Metaheurísticas e Formulações para a resolução do Problema de Roteamento e Alocação de Comprimentos de Onda em Redes Ópticas

Alexandre Xavier Martins

▶ **To cite this version:**

Alexandre Xavier Martins. Metaheurísticas e Formulações para a resolução do Problema de Roteamento e Alocação de Comprimentos de Onda em Redes Ópticas. Other. Université Blaise Pascal - Clermont-Ferrand II, 2011. Portuguese. NNT : 2011CLF22160 . tel-00864176

HAL Id: tel-00864176

https://theses.hal.science/tel-00864176

Submitted on 20 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSIDADE FEDERAL DE MINAS GERAIS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Metaheurísticas e Formulações para a resolução do Problema de Roteamento e Alocação de Comprimentos de Onda em Redes Ópticas

Autor: Alexandre Xavier Martins

PPGEE - EEUFMG

Orientador: Rodney Rezende Saldanha

Co-orientador: Mauricio Cardoso de Souza

Université Blaise Pascal

Orientador: Philippe Mahey

Co-orientador: Christophe Duhamel

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte integrante dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica, área de concentração: Engenharia de Computação e Telecomunicações.

Belo Horizonte, 3 de Abril de 2013

–

Agradecimentos

Em primeiro lugar agradeço à Deus por todas as graças recebidas, pela força para superar cada um dos obstáculos e por estar ao meu lado sempre.

Ao Professor Mauricio Cardoso de Souza pela orientação, tempo dedicado e oportunidade de desenvolver este trabalho.

Ao Professor Rodney Rezende Saldanha pela orientação e pela oportunidade aberta no programa da engenharia elétrica.

Ao Professor Marcone Jamilson Freitas Souza que me apresentou à P.O. em 2001 e desde então tem sido não só um orientador e companheiro de trabalho, mas também um amigo.

Ao Professor Philippe Mahey por me receber no LIMOS e pela grande oportunidade oferecida.

Ao Professor Christophe Duhamel pela orientação e tempo dedicado para a realização desta tese.

Aos Professores e amigos Ricardo Saraiva de Camargo e Gilberto de Miranda Junior, pelas conversas, conselhos e paciência.

À amiga Cássia Ottoni, sempre disposta a ajudar no que for preciso.

Aos meus pais, João Francisco Martins e Ana Maria Xavier Martins, por terem sempre apoiado as minhas decisões.

Aos meus irmãos, Alisson Xavier Martins e Andre Xavier Martins pelo companheirismo.

Aos meus tios, Rômulo Angelo dos Santos e Maria Ocarlina Martins de Souza Santos, e meus primos, Melissa Martins de Souza Santos e Matheus Martins de Souza Santos, pelo apoio. Nem todas as páginas desta tese seriam suficientes para expressar a gratidão que tenho por vocês.

À família Vieira, José Carlos, Mira, e os fanfarrões, Tony, Francis e especialmente

à Carol por me aturar nesses 4 anos.

Aos amigos da França, Datta Ramadasan e Renata, Djellul Mameri, Edna, Erol Ozgur, Lahkdar Akroun, Libo Ren, Nardjees Manadjelia, Natália Vaz, Nathalie Bernard, Monica Unau e Raksmey Phan. Agradeço especialmente ao casal mais porreta que encontrei naquelas bandas, Heitor Liberalino e Emilie Raymond.

À Capes pela bolsa concedida no período que estive na França.

E à todos aqueles que de alguma forma contribuíram para a realização desta tese.

Resumo

Resumo do texto da tese apresentada à UFMG como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia de Elétrica

Nesta tese, apresenta-se um estudo sobre o Problema de Roteamento e Alocação de Comprimentos de Onda em redes ópticas WDM independentemente da topologia física subjacente. Inicialmente é feita uma revisão da literatura apresentando alguns modelos matemáticos formulados para a resolução do problema, também são revistos métodos para definição de limites inferiores e métodos heurísticos. Por se tratar de um problema da classe NP-difícil muitas heurísticas vêm sendo propostas para a sua resolução.

Apresentamos neste trabalho uma metodologia baseada na metaheurística Descida em Vizinhança Variável (*Variable Neighborhood Descent-VND*), que chamamos VND-BFD, primeiramente com o foco na eliminação dos comprimentos de onda e um método híbrido VND-BT para a resolução do problema.

Depois introduzimos uma nova abordagem também baseada no VND, mas dessa vez com o foco voltado para o rearranjo das requisições, quando esta nova versão do VND falha o método ativa um procedimento de perturbação. O novo método alterna entre uma busca através do VND e de uma perturbação, como na metaheurística *Iterated Local Search*. Definimos quatro variantes para este método, que chamamos de VND_r-ILS_p, VND_e-ILS_p, VND_r-ILS_{5p} e VND_e-ILS_{5p}.

As comparações feitas mostram que a abordagem com foco nas requisições se mostrou a mais eficiente com destaque para a versão VND_e-ILS_{5p}. O método proposto é competitivo com relação aos melhores métodos apresentados na literatura

até então, e foi capaz de melhorar a grande maioria dos limites superiores para as instâncias ainda em aberto.

Por fim, apresentamos modelos compactos voltados para a maximização da quantidade de requisições atendidas e algumas simplificações são propostas com o intuito de acelerar a resolução dos problemas. Ainda apresentamos alguns modelos da literatura baseados em geração de colunas e uma nova metodologia é proposta. Nos testes realizados esta nova metodologia, que chamamos PG-MAX-IS-IRC, foi capaz de resolver todas as instâncias em tempo sempre inferior ao método da literatura que usamos na comparação e sempre encontrado o mesmo limite superior deste.

PALAVRAS CHAVE: Otimização, telecomunicação, roteamento e atribuição de comprimentos de onda, programação matemática, geração de colunas, metaheurísticas.

Abstract

This work deals with the routing and Wavelength assignment (RWA) in optical WDM networks independently on the underlying physical topology. We begin with a review of the literature presented some mathematical models formulated to solve the problem, are also reviewed methods for setting lower bounds and heuristic methods. This problem has been shown to be NP-hard and several heuristic algorithms have been developed to solve it.

We present in this work a methodology based on metaheuristic Variable Neighborhood Descent (VND), which we call VND-BFD, primarily with the focus on the elimination of wavelengths and a hybrid method VND-BT to solve the problem.

Then we introduce a new approach also based on VND, but this time with the focus on the rearrangement of the requests, when this new version of the VND fails the procedure activates a disturbance, as the metaheuristic Iterated Local Search. We define four variants to this method, which we call VND_r-ILS_p, VND_e-ILS_p, VND_r-ILS_{5p} and VND_e-ILS_{5p}.

The computational experiments show that the approach with the focus on requests proved more efficient, especially the version VND_e-ILS_{5p}. The proposed method is competitive with respect to the best methods in the literature.

Finally, we present compact models aimed at maximizing the number of requests accepted and some simplifications are proposed in order to speed up the resolution of problems. Although we present some models of literature based on column generation and a new methodology is proposed. The new methodology, which we call PG-MAX-IS-IRC, was able to solve all instances faster than the method of the literature and always found the same upper bound.

KEYWORDS: Optimization, telecommunication, routing and wavelength alloca-

tion, mathematical programming, column generation, metaheuristic.

Résumé

Notre travail porte sur l'étude du Problème de Routage et d'Allocation de Longueur d'Onde (*Routing and Wavelength Allocation* - RWA) dans des réseaux optiques WDM, indépendamment de la topologie physique sous-jacente. Le problème a été identifié comme étant NP-difficile et plusieurs approches, tant exactes qu'approchées, existent. Nous fournissons d'abord une revue de littérature dans laquelle nous présentons quelques formulations mathématiques pour le problème ainsi que plusieurs manières d'obtenir des bornes inférieures et des heuristiques.

Nous considérons le problème min-RWA dans lequel on doit satisfaire un certain nombre de requêtes avec le moins de longueurs d'onde possible. Nous présentons une méthodologie reposant sur une recherche locale de type Descente à Voisinage Variable (*Variable Neighborhood Descent* - VND) que l'on appelle VND-BFD. Son objectif principal est de supprimer des longueurs d'onde. Nous présentons également une méthode hybride VND-BT.

Ensuite, nous proposons une nouvelle approche, elle-aussi reposant sur la VND. Elle consiste à ré-arranger les requêtes entre les longueurs d'onde disponibles. Lorsqu'elle atteint un optimum local, une procédure de perturbation est appliquée et le schéma est similaire à la Recherche Locale Itérée (*Iterated Local Search* - ILS). Quatre variantes sont définies selon les stratégies appliquées dans VND et ILS : VND$_r$-ILS$_p$, VND$_e$-ILS$_p$, VND$_r$-ILS$_{5p}$ et VND$_e$-ILS$_{5p}$. Les résultats expérimentaux montrent que cette nouvelle approche est plus performante, en particulier la version VND$_e$-ILS$_{5p}$. La méthode est compétitive avec les meilleures méthodes de la littérature puisque VND$_e$-ILS$_{5p}$ a permis d'améliorer une grande partie des meilleures solutions connues sur les instances standard du min-RWA.

Enfin, nous considérons aussi le problème max-RWA dans lequel on doit max-

imiser le nombre de requêtes traitées avec un nombre donné de longueurs d'onde. Nous proposons des modèles compacts ainsi que des améliorations destinées à accélérer la résolution par des solveurs en nombre entiers. Après avoir décrit des modèles existants utilisant la génération de colonnes, nous proposons un nouveau modèle, PG-MAX-IS-IRC, utilisant lui-aussi la génération de colonnes. Il permet d'obtenir des bornes supérieures de même qualité en un temps très fortement réduit.

MOTS-CLÉ: Optimisation, télécommunication, routage et allocation de longueur d'onde, modèle mathématique, génération de colonnes, métaheuristique.

Conteúdo

Resumo **v**

Abstract **vii**

Résumé **ix**

Résumé Étendu **xix**

1 Introdução **1**

1.1 Contexto Geral 1

1.2 Objetivos 5

1.3 Contribuições 6

1.4 Organização do Trabalho 7

2 Roteamento e Alocação de Comprimentos de Onda **8**

2.1 Introdução 8

2.2 Formulações para minimizar o número de comprimentos de onda . . . 11

2.3 Limites inferiores 15

2.4 Heurísticas e Metaheurísticas 17

2.5 Conclusão 22

3 Metaheurísticas aplicadas ao Problema de Roteamento e Alocação de Comprimentos de Onda **25**

3.1 Representação da Solução 25

3.2 Abordagens VND com foco nos Comprimentos de Onda 26

3.2.1 Descida em Vizinhança Variável 26

3.2.2 Método Híbrido VND-BT 30
3.3 Abordagem VND-ILS com foco nas requisições 31
3.3.1 Descida em Vizinhança Variável 32
3.3.2 Busca Local Iterativa 35
3.4 Resultados Computacionais 36
3.4.1 Resultados para o VND-BFD e VND-BT 38
3.4.2 Resultados para o VND-ILS 41
3.4.3 Comparações com a literatura 46
3.4.4 Análise de componentes $\text{VND}_e\text{-ILS}_{5p}$ 53
3.5 Conclusões 62

4 Formulações para maximizar o número de requisições estabelecidas 64
4.1 Modelos compactos 64
4.1.1 Formulações propostas 68
4.1.2 Resultados Computacionais 74
4.2 Modelos Estendidos 79
4.2.1 Abordagem proposta 84
4.2.2 Resultados Computacionais 88
4.3 Conclusão 96

5 Conclusões e Trabalhos Futuros 97
5.1 Conclusões 97
5.2 Trabalhos Futuros 98

Referências Bibliográficas 99

Lista de Tabelas

2.1 Comparação de limites inferiores [18] 17

3.1 Instâncias Realísticas 38
3.2 Resultados VND-BFD X VND-BT - Instâncias Realísticas 40
3.3 Resultados VND-BFD X VND-BT - Instâncias Z 41
3.4 Resultados VND-BFD X VND-BT - Conjunto Y 42
3.5 Resultados VND-ILS - Instâncias Realísticas 44
3.6 Resultados VND-ILS - Conjunto Z 45
3.7 Resultados VND-ILS - Conjunto Y3 46
3.8 Resultados VND-ILS - Conjunto Y4 47
3.9 Resultados VND-ILS - Conjunto Y5 48
3.10 Comparação com o multistart-BFD sobre as instâncias realísticas . . 49
3.11 Comparação com o multistart-BFD sobre o conjunto Z 50
3.12 Comparação com o multistart-BFD sobre o conjunto Y3 51
3.13 Comparação com o multistart-BFD sobre o conjunto Y4 52
3.14 Comparação com o multistart-BFD sobre o conjunto Y5 53
3.15 Resultados GA-RWA × VND_e-ILS_{5p} 59
3.16 Análise da dependência de cada vizinhança 61
3.17 Porcentagem de tempo e efetividade de cada etapa 62

4.1 Requisições da rede NSFnet 67
4.2 Resultados da relaxação dos modelos - Instâncias Z 75
4.3 Resultados da relaxação dos modelos - Instâncias NFSnet 76
4.4 Tempo até o ótimo - Instâncias Z 77
4.5 Tempo até o ótimo - Instâncias NFSnet 77

4.6 Instâncias Realísticas 89
4.7 MAX-IRC × PG-MAX-IS - W = 10 90
4.8 MAX-IRC × PG-MAX-IS - W = 20 91
4.9 MAX-IRC × PG-MAX-IS - W = 30 92
4.10 Resultados para W = 10 93
4.11 Resultados para W = 20 94
4.12 Resultados para W = 30 95

Lista de Símbolos

G	Grafo
V	Conjunto de vértices
E	Conjunto de arestas
v	Cardinalidade do conjunto de vértices ou número de vértices
Γ	Conjunto de Requisições por caminhos ópticos
r	Requisição individual definido por um par origem-destino
s_r	Origem da requisição r
d_r	Destino da requisição r
R	Matriz de requisições
r_{sd}	Número de requisições entre os vértices s e d
λ_{ij}^{sd}	Número de requisições entre os vértices s e d que utilizam o arco (i, j)
c	Configuração de Roteamento Independente
a_c	Vetor que representa c
a_{sd}^c	Número de caminhos entre os vértices s e d na configuração c
C	Conjunto de todas as possíveis configurações de roteamento independentes
Z_c	Número de vezes que uma determinada configuração de roteamento independente é utilizada
P	Conjunto de todos os caminhos elementares
P_{sd}	Conjunto de todos os caminhos elementares entre os vértices s e d
Λ	Conjunto de comprimentos de onda disponíveis
δ_{pl}	Indica se o caminho p pertence ao conjunto independente l
w_l	Número de comprimentos de onda associado ao conjunto independente l
y_{sd}	Número de requisições atendidas entre os vértices s e d

Lista de Figuras

1.1 Uma fibra transmitindo três comprimentos de onda distintos – [23] . 2

2.1 Instância estática do RWA 9
2.2 Representação da Solução 10
2.3 Exemplos de topologias em árvores 11
2.4 Exemplo de solução viável com 3 comprimentos de onda 13
2.5 Pseudocódigo do algoritmo Greedy-EDP-RWA - Adaptado de [39] . . 18
2.6 Pseudocódigo do algoritmo OneStepCD - Adaptado de [45] 19
2.7 Pseudocódigo do algoritmo TS-PCP - Adaptado de [45] 23
2.8 Pseudocódigo do algoritmo BFD-RWA - Adaptado de [50] 24

3.1 Representação da Solução 26
3.2 Algoritmo *VND* 27
3.3 Subgrafos sem os arcos já utilizados 28
3.4 Seleção dos subgrafos para a primeira vizinhança 29
3.5 Algoritmo Busca Tabu 31
3.6 Vizinhanças 34
3.7 Perturbação ILS por meio de um problema de atribuição 36
3.8 Instância Finland 39
3.9 Exemplo de Instância (Rede Z) - [43] 39
3.10 Gráfico para instância Att - Alvo 20 54
3.11 Gráfico para instância Finland - Alvo 46 54
3.12 Gráfico para instância NSF.12 - Alvo 38 55
3.13 Gráfico para instância y.4.20.4 - Alvo 19 55
3.14 Gráfico para instância y.5.80.2 - Alvo 59 56

3.15 Gráfico para instância y.5.100.2 - Alvo 73 56
3.16 Gráfico para instância y.4.80.1 - Alvo 72 57
3.17 Gráfico para instância y.4.100.1 - Alvo 89 57
3.18 Gráfico para instância z.10x10.60 - Alvo 86 58
3.19 Gráfico para instância z.10x10.80 - Alvo 114 58

4.1 NSFnet - [45] 66
4.2 Solução do modelo KS 68
4.3 Solução do modelo JMT 69
4.4 Solução dos modelos RWA1 e RWA2 74
4.5 Exemplo de conjuntos independentes [27] 79
4.6 Exemplo de Instância [27] 85
4.7 Conjunto Convexo para o problema MAX-IS 87

Lista de Siglas

BF Best Fit

BFD Best Fit Decreasing

DEMUX Demultiplexador

EDP Edge Disjoint Path

FDM Frequency-Division Multiplexing

FF First Fit

FFD First Fit Decreasing

LD Laser Diode

LED Light Emitting Diode

MUX Multiplexador

OXC Optical Crossconnect

PCT Problema de Concentração de Tráfego

PM Problema Mestre

PMR Problema Mestre Restrito

PTD Physical Topology Design

RWA Routing and Wavelength Assignment

SDM Space-Division Multiplexing

TDM Time-Division Multiplexing

TS-PCP Tabu Search-Partition Coloring Problem

WDM Wavelength Division Multiplexing

VTD Virtual Topology Design

Résumé

Introduction

Les réseaux optiques ont favorisé une forte croissance dans les télécommunications en général. En effet, ils possèdent une grande bande passante, ils ne sont pas affectés par les interférences électromagnétiques du fait de l'utilisation de matériaux diélectriques, ils résistent bien à la corrosion et ont un faible taux de perte. De plus il est possible de mettre en place des systèmes de transmission longue distance avec un espacement raisonnablement grand entre les répétiteurs, ce qui limite le coût et la complexité [9].

Cependant, la vitesse de transmission théoriquement offerte par la fibre optique est rarement atteinte à cause des traitements électroniques sur les routeurs [42]. Le mécanisme de commutation actuel nécessite du stockage et du traitement sur les routeurs. Il induit donc du retard dans la transmission et transforme les nœuds de commutation en points de congestion. De plus, la bande passante offerte par les fibres optiques n'est pas suffisante pour garantir l'efficacité du réseau [11]. Une clé pour obtenir des réseaux à grande vitesse est de maintenir le signal sous forme optique en évitant ainsi les transformations coûteuses sur les convertisseurs opto-électroniques, tant en entrée qu'en sortie du signal [36].

Un système de transmission des fibres optiques est fondamentalement constitué par un émetteur optique et un récepteur, et un câble de fibre optique. Le transmetteur se compose d'un dispositif émettant de la lumière et un circuit associé, qui a des fonctions de polarisation et de contrôle de puissance de sortie par le dispositif émetteur de lumière. Le dispositif émetteur de lumière est l'élément actif du système, il est responsable de la conversion opto-électronique des signaux. Deux

types de dispositifs sont couramment utilisés comme sources de lumière dans les systèmes de transmission par des fibres optiques: diodes laser (LD's) et diodes électroluminescentes (LED's). Plusieurs dispositifs ont été développés pour améliorer cette technologie, par exemple, des commutateurs optiques (Optical CrossConnect − OXC) et des amplificateurs optiques (OA).

De nombreuses recherches sont donc orientées multiplexage pour permettre de profiter des grandes capacités offertes. Les différentes techniques de multiplexage consiste à envoyer un signal sur chaque canal et introduire une séparation temporelle, spatiale et fréquentielle. Les multiplexages les plus répandues sont:

- Le multiplexage à répartition spatiale (*Spatial Division Multiplexing* − SDM): il consiste à partitionner l'espace physique pour augmenter la bande passante, par exemple, en regroupant un ensemble de fibres dans un câble, ou en utilisant des câbles multiples en un seul lien;

- Le multiplexage à répartition temporelle (*Time Division Multiplexing* − TDM): il consiste à consacrer la totalité de la bande passante à l'échange d'une donnée pendant un intervalle de temps avant de passer à la donnée suivante; l'incovénient de cette méthode est le gaspillage de la bande passante du fait que toutes les données envoyées n'ont pas la même taille; il arrive donc de consacrer un intervalle de temps très grand pour envoyer une donnée minimale;

- Le multiplexage à répartition fréquentielle (*Wavelength Division Multiplexing* − WDM): il consiste à séparer le lien en canaux, généralement 1nm; ces canaux correspondent à différentes longueurs d'onde. Plusieurs données sont envoyées en même temps sur différents canaux. Techniquement, on utilise des diodes lasers émettant à des longueurs d'ondes différentes mais assez proches et des multiplexeurs/démultiplexeurs optiques. Ce multiplexage n'est possible qu'avec des fibres optiques monomodes. Grâce au WDM, il est possible d'atteindre un débit de 10 Térabits. En plus, il est indépendant du protocole utilisé.

L'utilisation du multiplexage par division de longueur d'onde est devenu néces-

saire pour faire face à la croissance rapide de la demande pour les informations qui circulent à travers les réseaux. Le multiplexage est l'envoi d'un nombre de canaux par les mêmes moyens de communication. Cette technique permet d'expansion de la capacité des systèmes optiques sans besoin d'installer de nouvelles fibres. Cette technique permet d'expansion de la capacité des systèmes optiques sans besoin d'installer de nouvelles fibres, il permet d'utiliser une fibre optique unique comme plusieurs fibres virtuelles, portant chacune l'une de ces signaux de longueurs d'onde différentes, quelque soit le format de données effectué. Les connexions entre les nœuds d'un réseau WDM tout-optique sont établies par des chemins optiques (*Lightpaths*).

Les chemins optiques peuvent concerner un ou plusieurs nœuds du réseau. Le chemin optique exige l'utilisation d'une longueur d'onde (canal), c'est à dire chaque requête est associée à une longueur d'onde spécifique. Un sommet du réseau se compose d'un émetteur/récepteur, multiplexeur/démultiplexeur (MUX / DEMUX), un amplificateur optique et Ajoute/Extrait (Add / Drop − OADM), qui vise à extraire ou à ajouter le trafic qui circule sur le réseau.

Le OADM reçoit les informations, elles sont multiplexés puis envoyé à l'émetteur et se connecte ensuite à un amplificateur de ligne optique. Ensuite, le signal passe par un chemin optique pour atteindre la destination. Puis, le signal atteint le récepteur, qui a un démultiplexeur (DEMUX), où les longueurs d'onde sont de nouveau séparées et est alors dirigé vers l'OADM. S'il y a des sommets intermédiaires, la clé OXC est utilisée pour changer la lumière d'un sommet à l'autre, qui peut être un autre sommet intermédiaire ou même le sommet de destination.

La clé de ces réseaux à haut débit est de maintenir le signal sous forme optique, en évitant la surcharge de la conversion sous forme électrique, soit à l'arrivée ou au signal de sortie [36].

Associé au problème de répondre aux exigences d'un réseau à l'aide WDM, on peut citer les problèmes et leurs caractéristiques suivantes:

- La conception de la topologie virtuelle: il s'agit essentiellement d'un choix de chemins optiques du réseau. Pour ces connexions, on ne fait pas de traitement électronique pour le routage.

- Le routage et l'affectation de longueur d'onde: ce problème a essentiellement deux objectifs, accueillir les chemins optiques de la conception de la topologie virtuelle au cours de la topologie physique et l'affectation des longueurs d'onde sur chaque lien, de sorte qu'une longueur d'onde ne soit pas affectée à deux chemins optiques dans le même lien.

- Problème de concentration du trafic (*Traffic Grooming*): ce problème est défini comme le problème de la combinaison des demandes de faible vitesse sur les canaux de capacité de transmission élevée. En général, il a déjà prédéterminé les chemins optiques et le problème de la concentration du trafic est de combiner le trafic de ces voies [28].

Dans ce travail, nous allons traiter le problème du routage et l'affectation de longueur d'onde. En l'abscence de conversion, la contrainte de continuité de longueur d'onde correspond à l'utilisation d'une seule longueur d'onde pour chaque chemin optique d'une origine à une destination [5]. Dans la litérature, le problème de routage et affectation de longueur d'onde (Routing and Wavelength Assignement - RWA) est souvent considéré de deux manières différentes : soit maximiser le nombre de requêtes traitées, soit minimiser le nombre de longueurs d'onde utilisées.

Routage et l'affectation de longueur d'onde - RWA

Soit $G = (V; E)$ un graphe orienté où V est l'ensemble des $v = |V|$ sommets (les commutateurs) et E est l'ensemble des arcs (les fibres optiques). On note Γ l'ensemble des requêtes qui exigent de chemins optiques où chaque $r \in \Gamma$ est défini par une paire source-destination $(s_r, d_r) \in V \times V$. Notez que nous pouvons avoir deux requêtes différentes pour la même paire source-destination et dans ce cas si elles sont acheminées vers la même longueur d'onde, alors elles doivent être achemineés par des chemins disjoints dans G.

Il existe essentiellement deux types de problème de routage et d'allocation de longueurs d'onde, le problème statique dans lequel les requêtes entre les paires source destination sont pré-établies et le problème dynamique dans lequel les requêtes ar-

rivent de temps en temps. Initialement pour qu'une planification du réseau soit plus pratique pour résoudre le problème statique, le trafic initial sera évalué et mesuré. Au cours du temps, occasionnellement il y aura l'émergence de nouvelles exigences et des préoccupations ainsi il y aura une attribution successive de chemins individuels, provenant sur la scène dans le problème dynamique.

Le RWA est habituellement traité avec deux objectifs distincts. Le premier objectif est de minimiser le nombre de longueurs d'onde, où le problème est de trouver une partition minimum de Γ sur W sous-ensembles telles que les requêtes pour chaque Γ_w, $w = 1, \ldots, W$, peuvent être acheminées sur les chemins disjoints dans G. L'autre alternative est de maximiser le nombre de connexions établies pour un nombre limité de longueurs d'onde. Pour répondre à chaque demande, nous avons besoin de déterminer une route et d'affecter une longueur d'onde à cette demande. La restriction qui est imposée est que la même longueur d'onde ne peut pas être utilisée pour répondre à deux ou plusieurs requêtes différentes si les chemins de ces requêtes utilisent le même arc. En outre, nous avons la contrainte de continuité, où la longueur d'onde affectée à un chemin doit être utilisée du début à la fin de ce chemin, sans conversion. Cette restriction sera adoptée dans tous les travaux.

Chlamtac et al. [11] ont montré que le RWA est NP-complet pour des graphes de topologie arbitraire en réduisant le problème au problème de coloration de graphes. C'est à dire que trouver le nombre minimum de longueurs d'onde et que d'affecter toutes les requêtes est le même problème que trouver le nombre minimum nécessaire pour colorer un graphe tel qu'aucun nœud adjacent reçoit la même couleur, où le nombre de couleurs est le nombre de longueurs d'onde.

Pour résoudre le problème du routage et d'affectation de longueur d'onde sont proposés certains modèles de programmation mathématique dans la littérature. La plupart de ces modèles visent à maximiser le nombre de requêtes attribuées.

Ramaswami et Sivarajan [46] présentent une formulation pour le problème de routage et d'affectation des longueurs d'onde, en introduisant l'idée d'ensembles indépendants.

Lee et al. [36] ont présenté une formulation basée sur les chemins. Pour cela,

ils ont introduit la notion de configuration de routage indépendante. L'ensemble c est une configuration de routage indépendante si on peut faire a_{sd}^c connexions pour chaque $(s,d) \in R$ sur le graphe G simultanément en utilisant une seule longueur d'onde. Généralement, il ya un nombre exponentiel de configurations de routage indépendante, c'est à dire que le problème maître a un nombre exponentiel de variables de décision. Pour cette raison, Lee et al. [36] ont également présenté une stratégie basée sur la génération de colonnes [20] et un algorithme *branch-and-price* pour résoudre le problème, la méthode a été en mesure d'obtenir de bonnes solutions.

Krishnaswamy et Sivarajan [33] ont présenté un modèle avec l'objectif de maximiser le nombre de requêtes défini basé sur des arcs.

Des comparaisons des modèles de programmation mathématique pour le RWA avec l'objectif de maximiser le nombre de connexions établies peuvent être trouvées dans [24], [25] et [26].

Dans [27] est fait un examen des modèles de génération de colonnes pour le RWA afin de maximiser le nombre du connexions établies et quelques améliorations sont proposées. En outre, combinant les idées de Ramaswami et Sivarajan [46] et Lee et al. [36], Jaumard et al. [27] ont présenté une nouvelle formulation et un algorithme *Branch and Price* qui a été en mesure de résoudre certains cas de taille moyenne à grande.

Pour les problèmes qui ont un trafic uniforme, c'est à dire qu'une connexion doit être établi entre toutes les paires de sommets ($v \times (v-1)$ demandes), Baroni et Bayvel [7] ont obtenu des bornes inférieures introduisant des réductions dans le graphe.

Sharafat et Ma'rouzi [49] ont présenté une généralisation du travail de Baroni et Bayvel [7] en calculant des bornes inférieures pour des graphes de différentes matrices de trafic et de topologies arbitraires.

Skorin-Kapov [50], inspirée par la méthode introduite par Ramaswami et Sivarajan [47] à des problèmes de définition de topologies virtuelles, a présenté une méthode simple pour obtenir des bornes inférieures.

En 2009, Fisher et al. [18] ont adapté le modèle présenté par Banerjee et Mukher-

jee [4]. Dans ce travail, la contrainte de continuité sur la longueur d'onde a été assouplie et un algorithme pour obtenir des bornes inférieures pour la RWA a été fait.

Plusieurs travaux sont consacrés à la résolution du RWA minimisant le nombre de longueurs d'onde. Certaines stratégies décomposent le problème en deux parties [5, 37, 45]. Une route est d'abord déterminée pour chaque connexion puis une longueur d'onde est affectée à chaque route en résolvant un problème de coloration de graphe [10]. D'autres stratégies traitent les deux problèmes simultanément [39, 50].

Inspiré par Conway et al. [12], Chlamtac et al. [11] ont développé une heuristique gloutonne pour l'attribution des longueurs d'onde. Dans cet article, les chemins sont déterminés par un algorithme de plus court chemin. Ensuite, ils sont triés par ordre décroissant de taille, puis attribués à chaque longueur d'onde à tous les chemins disjoints possible. L'idée est que les routes plus grandes sont plus difficiles à attribuer et doivent donc être affectés en premier lieu.

Manohar et al. [39] ont proposé un algorithme constructive et randomisé pour RWA, le Greedy-EDP-RWA, cet algorithme est utilisé comme un algorithme de multipartite, exécuté par le nombre fixe d'itérations, en retournant la meilleure solution trouvée. L'idée est similaire aux travaux présentés par Chlamtac et al. [11] d'affecter le même longueur d'onde sur les chemins disjoints, mais ici, les chemins sont générés à chaque itération, l'ordre de toutes les demandes est tiré au hasard.

Banerjee et Sharan [6] ont utilisé un algorithme évolutionnaire pour déterminer le routage et de résoudre le problème de l'affectation des longueurs d'onde comme un problème de coloration de graphes.

Li et Simha [37] ont proposé pour résoudre le problème du routage et d'affectation des longueurs d'onde en résolvant le problème de coloration de partitions, une extension du problème de coloration de graphes. Dans ce travail, les auteurs ont adapté six méthodes de construction appliquées à la coloration de graphes. Les meilleurs résultats ont été obtenus avec l'algorithme onestepCD (One Step Color Degree).

Noronha et al. [45] ont utilisé la stratégie exposée dans le travail de Manohar et al. [39] pour générer deux chemins pour chaque connexion, puis ils ont utilisé une

procédure basée sur la *Tabu Search* pour déterminer les affectations des longueurs d'onde comme un problème de coloration partitions [37].

Skorin-Kapov [50] propose quatre heuristiques constructives basées sur des algorithmes similaires utilisées dans la résolution du *Bin Packing*. First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) et Best Fit Decreasing (BFD) sont quatre heuristiques connues pour résoudre le problème du *Bin Packing*. FF et BF font le tri des éléments au hasard, tandis que FFD et BFD sont triés selon le plus court chemin. FF et FFD font l'affectation à la première longueur d'onde faisable, tandis que BF et BFD font l'affectation à la meilleure longueur d'onde faisable.

Noronha et al. [43] ont amélioré les performances des méthodes présentées dans [50]. Les auteurs ont travaillé dans la structure des données et des stratégies de mise en œuvre, telle que la liste d'adjacence doublement enchaînée soit mise à jour dynamique des plus courts chemins. Les meilleurs résultats sont obtenus avec une variante de l'algorithme BFD.

Noronha et al. [44] ont proposé un algorithme génétique avec clés aléatoires pour trier les demandes. Les éléments sont triés en fonction de la taille et la valeur de la clé. Les chromosomes sont des vecteurs de nombres réels, appelées clés. Chaque demande est associée à une valeur de clé entre $[0, 1]$, cette valeur dans la population initiale est générée aléatoirement. Les clés sont utilisées pour décoder en générant des solutions réalisables. Les demandes sont triées en fonction de la somme de la taille de son plus court chemin entre l'origine et la destination, et la valeur de la clé, puis ensuite le BFD est appliqué. Pour générer une nouvelle population, la méthodologie proposée dans DeJong et Britney [51] est utilisée. Les résultats numériques ont été réalisés sur la plupart des cas étudiés dans la littérature ainsi que de nouvelles instances introduites dans [43]. L'algorithme a obtenu de meilleurs résultats que les méthodes proposées jusqu'à présent.

Heuristiques appliquée au problème de RWA

Nous proposons dans cette thèse deux approches heuristiques pour résoudre le problème de RWA.

Dans la première stratégie nous avons développé un algorithme VND, qui, avec l'heuristique BFD tente d'éliminer à chaque itération une longueur d'onde. En outre, malgré la première stratégie, nous combinons l'algorithme VND avec la Tabu Search présentés dans Noronha et al. [45].

La première étape de la méthode consiste à sélectionner deux longueurs d'onde. Le critère utilisé pour ce premier choix est de sélectionner deux sous-graphes (longueurs d'onde) qui forment ensemble le sous-graphe le plus proche du graphe d'origine. Le concept de sous-graphe le plus proche est défini comme suit: soit une collection de sous-graphes de $G = (V, E)$, deux sous-graphes $G' = (V, E')$ et $G'' = (V, E'')$ qui forment la collection plus proche du graphe G cas la différence entre la cardinalité de E et $E' \cup E''$ est la plus petite possible d'obtenir avec deux sous-graphes de la collection.

Avec les deux sous-graphes déterminés, l'étape suivante consiste à affecter les requêtes dans ces sous-graphes dans un autre graphe, quelque soient les tailles du chemins. Cette étape vise à libérer des arcs appartenant à des sous-graphes sélectionnés.

Enfin, on applique l'algorithme BFD sur les requêtes restantes dans sous-graphes sélectionnés. Il s'agit du voisinage 1-opt, où nous travaillons avec deux copies du graphe original. Si la solution retournée par BFD est meilleure, c'est à dire que si la méthode affecte toutes les requêtes en utilisant moins de sous-graphes (longueurs d'onde) qu'au début du processus, alors nous acceptons la solution, sinon nous irons vers le prochain voisinage en ajoutant un sous-graphe à la procédure. Dans cette étape, chaque nouveau graphe ajouté, est celui qui a le plus grand nombre d'arcs sans utilisation. Cette opération est répétée jusqu'à ce que le nombre de sous-graphes utilisés dans le mouvement atteint une limite prédéterminée, qui est un paramètre de la méthode.

La méthode BFD est utilisée pour générer la solution, ainsi que dans le raffinement de la solution. Dans le raffinement, il est utilisé sur un sous-ensemble de requêtes appartenant à un certain nombre de sous-graphes, en fonction du voisinage exploré.

Le *Tabu Search* présenté dans [45] a été adapté pour faire partie d'une procédure hybride pour résoudre le RWA. La méthode hybride fonctionne comme suit: lorsque les voisinages explorés par le VND ont été épuisés, nous avons affecté un sous-ensemble de requêtes dans un sous-ensemble de graphes (longueurs d'onde). Pour chaque requête, à ce point, nous avons deux chemins, un chemin généré lors de la construction de la solution initiale et un autre produit dans la dernière itération du VND. Ainsi, ces deux chemins de chaque requête sont passés à *Tabu Search*. Le *Tabu Search* est exactement égal à celui présenté dans [45].

Dans la seconde stratégie le focus du mouvement devient les requêtes individuelles.

Nous présentons une heuristique hybride, combinant deux méthodes approchées, une autre version du VND et une heuristique ILS. La solution initiale est construite par une heuristique classique, EDP ou BFD. La VND est utilisée comme Recherche Locale pour améliorer la solution courante. Ses voisinages reposent sur la modification de la longeur d'onde affectée à chaque connexion. La VND est intégrée dans une ILS. La solution obtenue à la fin de la VND est perturbée puis ré-optimisée par la VND. Le principe est d'explorer les bassins de solutions proches de la solution courante en essayant de conserver la plupart des bonnes caractéristiques de celle-ci. De plus, en partant d'une solution proche d'un optimum local, l'effort lié à la nouvelle recherche locale est fortement réduit. Nous proposons aussi une perturbation ne modifiant pas le nombre total de longueurs d'onde utilisées.

Avant d'appliquer un mouvement, il est nécessaire de choisir un sous-graphe. Nous considérons deux alternatives pour choisir sous-graphe. Dans la première variante, notée VND_r, le sous-graphe choisi est celui avec le moins de requêtes, c'est un choix naturel puisque l'objectif est de vider le sous-graphe. Dans la deuxième variante, notée VND_e, le sous-graphe choisi est celui qui a le plus grand nombre d'arcs sans utilisation.

Soit une solution réalisable, le VND_r ou VND_e applique trois types de mouvements pour réorganiser les requêtes entre les sous-graphes afin d'éliminer l'un d'eux. Dans les deux cas, les requêtes appartenant au sous-graphe choisis appartiennent à

une liste triée par ordre décroissant en fonction de la taille du plus court chemin dans le graphe G. Le VND parcourt la liste en utilisant les voisinages Φ_1, Φ_2 et Φ_3 pour essayer de déplacer les requêtes. Si le VND est réussi, le sous-graphe est vide et, par conséquent le nombre de longueurs d'onde peuvent être réduit. Si l'algorithme VND échoue dans le voisinage Φ_3 phase ILS de la méthode est appelée.

Nous attirons l'attention sur un mouvement d'amélioration, si cela se produit, ce sera uniquement lorsque la dernière requête du sous-graphe sera transférée dans un autre sous-graphe.

Un mouvement dans le voisinage Φ_1 tente d'affecter la première requête, requête r, dans la liste dans un autre sous-graphe de la solution. En cas de succès, nous prenons la prochaine requête de la liste, le cas échéant, on met à jour nombre de longueurs d'onde dans le cas où la liste est vide. En cas d'échec, on passe à la recherche suivante dans le voisinage Φ_2.

Dans le voisinage Φ_2 la première étape est d'essayer de trouver un nouvel espace pour la requête r dans un autre sous-graphe en déplaçant le maximum de requêtes de cette sous-graphe. Si la méthode peut déplacer au moins une requête du sous-graphe, alors nous pouvons essayer d'affecter la requête r dans ce sous-graphe. En cas de succès, nous passons à la prochaine requête sur la liste, le cas échéant, dans le voisinage Φ_1. Dans le contraire, la méthode passe au voisinage Φ_3 toujours avec la même requête r considérée. À chaque nouvelle recherche via Φ_2 un seul sous-graphe est choisi selon l'ordre de la création du sous-graphes.

Un mouvement dans le voisinage Φ_3 essaie de modifier la requête r avec une autre requête appartenant à un autre sous-graphe. Les échanges sont acceptés uniquement pour les requêtes qui ont le chemin plus court que celui de la requête r. Notez que le mouvement dans le voisinage Φ_3 ne réduit pas la cardinalité du sous-graphe. Si un changement n'est pas possible dans le voisinage Φ_3, on passe à l'étape de perturbation de la méthode.

La perturbation est appliqué lorsque le VND a échoué à changer une requête dans le voisinage Φ_3. Notez qu'à ce stade, le VND peut avoir fait quelques pas. Nous n'annulons pas ces mouvements faits. La perturbation se trouve sur un problème

de l'attribution construit pour réorganiser les requêtes entre les sous-graphes. La méthode choisit au hasard une requête r' pour chaque sous-graphe de la solution, à l'exception du sous-graphe que nous voulons éliminer.

Soit $G^a_{w'} = (V, E^a_{w'})$ le sous-graphe de G, où $E^a_{w'} = E - \cup_{r \in \Gamma_{w'} - \{r'\}} E_r$ (les arcs utilisés pour acheminer r' sont réactivés dans $G^a_{w'}$). La perturbation est fait pour trouver la meilleure façon d'attribuer $W - 1$ requêtes en $W - 1$ sous-graphes.

Le coût de l'attribution de la requête $i \in \Gamma_{w^i}$ pour le sous-graphe j est donnée comme suit:

- $c_{ij} = 2$, si $j = w^i$; {Coût de retourner la requête au le sous-graphe d'origine}
- $c_{ij} = \infty$, si il n'existe aucun moyen d'acheminer la requête i dans le sous-graphe G^a_j;
- $c_{ij} = 1 - \frac{\min_{spl}(i)}{\min_{spl}(i)(G^a_j)} - \max(\min_{spl}(i)(G^a_i) - \min_{spl}(G^a_j), 0)$, cas contraire,

où $\min_{spl}(i)(G')$ est le nombre d'arcs du plus court chemin entre s_i et d_i sur le graphe G' et $\min_{spl}(i)$ est le nombre d'arcs du plus court chemin de la requête i sur le graphe G.

Après avoir résolu le problème d'attribution, nous vérifions si l'affectation des requêtes dans les sous-graphes a changé. En cas de succès, la méthode revient à l'étape VND essayant le mouvement avec r dans le voisinage Φ_1. En cas d'échec, la perturbation se fait à nouveau.

Nous proposons deux façons de choisir la requête r' pour chaque sous-graphe pour construire le problème d'attribution. Dans la première variante, notée ILS$_p$, toutes les requêtes ont la même probabilité d'être choisies. Dans le deuxième cas, notée ILS$_5p$, les requêtes qui ont des chemins qui passent par l'origine ou la destination de la requête r dans lequel le VND a échoué de faire le mouvement dans le voisinage Φ_3, ont une plus grande probabilité d'être choisies. En fait, ces requêtes ont cinq fois plus de chance d'être choisies.

Résultats des heuristiques

Pour les tests, nous avons utilisé trois séries d'instance. Le premier ensemble est constitué d'instances qui peuvent être trouvé dans la littérature. Les deux autres ensembles sont des instances proposé dans [43].

Les algorithmes ont été implémentés en C++, et les tests ont été effectués sur un PC Core 2 Duo avec 4 Go de Ram, XP. Chaque méthode lance 5 fois chaque instance, 5 minutes pour chaque essai.

Le VND d'abord proposé produit des résultats similaires lorsque l'on combine la *Tabu Search*. Les variations du VND-ILS testé ont réussi à résoudre toutes les instances réalistes.

Le point culminant a été les variante $\text{VND}_e\text{-ILS}_{5p}$. Nous montrons que cette méthode est capable de produire de meilleures solutions que le BFD dans toutes les instances dans lesquelles le BFD n'est pas en mesure de trouver la solution optimale.

Ensuite, nous présentons une comparaison avec le GA-RWA proposé dans Noronha et al. [44]. Cette comparaison nous emmène à observer que la méthode proposée dans cette thèse est en mesure d'améliorer les limites de toutes les instances pour lesquelles le GA-RWA n'est pas en mesure de trouver la solution optimale.

Méthodes exactes appliquées au problème de RWA

Dans cette thèse, nous présentons quelques modifications aux modèles mathématiques basés sur des arches existantes à ce problème de routage et l'affectation des longueurs d'onde afin de maximiser le nombre de connexions établies.

Le but ici est de rendre les modèles plus "légers"et de pouvoir résoudre des instances de même taille dans un temps plus court ou de même instances plus grandes qui ne sont pas traitées par les modèles existants.

Nous présentons les modèles que nous appelons KS [33] et JMT [26]. Ensuite, nous montrons comment simplifier ces deux modèles et nous présentons les modèles RWA1 et RWA2. Nous présentons un test préliminaire pour montrer que les solutions fournies par les deux modèles modifiés sont des solutions viables à ce problème de

routage et d'affectation de longueur d'onde.

En outre, nous discutons plus en détail deux modèles basés sur la génération colonnes pour la résolution du RWA, en particulier les modèles MAX-IS [46] et MAX-IRC [27].Premièrement, nous introduisons une nouvelle restriction sur le modèle MAX-IS afin d'être en mesure de fournir des limites aussi bonnes que celles du modèle MAX-IRC.

Nous montrons que, pour garantir l'optimalité, le modèle MAX-IS doit générer tous les chemins possibles pour chacune des requêtes, mais ce modèle peut apporter de bonnes solutions si "bonnes"chemins sont transmis comme entrée. Pour cela, nous avons adapté le modèle présenté dans [6] et construit un modèle de génération de colonnes, appelée Path-Gen, uniquement pour fournir les chemins pour le modèle MAX-IS. Avec cela, nous avons observé que le modèle MAX-IS est capable de trouver la solution optimale pour la plupart des cas utilisés dans les tests de calcul. Cependant, ce n'est pas toujours possible. À la fin, nous avons adapté le modèle MAX-IRC et nous avons construit un modèle hybride, appelé MAX-IS-IRC, ce qui a permis de trouver les mêmes limites que celles du modèle MAX-IRC pour tous les cas avec un temps plus court, et pourtant cette nouvelle modèle est parvenue à résoudre une instance que le modèle MAX-IRC n'a pas pu résoudre.

Résultats des méthodes exactes

Les tests développés pour les modèles de programmation en nombres entiers au point ont été effectués sur des instances créées à l'aide de la méthodologie présentée dans [43] et aussi sur le réseau NSF avec le trafic généré de façon aléatoire. Le réseau NSF a 15 sommets, et les réseaux générés ont entre 12 et 16 sommets. Dans les essais, le nombre de longueurs d'onde a varié entre [3, 5, 7].

Tous les modèles sont capables de trouver la solution optimale pour toutes les instances, mais les modèles RWA1 et RWA2 résolvent toutes les instances en moins de temps que des modèles KS et JMT.

Pour les tests liés à la génération de colonnes on a utilisé 24 instances de la littérature, appelée instances réalistes. Les tests ont été effectués comme suit: pour

chaque méthode effectuée toutes les instances ont un nombre de longueurs d'onde disponibles dans [10, 20, 30].

Les méthodes testées sont: la version de la littérature MAX-IRC [27], le modèle de la génération de chemins combiné avec le modèle MAX-IS (PG-MAX-IS) et le modèle Path-Gen combiné avec modèle MAX-IS-IRC (PG-MAX-IS-IRC).

Nous pouvons voir que la stratégie de génération de chemins développé pour l'utilisation du modèle MAX-IS fonctionne bien et ne compromette pas la qualité des solutions pour les instances testés lorsque nous utilisons $W = 10$. Nous notons également que la méthode PG-MAX-IS-IRC est en mesure de contourner les limitations de la méthode PG-MAX-IS comme cela était prévu et on trouve toutes les solutions linéaire optimal.

Lorsque nous utilisons le modèle PG-MAX-IS pour $W = 20$, nous pouvons voir que ce modèle commence à montrer des pertes de qualité de la solution. Nous avons également montré que la méthode PG-MAX-IS-IRC a été en mesure de trouver les solutions optimales avec des limites aussi bonnes que la méthode de MAX-IRC.

Pour $W = 30$ la méthode PG-MAX-IS n'a pas de perte importante et, là encore, la méthode PG-MAX-IS-IRC est capable de trouver les mêmes solutions que celles du modèle MAX-IRC.

Enfin, nous présentons une comparaison entre le temps utilisé par les modèles MAX-IS et PG-MAX-IS-IRC pour résoudre chaque instance. Dans tous les tests la méthode PG-MAX-IS-IRC est en mesure de résoudre les instances dans le temps bien au-dessous au modèle MAX-IRC. Dans le meilleur des cas le modèle MAX-IRC résoud l'une des instances en utilisant un temps de 453.5% de plus que la méthode présentée et dans le pire des cas cette différence est de 23147.8%.

Conclusions

Dans cette thèse, une étude a été faite sur le problème de routage et d'affectation des longueurs d'onde dans les réseaux optiques WDM. Quelques modèles mathématiques et les principales heuristiques et métaheuristiques proposées dans la littérature ont été présentés.

Nous avons développé une nouvelle approche en utilisant métaheuristique VND, où l'accent est le réarrangement des requêtes. Le VND est utilisé comme la recherche locale dans une procédure métaheuristique basée sur *Iterated Local Search* et nous avons développé une perturbation basé sur un problème d'allocation qui n'affecte pas la valeur de la fonction objectif. Nous avons développé quatre variantes de cette approche. Cette approche a été très efficace et compétitive en comparaison avec les méthodes proposées dans la littérature, en particulier la variante $\text{VND}_e\text{-ILS}_{5p}$. Avec cette variante, toutes les limites supérieures des instances qui restent ouvertes ont été améliorées, et la méthode a été capable de résoudre toutes les instances réalistes. Nous menons des tests de calcul afin d'étudier chaque composante du $\text{VND}_e\text{-ILS}_{5p}$ montrant ainsi l'importance de chacun.

En ce qui concerne les méthodes exactes, nous commençons par présenter deux modèles compacts proposés dans la littérature pour résoudre le problème de routage et d'affectation des longueurs d'onde afin de maximiser le nombre de requêtes satisfaites. Nous avons ensuite proposé quelques simplifications sur ces modèles et nous avons montrer, avec des tests de calcul, que les simplifications n'affectent pas les limites trouvées par les modèles et accélérer la résolution des problèmes. En outre, nous présentons quelques modèles basés sur la génération de colonne dans la littérature, aussi pour résoudre le RWA afin de maximiser le nombre de requêtes satisfaites. Enfin, nous présentons une nouvelle approche de génération de colonnes combinant les modèles existants. Dans les expériences menées, l'approche proposée est capable de résoudre toutes les instances dans un temps plus court que le modèle le plus efficace dans la littérature.

Comme travaux futurs, nous proposons une étude plus approfondie de la nouvelle approche basée sur la génération de colonnes développée dans la thèse et le développement d'un algorithme *branch-and-price* pour résoudre le problème. Aussi, nous suggérons la mise en œuvre d'autres méthodes heuristiques et l'étude de l'influence de certaines requêtes qui entravent la résolution du problème.

Aussi pour les travaux futurs, nous proposons l'étude du problème de la concentration du trafic (*traffic grooming*), une extension du problème du routage et

d'affectation des longueurs d'onde, qui est encore peu étudié dans la littérature.

Capítulo 1

Introdução

1.1 Contexto Geral

As redes ópticas propiciaram um grande crescimento da área de telecomunicações em geral. Tais redes possuem uma grande largura de banda, não são afetadas por interferências eletromagnéticas por serem compostas de material dielétrico, são imunes à ação corrosiva de vários elementos químicos presentes no ar e possuem baixa taxa de perda. Portanto, é possível implantar sistemas de transmissão à longa distância com espaçamento razoavelmente grande entre os repetidores, o que diminui o custo e a complexidade [9]. Porém, a velocidade na qual é possível modular eletronicamente os sinais ainda causa diferença entre as velocidades de transmissão óptica e o processamento eletrônico [42]. O atual processo de comutação, processamento e tecnologias de armazenamento causam atrasos às capacidades de transmissão, transformando os nós de comutação nos focos de congestionamento. Portanto, a largura de banda fornecida pelas fibras de comunicação óptica não podem ser facilmente traduzida em eficiência da rede, ou seja, usar a largura de banda disponível [11].

Um sistema de transmissão por fibras ópticas é constituído basicamente por um transmissor e um receptor ópticos, mais um cabo de fibra óptica. O transmissor é composto por um dispositivo emissor de luz e um circuito associado que tem funções de polarização elétrica e de comando da emissão de potência luminosa pelo dispositivo emissor de luz. O dispositivo emissor de luz é o elemento ativo básico do sistema, responsável pela conversão eletro-óptica dos sinais. Dois tipos de dispositivos são normalmente utilizados como fontes luminosas em sistemas de transmissão

por fibras ópticas: diodos laser (LD's) e diodos eletroluminescentes (LED's). Diversos equipamentos foram desenvolvidos para o melhoramento desta tecnologia, como por exemplo, as chaves ópticas (Optical Crossconnect − OXC) e os amplificadores ópticos (AO).

Para fazer frente ao rápido crescimento na demanda de informações que atravessam uma rede começou-se a fazer uso da multiplexação por divisão de comprimento de onda. Multiplexar é enviar um certo número de canais pelo do mesmo meio de transmissão. A multiplexação por divisão de comprimento de onda (WDM - *Wavelenght Division Multiplexing*) é uma técnica que vem sendo utilizada juntamente com técnicas de encaminhamento de informações por uma rede óptica, auxiliando as telecomunicações [55]. Esta técnica permite a expansão da capacidade dos sistemas ópticos sem a necessidade de instalação de novas fibras, pois permite transformar uma única fibra óptica em várias fibras virtuais, cada uma destas transportando sinais com comprimentos de onda diferentes, independentemente do formato dos dados transportados. As conexões entre os nós de uma rede totalmente óptica WDM são estabelecidas por caminhos ópticos [11].

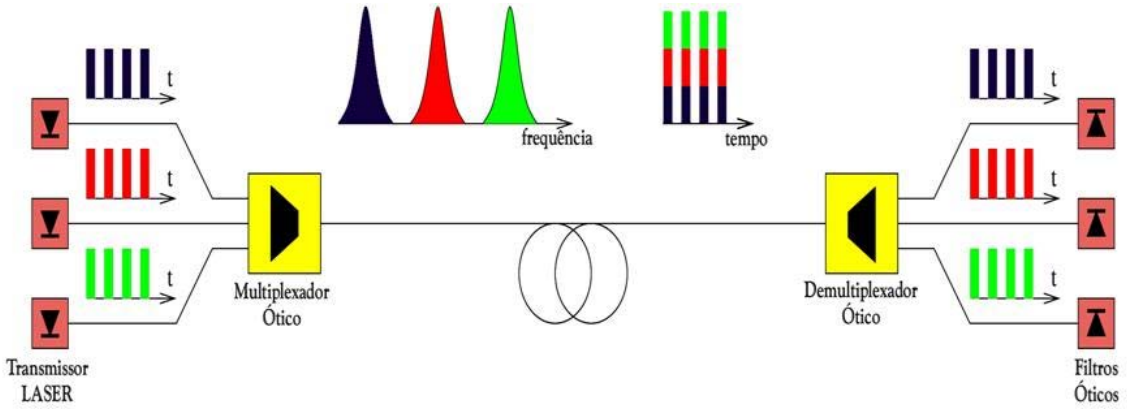


Figura 1.1: Uma fibra transmitindo três comprimentos de onda distintos − [23]

Os caminhos ópticos podem envolver um ou mais vértices da rede. O caminho óptico necessita do uso de um comprimento de onda, ou seja, cada requisição está associada a um comprimento de onda específico. Um vértice da rede é formado por transmissor/receptor, multiplexador/demultiplexador (MUX/DEMUX), amplificador óptico e Adiciona/Extrai (Add/Drop − OADM).

As informações chegam a um vértice, que pode ser um OADM, o qual tem por finalidade extrair ou adicionar ao sistema o tráfego que passa pelo vértice, ou um OXC que tem por finalidade comutar comprimentos de onda de uma porta de entrada para uma porta de saída. Em cada vértice são multiplexados os comprimentos de onda que passam pelo transmissor e, em seguida, conecta-se a um amplificador óptico de linha (AO). Depois passam por um caminho óptico até chegar ao vértice destino. Continuando, o sinal chega ao receptor, o qual tem um demultiplexador (DEMUX), onde os comprimentos de onda são separados novamente e, em seguida, é direcionado a outro vértice. Caso possua vértices intermediários é usada a chave OXC para chavear a luz de um vértice anterior para o próximo, que pode ser outro vértice intermediário ou até mesmo o vértice destino. A Figura 1.1 mostra um exemplo da técnica de multiplexagem: três sinais são modulados em três comprimentos de onda (três diferentes cores), mas são transmitidos ao mesmo tempo por uma única fibra.

A chave para a alta velocidade nestas redes é manter o sinal em sua forma óptica, evitando a sobrecarga da conversão na forma elétrica tanto na chegada quanto na saída do sinal [36].

Desde que não haja conversão, cada caminho óptico utiliza um comprimento de onda único em todos os arcos da origem ao destino, isto é chamado de restrição de continuidade do comprimento de onda [5].

Diferentes técnicas de multiplexação podem ser usadas nas redes ópticas WDM como, por exemplo:

- Multiplexação por divisão de espaço (SDM): particiona o espaço físico para aumentar a largura de banda, por exemplo, agrupando um conjunto de fibras em um único cabo, ou usando vários cabos em um único enlace [8].

- Multiplexação por divisão de freqüência (FDM): particiona o espectro de freqüência disponível em um conjunto de canais independentes. A utilização de FDM em uma rede óptica se reduz ao WDM, multiplexação por divisão de comprimento de onda que permite uma fibra levar tráfego em muitos comprimentos de onda distintos [8].

- Multiplexação por divisão de tempo (TDM): divide o domínio de tempo da banda em intervalos de duração fixa. Utilizando a TDM, múltiplos sinais podem compartilhar um determinado comprimento de onda se eles não se sobrepõe no tempo [8].

Ligados ao problema da acomodação das demandas em uma rede usando sistemas WDM, pode-se citar os seguintes problemas e características:

- Projeto da Topologia Virtual: consiste basicamente em uma escolha dos caminhos ópticos da rede. Estes são definidos com conexões ponto-a-ponto que são transparentes aos nós intermediários, isto é, para estas conexões não é realizado processamento eletrônico para fins de roteamento em tais nós [34].

- Roteamento e Atribuição dos Comprimentos de Onda: este problema possui basicamente dois objetivos: acomodar os caminhos ópticos provenientes do projeto da topologia virtual sobre a topologia física e atribuir os comprimentos de onda em cada enlace, afim de que um comprimento de onda não seja atribuído a dois caminhos ópticos em um mesmo enlace.

- Problema de Concentração de Tráfego (*Traffic Grooming*): este problema é definido como o problema de combinar demandas de baixa velocidade em canais de transmissão de alta capacidade. Em geral considera-se uma configuração de caminhos ópticos pré-estabelecidos e o problema de concentração de tráfego consiste em combinar o tráfego dentro destes caminhos [28].

Uma topologia física G_p é um grafo em que os arcos representam um par de fibras (um em cada direção) e os nós representam nós roteadores de comprimento de onda. O conjunto de todos os caminhos ópticos unidirecionais ao longo dos roteadores forma a topologia virtual G_v ou topologia de caminhos ópticos.

Para uma rede com V vértices, o ideal seria configurar caminhos ópticos para todos os $V(V-1)$ pares. Entretanto, isso não é usualmente possível primeiramente pelo número de comprimentos de onda disponíveis que impõe um limite na quantidade de caminhos ópticos que podem ser configurados (isto é também uma função

da distribuição de tráfego) e ainda cada nó pode ser fonte e destino de um número limitado de caminhos ópticos. Isto é determinado pela quantidade de hardware óptico que pode ser provido (transmissores e receptores) e pela quantidade total de informações que um nó pode processar [47].

A topologia virtual é dita simétrica se, sempre que existir uma ligação lógica do vértice i para o vértice j, existe uma outra ligação lógica de j para i, caso contrário a topologia virtual é dita assimétrica [34].

Definida a topologia virtual o próximo passo é saber como as ligações ponto a ponto desta topologia irão ser feitas sobre a topologia física, ou seja, como será feito o roteamento na topologia física e a alocação de comprimentos de onda para os enlaces virtuais estabelecidos previamente. Esse problema é conhecido como Roteamento e Alocação de Comprimentos de Onda (RWA - Routing and Wavelength Assignment).

O RWA visa atender às demandas definidas na etapa do projeto de topologia virtual, em geral minimizando recursos como a quantidade de comprimentos de onda utilizados ou maximizando a quantidade de requisições atendidas com um número limitado de comprimentos de onda disponíveis.

A fim de utilizar eficientemente a largura de banda oferecida pelas redes ópticas, fluxos independentes com baixa taxa de tráfego devem ser multiplexados em um único caminho óptico [15]. Esta observação dá origem ao conceito de concentração de tráfego (PCT), ou *Traffic Grooming Problem*, que refere-se às técnicas utilizadas para combinar os componentes de baixa velocidade em comprimentos de onda disponíveis para atender aos objetivos do projeto de rede, como a minimização de custos ou a maximização do fluxo na rede.

O RWA pode ser considerado um caso especial do PCT. A diferença é que no RWA, cada requisição ocupa a capacidade inteira de um comprimento de onda enquanto no PCT essa capacidade pode ser dividida para mais de uma requisição.

1.2 Objetivos

O objetivo desta tese é estudar o RWA independentemente da topologia da rede física subjacente. Para isso duas abordagens distintas serão apresentadas. Uma

abordagem heurística, visando soluções viáveis de boa qualidade em um tempo computacional razoável e uma abordagem exata, através de técnicas como a geração de colunas para a obtenção de limites inferiores. Como objetivos específicos deste trabalho, podemos destacar:

- Revisão de Literatura sobre o RWA;
- Construção de Heurísticas para a resolução do RWA;
- Testes computacionais das heurísticas propostas;
- Modelagem matemática para RWA;
- Testes computacionais para os modelos propostos;
- Desenvolvimento de métodos para determinação de limites inferiores;
- Comparações com a literatura.

1.3 Contribuições

Os seguintes elementos são as principais contribuições desta tese:

- Desenvolver e testar três buscas locais que visam a realocação de requisições;
- Desenvolver e testar um método de perturbação de soluções que não alteram o valor da função objetivo;
- Apresentar e testar simplificações aos modelos compactos existentes para a resolução do problema;
- Desenvolver um modelo de geração de colunas para a geração de caminhos;
- Desenvolver e testar uma nova abordagem de geração de colunas para a resolução do problema;

1.4 Organização do Trabalho

Este trabalho está organizado como segue: No Capítulo 1 é apresentada uma breve introdução e os objetivos são definidos. No Capítulo 2 é feita uma revisão da literatura a respeito do problema de roteamento e alocação de comprimentos de onda, sendo apresentados alguns modelos de programação matemática. Também são discutidos alguns métodos para a obtenção de limites inferiores e métodos baseados em heurísticas e metaheurísticas são apresentados. No Capítulo 3 é apresentada a metodologia proposta e desenvolvida para a obtenção de soluções para o problema de roteamento e alocação de comprimentos de onda; ao fim, os resultados são apresentados e comparados com métodos da literatura. No Capítulo 4 são apresentadas duas simplificações em modelos já existentes na literatura além de uma nova abordagem para a geração de colunas aplicada ao RWA. Por fim, no Capítulo 5, são apresentadas as conclusões e o que se pretende desenvolver posteriormente.

Capítulo 2

Roteamento e Alocação de Comprimentos de Onda

Neste capítulo é apresentada uma revisão acerca do problema de roteamento e alocação de comprimentos de onda (RWA - *Routing and Wavelength Assignment*). Primeiramente o problema é apresentado e definido. A seguir, são mostrados alguns modelos de programação matemática e métodos para obter limites inferiores. Por fim, é feita uma revisão sobre métodos heurísticos e sobre as metaheurísticas utilizadas para a resolução do problema.

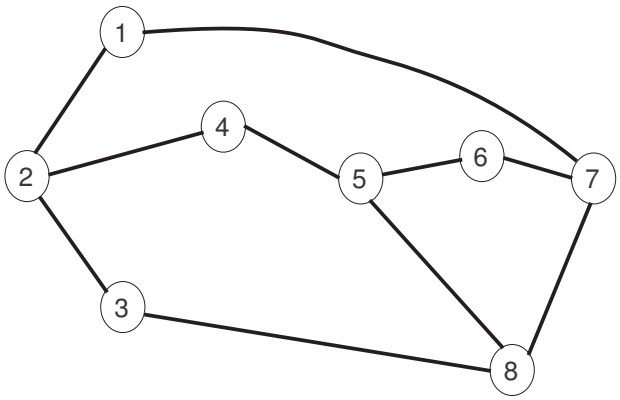
2.1 Introdução

Seja $G = (V, E)$ um grafo direcionado onde V, $v = |V|$, representa o conjunto de vértices (comutadores) e E o conjunto de arestas (fibras óticas). Nós denotamos por Γ o conjunto de requisições por caminhos ópticos onde cada $r \in \Gamma$ é definido por um par origem-destino $(s_r, d_r) \in V \times V$. Note que podemos ter duas diferentes requisições com o mesmo par origem-destino e neste caso se elas são roteadas com o mesmo comprimento de onda, então elas devem ser roteadas através de caminhos disjuntos por arestas em G.

Basicamente existem dois tipos de problemas de roteamento e alocação de comprimentos de onda, o estático no qual as requisições entre os pares origem destino já estão pré-estabelecidas, e o dinâmico em que as requisições chegam ao longo do tempo.

Na literatura do RWA podemos encontrar usualmente pelo menos duas aborda-

gens em se tratando do objetivo a ser alcançado. A primeira variante consiste em minimizar o número de comprimentos de onda utilizados, onde o problema é encontrar a mínima partição de Γ em W subconjuntos tal que as requisições em cada Γ_w, $w = 1, \ldots, W$, podem ser roteadas em caminhos disjuntos por arestas em G. A outra alternativa é maximizar o número de conexões estabelecidas para um número limitado de comprimentos de onda. Para atender cada requisição entre os pares, origem e destino, necessitamos determinar um caminho direcionado e alocar um comprimento de onda a esta requisição. A restrição que se impõe é que um mesmo comprimento de onda não pode ser utilizado para atender duas ou mais requisições diferentes, se os caminhos dessas requisições utilizam um mesmo arco. Além disso, podemos ter a restrição de continuidade, onde o comprimento de onda alocado a um caminho deve ser utilizado do início ao fim deste caminho, sem conversão. Essa restrição será adotada em todo trabalho.



Source	Destination
1	6
1	7
1	4

Figura 2.1: Instância estática do RWA

A Figura 2.1 apresenta uma instância estática para o RWA com o grafo e suas arestas e as requisições demandadas. A Figura 2.2 apresenta uma solução viável para a instância da Figura 2.1 com o objetivo de minimizar o número de comprimentos de onda, onde podemos ver que são utilizados dois comprimentos de onda, W_1 atendendo as requisições entre os pares $(1, 4)$ e $(1, 6)$ e W_2 atendendo a requisição entre o par $(1, 7)$.

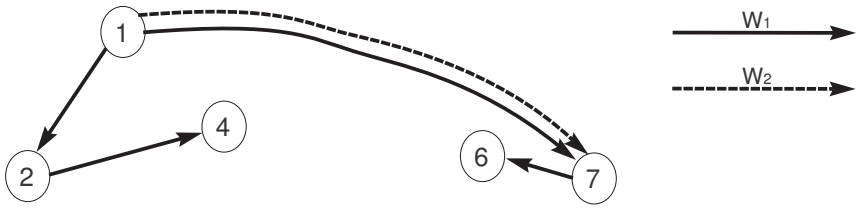


Figura 2.2: Representação da Solução

Note que, se G é uma árvore, cada par de nós pode ser conectado por somente um caminho; assim, a primeira parte do problema já estará resolvido, o roteamento. A complexidade do problema de minimizar número de comprimentos de onda alocados dependerá da topologia da árvore. Se G for uma árvore em cadeia (árvore com grau máximo igual a dois), estrela ou *spider* (árvore com exatamente um vértice de grau maior que dois) o problema de minimizar o número de comprimentos de onda pode ser resolvido em um tempo polinomial [15]. A Figura 2.3 mostra exemplos das topologias citadas. Ainda segundo Dutta e Rouskas [15], se G é uma árvore em cadeia, a segunda parte do problema, a alocação dos comprimentos de onda, é equivalente ao problema de coloração em grafos de intervalos e pode ser resolvido em tempo linear por um algoritmo guloso. Se G é uma árvore em estrela, a alocação é equivalente a encontrar o mínimo número de cores em arestas de um grafo bipartido, o qual pode ser resolvido em tempo polinomial combinando os teoremas de Hall e Konig [53]. Para árvores com topologia *spider* encontrar o número mínimo de comprimentos de onda é uma tarefa que também pode ser feita em tempo polinomial [54]. Segundo Corteel et al. [14], caso G seja uma árvore com uma topologia geral, ou seja, uma topologia diferente das definidas até então, o problema de minimizar o número de comprimentos de onda é NP-difícil.

Mostrando que o problema de estabelecer caminhos ópticos é equivalente ao problema de coloração de grafos, Chlamtac et al. [11] mostraram que o RWA é NP-completo para grafos com topologia arbitrária. Isto é, encontrar o número mínimo de comprimentos de onda para alocar todas as requisições é o mesmo que encontrar o

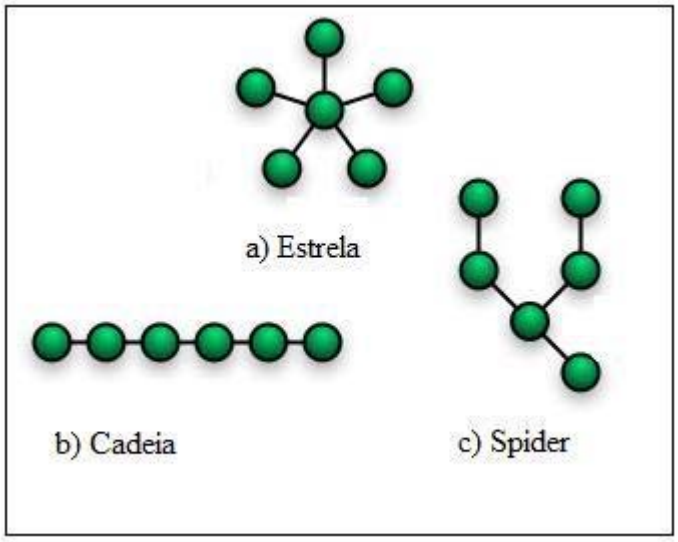


Figura 2.3: Exemplos de topologias em árvores

número mínimo necessário para colorir um grafo tal que nenhum nó adjacente receba a mesma cor, onde o número de cores corresponde ao número de comprimentos de onda. Erlebach e Jansen [16] mostraram que o RWA é um problema NP-difícil para redes com topologia em anel.

2.2 Formulações para minimizar o número de comprimentos de onda

Para a resolução do problema de roteamento e alocação de comprimentos de onda existem alguns modelos de programação matemática na literatura. Neste capítulo serão apresentados dois com o objetivo de minimizar o número de comprimentos de onda. Os modelos que serão apresentados tratam da versão estática e sem conversão dos comprimentos de onda ao longo do caminho.

Banerjee e Mukherjee [4] apresentam uma formulação para o problema baseada no problema de fluxo multiproduto. O modelo de programação matemática é usado para identificar um limite inferior para o número mínimo de comprimentos de onda necessário para atender certa configuração de um problema estático. O tráfego é definido por uma matriz R, $v \times v$ onde r_{sd} define o número de requisições demandadas entre os vértices s e d, onde $r_{sd} \geq 0$. Seja λ_{ij}^{sd} o número de requisições entre os pares

origem destino que usam o arco (i, j). O modelo de programação matemática para o problema é dado como segue:

$$\min \lambda_{max} \tag{2.1}$$

s.a:

$$\lambda_{max} \geq \sum_{(s,d)\in R} \lambda_{ij}^{sd}, \quad \forall\ (i,j) \in E \tag{2.2}$$

$$\sum_{j\in V|(i,j)\in E} \lambda_{ij}^{sd} - \sum_{j\in V|(j,i)\in E} \lambda_{ji}^{sd} = \begin{cases} r_{sd} & i = s, \\ -r_{sd} & i = d, \\ 0 & \text{caso contrário} \end{cases} \quad \forall\ (s,d) \in R, i \in V \tag{2.3}$$

$$\lambda_{ij}^{sd} \in \{0,1\}, \qquad \forall\ (s,d) \in R, (i,j) \in E \tag{2.4}$$

Nesta formulação, cada 'produto' é um caminho óptico da origem até o destino; neste caso não limita-se a capacidade das arestas, mas se necessário a capacidade pode ser entendida como o número de comprimentos de onda suportados em cada fibra. Banerjee e Mukherjee [5] usaram a relaxação deste modelo juntamente com um procedimento iterativo que a partir da solução do modelo de programação matemática, encontrava possíveis caminhos para cada requisição, aleatoriamente selecionava um destes caminhos e depois realizava a alocação. Esta formulação é frequentemente utilizada para obter limites inferiores para o problema. Isto se deve ao fato de que mesmo se resolvido na integralidade o modelo não apresenta uma solução completa. Por exemplo, se resolvermos a instância apresentada na Figura 2.1 obteriamos a seguinte solução:

$$\lambda_{max} = 2$$

Requisição $(1, 6)$:

$$\lambda[1, 6, 1, 7] = 1.0$$

$$\lambda[1, 6, 7, 6] = 1.0$$

Requisição $(1, 7)$:

$$\lambda[1,7,1,7] = 1.0$$

Requisição $(1,4)$:

$$\lambda[1,4,1,2] = 1.0$$

$$\lambda[1,4,2,4] = 1.0$$

Por esta solução, podemos ver que o número de comprimentos de onda é igual a 2. Através da variável λ podemos recuperar os caminhos de cada requisição, onde os dois primeiros índices representam respectivamente a origem e o destino e os dois últimos representam o arco usado, por exemplo $\lambda[1,4,1,2] = 1.0$ indica que a requisição $(1,4)$ utiliza o arco $(1,2)$, mas não sabemos como os comprimentos de onda foram alocados. Para este exemplo seria fácil encontrar a alocação ótima, mas para instâncias realísticas isso não é trivial. Além disso, outro problema com este modelo é que ele não impede a conversão do comprimento de onda, ou seja, uma mesma requisição pode utilizar vários comprimentos de onda ao longo do caminho. Na verdade o modelo nos indica um limite com relação ao arco mais congestionado, o que não quer dizer, necessariamente, que usando este limite em número de comprimentos de onda se consiga estabelecer todas as conexões. A Figura 2.4 apresenta uma solução viável para uma rede qualquer utilizando três comprimentos de onda. Pelo modelo apresentado anteriormente o número de comprimentos de onda necessários seriam de apenas dois, já que os arcos mais congestionados suportam somente dois comprimentos de onda, mas na verdade o número necessário seria realmente de três, já que os três comprimentos de onda dividem ao menos um arco na solução.

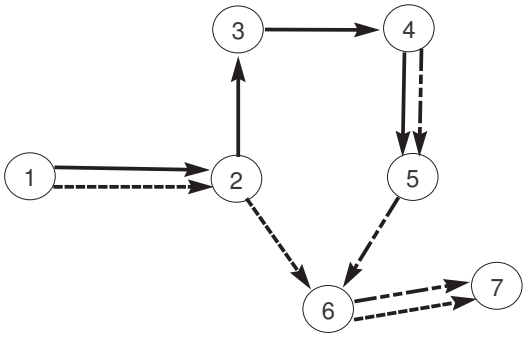


Figura 2.4: Exemplo de solução viável com 3 comprimentos de onda

Lee et al. [36] apresentaram uma formulação baseada em caminhos. Para apre-

sentar esta formulação será introduzido o conceito de configuração de roteamento independente. Uma configuração de roteamento independente c é representada por um vetor não-negativo $a^c \in \mathbb{R}^{v(v-1)}$, tal que $a^c_{sd} \leq r_{sd}$ para todo $(s,d) \in R$, onde a^c_{sd} representa o número de caminhos entre os vértices (s,d) na configuração c.

Uma configuração de roteamento c é independente se pode realizar a^c_{sd} conexões para todo $(s,d) \in R$ sobre o grafo G simultaneamente usando somente um comprimento de onda. Seja C o conjunto de todos as possíveis configurações de roteamento independentes, então a formulação de programação inteira é dada como segue:

$$(\text{MP}) \min \sum_{c \in C} Z_c \tag{2.5}$$

s.a:

$$\sum_{c \in C} a^c_{sd} Z_c \geq r_{sd}, \qquad \forall\ (s,d) \in R \tag{2.6}$$

$$Z_c \in \mathbb{N}, \qquad \forall\ c \in C \tag{2.7}$$

Cada variável de decisão é $Z_c = l$, se uma configuração de roteamento independente é realizada l vezes usando l comprimentos de onda distintos, onde l é um inteiro não-negativo, ou seja, uma mesma configuração de roteamento independente pode ser usada mais de uma vez, sendo que cada vez utiliza-se de um comprimento de onda diferente. A restrição (2.6) garante que todas as requisições serão atendidas. A relaxação desta formulação pode ser obtida retirando as restrições de integralidade impostas às variáveis de decisão.

Geralmente, existe um número exponencial de configurações de roteamento independente para uma instância do RWA, isto é, o problema mestre tem um número exponencial de variáveis de decisão. É então, impraticável enumerar todas as possíveis configurações de roteamento independentes. Devido a isso, Lee et al. [36] apresentaram também uma estratégia baseada em Geração de Colunas [20] e um algoritmo *branch-and-price* para a resolução do problema. O método se mostrou capaz de conseguir boas soluções e, ao mesmo tempo, limites inferiores bem fortes.

2.3 Limites inferiores

Nesta seção serão apresentadas algumas metodologias para obtenção de limites inferiores para o problema de roteamento e alocação de comprimentos de onda com o objetivo de minimizar o número de comprimentos de onda. Geralmente, para a obtenção de limites superiores, quando o objetivo é maximizar o número de requisições atendidas, são usadas as próprias formulações apresentadas na seção anterior.

Para problemas com uma matriz de tráfego onde uma conexão deve ser estabelecida entre todos os pares de vértices ($v \times (v-1)$ requisições) Baroni e Bayvel [7] obtiveram limites inferiores introduzindo cortes no grafo. Seja um corte no grafo G definido por V_1 e V_2, isto é, $V_1 \cap V_2 = \oslash$ e $V_1 \cup V_2 = V$ e seja C o conjunto de arestas do corte, isto é, C é o subconjunto de arestas de E com uma extremidade em V_1 e outra em V_2. Temos então $|V_1| \times |V_2|$ pares de vértices que demandam uma conexão através das arestas de C. O número mínimo de comprimentos de onda para um determinado conjunto C pode ser obtido por: $LI_C = \lceil \frac{|V_1| \times |V_2|}{|C|} \rceil$, onde $\lceil x \rceil$ é igual ao menor inteiro maior ou igual a x. Vários conjuntos de arestas podem ser utilizadas para formar C obtendo-se valores diferentes para LI_C. O conjunto que determinar o maior valor para LI_C determina o limite inferior desta metodologia.

Sharafat e Ma'rouzi [49] apresentaram uma generalização do trabalho de Baroni e Bayvel [7] calculando limites inferiores para grafos com matrizes de tráfego distintas e topologias arbitrárias.

Skorin-Kapov [50], inspirada na metodologia apresentada por Ramaswami e Sivarajan [47] para problemas de definição de topologias virtuais, apresentou uma metodologia simples para obtenção de limites inferiores. Neste trabalho são considerados dois limites inferiores, sendo o primeiro calculado da seguinte maneira: para cada vértice, dividimos o número de requisições que devem ser estabelecidas a partir do vértice (grau lógico) pelo número de arestas que saem do vértice (grau físico), o maior valor entre todos os vértices é o primeiro limite inferior. Matematicamente falando temos $max_{i \in V} \lceil \Delta_{l(i)} / \Delta_{f(i)} \rceil$. O outro limite inferior é obtido pelo somatório dos tamanhos (em número de arestas ou número de saltos) dos caminhos mínimos

de todas as requisições dividido pelo dobro do número de arestas. Matematicamente temos, $\lceil \sum_{r \in R} l(CM_r)/(2 \times |E|) \rceil$. Imagine que todas as requisições sejam atendidas utilizando-se o caminho mínimo entre a origem e o destino da requisição. No melhor caso vamos precisar de $\lceil \sum_{r \in R} l(CM_r)/(2 \times |E|) \rceil$ comprimentos de onda para acomodar estas requisições, sendo assim, $\lceil \sum_{r \in R} l(CM_r)/(2 \times |E|) \rceil$ é um limite inferior para o RWA. Obviamente os dois limites inferiores podem apresentar valores diferentes, assim podemos considerar aquele que resultar o maior valor.

Em 2009, Fisher et al. [18] adaptaram o modelo apresentado por Banerjee e Mukherjee [4] e relaxando a restrição de continuidade do comprimento de onda apresentaram um algoritmo para obter limites inferiores para o RWA. A Tabela 2.1, apresentada em [18], mostra uma comparação dos valores para os limites inferiores obtidos entre a metodologia apresentada por Fisher et al. [18] e Skorin-Kapov [50]. A primeira coluna mostra as instâncias testadas, a segunda coluna mostra os resultados do modelo (MCF) e a última coluna os resultados do método apresentado em Skorin-Kapov [50].

$$(\text{MCF}) \min(\max_{(i,j) \in E} \sum_{(s,d) \in R} \lambda_{ij}^{sd} + \sum_{(s,d) \in R} \lambda_{ji}^{sd}) \tag{2.8}$$

s.a:

$$\lambda_{ij}^{sd} + \lambda_{ji}^{sd} \leq r_{sd}, \qquad \forall\ (s,d) \in R, (i,j) \in V \tag{2.9}$$

$$\sum_{j \in V | (i,j) \in E} \lambda_{ij}^{sd} - \sum_{j \in V | (j,i) \in E} \lambda_{ji}^{sd} = \begin{cases} r_{sd}, & i = s \\ -r_{sd}, & i = d,\ \forall\ (s,d) \in R, i \in V \\ 0, & \text{caso contrário} \end{cases} \tag{2.10}$$

$$\lambda_{ij}^{sd} \geq 0, \qquad \forall\ (s,d) \in R, (i,j) \in E \tag{2.11}$$

Tabela 2.1: Comparação de limites inferiores [18]

Instância	MCF-LB	SK-LB
10.50.deg4	225	158
15.50.deg4	363	363
15.50.deg8	258	120
20.50.deg4	432	375
Atlanta	1256	1147
Germany50	147	147
Nobel-us	670	646
Norway	543	325
Polska	1682	1178
zib54	705	457

2.4 Heurísticas e Metaheurísticas

Na literatura são encontradas diversas estratégias para a resolução do problema de roteamento e alocação de comprimentos de onda com o objetivo de minimizar o número de comprimentos de onda utilizados. Algumas dessas estratégias decompõem o problema em duas partes ([37], [5], [45]). Primeiramente é calculada uma rota para cada conexão, em seguida são atribuídos comprimentos de onda para cada uma dessas rotas, de acordo com uma solução de um problema de coloração de grafos [10]. Outras estratégias tratam dos dois problemas em conjunto ([50], [39]).

Inspirados em Conway et al. [12], Chlamtac et al. [11] desenvolveram uma heurística gulosa para a alocação dos comprimentos de onda. Neste trabalho os caminhos são determinados por algum algoritmo de caminho mínimo. Depois são ordenados em ordem decrescente de tamanho, então aloca-se cada comprimento de onda a todos os possíveis caminhos disjuntos. A ideia é que caminhos maiores são mais difíceis de serem roteados e portanto devem ser roteados primeiro. Neste trabalho também são apresentadas heurísticas para outras variações do problema, por exemplo, com o número de comprimentos de onda limitado e com requisições chegando dinamicamente.

Manohar et al. [39] propuseram um algoritmo construtivo randomizado para o RWA ao qual chamaram de Greedy-EDP-RWA. O algoritmo é utilizado como um algoritmo de multipartida, executado um número fixo de iterações, retornando a me-

lhor solução encontrada. A ideia é semelhante ao trabalho apresentado por Chlamtac et al. [11] de alocar o mesmo comprimento de onda aos caminhos disjuntos, mas, aqui, os caminhos são gerados a cada iteração e a ordem de todas as requisições é estabelecida aleatoriamente. Além disso o caminho mínimo inserido em uma solução é limitado por $max(diametro(G), \sqrt{|E|})$, onde $diametro(G)$ representa o diâmetro do grafo que é dado pelo tamanho do maior dos caminhos mínimos do grafo. Este limite foi proposto por Kleinberg [32] para a resolução do problema de maximizar o número de caminhos com arestas disjuntas em um grafo. O pseudocódigo deste algoritmo pode ser visto na Figura 2.5.

```
procedimento Greedy − EDP − RWA(G, R)
1  d ← max(diametro(G), √E);
2  R' ← R; {Conjunto auxiliar com as requisições ainda não alocadas}
3  λ ← 0; {Contador do número de comp. de onda usados}
4  Enquanto (R' ≠ ∅) faça
5      λ ← λ + 1;
6      G' ← G; {Grafo auxiliar inicialmente igual ao original}
7      para i = 1 até |R'| faça
8          Seja r a i-ésima conexão em R' selecionada aleatoriamente;
9          Calcule o caminho mínimo em G' entre os vértices de r;
10         se o tamanho do caminho c ≤ d então
11             R' ← R' \ {r};
12             Remova os arcos do caminho c de G';
13         fim-se
14     fim-para
15 fim-enquanto
16 Retorne λ;
fim Greedy − EDP − RWA
```

Figura 2.5: Pseudocódigo do algoritmo Greedy-EDP-RWA - Adaptado de [39]

Banerjee e Sharan [6] utilizaram um algoritmo evolucionário para determinar o roteamento e resolveram o problema de alocação dos comprimentos de onda como um problema de coloração de grafos. A alocação de comprimentos de onda é resolvida como um problema de coloração de grafos onde cada conexão da topologia virtual é um vértice em um novo grafo, dito $P(G)$. Cada vértice em $P(G)$ é conectado a outro vértice por uma aresta não direcionada se o seu caminho divide uma mesma aresta na topologia física [39].

Li e Simha [37] propuseram resolver o problema de roteamento e alocação de comprimentos de onda através da resolução do problema de coloração de partições (PCP), uma extensão do problema de coloração de grafos. Neste trabalho os autores adaptaram seis métodos de construção aplicados à coloração de grafos. Os melhores resultados foram obtidos com o algoritmo onestepCD (*One Step Color Degree*). Este algoritmo é uma variante da heurística de coloração baseada no grau de cada vértice [10], adaptada ao PCP. Primeiramente, todas as arestas que conectam vértices na mesma partição são removidas. O grau-cor de cada vértice é definido como o número de vértices já coloridos que são a ele adjacentes, $CD(i)$ retorna o grau-cor do vértice i. A cada iteração, o vértice com o menor grau-cor em cada componente da partição é determinado. O vértice com o maior grau-cor dentre estes últimos é selecionado. Este vértice é colorido com a menor cor que não provoca conflitos e todos os demais vértices na mesma componente são removidos. Estes passos são repetidos até que cada componente tenha um vértice colorido. No fim, o número de cores utilizadas representa o número de comprimentos de onda usados. A Figura 2.6 apresenta o pseudocódigo do método *OneStepCD*, onde V^k representa os vértices da partição k para $k = 1, \ldots, q$.

procedimento *OneStepCD*(G)
1 Remova de G todas as arestas $(i, j) \in E : i, j \in V^k$ para todas as partições $k = 1, \ldots, q$;
2 $V' \leftarrow \oslash$; {Conjunto de vértices já coloridos}
3 Enquanto $|V'| < q$ faça
4 $X \leftarrow \oslash$; {Conjunto de vértices candidatos a receber uma cor}
5 para $k = 1$ até $q : V^k \cap V' = \oslash$ faça;
6 $X \leftarrow X \cup argmin\{CD(i) : i \in V^k\}$;
7 fim-para
8 $x \leftarrow argmax\{CD(i) : i \in X\}$;
9 $V' \leftarrow V' \cup \{x\}$;
10 Assinale a menor cor possível a x; {Cor que não gera conflito de coloração}
11 Remova de G todos os vértices em $V_{c(x)} \setminus \{x\}$;
12 fim-enquanto
fim *OneStepCD*

Figura 2.6: Pseudocódigo do algoritmo OneStepCD - Adaptado de [45]

Noronha et al. [45] utilizaram a estratégia apresentada no trabalho de Manohar

et al. [39] para gerar dois caminhos para cada conexão, o caminho é descartado caso seja criado um caminho igual, e depois utilizaram um procedimento baseado em Busca Tabu [22] para definir as atribuições dos comprimentos de onda como um problema de coloração de partições [37]. Depois de definidos os caminhos de cada requisição a solução inicial é gerada pelo algoritmo *OneStepCD* proposto por Li e Simha [37] usando $(\lambda + 1)$ cores. O procedimento de Busca Tabu então aplica uma busca local como descrito a seguir: uma solução inviável com λ cores é criada recolorindo-se aleatoriamente com uma das λ primeiras cores todos os vértices originalmente coloridos com a cor $(\lambda + 1)$. Um conflito de coloração ocorre quando dois vértices adjacentes em duas partições diferentes são coloridos com a mesma cor. O procedimento de busca local investiga uma a uma as componentes cujos vértices coloridos estejam envolvidos em um conflito de coloração. O método procura dentro de cada componente um vértice não-colorido que possa ser colorido ou um vértice colorido que possa ser recolorido com outra cor de modo a reduzir o número total de conflitos de coloração. Caso exista um vértice nesta situação, o algoritmo aplica as modificações apropriadas de coloração e move-se para uma nova solução, a partir da qual a busca recomeça. Caso contrário, uma nova componente é investigada. A busca é interrompida quando uma solução viável com λ cores é identificada. A seguir, a viabilidade desta coloração é destruída e dispara-se a busca por outra coloração, agora com $(\lambda - 1)$ cores. Este procedimento termina após a realização de um número máximo de iterações ter sido efetuado sem que uma solução viável tenha sido encontrada. A cada iteração da busca tabu, um par (vértice, cor) define um movimento a ser inserido na lista tabu. A Figura 2.7 apresenta um pseudocódigo do algoritmo descrito, onde $Conflitos(S)$ retorna o número de conflitos em uma dada solução S, os conflitos indicam que vértices adjacentes possuem a mesma cor. A variável $maxC$ representa o número de cores usadas na solução e mais uma vez V^k é o conjunto de vértices da partição k.

Skorin-Kapov [50] propõe quatro heurísticas construtivas similares baseadas em algoritmos utilizados na resolução do *Bin Packing*. *Bin Packing* é um problema clássico na literatura da otimização combinatória [19]. Seja uma lista de n itens

de vários tamanhos e um conjunto de *bins* idênticos com capacidade limitada. O problema consiste em armazenar todos os itens da lista no número mínimo de *bins* sem violar a restrição de capacidade. Como o problema é NP-difícil [19] diversos algoritmos têm sido propostos para a sua resolução. Um estudo mais aprofundado sobre o problema pode ser encontrado nos trabalhos de Alvim et al. [3], Coffman et al. [30] e Coffman et al. [29].

First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) e Best Fit Decreasing (BFD) são quatro heurísticas conhecidas para a resolução do problema de *Bin Packing*. As heurísticas FF e BF ordenam aleatoriamente os itens e percorrem o conjunto de *bins* em uma ordem sequencial alocando os itens. O algoritmo FF coloca o item no primeiro *bin* possível enquanto o algoritmo BF analisa cada *bin* e aloca o item naquele considerado "melhor". No caso do RWA, cada requisição é considerada um item. O conceito de melhor *bin* muda para o problema original e na abordagem adotada para a resolução do RWA. No *Bin Packing* clássico o melhor *bin* é aquele que fica com menos espaço vazio após a alocação de um item, enquanto no RWA o melhor *Bin* é aquele que aloca uma requisição utilizando o menor número de arcos [50]. Os algoritmos FFD e BFD ordenam os itens em ordem decrescente de tamanho e depois alocam esses itens da mesma maneira dos algoritmos FF e BF respectivamente.

Noronha et al. [43] melhoraram o desempenho dos métodos apresentados em [50], trabalharando na estrutura de dados e em estratégias de implementação, tais como, lista de adjacência duplamente encadeada e atualização dinâmica dos caminhos mínimos. Os melhores resultados são observados com uma variante do algoritmo BFD, em que as requisições são ordenadas em ordem decrescente de acordo com o tamanho do menor caminho (tamanho do caminho definido pelo número de arestas) entre a origem e o destino no grafo original. Iterativamente para cada demanda e cada comprimento de onda já atribuido é checado se a requisição pode ser roteada com este mesmo comprimento de onda. O melhor comprimento de onda é definido como aquele com o menor caminho viável entre a origem e o destino. Se nenhum comprimento de onda pode ser atribuido a esta demanda ou se todos os menores caminhos

são maiores que um determinado limite, um novo comprimento de onda é gerado e atribuído a esta demanda. A dificuldade do algoritmo BFD [50] é que muitas requisições apresentam o mesmo tamanho. Sendo assim as requisições de mesmo tamanho são ordenadas aleatoriamente.

Para contornar o fato do BFD ordenar muitas requisições de maneira aleatória, Noronha et al. [44] propuseram um algoritmo genético com chaves randômicas para ordenar as requisições. Os elementos são ordenados segundo seu tamanho e o valor da chave. Os cromossomos são vetores de números reais, denominadas chaves. A cada requisição é associada uma chave de valor entre $[0, 1]$; na população inicial esse valor é gerado aleatoriamente. As chaves são usadas para a decodificação gerando as soluções viáveis. As requisições são ordenadas de acordo com a soma do tamanho do seu caminho mínimo, entre a origem e o destino, e o valor da chave, depois, então o BFD é aplicado. A resposta do BFD, o número de comprimentos de onda, é usada como *fitness* de cada indivíduo. Para gerar os filhos é utilizada a metodologia proposta em Spears e deJong [51]. Experimentos computacionais foram conduzidos sobre as instâncias mais estudadas na literatura bem como as novas instâncias introduzidas em [43]. O algoritmo genético proposto alcançou melhores resultados do que os métodos propostos até então. Noronha et al. [44] mostraram que o método apresentou soluções melhores ou iguais a aquelas encontradas por um multistart-BFD usando em média 23% de tempo a menos e reduziram em média à metade os desvios encontrados pelo esquema de decomposição apresentado em [43].

2.5 Conclusão

Neste capítulo foi apresentada uma revisão acerca do problema de roteamento e alocação de comprimentos de onda. Foram mostradas as formas como este problema é tratado na literatura, com variações na função objetivo e na maneira em que as requisições são tratadas. Além disso, foram discutidos os principais métodos heurísticos para a resolução do RWA com o objetivo de minimizar o número de comprimentos de onda em sua versão estática.

procedimento $TS - PCP(G, S, maxC)$
1 Construa uma solução S' recolorindo aleatoriamente com $maxC - 1$ cores todos os vértices de cor $maxC$;
2 Inicialize a Lista Tabu;
3 $Iter \leftarrow 0$;
4 Seja Q o conjunto formado por todas as componentes envolvidos em conflitos em S';
5 Enquanto $Q \neq \oslash$ faça
6 Aleatoriamente selecione $k \in Q$ e atualize $Q \leftarrow Q \setminus \{k\}$;
7 $Reduziu \leftarrow .False$;
8 $maxConflitos \leftarrow \infty$;
9 Para cada $i \in V^k$ para cada $l = 1, \ldots, maxC - 1$ faça
10 Enquanto $!Reduziu$ faça
11 Se $(i, l) \notin$ Lista Tabu ou Se satisfaz o critério de aspiração então
12 Obtenha uma solução S'' recolorindo o vértice i com a cor l;
13 Se $Conflitos(S'') < maxConflitos$ então
14 $maxConflitos \leftarrow Conflitos(S'')$; $S^* \leftarrow S''$; $k^* \leftarrow k$;
15 Se $Conflitos(S'') < Conflitos(S')$então
16 $S' \leftarrow S''$; $Reduziu \leftarrow .True$; $Iter \leftarrow Iter + 1$; atualize Q;
17 fim-se
18 fim-enquanto
19 fim-para
20 fim-enquanto
21 Se $Conflitos(S') = 0$ então
22 $S \leftarrow S'$; $maxC \leftarrow maxC - 1$; Retorne ao passo 1;
23 Senão
24 Seja i^* o vértice de cor $l^* \in V^{k^*}$ de S'; $\{k^*$ definido na linha 14$\}$
25 Insira o par $(i^*, l^*$ na Lista Tabu por $TabuIter$ iterações;
26 $S' \leftarrow S^*$;
27 $Iter \leftarrow Iter + 1$;
28 Se $Iter < maxIter$ então Retorne à linha 4;
29 fim-se
30 Retorne S;
fim $TS - PCP$

Figura 2.7: Pseudocódigo do algoritmo TS-PCP - Adaptado de [45]

procedimento $BFD - RWA(G, R)$
1 $d \leftarrow max(diametro(G), \sqrt{|E|})$;
2 $S \leftarrow \oslash$; {Conjunto que guarda a solução}
3 $B \leftarrow \oslash$; {Conjunto que guarda as cópias do grafo original DEFINIR}
4 para $i = 1$ até $|R|$ faça
5 se $\nexists$ caminho disjunto $\leq d$ em nenhuma cópia de $G \in B$ então
6 Crie uma nova cópia $\overline{G}$ de G;
7 $B \leftarrow B \cup \{\overline{G}\}$;
8 fim-se
9 Encontre a j-ésima cópia $G_j \in B$ onde a i-ésima requisição possui o menor caminho;
10 $c \leftarrow$ o menor caminho para i no grafo G_j;
11 $S \leftarrow S \cup \{(i, c, j)\}$; {Para cada requisição i guarda-se o caminho c e o índice j em que ela foi alocada}
12 Remova os arcos do caminho c da cópia G_j;
13 fim-para
14 Retorne S;
fim $BFD - RWA$

Figura 2.8: Pseudocódigo do algoritmo BFD-RWA - Adaptado de [50]

Capítulo 3

Metaheurísticas aplicadas ao Problema de Roteamento e Alocação de Comprimentos de Onda

Neste capítulo serão apresentados alguns métodos para a resolução do problema de roteamento e alocação de comprimentos de onda com o objetivo de minimizar o número de comprimentos de onda utilizados para uma certa configuração de demanda, onde as demandas são predeterminadas, ou seja, a versão estática do problema.

Antes, definimos uma solução viável por uma partição de Γ em W subconjuntos tais que existam caminhos disjuntos para todas as requisições pertencentes a cada Γ_w, $w = 1, \ldots, W$, em G. Cada $\Gamma_w \subseteq \Gamma$ induz um subgrafo $G_w = (V, E_w)$ de G onde $E_w = E - \bigcup_{r \in \Gamma_w} E_r$. Em outras palavras, G_w contém os arcos não usados para rotear requisições em Γ_w.

3.1 Representação da Solução

A solução é representada por um conjunto de caminhos, um caminho para cada requisição, e um conjunto de índices que representa em qual subgrafo a requisição está alocada, lembrando que cada subgrafo representa um comprimento de onda. A Figura 3.1 mostra uma representação parcial de uma solução. A Figura 3.1 (a) mostra uma matriz em que cada linha representa uma requisição, na primeira coluna mostramos o índice da requisição e sua origem e destino. As outras colunas representam os vértices. Esta matriz é responsável por guardar os caminhos que

podem ser recuperados pelos predecessores. Por exemplo, a requisição 1 deve ser uma ligação entre os vértices 1 e 4. Pela matriz podemos ver que para chegar em 4 passamos antes pelo vértice 2 e antes desse pelo vértice 1, que é o vértice origem da requisição 1.

A Figura 3.1 (b) mostra o vetor usado para representar a alocação dos comprimentos de onda. Em cada posição este vetor nos diz a qual subgrafo a requisição e seu respectivo caminho estão alocados. A Figura 3.1 (c) apresenta os caminhos para as três requisições do exemplo.

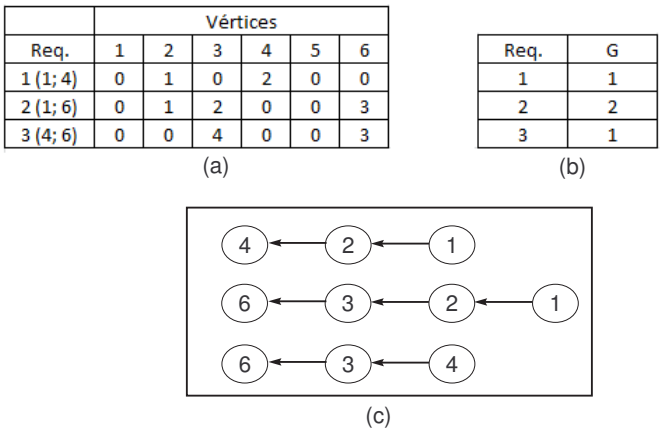


Figura 3.1: Representação da Solução

3.2 Abordagens VND com foco nos Comprimentos de Onda

Nesta seção serão apresentados a forma de representação computacional de uma solução para o problema, a forma de geração de uma solução inicial, a estrutura de vizinhança e os métodos usados para a resolução do RWA, no caso, o método de descida em vizinhança variável e a metaheurística Busca Tabu.

3.2.1 Descida em Vizinhança Variável

O Método de Descida em Vizinhança Variável (*Variable Neighborhood Descent*, VND), proposto por Mladenovic e Hansen [41], é um método de refinamento que consiste em explorar o espaço de soluções através de trocas sistemáticas de estru-

turas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada.

O pseudocódigo desse algoritmo, em que se considera o refinamento de uma solução s utilizando uma função de avaliação f, a ser minimizada, e um conjunto Φ de k_{max} diferentes vizinhanças $\Phi = \{\Phi^{(1)}, \Phi^{(2)}, \cdots, \Phi^{(k_{max})}\}$, é apresentado pela Figura 3.2.

```
procedimento VND(f(.),Φ(.),k_max,s)
1  Seja k_max o número de estruturas diferentes de vizinhança;
2  k ← 1;              {Tipo de estrutura de vizinhança corrente}
3  enquanto (k ≤ k_max) faça
4      Encontre o melhor vizinho s' ∈ Φ^(k)(s);
5      se (f(s') < f(s))
6          então
7              s ← s';
8              k ← 1;
9          senão
10             k ← k + 1;
11     fim-se;
12 fim-enquanto;
13 Retorne s;
fim VND;
```

Figura 3.2: Algoritmo *VND*

Dependendo do problema abordado, a busca pelo melhor vizinho (linha 4 da Figura 3.2) pode ser cara computacionalmente. Nessa situação é comum fazer a busca pela primeira solução de melhora. Outra alternativa é considerar a exploração apenas em um certo percentual da vizinhança.

Em nosso caso, o primeiro método desenvolvido, que chamaremos de VND-BFD, é aplicado da seguinte maneira: Seja uma solução inicial, lembrando que, em uma solução qualquer teremos um conjunto de caminhos e um conjunto de subgrafos, cada subgrafo está associado a um subconjunto da partição de Γ, aos quais estes caminhos estão alocados. Cada subgrafo pode possuir ao final da geração da solução alguns arcos ainda não utilizados. A Figura 3.3 mostra um exemplo onde temos um grafo original G e três subgrafos (G_1, G_2 e G_3) com os arcos que ainda não foram utilizados, os arcos utilizados foram eliminados.

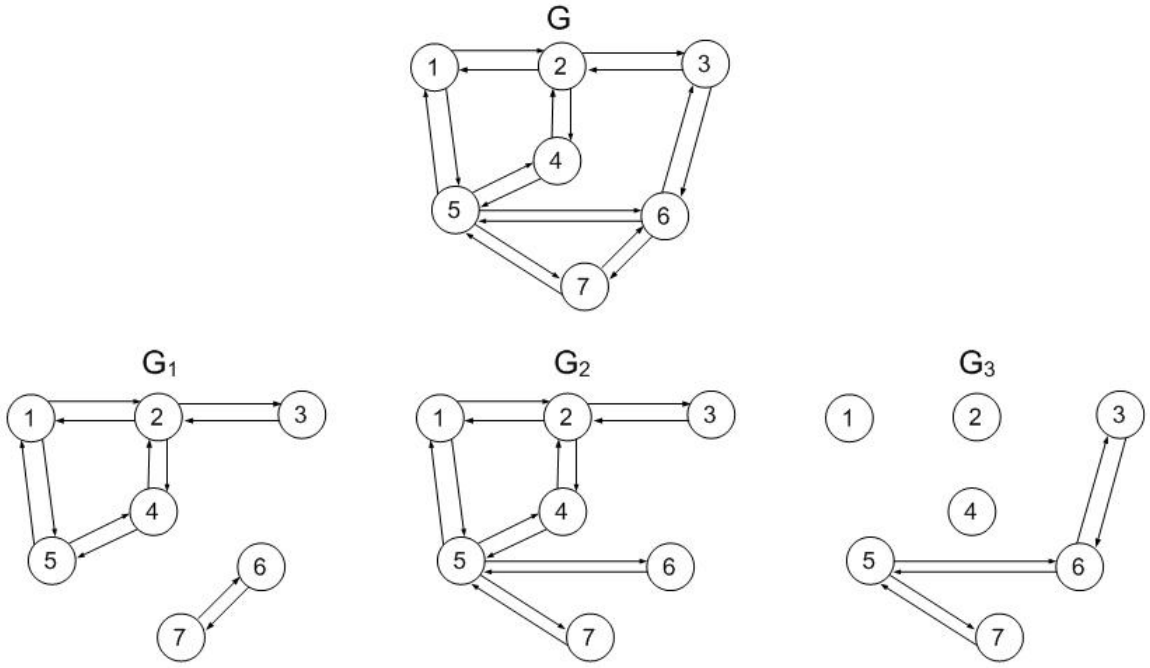


Figura 3.3: Subgrafos sem os arcos já utilizados

Primeiro selecionamos dois subgrafos da solução inicial que combinados formam o grafo mais próximo ao grafo original. O conceito de mais próximo é definido da seguinte forma: Dada uma coleção de subgrafos de $G = (V, E)$, dois subgrafos $G' = (V, E')$ e $G'' = (V, E'')$ da coleção formam o subgrafo mais próximo de G se a diferença entre a cardinalidade de E e $E' \cup E''$ é a menor possível de se obter com dois subgrafos da coleção. A Figura 3.4 exemplifica o processo. Os subgrafos são comparados par a par. No exemplo, podemos ver que G_2 individualmente é o subgrafo mais próximo do grafo original, mas quando a combinação é feita com G_1 e G_2 ou G_2 e G_3 o grafo formado não é o mais próximo ao grafo original que a combinação dos sugrafos G_1 e G_3 que formam um grafo exatamente igual ao grafo original. Sendo assim, os subgrafos selecionados neste caso seriam G_1 e G_3. Essa situação, onde dois subgrafos combinados são exatamente iguais ao grafo original seria a situação ideal. Assim poderiamos alocar todas as requisições de um dos subgrafos no outro, eliminando um subgrafo, ou seja, eliminando um comprimento de onda. No entanto, o próprio BFD não permite que tal situação ocorra. Podemos ver pela Figura 3.4 (b) que G_1 e G_3 combinados formam um grafo igual ao original. Pelo algoritmo BFD, antes de G_3 ser criado, pelo menos uma requisição não pode

ser alocada em G_1 nem em G_2, mas pela Figura 3.4 todas as requisições em G_3 podem ser alocadas em G_1, logo G_3 nunca seria criado.

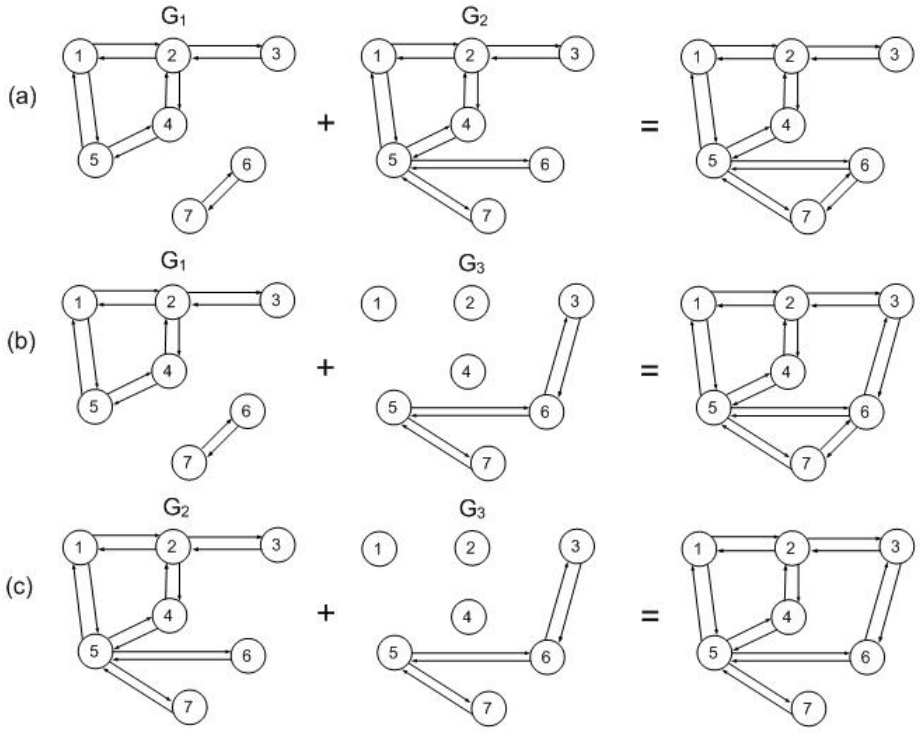


Figura 3.4: Seleção dos subgrafos para a primeira vizinhança

Assim que determinamos em quais subgrafos iremos trabalhar, o próximo passo é tentar alocar as requisições que estão nestes subgrafos em outro subgrafo qualquer, sem importar o tamanho do caminho. Com isso esperamos deixar os subgrafos selecionados mais livres, com mais arcos sobrando. Por fim, aplicamos o algoritmo BFD nas requisições que restaram nos subgrafos selecionados. Essa é a vizinhança 1-opt, onde trabalhamos com dois subgrafos. Caso a solução retornada pelo BFD seja melhor, ou seja, o método aloca todas as requisições usando menos subgrafos (comprimentos de onda) do que no início do processo, aceitamos a solução, caso contrário, iremos para a próxima vizinhança. Em cada vizinhança trabalhamos com um subgrafo a mais que a cardinalidade da vizinhança, por exemplo, para $k = 1$ o procedimento é feito sobre dois subgrafos, para $k = 2$ é feito sobre três subgrafos e assim sucessivamente. Para $k > 1$, cada novo subgrafo adicionado é aquele que apresentar o maior número de arcos restantes. Isso se repete até que o número de subgrafos utilizados no movimento atinja um limite pré-determinado, que é dado

por k_{max}.

O método BFD é usado tanto na geração da solução quanto no refinamento da solução. No refinamento o BFD é chamado sobre o subconjunto de requisições pertencentes aos subgrafos predefinidos, de acordo com a vizinhança explorada. Caso o número de subgrafos necessários para acomodar este subconjunto de requisições seja menor ao fim da chamada do método, isto significa que houve uma melhora e a solução é atualizada.

3.2.2 Método Híbrido VND-BT

A metaheurística Busca Tabu é um procedimento de otimização local que admite soluções de piora. Em sua forma clássica, a cada iteração procura-se um ótimo local selecionando-se o melhor vizinho s' de um subconjunto V da vizinhança $\Phi(s)$ da solução corrente s. Independentemente de $f(s')$ ser melhor ou pior que $f(s)$, s' será sempre a nova solução corrente. Entretanto, apenas esse mecanismo não é suficiente para escapar de ótimos locais, uma vez que pode haver retorno a uma solução previamente gerada. Para evitar isso, o algoritmo usa o conceito de lista tabu. Esta lista define todos os movimentos que têm um certo atributo como sendo tabu por um determinado número de iterações, conhecido como tempo tabu. Tais movimentos são proibidos a menos que a solução satisfaça a um certo critério de aspiração A, em geral que essa solução seja melhor que a melhor solução encontrada até então. Os atributos são escolhidos para prevenir o retorno a soluções visitadas recentemente e são escolhidos por características que são fáceis para detectar. O pseudocódigo genérico do algoritmo é apresentado pela Figura 3.5. Detalhes adicionais desse algoritmo podem ser encontrados em [22, 21].

A Busca Tabu apresentada em [45] foi adaptada para fazer parte de um procedimento híbrido para a resolução do RWA. O método híbrido age da seguinte maneira: ao se esgotarem as vizinhanças exploradas pelo VND apresentado na Seção 3.2.1 temos um subconjunto de requisições alocados em um subconjunto de subgrafos (Comprimentos de Onda). Para cada requisição já temos nesse ponto dois caminhos, um gerado durante a construção da solução inicial e outro gerado na última

```
procedimento BuscaTabu(f(.),N(.),s)
1  Iter ← 0;
2  MelhorIter ← 0;
3  T ← ∅;        {Lista Tabu}
4  Inicialize a função de aspiração A;
5  s* ← s;       {Melhor Solução até então}
6  enquanto Critério de parada não satisfeito faça
7      Iter ← Iter + 1;
8      Seja s' o melhor vizinho que não seja Tabu ou atenda ao critério de aspiração;
9      Atualize a lista tabu T;
10     s ← s';
11     se f(s) < f(s*) então
12         s* ← s;
13         MelhorIter ← Iter;
14     fim-se
15     Atualize a função de aspiração A;
16 fim-enquanto
17 Retorne s*;
fim BuscaTabu;
```

Figura 3.5: Algoritmo Busca Tabu

iteração do VND. Assim esses dois caminhos de cada requisição são passados para a Busca Tabu. A partir disto o funcionamento da Busca Tabu é exatamente igual ao apresentado em [45]. No fim, se o número de cores retornadas pela Busca Tabu for menor que a cardinalidade do subconjunto de subgrafos passado ao método significa que houve uma melhora e a solução é atualizada. Cada iteração do método híbrido consiste em uma fase VND e uma chamada na Busca Tabu ao fim do VND.

3.3 Abordagem VND-ILS com foco nas requisições

Dizemos que o foco desta abordagem são as requisições pois o objetivo inicial é o de transferir as requisições, uma a uma, de um determinado conjunto até que este conjunto esteja vazio. Assim, mesmo que um movimento seja de sucesso, ou seja, a requisição seja transferida, não podemos garantir que ele causará um impacto positivo na função objetivo do RWA. Uma outra característica desta metodologia é que ao escolhermos determinada requisição só passamos a outra quando esta foi transferida para outro conjunto, não importando o número de iterações utilizadas

para que isso ocorra.

Dada uma solução viável, a fase VND aplica três tipos de movimentos para rearranjar as requisições entre os subgrafos associados aos subconjuntos Γ no intuito de eliminar um deles. Quando o VND falha, utilizamos um procedimento para perturbar a solução corrente visando encontrar uma nova distribuição de requisições entre os subconjuntos da partição Γ. Assim, uma iteração do algoritmo alterna entre uma fase VND e uma perturbação. O VND é usado como a busca local de um procedimento ILS (*Iterated Local Search*).

3.3.1 Descida em Vizinhança Variável

Propomos aqui um algoritmo VND com $k_{max} = 3$. Seja $\Gamma_{\bar{w}}$ um subconjunto da partição que caracteriza uma solução viável. Nós consideramos duas alternativas para escolher $\Gamma_{\bar{w}}$. Na primeira alternativa, denotada por VND_r, $\Gamma_{\bar{w}}$ é o subconjunto da partição com o menor número de requisições, isto é, $\bar{w} = \arg\min\{|\Gamma_w| : w = 1, \ldots, W\}$, esta é uma escolha natural já que o objetivo é de esvaziar o conjunto. Na segunda alternativa, denotada por VND_e, $\Gamma_{\bar{w}}$ é o subconjunto no qual o grafo induzido $G_{\bar{w}}$ tem o maior número de arcos, isto é, $\bar{w} = \arg\max\{|E_w| : w = 1, \ldots, W\}$. A ideia aqui é que mesmo que exista um conjunto possuindo um número menor de requisições, como este conjunto utiliza menos arcos, provavelmente ele possui um número grande de requisições com caminhos mínimos pequenos, que, teoricamente, serão mais fáceis de serem realocados. Em ambos os casos, as requisições pertencentes a $\Gamma_{\bar{w}}$ formam uma lista $L_{\bar{w}}$ ordenada em ordem decrescente segundo o tamanho do caminho mínimo em G entre os nós origem e destino. O método VND percorre $L_{\bar{w}}$ utilizando as vizinhanças Φ_1, Φ_2 e Φ_3 para tentar realocar as requisições. Se o VND for bem sucedido, o subconjunto $\Gamma_{\bar{w}}$ ficará vazio e consequentemente o número de comprimentos de onda poderá ser reduzido. Se o VND falha para alguma requisição na vizinhança Φ_3 a fase de perturbação do método é chamada.

Chamamos atenção para que um movimento de melhora, se ele acontece, ocorre somente quando a última requisição de $\Gamma_{\bar{w}}$ é realocada em outro subconjunto, enquanto para $|\Gamma_{\bar{w}}| \geq 2$ os movimentos não reduzem o valor da função objetivo.

Quando o VND consegue esvaziar $\Gamma_{\bar{w}}$, nós atualizamos os subconjuntos restantes com respeito ao critério para selecionar o próximo subconjunto no qual o VND será aplicado, de acordo com as alternativas VND_r ou VND_e.

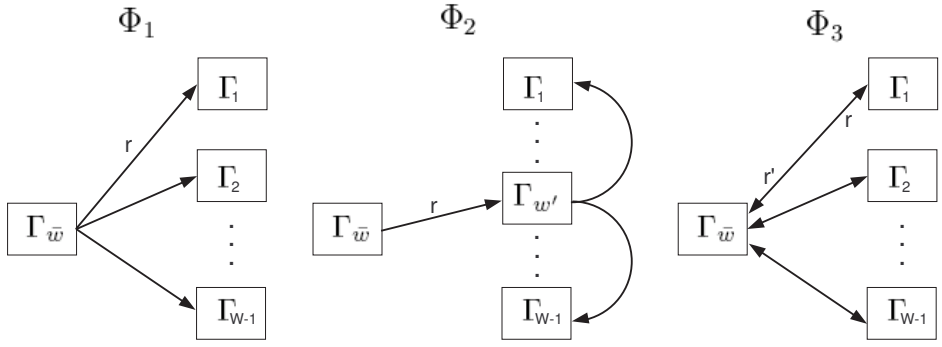
Suponha que a busca considere uma requisição r. Um movimento na vizinhança Φ_1 tenta realocar r em outro subconjunto de Γ. Se a vizinhança Φ_1 não é vazia com respeito a uma partição e uma requisição, então existe $\Gamma_{w'} \subseteq \Gamma$, $w' \neq \bar{w}$, no qual o grafo induzido $G_{w'}$ tem um caminho entre s_r e d_r. Neste caso a busca em Φ_1 é interrompida quando o primeiro grafo $G_{w'}$ para acomodar r é encontrado, e r é transferido de $\Gamma_{\bar{w}}$ para $\Gamma_{w'}$. A busca então continua com a próxima requisição em $L_{\bar{w}}$, ou o número de comprimentos de onda é reduzido se r foi a última requisição em $L_{\bar{w}}$. Se Φ_1 é vazio, o VND passa para a vizinhança Φ_2 com a mesma requisição r sob análise. Para verificar a existência de um caminho em cada conjunto utilizamos um algoritmo de busca em largura de ordem $O(|E| + |V|)$ [13], logo a complexidade total deste movimento é de $O((|E| + |V|) \cdot (W - 1))$.

Na vizinhança Φ_2 o primeiro passo é tentar arrumar um novo espaço para r no grafo $G_{w'}$ transferindo o máximo de requisições possíveis de $\Gamma_{w'}$ para outros subconjuntos de Γ diferentes de $\Gamma_{\bar{w}}$. Se é possível realocar ao menos uma requisição de $\Gamma_{w'}$, então podemos tentar alocar r neste subconjunto. Em caso de sucesso, passamos para a próxima requisição em $L_{\bar{w}}$, se existir, através da vizinhança Φ_1. Caso contrário, passamos para a vizinhança Φ_3 ainda com a mesma requisição r sob análise. Assuma que os subconjuntos da partição foram construídos em uma dada ordem, $w = 1, \ldots, W$. Em cada nova busca através de Φ_2 somente uma partição é selecionada, de acordo com a ordem de criação dos subconjuntos. Isto é, se a busca falha, nós não tentamos aplicar um movimento considerando um outro grafo, mas passamos à vizinhança Φ_3. Então, na próxima chamada à vizinhança Φ_2 o grafo a ser considerado será $G_{w'+1}$ (assumindo $\bar{w} \neq w' + 1$), se $w' + 1 > W$, recomeçamos em G_1. Como devemos verificar a existência de um caminho para cada requisição de $\Gamma_{w'}$ e no pior caso iremos verificar em todos os subconjuntos, a complexidade deste movimento é de $O((|\Gamma_{w'}| \cdot (|E| + |V|)) \cdot (W - 2))$.

Um movimento na vizinhança Φ_3 tenta trocar a requisição r com uma outra

requisição pertencente a um subconjunto de Γ diferente de $\Gamma_{\bar{w}}$. Dada uma requisição u, seja $\min_{spl}(u)$ o número de arcos do caminho mínimo entre s_u e d_u em G. A busca considera requisições com $\min_{spl}$ menor que $\min_{spl}(r)$, ordenadas em ordem crescente do $\min_{spl}$. Se uma requisição $r' \in \Gamma_{w'}$, tal que r pode ser roteado no subgrafo induzido em G por $\Gamma_{w'} - \{r'\}$, e r' no subgrafo induzido por $\Gamma_{\bar{w}} - \{r\}$, é encontrado, então o movimento de troca com r e r' é feito. Assim, depois de um movimento em Φ_3 nós temos $r' \in \Gamma_{\bar{w}}$ e $r \in \Gamma_{w'}$. Note que um movimento em Φ_3 não reduz $|\Gamma_{\bar{w}}|$, mas como ele troca r por r', e $\min_{spl}(r') < \min_{spl}(r)$, isto pode permitir uma realocação de r' usando Φ_1 ou Φ_2 na sequência. Se nenhuma troca é possível em Φ_3, passamos à fase de perturbação. Isto significa que o método VND falhou em esvaziar $L_{\bar{w}}$, e ele não será aplicado a outro subconjunto da partição. Ao invés disto, um procedimento é aplicado para perturbar a solução corrente antes de uma nova tentativa do VND. A complexidade do movimento é de $O(|\Gamma| \cdot (|E| + |V|))$.

A Figura 3.6 ilustra os movimentos de busca através das vizinhanças do VND. A ideia na parte 3.6a é mover uma requisição r de $\Gamma_{\bar{w}}$ para um subconjunto Γ_i, $i = 1, \ldots, W$, com $i \neq \bar{w}$. A parte 3.6b mostra a tentativa de acomodar r em $\Gamma_{w'}$ realocando o máximo possível de requisições pertencenter a $\Gamma_{w'}$. A parte 3.6c representa movimentos de troca.



(a) Aloca (b) Aloca/Realoca (c) Troca

Figura 3.6: Vizinhanças

3.3.2 Busca Local Iterativa

O método ILS (*Iterated Local Search*) emprega técnicas de perturbação para escapar de um ótimo local. Seja s^* a solução corrente, e seja inicialmente s igual a s^*, então a fase ILS tem três componentes: (i) uma busca local é aplicada em s para gerar s'; (ii) um critério de aceitação para atualizar ou não s^*; e (iii) uma técnica de perturbação aplicada em s^* para gerar uma nova solução s. A melhor solução é retornada ao fim do processo. O leitor pode se dirigir a [38] para uma discussão mais ampla sobre o método ILS.

O ILS utiliza como busca local o VND apresentado na seção anterior e quando este falha ao trocar uma requisição $r \in \Gamma_{\bar{w}}$ através de Φ_3 uma perturbação é feita. Note que neste ponto o VND pode já ter feito alguns movimentos. Nós não desfazemos estes movimentos feitos. A perturbação recai sobre um problema de atribuição construído para rearranjar as requisições entre os subconjuntos que definem a partição dos elementos de $\Gamma - \Gamma_{\bar{w}}$. Nós escolhemos aleatoriamente uma requisição r' em cada subconjunto $\Gamma_{w'}$, $\Gamma_{w'} \subseteq \Gamma - \Gamma_{\bar{w}}$, da partição.

Seja $G^a_{w'} = (V, E^a_{w'})$ o subgrafo de G onde $E^a_{w'} = E - \cup_{r \in \Gamma_{w'} - \{r'\}} E_r$ (os arcos usados para rotear r' são reativados em $G^a_{w'}$). A perturbação é feita para encontrar a melhor maneira de atribuir $W - 1$ requisições em $W - 1$ subconjuntos.

O custo de atribuir a requisição $i \in \Gamma_{w^i}$ ao subconjunto Γ_j é dado como segue:

- $c_{ij} = 2$, se $j = w^i$; {Custo de retornar a requisição ao conjunto de origem}
- $c_{ij} = \infty$, se não existe caminho para rotear a requisição i no grafo G^a_j;
- $c_{ij} = 1 - \frac{\min_{spl}(i)}{\min_{spl}(i)(G^a_j)} - \max\{\min_{spl}(i)(G^a_i) - \min_{spl}(G^a_j\}, 0)$, caso contrário,

onde $\min_{spl}(i)(G')$ é o número de arcos do caminho mínimo entre s_i e d_i no grafo G' e $\min_{spl}(i)$ é o número de arcos do caminho mínimo da requisição i no grafo G original. Depois de resolver o problema de atribuição, verificamos se a alocação das requisições no subconjunto da partição $\Gamma - \Gamma_{\bar{w}}$ foi alterada. Se sim, o método volta à fase VND tentando o movimento com $r \in \Gamma_{\bar{w}}$ através de Φ_1. Caso contrário, a perturbação é feita novamente.

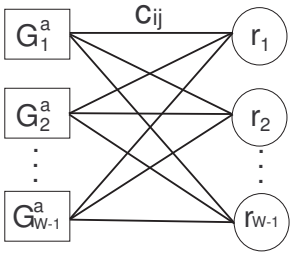


Figura 3.7: Perturbação ILS por meio de um problema de atribuição

A Figura 3.7 ilustra o problema de atribuição construído para servir de base para a perturbação ILS. Note que o subgrafo gerado pelo subconjunto $\Gamma_{\bar{w}}$ não faz parte do problema.

Nós propomos duas variações para escolher a requisição r' em cada subconjunto $\Gamma_{w'}$ para construir o problema de atribuição. Na primeira, denotada por ILS_p, todas as requisições pertencentes a $\Gamma_{w'}$ têm a mesma probabilidade de serem escolhidas. Na segunda, denotada por ILS_{5p}, as requisições que possuem caminhos que passam através da origem s_r ou destino d_r da requisição $r \in \Gamma_{\bar{w}}$ na qual o VND falhou ao fazer o movimento através de Φ_3 têm uma maior probabilidade de ser escolhida. De fato, se o caminho usado para rotear a requisição $r' \in \Gamma_{w'}$ em G tem um arco saindo (resp. entrando) de s_r (resp. em d_r), então r' tem cinco vezes mais chances de ser escolhida.

3.4 Resultados Computacionais

Nesta seção serão apresentados os testes comparando os métodos desenvolvidos, o VND-BFD, VND-BT, e as variações do VND-ILS apresentadas na Seção 3.3.

Os testes foram estruturados da seguinte forma, primeiramente uma série de experimentos apresentam os resultados dos métodos propostos com o foco na eliminação de um ou mais comprimentos de onda com um movimento apenas, ou seja, o VND-BFD e o VND-BT apresentados nas seções 3.2.1 e 3.2.2. Depois apresentamos resultados para a outra alternativa em que o foco passa a ser a transferência de requisições, uma a uma, ou seja, as variações do VND-ILS apresentado na seção 3.3. Por fim, apresentamos os métodos que apresentaram os melhores resultados e

fazendo uma comparação com a literatura recente.

Para a realização dos testes foram utilizados três conjuntos de instâncias. O primeiro conjunto, apresentado na Tabela 3.1 são instâncias encontradas na literatura, disponíveis na web[1] ou disponibilizadas por Noronha[2], e são conhecidas por se tratarem de instâncias realísticas. As instâncias marcadas com (*) tiveram a matriz de demandas gerada aleatoriamente de maneira a formar instâncias assimétricas. A Figura 3.8 apresenta a topologia física de uma destas instâncias, a instância Finland.

Os outros dois conjuntos, são as instâncias propostas em [43]. O conjunto de instâncias Z foi gerado aleatoriamente e foi criado baseado em grids $n \times m$ onde cada vértice tem uma ligação com os quatro vértices mais próximos. Foram criadas cinco redes com aproximadamente 100 vértices (10×10, 8×13, 6×17, 5×20, 4×25). Para cada rede foram geradas também aleatoriamente cinco matrizes de tráfego com probabilidades P_l igual a 0.2, 0.4, 0.6, 0.8, ou 1.0 de existir uma requisição entre um par de vértices. A Figura 3.9 apresenta um exemplo de instância de 12 nós criada a partir de um grid 3×4.

As instâncias do conjunto Y também foram geradas aleatoriamente e todas têm exatamente 100 vértices. A probabilidade P_e de existir uma ligação entre um par de vértices é igual a 0.03, 0.04, ou 0.05, e a probabilidade P_l de existir uma requisição entre um par de vértices é igual 0.2, 0.4, 0.6, 0.8, ou 1.0. Os vértices na rede com P_e igual a 0.04 ou 0.05 têm grau no mínimo igual a 2. O diâmetro das redes com P_e igual a 0.05, 0.04 e 0.03 são, respectivamente, 5, 6 e 7.

Os testes foram realizados da seguinte maneira: Cada instância foi resolvida por cada um dos métodos 5 vezes partindo-se de sementes aleatórias diferentes. O tempo máximo para cada método foi de 5 minutos.

Os algoritmos foram implementados na linguagem C++ usando o compilador Borland C++ Builder 6.0 e testados em um microcomputador Intel Core 2 Duo, 2.54 GHz, com 4 GB de RAM sob sistema operacional Windows XP.

[1] http://dag.cs.uni-kl.de/research/rwa/

[2] comunicação pessoal

Tabela 3.1: Instâncias Realísticas

Instância	Vértices	Arestas	Conexões
Atlanta20	15	22	13680
ATT	90	137	359
ATT2	71	175	2918
Brasil	27	70	1370
Cost266*	37	57	6543
Dfn-bwin*	10	45	4840
Dfn-gwin*	11	47	3771
EON	20	39	368
Finland	31	51	930
France*	25	45	15398
Germany50	50	88	4730
Giul	39	86	14732
Janos-us*	26	42	3262
Nobel-eu	28	41	3796
Nobel-germany	17	26	1320
Norway	27	51	10696
NSF.1	14	21	284
NSF.3	14	22	258
NSF.12	14	21	551
NSF.48	14	21	547
NSF2.1	14	22	284
NSF2.3	14	22	285
NSF2.12	14	22	551
NSF2.48	14	22	547
Sun	27	51	952

3.4.1 Resultados para o VND-BFD e VND-BT

A Tabela 3.2 apresenta os resultados da comparação entre o VND-BFD e o método híbrido VND-BT para as instâncias realísticas. A primeira coluna apresenta a instância, para cada método apresentamos a melhor solução encontrada nas cinco execuções e o *gap* médio com relação ao limite inferior (limite obtido através da relaxação do modelo apresentado por [4]), a última coluna apresentamos o limite inferior. Podemos observar que para as instâncias realísticas os métodos obtêm um resultado praticamente igual. O VND-BT apresenta um resultado melhor para as instâncias ATT2 e Finland, enquanto o VND-BFD consegue melhor resultado na instância NSF.3. Nas demais instâncias os métodos apresentam resultados exatamente iguais.

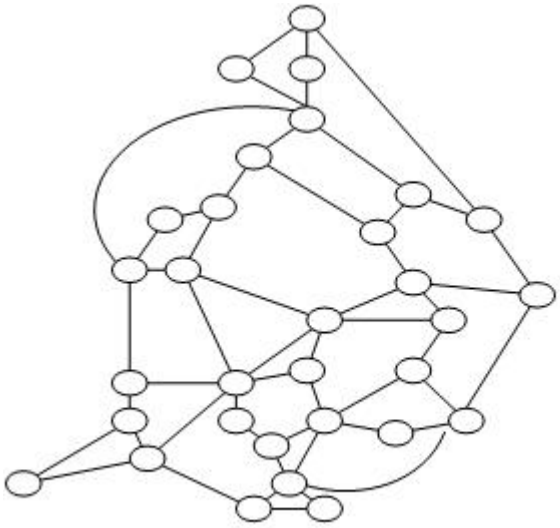


Figura 3.8: Instância Finland

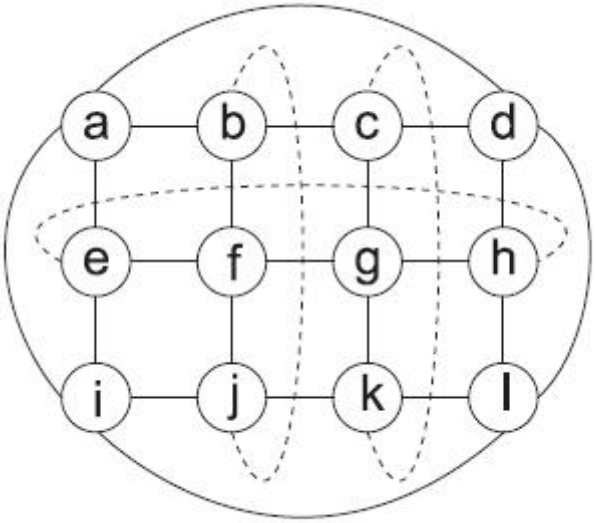


Figura 3.9: Exemplo de Instância (Rede Z) - [43]

Para as instâncias do conjunto Z os resultados são apresentados na Tabela 3.3. Para este conjunto de instâncias também podemos observar bastante equilíbrio entre os métodos, mas o VND-BFD consegue obter melhores resultados em algumas instâncias, com destaque para a instância z.10x10.40 onde o VND-BFD tem um desvio médio de 9.80% contra 11.37% do VND-BT.

Por fim, a Tabela 3.4 apresenta a comparação entre o VND-BFD e o VND-BT para as instâncias do conjunto Y. Novamente podemos observar que os resultados são bem semelhantes. O VND-BFD é capaz de encontrar limites melhores para duas

instâncias, y.3.60.5 e y.4.100.1 enquanto o VND-BT é capaz de encontrar um limite melhor para a instância y.3.80.5.

Tabela 3.2: Resultados VND-BFD X VND-BT - Instâncias Realísticas

Métodos	VND-BFD		VND-BT		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
ATT	23	18.0	23	18.0	20
ATT2	113	0.7	113	0.0	113
Brasil	48	0.0	48	0.0	48
EON	22	0.0	22	0.0	22
Finland	47	2.2	46	1.7	46
NSF.3	22	2.7	23	4.6	22
NSF.12	39	2.6	39	2.6	38
NSF2.12	35	0.0	35	0.0	35
Média		3.3		3.4	

Tabela 3.3: Resultados VND-BFD X VND-BT - Instâncias Z

Métodos	VND-BFD		VND-BT		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
z.10x10.20	30	14.07	31	14.81	27
z.8x13.20	35	6.06	35	6.06	33
z.6x17.20	46	4.55	46	4.55	44
z.5x20.20	54	1.48	54	1.48	54
z.4x25.20	68	3.03	68	3.03	66
z.10x10.40	56	9.80	56	11.37	51
z.8x13.40	66	4.76	66	4.76	63
z.6x17.40	86	2.38	86	2.38	84
z.5x20.40	103	1.98	103	1.98	101
z.4x25.40	129	2.38	129	2.38	126
z.10x10.60	84	9.09	84	9.09	77
z.8x13.60	100	4.17	100	4.17	96
z.6x17.60	131	2.34	131	2.50	128
z.5x20.60	157	1.95	157	1.95	154
z.4x25.60	196	2.08	196	2.08	192
z.10x10.80	111	7.77	111	7.77	103
z.8x13.80	133	3.10	133	3.10	129
z.6x17.80	174	2.11	174	2.22	171
z.5x20.80	207	1.27	208	1.46	205
z.4x25.80	260	1.17	260	1.17	257
z.10x10.100	136	8.80	136	8.80	125
z.8x13.100	170	1.19	170	1.19	168
z.6x17.100	219	1.39	219	1.39	216
z.5x20.100	254	1.68	254	1.92	250
z.4x25.100	318	1.92	318	1.99	312
Média		4.02		4.14	

3.4.2 Resultados para o VND-ILS

Nesta seção apresentamos os resultados das variações para o VND-ILS. Foram testadas 4 versões do VND-ILS, combinando as variantes VND_e e VND_r com as variantes ILS_p e ILS_{5p}. Relembrando que no VND_e o subconjunto a ser elimanado é o que apresenta o maior número de arcos sem utilização, no VND_r é o que apresenta o menor número de requisições; no ILS_p todas as requisições têm as mesmas chances de fazer parte do problema de atribuição, e no ILS_{5p} as requisições roteadas por arcos adjacentes a s_r e d_r têm cinco vezes mais chances. As soluções iniciais de todas as variantes são geradas pelo BFD.

A Tabela 3.5 apresenta os resultados das variações do VND-ILS para as instâncias

Tabela 3.4: Resultados VND-BFD X VND-BT - Conjunto Y

Métodos	VND-BFD		VND-BT		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
y.4.20.4	20	5.26	20	7.37	19
y.3.40.5	60	13.21	60	13.21	53
y.3.60.5	86	12.73	87	12.99	77
y.4.60.5	57	17.55	57	17.55	49
y.5.60.1	37	12.12	37	12.12	33
y.3.80.1	122	15.47	122	15.66	106
y.3.80.5	114	9.62	113	9.42	104
y.4.80.1	72	53.19	72	53.19	47
y.4.80.5	74	14.77	74	14.46	65
y.5.80.1	48	13.49	48	13.02	43
y.5.80.2	60	2.37	60	2.37	59
y.4.100.1	88	16.84	89	17.11	76
y.5.100.1	59	7.27	59	7.27	55
y.5.100.2	76	4.11	76	4.11	73
Média		14.14		14.28	

realísticas. Todas as variações foram capazes de encontrar a solução ótima para todas as instâncias em todas as execuções.

A Tabela 3.6 apresenta os resultados das variações do VND-ILS para as instâncias do conjunto Z. Podemos observar que as variações VND_e e VND_r não causam tanto impacto nas soluções, mas o ILS_{5p} apresenta resultados melhores que o ILS_p. A diferença na média das soluções do VND_e-ILS_{5p} e VND_r-ILS_{5p} é mínima, 2.80 e 2.77, no entanto o VND_e-ILS_{5p} é capaz de encontrar um limite superior melhor, no caso, para a instância z.6x17.80. Para todas as outras instâncias o limite superior foi sempre o mesmo. O VND_e-ILS_{5p} é capaz de encontrar a solução ótima para seis instâncias deste conjunto.

Nas Tabelas 3.7, 3.8 e 3.9 novamente podemos ver que o ILS_{5p} causa um impacto positivo nas soluções e o VND_e-ILS_{5p} apresenta limites melhores em quatro instâncias, y.3.20.5, y.3.60.5, y.3.100.4 e y.4.100.1. Todas as variações são capazes de encontrar a solução ótima para quarenta e cinco instâncias deste conjunto, sendo os métodos que utilizam o ILS_{5p} capazes de encontrar a solução ótima para outras três instâncias.

A partir dos resultados expostos serão feitas comparações da variante VND_e-

ILS_{5p} com outros métodos da literatura.

A Tabela 3.10 apresenta os resultados de quatro métodos, multistart-BFD, BFD-VND_e, VND_e-ILS_{5p} e EDP-VND_e-ILS_{5p}, para as instâncias realísticas. O objetivo deste teste é mostrar que a variante VND_e-ILS_{5p} é capaz de encontrar melhores soluções que o multistart-BFD. Na variante BFD-VND_e eliminamos a fase de perturbação e assim podemos ver o seu impacto. Na variante EDP-VND_e-ILS_{5p} utilizamos o EDP para a geração da solução inicial e assim podemos observar a robustez do método proposto. Para cada método executamos novamente cinco rodadas sobre cada instância, apresentando a melhor solução e o *gap* médio, com limite de tempo de 5 minutos. Como observamos anteriormente, as instâncias realísticas são mais fáceis de resolver do que as instâncias dos conjuntos Y e Z, contudo, mesmo para aquelas instâncias os métodos multistart-BFD e BFD-VND_e não foram capazes de encontrar as soluções ótimas em todos os casos, enquanto o VND_e-ILS_{5p} e EDP-VND_e-ILS_{5p} são capazes. Podemos observar que o BFD-VND_e é capaz de melhorar os resultados do multistart-BFD reduzindo a média geral de 3.27% para 2.50%, mas ainda apresenta alguns desvios bem grandes como 20% e 15.37%. No entanto, quando o ILS_{5p} é combinado ao VND_e, todas as soluções ótimas são encontradas, independentemente da solução inicial.

A Tabela 3.11 apresenta os resultados dos quatro métodos para o conjunto Z. Como podemos observar o multistart-BFD não foi capaz de encontrar nenhum dos limites inferiores e que o VND_e-ILS_{5p} foi capaz de obter desvios melhores em todas as instâncias deste conjunto. Além disso, o VND_e-ILS_{5p} foi capaz de encontrar a solução ótima para 6 instâncias. O uso do VND e o ILS é robusto, mesmo partindo de soluções iniciais diferentes como podemos ver pelo EDP-VND_e-ILS_{5p} que encontra melhores desvios em relação ao multistart-BFD para 14 instâncias.

As Tabelas 3.12 a 3.14 apresentam os resultados para as instâncias do conjunto Y. Comparando os quatro métodos eles seguem a tendência das tabelas anteriores. As versões com o ILS apresentam desvios menores que o multistart-BFD, em particular para instâncias em que o multistart-BFD não é capaz de encontrar a solução ótima. Podemos observar que o VND_e-ILS_{5p} é capaz de encontrar melhores desvios

para 42 das 75 instâncias e que o EDP-VND$_e$-ILS$_{5p}$ é capaz de encontrar melhores desvios em 35 destas 42 instâncias. O uso do BFD como solução inicial, contudo, conduz a melhores soluções como podemos ver pelos resultados do VND$_e$-ILS$_{5p}$ em comparação com o EDP-VND$_e$-ILS$_{5p}$.

Tabela 3.5: Resultados VND-ILS - Instâncias Realísticas

Métodos	VND$_e$-ILS$_p$		VND$_r$-ILS$_p$		VND$_e$-ILS$_{5p}$		VND$_r$-ILS$_{5p}$		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
Atlanta20	1256	0.0	1256	0.0	1256	0.0	1256	0.0	1256
ATT	20	0.0	20	0.0	20	0.0	20	0.0	20
ATT2	113	0.0	113	0.0	113	0.0	113	0.0	113
Brasil	48	0.0	48	0.0	48	0.0	48	0.0	48
Cost266	446	0.0	446	0.0	446	0.0	446	0.0	446
Dfn-bwin	73	0.0	73	0.0	73	0.0	73	0.0	73
Dfn-gwin	316	0.0	316	0.0	316	0.0	316	0.0	316
EON	22	0.0	22	0.0	22	0.0	22	0.0	22
Finland	46	0.0	46	0.0	46	0.0	46	0.0	46
France	946	0.0	946	0.0	946	0.0	946	0.0	946
Germany50	147	0.0	147	0.0	147	0.0	147	0.0	147
Giul	379	0.0	379	0.0	379	0.0	379	0.0	379
Janos-us	207	0.0	207	0.0	207	0.0	207	0.0	207
Nobel-eu	214	0.0	214	0.0	214	0.0	214	0.0	214
Nobel-germany	78	0.0	78	0.0	78	0.0	78	0.0	78
Nobel-us	39	0.0	39	0.0	39	0.0	39	0.0	39
Norway	274	0.0	274	0.0	274	0.0	274	0.0	274
NSF.1	22	0.0	22	0.0	22	0.0	22	0.0	22
NSF.3	22	0.0	22	0.0	22	0.0	22	0.0	22
NSF.12	38	0.0	38	0.0	38	0.0	38	0.0	38
NSF.48	41	0.0	41	0.0	41	0.0	41	0.0	41
NSF2.1	21	0.0	21	0.0	21	0.0	21	0.0	21
NSF2.3	21	0.0	21	0.0	21	0.0	21	0.0	21
NSF2.12	35	0.0	35	0.0	35	0.0	35	0.0	35
NSF2.48	39	0.0	39	0.0	39	0.0	39	0.0	39
Sun	59	0.0	59	0.0	59	0.0	59	0.0	59
Média		0.0		0.0		0.0		0.0	

Tabela 3.6: Resultados VND-ILS - Conjunto Z

Métodos	$\text{VND}_e\text{-ILS}_p$		$\text{VND}_r\text{-ILS}_p$		$\text{VND}_e\text{-ILS}_{5p}$		$\text{VND}_r\text{-ILS}_{5p}$		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
z.10x10.20	29	7.41	29	7.41	29	7.41	29	7.41	27
z.8x13.20	34	3.03	34	3.03	34	3.03	34	3.03	33
z.6x17.20	45	2.27	44	1.82	44	1.36	44	0.91	44
z.5x20.20	54	0.00	54	0.00	54	0.00	54	0.00	54
z.4x25.20	66	0.91	66	0.61	66	0.00	66	0.61	66
z.10x10.40	56	9.80	56	9.80	55	8.63	55	8.24	51
z.8x13.40	65	3.17	65	3.17	64	2.54	64	2.22	63
z.6x17.40	85	1.43	85	1.19	85	1.19	85	1.19	84
z.5x20.40	102	0.99	101	0.79	101	0.59	101	0.40	101
z.4x25.40	127	0.95	127	0.79	127	0.79	127	0.79	126
z.10x10.60	85	10.39	85	10.39	84	9.09	84	9.35	77
z.8x13.60	99	3.13	98	2.92	98	2.08	98	2.08	96
z.6x17.60	129	0.78	129	0.94	129	0.78	129	0.78	128
z.5x20.60	155	0.65	155	0.65	154	0.26	154	0.39	154
z.4x25.60	193	0.62	193	0.52	193	0.52	193	0.52	192
z.10x10.80	113	9.90	113	9.90	112	9.51	112	9.13	103
z.8x13.80	131	1.55	131	1.55	130	1.09	130	1.24	129
z.6x17.80	172	0.58	172	0.58	171	0.47	172	0.58	171
z.5x20.80	206	0.49	206	0.49	206	0.49	206	0.49	205
z.4x25.80	259	0.78	258	0.62	258	0.47	258	0.54	257
z.10x10.100	140	12.00	140	12.00	139	11.52	139	11.36	125
z.8x13.100	174	3.57	174	3.57	173	3.21	173	3.33	168
z.6x17.100	220	1.94	220	1.94	220	1.85	220	1.85	216
z.5x20.100	253	1.52	253	1.52	253	1.52	253	1.28	250
z.4x25.100	317	1.73	317	1.73	317	1.60	317	1.60	312
Média		3.18		3.12		2.80		2.77	

Tabela 3.7: Resultados VND-ILS - Conjunto Y3

Métodos	VND$_e$-ILS$_p$		VND$_r$-ILS$_p$		VND$_e$-ILS$_{5p}$		VND$_r$-ILS$_{5p}$		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
y.3.20.1	30	11.11	30	11.11	29	8.89	29	8.89	27
y.3.20.2	33	0.00	33	0.00	33	0.00	33	0.00	33
y.3.20.3	29	0.00	29	0.00	29	0.00	29	0.00	29
y.3.20.4	28	10.00	28	8.46	28	8.46	28	9.23	26
y.3.20.5	29	3.57	29	3.57	28	2.86	29	3.57	28
y.3.40.1	58	9.43	57	9.06	57	7.55	57	7.55	53
y.3.40.2	59	0.00	59	0.00	59	0.00	59	0.00	59
y.3.40.3	61	0.00	61	0.00	61	0.00	61	0.00	61
y.3.40.4	54	8.80	54	9.20	54	8.00	54	8.40	50
y.3.40.5	57	7.55	57	7.55	56	6.79	56	6.42	53
y.3.60.1	87	8.15	87	8.40	87	7.41	87	7.41	81
y.3.60.2	89	0.00	89	0.00	89	0.00	89	0.00	89
y.3.60.3	91	0.00	91	0.00	91	0.00	91	0.00	91
y.3.60.4	80	3.59	81	3.85	80	2.82	80	2.56	78
y.3.60.5	83	8.05	83	8.05	82	7.53	83	7.79	77
y.3.80.1	116	9.81	116	9.43	115	9.25	115	9.06	106
y.3.80.2	117	0.00	117	0.00	117	0.00	117	0.00	117
y.3.80.3	118	0.00	118	0.00	118	0.00	118	0.00	118
y.3.80.4	107	1.90	107	1.90	106	1.14	106	1.33	105
y.3.80.5	110	5.77	110	5.96	109	4.81	109	5.00	104
y.3.100.1	144	10.08	144	10.08	143	9.31	143	9.16	131
y.3.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.3	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.4	133	1.83	133	1.83	132	1.22	133	1.53	131
y.3.100.5	137	6.51	137	6.51	136	5.58	136	5.74	129
Média		4.25		4.20		3.66		3.75	

3.4.3 Comparações com a literatura

Nesta seção serão apresentadas algumas comparações com a literatura. As instâncias utilizadas nestes testes são um subconjunto das instâncias apresentadas na Seção 3.4.2. Serão apresentados somente os resultados de algumas instâncias que apresentaram maior dificuldade de serem resolvidas segundo estudos realizados por Noronha et al. [44]. Os testes foram realizados da seguinte maneira: Cada instância foi resolvida pelo VND$_e$-ILS$_{5p}$ 5 vezes partindo-se de sementes aleatórias diferentes. O tempo máximo para cada método foi de 5 minutos. Para fins de verificar o desvio padrão da função objetivo, cada instância foi resolvida mais 25 vezes.

O algoritmo foi implementado na linguagem C++ usando o compilador Borland

Tabela 3.8: Resultados VND-ILS - Conjunto Y4

Métodos	VND$_e$-ILS$_p$		VND$_r$-ILS$_p$		VND$_e$-ILS$_5p$		VND$_r$-ILS$_5p$		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
y.4.20.1	19	11.76	19	11.76	19	11.76	19	11.76	17
y.4.20.2	28	0.00	28	0.00	28	0.00	28	0.00	28
y.4.20.3	23	0.00	23	0.00	23	0.00	23	0.00	23
y.4.20.4	19	0.00	19	0.00	19	0.00	19	0.00	19
y.4.20.5	19	11.76	19	11.76	19	11.76	19	11.76	17
y.4.40.1	36	16.13	36	16.13	35	12.90	35	12.90	31
y.4.40.2	57	0.00	57	0.00	57	0.00	57	0.00	57
y.4.40.3	43	0.00	43	0.00	43	0.00	43	0.00	43
y.4.40.4	38	0.00	38	0.00	38	0.00	38	0.00	38
y.4.40.5	37	12.12	37	12.12	37	12.12	37	12.12	33
y.4.60.1	53	12.77	53	12.77	53	12.77	53	12.77	47
y.4.60.2	86	0.00	86	0.00	86	0.00	86	0.00	86
y.4.60.3	64	0.00	64	0.00	64	0.00	64	0.00	64
y.4.60.4	58	0.00	58	0.00	58	0.00	58	0.00	58
y.4.60.5	55	13.06	55	12.24	55	12.24	55	12.24	49
y.4.80.1	70	49.36	70	49.36	70	48.94	70	48.94	47
y.4.80.2	118	0.00	118	0.00	118	0.00	118	0.00	118
y.4.80.3	81	0.00	81	0.00	81	0.00	81	0.00	81
y.4.80.4	78	0.00	78	0.00	78	0.00	78	0.00	78
y.4.80.5	72	11.69	72	12.00	72	10.77	72	10.77	65
y.4.100.1	87	14.47	87	14.47	86	14.21	87	14.47	76
y.4.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.4.100.3	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.4	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.5	90	12.50	90	12.50	89	11.75	89	12.00	80
Média		6.62		6.60		6.37		6.39	

C++ Builder 6.0 e testado em um microcomputador Intel Core i5-2410M, 2.30 GHz, com 4 GB de RAM sob sistema operacional Windows 7.

A comparação é feita com o algoritmo genético GA-RWA proposto por Noronha et al.[44]. Os resultados do GA-RWA foram obtidos com um limite de 10 minutos sob um Pentium IV com 3.4GHz.

Os resultados dessa comparação são apresentados na Tabela 3.15. Na tabela apresentamos os resultados referentes às 5 primeiras rodadas do método para maior justiça na comparação, já que em [44] apenas 5 execuções foram feitas. Também apresentamos os resultados referentes às 30 execuções, mostrando, além do valor mínimo e a média, o desvio padrão (σ) da função objetivo e o tempo médio em que

Tabela 3.9: Resultados VND-ILS - Conjunto Y5

Métodos	VND$_e$-ILS$_p$		VND$_r$-ILS$_p$		VND$_e$-ILS$_5p$		VND$_r$-ILS$_5p$		
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	LI
y.5.20.1	13	0.00	13	0.00	13	0.00	13	0.00	13
y.5.20.2	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.3	13	8.33	12	6.67	12	5.00	12	5.00	12
y.5.20.4	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.5	15	0.00	15	0.00	15	0.00	15	0.00	15
y.5.40.1	24	1.67	24	1.67	24	0.00	24	0.00	24
y.5.40.2	31	0.00	31	0.00	31	0.00	31	0.00	31
y.5.40.3	23	4.55	23	4.55	23	4.55	23	4.55	22
y.5.40.4	33	0.00	33	0.00	33	0.00	33	0.00	33
y.5.40.5	28	0.00	28	0.00	28	0.00	28	0.00	28
y.5.60.1	36	9.09	35	8.48	35	6.67	35	7.27	33
y.5.60.2	45	0.00	45	0.00	45	0.00	45	0.00	45
y.5.60.3	34	0.00	34	0.00	34	0.00	34	0.00	34
y.5.60.4	48	0.00	48	0.00	48	0.00	48	0.00	48
y.5.60.5	40	0.00	40	0.00	40	0.00	40	0.00	40
y.5.80.1	46	8.37	46	7.91	46	6.98	46	6.98	43
y.5.80.2	59	0.34	59	0.00	59	0.00	59	0.00	59
y.5.80.3	44	3.26	44	3.26	44	2.33	44	2.33	43
y.5.80.4	63	0.00	63	0.00	63	0.00	63	0.00	63
y.5.80.5	53	0.00	53	0.00	53	0.00	53	0.00	53
y.5.100.1	57	4.36	57	4.00	57	3.64	57	3.64	55
y.5.100.2	73	1.10	73	1.10	73	0.00	73	0.00	73
y.5.100.3	55	3.77	55	3.77	54	3.40	54	3.40	53
y.5.100.4	77	0.00	77	0.00	77	0.00	77	0.00	77
y.5.100.5	66	0.00	66	0.00	66	0.00	66	0.00	66
Média		1.79		1.66		1.30		1.33	

a melhor solução foi encontrada ($\bar{t}_{best}$).

Podemos observar pela Tabela 3.15 que o VND$_e$-ILS$_{5p}$ apresenta um desempenho superior ao método GA-RWA. Para todas as instâncias em que o GA-RWA ainda não havia encontrado a solução ótima o VND$_e$-ILS$_{5p}$ consegue um limite superior melhor mesmo se olharmos somente as 5 primeiras execuções. A média do VND$_e$-ILS$_{5p}$ é exatamente igual para 5 ou 30 execuções e o maior desvio padrão observado é igual a 0.5 que mostra que as soluções encontradas pelo método apresentam pouca variação. Das 30 instâncias o método proposto consegue melhorar o limite superior conhecido em 26 e obtem os mesmos limites nas outras 4 instâncias. Além disso o VND$_e$-ILS$_{5p}$ encontra a solução ótima para 11 instâncias sendo que em 9 destas a

Tabela 3.10: Comparação com o multistart-BFD sobre as instâncias realísticas

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instâncias	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	LI
Atlanta20	1342	6.91	1284	2.37	1256	0.00	1256	0.00	1256
ATT	25	25.00	24	20.00	20	0.00	20	0.00	20
ATT2	113	0.71	113	0.71	113	0.00	113	0.00	113
Brasil	48	0.00	48	0.00	48	0.00	48	0.00	48
Cost266*	446	0.00	446	0.00	446	0.00	446	0.00	446
Dfn-bwin*	73	0.00	73	0.00	73	0.00	73	0.00	73
Dfn-gwin*	316	0.00	316	0.00	316	0.00	316	0.00	316
EON	22	0.00	22	0.00	22	0.00	22	0.00	22
Finland	47	2.17	47	2.17	46	0.00	46	0.00	46
France*	946	0.00	946	0.00	946	0.00	946	0.00	946
Germany50	169	15.37	169	15.37	147	0.00	147	0.00	147
Giul	402	6.07	401	5.91	379	0.00	379	0.00	379
Janos-us*	215	4.15	215	4.15	207	0.00	207	0.00	207
Nobel-eu	304	0.00	304	0.00	304	0.00	304	0.00	304
Nobel-germany	89	4.94	89	4.71	85	0.00	85	0.00	85
Norway	543	0.00	543	0.00	543	0.00	543	0.00	543
NSF.1	23	4.55	22	0.00	22	0.00	22	0.00	22
NSF.3	22	3.64	22	2.73	22	0.00	22	0.00	22
NSF.12	39	2.63	39	2.63	38	0.00	38	0.00	38
NSF.48	41	0.49	41	0.00	41	0.00	41	0.00	41
NSF2.1	21	0.00	21	0.00	21	0.00	21	0.00	21
NSF2.3	21	0.00	21	0.00	21	0.00	21	0.00	21
NSF2.12	35	1.71	35	0.00	35	0.00	35	0.00	35
NSF2.48	39	0.00	39	0.00	39	0.00	39	0.00	39
Sun	61	3.39	60	1.69	59	0.00	59	0.00	59
Média		3.27		2.50		0.00		0.00	

solução ótima é encontrada em todas as execuções.

Para finalizar esta seção de testes o VND$_e$-ILS$_{5p}$ foi executado 200 vezes para algumas instâncias, também variando-se a semente utilizada pelo gerador de números pseudo-aleatórios. O algoritmo foi programado para terminar sua execução quando encontrasse uma solução com um número de comprimentos de onda para cada instância dado a priori. Nas Figuras 3.10 à 3.19, cada ponto da curva associada ao algoritmo representa um par (t_i, p_i), para cada execução de $i = 1, \ldots, 200$, onde t_i é o tempo da i-ésima execução mais rápida deste algoritmo e $p_i = (i - 0.5)/200$. De acordo com a metodologia descrita em [1] e [2] (time-to-target (TTT) plots) e utilizada em vários trabalhos, veja por exemplo [17, 48, 45, 44], cada um destes gráfi-

Tabela 3.11: Comparação com o multistart-BFD sobre o conjunto Z

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instâncias	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	LI
Z.10x10.20	31	17.04	31	17.04	29	7.41	30	11.11	27
Z.8x13.20	35	8.48	35	8.48	34	3.03	34	3.03	33
Z.6x17.20	46	4.55	46	4.55	44	1.36	45	2.27	44
Z.5x20.20	55	1.85	55	1.85	54	0.00	54	0.00	54
Z.4x25.20	68	3.03	68	3.03	66	0.00	67	2.12	66
Z.10x10.40	59	15.69	58	14.51	55	8.63	57	12.16	51
Z.8x13.40	67	6.35	67	6.35	64	2.54	66	4.76	63
Z.6x17.40	87	4.29	87	3.57	85	1.19	86	2.38	84
Z.5x20.40	104	2.97	104	2.97	101	0.59	103	2.18	101
Z.4x25.40	129	2.70	129	2.38	127	0.79	129	2.38	126
Z.10x10.60	88	14.29	87	14.03	84	9.09	87	12.99	77
Z.8x13.60	101	5.63	101	5.21	98	2.08	100	4.38	96
Z.6x17.60	133	3.91	133	3.91	129	0.78	131	2.66	128
Z.5x20.60	158	2.60	158	2.60	154	0.26	158	2.60	154
Z.4x25.60	195	2.08	195	1.98	193	0.52	197	2.71	192
Z.10x10.80	116	12.62	116	12.62	112	9.51	117	13.98	103
Z.8x13.80	134	4.50	134	4.50	130	1.09	134	3.88	129
Z.6x17.80	176	3.39	176	3.27	171	0.47	175	2.57	171
Z.5x20.80	209	1.95	209	1.95	206	0.49	210	2.63	205
Z.4x25.80	261	1.56	261	1.56	258	0.47	264	2.72	257
Z.10x10.100	142	13.60	142	13.60	139	11.52	146	17.44	125
Z.8x13.100	175	4.17	175	4.17	173	3.21	178	6.31	168
Z.6x17.100	222	3.06	222	3.06	220	1.85	225	4.17	216
Z.5x20.100	256	2.56	256	2.48	253	1.52	260	4.16	250
Z.4x25.100	319	2.24	318	2.18	317	1.60	325	4.42	312
Média		5.80		5.67		2.80		5.20	

cos descreve uma aproximação da distribuição de probabilidade da variável aleatória tempo para alcançar uma solução tão boa quanto o valor alvo.

Utilizamos neste último teste três instâncias realísticas que usamos para comparar com o GA-RWA, ATT, Finland e NSF.12, todas tendo como alvo a solução ótima de cada uma das instâncias. Ainda utilizamos as instâncias y.4.20.4, y.5.80.2 e y.5.100.2 também tendo como alvo a solução ótima de cada uma das instâncias. Para finalizar utilizamos também duas instâncias do conjunto Y e duas do conjunto Z nas quais os algoritmos apresentaram os maiores desvios. No caso as instâncias y.4.80.1, y.4.100.1, z.10x10.60 e z.10x10.80, com alvos iguais a uma unidade a menos da melhor solução encontrada pelo algoritmo GA-RWA, respectivamente, 72, 89, 86

Tabela 3.12: Comparação com o multistart-BFD sobre o conjunto Y3

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instâncias	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	LI
y.3.20.1	33	22.22	33	22.22	29	8.89	30	11.11	27
y.3.20.2	33	0.00	33	0.00	33	0.00	33	0.00	33
y.3.20.3	31	6.90	31	6.90	29	0.00	29	0.00	29
y.3.20.4	31	19.23	31	19.23	28	8.46	29	11.54	26
y.3.20.5	31	10.71	31	10.71	28	2.86	29	3.57	28
y.3.40.1	62	18.49	62	18.11	57	7.55	59	11.32	53
y.3.40.2	59	0.00	59	0.00	59	0.00	59	0.00	59
y.3.40.3	61	0.00	61	0.00	61	0.00	61	0.00	61
y.3.40.4	58	16.00	58	16.00	54	8.00	56	12.00	50
y.3.40.5	60	13.21	60	13.21	56	6.79	58	9.43	53
y.3.60.1	93	14.81	93	14.81	87	7.41	90	12.35	81
y.3.60.2	89	0.00	89	0.00	89	0.00	89	0.00	89
y.3.60.3	91	0.00	91	0.00	91	0.00	91	0.00	91
y.3.60.4	85	9.23	85	8.97	80	2.82	83	6.41	78
y.3.60.5	86	12.73	86	12.73	82	7.53	85	11.69	77
y.3.80.1	123	16.04	123	16.04	115	9.25	122	15.09	106
y.3.80.2	117	0.00	117	0.00	117	0.00	117	0.00	117
y.3.80.3	118	0.17	118	0.17	118	0.00	118	0.00	118
y.3.80.4	112	6.67	111	6.48	106	1.14	110	5.71	105
y.3.80.5	114	9.62	114	9.62	109	4.81	114	9.62	104
y.3.100.1	151	15.27	151	15.57	143	9.31	152	16.03	131
y.3.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.3	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.4	138	5.80	138	5.80	132	1.22	139	6.87	131
y.3.100.5	141	9.30	141	9.30	136	5.58	143	10.85	129
Média		8.26		8.23		3.66		6.14	

e 114.

Para as instâncias realísticas, Figuras 3.10, 3.11 e 3.12, podemos ver que o VND$_e$-ILS$_{5p}$ não apresenta dificuldade em encontrar a solução alvo para nenhuma das instâncias, sendo as soluções das instâncias ATT e Finland encontradas em menos de 10 segundos e para a instância NSF.12 os piores casos não passam de 2 segundos.

Para as instâncias y.4.20.4, y.5.80.2 e y.5.100.2, que também tiveram como alvo a solução ótima, podemos ver pelas Figuras 3.13 a 3.15 que apenas a última instância não foi resolvida todas as vezes em menos de 5 minutos. Contudo, vemos pela Figura 3.15 que mais de 90% das rodadas a solução ótima foi encontrada antes deste tempo. A instância y.4.20.4 é sempre resolvida com menos de 1 minuto como podemos ver

Tabela 3.13: Comparação com o multistart-BFD sobre o conjunto Y4

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instâncias	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	λ_{min}	$\overline{gap}$(%)	LI
y.4.20.1	21	23.53	21	23.53	19	11.76	19	11.76	17
y.4.20.2	28	0.00	28	0.00	28	0.00	28	0.00	28
y.4.20.3	23	0.00	23	0.00	23	0.00	23	0.00	23
y.4.20.4	20	8.42	20	7.37	19	0.00	19	0.00	19
y.4.20.5	21	23.53	21	23.53	19	11.76	19	15.29	17
y.4.40.1	38	22.58	38	22.58	35	12.90	36	16.13	31
y.4.40.2	57	0.00	57	0.00	57	0.00	57	0.00	57
y.4.40.3	43	0.00	43	0.00	43	0.00	43	0.00	43
y.4.40.4	38	0.00	38	0.00	38	0.00	38	0.00	38
y.4.40.5	40	21.21	40	21.21	37	12.12	38	15.15	33
y.4.60.1	56	19.15	56	19.15	53	12.77	54	14.89	47
y.4.60.2	86	0.00	86	0.00	86	0.00	86	0.00	86
y.4.60.3	64	0.00	64	0.00	64	0.00	64	0.00	64
y.4.60.4	58	0.00	58	0.00	58	0.00	58	0.00	58
y.4.60.5	58	19.18	58	18.78	55	12.24	57	16.33	49
y.4.80.1	73	57.02	73	56.60	70	48.94	72	54.04	47
y.4.80.2	118	0.00	118	0.00	118	0.00	118	0.00	118
y.4.80.3	81	0.00	81	0.00	81	0.00	81	0.00	81
y.4.80.4	78	0.00	78	0.00	78	0.00	78	0.00	78
y.4.80.5	76	16.92	76	16.92	72	10.77	75	15.38	65
y.4.100.1	91	19.74	91	19.74	86	14.21	90	19.47	76
y.4.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.4.100.3	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.4	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.5	93	16.50	93	16.50	89	11.75	93	17.25	80
Média		9.91		9.84		6.37		7.83	

pela Figura 3.13, enquanto na instância y.5.80.2 apenas em uma rodada uma rodada (execução) a solução ótima é encontrada acima de 200 segundos.

Para as instâncias y.4.80.1, y.4.100.1, z.10x10.60 e z.10x10.80 em que verificamos o tempo gasto para encontrar uma solução com um comprimento de onda a menos do que a melhor solução encontrada pelo GA-RWA podemos observar pelas Figuras 3.16, 3.17, 3.18 e 3.19 que isto é quase sempre realizado antes de 100 segundos. Apenas para a instância z.10x10.80 em pouco menos de 10% das execuções as soluções alvo são encontradas após 100 segundos.

Tabela 3.14: Comparação com o multistart-BFD sobre o conjunto Y5

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instâncias	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	LI
y.5.20.1	14	7.69	14	7.69	13	0.00	13	0.00	13
y.5.20.2	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.3	13	8.33	13	8.33	12	5.00	13	8.33	12
y.5.20.4	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.5	15	0.00	15	0.00	15	0.00	15	0.00	15
y.5.40.1	25	4.17	25	4.17	24	0.00	24	0.83	24
y.5.40.2	31	1.29	31	1.29	31	0.00	31	0.00	31
y.5.40.3	24	9.09	24	9.09	23	4.55	23	6.36	22
y.5.40.4	33	0.00	33	0.00	33	0.00	33	0.00	33
y.5.40.5	28	0.00	28	0.00	28	0.00	28	0.00	28
y.5.60.1	36	11.52	36	11.52	35	6.67	36	9.09	33
y.5.60.2	45	1.33	45	0.89	45	0.00	45	0.00	45
y.5.60.3	35	2.94	35	2.94	34	0.00	34	0.00	34
y.5.60.4	48	0.00	48	0.00	48	0.00	48	0.00	48
y.5.60.5	40	0.00	40	0.00	40	0.00	40	0.00	40
y.5.80.1	47	11.16	47	11.16	46	6.98	47	9.30	43
y.5.80.2	60	1.69	60	1.69	59	0.00	59	1.02	59
y.5.80.3	45	4.65	45	4.65	44	2.33	45	5.12	43
y.5.80.4	63	0.00	63	0.00	63	0.00	63	0.00	63
y.5.80.5	53	0.00	53	0.00	53	0.00	53	0.00	53
y.5.100.1	59	7.27	59	7.27	57	3.64	58	5.82	55
y.5.100.2	75	2.74	75	2.74	73	0.00	74	1.37	73
y.5.100.3	56	5.66	56	5.66	54	3.40	56	5.66	53
y.5.100.4	77	0.00	77	0.00	77	0.00	77	0.00	77
y.5.100.5	66	0.00	66	0.00	66	0.00	66	0.00	66
Média		3.18		3.16		1.30		2.12	

3.4.4 Análise de componentes VND$_e$-ILS$_{5p}$

Nesta seção apresentaremos algumas análises feitas sobre o impacto das componentes isoladas do método VND$_e$-ILS$_{5p}$. Todas as análises foram feitas utilizando as mesmas instâncias usadas para a comparação com o método GA-RWA [44]. Estas instâncias, além de se mostrarem as mais difíceis de serem resolvidas, são bem representativas.

A primeira análise executada foi com respeito à dependência do método sobre cada uma das vizinhanças. Realizamos cinco experimentos, utilizando somente a vizinhança Φ_1, somente a vizinhança Φ_2, e as combinações $\Phi_1 + \Phi_2$, $\Phi_1 + \Phi_3$ e $\Phi_2 + \Phi_3$. A vizinhança Φ_3 não foi testada isoladamente já que ela não é capaz de mudar a cardinalidade do comprimento de onda que se deseja eliminar, pois nesta

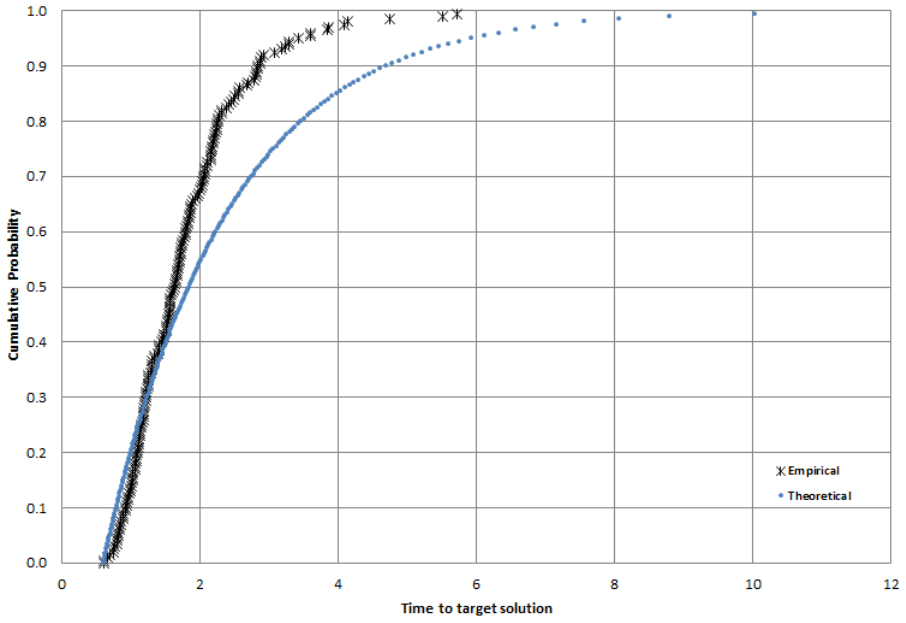


Figura 3.10: Gráfico para instância Att - Alvo 20

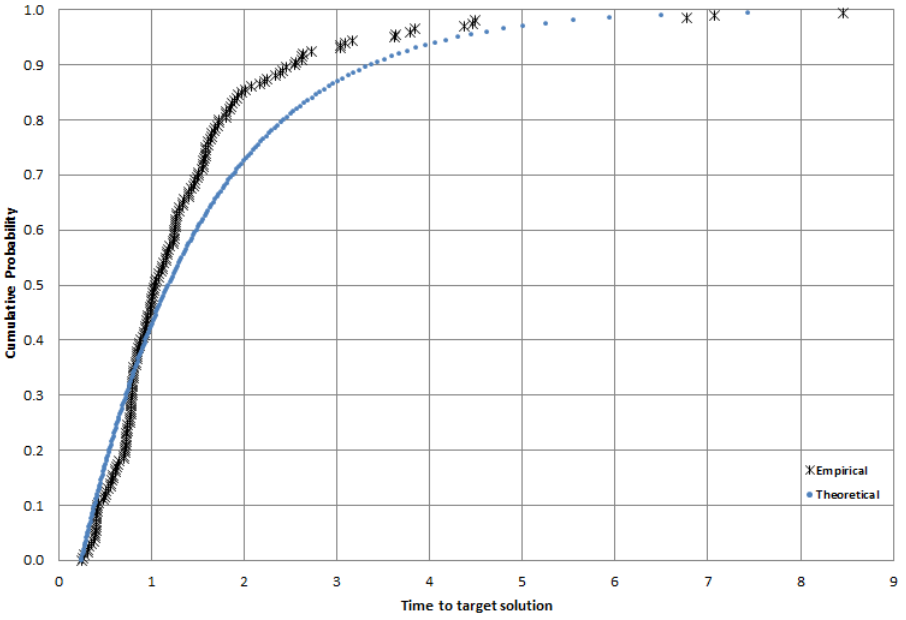


Figura 3.11: Gráfico para instância Finland - Alvo 46

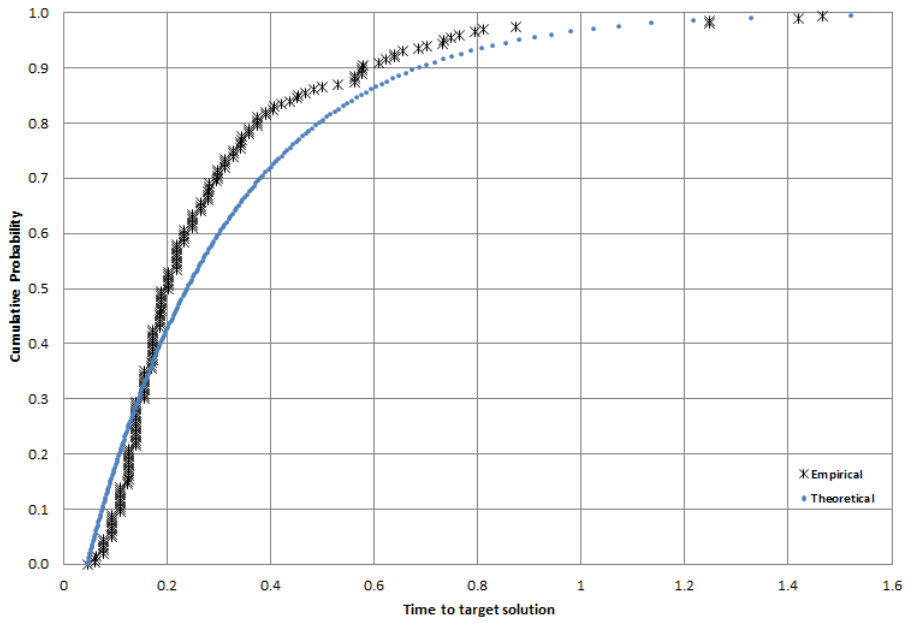


Figura 3.12: Gráfico para instância NSF.12 - Alvo 38

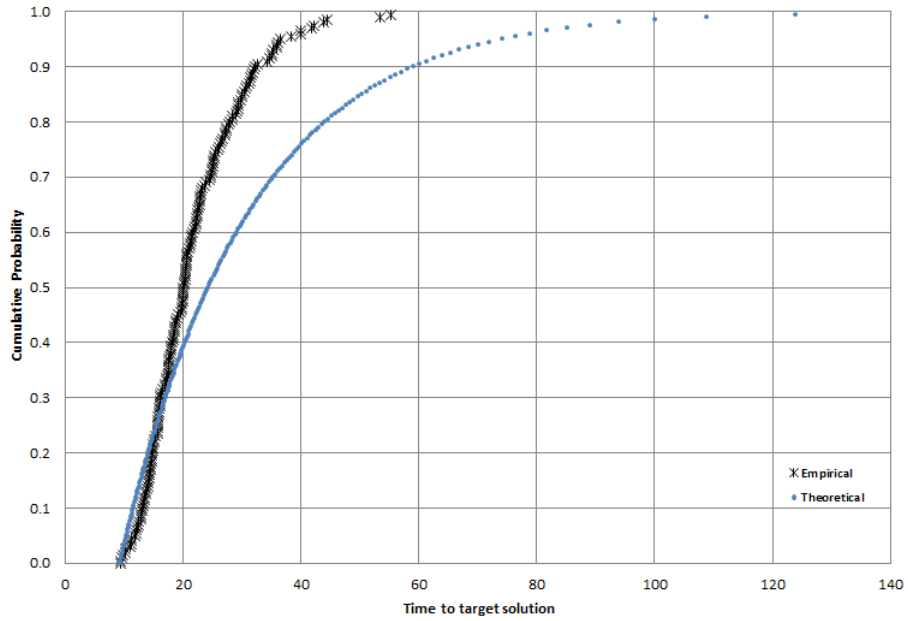


Figura 3.13: Gráfico para instância y.4.20.4 - Alvo 19

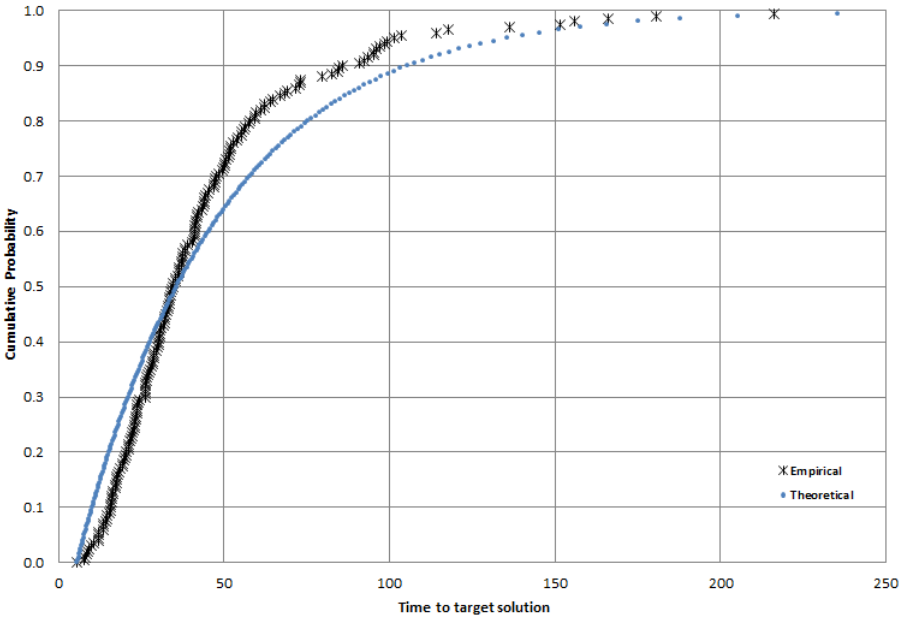


Figura 3.14: Gráfico para instância y.5.80.2 - Alvo 59

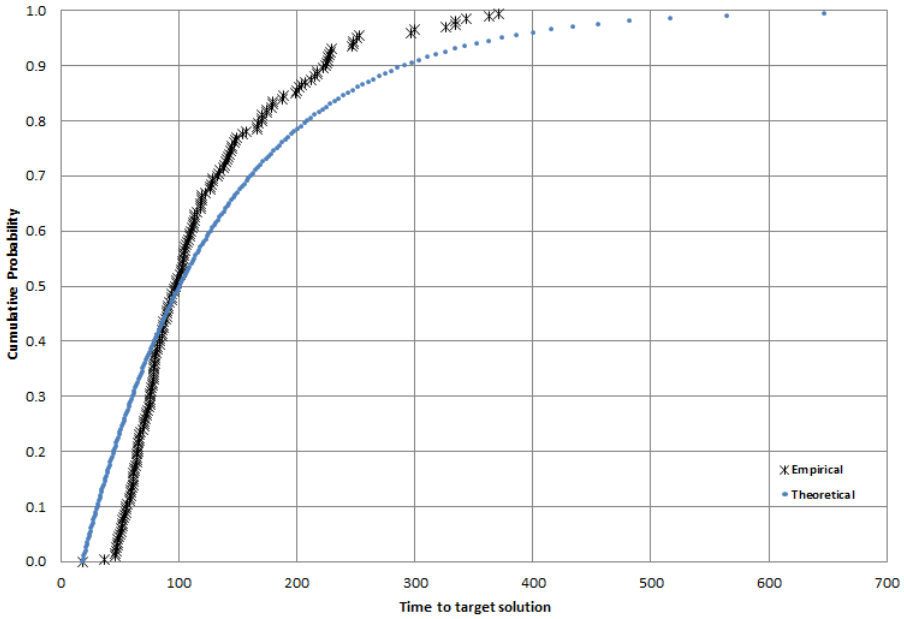


Figura 3.15: Gráfico para instância y.5.100.2 - Alvo 73

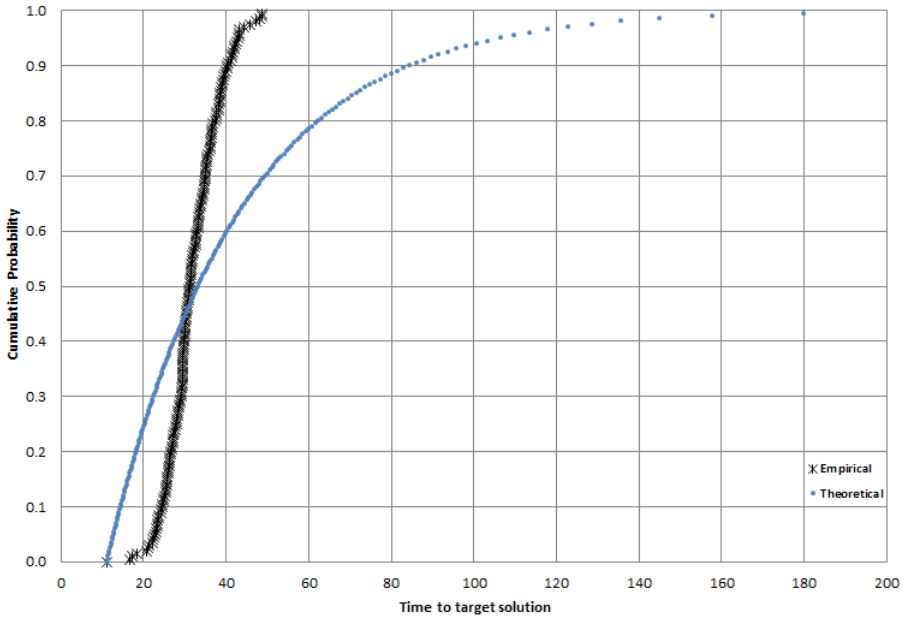


Figura 3.16: Gráfico para instância y.4.80.1 - Alvo 72

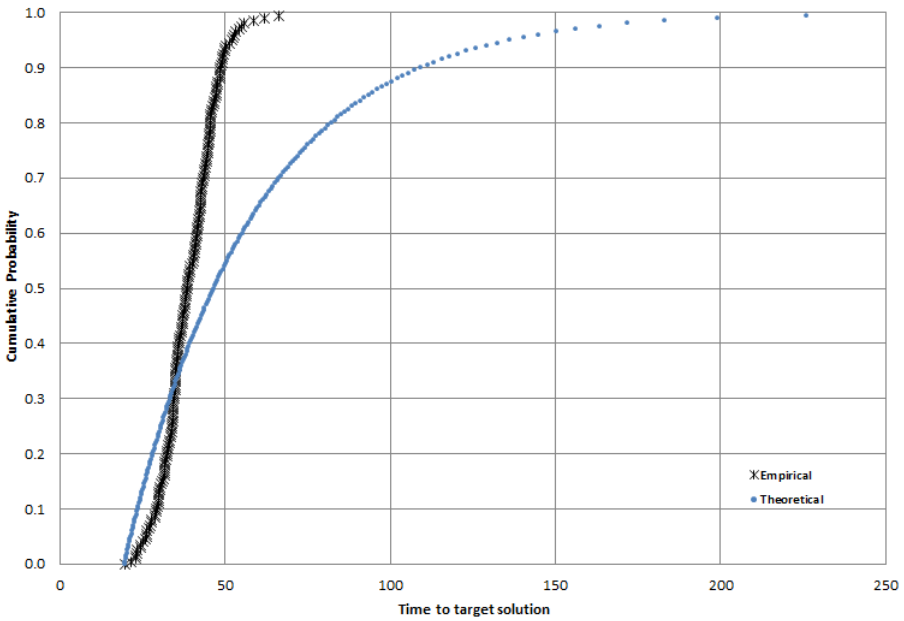


Figura 3.17: Gráfico para instância y.4.100.1 - Alvo 89

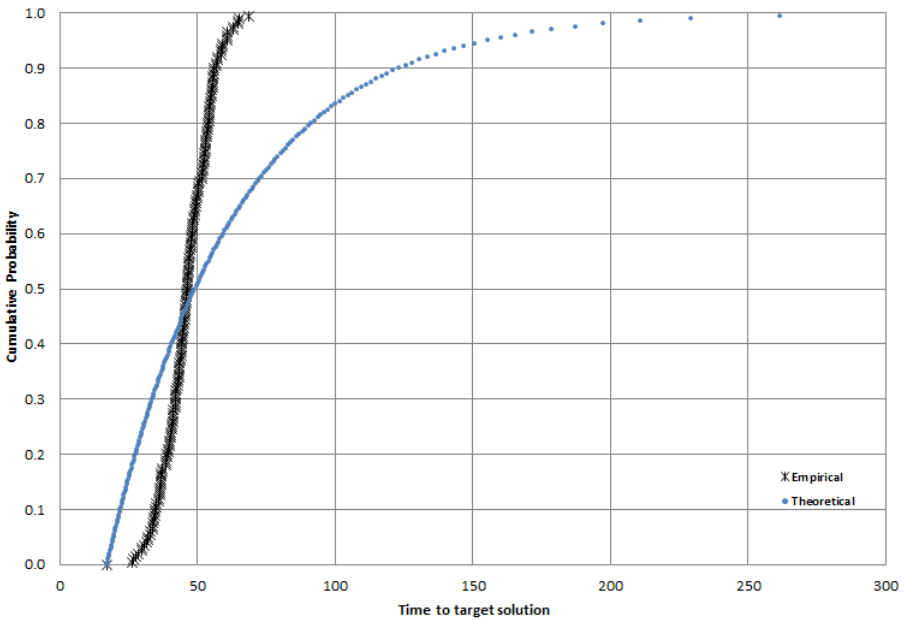


Figura 3.18: Gráfico para instância z.10x10.60 - Alvo 86

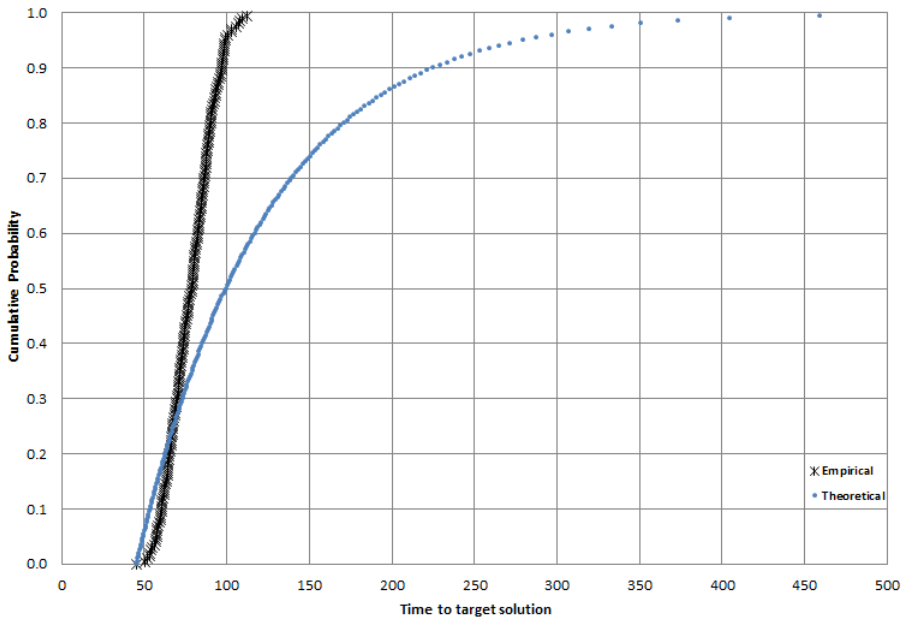


Figura 3.19: Gráfico para instância z.10x10.80 - Alvo 114

Tabela 3.15: Resultados GA-RWA × VND$_e$-ILS$_{5p}$

Métodos	GA-RWA		(5) VND$_e$-ILS$_{5p}$		(30) VND$_e$-ILS$_{5p}$				
Instâncias	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	$\lambda_{\min}$	$\overline{gap}(\%)$	σ	$\bar{t}_{best}$	LI
ATT	24	20.0	20	0.0	20	0.0	0.0	1.9	20
ATT2	113	0.0	113	0.0	113	0.0	0.0	4.3	113
Finland	46	0.4	46	0.0	46	0.0	0.0	1.6	46
NSF.3	22	0.9	22	0.0	22	0.0	0.0	0.1	22
NSF.12	39	2.6	38	0.0	38	0.0	0.0	0.3	38
NSF2.12	35	0.6	35	0.0	35	0.0	0.0	0.1	35
z.10x10.20	31	15.6	29	7.4	29	7.4	0.0	75.1	27
z.6x17.40	87	4.0	85	1.2	85	1.2	0.0	54.9	84
z.10x10.60	87	13.2	84	9.1	84	9.1	0.0	202.8	77
z.4x25.60	195	2.0	193	0.5	193	0.5	0.0	152.8	192
z.10x10.80	115	12.4	112	8.7	112	8.7	0.0	238.1	103
z.8x13.80	134	3.9	130	0.8	130	1.0	0.4	209.8	129
z.6x17.80	176	3.0	171	0.2	171	0.3	0.5	190.8	171
z.5x20.80	209	2.0	206	0.5	205	0.4	0.4	151.0	205
z.4x25.80	260	1.3	258	0.5	258	0.4	0.3	184.2	257
z.5x20.100	257	2.8	253	1.3	253	1.3	0.3	227.7	250
y.4.20.4	20	6.3	19	0.0	19	0.0	0.0	24.0	19
y.3.40.5	59	12.8	56	6.0	56	6.2	0.5	160.0	53
y.3.60.5	86	12.5	82	6.8	82	6.7	0.4	215.7	77
y.4.60.5	58	18.4	55	12.2	55	12.2	0.0	132.5	49
y.5.60.1	36	9.7	35	6.7	35	6.6	0.4	140.2	33
y.3.80.1	122	15.5	115	8.5	114	8.4	0.3	237.9	106
y.3.80.5	113	8.8	109	4.8	108	4.8	0.3	203.0	104
y.4.80.1	73	55.3	70	48.9	70	48.9	0.0	152.6	47
y.4.80.5	75	16.0	72	10.8	72	10.8	0.0	204.7	65
y.5.80.1	47	11.2	46	7.0	46	7.2	0.3	124.5	43
y.5.80.2	60	1.7	59	0.0	59	0.0	0.0	51.2	59
y.4.100.1	90	18.4	86	13.9	86	13.4	0.4	242.3	76
y.5.100.1	58	5.5	57	3.6	57	3.7	0.2	184.4	55
y.5.100.2	74	1.6	73	0.0	73	0.0	0.0	100.6	73
Média		9.3		5.3		5.3			

vizinhança apenas trocas são aceitas. Os testes foram rodados como anteriormente, ou seja, cada instância foi resolvida 5 vezes por cada variação do método tendo como limite 5 minutos para cada execução. A fase de perturbação foi utilizada em todos os testes.

A Tabela 3.16 apresenta os resultados da análise da dependência de cada vizinhança. Para cada variação apresentamos a melhor solução obtida e o desvio médio com relação ao limite inferior. Vemos que quando utilizamos somente a vizinhança

Φ_1 já é possível obter uma média geral melhor que os resultados dos métodos apresentados na literatura até então. A vizinhança Φ_2 quando utilizada sozinha é a que apresenta os piores resultados. É interessante observar que a vizinhança Φ_3 apesar de não ter efeito algum se utilizada isoladamente mostra-se indispensável ao método como podemos ver pelos resultados das combinações $\Phi_1 + \Phi_3$ e $\Phi_2 + \Phi_3$ que apresentam resultados próximos ao VND$_e$-ILS$_{5p}$.

Outra análise feita foi com relação ao tempo gasto em cada etapa e a efetividade de cada uma das vizinhanças desenvolvidas. A Tabela 3.17 apresenta os resultados deste estudo. As colunas indicadas com t apresentam a porcentagem média de tempo que a etapa correspondente utilizou para cada instância. Observamos que a média foi feita sobre 5 execuções. Por exemplo, a coluna t_{Φ_1} indica a porcentagem de tempo gasto pelo método na vizinhança Φ_1 e t_p indica a porcentagem de tempo na fase de perturbação do método. As colunas indicadas com e indicam a efetividade de cada vizinhança, ou seja, a porcentagem de vezes que o movimento foi bem sucedido, entendendo como tentativa bem sucedida quando é possível realizar um movimento na vizinhança com a requisição sob análise. No caso da coluna e_p é apresentada a porcentagem de vezes em que a fase de perturbação foi bem sucedida, ou seja, conseguiu fazer ao menos uma modificação na estrutura da solução.

Como podemos observar pela Tabela 3.17 as vizinhanças Φ_1 e Φ_2 são as que consomem menos tempo na execução do método com médias de 0.5% e 2.2% respectivamente. Das vizinhanças, Φ_3 é a que consome mais tempo, tendo uma média de 24.4% do tempo total, no entanto, como mostrado na análise anterior apresentada pela Tabela 3.16, esta vizinhança, comparada as outras duas, é a que traz maiores benefícios à solução final. De maneira global, a fase de perturbação é a que demanda mais tempo durante uma execução com uma média de 70.3% do tempo total, mas os testes preliminares mostraram que o VND sem a fase ILS produz soluções pouco melhores ou iguais ao BFD [50]. A fase de perturbação também demanda muito tempo devido a baixa efetividade das vizinhanças desenvolvidas. Para os testes feitos, as buscas na vizinhança Φ_1 obtêm movimentos de sucesso 5.9% das vezes em média, 7.9% para as buscas em Φ_2 e apenas 2% das vezes em Φ_3. Isto mostra o quão

difícil é rearranjar as requisições entre os grafos após uma solução ser construída e essa dificuldade se torna cada vez maior à medida que nos aproximamos da solução ótima. A efetividade da fase de perturbação não apresenta um padrão único para todas as instâncias, variando entre 9.7% à 100.0% e tendo como média 69.6%.

Tabela 3.16: Análise da dependência de cada vizinhança

	Φ_1		Φ_2		$\Phi_1 + \Phi_2$		$\Phi_1 + \Phi_3$		$\Phi_2 + \Phi_3$		
Instâncias	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	λ_{min}	$\overline{gap}(\%)$	LI
ATT	20	0.0	26	33.3	20	0.0	20	0.0	20	0.0	20
ATT2	113	0.0	115	2.0	113	0.0	113	0.0	113	0.0	113
Finland	46	0.0	48	5.2	46	0.0	46	0.0	46	0.0	46
NSF.3	22	0.0	22	0.0	22	0.0	22	0.0	22	0.0	22
NSF.12	38	0.0	38	0.0	38	0.0	38	0.0	38	0.0	38
NSF2.12	35	0.0	35	0.0	35	0.0	35	0.0	35	0.0	35
Z.10x10.20	30	14.1	32	20.7	30	11.1	29	7.4	29	7.4	27
Z.6x17.40	87	3.6	88	5.5	85	2.1	86	2.4	85	1.2	84
Z.10x10.60	88	14.3	89	15.8	86	12.7	85	10.4	85	10.4	77
Z.4x25.60	195	2.2	196	3.1	193	1.0	195	1.7	193	0.9	192
Z.10x10.80	115	12.2	116	13.6	115	11.8	114	10.7	113	9.7	103
Z.8x13.80	134	3.9	135	5.0	132	2.9	131	1.9	131	1.6	129
Z.6x17.80	175	2.7	177	4.0	173	1.3	173	1.2	172	0.7	171
Z.5x20.80	209	2.1	210	2.8	206	0.9	207	1.4	206	0.6	205
Z.4x25.80	260	1.5	262	2.2	259	0.8	260	1.3	259	0.8	257
Z.5x20.100	256	2.5	257	3.0	254	1.9	255	2.3	254	1.7	250
y.4.20.4	19	0.0	21	11.6	19	4.2	19	0.0	19	0.0	19
y.3.40.5	58	10.9	60	14.7	58	9.4	57	7.5	57	7.5	53
y.3.60.5	85	11.4	87	13.5	84	10.1	83	8.3	83	7.8	77
y.4.60.5	58	18.4	59	21.2	58	18.4	55	13.5	55	12.2	49
y.5.60.1	36	10.9	37	14.5	36	10.3	35	6.7	35	8.5	33
y.3.80.1	121	14.5	123	17.2	119	12.8	116	10.0	116	9.6	106
y.3.80.5	113	8.7	114	10.0	111	7.5	110	5.8	109	5.4	104
y.4.80.1	73	55.7	74	58.3	73	55.3	70	49.4	70	49.4	47
y.4.80.5	75	15.7	76	17.8	75	15.4	73	12.3	72	11.7	65
y.5.80.1	47	11.2	49	14.0	47	11.2	46	7.0	46	7.9	43
y.5.80.2	59	1.4	60	2.7	59	1.0	59	0.0	59	0.0	59
y.4.100.1	90	18.9	91	20.3	90	18.7	87	14.5	87	14.5	76
y.5.100.1	58	5.5	59	8.4	58	5.5	57	3.6	57	3.6	55
y.5.100.2	74	1.4	76	4.1	74	1.4	73	0.5	73	0.5	73
Média		8.1		11.5		7.6		6.0		5.8	

Tabela 3.17: Porcentagem de tempo e efetividade de cada etapa

Instâncias	t_{Φ_1}	t_{Φ_2}	t_{Φ_3}	t_p	e_{Φ_1}	e_{Φ_2}	e_{Φ_3}	e_p
ATT	1.0	2.8	27.1	65.8	8.0	5.8	1.2	93.6
ATT2	0.8	0.1	5.3	66.9	27.5	24.3	11.2	100.0
Finland	0.8	1.7	17.7	75.6	8.7	9.4	3.8	100.0
NSF.3	5.5	2.6	24.6	61.9	5.4	2.9	1.7	100.0
NSF.12	1.4	0.1	8.5	87.1	6.4	3.6	1.7	100.0
NSF2.12	3.1	3.1	12.0	75.8	19.5	10.7	5.3	100.0
z.10x10.20	0.1	1.9	14.6	83.2	0.9	1.4	0.6	9.7
z.6x17.40	0.1	2.1	22.4	74.9	1.2	3.5	0.5	97.1
z.10x10.60	0.1	2.1	22.6	74.9	2.6	4.5	1.5	48.3
z.4x25.60	0.2	4.9	6.1	87.3	7.1	20.9	0.8	100.0
z.10x10.80	0.1	2.2	23.2	73.9	3.1	4.5	1.4	66.4
z.8x13.80	0.1	2.9	24.3	71.7	3.4	7.0	1.8	96.7
z.6x17.80	0.2	4.3	11.6	82.1	6.8	15.1	1.7	100.0
z.5x20.80	0.2	5.3	7.4	84.8	8.7	21.6	1.3	100.0
z.4x25.80	0.2	4.0	4.5	88.7	9.5	27.9	0.8	100.0
z.5x20.100	0.2	5.8	6.0	84.6	9.2	24.7	1.3	100.0
y.4.20.4	0.1	1.1	31.1	67.0	3.2	2.3	1.6	11.3
y.3.40.5	0.1	1.3	32.2	66.0	1.6	1.6	0.7	52.0
y.3.60.5	0.1	1.5	34.7	62.7	2.6	3.3	1.4	73.8
y.4.60.5	0.1	2.1	34.6	62.7	3.1	3.5	2.2	28.7
y.5.60.1	0.1	0.8	37.3	61.4	2.3	1.8	0.9	14.2
y.3.80.1	0.1	2.0	27.4	69.1	5.3	6.8	2.9	93.0
y.3.80.5	0.1	1.8	29.0	67.7	3.9	4.4	1.7	86.1
y.4.80.1	0.1	2.0	31.5	65.7	3.7	4.4	2.8	36.3
y.4.80.5	0.1	1.7	32.2	65.2	3.6	4.1	2.8	38.5
y.5.80.1	0.1	1.3	36.1	61.9	3.7	3.0	1.5	22.9
y.5.80.2	0.1	0.3	50.1	46.7	3.9	1.4	0.7	74.0
y.4.100.1	0.1	2.5	32.1	64.3	4.5	5.8	3.2	47.1
y.5.100.1	0.1	1.3	37.5	60.2	3.9	2.9	1.5	25.3
y.5.100.2	0.1	0.6	47.0	49.8	5.1	2.6	0.8	73.6
Média	0.5	2.2	24.4	70.3	5.9	7.9	2.0	69.6

3.5 Conclusões

Neste capítulo foi apresentado o principal método proposto para obtenção de construção de soluções para o Problema de Roteamento e Alocação de Comprimentos de Onda com o objetivo de minimizar o número de comprimentos de onda em sua versão estática com relação às requisiçoes, o *Best Fit Decreasing* (BFD).

Para tentar um refinamento maior das soluções geradas, foi proposto um algoritmo baseado em VND que utilizava em sua busca local o próprio método de geração

da solução para um pequeno conjunto de requisições. Também foi acrescentado um método híbrido que usava uma Busca Tabu sempre que a busca local do VND era encerrada. Nesta comparação foi verificado que o VND-BFD apresentou os melhores resultados para a maioria das instâncias.

Depois apresentamos uma nova abordagem para o VND onde o foco se concentra nas requisições e alguns movimentos podem não causar uma melhora imediata na função objetivo. Para perturbar as soluções geradas pelo VND criamos um algoritmo ILS com duas variações, a primeira onde todas as requisições têm a mesma probabilidade de fazer parte de um problema de atribuição criado para a perturbação e o segundo onde certas requisições apresentam uma probabilidade cinco vezes maior de fazer parte deste problema. Verificamos através de testes computacionais que esta segunda versão, chamada de VND_e-ILS_{5p}, apresentou os melhores resultados. Para esta variante mostramos que os melhores resultados são encontrados quando utillizamos o BFD para a geração da solução inicial, mas mesmo utilizando outro método, o EDP, o algoritmo é capaz de encontrar boas soluções.

Foi mostrada uma comparação do VND_e-ILS_{5p} com o Algoritmo Genético apresentado em [44] que até então apresenta os melhores resultados para este problema. O método VND_e-ILS_{5p} proposto se mostrou eficiente, tendo melhorado a maioria das instâncias ainda em aberto.

Para finalizar, apresentamos um estudo das componentes do VND_e-ILS_{5p}, mostrando o tempo de cada uma das fases e a efetividade destas. Nesta parte, concluímos que as componentes que demandam mais tempo são as que geram maior benefício para a função objetivo.

Capítulo 4

Formulações para maximizar o número de requisições estabelecidas

Neste capítulo serão apresentadas algumas formulações para o RWA. Na literatura, a grande parte dos modelos e métodos exatos são voltados para a resolução do problema com o objetivo de maximizar o número de requisições atendidas. Apresentaremos ainda modificações aos modelos matemáticos baseados em arcos existentes para o problema. O objetivo aqui é deixar os modelos mais 'leves' e com isso conseguirmos resolver instâncias de mesmo tamanho em um tempo menor ou até mesmo instâncias maiores que não são resolvidas pelos modelos existentes. Na literatura recente podemos encontrar trabalhos onde se combinam métodos exatos e métodos heurísticos, em que as heurísticas dividem o problema original em subproblemas menores e estes subproblemas são resolvidos de forma exata. Dessa forma, é importante que se tenham modelos matemáticos que resolvam pequenas instâncias em um tempo razoável ([40], [52]). Ainda apresentaremos alguns modelos baseados na estratégia de geração de colunas e uma nova abordagem será proposta.

4.1 Modelos compactos

Krishnaswamy e Sivarajan [33] apresentaram um modelo com o objetivo de maximizar o número de requisições estabelecidas baseado em arcos. Seja $x_i = 1$ se a requisição i é estabelecida e $x_i = 0$ caso contrário. Seja $x_i^\lambda = 1$ se o comprimento de onda λ suporta a requisição i e $x_i^\lambda = 0$ caso contrário. Seja $x_{ie}^\lambda = 1$ se o comprimento de onda λ suporta a requisição i usando o arco e e $x_{ie}^\lambda = 0$ caso contrário. Con-

sidere $\omega^+(j)$ e $\omega^-(j)$, respectivamente, o conjunto de arcos que saem e que entram no vértice j. Neste modelo temos um número limitado de comprimentos de onda disponíveis, assim consideramos também Λ o conjunto de comprimentos de onda.

$$(\text{KS}) \max \sum_{i \in R} x_i \tag{4.1}$$

s.a:

$$\sum_{e \in \omega^+(j)} x_{ie}^{\lambda} = \sum_{e \in \omega^-(j)} x_{ie}^{\lambda}, \quad \forall\ i \in R, j \in V \setminus \{s_i, d_i\}, \lambda \in \Lambda \tag{4.2}$$

$$\sum_{e \in \omega^+(s_i)} x_{ie}^{\lambda} - \sum_{e \in \omega^-(s_i)} x_{ie}^{\lambda} = x_i^{\lambda}, \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.3}$$

$$\sum_{e \in \omega^-(d_i)} x_{ie}^{\lambda} - \sum_{e \in \omega^+(d_i)} x_{ie}^{\lambda} = x_i^{\lambda}, \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.4}$$

$$\sum_{i \in R} x_{ie}^{\lambda} \leq 1, \quad \forall\ e \in E, \lambda \in \Lambda \tag{4.5}$$

$$\sum_{\lambda \in \Lambda} x_i^{\lambda} = x_i, \quad \forall\ i \in R \tag{4.6}$$

$$x_{ie}^{\lambda} \leq x_i^{\lambda}, \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.7}$$

$$x_i, x_i^{\lambda}, x_{ie}^{\lambda} \in \{0,1\}, \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.8}$$

Os três primeiros conjuntos de restrições garantem a continuidade do comprimento de onda. O conjunto de restrições (4.5) garante que um mesmo comprimento de onda não será usado em duas ou mais requisições utilizando um mesmo arco. O conjunto de restrições (4.6) faz com que se uma requisição é atendida ela será atendida por um único comprimento de onda. Finalmente, o conjunto de restrições (4.7) garante a consistência entre as variáveis x_{ie}^{λ} e x_i^{λ}.

As soluções apresentadas por este modelo podem conter ciclos; no entanto, estes ciclos não afetam o valor da função objetivo. No trabalho apresentado por Jaumard et al. [26], o modelo (KS) é alterado eliminando-se o conjunto de variáveis x_i que indicam se a requisição i é atendida ou não, substituindo as restrições (4.3), (4.4) e (4.6), respectivamente pelas restrições (4.11), (4.12) e (4.14) e introduzindo dois novos conjuntos de restrições, um para impedir que arcos sejam ativados para

requisições sem que existam comprimentos de onda alocados para estas requisições (4.15), e outro de modo a eliminar os possíveis ciclos que envolvem os vértices origem e destino das requisições (4.16). O modelo é apresentado a seguir:

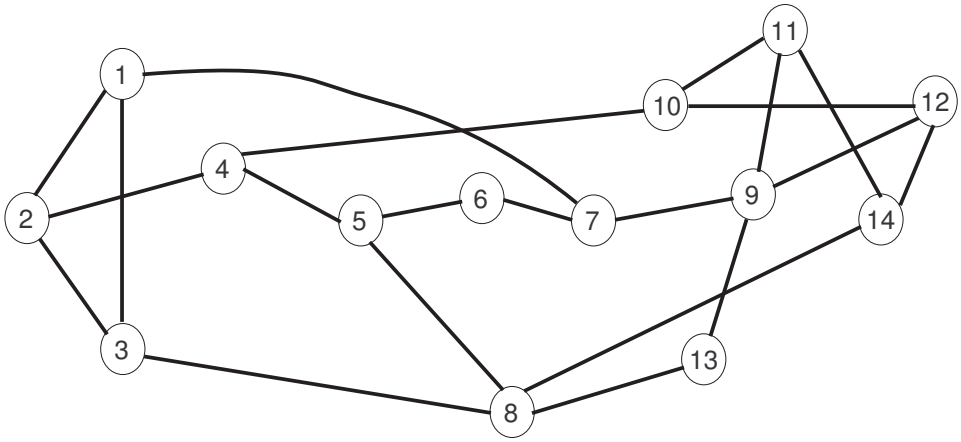


Figura 4.1: NSFnet - [45]

$$\text{(JMT)} \quad \max \sum_{\lambda \in \Lambda} \sum_{i \in R} x_i^{\lambda} \tag{4.9}$$

s.a:

$$\sum_{e \in \omega^+(j)} x_{ie}^{\lambda} = \sum_{e \in \omega^-(j)} x_{ie}^{\lambda}, \quad \forall\, i \in R, j \in V \setminus \{s_i, d_i\}, \lambda \in \Lambda \tag{4.10}$$

$$\sum_{e \in \omega^+(s_i)} x_{ie}^{\lambda} = x_i^{\lambda}, \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.11}$$

$$\sum_{e \in \omega^-(d_i)} x_{ie}^{\lambda} = x_i^{\lambda}, \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.12}$$

$$\sum_{i \in R} x_{ie}^{\lambda} \leq 1, \quad \forall\ e \in E, \lambda \in \Lambda \tag{4.13}$$

$$\sum_{\lambda \in \Lambda} x_i^{\lambda} \leq 1, \quad \forall\ i \in R \tag{4.14}$$

$$x_{ie}^{\lambda} \leq x_i^{\lambda}, \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.15}$$

$$\sum_{e \in \omega^-(s_i)} x_{ie}^{\lambda} = \sum_{e \in \omega^+(d_i)} x_{ie}^{\lambda} = 0, \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.16}$$

$$x_i^{\lambda}, x_{ie}^{\lambda} \in \{0,1\}, \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.17}$$

Uma comparação de modelos de programação matemática para o RWA com o

objetivo de maximizar o número de conexões estabelecidas pode ser encontrada em [24], [25] e [26]. Nestes trabalhos podemos verificar que, apesar das abordagens diferentes dadas aos modelos, os limites superiores encontrados por eles são sempre os mesmos e que a relaxação fornece valores muito próximos ou iguais aos valores ótimos. Com algumas adaptações na formulação baseada em arcos, Jaumard et al. [26] foram capazes de resolver algumas instâncias realísticas pela primeira vez.

A seguir será mostrado que o modelo (JMT), apesar de impedir ciclos envolvendo os vértices origem e destino de cada requisição, não é capaz de prevenir todos os ciclos. Para alguns testes preliminares será usada a rede realística NSFnet apresentada na Figura 4.1. A Tabela 4.1 mostra as requisições com suas origens e seus destinos.

Para o número de comprimentos de onda igual a dois a solução apresentada pelo modelo (KS) pode ser observada na Figura 4.2. São apresentados na figura somente os arcos que foram usados na solução, os arcos que não foram usados foram eliminados. Cada subgrafo representa as requisições atendidas por um mesmo comprimento de onda; no exemplo, o comprimento de onda 1 atende às requisições $(\{3,11\};\{4,12\};\{6,14\};\{7,1\};\{11,5\};\{13,4\})$, enquanto o comprimento de onda 2 atende às requisições $(\{1,11\};\{2,14\};\{4,14\};\{5,1\})$.

Tabela 4.1: Requisições da rede NSFnet

Requisição	Origem	Destino
1	1	11
2	2	14
3	3	11
4	4	12
5	4	14
6	5	1
7	6	14
8	7	1
9	11	5
10	13	4

Observe que na solução apresentada pelo modelo (KS) existe um ciclo entre os vértices $\{8,14\}$ no grafo associado ao comprimento de onda 1. Este ciclo foi

gerado no atendimento à requisição 8 entre os vértices $\{7, 1\}$. Contudo, como já dito anteriormente, isso não prejudica a recuperação da solução e a eliminação deste ciclo não influencia o valor da função objetivo, que para o exemplo é igual a 10, ou seja, todas as requisições são atendidas.

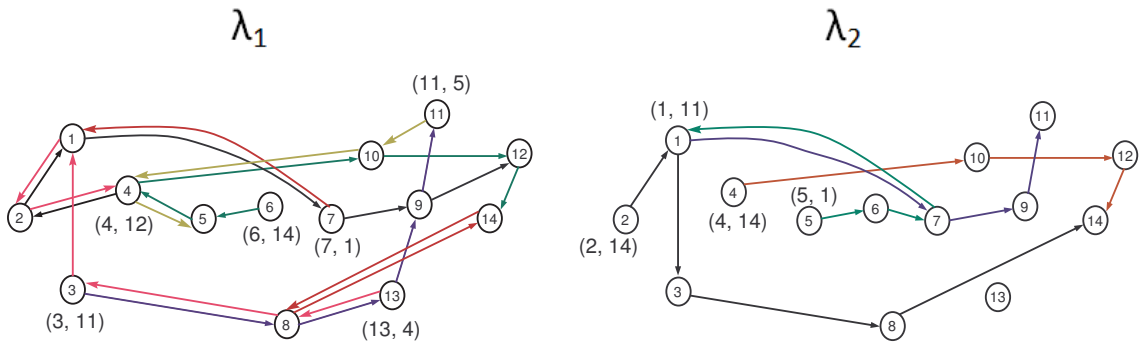


Figura 4.2: Solução do modelo KS

Para o mesmo exemplo, ao resolvermos pelo modelo (JMT) obtemos a solução apresentada pela Figura 4.3, que também é ótima, atendendo todas as requisições. Na solução podemos ver que não existe nenhum ciclo envolvendo os vértices origem e destino de nenhuma requisição, mas mesmo assim ainda podemos observar ciclos na solução. Os ciclos apresentados por esta solução estão todos associados ao comprimento de onda 1. Os ciclos podem ser vistos entre os vértices $\{12, 14\}$, $\{10, 12\}$ e $\{11, 14\}$, gerados no atendimento das requisições 3 entre os vértices $\{3, 11\}$, 7 entre os vértices $\{6, 14\}$ e 10 entre os vértices $\{13, 4\}$. Novamente, os ciclos gerados não impedem a recuperação da solução e a eliminação destes não causa impacto no valor da função objetivo.

4.1.1 Formulações propostas

Nesta seção será feita uma análise de algumas restrições dos modelos (KS) e (JMT) e algumas alterações serão sugeridas de forma a facilitar a resolução dos modelos.

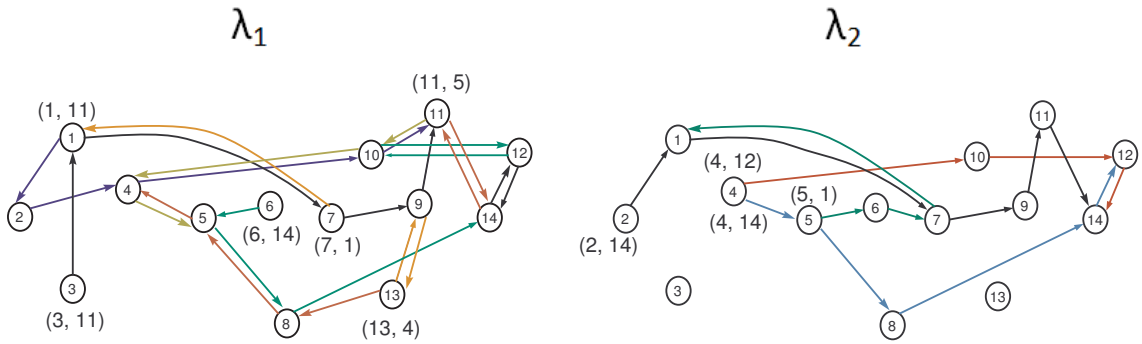


Figura 4.3: Solução do modelo JMT

4.1.1.1 Formulação baseada no modelo KS

As restrições (4.3) e (4.4), garantem que, sempre que uma requisição seja atendida o fluxo do vértice da origem desta requisição seja maior em uma unidade ao fluxo que entra e que o fluxo que entra no vértice destino seja uma unidade maior ao fluxo que sai. Dessa maneira, é possível que haja algum ciclo envolvendo os vértices origem e destino. Combinadas a estas duas restrições a restrição (4.2) garante a continuidade do fluxo e a continuidade do comprimento de onda.

Para impedir os ciclos Jaumard et al. [26] forçam que o segundo membro do lado esquerdo das equações (4.3) e (4.4) sejam iguais a zero introduzindo a restrição (4.16).

As restrições (4.3) e (4.4) formam um conjunto de $2 \times |R| \times |\lambda|$ restrições, mas para formar um caminho entre a origem e o destino de uma requisição podemos simplificar estes dois conjuntos de restrições substituindo as restrições (4.3) e (4.4) por:

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(s_i)} x_{ie}^{\lambda} = x_i, \quad \forall\ i \in R \tag{4.18}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(d_i)} x_{ie}^{\lambda} = x_i, \quad \forall\ i \in R \tag{4.19}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(s_i)} x_{ie}^{\lambda} = \sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(d_i)} x_{ie}^{\lambda} = 0, \quad \forall\ i \in R \tag{4.20}$$

Note que, dessa forma, se uma requisição for atendida, as restrições (4.18) e (4.19) não garantem que o mesmo comprimento de onda que sai do vértice origem chegue ao vértice destino, mas ao combinarmos essas duas restrições à restrição (4.2) isso passa a ser garantido, respeitando-se a restrição de continuidade do comprimento de onda.

Continuando, para evitar ciclos onde requisições estejam desativadas podemos usar a seguinte restrição no modelo (KS):

$$\sum_{\lambda \in \Lambda} \sum_{e \in E} x_{ie}^{\lambda} \leq \quad (|V| - 1) \times x_i, \quad \forall \ i \in R \tag{4.21}$$

já que para qualquer grafo o maior caminho sem ciclo é sempre menor ou igual a $(|V| - 1)$. Dessa forma, se uma requisição estiver ativa permitimos que arcos sejam associados a ela, senão, o número de arcos associados a ela deve ser igual a zero.

A restrição (4.7) garante que um arco só pode ser usado para uma requisição usando um comprimento de onda se este comprimento de onda estiver associado a esta requisição. Isto além de contribuir para a continuidade do comprimento de onda, ainda impede a formação de ciclos para requisições que não foram ativadas. Como a continuidade do comprimento de onda já é garantida pelas restrições mostradas anteriormente e a formação de ciclos não impede a recuperação da solução e nem afeta o valor da função objetivo esta restrição pode ser eliminada, diminuindo $|R| \times |E| \times |\Lambda|$ restrições do modelo.

Com isso obtemos a formulação modificada, sobre o modelo (KS), que chamaremos de (RWA1) mostrada pelas equações (4.22)-(4.29), a seguir:

$$(\text{RWA1}) \max \sum_{i \in R} x_i \tag{4.22}$$

s.a:

$$\sum_{e \in \omega^+(j)} x_{ie}^{\lambda} - \sum_{e \in \omega^-(j)} x_{ie}^{\lambda} = 0 \quad , \quad \forall\ i \in R, j \in V \setminus \{s_i, d_i\}, \lambda \in \Lambda \tag{4.23}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(s_i)} x_{ie}^{\lambda} = x_i \quad , \quad \forall\ i \in R \tag{4.24}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(d_i)} x_{ie}^{\lambda} = x_i \quad , \quad \forall\ i \in R \tag{4.25}$$

$$\sum_{i \in R} x_{ie}^{\lambda} \leq 1 \quad , \quad \forall\ e \in E, \lambda \in \Lambda \tag{4.26}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in E} x_{ie}^{\lambda} \leq (|V| - 1) \times x_i \quad , \quad \forall\ i \in R \tag{4.27}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(s_i)} x_{ie}^{\lambda} = \sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(d_i)} x_{ie}^{\lambda} = 0 \quad , \quad \forall\ i \in R, \lambda \in \Lambda \tag{4.28}$$

$$x_i, x_{ie}^{\lambda} \in \{0, 1\} \quad , \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.29}$$

Repare que neste modelo as variáveis x_i^{λ} foram eliminadas.

4.1.1.2 Formulação baseada no modelo JMT

As modificações feitas no modelo JMT seguem a mesma lógica das modificações feitas anteriormente. Primeiramente alteramos as restrições (4.11), (4.12) e (4.16) para garantir que apenas uma unidade de fluxo sairá da origem da requisição e chegará no destino caso a requisição seja atendida. Também garantimos que não haverá fluxo chegando à origem nem deixando o destino. Para tal, as novas restrições ficam como segue:

$$\sum_{\lambda\in\Lambda}\sum_{e\in\omega^+(s_i)} x_{ie}^{\lambda} = \sum_{\lambda\in\Lambda} x_i^{\lambda}, \quad \forall\ i \in R \tag{4.30}$$

$$\sum_{\lambda\in\Lambda}\sum_{e\in\omega^-(d_i)} x_{ie}^{\lambda} = \sum_{\lambda\in\Lambda} x_i^{\lambda}, \quad \forall\ i \in R \tag{4.31}$$

$$\sum_{\lambda\in\Lambda}\sum_{e\in\omega^-(s_i)} x_{ie}^{\lambda} = \sum_{\lambda\in\Lambda}\sum_{e\in\omega^+(d_i)} x_{ie}^{\lambda} = 0, \quad \forall\ i \in R \tag{4.32}$$

Para impedir ciclos formados por requisições não atendidas podemos substituir a restrição (4.15) do modelo (JMT) por:

$$\sum_{\lambda\in\Lambda}\sum_{e\in E} x_{ie}^{\lambda} \leq (|V|-1) \times \sum_{\lambda\in\Lambda} x_i^{\lambda}, \quad \forall\ i \in R \tag{4.33}$$

O segundo modelo é obtido com as modificações sobre o modelo (JMT) que chamaremos de (RWA2) e é apresentado a seguir pelas equações (4.34)-(4.42):

$$\text{(RWA2)} \quad \max \sum_{\lambda \in \Lambda} \sum_{i \in R} x_i^\lambda \tag{4.34}$$

s.a:

$$\sum_{e \in \omega^+(j)} x_{ie}^\lambda - \sum_{e \in \omega^-(j)} x_{ie}^\lambda = 0 \quad , \quad \forall\, i \in R, j \in V \setminus \{s_i, d_i\}, \lambda \in \Lambda \tag{4.35}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(s_i)} x_{ie}^\lambda = \sum_{\lambda \in \Lambda} x_i^\lambda \quad , \quad \forall\ i \in R \tag{4.36}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(d_i)} x_{ie}^\lambda = \sum_{\lambda \in \Lambda} x_i^\lambda \quad , \quad \forall\ i \in R \tag{4.37}$$

$$\sum_{i \in R} x_{ie}^\lambda \leq 1 \quad , \quad \forall\ e \in E, \lambda \in \Lambda \tag{4.38}$$

$$\sum_{\lambda \in \Lambda} x_i^\lambda \leq 1 \quad , \quad \forall\ i \in R \tag{4.39}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in E} x_{ie}^\lambda \leq (|V| - 1) \times \sum_{\lambda \in \Lambda} x_i^\lambda \quad , \quad \forall\ i \in R \tag{4.40}$$

$$\sum_{\lambda \in \Lambda} \sum_{e \in \omega^-(s_i)} x_{ie}^\lambda = \sum_{\lambda \in \Lambda} \sum_{e \in \omega^+(d_i)} x_{ie}^\lambda = 0 \quad , \quad \forall\ i \in R \tag{4.41}$$

$$x_i^\lambda, x_{ie}^\lambda \in \{0, 1\} \quad , \quad \forall\ i \in R, e \in E, \lambda \in \Lambda \tag{4.42}$$

Repare que para o modelo RWA2 as variáveis permanecem as mesmas do modelo JMT.

A seguir apresentamos um exemplo para ilustrar que as soluções providas pelos dois modelos modificados são soluções viáveis para o problema de roteamento e alocação de comprimentos de onda. O teste foi feito sobre a mesma instância da seção anterior.

As soluções encotrandas pelos dois modelos foram exatamente iguais e são apresentadas pela Figura 4.4. Note que a solução apresentada não apresenta ciclo algum, apesar disto não ter sido evitado explicitamente.

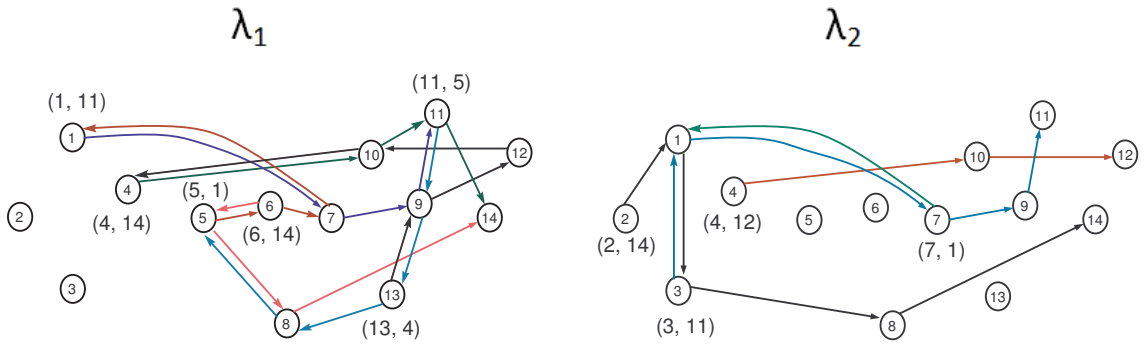


Figura 4.4: Solução dos modelos RWA1 e RWA2

4.1.2 Resultados Computacionais

Nesta seção serão apresentados os testes comparando os modelos propostos com os modelos da literatura.

Serão utilizadas nos testes desta seção dois conjuntos de instâncias. Um conjunto com a rede NFSnet com variações no número de requisições, com NFS.1 representando a instância com 100% de requisições, ou seja, requisições entre todos os pares de vértices, NFS.08 com 80% e assim sucessivamente, onde a matriz de tráfego foi construída aleatoriamente. O outro conjunto de instâncias foi criado de acordo com a metodologia proposta em [43] para as instâncias do conjunto Z. De acordo com esta metodologia foram criadas duas redes com 12 e 16 vértices e são apresentadas variações no número de requisições da mesma maneira descrita anteriormente.

As instâncias foram resolvidas utlizando o software CPLEX versão 9.1 em um computador Dell Studio Core 2 Duo, 2.10 GHz, com 4 GB de RAM sob sistema operacional Ubuntu.

A Tabela 4.2 apresenta os resultados para as quatro variações de modelos matemáticos com relação ao GAP encontrado pela relaxação linear e o tempo necessário para resolver o modelo relaxado. Na última coluna é apresentado o número de comprimentos de onda disponíveis que nos testes foi variado de 3, 5 e 7. Estes resultados são para as instâncias do conjunto Z. A Tabela 4.3 apresenta os mesmos resultados com relação à variações na matriz de tráfego da instância NFSnet.

Tabela 4.2: Resultados da relaxação dos modelos - Instâncias Z

Modelos	RWA1		RWA2		KS		JMT		
Instâncias	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	Λ
Z.4x4-1	0.00	33.12	0.00	32.00	0.00	269.92	0.00	188.04	
Z.4x3-1	0.00	3.52	0.00	3.54	0.00	19.30	0.00	10.14	
Z.4x4-08	0.00	36.4	0.00	35.92	0.00	180.82	0.00	173.58	
Z.4x3-08	0.00	3.12	0.00	3.10	0.00	8.16	0.00	7.58	
Z.4x4-06	0.67	16.44	0.67	16.26	0.67	90.70	0.67	64.04	3
Z.4x3-06	0.00	1.16	0.00	1.16	0.00	2.46	0.00	2.02	
Z.4x4-04	0.00	2.76	0.00	2.84	0.00	13.80	0.00	14.38	
Z.4x3-04	0.00	0.22	0.00	0.22	0.00	0.34	0.00	0.46	
Z.4x4-02	0.00	0.00	0.00	0.02	0.00	0.32	0.00	0.40	
Z.4x3-02	0.00	0.02	0.00	0.02	0.00	0.06	0.00	0.14	
Z.4x4-1	0.18	12.00	0.18	12.24	0.18	1657.14	0.18	1067.00	
Z.4x3-1	0.00	105.00	0.00	121.44	0.00	75.46	0.00	59.20	
Z.4x4-08	0.30	69.48	0.30	83.88	0.30	718.18	0.30	530.86	
Z.4x3-08	0.00	5.74	0.00	5.80	0.00	26.78	0.00	37.40	
Z.4x4-06	0.00	44.02	0.00	43.98	0.00	252.38	0.00	290.26	
Z.4x3-06	0.00	0.38	0.00	0.38	0.00	0.60	0.00	2.76	5
Z.4x4-04	0.00	0.20	0.00	0.20	0.00	10.38	0.00	12.72	
Z.4x3-04	0.00	0.02	0.00	0.02	0.00	0.12	0.00	0.36	
Z.4x4-02	0.00	0.04	0.00	0.04	0.00	0.22	0.00	0.64	
Z.4x3-02	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.04	
Z.4x4-1	0.00	223.30	0.00	223.96	0.00	4503.64	0.00	3458.18	
Z.4x3-1	0.00	9.00	0.00	8.96	0.00	183.58	0.00	150.56	
Z.4x4-08	0.00	194.86	0.00	194.64	0.00	2997.16	0.00	3057.54	
Z.4x3-08	0.00	1.68	0.00	1.74	0.00	62.82	0.00	29.38	
Z.4x4-06	0.00	10.68	0.00	10.64	0.00	842.78	0.00	624.02	
Z.4x3-06	0.00	0.10	0.00	0.08	0.00	1.36	0.00	1.60	7
Z.4x4-04	0.00	0.18	0.00	0.16	0.00	2.40	0.00	11.50	
Z.4x3-04	0.00	0.02	0.00	0.04	0.00	0.08	0.00	0.30	
Z.4x4-02	0.00	0.02	0.00	0.02	0.00	0.16	0.00	0.90	
Z.4x3-02	0.00	0.00	0.00	0.02	0.00	0.04	0.00	0.06	
Total	1.16	773.48	1.16	803.32	1.16	11921.20	1.16	9796.06	

Os resultados apresentados nas Tabelas 4.2 e 4.3 mostram que as modificações propostas não alteram os valores encontrados pelas relaxações dos modelos, que, no caso, são sempre iguais à solução ótima ou um valor muito próximo. Por outro lado, podemos perceber que os tempos necessários para a resolução do modelo relaxado caem significativamente para os modelos RWA1 e RWA2.

As Tabelas 4.4 e 4.5 apresentam os tempos computacionais gastos até cada um dos modelos testados encontrar a solução ótima. A penúltima coluna das tabelas apresenta o valor ótimo de cada instância, ou seja, o número de requisições atendidas, enquanto a última coluna mais uma vez apresenta o número de comprimentos de onda disponíveis.

Tabela 4.3: Resultados da relaxação dos modelos - Instâncias NFSnet

Modelos	RWA1		RWA2		KS		JMT		
Instâncias	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	GAP(%)	Tempo(s)	Λ
NFS.1	0.00	5.04	0.00	5.00	0.00	64.32	0.00	29.52	
NFS.08	0.00	2.92	0.00	2.80	0.00	28.32	0.00	16.66	
NFS.06	1.04	2.56	1.04	2.56	1.04	10.40	1.04	8.70	3
NFS.04	0.86	2.14	0.86	2.14	0.86	6.96	0.86	4.34	
NFS.02	0.00	0.60	0.00	0.61	0.00	1.42	0.00	1.22	
NFS.1	0.00	21.38	0.00	21.48	0.00	169.86	0.00	167.68	
NFS.08	0.64	15.10	0.64	15.02	0.64	94.76	0.64	77.70	
NFS.06	0.00	7.48	0.00	7.46	0.00	28.46	0.00	24.20	5
NFS.04	0.00	4.04	0.00	4.08	0.00	20.42	0.00	15.18	
NFS.02	0.00	0.28	0.00	0.28	0.00	1.26	0.00	1.04	
NFS.1	0.00	56.62	0.00	42.24	0.00	381.24	0.00	432.06	
NFS.08	0.00	52.66	0.00	40.30	0.00	190.94	0.00	239.82	
NFS.06	0.00	22.76	0.00	22.82	0.00	73.04	0.00	152.58	7
NFS.04	0.00	1.62	0.00	1.72	0.00	26.18	0.00	12.60	
NFS.02	0.00	0.08	0.00	0.08	0.00	0.22	0.00	0.30	
Total	2.54	195.28	2.54	168.59	2.54	1097.8	2.54	1183.6	

Conforme podemos ver nas Tabelas 4.4 e 4.5 as modificações propostas causam um grande impacto nos tempos computacionais. Todos os modelos são capazes de encontrar a solução ótima para todas as instâncias, mas os modelos RWA1 e RWA2 resolvem todas as instâncias em tempo inferior aos modelos KS e JMT.

Tabela 4.4: Tempo até o ótimo - Instâncias Z

Modelos	RWA1	RWA2	KS	JMT		
Instâncias	Tempo(s)	Tempo(s)	Tempo(s)	Tempo(s)	Ótimo	Λ
Z.4x4-1	74	73	471	314	128	
Z.4x3-1	4	4	20	26	96	
Z.4x4-08	60	58	187	241	121	
Z.4x3-08	5	5	9	16	79	
Z.4x4-06	36	36	447	254	106	3
Z.4x3-06	2	2	12	6	64	
Z.4x4-04	7	8	95	31	80	
Z.4x3-04	1	1	1	1	43	
Z.4x4-02	0	1	1	1	39	
Z.4x3-02	1	0	1	1	21	
Z.4x4-1	569	595	6336	3768	181	
Z.4x3-1	49	35	188	104	120	
Z.4x4-08	338	447	13181	1497	167	
Z.4x3-08	21	16	155	69	102	
Z.4x4-06	192	151	733	731	137	5
Z.4x3-06	1	2	1	21	70	
Z.4x4-04	1	1	21	29	80	
Z.4x3-04	0	1	0	1	43	
Z.4x4-02	0	0	0	2	39	
Z.4x3-02	0	0	0	1	21	
Z.4x4-1	1535	3823	15593	8514	224	
Z.4x3-1	26	26	568	383	132	
Z.4x4-08	1110	962	6270	25119	197	
Z.4x3-08	15	15	220	48	105	
Z.4x4-06	63	62	1746	1011	141	7
Z.4x3-06	1	1	2	4	70	
Z.4x4-04	1	0	4	16	80	
Z.4x3-04	0	0	0	1	43	
Z.4x4-02	1	0	1	2	39	
Z.4x3-02	1	0	1	0	21	
Total	4114	6325	46264	42212		

Tabela 4.5: Tempo até o ótimo - Instâncias NFSnet

Modelos	RWA1	RWA2	KS	JMT		
Instâncias	Tempo(s)	Tempo(s)	Tempo(s)	Tempo(s)	Ótimo	Λ
NFS.1	6	6	105	30	84	
NFS.08	4	3	31	18	76	
NFS.06	6	5	31	18	64	3
NFS.04	3	3	12	5	58	
NFS.02	1	1	3	4	41	
NFS.1	40	41	206	189	115	
NFS.08	86	87	165	305	104	
NFS.06	17	17	34	42	86	5
NFS.04	10	10	36	23	75	
NFS.02	1	1	4	5	46	
NFS.1	109	94	679	519	140	
NFS.08	70	80	304	276	125	
NFS.06	32	37	209	279	102	7
NFS.04	7	8	139	95	81	
NFS.02	1	0	2	0	46	
Total	393	393	1960	1808		

4.2 Modelos Estendidos

Baseados na ideia de conjuntos independentes, Ramaswami e Sivarajan [46] apresentaram uma formulação para o problema de roteamento e alocação de comprimentos de onda com o objetivo de maximizar o número de requisições atendidas. Seja G_p, dito grafo de conflitos, um novo grafo onde cada vértice corresponde a um caminho em G e dois vértices em G_p são adjacentes se os dois caminhos correspondentes em G têm pelo menos um arco em comum. Os caminhos podem ser um dado de entrada ou podem ser gerados por algum procedimento (Note que o número de caminhos entre os vértices origem-destino pode crescer exponencialmente [46]). A parte da alocação é tratada como um problema de conjuntos independentes. Em outras palavras, um conjunto de caminhos em G pode receber um mesmo comprimento de onda se e somente se os correspondentes vértices em G_p formam um conjunto independente.

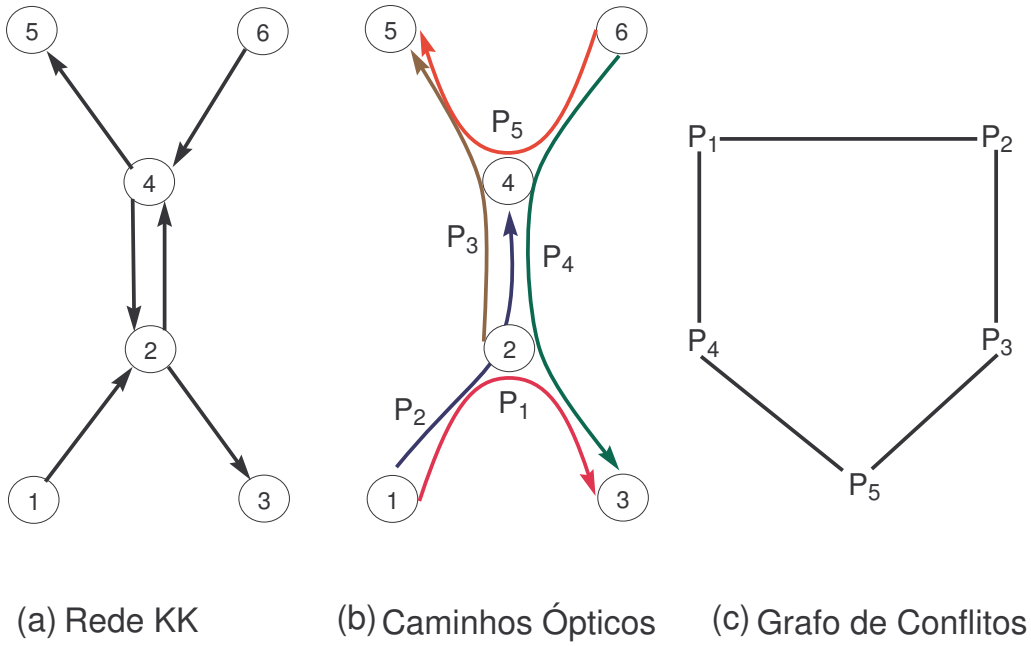


Figura 4.5: Exemplo de conjuntos independentes [27]

A Figura 4.5 apresenta uma ideia sobre conjuntos independentes. A Figura 4.5 (a) (rede KK extraída de Kleinberg e Kumar [31]) apresenta o grafo original com os vértices e as ligações físicas. Assuma que a matriz de tráfego seja $r_{13} = r_{14} = r_{25} = r_{63} = r_{65} = 1$, e que para todos os outros pares de vértices $r_{ij} = 0$. Os caminhos

definidos para realizar as requisições são $p_1 = \{1, 2, 3\}$, $p_2 = \{1, 2, 4\}$, $p_3 = \{2, 4, 5\}$, $p_4 = \{6, 4, 2, 3\}$ e $p_5 = \{6, 4, 5\}$, veja Figura 4.5 (b). A Figura 4.5 (c) apresenta o grafo de conflitos, onde cada vértice representa um caminho e cada aresta representa que os dois caminhos compartilham um arco no grafo original. As arestas do grafo de conflitos pertencem ao conjunto E_W.

Pelo grafo de conflito podemos ver que os conjuntos independentes deste exemplo são $\{p_1\}$, $\{p_2\}$, $\{p_3\}$, $\{p_4\}$, $\{p_5\}$, $\{p_1, p_3\}$, $\{p_1, p_5\}$, $\{p_2, p_4\}$, $\{p_2, p_5\}$ e $\{p_3, p_4\}$, onde os conjuntos com dois elementos são os conjuntos independentes maximais.

O conjunto de comprimentos de onda disponíveis é denotado por $\Lambda = \lambda_1, \lambda_2, \ldots, \lambda_W$ com $W = |\Lambda|$. Seja P o conjunto de todos os caminhos elementares (isto é, caminhos sem ciclos) e P_{sd} o conjunto de caminhos elementares entre s e d. Seja L_{max} o conjunto contendo todos os conjuntos independentes em G_p. Seja δ_{pl} a matriz que indica se um caminho p pertence ao conjunto independente l e $\sum_{p \in P_{sd}} \delta_{pl} = a^l_{sd}$, para $l \in L_{max}$. Seja w_l o número de comprimentos de onda associado ao conjunto independente l, ou seja, cada conjunto independente pode ser usado w_l vezes, e a^l_{sd} o número de requisições entre o par $\{s, d\}$ atendidas pelo conjunto l. O modelo MAX-IS é dado como segue:

$$\text{(MAX-IS) max } Z_{max-is}(w, y) = \sum_{(s,d) \in R} y_{sd} \tag{4.43}$$

s.a:

$$\sum_{l \in L_{max}} w_l \leq W \tag{4.44}$$

$$y_{sd} \leq \sum_{l \in L_{max}} a^l_{sd} w_l, \qquad \forall\ (s, d) \in R \tag{4.45}$$

$$y_{sd} \leq r_{sd}, \qquad \forall\ (s, d) \in R \tag{4.46}$$

$$w_l \in \mathbb{N}, \qquad \forall\ l \in L_{max} \tag{4.47}$$

A variável y_{sd} conta o número de requisições atendidas entre os vértices s e d para todo (s, d) em R. Ramaswami e Sivarajan [46] propuseram a formulação MAX-IS

mas não a utilizaram. Jaumard et al. [27] mostraram como resolver este modelo usando geração de colunas.

Seja lp-MAX-IS a relaxação do modelo ao trocarmos a restrição (4.47) por $w_l \geq 0$ para todo $l \in L_{max}$. Como o número de conjuntos independentes maximais pode ser exponencial, trabalha-se no então chamado Problema Mestre Restrito (PMR) com todas as variáveis y_{sd} tal que $(s,d) \in R$ e um subconjunto de $\{w_l : l \in L_{max}\}$. Para verificar se a solução ótima do PMR é também solução ótima para o problema original é necessário verificar se existe uma variável w_l com um custo reduzido positivo. Se existe tal variável, ela é adicionada ao PMR, senão a solução é ótima para o problema original.

Seja u^0 a variável dual associada à restrição (4.44) e u^1_{sd} a variável dual associada à restrição (4.45). O custo reduzido para w_l é:

$$\overline{c}(w_l) = -u^0 + \sum_{(s,d)\in R} a^l_{sd} u^1_{sd}$$

Assim, o subproblema pode ser escrito como:

$$\max\ \overline{c}_{AUX-MAX-IS}(\alpha) = -u^0 + \sum_{(s,d)\in R} \sum_{p\in P_{sd}} u^1_{sd}\alpha_p \tag{4.48}$$

s.a:

$$\alpha_p + \alpha_{p'} \leq 1, \qquad \forall\ (r_p, r_{p'}) \in E_W \tag{4.49}$$

$$\alpha_p \in \{0,1\}, \qquad \forall\ p \in P \tag{4.50}$$

onde $\alpha_p = 1$ se o vértice r_p pertence ao conjunto independente e 0 caso contrário.

O problema deste modelo é que o grafo de conflitos pode envolver um número exponencial de vértices.

Lee et al. [35] introduziram o conceito de configuração de roteamento independente (*Independent Routing Configuration*), onde cada configuração é implicitamente associada a um conjunto de caminhos que podem ser usados para satisfazer

uma fração das requisições utilizando o mesmo comprimento de onda. Uma configuração de roteamento independente C é representada por um vetor a^c tal que:

a^c_{sd} = número de requisições entre (s,d) suportadas em C

$a^c_{sd} \leq r_{sd}$ para todo $(s,d) \in R$.

Seja $\mathcal{C}$ o conjunto de todas as possíveis configurações de roteamento independentes. Seja w_c a variável que indica quantas vezes uma dada configuração de roteamento independente é usada, sendo a cada ocorrência associado um comprimento de onda diferente. O modelo, denotado por IRC, pode ser expresso como:

$$\text{(IRC)} \quad \max \quad Z_{IRC}(w) = \sum_{c \in \mathcal{C}} \sum_{(s,d) \in R} a^c_{sd} w_c \tag{4.51}$$

s.a:

$$\sum_{c \in \mathcal{C}} w_c \leq W \tag{4.52}$$

$$\sum_{c \in \mathcal{C}} a^c_{sd} w_c \leq r_{sd}, \qquad \forall \ (s,d) \in R \tag{4.53}$$

$$w_c \in \mathbb{N}, \qquad \forall \ c \in \mathcal{C} \tag{4.54}$$

Sejam (u^0, u^1_{sd}) as variáveis duais da solução ótima do modelo IRC relaxado. O custo reduzido $\overline{c}(w_c)$ pode ser calculado como:

$\overline{c}(w_c) = -u^0 + \sum_{(s,d) \in R} (1 - u^1_{sd}) a^c_{sd}$. O subproblema do modelo IRC apresenta as mesmas restrições do subproblema MAX-IRC que será apresentado a seguir. A diferença é somente a função objetivo.

Em [27] é feita uma revisão acerca de modelos de geração de colunas para o RWA com o objetivo de maximizar o número de conexões estabelecidas e algumas melhorias são sugeridas. Além disso, combinando as ideias de Ramaswami e Sivarajan [46] e Lee et al. [35], Jaumard et al. [27] apresentaram uma nova formulação em que utilizam somente as configurações de roteamento independentes maximais, onde uma configuração de roteamento independente C é maximal se não existe outra configuração de roteamento independente C' tal que $a'_c \geq a_c$.

Seja C_{max} o conjunto de todas as configurações de roteamento independentes maximais e seja w_c o número de vezes que uma determinada configuração é utilizada. O modelo pode ser dado como segue:

$$\text{(MAX-IRC)} \quad \max \quad Z_{MAX-IRC}(w,y) = \sum_{(s,d)\in R} y_{sd} \tag{4.55}$$

s.a:

$$\sum_{c\in C_{max}} w_c \leq W \tag{4.56}$$

$$y_{sd} \leq \sum_{c\in C_{max}} a^c_{sd} w_c, \qquad \forall\ (s,d) \in R \tag{4.57}$$

$$y_{sd} \leq r_{sd}, \qquad \forall\ (s,d) \in R \tag{4.58}$$

$$w_c \in \mathbb{N}, \qquad \forall\ c \in C_{max} \tag{4.59}$$

Sejam u^0 e u^1_{sd} as variáveis duais associadas respectivamente às restrições (4.56) e (4.57) na solução ótima do problema (MAX-IRC). O custo reduzido para a variável w_c é $-u^0 + \sum_{(s,d)\in R} a^C_{sd} u^1_{sd}$. O subproblema pode ser definido como segue:

$$\max\ \overline{c}_{AUX-MAX-IRC}(\alpha) = -u^0 + \sum_{(s,d)\in R} \sum_{e\in\omega^+(s)} u^1_{sd} \alpha^{sd}_e \tag{4.60}$$

s.a:

$$\sum_{(s,d)\in R} \alpha^{sd}_e \leq 1, \qquad \forall\, e \in E \tag{4.61}$$

$$\sum_{e\in\omega^+(i)} \alpha^{sd}_e = \sum_{e\in\omega^-(i)} \alpha^{sd}_e, \qquad \forall\, (s,d) \in R,\ i \in V \setminus \{s,d\} \tag{4.62}$$

$$\sum_{e\in\omega^+(s)} \alpha^{sd}_e \leq r_{sd}, \qquad \forall\, (s,d) \in R \tag{4.63}$$

$$\sum_{e\in\omega^-(s)} \alpha^{sd}_e = 0, \qquad \forall\, (s,d) \in R \tag{4.64}$$

$$\alpha^{sd}_e \in \{0,1\}, \qquad \forall\, (s,d) \in R, e \in E \tag{4.65}$$

Jaumard et al. [27] mostraram que esta formulação apresenta limites superiores tão bons quanto o modelo apresentado em [35], mas, no entanto, usam menos variáveis. Sendo assim, no mesmo trabalho foi apresentado um algoritmo *Branch-and-price* baseado nesta última formulação.

4.2.1 Abordagem proposta

Recapitulando, o modelo MAX-IS foi proposto por Ramaswami e Sivarajan [46] e não foi utilizado pelo fato de que para se ter garantia que a solução ótima será encontrada é necessário que todos os caminhos para cada uma das requisições sejam definidos a priori ou pelo menos todos os caminhos que farão parte da solução ótima. Jaumard et al. [27] aproveitaram as ideias do modelo MAX-IS e do modelo IRC [35] e propuzeram o modelo MAX-IRC. Neste trabalho foi mostrado que se relaxarmos as restrições (4.63), o modelo resultante, MAX-IRC-RL, apresenta soluções lineares iguais ao modelo MAX-IS, ou seja, $Z^{LP}_{MAX-IS} = Z^{LP}_{MAX-IRC-RL} \geq Z^{LP}_{MAX-IRC}$. Note que esta restrição impede que uma configuração de roteamento independente seja formada atendendo um par origem destino com mais caminhos do que demandado. Note também que no modelo MAX-IRC os caminhos são gerados a cada chamada ao subproblema (caminhos implícitos), enquanto no modelo MAX-IS os caminhos são pré-definidos e apenas a combinação é definida no subproblema (caminhos explícitos).

Se o modelo MAX-IS gera soluções com limites piores que o modelo MAX-IRC pelo fato de poder gerar conjuntos independentes que não respeitam a demanda podemos introduzir a restrição (4.66), a seguir,

$$\sum_{p \in P_{sd}} \alpha_p \leq r_{sd}, \qquad \forall\ (s,d) \in R \tag{4.66}$$

no subproblema MAX-IS impedindo que um conjunto independente possa ser gerado com um número maior de caminhos explícitos do que a demanda solicitada. Seja a^l um vetor de coeficientes para uma variável w_l em MAX-IS, onde cada elemento $a^l_{sd} \leq r_{sd}$ para todo $(s,d) \in R$. Cada elemento do vetor representa o número de caminhos explícitos que podemos utilizar para atender uma fração da demanda com um mesmo comprimento de onda. Todos os caminhos usados para construir o vetor a^l são disjuntos; logo, existe um vetor a^c, de coeficientes para uma variável w_c em MAX-IRC, igual a a^l. Assim, se as restrições (4.66) forem introduzidas no subproblema MAX-IS, qualquer variável em MAX-IS tem uma variável equivalente

em MAX-IRC, logo, $Z^{LP}_{MAX-IS} \leq Z^{LP}_{MAX-IRC}$.

Por outro lado, se o modelo MAX-IS possui todos os caminhos possíveis para cada $(s,d) \in R$, cada vetor a^c de caminhos implícitos em MAX-IRC pode ser convertido em pelo menos um vetor a^l em MAX-IS, mesmo com as restrições (4.66), já que $a^c_{sd} \leq r_{sd}$ para todo $(s,d) \in R$. Assim temos que se as restrições (4.66) forem utilizadas $Z^{LP}_{MAX-IS} = Z^{LP}_{MAX-IRC}$.

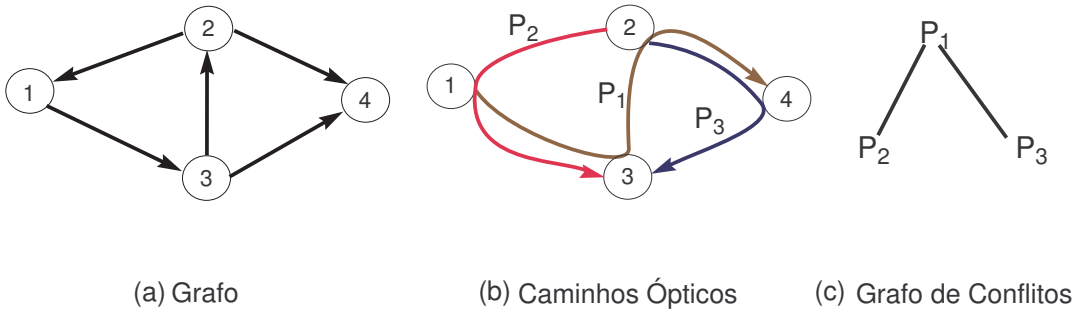


Figura 4.6: Exemplo de Instância [27]

Para exemplificar imagine a instância apresentada na Figura 4.6. Existem demandas entre os vértices $\{1,4\}$ e $\{2,3\}$. Sendo assim, temos 3 caminhos possíveis (parte (b)), e o grafo de conflitos é apresentado na parte (c). No modelo MAX-IRC os coeficientes das variáveis seriam representadas por:

$$a^{c_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, a^{c_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, a^{c_3} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

Repare que o vetor de coeficientes a^{c_2} indica que é possível construir uma configuração de roteamento independente com apenas um caminho atendendo a requisição 2, sem especificar qual caminho será este (caminho implícito).

Para o modelo MAX-IS os coeficientes seriam representados por:

$$a^{l_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a^{l_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a^{l_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, a^{l_3} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

Se as demandas forem unitárias e se utilizarmos as restrições (4.63) no subproblema do modelo MAX-IRC e as restrições (4.66) no subproblema do modelo MAX-IS, as variáveis que utilizam os coeficientes a^{c_3} e a^{l_4} não serão geradas.

Até então o modelo MAX-IS não foi utilizado pela necessidade de se ter todos os caminhos possíveis para que a solução encontrada tenha garantia de ser ótima. Como o número de caminhos pode ser exponencial, o subproblema pode ser intratável. Suponhamos então que tenhamos a priori todos os caminhos da solução ótima faltando apenas definir a atribuição de cada caminho ao comprimento de onda que lhe convém. Assim podemos trabalhar usando somente os caminhos definidos a priori, ou seja, teremos um subconjunto $L_r \subset L_{max}$ de variáveis w^l que poderão fazer parte do problema.

O modelo apresentado por [4] pode ser reformulado para dar uma boa previsão de quais serão os caminhos presentes em uma solução ótima. Dessa forma apresentamos um modelo de geração de colunas que fornecerá os caminhos que serão usados na fase de atribuição. Seja x_p o número de vezes que o caminho p será utilizado e $a_e = 1$ se o caminho p utiliza o arco e e $a_e = 0$ caso contrário. O modelo pode ser definido como segue:

$$\text{(Path-Gen)} \quad \max \quad Z_{path-gen}(x) = \sum_{(s,d)\in R} \sum_{p\in P_{sd}} x_p \tag{4.67}$$

s.a:

$$\sum_{(s,d)\in R} \sum_{p\in P_{sd}} a_e x_p \leq W \qquad \forall\ e \in E \tag{4.68}$$

$$\sum_{p\in P_{sd}} x_p \leq r_{sd}, \qquad \forall\ (s,d) \in R \tag{4.69}$$

$$x_p \in \mathbb{N}, \qquad \forall\ p \in P \tag{4.70}$$

O subproblema da formulação Path-Gen pode ser resolvido por um algoritmo de caminho mínimo para cada par (s, d) em que o custo dos arcos é dado pelo valor dual das restrições (4.68). Assim, o primeiro algoritmo apresentado, PG-MAX-IS, utiliza somente os caminhos gerados pela formulação Path-Gen no modelo MAX-IS. Obviamente a otimalidade da solução encontrada por esta estratégia não pode ser garantida, já que não se pode garantir que os caminhos gerados pelo modelo Path-Gen estarão realmente na solução ótima.

A ideia pode ser visualizada pela Figura 4.7. A área maior representa o conjunto convexo formado quando temos a priori todos os caminhos possíveis para a resolução do modelo MAX-IS. Nessa situação resolvemos o problema P, que tem a solução ótima indicada pela letra (a). A situação ideal para o modelo Path-Gen é gerar um subconjunto representado pelo problema P', que apesar de não englobar todas as soluções de P, engloba a solução ótima. Como não podemos garantir que ao final da execução do modelo Path-Gen todas os caminhos da solução ótima serão gerados, podemos chegar à situação apresentada pelo problema P'' em que a solução mostrada pela letra (b) não é a solução ótima do problema original.

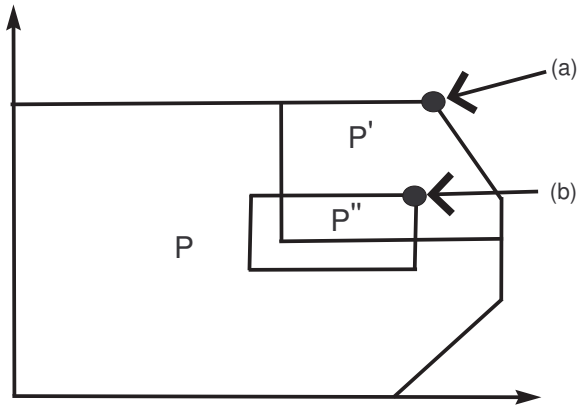


Figura 4.7: Conjunto Convexo para o problema MAX-IS

Para contornar este problema apresentamos uma segunda estratégia que é a união dos modelos MAX-IS e MAX-IRC, onde o modelo MAX-IS, novamente, utiliza somente os caminhos gerados pelo modelo Path-Gen. O modelo MAX-IS-IRC pode ser definido como segue:

$$\text{(MAX-IS-IRC)} \quad \max \quad Z_{MAX-IS-IRC}(w, y) = \sum_{(s,d)\in R} y_{sd} \tag{4.71}$$

s.a:

$$\sum_{l\in L_r} w_l + \sum_{c\in C_{max}} w_c \leq W \tag{4.72}$$

$$y_{sd} \leq \sum_{l\in L_r} a^l_{sd} w_l + \sum_{c\in C_{max}} a^c_{sd} w_c, \quad \forall\ (s,d) \in R \tag{4.73}$$

$$y_{sd} \leq r_{sd}, \quad \forall\ (s,d) \in R \tag{4.74}$$

$$w_c \in \mathbb{N}, \quad \forall\ c \in C_{max} \tag{4.75}$$

$$w_l \in \mathbb{N}, \quad \forall\ l \in L_r \tag{4.76}$$

Repare que, agora, com o modelo MAX-IS-IRC, temos a opção de gerar variáveis w_l e w_c, utilizando os subproblemas já apresentados anteriormente para cada um dos modelos.

Em nosso algoritmo, PG-MAX-IS-IRC, utilizamos a seguinte estratégia: primeiramente geramos os caminhos através do modelo Path-Gen e geramos o grafo de conflitos. Em seguida, utilizamos o modelo MAX-IS-IRC sempre gerando variáveis w_l, que em geral é mais fácil de ser resolvido já que os caminhos já estão calculados. Somente tenta-se gerar uma variável w_c quando o custo reduzido da variável w_l for menor ou igual zero. Se os custos reduzidos de w_l e w_c são menores ou iguais a zero em uma iteração significa que a solução é ótima.

4.2.2 Resultados Computacionais

Os algoritmos foram implementados na linguagem C++ usando o compilador Visual Studio e testados em um microcomputador Dell Optiplex 780, 3.00 GHz, com 3.25 GB de RAM sob sistema operacional Windows XP.

Para os testes foram usadas 24 instâncias disponíveis na literatura, conhecidas como instâncias realísticas. As características, como número de requisições, número de arestas e número de nós, de cada uma das instâncias estão descritas na Tabela 4.6.

Os testes computacionais foram executados da seguinte maneira: para cada método rodamos todas as instâncias com o número de comprimentos de onda disponíveis entre 10, 20 e 30. Os métodos executados são a versão de geração de colunas da literatura (MAX-IRC [27]), o modelo de geração de caminhos combinado com o modelo MAX-IS (PG-MAX-IS) e o modelo de Path-Gen combinado com o modelo MAX-IS-IRC (PG-MAX-IS-IRC).

Tabela 4.6: Instâncias Realísticas

Instâncias	Requisições	Arestas	Nós
att	359	137	90
att2	2918	175	71
brasil	1370	70	27
cost266	6543	57	37
dfn-bwin	4840	45	10
dfn-gwin	3771	47	11
eon	373	39	20
fin	930	51	31
france	15398	45	25
giul	14732	86	39
janos-us	3262	42	26
nobel-eu	1898	41	28
nobel-germany	660	26	17
nobel-us	478	21	14
norway	5348	51	27
nsf1	284	21	14
nsf3	285	21	14
nsf12	551	21	14
nsf48	547	21	14
nsf2-1	284	22	14
nsf2-3	285	22	14
nsf212	551	22	14
nsf2-48	547	22	14
sun	952	51	27

A Tabela 4.7 apresenta os resultados da comparação entre o método proposto em [27] e os métodos propostos neste trabalho, somente com relação à função objetivo. Podemos ver que a estratégia de geração de caminhos desenvolvida para a utilização do modelo MAX-IS apresenta um bom desempenho e não compromete a qualidade das soluções para nenhuma das instâncias testadas quando usamos $W = 10$. Obeservamos isso analisando que todas as instâncias onde o método não consegue chegar na solução linear ótima, os valores são sempre menores que uma unidade, dessa

maneira, por exemplo, para a instância *brasil*, onde a solução linear ótima é 721.50, sabemos que o limite superior para solução inteira será então 721, valor encontrado pelo método PG-MAX-IS. Podemos observar também que o método PG-MAX-IS-IRC é capaz de contornar as limitações do PG-MAX-IS como era de se esperar e encontra todas as soluções lineares ótimas. Observe que para a instância *att*2 o método MAX-IRC é interrompido com uma solução igual a 351.00 devido ao estouro de memória.

Tabela 4.7: MAX-IRC × PG-MAX-IS - W = 10

Modelos Instâncias	MAX-IRC Fo	PG-MAX-IS Fo	PG-MAX-IS-IRC Fo
att	253.00	253.00	253.00
att2	351.00(*)	895.00	895.00
brasil	721.50	721.00	721.50
cost266	788.50	788.00	788.50
dfn-bwin	884.00	884.00	884.00
dfn-gwin	874.00	874.00	874.00
eon	285.00	285.00	285.00
fin	444.77	444.00	444.77
france	610.00	610.00	610.00
giul	1561.00	1561.00	1561.00
janos-us	600.00	600.00	600.00
nobel-eu	346.00	346.00	346.00
nobel-germany	228.00	228.00	228.00
nobel-us	190.00	190.00	190.00
norway	834.00	834.00	834.00
nsf1	197.00	197.00	197.00
nsf3	195.50	195.00	195.50
nsf12	264.00	264.00	264.00
nsf48	254.00	254.00	254.00
nsf2-1	205.00	205.00	205.00
nsf2-3	206.00	206.00	206.00
nsf212	280.33	280.00	280.33
nsf2-48	266.33	266.00	266.33
sun	264.00	264.00	264.00

Na Tabela 4.8 podemos ter uma visão mais esclarecedora sobre o desempenho do método PG-MAX-IS. Observamos que a metodologia é bastante eficiente e encontra soluções tão boas quanto as outras para a maioria das instâncias, contudo podemos verificar nas instâncias *att* e $france$ que as perdas começam a ser maiores que uma unidade, respectivamente, 1.5 e 7. Verificamos ainda que, novamente, a metodologia

PG-MAX-IS-IRC foi capaz de encontrar as soluções ótimas com limites tão bons quanto o método MAX-IRC. Mais uma vez observamos que o método MAX-IRC não foi capaz de resolver a instância *att*2 devido a estouro de memória.

Tabela 4.8: MAX-IRC × PG-MAX-IS - W = 20

Modelos Instâncias	MAX-IRC Fo	PG-MAX-IS Fo	PG-MAX-IS-IRC Fo
att	359.00	357.50	359.00
att2	351.00(*)	1298.00	1298.00
brasil	1080.67	1080.00	1080.67
cost266	1325.00	1325.00	1325.00
dfn-bwin	1733.00	1733.00	1733.00
dfn-gwin	1519.00	1519.00	1519.00
eon	369.00	369.00	369.00
fin	642.00	642.00	642.00
france	1188.00	1181.00	1188.00
giul	2519.00	2519.00	2519.00
janos-us	981.00	981.00	981.00
nobel-eu	596.00	596.00	596.00
nobel-germany	384.00	384.00	384.00
nobel-us	326.50	326.50	326.50
norway	1377.00	1377.00	1377.00
nsf1	278.00	278.00	278.00
nsf3	277.00	277.00	277.00
nsf12	408.00	408.00	408.00
nsf48	389.00	389.00	389.00
nsf2-1	282.00	282.00	282.00
nsf2-3	284.00	284.00	284.00
nsf212	427.00	427.00	427.00
nsf2-48	413.00	413.00	413.00
sun	502.00	502.00	502.00

Na Tabela 4.9 apresentamos os resultados somente para instâncias em que são necessários mais que 30 comprimentos de onda para atender todas as requisições. Vemos nesta tabela que para $W = 30$ o desempenho do método PG-MAX-IS é comparável aos demais e que a perda para estes testes, no pior caso foi de 0.50 para a instância *nobel-us*. Novamente o método PG-MAX-IS-IRC é capaz de encontrar exatamente os mesmos limites superiores do método MAX-IRC, que por sua vez novamente não foi capaz de resolver a instância *att*2.

Apesar de ter apresentado um bom desempenho, o método PG-MAX-IS não pode garantir que a solução encontrada é verdadeiramente a solução ótima. Sendo assim,

as tabelas (4.10) a (4.12) mostram uma comparação de desempenho somente dos métodos MAX-IRC e PG-MAX-IS-IRC. Nestas tabelas apresentamos o tempo gasto para encontrar a solução ótima de cada método e o número de variáveis geradas.

Tabela 4.9: MAX-IRC × PG-MAX-IS - W = 30

Modelos Instâncias	MAX-IRC Fo	PG-MAX-IS Fo	PG-MAX-IS-IRC Fo
att2	351.00(*)	1648.00	1648.00
brasil	1241.00	1241.00	1241.00
cost266	1728.94	1728.71	1728.94
dfn-bwin	2555.50	2555.50	2555.50
dfn-gwin	1945.00	1945.00	1945.00
fin	774.00	774.00	774.00
france	1730.00	1730.00	1730.00
giul	3307.50	3307.50	3307.50
janos-us	1239.00	1239.00	1239.00
nobel-eu	822.00	822.00	822.00
nobel-germany	489.00	489.00	489.00
nobel-us	432.50	432.00	432.50
norway	1782.00	1782.00	1782.00
nsf12	499.00	499.00	499.00
nsf48	469.00	469.00	469.00
nsf212	522.00	522.00	522.00
nsf2-48	505.00	505.00	505.00
sun	722.00	722.00	722.00

A Tabela 4.10 apresenta a comparação entre o MAX-IRC e o PG-MAX-IS-IRC para $W = 10$. O método MAX-IRC não foi capaz de resolver a instância $att2$. Podemos ver que o método PG-MAX-IS-IRC sempre foi capaz de resolver as instâncias em um tempo inferior ao MAX-IRC. Das instâncias resolvidas pelo MAX-IRC a instância Finland foi a que demandou mais tempo deste método, 27780.5 segundos. Para esta instância, o método proposto, PG-MAX-IS-IRC, gasta apenas 269.2 segundos. Com relação ao número de variáveis geradas observamos que em geral o MAX-IRC necessita gerar um número menor de variáveis para encontrar a solução ótima, mas isto nem sempre é verdade como podemos verificar nas instâncias $brasil$, $dfn\text{-}gwin$, $france$ e $nsf3$ por exemplo.

Tabela 4.10: Resultados para W = 10

Modelos	MAX-IRC		PG-MAX-IS-IRC	
Instâncias	Tempo(s)	♯Var	Tempo(s)	♯Var
att	653.9	106	18.2	325
att2*	-	1	567.4	997
brasil	7809.6	663	44.5	598
cost266	2002.7	178	27.0	265
dfn-bwin	0.3	3	0.0	9
dfn-gwin	4.0	31	0.1	28
eon	109.9	132	2.8	207
fin	27780.5	1285	269.2	1308
france	9.8	9	1.1	7
giul	4015.8	409	102.0	525
janos-us	105.6	67	5.6	97
nobel-eu	54.3	54	1.5	61
nobel-germany	2.6	25	0.1	19
nobel-us	0.9	17	0.1	37
norway	134.9	61	7.8	118
nsf1	17.9	108	0.8	144
nsf3	22.0	138	0.6	107
nsf12	14.6	102	0.8	128
nsf48	22.8	139	0.7	118
nsf2-1	22.0	114	0.8	149
nsf2-3	19.0	110	0.8	160
nsf212	17.6	118	1.0	110
nsf2-48	23.7	134	1.1	125
sun	6.0	11	0.1	10

Na Tabela 4.11 apresentamos a comparação de desempenho entre o MAX-IRC e o PG-MAX-IS-IRC para $W = 20$. Observamos que o comportamento permanece semelhante ao apresentado anteriormente. O método PG-MAX-IS-IRC sempre encontra a solução ótima em um tempo inferior ao MAX-IRC, este, por sua vez, não é capaz de encontrar a solução ótima para a instância *att*2. Também podemos ver que a instância *Finland* é a que demanda mais tempo do método MAX-IRC, 14078.7 segundos, enquanto o PG-MAX-IS-IRC gasta 629.2 segundos. Novamente, com relação ao número de variáveis geradas, o MAX-IRC, em geral, é capaz de encontrar a solução ótima com menos variáveis, mas isso nem sempre ocorre.

Para finalizar apresentamos a Tabela 4.12 com os resultados do desempenho entre o MAX-IRC e o PG-MAX-IS-IRC para $W = 30$. O mesmo comportamento dos testes anteriores pode ser observado. O MAX-IRC em geral resolvendo as instâncias com menos variáveis, mas sempre utilizando um tempo maior. Dessa vez a instância que demandou mais tempo do MAX-IRC foi a instância *giul*, 21809.4 segundos.

Tabela 4.11: Resultados para W = 20

Modelos	MAX-IRC		MAX-IS-IRC	
Instâncias	Tempo(s)	♯Var	Tempo(s)	♯Var
att	254.9	106	44.7	531
att2*	-	1	3074.4	1626
brasil	11111.6	930	110.8	967
cost266	11498.2	828	220.9	832
dfn-bwin	1.8	23	0.0	25
dfn-gwin	12.1	105	0.2	75
eon	21.2	64	3.8	270
fin	14078.7	660	629.2	1554
france	22.1	16	1.5	18
giul	12400.4	572	107.9	492
janos-us	512.4	268	20.1	319
nobel-eu	108.6	113	3.2	108
nobel-germany	6.9	59	0.4	70
nobel-us	4.7	82	0.2	78
norway	1269.1	468	29.1	393
nsf1	9.6	99	1.1	197
nsf3	8.8	95	0.8	146
nsf12	25.3	181	1.1	169
nsf48	17.4	116	0.8	135
nsf2-1	7.0	71	0.6	125
nsf2-3	10.8	111	0.6	126
nsf212	24.8	159	1.0	160
nsf2-48	24.8	151	1.1	171
sun	21.9	35	0.3	70

Para esta instância, o PG-MAX-IS-IRC gasta apenas 128.3 segundos para encontrar a solução ótima.

Tabela 4.12: Resultados para W = 30

Modelos	MAX-IRC		PG-MAX-IS-IRC	
Instâncias	Tempo(s)	♯Var	Tempo(s)	♯Var
att2*	-	1	7149.7	1925
brasil	1242.6	198	58.8	633
cost266	19060.1	1226	339.2	1136
dfn-bwin	2.8	36	0.1	33
dfn-gwin	7.5	63	0.2	76
fin	15849.0	777	717.0	1545
france	30.9	24	1.7	27
giul	21809.4	702	128.3	570
janos-us	649.2	300	22.8	349
nobel-eu	324.9	272	7.2	226
nobel-germany	10.7	95	0.5	79
nobel-us	11.3	180	0.6	159
norway	1996.2	609	45.7	562
nsf12	16.4	134	0.8	130
nsf48	11.1	87	0.6	111
nsf212	17.7	151	1.2	175
nsf2-48	21.1	148	1.1	168
sun	78.3	112	1.8	230

4.3 Conclusão

Neste capítulo apresentamos primeiramente algumas variações nos modelos existentes para a resolução do max-RWA com o objetivo de simplificar esses modelos e assim poder resolver instâncias um pouco maiores dos que as resolvidas até então. Apresentamos duas variações que chamamos de RWA1 e RWA2. As duas propostas apresentaram um desempenho em termos de tempo de computação superior (menor tempo) aos modelos que usamos para fazer as comparações.

Depois apresentamos uma nova proposta em termos de geração de colunas, também, para a resolução do MAX-RWA. Na verdade, combinamos duas estratégias já apresentadas na literatura, MAX-IRC e MAX-IS, para a resolução dos problemas. Observamos que a estratégia MAX-IS ainda não tinha sido utilizada devido à limitação de se ter a necessidade de que todos os possíveis caminhos entre os pares origem e destino tenham sido gerados para que se possa ter garantia que a solução gerada é ótima. Para tentar resolver esta limitação desenvolvemos uma boa estratégia para a geração de caminhos que chamamos de Path-Gen. Mesmo com a estratégia de geração de caminhos vimos que as vezes o modelo MAX-IS não é capaz de encontrar a solução ótima para o problema original. Sendo assim, surgiu a ideia de combinar os modelos MAX-IS e MAX-IRC, onde em cada iteração temos a opção de gerar variáveis utilizando um ou outro subproblema. No caso, a estratégia usada é de gerar variáveis com o subproblema do MAX-IRC somente quando o subproblema MAX-IS não é capaz de encontrar uma variável de custo reduzido maior que zero. A nova estratégia se mostrou capaz de resolver todas as instâncias testadas em um tempo inferior ao modelo MAX-IRC.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Neste trabalho foi feita uma revisão sobre o problema de roteamento e alocação de comprimentos de onda em redes ópticas WDM. Foram apresentados alguns modelos matemáticos e as principais heurísticas e metaheurísticas propostas na literatura.

Em relação aos métodos heurísticos para a minimização do número de comprimentos de onda foi verificado que o principal método de geração de soluções até então é o *Best Fit Decreasing*, proposto em 2007 por Skorin-Kapov [50]. Como método de refinamento, primeiramente foi desenvolvido um algoritmo baseado na metaheurística VND e também um método híbrido usando a Busca Tabu apresentada por Noronha et al. [45], respectivamente VND-BFD e VND-BT. Estes métodos não apresentaram melhoras significativas com relação aos métodos já apresentados na literatura.

Desenvolvemos então uma nova abordagem, também através do VND, onde o foco é o rearranjo das requisições. Esta nova versão do VND é utilizada como busca local de um procedimento baseado na metaheurística *Iterated Local Search* e desenvolvemos uma perturbação baseada em um problema de atribuição que não afeta o valor da função objetivo. Desenvolvemos quatro variantes desta abordagem que variavam pela maneira de escolher um subgrafo para ser eliminado e na maneira de escolher as requisições para o problema de atribuição na fase de perturbação. Esta abordagem se mostrou muito eficiente e competitiva em comparação com os métodos propostos na literatura, com destaque para a variante $\text{VND}_e\text{-ILS}_{5p}$, na

qual o subgrafo escolhido para eliminação é aquele que possui o maior número de arcos não utilizados e as requisições que utilizam arcos adjacentes à origem ou o destino da requisição que se deseja rearranjar possuem cinco vezes mais chances de serem escolhidas para o problema de atribuição na fase de perturbação. Com esta variante todos os limites superiores das instâncias ainda em aberto foram melhorados e o método foi capaz de resolver todas as instâncias realísticas. Conduzimos testes computacionais com o intuito de estudar cada uma das componentes do VND_e-ILS_{5p} mostrando, assim, a importância de cada uma.

Com relação aos métodos exatos, começamos apresentando dois modelos compactos propostos na literatura para a resolução do problema de roteamento e alocação de comprimentos de onda com o objetivo de maximizar o número de requisições atendidas. Propusemos, então, algumas simplificações sobre estes modelos e mostramos através de testes computacionais que as simplificações não prejudicam os limites encontrados pelos modelos e aceleram a resolução dos problemas. Além disso, apresentamos alguns modelos baseados em geração de colunas presentes na literatura, também para a resolução do RWA com o objetivo de maximizar o número de requisições atendidas. Para finalizar, apresentamos uma nova abordagem para a geração de colunas combinando os modelos já existentes. Nos experimentos computacionais conduzidos, a abordagem proposta é capaz de resolver todas as instâncias em um tempo menor do que o modelo mais eficiente da literatura até então garantindo que a solução encontrada é ótima.

5.2 Trabalhos Futuros

Como trabalhos futuros sugere-se aprofundar o estudo da nova abordagem para geração de colunas proposta e o desenvolvimento de um algoritmo *Branch-and-price* para a resolução do problema. Também sugere-se a implementação de outros métodos heurísticos e o estudo da influência de certas requisições que dificultam a resolução do problema.

Ainda como trabalhos futuros sugerimos o estudo sobre o problema de concentração de tráfego (*traffic grooming*), uma extensão do problema de roteamento e

alocação de comprimentos de onda, que ainda é pouco estudado na literatura.

Bibliografia

[1] R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. Probability distribution of solution time in GRASP: An experimental investigation. *Journal of Heuristics*, 8:343–373, 2002.

[2] R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. TTT plots: a pearl program to create time-to-target plots. *Optimization Letters*, 1:355–366, 2007.

[3] A.C.F. Alvim, F. Glover, and C.C. Ribeiro. A hybrid improvement heuristic for the one-dimensional bin packing problem. *Journal of Heuristics*, 10:205–229, 2004.

[4] D. Banerjee and B. Mukherjee. Practical approaches for routing and wavelength assignment in large all-optical wavelength routed networks. In *Proceedings IEEE Infocom 94*, pages 110–119, Toronto, Canada, 1994.

[5] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):903–908, 1996.

[6] N. Banerjee and S. Sharan. A evolutionary algorithm for solving the single objective static routing and wavelength assignment problem in WDM networks. In *Proceedings of international conference on intelligent sensing and information processing*, pages 13–18, Chennai, India, 2004.

[7] S. Baroni and P. Bayvel. Wavelength requirements in arbitrarily connected wavelength-routed optical networks. *Journal of Lighwave Technology*, 15(2):242–251, 1997.

[8] R.S. Barr and R.A. Patterson. Grooming telecommunication networks. *Optical Networks Magazine*, 2(3):20–23, 2001.

[9] M.S. Borella, J.P. Jue, D. Banerjee, and B. Mukherjee. Optical components for WDM lightwave networks. *Proceedings of the IEEE*, 85(2):1274–1307, 1997.

[10] D. Brelaz. New methods to color the vertices of a graph. *Communications of the ACM*, 256:251–256, 1979.

[11] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high-bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992.

[12] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Reading, 1967.

[13] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill Book Company, New York, USA, second edition, 2003.

[14] S. Corteel, D. Gardy, M. Valencia-Pabon, D. Bath, and A. Denise. The permutation-path coloring problem on trees. *Theoretical Computer Science*, 297:119–143, 2003.

[15] R. Dutta and G.N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Networks*.

[16] T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255:33–50, 2001.

[17] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the MAX-CUT problem. *Optimization Methods and Software*, 7:1033–1058, 2002.

[18] T. Fischer, K. Bauer, and P. Merz. Solving the routing and wavelength assignment problem with a multilevel distributed memetic algorithm. *Memetic Computing*, 1(2):101–123, 2009.

[19] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, USA, 1979.

[20] P.C. Gilmore and R.E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.

[21] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[22] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.

[23] G. Huiban. O problema da reconfiguração nas redes WDM multifibras. Tese de doutorado, Universidade Federal de Minas Gerais, Programa de Pós-Graduação em Ciência da Computação, Belo Horizonte, 2006.

[24] B. Jaumard, C. Meyer, and B. Thiongane. ILP Formulations and Optimal Solutions for the RWA Problem. In *Proceedings of the IEEE GLOBECOM*, volume 3, pages 1918–1924, Dallas, Texas, USA, 2004.

[25] B. Jaumard, C. Meyer, and B. Thiongane. *ILP formulations for the RWA problem for symmetrical systems*. In: P. Pardalos, M. Resende (Eds.), 2006.

[26] B. Jaumard, C. Meyer, and B. Thiongane. Comparison of ILP formulations for the RWA problem. *Optical Switching and Networking*, 4:157–172, 2007.

[27] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. *Discrete Applied Mathematics*, 157:1291–1308, 2009.

[28] Z.K.G. Patrocínio Júnior. Planejamento de topologia virtual com combinação de tráfegos em redes Ópticas multiplexadas por divisão de comprimentos de onda. Tese de doutorado, Universidade Federal de Minas Gerais, Programa de Pós-Graduação em Ciência da Computação, Belo Horizonte, 2005.

[29] E.G. Coffman Jr., J. Csirik, and G. Woeginger. *Bin packing theory.* In: P. Pardalos, M. Resende (Eds.), Oxford University Press, New York, USA, 2002.

[30] E.G. Coffman Jr., M.R. Garey, and D.S. Johnson. *Bin packing approximation algorithms: A survey.* In: D. Hochbaum (Ed.), PWS Publishing Co., Boston, MA, 1996.

[31] J. Kleinberg and A. Kumar. Wavelength conversion in optical networks. *Journal of Algorithms*, 38(1):25–50, 2001.

[32] J.M. Kleinberg. Approximation algorithms for disjoint paths problems. Tese de doutorado, Departament of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, 1996.

[33] R.M. Krishnaswamy and K.N. Sivarajan. Algorithms for routing and wavelength assignment based on solutions of lp-relaxation. *IEEE Communications Letters*, 5(10):435–437, 2001.

[34] R.M. Krishnaswamy and K.N. Sivarajan. Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Transactions on Networking*, 9(2):186–198, 2001.

[35] K. Lee, K. Kang, T. Lee, and S. Park. Optimal routing and wavelength assignment in WDM ring networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2146–2154, 2000.

[36] K. Lee, K. Kang, T. Lee, and S. Park. An optimization approach to routing and wavelength assignment in WDM all-optical mesh networks without wavelength conversion. *ETRI Journal*, 24(2):131–141, 2002.

[37] G. Li and R. Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Proceedings of the First Workshop on Optical Networks*, pages 1–19, Dallas, Texas, USA, 2000.

[38] H.R. Lourenço, O. Martin, and T. Stuetzle. *Iterated Local Search.* In: F. Glover, G. Kochenberger (Eds.), Kluwer Academic Publishers, Norwell, MA, 2002.

[39] P. Manohar, D. Manjunath, and R.K. Shevgaonkar. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *IEEE Communications Letters*, 6(5):211–213, 2002.

[40] A.X. Martins, M.C. de Souza, M.J.F. Souza, and T.A.M. Toffolo. Grasp with hybrid heuristic-subproblem optimization for the multi-level capacitated minimum spanning tree problem. *Journal of Heuristics*, 15:133–151, 2009.

[41] N. Mladenovic and P. Hansen. A variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

[42] C.S.R. Murthy and M. Gurusamy. *WDM optical Networks: Concepts, Design and Algorithms*. Prentice Hall, 2002.

[43] T.F. Noronha, M.G.C. Resende, and C.C. Ribeiro. Efficient implementation of heuristics for routing and wavelength assignment. *Lecture Notes in Computer Science*, 5038:169–180, 2008.

[44] T.F. Noronha, M.G.C. Resende, and C.C. Ribeiro. A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, 50:503–518, 2011.

[45] T.F. Noronha and C.C. Ribeiro. Routing and wavelength assignment by partition colouring. *European Journal of Operational Research*, 171(3):797–810, 2006.

[46] R. Ramaswami and K.N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking*, 3(5):489–501, 1995.

[47] R. Ramaswami and K.N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):840–851, 1996.

[48] M. G. C. Resende and C. C. Ribeiro. A GRASP with pathrelinking for private virtual circuit routing. *Networks*, 41:104–114, 2003.

[49] A.R. Sharafat and O.R. Ma'rouzi. The most congested cutset: deriving a tight lower bound for the chromatic number in the RWA problem. *IEEE Communications Letters*, 8(7):473–475, 2004.

[50] N. Skorin-Kapov. Routing and wavelength assignment in optical networks using bin packing based algorithms. *European Journal of Operational Research*, 177(2):1167–1179, 2007.

[51] W. Spears and K. deJong. On the virtues of parameterized uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236, San Mateo, 1991.

[52] E. Uchoa, T.A.M. Toffolo, M.C. de Souza, A.X. Martins, and R. Fukasawa. Branch-and-cut and hybrid local search for the multi-level capacitated minimum spanning tree problem. *Networks*, To Appear, 2011.

[53] D. West. *Introduction to Graph Theory*. Prentice Hall, 1996.

[54] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proceedings on the 9th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 333–341, San Francisco, California, USA, 1998.

[55] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Network Magazine*, 1:47–60, 2000.