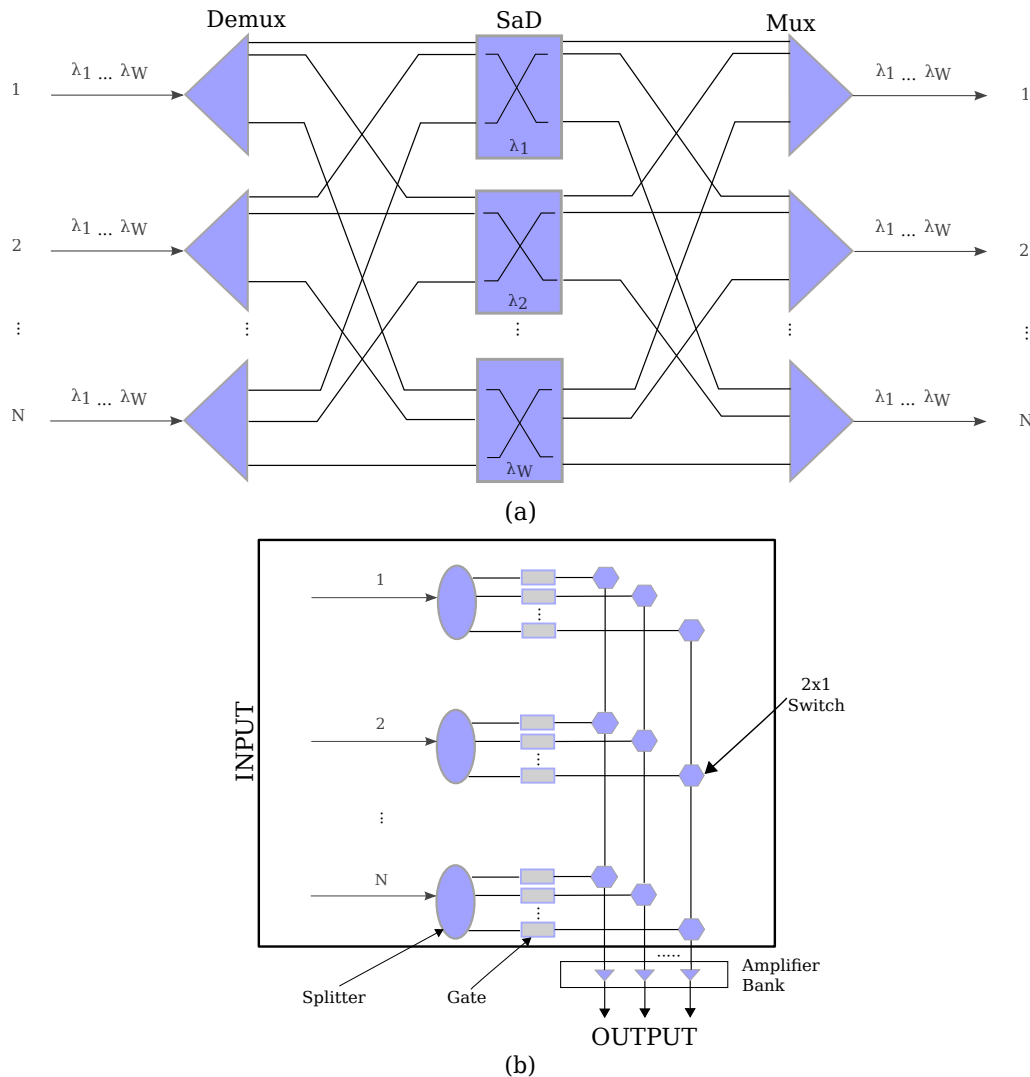


Figura A.1: Arquitectura de um (a) SaD-OXC $N \times N$ baseado em (b) comutadores SaD. Imagem adaptada de [4].

Existem também arquitecturas que introduzem capacidade de conversão de comprimento de onda nos nós MC [18, 24]. Na arquitectura proposta em [24], os sinais de entrada são desmultiplexados e passam pelo primeiro comutador óptico, sendo apenas os sinais Multicast comutados para um *optical-splitter*. De seguida, as cópias do sinal atravessam um segundo comutador que encaminha os sinais para os portos de saída apropriados. Em [18] é introduzida uma nova arquitectura baseada em partilha de *optical-splitters* e conversores de comprimento de onda por todos os sinais de entrada. Assim, algumas sessões poderão ser bloqueadas dependendo da disponibilidade destes recursos. Os resultados desta arquitectura, indicam que equipando 50% dos nós com *optical-splitters* e 15% dos nós com conversores de comprimento de onda é possível melhorar o desempenho do sistema, aproximando a probabilidade de bloqueio de uma sessão Multicast para o caso ideal.

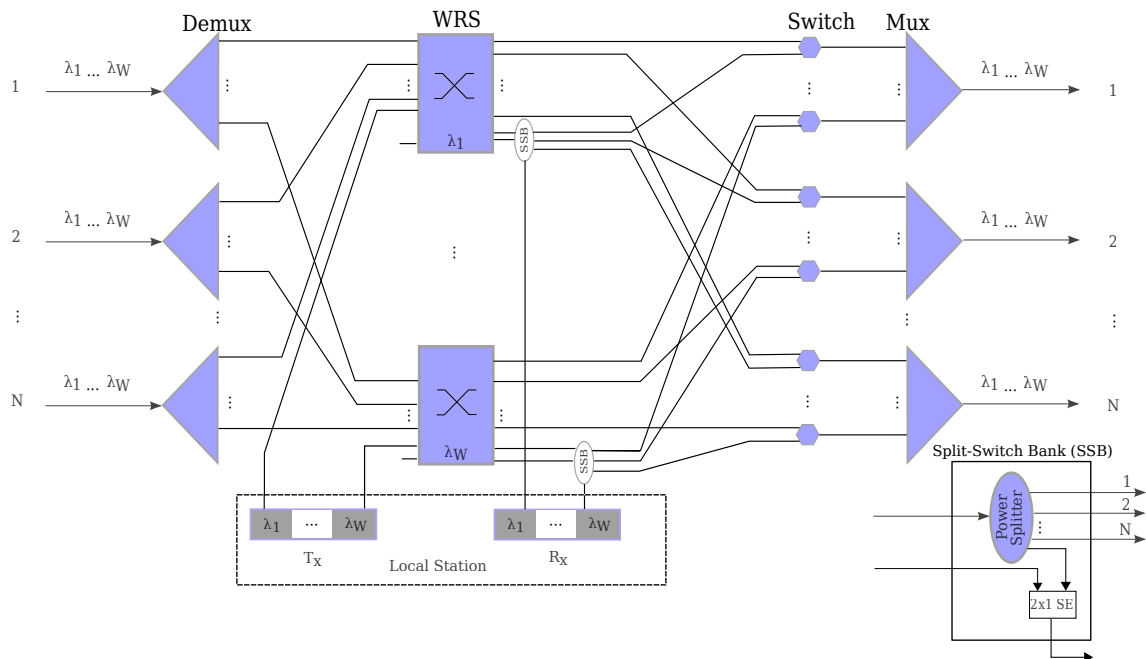


Figura A.2: Arquitectura de um MOSaD-OXC $N \times N$. Imagem adaptada de [4].

Os nós MI DaC foram propostos em [3] para diminuir o impacto do custo dos nós MC nas *light-trees*. A sua estrutura é semelhante à do MOSaD-OXC, porém o módulo SSB é trocado por um novo módulo, *Tap-and-Continue Module (TCM)* representado na figura A.3. Nos TCMs, uma fracção pequena do sinal é libertada no nó local, e o restante sinal (com potência na ordem dos 99,9%) é comutado para uma outra das $N - 1$ saídas.

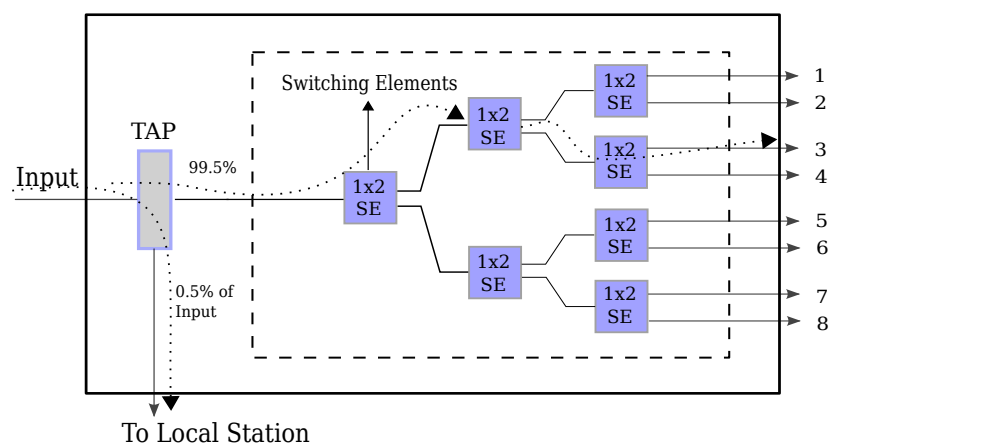


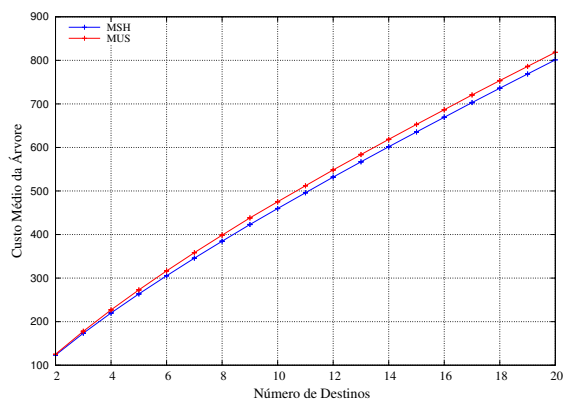
Figura A.3: Exemplo de um módulo TCM 1×8 . Imagem adaptada de [3].

B

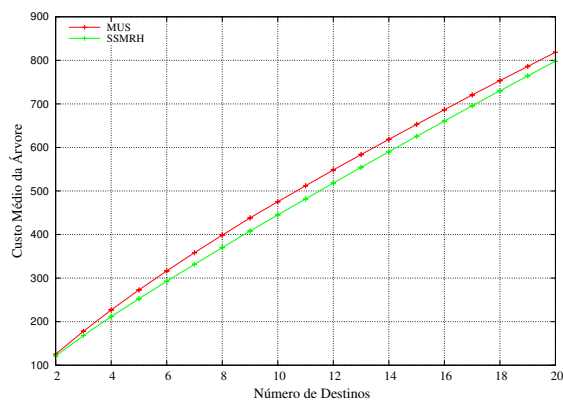
Resultados Obtidos

São apresentados neste apêndice os resultados obtidos para as heurísticas implementadas e para os métodos de protecção desenvolvidos. Para as heurísticas implementadas é avaliado o custo médio das árvores Multicast calculadas por número de destinos. Os resultados dos métodos de protecção são expressos pelo custo médio das duas árvores Multicast calculadas (árvore primária e secundária) e taxa de bloqueio por número de destinos.

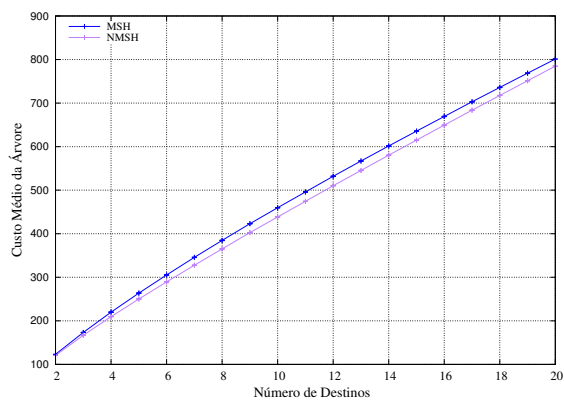
B. Resultados Obtidos



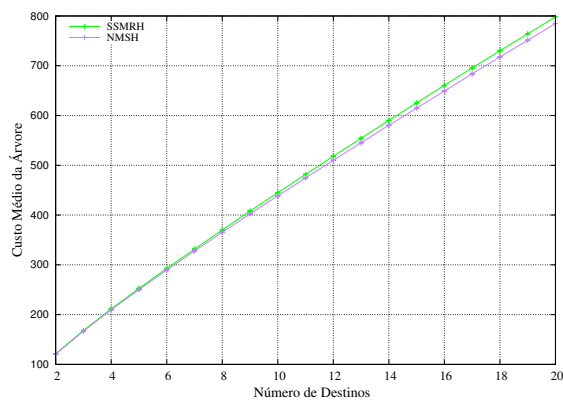
(a) MSH vs MUS.



(b) MUS vs SSMRH.

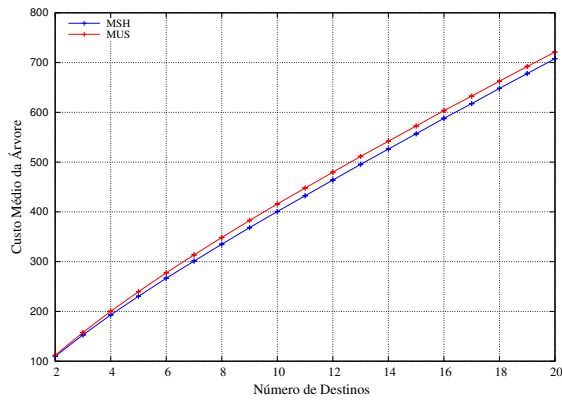


(c) MSH vs NMSH.

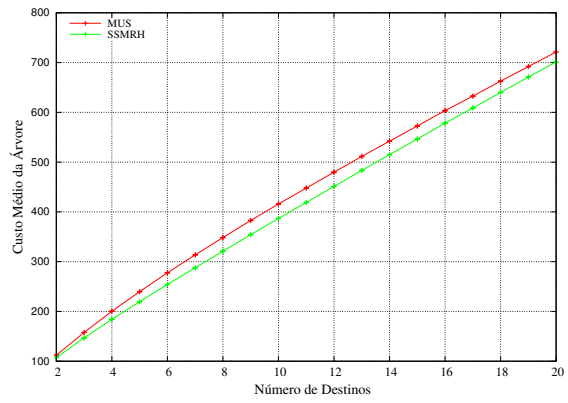


(d) SSMRH vs NMSH.

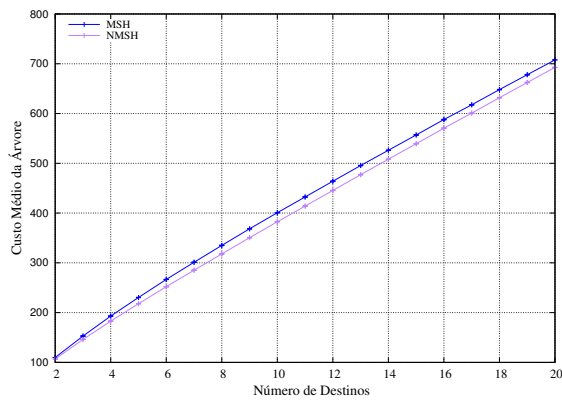
Figura B.1: Custo médio da árvore por número de destinos para a rede “rede40-0” com 5 nós MC.



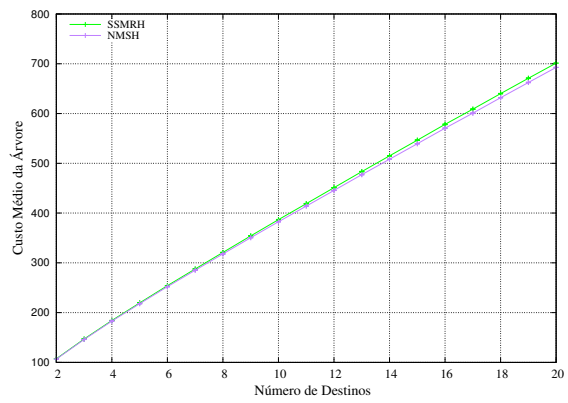
(a) MSH vs MUS.



(b) MUS vs SSMRH.



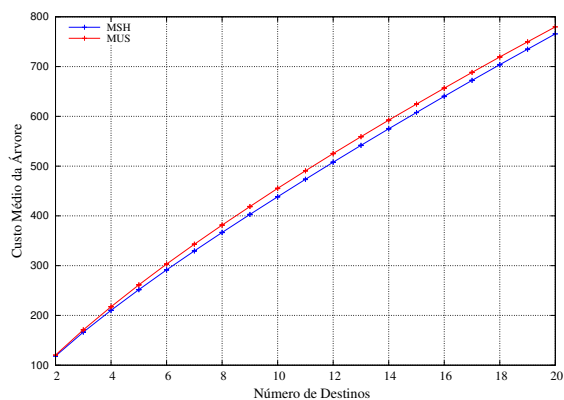
(c) MSH vs NMSH.



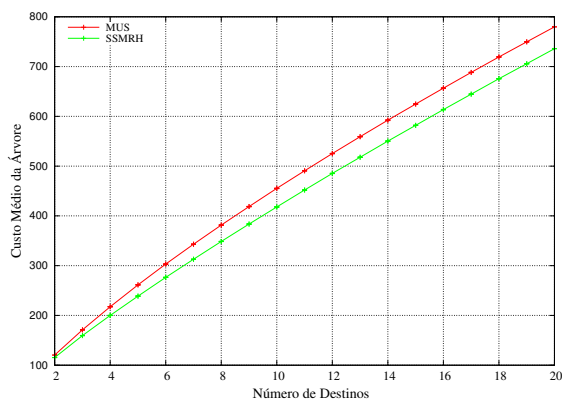
(d) SSMRH vs NMSH.

Figura B.2: Custo médio da árvore por número de destinos para a rede “rede40-1” com 5 nós MC.

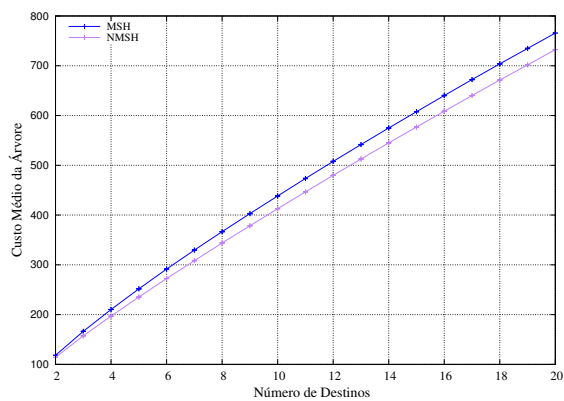
B. Resultados Obtidos



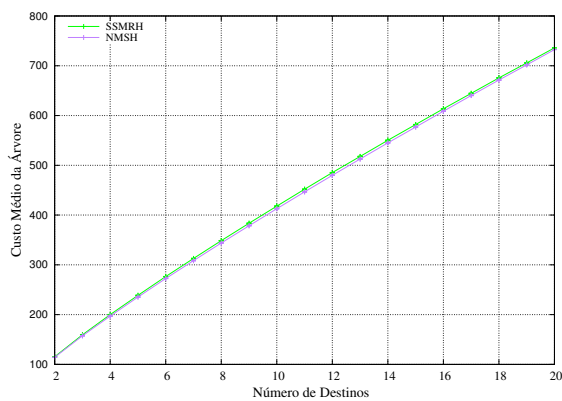
(a) MSH vs MUS.



(b) MUS vs SSMRH.

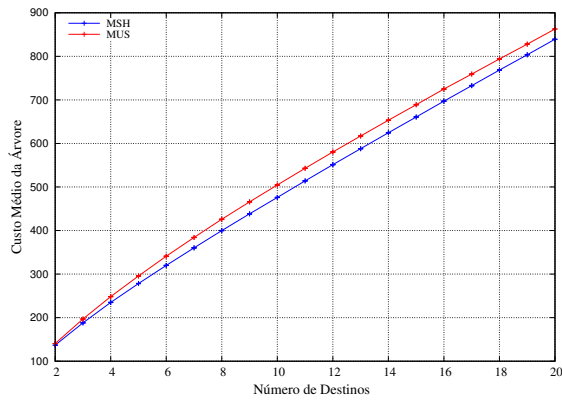


(c) MSH vs NMSH.

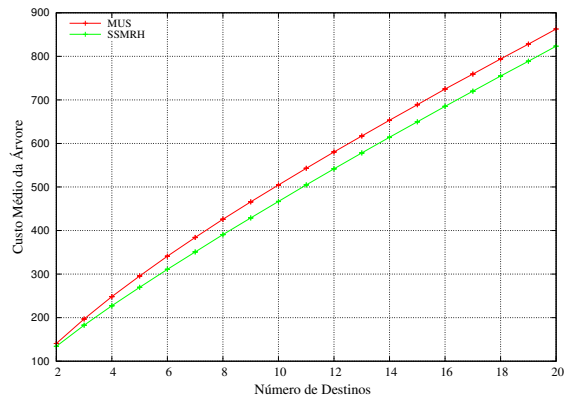


(d) SSMRH vs NMSH.

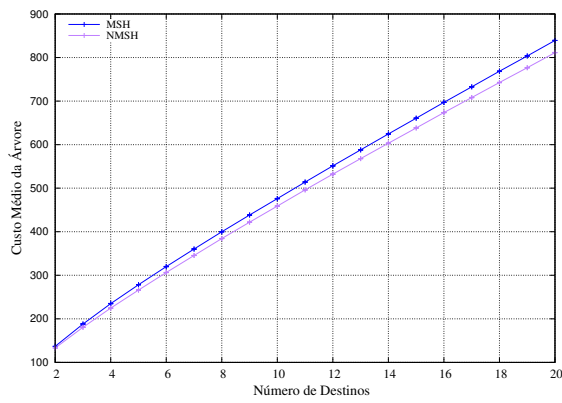
Figura B.3: Custo médio da árvore por número de destinos para a rede “rede40-2” com 5 nós MC.



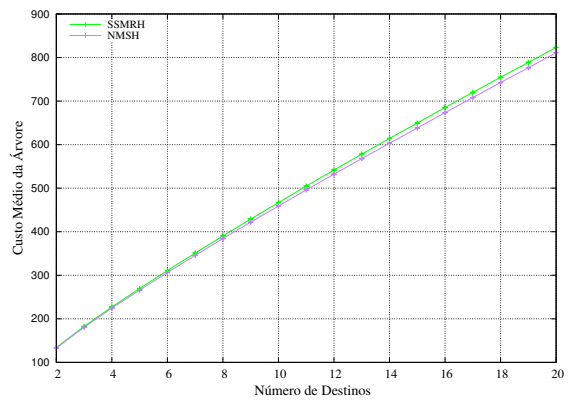
(a) MSH vs MUS.



(b) MUS vs SSMRH.



(c) MSH vs NMSH.



(d) SSMRH vs NMSH.

Figura B.4: Custo médio da árvore por número de destinos para a rede “rede50” com 5 nós MC.

B. Resultados Obtidos

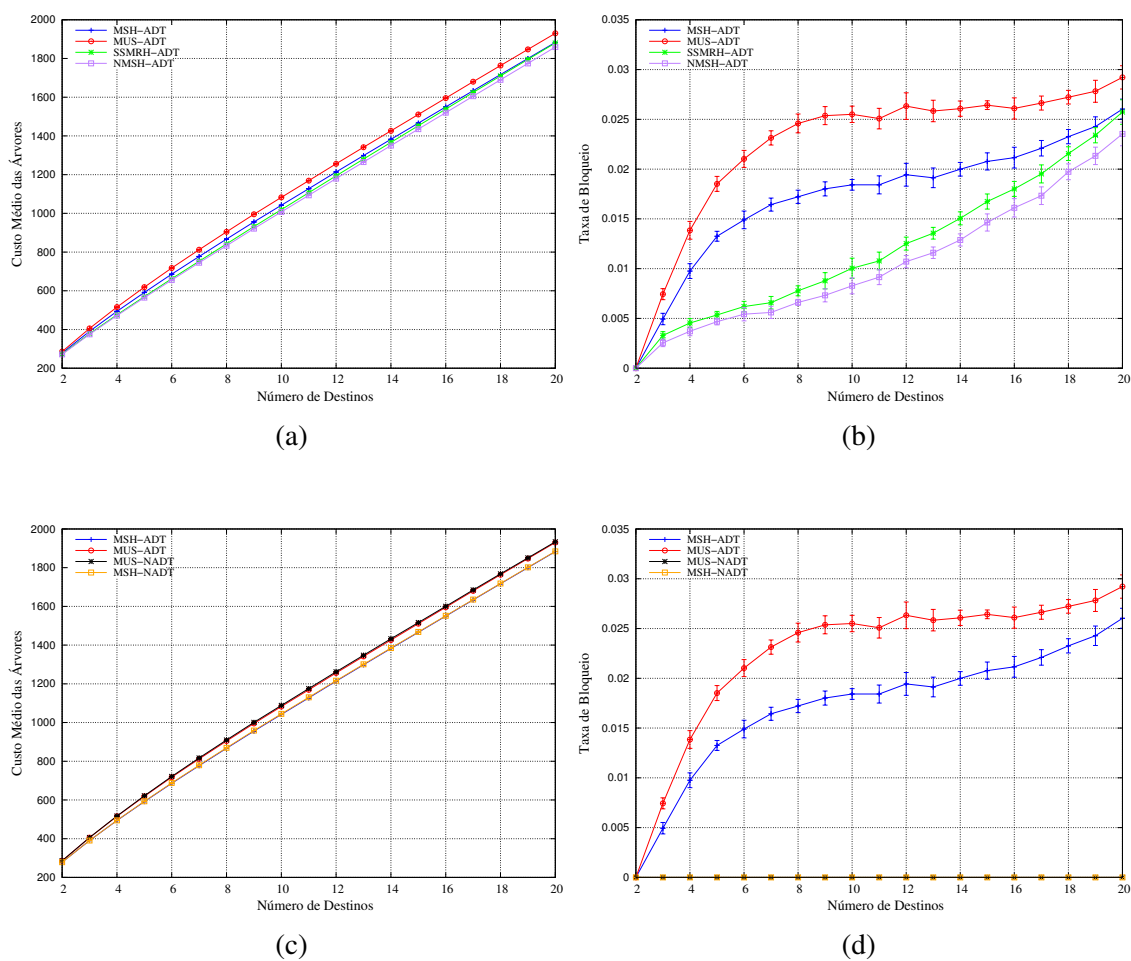


Figura B.5: (a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-0” com 5 nós MC.

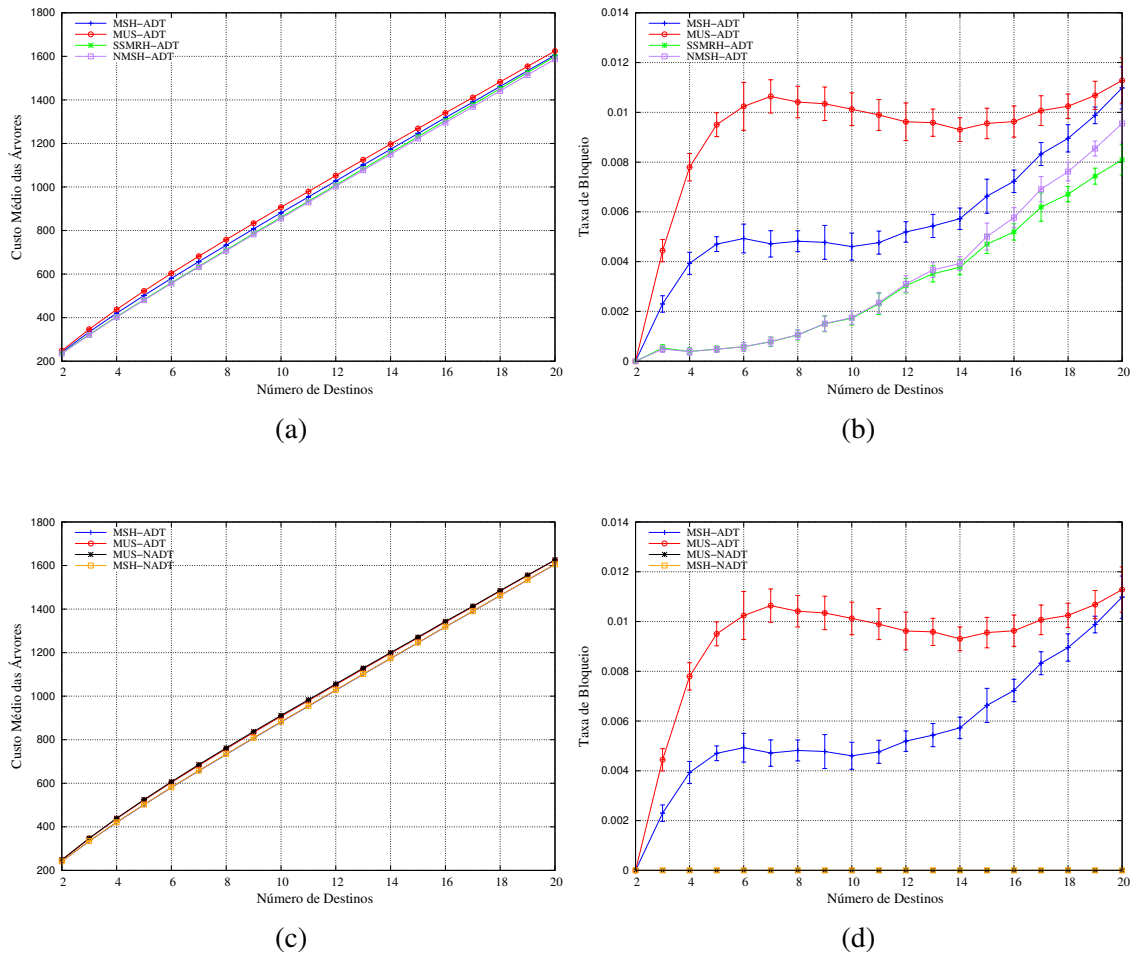


Figura B.6: (a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-1” com 5 nós MC.

B. Resultados Obtidos

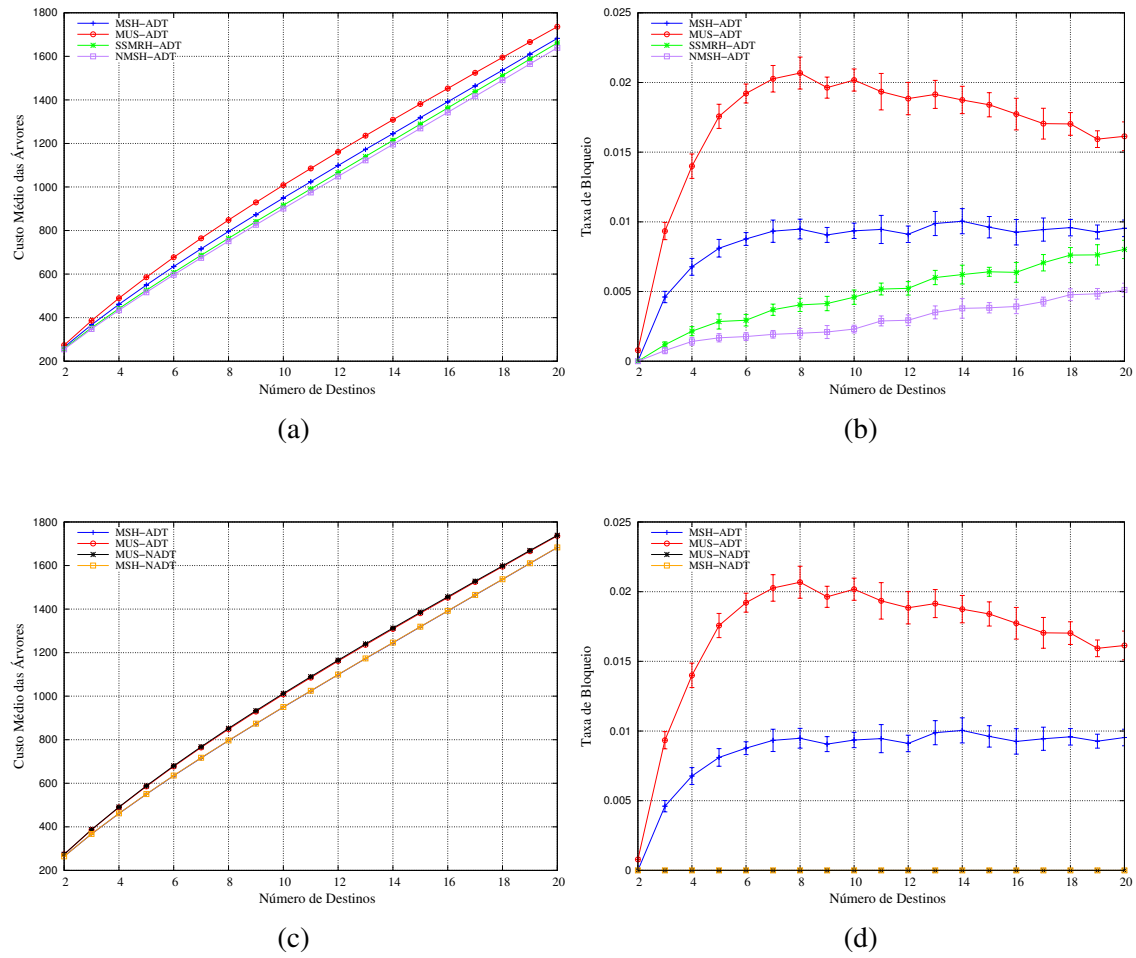


Figura B.7: (a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede40-3” com 5 nós MC.

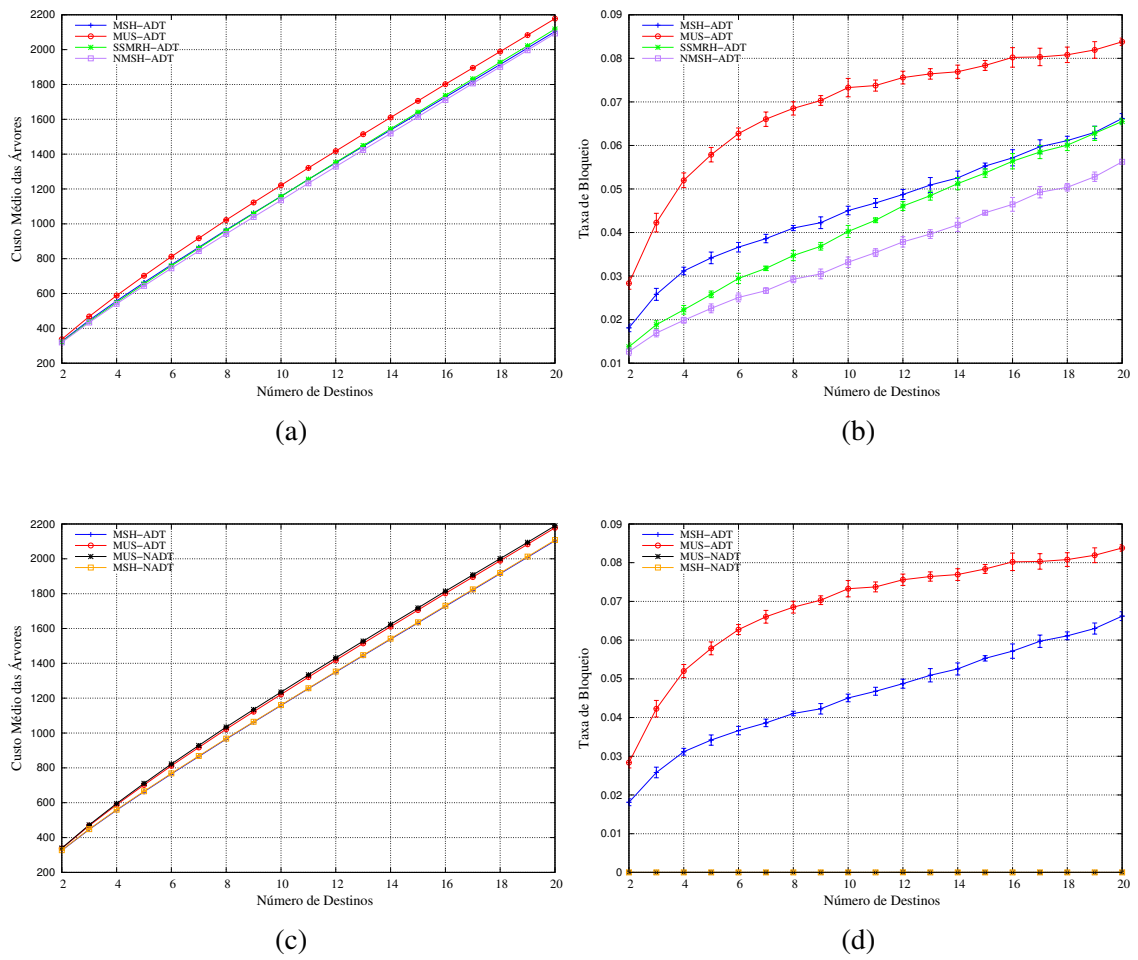


Figura B.8: (a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rede50” com 5 nós MC.

B. Resultados Obtidos

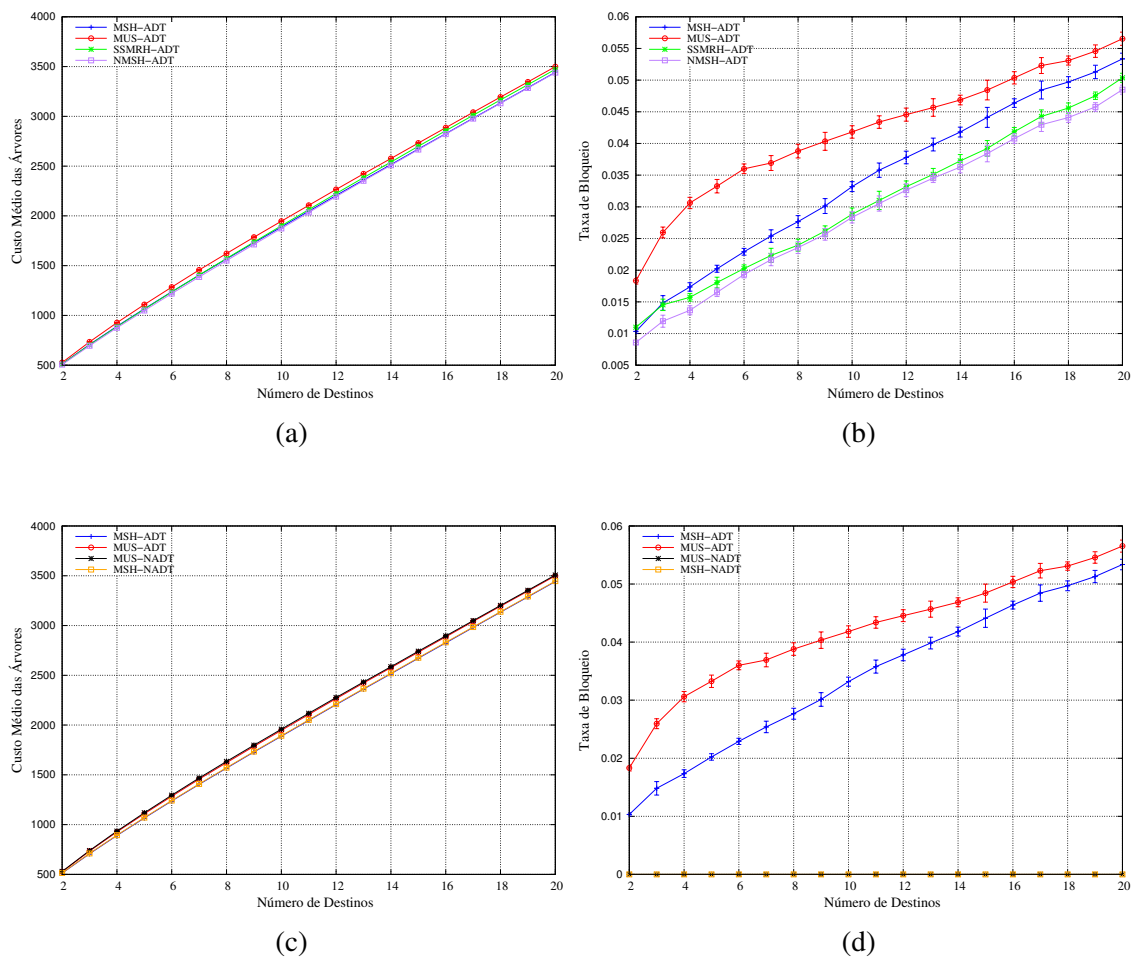


Figura B.9: (a) e (c) Custo médio das árvores disjuntas nos arcos por número de destinos e (b) e (d) taxa de bloqueio por número de destinos para a “rnd40” com 5 nós MC.

Bibliografia

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [2] M. Ali, “Optimization of splitting node placement in wavelength-routed optical networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1571–1579, Oct 2002.
- [3] M. Ali and J. Deogun, “Cost-effective implementation of multicasting in wavelength-routed networks,” *Lightwave Technology, Journal of*, vol. 18, no. 12, pp. 1628–1638, Dec 2000.
- [4] ———, “Power-efficient design of multicast wavelength-routed networks,” *Selected Areas in Communications, IEEE Journal On*, vol. 18, no. 10, pp. 1852–1862, October 2000.
- [5] Y. Cao and O. Yu, “Optimal placement of light splitters and wavelength converters for multicast in WDM networks,” in *Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on*, vol. 1, May 2005, pp. 585–589 Vol. 1.
- [6] C.-Y. Cheng-Yu Hsieh and W. Liao, “All-optical multicast routing in sparse splitting WDM networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 6, pp. 51–62, August 2007.
- [7] S.-H. Cho, T.-J. Lee, M. Y. Chung, and H. Choo, “Minimum cost multicast routing based on high utilization mc nodes suited to sparse-splitting optical networks.” in *ICCSA (2)*, ser. Lecture Notes in Computer Science, M. L. Gavrilova, O. Gervasi, V. Kumar, C. J. K. Tan, D. Taniar, A. Laganá, Y. Mun, and H. Choo, Eds., vol. 3981. Springer, 2006, pp. 288–297.
- [8] C. Constantinou and G. Ellinas, “Calculation of arc-disjoint trees in mixed-graph arbitrary mesh optical networks,” in *Ultra Modern Telecommunications and Control*

- Systems and Workshops (ICUMT), 2012 4th International Congress on*, Oct 2012, pp. 725–730.
- [9] —, “Survivable multicast routing in mixed-graph mesh optical WDM networks,” in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*, April 2012, pp. 1–4.
- [10] —, “A heuristic algorithm for multicast routing in sparse-splitting optical WDM networks,” in *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*, April 2013, pp. 65–69.
- [11] —, “Survivable multicast routing in mixed-graph sparse-splitting optical networks,” in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on*, Sept 2013, pp. 68–73.
- [12] C. Constantinou, G. Ellinas, and K. Manousakis, “Survivability of multicast requests in mesh optical networks,” in *Optical Network Design and Modeling, 2014 International Conference on*, May 2014, pp. 7–12.
- [13] D. Danh Le, M. Molnar, and J. Palaysi, “An improved multicast routing algorithm in sparse splitting WDM networks,” in *Computing, Management and Telecommunications (ComManTel), 2013 International Conference on*, Jan 2013, pp. 99–104.
- [14] E. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [15] M. Doar and I. Leslie, “How bad is naive multicast routing?” in *INFOCOM '93. Proceedings. Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future, IEEE*, 1993, pp. 82–89 vol.1.
- [16] W. Hu and Q. Zeng, “Multicasting optical cross connects employing splitter-and-delivery switch,” *Photonics Technology Letters, IEEE*, vol. 10, no. 7, pp. 970–972, July 1998.
- [17] W. Huang, L. Tang, M. Razo, A. Sivasankaran, M. Tacca, and A. Fumagalli, “Routing and wavelength assignment computed jointly for a given set of multicast trees reduces the total wavelength conversion,” in *Transparent Optical Networks (IC-TON), 2010 12th International Conference on*, June 2010, pp. 1–5.
- [18] P. Jia, Y. Liang, S. Huang, J. Zhang, and W. Gu, “Multicast with a new switching structure in optical networks,” *Photonic Network Communications*, vol. 15, no. 1, pp. 83–89, 2008.

- [19] H.-C. Lin and S.-W. Wang, "Splitter placement in all-optical WDM networks," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 1, Nov 2005, pp. 5 pp.–.
- [20] R. Malli, X. Zhang, and C. Qiao, "Benefits of multicasting in all-optical networks," *Proc. SPIE*, vol. 3531, pp. 209–220, November 1998.
- [21] B. Mukherjee, *Optical WDM Networks*. Springer Science and Business Media, Inc., 2006.
- [22] J. M. S. S. Pereira, *Matemática Discreta: Grafos, Redes, Aplicações*, 1st ed., ser. Ensaio e Livros Técnicos. Luz da Vida, Maio 2009.
- [23] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 2nd ed. Morgan Kaufmann Publishers, 2002.
- [24] L. Sahasrabudde and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *Communications Magazine, IEEE*, vol. 37, no. 2, pp. 67–73, Feb 1999.
- [25] N. Singhal and B. Mukherjee, "Protecting multicast sessions in WDM optical mesh networks," *Lightwave Technology, Journal of*, vol. 21, no. 4, pp. 884–892, April 2003.
- [26] N. Singhal, L. Sahasrabudde, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks," *Lightwave Technology, Journal of*, vol. 21, no. 11, pp. 2587–2594, Nov 2003.
- [27] N. Skorin-Kapov, "Routing and wavelength assignment in optical networks using bin packing based algorithms," *European Journal of Operational Research*, vol. 177, no. 2, pp. 1167 – 1179, 2007.
- [28] S. Subramaniam, M. Azizoglu, and A. Somani, "All-optical networks with sparse wavelength conversion," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 4, pp. 544–557, Aug 1996.
- [29] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Mathematica Japonica*, vol. 24, no. 6, pp. 573–577, 1980.
- [30] A. Tzanakaki, I. Zacharopoulos, and I. Tomkos, "Broadband building blocks [optical networks]," *Circuits and Devices Magazine, IEEE*, vol. 20, no. 2, pp. 32–37, Mar 2004.

Bibliografia

- [31] X. Wang, S. Wang, L. Li, and Y. Chuan, "Multicast routing in WDM networks with sparse light splitting," in *Communications, Circuits and Systems, 2007. ICCCAS 2007. International Conference on*, July 2007, pp. 578–582.
- [32] S. Yan, J. S. Deogun, and M. Ali, "Routing in sparse splitting optical networks with multicast traffic," *Computer Networks*, vol. 41, no. 1, pp. 89 – 113, 2003.
- [33] H. Zang and J. P. Jue, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, pp. 47–60, 2000.
- [34] X. Zhang, J. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting," *Lightwave Technology, Journal of*, vol. 18, no. 12, pp. 1917–1927, Dec 2000.
- [35] F. Zhou, M. Molnar, and B. Cousin, "Light-hierarchy: the optimal structure for multicast routing in WDM mesh networks," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, June 2010, pp. 611–616.
- [36] F. Zhou, M. Molnar, B. Cousin, and C. Qiao, "Cost bounds and approximation ratios of multicast light-trees in WDM networks," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 3, no. 4, pp. 323–334, April 2011.

