

**FORMULAÇÕES E HEURÍSTICAS PARA O PROBLEMA DE  
ESCALONAMENTO DE CONEXÕES COM MÚLTIPLAS VELOCIDADES  
DE TRANSMISSÃO E EM CANAIS COM LARGURA DE BANDA  
VARIÁVEL**



JOSÉ MAURÍCIO COSTA

**FORMULAÇÕES E HEURÍSTICAS PARA O PROBLEMA DE  
ESCALONAMENTO DE CONEXÕES COM MÚLTIPLAS VELOCIDADES  
DE TRANSMISSÃO E EM CANAIS COM LARGURA DE BANDA  
VARIÁVEL**

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

**ORIENTADOR: THIAGO FERREIRA DE NORONHA  
CO-ORIENTADOR: MARCOS AUGUSTO MENEZES VIEIRA**

Belo Horizonte

Maio de 2019

© 2019, José Maurício Costa  
Todos os direitos reservados

**Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG**

Costa, José Maurício.

C837f      Formulações e heurísticas para o problema de escalonamento de conexões com múltiplas velocidades de transmissão e em canais com largura de banda variável / José Maurício Costa — Belo Horizonte, 2019. xx, 73 f.: il.; 29 cm.

Tese (doutorado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Thiago Ferreira de Noronha  
Coorientador: Marcos Augusto Menezes Vieira

1. Computação – Teses. 2. Otimização combinatória. 3. Programação inteira. 4. Programação heurística. 5. Redes de sensores sem fio. I. Orientador. II. Coorientador. III. Título.

CDU 519.6.\*61(043)



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

## FOLHA DE APROVAÇÃO

Formulações e Heurísticas para o Problema de Escalonamento de Conexões  
com Múltiplas Velocidades de Transmissão e em Canais com Largura de  
Banda Variável

**JOSÉ MAURÍCIO COSTA**

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. THIAGO FERREIRA DE NORONHA - Orientador  
Departamento de Ciência da Computação - UFMG

PROF. MARCOS AUGUSTO MENEZES VIEIRA - Coorientador  
Departamento de Ciência da Computação - UFMG

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

PROFA. OLGA NIKOLAEVNA GOUSSEVSKAIA  
Departamento de Ciência da Computação - UFMG

PROF. MARCONE JAMILSON FREITAS SOUZA  
Departamento de Computação - UFOP

PROF. RICARDO MARTINS DE ABREU SILVA  
Centro de Informática - UFPE

Belo Horizonte, 20 de Maio de 2019.



# Agradecimentos

Agradeço a Deus pelo dom da vida e por ter me mantido sempre de pé para conseguir realizar mais este sonho.

Agradeço os meus pais Adauto e Vilani por todo o amor, carinho e compreensão que tiveram comigo, apesar da distância. Gostaria de destacar a minha gratidão para a minha mãe pela incondicional paciência e compreensão para me ouvir durante todo esse período.

Agradeço o meu orientador Thiago Noronha por todos os conselhos, ensinamentos e orientações que foram importantes tanto para o desenvolvimento da pesquisa quanto para a minha formação. Acredito que os mesmos continuarão a me guiar durante a minha carreira. Agradeço também ao meu orientador Marcos Vieira por todos os ensinamentos e orientações compartilhados que, por sua vez, foram essenciais para o desenvolvimento desta pesquisa. Agradeço também ao meu colega de trabalho Pedro que desde quando começou a participar da pesquisa ajudou com as suas ideias e entusiasmo, contribuindo para que o trabalho se tornasse ainda melhor.

Agradeço aos meus amigos do LAPO pela amizade e apoio que tive durante todos esses anos de estudo e que foram essenciais para a minha caminhada. Agradeço em especial os meus colegas Amadeu, Armando, Bráulio, Iago, Luis Henrique, Marcelo e Mateus por todos os ensinamentos compartilhados e a ajuda que me deram durante esse período. Gostaria de agradecer também aos meus amigos que conheci no departamento Jeferson, Cleice, Évelin, Kattiana e Johnatan por toda a amizade, apoio e conselhos que me ajudaram a crescer pessoalmente e profissionalmente para chegar até aqui. Por fim, agradeço também aos meus amigos Raimundo, Nilmar, Alline e Honório por toda amizade, apoio e lições de vida compartilhadas por vocês que foram importantes para que eu conseguisse chegar até aqui.

Por fim, gostaria de agradecer a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), e a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro.





*“I have no special talent. I am only passionately curious.”*

(Albert Einstein)



# Resumo

O padrão IEEE 802.11ac permite a transmissão de dados em velocidades mais altas do que as admitidas pelos padrões da família IEEE 802.11 anteriores, pois ele adota diversas melhorias como a utilização do método MU-MIMO (do inglês *Multi-User Multiple-Input Multiple-Output*) e o aumento do número de fluxos espaciais para o envio de dados, bem como o uso canais de comunicação com larguras de banda maiores. Neste caso, o padrão IEEE 802.11ac admite canais de comunicação com diferentes larguras de banda, variando de 20 MHz a 160 MHz. Neste trabalho é introduzido o Problema de Escalonamento de Conexões com Múltiplas Velocidades de Transmissão e em Canais com Largura de Banda Variável (VRBSP, do inglês *Variable Rate and Variable Bandwidth Scheduling Problem*), que é uma generalização do clássico Problema de Escalonamento de Conexões com Velocidade de Transmissão Variável (VRSP, do inglês *Variable Rate Scheduling Problem*) em redes sem fio. Duas formulações de Programação Linear Inteira Mista (MILP, do inglês *Mixed Integer Linear Programming*) foram propostas para representar o VRBSP. Estas formulações foram usadas em um *solver* de MILP com o objetivo de encontrar soluções ótimas para o VRBSP em instâncias de pequeno porte. Esta abordagem também pode ser usada para resolver o VRSP. Dado que o VRBSP é NP-Difícil, também são propostas duas heurísticas baseadas nas metaheurísticas BRKGA (do inglês *Biased Random-Key Genetic Algorithm*) e VNS (do inglês *Variable Neighborhood Search*) para encontrar soluções de tal problema próximas do ótimo para instâncias de médio e grande porte. Os experimentos computacionais foram realizados com um conjunto de instâncias da literatura que possuem até 2048 conexões. Eles mostraram que o VNS obteve os melhores resultados dentre as heurísticas e que os algoritmos exatos foram capazes de encontrar soluções ótimas de VRBSP para instâncias com até 256 conexões e soluções ótimas de VRSP para instâncias com até 1024 conexões.

**Palavras-chave:** Problema de Escalonamento em Redes Sem Fio, Programação Inteira, Heurísticas, Otimização Combinatória.



# Abstract

The IEEE 802.11ac standard enables a higher transmission speed than the previous IEEE 802.11 standards, because it implements several improvements like the adoption of the MU-MIMO (Multi-User Multiple-Input Multiple-Output) method and the increasing of the number of spatial streams to send data, as well the utilization of larger communication channels. The IEEE 802.11ac allows the network frequency spectrum to be divided into communication channels with different bandwidths, varying from 20 MHz to 160 MHz. In this work, we introduce the Variable Rate and Variable Bandwidth Scheduling Problem (VRBSP), which is a generalization of the classical Variable Rate Scheduling Problem (VRSP) on wireless networks. We propose two Mixed Integer Linear Programming (MILP) formulations to model the VRBSP. These two formulations are used within a MILP solver to seek optimal VRBSP schedules for small-sized networks. This approach can also be used to solve VRSP. As VRBSP is NP-Hard, we also propose a Biased Random-Key Genetic Algorithm (BRKGA) and a Variable Neighborhood Search (VNS) heuristic to find near optimal VRBSP solutions for medium-sized and large-sized networks. The computational experiments were carried out on classical network instances from the literature with up to 2048 links. They show that the best heuristic results were obtained by VNS, and that the MILP-based exact algorithms were able to find optimal VRBSP schedules for networks with up to 256 links and optimal VRSP schedules for networks with up to 1024 links.

**Keywords:** Wireless Scheduling Problem, Integer Programming, Heuristics, Combinatorial Optimization.



# Lista de Figuras

1.1	Exemplo de rede sem fio com infraestrutura [Tanenbaum & Wetherall, 2011].	2
1.2	Faixas de frequência em que os canais operam conforme o padrão IEEE 802.11ac [Gast, 2013]. . . . .	4
4.1	Ilustração do processo de transição entre consecutivas gerações do algoritmo genético com chaves aleatórias tendencioso. . . . .	25
5.1	Desempenho das heurísticas em relação ao <i>throughput</i> das soluções obtidas utilizando as instâncias do conjunto <i>D250</i> . . . . .	52
5.2	Distribuição média do <i>throughput</i> das soluções obtidas pelas heurísticas como uma função de distribuição acumulada para as instâncias do conjunto <i>D250</i> com 1024 conexões. . . . .	53
5.3	Distribuição média do <i>throughput</i> das soluções obtidas pelas heurísticas como uma função de distribuição acumulada para as instâncias do conjunto <i>D250</i> com 2048 conexões. . . . .	54
5.4	Box plot referente a distribuição do <i>throughput</i> das soluções obtidas pelas heurísticas para as instâncias do conjunto <i>D250</i> com 1024 conexões. . . . .	54
5.5	Box plot referente a distribuição do <i>throughput</i> das soluções obtidas pelas heurísticas para as instâncias do conjunto <i>D250</i> com 2048 conexões. . . . .	55
5.6	Evolução das melhores soluções encontradas pelo BRKGA e o VNS ao longo de 600 segundos de execução para as instâncias do conjunto <i>D250</i> com 1024 conexões. . . . .	56
5.7	Evolução das melhores soluções encontradas pelo BRKGA e o VNS ao longo de 600 segundos de execução para as instâncias do conjunto <i>D250</i> com 2048 conexões. . . . .	57
5.8	Distribuição média do número de canais usados (figura 5.8a) e das conexões ativas (figura 5.8b), de acordo com a largura de banda dos canais, das soluções com 32 conexões obtidas pelas heurísticas propostas. . . . .	58

5.9	Distribuição média do número de canais usados (figura 5.9a) e das conexões ativas (figura 5.9b), de acordo com a largura de banda dos canais, das soluções com 64 conexões obtidas pelas heurísticas propostas. . . . .	58
5.10	Distribuição média do número de canais usados (figura 5.10a) e das conexões ativas (figura 5.10b), de acordo com a largura de banda dos canais, das soluções com 128 conexões obtidas pelas heurísticas propostas. . . . .	58
5.11	Distribuição média do número de canais usados (figura 5.11a) e das conexões ativas (figura 5.11b), de acordo com a largura de banda dos canais, das soluções com 256 conexões obtidas pelas heurísticas propostas. . . . .	59
5.12	Distribuição média do número de canais usados (figura 5.12a) e das conexões ativas (figura 5.12b), de acordo com a largura de banda dos canais, das soluções com 512 conexões obtidas pelas heurísticas propostas. . . . .	59
5.13	Distribuição média do número de canais usados (figura 5.13a) e das conexões ativas (figura 5.13b), de acordo com a largura de banda dos canais, das soluções com 1024 conexões obtidas pelas heurísticas propostas. . . . .	59
5.14	Distribuição média do número de canais usados (figura 5.14a) e das conexões ativas (figura 5.14b), de acordo com a largura de banda dos canais, das soluções com 2048 conexões obtidas pelas heurísticas propostas. . . . .	60



# Lista de Tabelas

1.1	Velocidades de transmissão (Mbps) com que os dados são enviados de acordo com a largura de banda do canal e a qualidade do sinal SINR(dB), conforme o padrão IEEE 802.11ac . . . . .	8
5.1	SINR(dB) e Velocidade (Mbps) para o padrão IEEE 802.11b. . . . .	41
5.2	SINR(dB) e Velocidade (Mbps) para o padrão IEEE 802.11n. . . . .	41
5.3	Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto <i>D10000</i> referentes a variante do VRSP que corresponde ao padrão IEEE 802.11b. . . . .	43
5.4	Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto <i>D250</i> referentes a variante do VRSP que corresponde ao padrão IEEE 802.11b. . . . .	43
5.5	Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto <i>D10000</i> referentes a variante do VRSP que corresponde ao padrão IEEE 802.11n. . . . .	44
5.6	Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto <i>D250</i> referentes a variante do VRSP que corresponde ao padrão IEEE 802.11n. . . . .	44
5.7	Resultados do B&C-F1 e B&C-F2 para o conjunto <i>D10000</i> referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac. . . . .	48
5.8	Resultados do B&C-F1 e B&C-F2 para o conjunto <i>D250</i> referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac. . . . .	48
5.9	Resultados do BRKGA, CH e VNS para o conjunto <i>D10000</i> referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac. . . . .	50
5.10	Resultados do BRKGA, CH e VNS para o conjunto <i>D250</i> referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac. . . . .	50



# Sumário

Agradecimentos	vii
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
<b>1 Introdução</b>	<b>1</b>
1.1 Definição do problema . . . . .	4
1.2 Objetivos e metodologia . . . . .	8
1.3 Roteiro da tese . . . . .	9
<b>2 Trabalhos Relacionados</b>	<b>11</b>
2.1 Problemas com uma única velocidade de transmissão . . . . .	11
2.2 Problemas com múltiplas velocidades de transmissão . . . . .	14
2.3 Problemas com velocidades de transmissão variáveis . . . . .	15
<b>3 Formulações para VRBSP</b>	<b>17</b>
3.1 Formulação não-linear . . . . .	17
3.2 Formulações lineares . . . . .	19
<b>4 Heurísticas para o VRBSP</b>	<b>23</b>
4.1 Algoritmo genético com chaves aleatórias tendencioso . . . . .	23
4.2 Heurística de busca em vizinhanças variáveis . . . . .	29
4.2.1 Heurística construtiva . . . . .	29
4.2.2 Modelagem do espaço de busca do VNS . . . . .	31
4.2.3 VNS . . . . .	32

<b>5 Experimentos Computacionais</b>	<b>39</b>
5.1 Experimentos com algoritmos exatos . . . . .	41
5.1.1 Resultados para o VRSP . . . . .	41
5.1.2 Resultados para o VRBSP . . . . .	46
5.2 Experimentos com heurísticas . . . . .	49
<b>6 Conclusões</b>	<b>63</b>
<b>Referências Bibliográficas</b>	<b>67</b>

# Capítulo 1

## Introdução

As redes sem fio estão se tornando cada vez mais presentes, sendo estas utilizadas em diversos locais como instituições de ensino, centros de pesquisa, residências, e empresas, devido ao constante crescimento do número de dispositivos sem fio que operam em tais redes. Além disso, as aplicações que executam em tais dispositivos demandam velocidades de transmissão cada vez maiores. Portanto, novas tecnologias e padrões de comunicação têm sido desenvolvidos para lidar com estas questões [Cisco, 2012; IEEE Computer Society, 2013].

As redes sem fio com infraestrutura (do inglês, *Infrastructure Wireless Networks*) consistem em um ambiente de rede formado por um dispositivo controlador (do inglês *controller*) e um conjunto de pontos de acesso (APs, do inglês *Access Points*). Os pontos de acesso são os dispositivos responsáveis pela transmissão de dados na rede. A comunicação entre os dispositivos clientes (*laptops, smartphones*, dispositivos móveis, etc) e os APs é direta (*single-hop*) por meio de um protocolo de comunicação, tal como o empregado pelo padrão IEEE 802.11ac. A figura 1.1 ilustra a arquitetura de uma rede sem fio. À esquerda encontram-se dois dispositivos conectados a um AP, enquanto que à direita pode-se observar outros dois dispositivos conectados a um outro AP. Os APs podem ser interconectados, via cabo, a um dispositivo controlador que é

por sua vez conectado à Internet [Tanenbaum & Wetherall, 2011]. Este controlador é responsável por gerenciar os pontos de acesso. Desta forma, por meio de um dispositivo central, pode-se configurar o ponto de acesso para operar sobre uma rede sem fio de forma que ele se comunique com os demais dispositivos. Atualmente, a maioria das redes locais sem fio são redes com infraestrutura. Redes locais sem fio que operam deste modo são muito comuns em universidades, hotéis, empresas, e fábricas [Moura et al., 2015]. Assim sendo, o uso do controlador como dispositivo central possibilita o desenvolvimento de uma abordagem centralizada que foca na resolução do problema de se configurar um conjunto de pontos de acesso.

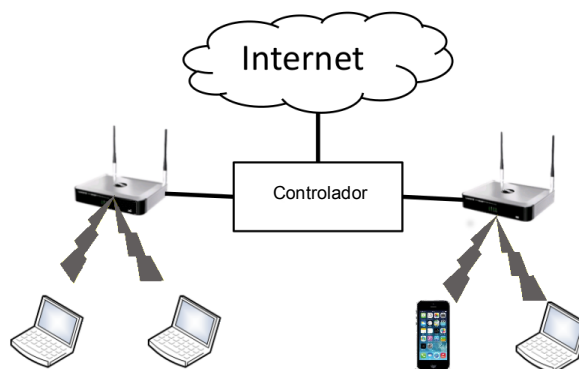


Figura 1.1: Exemplo de rede sem fio com infraestrutura [Tanenbaum & Wetherall, 2011].

Em redes de comunicação sem fio, uma conexão pode ser modelada por um par de dispositivos, sendo um dispositivo transmissor e outro receptor. Como o meio de transmissão é compartilhado, se duas conexões transmitem ao mesmo tempo, usando faixas de frequência do espectro eletromagnético que se sobrepõem, o transmissor de uma delas pode interferir significativamente sobre o sinal que chega ao receptor da outra. Quanto menor é o nível de interferência, maior é a qualidade do sinal que chega ao receptor. Além disso, quanto maior é a qualidade do sinal, mais rápida é a velocidade com que a conexão transmite. Por outro lado, o nível de interferência pode ser tão alto que a qualidade do sinal diminui ao ponto de que o receptor não seja capaz de decodificar o sinal [Goussevskaia et al., 2012].

O padrão de comunicação em redes Wi-Fi mais recente é o IEEE 802.11ac [Ong et al., 2011]. Este padrão permite a transmissão de dados em velocidades mais altas do que as admitidas pelos padrões anteriores IEEE 802.11b [Network, 2008] e IEEE 802.11n [Henry, 2010], pois ele adota diversas melhorias como a utilização do método MU-MIMO (do inglês *Multi-User Multiple-Input Multiple-Output*) e o aumento do número de fluxos espaciais para o envio de dados, bem como o uso canais de comunicação com larguras de banda maiores [Gast, 2013]. Neste caso, o padrão IEEE 802.11ac possibilita a divisão da faixa de frequência usada pela rede em canais de comunicação com diferentes larguras de banda, variando de 20 MHz a 160 MHz.

Nem todas as partições da faixa de frequência de transmissão em canais são permitidas pelo padrão IEEE 802.11ac. Por exemplo, nas figuras 1.2a, 1.2b e 1.2c são ilustradas as faixas de frequência em que os canais da rede podem operar de acordo com o padrão de comunicação IEEE 802.11ac. Na figura 1.2a é mostrada a faixa de frequência, que está entre 5.170 MHz e 5.330 MHz. Nela podem ser utilizados, sem que ocorra sobreposição, até 8 canais com largura de banda igual a 20 MHz (canais 1 a 8), 4 canais com largura de banda igual a 40 MHz (canais 26 a 29), 2 canais com largura de banda igual a 80 MHz (canais 38 e 39), ou um único canal com largura de banda igual a 160 MHz (canal 44). Na figura 1.2b é mostrada a faixa de frequência que está entre 5.490 MHz e 5.730 MHz. Nela podem ser utilizados, sem que ocorra sobreposição, até 12 canais com largura de banda igual a 20 MHz (canais 9 a 20), 6 canais com largura de banda igual 40 MHz (canais 30 a 35), 3 canais com largura de banda igual a 80 MHz (canais 40, 41 e 42) MHz, ou um único canal com largura de banda igual a 160 MHz (canal 45). Na figura 1.2c é mostrada a faixa de frequência que está entre 5.735 MHz e 5.835 MHz. Nela podem ser utilizados, sem que ocorra sobreposição, até 5 canais com largura de banda igual a 20 MHz (canais 21 a 25), 2 canais com largura de banda igual a 40 MHz (canais 36 e 37), ou um único canal com largura de banda igual a 80 MHz (canal 43). Sendo assim, algoritmos de escalonamento para estas redes devem ser capazes de decidir simultaneamente como particionar o espectro eletromagnético em

canais e como escalonar as conexões para estes canais.

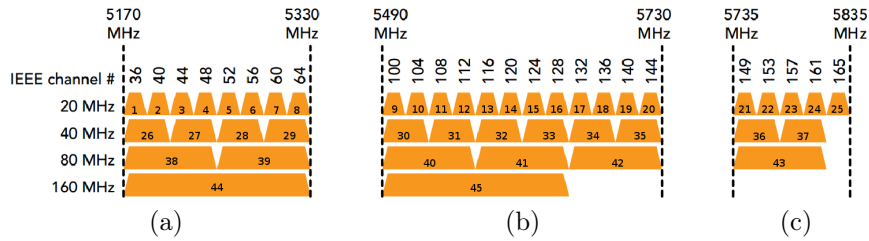


Figura 1.2: Faixas de frequência em que os canais operam conforme o padrão IEEE 802.11ac [Gast, 2013].

## 1.1 Definição do problema

Seja  $V = \{1, \dots, n\}$  o conjunto de dispositivos da rede, e  $L$  o conjunto de conexões, onde cada conexão  $(i, j) \in L$  é caracterizada por um dispositivo *transmissor*  $i \in V$  e um dispositivo *receptor*  $j \in V$ . Seja também  $B$  o conjunto de larguras de banda (em MHz). No caso do padrão IEEE 802.11ac [IEEE Computer Society, 2013], temos que  $B = \{20, 40, 80, 160\}$ . Além disso, seja  $C$  o conjunto de subdivisões do espectro eletromagnético em canais de comunicação, cuja largura de banda é uma das definidas em  $B$ . Estes canais podem estar sobrepostos uns aos outros. Portanto, também é definido o conjunto  $O^c \subseteq C$  com os canais que se sobrepõem a  $c \in C$  (incluindo ele mesmo). No caso do padrão IEEE 802.11ac, temos que  $C = \{1, \dots, 45\}$  e  $O^c$ , para todo  $c \in C$ , é dado pela figura 1.2 [Gast, 2013]. Por exemplo,  $O^{32} = \{13, 14, 32, 41, 45\}$  porque o canal 32 se sobrepõe aos canais 13, 14, 41, 45, e ele mesmo. Além disso, também define-se  $C^b \subseteq C$  como o subconjunto de canais cuja largura de banda é  $b \in B$ , e  $B^c \in B$  como a largura de banda do canal  $c \in C$ . Por exemplo,  $B^{44} = B^{45} = 160$ . Como estes são os únicos canais com largura de banda iguais a 160 MHz, temos que  $C^{160} = \{44, 45\}$ .

O Problema de Escalonamento de Conexões com Múltiplas Velocidades de Transmissão e em Canais com Largura de Banda Variável (VRBSP, do inglês *Variable Rate*



*and Variable Bandwidth Scheduling Problem*) em redes sem fio consiste em selecionar um subconjunto de conexões  $L$ , que transmitirão no mesmo intervalo de tempo, e atribuir cada conexão a um canal pertencente a  $C$ , tal que (i) duas conexões que compartilham um mesmo dispositivo não podem transmitir simultaneamente, e (ii) a interferência no receptor de cada conexão escalonada deve ser pequena o suficiente para que a mensagem transmitida possa ser decodificada. Uma solução para VRBSP é representada por um conjunto  $S \subseteq L \times C$ , onde  $\langle(i, j), c\rangle \in S$  denota que cada conexão  $(i, j) \in L$  é escalonada e atribuída a um canal  $c \in C$ . Por uma questão de simplicidade, denota-se também por  $L(S)$  o conjunto de *conexões escalonadas*, isto é,  $L(S) = \{(i, j) \in L : \exists \langle(i, j), c\rangle \in S\}$ . O objetivo do VRBSP é encontrar uma solução em que o *throughput* da rede é o máximo, isto é, o escalonamento em que o somatório das velocidades de transmissão das conexões escalonadas é maximizado. Assim sendo, esta tese tem como objetivo estudar e tratar o problema descrito acima, tendo como foco o padrão IEEE 802.11ac.

Neste trabalho é adotado o modelo SINR (SINR, do inglês *Signal to Interference plus Noise Ratio*) [Gupta & Kumar, 2000] para modelar a interferência e a qualidade do sinal que chegam no receptor de cada conexão  $\langle(i, j), c\rangle \in S$ . A interferência que ocorre no receptor da conexão  $\langle(i, j), c\rangle \in S$ , causada pelas demais conexões escalonadas que foram atribuídas a canais que se sobrepõem a  $c$ , é dada pela equação (1.1). Neste caso, os dispositivos transmissor e receptor da conexão  $\langle(i, j), c\rangle \in S$  são denotados por  $i \in L$  e  $j \in L$ , respectivamente. O conjunto de canais que se sobrepõem ao canal  $c \in C$  onde  $\langle(i, j), c\rangle \in S$  transmite é dado por  $O^c$ ,  $d_{uj}$  é a distância (em metros) entre o transmissor  $u$  da conexão  $\langle(u, v), \bar{c} \in O^c\rangle \in S \setminus \{\langle(i, j), c\rangle\}$  e o receptor  $j$ ,  $P$  é a potência de transmissão (em Watts), e  $\alpha \in \mathbb{R}$  é o fator de atenuação do meio [Rappaport, 2002], cujo valor tipicamente varia no intervalo  $2 < \alpha \leq 6$  [Goussevskaia et al., 2012; Gupta & Kumar, 2000; Goussevskaia et al., 2009]. Assim como o definido em [Goussevskaia et al., 2016, 2009; Chen & Chen, 2015], assume-se que não há interferência entre conexões atribuídas a canais adjacentes, e que todos os dispositivos transmissores enviam dados

com o mesmo nível de potência  $P$  (em Watts). Dado que a potência dos transmissores é uniforme, este trabalho não considera o problema de controle de potência. A qualidade do sinal que chega no receptor de cada conexão  $\langle(i, j), c\rangle \in S$  é denotada por  $SINR_{ij}$  e, junto com o valor de  $B^c$ , determina a velocidade de transmissão de  $(i, j)$ . O valor de  $SINR_{ij}$  é inversamente proporcional ao valor da interferência  $I_{ij}$ , que ocorre no receptor de  $(i, j)$ . Portanto, o valor de  $SINR_{ij}$  (em Decibéis) em uma solução  $S$  pode ser computado pela equação (1.2) [Gupta & Kumar, 2000], onde  $d_{ij}$  é a distância entre  $i \in L$  e  $j \in L$  (em metros), e  $N$  é o ruído ambiente (em Watts) [Gupta & Kumar, 2000], cujo valor tipicamente varia no intervalo  $0 \leq N \leq 8 \cdot 10^{-14}$  [Goussevskaia et al., 2012, 2016; Halldórsson & Mitra, 2012; Vieira et al., 2016].

$$I_{ij} = \sum_{\langle(u,v), \bar{c} \in O^c\rangle \in S \setminus \{\langle(i,j), c\rangle\}} \frac{P}{(d_{uj})^\alpha}, \forall \langle(i, j), c\rangle \in S \quad (1.1)$$

$$SINR_{ij} = \frac{\frac{P}{(d_{ij})^\alpha}}{I_{ij} + N}, \forall \langle(i, j), c\rangle \in S \quad (1.2)$$

No padrão IEEE 802.11ac, a velocidade de transmissão  $r_{ij}$  de uma conexão escalonada  $\langle(i, j), c\rangle \in S$  é computada de acordo com os valores de  $SINR_{ij}$  e  $B^c$ . Dados  $SINR_{ij}$  e  $B^c$ , pode-se computar qual é o esquema de modulação e codificação (MCS, do inglês *Modulation and Coding Scheme*) [Cisco, 2012] que resulta na maior velocidade de transmissão possível para  $(i, j)$ , ou seja  $r_{ij}$ . O padrão IEEE 802.11ac possui dez MCS [IEEE Computer Society, 2013]. Seja  $M = \{0, \dots, 9\}$  o conjunto de identificadores de MCS, denota-se por  $q_m^b$  o SINR mínimo que uma conexão deve ter para que ela seja capaz de transmitir com uma velocidade de transmissão  $\bar{r}_m^b$  usando um canal com largura de banda  $b \in B$  e o MCS  $m \in M$ . Os valores de  $q_m^b$  e  $\bar{r}_m^b$ , referentes ao padrão IEEE 802.11ac, são dados pela tabela 1.1. A coluna 1 identifica os MCS, e a coluna 2 apresenta o valor mínimo de SINR  $q_m^b$  que é necessário para transmitir com uma velocidade de transmissão  $\bar{r}_m^b$  (mostrado na coluna 3) usando um canal com largura de

banda  $b = 20$  e o MCS  $m \in M$ . A última linha das colunas 2 e 3 está preenchida com '—', porque a maior velocidade de transmissão com que uma conexão pode transmitir em um canal com largura de banda igual a 20 MHz é 78,0 Mbps. Neste caso, o MCS 9 não é usado pelas conexões que transmitem em canais com largura de banda igual a 20 MHz. No entanto, por questão de simplicidade e sem perda de generalidade, assume-se que  $q_9^{20} = q_8^{20} = 29,0$  e  $\bar{r}_9^{20} = \bar{r}_8^{20} = 78,0$ . Os mesmos dados são apresentados para canais com largura de banda igual a 40 MHz nas colunas 4 e 5, para canais com largura de banda igual 80 MHz nas colunas 6 e 7, e para canais com largura de banda igual 160 MHz nas colunas 8 e 9, respectivamente. Por exemplo, de acordo com a tabela 1.1, uma conexão  $\langle (i, j), c \rangle \in S$ , com  $B^c = 80$ , que tem  $SINR_{ij} = 21,0$  dB transmite a uma velocidade de 175,5 Mbps. Se  $SINR_{ij} \geq 37,0$  dB a mesma conexão transmite a uma velocidade de 390,0 Mbps. No entanto, se  $SINR_{ij} < 8,0$  dB o receptor desta conexão não é capaz de decodificar a mensagem, e esta conexão não pode ser escalonada. Sendo assim, dados  $SINR_{ij}$  e  $B^c$ , a velocidade de transmissão de  $\langle (i, j), c \rangle \in S$  é dada pela equação (1.3).

$$r_{ij} = \max_{m \in M: SINR_{ij} \geq q_m^{B^c}} \bar{r}_m^{B^c} \quad (1.3)$$

Portanto, VRBSP pode ser formalmente definido como na equação (1.4),

$$S^* = \operatorname{argmax}_{S \in \Delta} \sum_{(i,j) \in L(S)} r_{ij} \quad (1.4)$$

onde  $\Delta \subset 2^{L \times C}$  é o conjunto de todas as *soluções viáveis*, de forma que uma solução  $S \in \Delta$  é viável se (i)  $i \neq j \neq u \neq v$  para todo par  $\langle (i, j), c \rangle, \langle (u, v), \bar{c} \rangle \in S$ , e (ii)  $SINR_{ij} \geq \min_{m \in M} q_m^{B^c}$  para todo  $\langle (i, j), c \rangle \in S$ .

Quando  $|B| = |C| = 1$ , VRBSP se reduz ao Problema de Escalonamento de Conexões com Velocidade de Transmissão Variável em redes sem fio (VRSP, do inglês *Variable Rate Scheduling Problem*) [Kesselheim, 2012], que por sua vez é NP-Difícil. Sendo assim, VRBSP também é NP-Difícil.

Tabela 1.1: Velocidades de transmissão (Mbps) com que os dados são enviados de acordo com a largura de banda do canal e a qualidade do sinal SINR(dB), conforme o padrão IEEE 802.11ac

MCS Index	20 MHz		40 MHz		80 MHz		160 MHz	
	SINR $q_m^{20}$	Velocidade $\bar{r}_m^{20}$	SINR $q_m^{40}$	Velocidade $\bar{r}_m^{40}$	SINR $q_m^{80}$	Velocidade $\bar{r}_m^{80}$	SINR $q_m^{160}$	Velocidade $\bar{r}_m^{160}$
0	2,0	6,5	5,0	13,5	8,0	29,3	11,0	58,5
1	5,0	13,0	8,0	27,0	11,0	58,5	14,0	117,0
2	9,0	19,5	12,0	40,5	15,0	87,8	18,0	175,5
3	11,0	26,0	14,0	54,0	17,0	117,0	20,0	234,0
4	15,0	39,0	18,0	81,0	21,0	175,5	24,0	351,0
5	18,0	52,0	21,0	108,0	24,0	234,0	27,0	468,0
6	20,0	58,5	23,0	121,5	26,0	263,3	29,0	526,5
7	25,0	65,0	28,0	135,0	31,0	292,5	34,0	585,0
8	29,0	78,0	32,0	162,0	35,0	351,0	38,0	702,0
9	—	—	34,0	180,0	37,0	390,0	40,0	780,0

## 1.2 Objetivos e metodologia

O VRBSP é um problema de otimização combinatória NP-Difícil [Cormen, 2009] e, nesse caso, não são conhecidas técnicas de desenvolvimento de algoritmos capazes de encontrar soluções ótimas para problemas como esse em tempo polinomial. Problemas NP-Difíceis são geralmente solucionados por meio de técnicas baseadas em algoritmos exatos [Wolsey, 1998] e heurísticas [Talbi, 2009; Dorigo & Stützle, 2004]. Os algoritmos exatos conseguem garantir que as soluções encontradas sejam ótimas. Entretanto, o tempo computacional gasto é exponencial no pior caso [Hertz & Widmer, 2003]. Já os algoritmos heurísticos são capazes de obter soluções de boa qualidade em tempo polinomial, mas sem a garantia de otimalidade.

O número de dispositivos que operam em redes sem fio é cada vez maior e as aplicações que são executadas por eles demandam velocidades de transmissão cada vez mais altas. Essa crescente demanda faz com que seja necessário o constante desenvolvimento de tecnologias para a melhoria do *throughput* da rede com o objetivo de garantir a qualidade da transmissão de dados nela. Porém, à medida que cresce o número de conexões em uma rede sem fio, também aumenta a interferência que ocorre entre elas. Quando a quantidade de conexões que estão transmitindo na rede é muito grande, a interferência pode ser alta o suficiente para que o receptor de cada conexão não seja capaz de decodificar os dados que são enviados para ele. Além disso, o número de

canais para os quais as conexões são atribuídas com a finalidade de se minimizar os problemas de interferência é limitado.

Portanto, dados os problemas descritos acima e o fato de que o VRBSP é NP-Difícil, esta tese tem como objetivo propor algoritmos exatos e heurísticos que sejam capazes de solucionar este problema que considera canais com largura de banda variáveis. Para tanto, são propostas duas formulações de Programação Linear Inteira Mista (MILP, do inglês *Mixed Integer Linear Programming*) para o VRBSP, que são usadas em um resolvidor de MILP com o intuito de encontrar soluções ótimas para o VRBSP. Como o VRSP é um caso especial do VRBSP, esta abordagem pode ser aplicada também a VRSP. Além disso, são propostas duas heurísticas baseadas nas metaheurísticas BRKGA (do inglês, *Biased Random-Key Genetic Algorithm*)[Gonçalves & Resende, 2011] e VNS (do inglês *Variable Neighborhood Search*)[Mladenović & Hansen, 1997] para encontrar soluções do VRBSP próximas do ótimo, quando comparadas as soluções ótimas obtidas por meio de algoritmos exatos.

### 1.3 Roteiro da tese

Esta tese está organizada da seguinte forma. Os trabalhos relacionados são apresentados no capítulo 2. No capítulo 3 são propostas duas formulações de Programação Linear Inteira Mista para o VRBSP. No capítulo 4 são descritas as heurísticas BRKGA e VNS propostas para o VRBSP. Os experimentos computacionais que foram realizados com os algoritmos exatos e heurísticos propostos são discutidos no capítulo 5. Por fim, as considerações finais deste trabalho são apresentadas no capítulo 6.



# Capítulo 2

## Trabalhos Relacionados

Gupta & Kumar [2000] formalizaram dois modelos de interferência em redes sem fio: o modelo de protocolos e o SINR (SINR, do inglês *Signal to Interference plus Noise Ratio*). Eles desenvolveram experimentos para analisar o nível de precisão destes modelos para modelar a interferência e a qualidade do sinal que chegam no receptor de cada conexão na rede. Os resultados obtidos mostraram que o modelo de interferência física é mais preciso que o modelo de protocolos. Portanto, este trabalho e os trabalhos relacionados discutidos a seguir focam no modelo SINR.

### 2.1 Problemas com uma única velocidade de transmissão

Ephremides & Truong [1990] propuseram o primeiro problema de escalonamento de conexões em redes sem fio onde a qualidade do sinal é representada pelo modelo SINR. Neste problema, todas as conexões enviam dados com a mesma velocidade de transmissão e compartilham o mesmo canal de comunicação. O Problema de Escalonamento de Conexões em redes sem fio (WSP, do inglês *Wireless Scheduling Problem*) consiste em particionar o conjunto de conexões  $L$ , na menor quantidade possível de subconjuntos,

tal que  $SINR_{ij} \geq \beta$ , para cada conexão  $(i, j) \in L$ , onde  $\beta$  é a qualidade mínima do sinal requerida para que os dados enviados sejam decodificados. As conexões que estão no mesmo subconjunto são escalonadas para o mesmo intervalo de tempo. Uma prova de que WSP é NP-Difícil foi apresentada em [Goussevskaia et al., 2007].

Os primeiros trabalhos para WSP focaram em algoritmos aproximativos. Brar et al. [2006] propuseram o algoritmo GreedyPhysical, cujo fator de aproximação é  $O\left(n^{1-\frac{2}{\psi(\alpha)+\epsilon}} \cdot (\log(n))^{\frac{2}{\psi(\alpha)+\epsilon}}\right)$ , onde  $\alpha$  é o fator de atenuação,  $\psi(\alpha) = \frac{1}{2} + \frac{\sqrt{9\alpha-20\alpha+4}}{2(\alpha-2)}$ , e  $\epsilon > 0$  é um pequeno valor arbitrário. Experimentos computacionais mostraram que GreedyPhysical obteve soluções até três vezes melhores que a abordagem proposta por Riley [2003] que usa simulação. Em [Goussevskaia et al., 2007], foi proposto o algoritmo ApproxDiversity, cujo fator de aproximação é  $O(\log \frac{d^{max}}{d^{min}})$ , onde  $d^{max}$  é a maior distância entre o transmissor e o receptor de uma conexão e  $d^{min}$  é a menor distância entre o transmissor e o receptor de uma conexão. Goussevskaia et al. [2009] propuseram outro algoritmo, chamado ApproxLogN, cujo fator de aproximação é  $O(\log n)$ . Os experimentos computacionais realizados neste trabalho, em redes com até 12800 conexões, mostraram que ApproxLogN teve um desempenho médio melhor do que GreedyPhysical [Brar et al., 2006] e ApproxDiversity [Goussevskaia et al., 2007].

Djukic & Valaee [2009] propuseram uma variação de WSP, onde cada conexão ocupa um ou mais intervalos de tempo contíguos. Neste trabalho, os autores propuseram uma heurística que divide o problema em duas partes. Primeiramente, decide-se a ordem em que cada conexão é escalonada. Em seguida, decide-se em qual intervalo de tempo cada conexão começa a transmitir. O primeiro subproblema é modelado por meio de uma formulação de programação linear inteira, e resolvido na otimalidade usando um algoritmo de *branch and bound*. Além disso, os autores mostraram que, dada uma ordem qualquer para as conexões, o segundo subproblema pode ser resolvido aplicando o algoritmo de Bellman-Ford [Bellman, 1958; Ford Jr, 1956] em um grafo de conflitos. Experimentos computacionais mostraram que o tempo computacional requerido para resolver o primeiro subproblema é proibitivo. No entanto, este



subproblema pode ser resolvido em tempo polinomial em redes cuja topologia é uma árvore. Chilukuri & Sahoo [2015] estenderam esta técnica e propuseram uma heurística para resolver este primeiro subproblema. O segundo subproblema é resolvido da mesma forma que em Djukic & Valaee [2009]. Além de ser eficiente para outras topologias, experimentos computacionais em redes com até 10 conexões mostraram que esta nova abordagem encontra soluções tão boas quanto aquelas obtidas pela heurística de Djukic & Valaee [2009].

Vieira et al. [2016] propuseram uma heurística para resolver uma variante de WSP em que cada conexão é alocada para exatamente  $\tau$  intervalos de tempo distintos. A cada iteração, a heurística seleciona uma conexão de  $L$  e insere esta no primeiro intervalo de tempo onde ela e as outras conexões que já foram alocadas para aquele intervalo de tempo podem transmitir simultaneamente. Se a conexão não pode ser inserida em nenhum dos intervalos de tempo já utilizados, a conexão é alocada para um novo intervalo de tempo. Este procedimento é repetido  $\tau$  vezes, de forma que ao final, cada conexão é alocada a exatamente  $\tau$  intervalos de tempo. Experimentos computacionais em redes com até 12800 conexões mostraram que a heurística proposta foi capaz de encontrar soluções melhores na média do que as adaptações de GreedyPhysical [Brar et al., 2006] e ApproxLogN [Goussevskaia et al., 2009].

Goussevskaia et al. [2007] propuseram o Problema de Escalonamento de Conexões com Um Intervalo de Tempo em redes sem fio (SSSP, do inglês *Single-Shot Scheduling Problem*). Este problema consiste em escalonar um subconjunto  $S \subseteq L$  de conexões em um único intervalo de tempo, tal que o número de conexões que transmitem neste intervalo é maximizado e que  $SINR_{ij} \geq \beta$ , para toda conexão  $(i, j) \in S$ . Uma prova de que SSSP é NP-Difícil foi também apresentada em [Goussevskaia et al., 2007].

Em [Goussevskaia et al., 2007], um algoritmo para SSSP foi desenvolvido cujo fator de aproximação é igual a  $O(\log \frac{d^{max}}{d^{min}})$ . Já em [Goussevskaia et al., 2009] um algoritmo foi proposto para o mesmo problema com fator de aproximação igual a

$O((2 \cdot c + 1)^\alpha + (3^{\alpha+1} \cdot 5) + 1)$ , onde  $c = \max\left(2, \left(2^5 \cdot 3^2 \cdot \beta \cdot \left(\frac{\alpha-1}{\alpha-2}\right)\right)^{\frac{1}{\alpha}}\right)$ . Andrews & Dinitz [2009] estudaram uma variação de SSSP onde os transmissores operam com diferentes níveis de potência. Eles propuseram um algoritmo cujo fator de aproximação é igual a  $O(\log \frac{d^{max}}{d^{min}})$ . Outro algoritmo para esta variação de SSSP foi proposto por Halldórsson & Mitra [2011] cujo fator de aproximação é  $O(\log |L| + \log \log \frac{d^{max}}{d^{min}})$ . Em [Kesselheim, 2011] foi proposto o primeiro algoritmo para SSSP cujo fator de aproximação  $O(\log |L|)$  varia apenas com a quantidade conexões.

## 2.2 Problemas com múltiplas velocidades de transmissão

Goussevskaia et al. [2012] propuseram o Problema de Escalonamento de Conexões com Múltiplas Velocidades de Transmissão em redes sem fio (MRSP, inglês *Multi-Rate Scheduling Problem*). Neste problema, a velocidade de transmissão pode ser diferente de uma conexão para outra. Portanto, cada conexão  $(i, j) \in L$  possui um requisito  $\beta_{ij}$  diferente para a qualidade do seu sinal. MRSP consiste em escalonar um subconjunto  $S \subseteq L$  de conexões em um único intervalo de tempo, de forma que o *throughput* da rede neste intervalo é maximizado e que  $SINR_{ij} \geq \beta_{ij}$ , para todo  $(i, j) \in S$ . Este problema é uma generalização de SSSP [Goussevskaia et al., 2007], e conseqüentemente também é NP-Difícil.

Goussevskaia et al. [2012] propuseram um esquema de aproximação polinomial (PTAS, do inglês *Polynomial-Time Approximation Scheme*) [Arora et al., 1999], denominado de DISK-MRS. O fator de aproximação deste algoritmo é  $(1 + \frac{1}{K-1})^2$ , onde  $K > 1$  é uma constante. Experimentos computacionais, em redes com até 2048 conexões, mostraram que DISK-MRS obteve resultados melhores do que uma adaptação do algoritmo AproxDiversity [Goussevskaia et al., 2007, 2009] para MRSP.

Halldórsson & Mitra [2012] propuseram uma heurística baseada em uma formula-

ção de programação linear inteira para o MRSP. Este algoritmo consiste em (i) resolver a relaxação linear da formulação proposta, (ii) arredondar as variáveis fracionárias e (iii) construir uma solução inteira viável para o problema. Experimentos computacionais em redes com até 600 conexões mostraram que o algoritmo proposto obteve resultados equivalentes ou até melhores que uma adaptação da heurística de Halldórsson & Mitra [2011] para MRSP.

## 2.3 Problemas com velocidades de transmissão variáveis

Kesselheim [2012] propôs o Problema de Escalonamento de Conexões com Velocidade de Transmissão Variável em redes sem fio (VRSP, do inglês *Variable Rate Scheduling Problem*). Este problema é diferente do MRSP em que cada conexão já possui uma velocidade de transmissão fixa associada. Seja  $R$  o conjunto de velocidades de transmissão com as quais cada conexão pode transmitir. No VRSP é necessário escolher para cada conexão uma velocidade de transmissão  $r_{ij} \in R$  com a qual ela deve transmitir. Portanto, o VRSP consiste em selecionar um único subconjunto de conexões  $S \subseteq L$  e uma velocidade de transmissão  $r_{ij} \in R$  para cada conexão  $(i, j) \in S$  de forma que o *throughput* da rede é maximizado e que  $SINR_{ij} \geq \beta(r_{ij})$  para todo  $(i, j) \in S$ , onde  $\beta(r_{ij})$  é a qualidade mínima do sinal requerida para transmitir em uma velocidade  $r_{ij} \in R$ . Quando  $|R| = 1$ , VRSP se reduz a SSSP. Conseqüentemente, este problema também é NP-Difícil.

Kesselheim [2012] propôs um algoritmo para VRSP cujo fator de aproximação é igual a  $O(\log |L|)$ . Ásgeirsson et al. [2017] propuseram um *framework* para resolver vários problemas de escalonamento em redes sem fio. Este *framework*, quando aplicado ao VRSP, resulta em um algoritmo cujo fator de aproximação é igual a  $O(\log \log \frac{r^{max}}{r^{min}})$ , onde  $r^{max} = \max_{r' \in R} r'$  e  $r^{min} = \min_{r' \in R} r'$ .

Goussevskaia et al. [2016] mostraram que VRSP pode ser modelado como um grafo de conflitos e propuseram um esquema de aproximação polinomial. Este algoritmo, denominado Datarate PPTAS, é uma extensão do algoritmo DISK-MRS [Goussevskaia et al., 2012]. O fator de aproximação deste algoritmo é igual a  $(1 + \frac{1}{K-1})^2$ , onde  $K > 1$  é uma constante. No entanto, a complexidade computacional deste algoritmo é  $n^{O(K^2)}$ , o que torna a sua execução impraticável para redes de médio e grande porte. Experimentos computacionais em redes com até 32 conexões mostraram que o Datarate PPTAS obteve resultados melhores do que as adaptações de AproxDiversity [Goussevskaia et al., 2007, 2009] e DISK-MRS [Goussevskaia et al., 2012] para VRSP.

O Problema de Escalonamento de Conexões com Múltiplas Velocidades de Transmissão e em Canais com Largura de Banda Variável em redes sem fio (VRBSP), definido na seção 1.1, é uma generalização do VRSP. Quando  $|B| = |C| = 1$ , o VRBSP se reduz ao VRSP. Portanto, este problema também é NP-Difícil. Até onde se sabe, não há trabalhos na literatura que estudaram a aplicação de algoritmos exatos e heurísticos para esta versão do problema que considera canais com largura de banda variáveis. Portanto, esta tese propõe as primeiras formulações baseadas em programação linear inteira mista e os primeiros algoritmos heurísticos para este problema.

# Capítulo 3

## Formulações para VRBSP

Neste capítulo são apresentadas duas formulações baseadas em programação linear inteira mista para o VRBSP. Na seção 3.1, é apresentada uma formulação não-linear inteira mista, já que a relação entre a interferência e o SINR de uma conexão é intrinsecamente não-linear. Neste caso, a interferência e o SINR foram formulados de acordo com o modelo SINR [Gupta & Kumar, 2000; Goussevskaja et al., 2012, 2016]. Em seguida, na seção 3.2, são descritas duas abordagens para linearizar esta formulação.

### 3.1 Formulação não-linear

Dada uma instância de VRBSP que por sua vez é caracterizada pela tupla  $\langle V, L, B, C, O^c, C^b, M, q_s^b, \bar{r}_s^b, d_{ij}, \alpha, N \rangle$ , VRBSP pode ser modelado pelas variáveis  $x_{ij}^c \in \{0, 1\}$ , onde  $x_{ij}^c = 1$  se a conexão  $(i, j) \in L$  é escalonada e atribuída ao canal  $c \in C$ , e  $x_{ij}^c = 0$  caso contrário. Também são definidas as variáveis auxiliares  $y_{ij}^{bm} \in \{0, 1\}$ , para toda conexão  $(i, j) \in L$ , largura de banda  $b \in B$ , e identificador  $m \in M$ , tal que  $y_{ij}^{bm} = 1$  se a velocidade de transmissão de  $(i, j)$  é  $\bar{r}_m^b$ , e  $y_{ij}^{bm} = 0$  caso contrário. Além disso, também são usadas as variáveis auxiliares  $I_{ij} \in \mathbb{R}$ , para todo  $(i, j) \in L$ , que armazenam a interferência em  $j$ . Portanto, uma formulação não-linear para VRBSP é dada por (3.1)-(3.6).

$$\text{maximize } F(x) = \sum_{(i,j) \in L} \sum_{b \in B} \sum_{m \in M} \bar{r}_m^b \cdot y_{ij}^{bm} \quad (3.1)$$

Sujeito a:

$$\sum_{(i,j) \in L} \sum_{c \in C} x_{ij}^c + \sum_{(j,i) \in L} \sum_{c \in C} x_{ji}^c \leq 1, \quad \forall i \in V \quad (3.2)$$

$$\sum_{m \in M} y_{ij}^{bm} \leq \sum_{c \in C^b} x_{ij}^c, \quad \forall (i,j) \in L, b \in B \quad (3.3)$$

$$I_{ij} = \sum_{(u,v) \in L \setminus \{(i,j)\}} \frac{P}{(d_{uj})^\alpha} \cdot \left( \sum_{c \in C} \sum_{\bar{c} \in O^c} x_{ij}^c \cdot x_{uv}^{\bar{c}} \right), \quad \forall (i,j) \in L \quad (3.4)$$

$$\frac{\frac{P}{(d_{ij})^\alpha}}{I_{ij} + N} \geq \sum_{b \in B} \sum_{m \in M} q_m^b \cdot y_{ij}^{bm}, \quad \forall (i,j) \in L \quad (3.5)$$

$$y_{ij}^{bm} \in \{0, 1\}, \quad \forall (i,j) \in L, \forall b \in B \forall m \in M \quad (3.6)$$

$$x_{ij}^c \in \{0, 1\}, \quad \forall (i,j) \in L, \forall c \in C \quad (3.7)$$

A função objetivo (3.1) maximiza o *throughput* da rede, ou seja, o somatório das velocidades de transmissão das conexões escalonadas. As restrições (3.2) garantem que cada conexão é atribuída a no máximo um canal, e que duas conexões que compartilham o mesmo dispositivo não são escalonadas ao mesmo tempo. Além disso, as

restrições (3.3) correlacionam os valores das variáveis  $x$  e  $y$ , e garantem que  $(i, j) \in L$  transmite com no máximo uma velocidade de transmissão. As restrições (3.4) computam o valor da interferência  $I_{ij}$ , para todo  $(i, j) \in L$ . Neste caso, a interferência que ocorre no receptor de uma conexão  $(i, j) \in L$  é causada pelo transmissor de cada conexão  $(u, v) \in L \setminus \{(i, j)\}$  que foi escalonada e atribuída a um canal  $\bar{c} \in O^c$  que se sobrepõem a  $c$ , tal que  $x_{uv}^{\bar{c}} = 1$ . Já as restrições (3.5) impõem que  $(i, j)$  pode apenas transmitir a uma velocidade de transmissão  $\bar{r}_m^b$  se  $SINR_{ij} \geq q_m^b$ , onde  $SINR_{ij} = \frac{P}{I_{ij} + N}$ . A integralidade das variáveis  $y$  e  $x$  é garantida por (3.6) e (3.7), respectivamente.

## 3.2 Formulações lineares

As restrições (3.4) e (3.5) são claramente não-lineares. A fim de obter uma formulação linear para VRBSP, é realizada a linearização das restrições (3.5) que é baseada na abordagem apresentada em [Powell & Baker, 2009]. Primeiramente, as restrições (3.5) são reescritas como (3.8).

$$\frac{P}{(d_{ij})^\alpha} \geq \sum_{b \in B} \sum_{m \in M} q_m^b \cdot y_{ij}^{bm} \cdot (I_{ij} + N), \quad \forall (i, j) \in L \quad (3.8)$$

Em seguida, as restrições (3.8) são reescritas como (3.9).

$$\frac{\frac{P}{(d_{ij})^\alpha}}{\sum_{b \in B} \sum_{m \in M} q_m^b \cdot y_{ij}^{bm}} - N \geq I_{ij}, \quad \forall (i, j) \in L \quad (3.9)$$

Dado que cada conexão  $(i, j) \in L$  transmite com no máximo uma velocidade de transmissão, as restrições (3.9) podem ser escritas como (3.10), que é linear.

$$\sum_{b \in B} \sum_{m \in M} \left( \frac{P}{(d_{ij})^\alpha} - N \right) \cdot y_{ij}^{bm} \geq I_{ij}, \quad \forall (i, j) \in L \quad (3.10)$$

Duas abordagens são propostas para linearizar as restrições (3.4), que computam

o valor de  $I_{ij}$  para todo  $(i, j) \in L$ . Ambas fazem uso das variáveis  $z_{uv}^c \in \mathbb{R}$ , para todo  $(u, v) \in L$  e  $c \in C$ , tal que  $z_{uv}^c = 1$  se  $(u, v) \in L$  é atribuída a um canal  $\bar{c} \in O^c$  que se sobrepõe com  $c$ , e  $z_{uv}^c = 0$  caso contrário. O valor correto dessas variáveis é garantido pelas restrições (3.11).

$$z_{uv}^c = \sum_{\bar{c} \in O^c} x_{uv}^{\bar{c}}, \quad \forall (u, v) \in L, \forall c \in C \quad (3.11)$$

A primeira abordagem para linearizar (3.4) é baseada nos trabalhos de [Manne, 1960; Am et al., 2017; Arenales et al., 2015]. Além de (3.11), essa abordagem faz o uso das variáveis  $I_{ij}^c \in \mathbb{R}$ , que armazenam qual seria o valor de  $I_{ij}$  se  $x_{ij}^c = 1$ , para todo  $(i, j) \in L$  e  $c \in C$ , ou seja:

$$I_{ij}^c = \sum_{(u,v) \in L \setminus \{(i,j)\}} \frac{P}{(d_{uj})^\alpha} \cdot z_{uv}^c, \quad \forall (i, j) \in L, \forall c \in C$$

Portanto, o valor de  $I_{ij}$  pode ser linearmente computado por (3.12) e (3.13), onde a constante  $\mathcal{M}_{ij} = \sum_{(u,v) \in L \setminus \{(i,j)\}} \frac{P}{(d_{uj})^\alpha}$  é um limite superior ao valor de  $I_{ij}^c$ . Portanto, o VRBSP é linearmente formulado por (3.1)-(3.3),(3.6)-(3.13).

$$I_{ij} \geq I_{ij}^c - \mathcal{M}_{ij} \cdot (1 - x_{ij}^c), \quad \forall (i, j) \in L, \forall c \in C \quad (3.12)$$

$$I_{ij} \leq I_{ij}^c + \mathcal{M}_{ij} \cdot (1 - x_{ij}^c), \quad \forall (i, j) \in L, \forall c \in C \quad (3.13)$$

Como a formulação acima se baseia em uma abordagem que usa Big  $\mathcal{M}$ , esta pode resultar em uma relaxação linear com limites superiores fracos para o valor do *throughput* ótimo da rede [Bazaraa et al., 2011; Codato & Fischetti, 2006]. Assim sendo, uma segunda abordagem é proposta para linearizar a equação (3.4). Esta abordagem é baseada no trabalho de Bisschop [2006]. Além de (3.11), esta abordagem faz uso



das variáveis  $w_{ij}^{uv} \in [0, 1]$ , para toda conexão  $(i, j) \in L$  e  $(u, v) \in L \setminus \{(i, j)\}$ , tal que  $w_{ij}^{uv} = 1$  se as conexões  $(i, j)$  e  $(u, v)$  são atribuídas ao mesmo canal, e  $w_{ij}^{uv} = 0$  caso contrário. O valor correto dessas variáveis é garantido pelas restrições (3.14) e (3.15).

$$w_{ij}^{uv} \geq x_{ij}^c + z_{uv}^c - 1, \quad \forall (i, j) \in L, \forall (u, v) \in L \setminus \{(i, j)\}, \forall c \in C \quad (3.14)$$

$$0 \leq w_{ij}^{uv} \leq 1, \quad \forall (i, j) \in L, \forall (u, v) \in L \setminus \{(i, j)\}, \quad (3.15)$$

Portanto, o valor de  $I_{ij}$  pode ser linearmente computado por (3.16). Assim sendo, VRBSP é linearmente formulado por (3.1)-(3.3),(3.6)-(3.11),(3.14)-(3.16).

$$I_{ij} = \sum_{(u,v) \in L \setminus \{(i,j)\}} \frac{P}{(d_{uj})^\alpha} \cdot w_{ij}^{uv}, \quad \forall (i, j) \in L \quad (3.16)$$



# Capítulo 4

## Heurísticas para o VRBSP

Este capítulo apresenta duas heurísticas para o VRBSP. A primeira, descrita na seção 4.1, é baseada em algoritmos genéticos com chaves aleatórias tendenciosos (BRKGA, do inglês *Biased Random-Key Genetic Algorithm*) [Gonçalves & Resende, 2011]. Esta metaheurística já foi usada com sucesso para resolver vários problemas de otimização em redes [Buriol et al., 2005; Noronha et al., 2011; Reis et al., 2011; Brandão et al., 2016; Andrade et al., 2015]. Já a segunda heurística, descrita na seção 4.2, é composta por uma heurística construtiva seguida de uma busca em vizinhança variável (VNS, do inglês *Variable Neighborhood Search*) [Mladenović & Hansen, 1997]. A metaheurística VNS também já foi usada com sucesso para resolver vários problemas de otimização em redes [Castaño et al., 2014; Abdelkhalek et al., 2015; Lanza-Gutierrez & Gomez-Pulido, 2016; Plotnikov et al., 2016; Erzin et al., 2017].

### 4.1 Algoritmo genético com chaves aleatórias tendencioso

Um BRKGA evolui uma população de cromossomos que consistem em vetores de números reais no intervalo  $[0, 1)$ . Estes números são chamados de chaves e são gerados

aleatoriamente na população inicial. O *fitness* de um cromossomo é dado pelo custo da solução encontrada por um algoritmo decodificador, que recebe o vetor de chaves como entrada e retorna uma solução viável para o problema.

Este trabalho usa o operador de *recombinação uniforme parametrizado* (do inglês, *parameterized uniform crossover*) proposto em [Spears & deJong, 1991] que combina dois indivíduos pais para gerar um filho. Neste esquema, o indivíduo descendente herda cada uma de suas chaves com probabilidade 0,7 do pai mais apto e com probabilidade 0,3 do pai menos apto. Este algoritmo genético não usa o operador de mutação padrão, onde parte dos cromossomos são substituídos com uma pequena probabilidade. Ao invés disso, o conceito de *mutantes* é usado. A cada geração, um número fixo de cromossomos são inseridos na população. Eles são gerados da mesma forma que na população inicial. O processo de inserir indivíduos mutantes na população tem o papel de ajudar o algoritmo a escapar de ótimos locais.

A cada nova geração, a população é dividida em três conjuntos: *TOP*, *MID* e *BOT*. Conseqüentemente, o tamanho da população é  $|TOP| + |MID| + |REST|$ . As melhores soluções são mantidas em *TOP* enquanto as demais são colocadas em  $REST = MID \cup BOT$ . Como ilustrado na figura 4.1, os cromossomos em *TOP* são copiados, sem mudança, para a população da próxima geração. Os novos mutantes são colocados no conjunto *BOT*. Os demais indivíduos da nova população são obtidos pelo cruzamento entre um pai escolhido aleatoriamente do conjunto *TOP* e outro do conjunto *REST*. Desse modo, soluções elite têm uma maior probabilidade de passarem as suas chaves para a próxima geração. Isto distingue um algoritmo genético com chaves aleatórias tendencioso do algoritmo genético com chaves aleatórias de Bean [1994]. No último, os dois pais são selecionados aleatoriamente dentre todos os indivíduos da população. Dessa forma,  $|MID|$  soluções descendentes são geradas pelo operador de recombinação. O tamanho dos conjuntos *TOP*, *MID*, e *BOT* são parâmetros que devem ser ajustados.

Um pseudocódigo do BRKGA é descrito no algoritmo 1. Na linha 2, a população

inicial é gerada utilizando—se vetores com chaves aleatórias. O laço das linhas 3 – 16 é executado enquanto a condição de parada não é satisfeita. Essa condição de parada define que o BRKGA executa por até 600 segundos. No laço das linhas 4 e 5 o *fitness* de cada indivíduo é computado pelo algoritmo decodificador. Na linha 6, ordena—se a população e a divide em três grupos, de acordo com o *throughput* da solução, chamados de *TOP*, *MID* e *BOT*. No laço das linhas 7 e 8 copia—se o grupo *TOP* para próxima geração. No laço das linhas 9 – 12, realiza—se o cruzamento, em que os cromossomos de *MID* são gerados para próxima geração. No laço da linha 13 – 15 os elementos do grupo *BOT* são gerados, de forma aleatória, para a próxima geração. Por fim, na linha 16, a solução com o maior *throughput* é retornada.

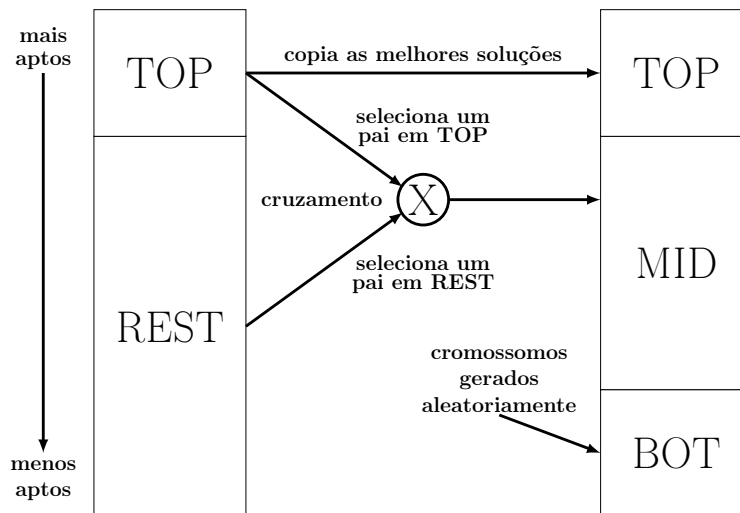


Figura 4.1: Ilustração do processo de transição entre consecutivas gerações do algoritmo genético com chaves aleatórias tendencioso.

O BRKGA para o VRBSP tem duas chaves associadas a cada conexão  $(i, j) \in L$ , chamadas de  $k_{ij}^1$  e  $k_{ij}^2$ . O decodificador analisa as conexões em ordem crescente do valor de  $k_{ij}^1$ , e tenta alocá-las em canais de largura de banda  $b_{ij}$ , onde  $b_{ij}$  é igual a 20, 40, 80, ou 160 MHz se  $k_{ij}^2$  está no intervalo  $[0, 0.25)$ ,  $[0.25, 0.5)$ ,  $[0.5, 0.75)$ , ou  $[0.75, 1)$ , respectivamente. Ele utiliza uma estratégia de *First Fit* [Johnson, 1973] e está dividido em três etapas. Na primeira, as conexões são inseridas no primeiro canal vazio de largura de banda  $b_{ij}$  que não se sobrepõe a nenhum outro canal. Caso tal

canal não exista, a conexão não é escalonada nesta etapa. Desta forma, não existe interferência no escalonamento resultante da primeira etapa. Na segunda etapa, as conexões que não foram escalonadas na etapa anterior são inseridas no primeiro canal vazio, dentre aqueles com a maior largura de banda, que não se sobrepõe a nenhum outro canal. Caso não exista tal canal, todas as bandas de frequência disponíveis estão sendo usadas por alguma conexão. Neste caso, a segunda fase é interrompida, pois não é mais possível escalonar conexões sem que ocorram interferências. Na terceira etapa, as conexões que não foram escalonadas nas etapas anteriores são inseridas no canal cujo *throughput* resultante da inserção é máximo. Entretanto, se a inserção da conexão resulta na diminuição do *throughput* da rede ou em uma solução inviável devido ao incremento na interferência, a conexão não é escalonada. Assim, toda a vez que a inserção da conexão em cada um dos canais é testada, tanto o *throughput* quanto a interferência da solução resultante são computados com a finalidade de se verificar a qualidade e viabilidade de tal solução.

No algoritmo 2 é descrito o pseudocódigo do algoritmo decodificador. Na linha 2 é inicializada a solução  $S$ . Nesta linha também é inicializado o conjunto de canais  $C'$  e o de conexões  $L'$  a serem usados. Na linha 3 os pares de chaves do cromossomo são ordenados em ordem crescente conforme o valor da chave  $k_{ij}^1$ . No laço das linhas 4 – 11, realiza-se a inserção de cada uma das conexões em canais vazios que possuem largura de banda igual a  $b_{ij}$ , conforme o valor da chave  $k_{ij}^2$ , tal que os mesmos não se sobreponham a nenhum dos canais utilizados pela solução corrente. Na linha 12 é atualizado o conjunto de conexões que ainda não foram escalonadas. No laço das linhas 13 – 17, cada uma das conexões, dentre aquelas que não foram escalonadas, são inseridas em canais vazios cuja a largura de banda seja a maior dentre aqueles que ainda não foram utilizados e que não se sobrepõem a nenhum outro canal que já está sendo usado. O conjunto de conexões que ainda não foram escalonadas é novamente atualizado na linha 18. No laço das linhas 19 – 22, cada uma das conexões restantes é inserida no canal, dentre os canais já utilizados, cujo o *throughput* resultante da

inserção é máximo. Por fim, na linha 23, o *throughput* da solução obtida é retornada.

---

**Algorithm 1:** BRKGA
 

---

```

1 begin
2   Gerar uma população inicial de vetores de chaves aleatórias;
3   while Condição de parada não satisfeita do
4     foreach elemento da população P do
5       Decodificador( $L$ , elemento);
6       Dividir a população nos conjuntos  $TOP$ ,  $MID$  e  $BOT$ ;
7       foreach elemento da população TOP do
8         Copiar o elemento para próxima geração;
9       for  $i$  de 1 até  $|MID|$  do
10        Seleccione aleatoriamente um cromossomo  $c^t$  de  $TOP$  e outro  $c^r$  de
11          $MID \cup BOT$ ;
12         $c^f \leftarrow Cruzamento(c^t, c^r)$ ;
13        Copiar  $c^f$  para a próxima geração;
14       for  $i$  de 1 até  $|BOT|$  do
15        Gerar um novo cromossomo  $c^a$  aleatoriamente;
16        Copiar  $c^a$  para a próxima geração;
17   return melhor solução de  $TOP$ ;
18 end

```

---

---

**Algorithm 2:** Decodificador

---

```

1 DECODIFICADOR( $L, cromossomo$ ) begin
2    $S \leftarrow \emptyset; C' \leftarrow \emptyset; L' \leftarrow \emptyset;$ 
3   Ordene os pares de chaves do cromossomo em ordem crescente conforme
   o valor de  $k_{ij}^1$ ;
4   foreach  $(i, j) \in L$  do
5     Defina o valor de  $b_{ij}$  de acordo com o valor de  $k_{ij}^2$ ;
6     foreach  $c \in C$  do
7       if  $B^c = b_{ij}$  &&  $c$  não se sobrepõe a nenhum canal  $\bar{c} \in C'$  then
8          $S \leftarrow Inserir(S, c, (i, j));$ 
9          $C' \leftarrow C' \cup c;$ 
10         $L' \leftarrow L' \cup (i, j);$ 
11         $C \leftarrow C \setminus \{c\};$ 
12   $L \leftarrow L - L';$ 
13  foreach  $(i, j) \in L$  do
14    Selecione um canal  $c \in C$  com a maior largura de banda e que não se
    sobrepõe a nenhum canal  $\bar{c} \in C'$ ;
15     $S \leftarrow Inserir(S, c, (i, j));$ 
16     $C' \leftarrow C' \cup c;$ 
17     $L' \leftarrow L' \cup \{(i, j)\};$ 
18   $L \leftarrow L - L';$ 
19  foreach  $(i, j) \in L$  do
20    foreach  $\bar{c} \in C'$  do
21       $S' \leftarrow Inserir(S, \bar{c}, (i, j));$ 
22       $S \leftarrow Melhor(S, S');$ 
23  return  $f(S);$ 
24 end

```

---



## 4.2 Heurística de busca em vizinhanças variáveis

A heurística apresentada nesta seção é composta por uma heurística construtiva, descrita na seção 4.2.1, e uma heurística baseada em VNS, descrita na seção 4.2.3, que usa como solução inicial a solução da heurística construtiva. Além disso, a heurística VNS faz uso de um algoritmo, descrito na seção 4.2.2, que tem a finalidade de obter soluções cujo o particionamento do espectro de transmissão em canais seja localmente ótimo. Este algoritmo consiste em explorar apenas o espaço de soluções correspondente à atribuição de conexões a canais de 20 MHz de forma que, a partir das soluções obtidas por meio de tais atribuições, seja computado o particionamento do espectro em canais. Neste caso, o particionamento localmente ótimo do espectro em canais é obtido por meio da junção de canais adjacentes de 20 MHz em canais com largura de banda maior e seleção da melhor solução resultante.

### 4.2.1 Heurística construtiva

A heurística construtiva (CH, do inglês *Constructive Heuristic*) recebe como entrada uma instância de VRBSP caracterizada pela tupla  $\langle V, L, B, C, O^c, C^b, M, q_s^b, r_s^b, d_{ij}, \alpha, N \rangle$ , definida no capítulo 1. Inicialmente, todo o espectro de transmissão é particionado em um conjunto  $C' \subseteq C$  de canais com as maiores larguras de banda possíveis e que não se sobrepõem. Por exemplo, na figura 1.2 temos que  $C' = \{25, 42, 43, 44, 45\}$ . Este conjunto inicial de canais é construído dessa forma com o intuito de fazer com que as conexões transmitam nas velocidades mais altas que puderem já que quanto maior é a largura de banda de um canal, maior é a velocidade com que elas transmitem. Em seguida, a cada iteração, uma conexão  $(i, j) \in L$  é escolhida aleatoriamente e tenta-se inseri-la na solução corrente da seguinte forma. Para cada canal  $c \in C'$ , duas soluções parciais são

avaliadas. A primeira consiste na inserção de  $(i, j)$  em  $c$ . Além disso, caso  $\frac{B^c}{2} \in B$ , a segunda solução consiste em, inserir  $(i, j)$  em  $c$ , e em seguida particionar  $c$  em dois canais de largura de banda  $\frac{B^c}{2}$ . Este processo é repetido para todos os canais em  $C'$  e a conexão é inserida no canal que resulta na solução parcial com maior *throughput*, caso esta seja melhor que a solução corrente. Caso contrário,  $(i, j)$  não é escalonada.

---

**Algorithm 3:** Heurística Construtiva
 

---

**Data:**  $\langle V, L, B, C, O^c, C^b, M, q_s^b, \bar{r}_s^b, d_{ij}, \alpha, N \rangle$

**Result:**  $S$ , melhor solução encontrada

```

1 begin
2    $C' \leftarrow$  conjunto inicial de canais;
3    $S \leftarrow \emptyset$ ;
4   foreach  $(i, j) \in L$  selecionado aleatoriamente do
5     foreach  $c \in C'$  do
6        $S' \leftarrow$  Inserir( $S, c, (i, j)$ );
7       if  $\frac{B^c}{2} \in B$  then  $S'' \leftarrow$  Dividir( $S', c$ );
8        $S \leftarrow$  Melhor( $S, S', S''$ );
9      $C' \leftarrow \{c \in C : \exists \langle (u, v), c \rangle \in S\}$ ;
10  return  $S$ ;
11 end

```

---

O algoritmo 3 apresenta o pseudocódigo da heurística CH. O algoritmo recebe como entrada a tupla  $\langle V, L, B, C, O^c, C^b, M, q_s^b, \bar{r}_s^b, d_{ij}, \alpha, N \rangle$ . O conjunto  $C'$  é inicializado na linha 2. O laço das linhas 4 a 9 é repetido para cada conexão  $(i, j) \in L$ . Neste caso, as conexões são avaliadas em uma ordem aleatória. O laço das linhas 5 a 8 é repetido para cada canal  $c \in C'$ . Na linha 6, computa-se a solução resultante da inserção da conexão  $(i, j)$  no canal  $c$ , enquanto na linha 7, computa-se a solução resultante da inserção da conexão  $(i, j)$  no canal  $c$ , seguida do particionamento do canal  $c$  em dois canais  $c'$  e  $c''$  cujas larguras de banda são iguais a  $\frac{B^c}{2}$ . Este particionamento é feito de

forma gulosa. Inicialmente,  $c'$  e  $c''$  contém uma única conexão cada um, que são o par de conexões com a maior interferência mútua dentre as conexões em  $c$ . Em seguida, as demais conexões em  $c$  são escolhidas aleatoriamente e inseridas no subconjunto que resulta no maior incremento do *throughput* da solução. A melhor solução conhecida é atualizada na linha 8, enquanto o conjunto  $C'$  é atualizando na linha 9. Por fim, a melhor solução obtida pela heurística CH é retornada na linha 10.

### 4.2.2 Modelagem do espaço de busca do VNS

Dada uma solução  $S \in \Delta$  para o VRBSP, onde todas as conexões estão alocadas a canais de 20 MHz, é possível obter novas soluções que são formadas por canais com largura de banda maior que por sua vez são formados pela junção dos canais adjacentes de  $S$ . Dados dois canais de comunicação  $c'$  e  $c''$  que são adjacentes e têm largura de banda igual a  $\frac{B^c}{2}$ , considera-se que a junção de tais canais pode ser realizada para formar um canal  $c$  com largura de banda  $B^c$  cuja a faixa de frequência ocupada por ele é a mesma de  $c'$  e  $c''$ . De acordo com a figura 1.2, a junção dos canais 3 e 4, cuja a largura de banda é igual a 20 MHz, pode ser feita para formar o canal 27 que tem largura de banda igual a 40 MHz e está na mesma faixa de frequência que 3 e 4, por exemplo. Seja  $S$  a solução que contém os canais  $c'$  e  $c''$ . Já a solução que é resultante da junção de  $c'$  e  $c''$  para formar o canal  $c$  é dada por  $S'$ . Neste caso, se o *throughput* da solução  $S'$  é maior que o *throughput* da solução  $S$ , a melhor solução é  $S'$  que contém o particionamento localmente ótimo com o canal  $c$ . Caso contrário, a melhor solução é  $S$  que contém o particionamento localmente ótimo com os canais  $c'$  e  $c''$ . Assim sendo, o particionamento localmente ótimo é obtido por meio da junção de canais adjacentes e seleção da solução com o maior *throughput*. Apesar de que existe uma quantidade exponencialmente grande de soluções que podem ser obtidas por meio da junção de canais adjacentes para formar um único canal com largura de banda maior. A escolha da melhor solução pode ser computada em tempo polinomial,

conforme o descrito anteriormente. Uma vez que o particionamento localmente ótimo dos canais pode ser computado em tempo polinomial. A exploração do espaço de busca pode ser feita apenas pela atribuição das conexões a canais de 20 MHz para que em seguida tais soluções sejam usadas como base para se computar o seu particionamento localmente ótimo e obter novas soluções. Na próxima seção é descrito o pseudocódigo do algoritmo que computa o particionamento localmente ótimo por meio do algoritmo 6. Este algoritmo é usado pela heurística VNS.

Desta forma, a heurística VNS descrita só precisa avaliar a atribuição de conexões em canais de 20 MHz, pois todas soluções que podem ser obtidas através da junção das conexões nos canais adjacentes nesta solução podem ser computadas pelo algoritmo descrito anteriormente.

### 4.2.3 VNS

A metaheurística Busca em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*) foi proposta em [Mladenović & Hansen, 1997]. Ela é um método iterativo, que sucessivamente aplica uma perturbação seguida de uma busca local no ótimo local corrente  $S$ , com o objetivo de gerar um novo ótimo local  $S''$ . A solução corrente é atualizada no caso de  $S''$  ser melhor que  $S$ . A perturbação consiste em selecionar um vizinho  $S'$  aleatoriamente na vizinhança de  $S$ . Já a busca local consiste em um algoritmo que retorna um ótimo local  $S''$  que está próximo de  $S'$ . O que caracteriza VNS e a diferencia de outras metaheurísticas é que a perturbação aplicada a  $S$  varia sistematicamente de uma iteração para a outra. Sejam  $N_1, \dots, N_{k_{max}}$  as funções de vizinhança utilizadas na perturbação, assume-se que  $N_k$  é utilizada antes da vizinhança  $N_{k+1}$ . No caso da perturbação em  $N_k$ , seguida da busca local, resultar em um ótimo local melhor do que  $S$ , utiliza-se  $N_1$  na perturbação da iteração seguinte. Caso contrário, utiliza-se  $N_{k+1}$ .

A heurística VNS para o VRBSP realiza uma busca local na união de duas

vizinhanças: *Add-drop* e *Reinsert*. O conjunto dos vizinhos *Add-drop* de uma solução  $S \in \Delta$  é computado da seguinte forma. Para cada conexão não escalonada  $(i, j) \in L \setminus L(S)$ , computa-se a solução resultante da inserção de  $(i, j)$  em cada um dos canais utilizados em  $S$ . A melhor dentre estas soluções é o vizinho de  $S$  correspondente a conexão  $(i, j)$ . O conjunto dos vizinhos *Reinsert* de  $S$  é computado de forma semelhante. Para cada conexão escalonada  $(i, j) \in L(S)$ , remove-se  $(i, j)$  de  $S$  e computa-se o vizinho *Add-drop* correspondente a  $(i, j)$  na solução resultante.

A heurística VNS para o VRBSP realiza a perturbação na união de duas vizinhanças  $\beta$ -*Add-drop* e  $\beta$ -*Reinsert*, que por sua vez são generalizações de *Add-drop* e *Reinsert*, respectivamente. Um vizinho  $\beta$ -*Add-drop* de uma solução  $S \in \Delta$  é selecionado aleatoriamente da seguinte forma. Inicialmente,  $\beta$  conexões não escalonadas são escolhidas aleatoriamente em  $L \setminus L(S)$ . Em seguida, estas conexões são individualmente inseridas em  $S$  da mesma forma que na vizinhança *Add-drop*, ou seja, no canal que maximiza o *throughput* da rede. Analogamente, um vizinho  $\beta$ -*Reinsert* de  $S$  é selecionado aleatoriamente da seguinte forma. Inicialmente,  $\beta$  conexões escalonadas são escolhidas aleatoriamente em  $L(S)$ . Em seguida, estas conexões são removidas de  $S$  e individualmente inseridas na solução resultante da mesma forma que na vizinhança *Reinsert*, ou seja, no canal que maximiza o *throughput* da rede.

O algoritmo 4 apresenta o pseudocódigo da heurística VNS. Ela recebe como entrada a tupla  $\langle V, L, B, C, O^c, C^b, M, q_s^b, \bar{r}_s^b, d_{ij}, \alpha, N \rangle$ , e a solução  $S^{CH}$  retornada pela heurística construtiva CH. A solução corrente  $S$  é inicializada na linha 2, com o ótimo local obtido a partir de  $S^{CH}$ . O laço das linhas 3-12 é executado enquanto o critério de parada não é satisfeito. Neste caso, o critério de parada do VNS foi fixado em 600 segundos. O tamanho da perturbação  $k$  é inicializado na linha 4. A cada iteração do laço das linhas 5-12 um novo ótimo local é gerado e avaliado. Essa solução é obtida aplicando-se uma perturbação de tamanho  $k$  a  $S$  na linha 6 e em seguida uma busca local na solução resultante na linha 7. Uma perturbação de tamanho  $k$  consiste em (i) selecionar aleatoriamente uma das vizinhanças  $\beta$ -*Add-drop* e

$\beta$ -*Reinsert* e em seguida (ii) selecionar aleatoriamente um vizinho de  $S$  na vizinhança escolhida com  $\beta = |L| \cdot k\%$ . O tamanho máximo da perturbação foi fixado em  $k_{max} = 50$ . Se a solução resultante  $S''$  é melhor do que a solução  $S$  (linha 8), a solução corrente é atualizada na linha 9, e o tamanho da perturbação é reinicializado na linha 10. Caso contrário, o valor de  $k$  é incrementado a cada 5 iterações do laço das linhas 5-12. Por fim, a melhor solução encontrada é retornada na linha 13.

---

**Algorithm 4:** Variable Neighborhood Search

---

**Data:**  $S^{CH}$  e  $\langle V, L, B, C, O^c, C^b, M, q_s^b, \bar{r}_s^b, d_{ij}, \alpha, N \rangle$

**Result:**  $S$ , Melhor solução encontrada

```

1 begin
2    $S \leftarrow \text{BuscaLocal}(S^{CH});$ 
3   while Condição de parada não satisfeita do
4      $k \leftarrow 1;$ 
5     while  $k \leq k_{max}$  do
6        $S' \leftarrow \text{Perturbação}(S, k);$ 
7        $S'' \leftarrow \text{BuscaLocal}(S');$ 
8       if  $f(S'') > f(S)$  then
9          $S \leftarrow S'';$ 
10         $k \leftarrow 1;$ 
11      else
12        Atualizar( $k$ );
13  return  $S;$ 
14 end

```

---

O pseudocódigo da busca local é descrito pelo algoritmo 5. Ela recebe como entrada uma solução  $S^*$ . Na linha 3, a solução  $S^*$  é salva em uma nova estrutura de dados em que todas as conexões estão armazenadas em canais com largura de banda iguais a 20MHz. Nas linhas 4 e 5 são inicializados  $S$  e  $S''$ , respectivamente. O laço

das linhas 6 – 19 é executado enquanto o critério de parada não é satisfeito. No laço das linhas 7 – 13 computa-se a solução resultante da reinserção de cada conexão  $(i, j) \in L(s)$  em todo o canal  $c \in C^{20}$ . Na linha 11, que faz parte desse laço, computa-se a junção de canais adjacentes em canais com larguras de banda maiores. Na linha 12,  $S''$  recebe a melhor solução até então obtida por meio da busca local. A solução corrente é atualizada na linha 14. Se a nova solução encontrada for melhor, a busca local continua. Caso contrário, a busca local é encerrada e a melhor solução encontrada por ela é retornada na linha 20.

No algoritmo 6 é descrito o pseudocódigo do algoritmo que computa o particionamento localmente ótimo dos canais. Ele recebe como entrada a solução e o canal cujo o respectivo espectro será analisado. Na linha 2 é inicializado o valor do melhor *throughput* encontrado. Essa função que analisa o particionamento localmente ótimo dos canais funciona de forma recursiva. Assim, a cada iteração são avaliadas duas soluções parciais. A primeira solução consiste em definir como parte da solução o canal  $c \in C$  que é recebido como entrada. Caso  $\frac{B^c}{2} \in B$ , a segunda solução consiste em definir como parte da solução os canais adjacentes  $c'$  e  $c''$  cuja a largura de banda é igual a  $\frac{B^c}{2}$  e o espectro é o mesmo de  $c \in C$ . Neste caso, é verificado na linha 3 se o canal  $c \in C$  pode ser particionado. Logo, se  $\frac{B^c}{2} \in B$ , é avaliado na linha 6 se o *throughput* da primeira solução é maior do que *throughput* da segunda. A solução que resulta no maior *throughput* é escolhida e os dados da solução corrente são atualizados nas linhas 7 e 8. Por fim, a solução com o particionamento localmente ótimo dos canais é retornada na linha 9.

---

**Algorithm 5:** Busca Local
 

---

```

1  BUSCALOCAL( $S^*$ ) begin
2  |    $melhora \leftarrow true$ ;
3  |    $S^* \leftarrow Converte(S^*)$ ;
4  |    $S \leftarrow S^*$ ;
5  |    $S'' \leftarrow S^*$ ;
6  |   while  $melhora$  do
7  |     |   foreach  $(i, j) \in L(s)$  do
8  |     |     |    $S' \leftarrow S \setminus \{(i, j), c\}$ ;
9  |     |     |   foreach  $\bar{c} \in C^{20} \setminus \{c\}$  do
10 |     |     |     |    $S' \leftarrow Insere(S', \bar{c}, (i, j))$ ;
11 |     |     |     |    $S' \leftarrow Solve(S', \bar{c})$ ;
12 |     |     |     |    $S'' \leftarrow Melhor(S', S'')$ ;
13 |     |     |     |    $S' \leftarrow S \setminus \{(i, j), c\}$ ;
14 |     |   if  $f(S'') > f(S^*)$  then
15 |     |     |    $S^* \leftarrow S''$ ;
16 |     |     |    $S \leftarrow S''$ ;
17 |     |     |    $melhora \leftarrow true$ ;
18 |     |   else
19 |     |     |    $melhora \leftarrow false$ ;
20 |   return  $S$ ;
21 end

```

---



---

**Algorithm 6:** Computa Largura de Banda

---

```
1 SOLVE(S, canal) begin
2   best ← canal.throughput;
3   if canal possui um filho then
4     bestC1 ← SOLVE(canal.C1);
5     bestC2 ← SOLVE(canal.C2);
6     if bestC1 + bestC2 > best then
7       best ← bestC1 + bestC2;
8       canal.inSolution ← false;
9   return best;
10 end
```

---



## Capítulo 5

# Experimentos Computacionais

Este capítulo descreve os experimentos computacionais que foram realizados com o objetivo de analisar e comparar os algoritmos exatos e heurísticos propostos nesta tese. Os algoritmos exatos são baseados nas formulações F1 e F2, dadas pelas equações (3.1)-(3.3),(3.6)-(3.13) e (3.1)-(3.3),(3.6)-(3.11),(3.14)-(3.16), respectivamente. Estas formulações foram resolvidas pelo algoritmo de *branch and cut* do *solver* IBM/ILOG CPLEX, versão 12.6, com os parâmetros padrão. Os algoritmos de *branch and cut* do CPLEX baseados nas formulações F1 e F2 são denominados B&C-F1 e B&C-F2, respectivamente. Já as heurísticas CH, BRKGA e VNS foram implementadas na linguagem C++ e compiladas com o GNU g++ versão 4.8.2. Todas as heurísticas utilizaram o gerador de número aleatórios *Mersenne Twister* proposto por [Matsumoto & Nishimura, 1998]. Os parâmetros das heurísticas BRKGA e VNS foram definidos de forma empírica por meio da realização de experimentos computacionais. Em relação aos parâmetros do BRKGA, o tamanho da população é igual a 100, a cardinalidade dos conjuntos *TOP*, *REST* e *BOT* são iguais a 25, 75 e 5, respectivamente. No BRKGA proposto um indivíduo descendente herda cada uma de suas chaves com probabilidade de 0,7 do pai mais apto e com probabilidade 0,3 do pai menos apto. Com relação ao VNS, o tamanho máximo da perturbação foi fixado em  $k_{max} = 50$ . Além disso, as heurísticas BRKGA

e VNS param após 10 minutos de execução. O código do algoritmo Datarate PPTAS foi disponibilizado pelos autores de [Goussevskaia et al., 2016]. Ele foi implementado na linguagem Java e executado por meio do JRE (*Java Runtime Environment*) 1.8.0. Os experimentos computacionais foram realizados em uma máquina com processador Intel Core i7 3.33 GHz e 24 GB de memória RAM, executando o sistema operacional Linux Ubuntu 14.04 LTS.

Neste trabalho, diferentes conjuntos de instâncias foram gerados. Neste caso, todos os conjuntos foram gerados por meio do mesmo gerador de instâncias que foi utilizado em [Goussevskaia et al., 2016], sendo que o seu código também foi disponibilizado pelos autores. O primeiro conjunto, denominado *D10000*, foi usado nos experimentos de [Goussevskaia et al., 2016]. Ele consiste em instâncias que possuem planos com dimensões iguais a  $10.000 \times 10.000$  metros. A distância entre os dispositivos na rede é Euclidiana. Os receptores são posicionados sobre o plano de forma aleatória, enquanto que os transmissores são posicionados dentro de um raio com comprimento igual a  $l_{max} = 6\sqrt{2}$  metros cujo centro é a coordenada do respectivo receptor. Assim como em [Goussevskaia et al., 2016], os valores de  $|L|$  variam por 8, 16, 32, 64, 128, 256, 512, 1024, e 2048 conexões. Desta forma, para cada um destes valores, foram geradas 30 instâncias. Portanto, o conjunto *D10000* é composto por 270 instâncias.

Além de *D10000*, foi usado um segundo conjunto de instâncias nos experimentos, denominado *D250*. Este último foi gerado da mesma forma que *D10000* e é também composto por 270 instâncias. No entanto, as instâncias neste conjunto são geradas em um plano de  $250 \times 250$  metros. Sendo assim, a densidade de conexões nestas instâncias é maior e conseqüentemente a interferência entre elas também se torna mais alta. Isto também faz com que o receptor de cada conexão tenha mais dificuldade para decodificar os dados que são enviados para ele. Portanto, espera-se que seja mais difícil obter soluções ótimas para as instâncias neste conjunto do que para aquelas em *D10000*.

## 5.1 Experimentos com algoritmos exatos

Esta seção apresenta os resultados dos experimentos realizados com os algoritmos exatos. Ela está dividida em duas partes. Na seção 5.1.1, B&C-F1, B&C-F2, e Datarate PPTAS [Goussevskaia et al., 2016] são avaliados em instâncias de VRSP, enquanto na seção 5.1.2, B&C-F1 e B&C-F2 são avaliados em instâncias de VRBSP.

### 5.1.1 Resultados para o VRSP

Duas variantes de VRSP são tratadas em [Goussevskaia et al., 2016], uma correspondente ao padrão IEEE 802.11b e outra correspondente ao padrão IEEE 802.11n. Em ambas, há apenas um canal disponível, ou seja,  $|C| = 1$ . No entanto, na primeira variante do VRSP o conjunto de larguras de banda é igual a  $B = \{20\}$ , e os valores de  $q_m^{20}$  e  $\bar{r}_m^{20}$ , para todo  $m \in M = \{0, 1, 2, 3\}$ , são dados pela tabela 5.1. Na segunda variante, o conjunto de larguras de banda é igual a  $B = \{40\}$ , e os valores de  $q_m^{40}$  e  $\bar{r}_m^{40}$ , para todo  $m \in M = \{0, \dots, 7\}$ , são dados pela tabela 5.2.

Tabela 5.1: SINR(dB) e Velocidade (Mbps) para o padrão IEEE 802.11b.

MCS Index	20 MHz	
	SINR $q_m^{20}$	Velocidade $\bar{r}_{ij}^{20}$
0	4,0	1,0
1	6,0	2,0
2	8,0	5,5
3	10,0	11,0

Tabela 5.2: SINR(dB) e Velocidade (Mbps) para o padrão IEEE 802.11n.

MCS Index	40 MHz	
	SINR $q_m^{40}$	Velocidade $\bar{r}_{ij}^{40}$
0	14,0	30,0
1	17,0	60,0
2	19,0	90,0
3	22,0	120,0
4	26,0	180,0
5	30,0	240,0
6	31,0	270,0
7	32,0	300,0

Os resultados de B&C-F1, B&C-F2, e Datarate PPTAS para a variante do VRSP correspondente ao padrão IEEE 802.11b são apresentados nas tabelas 5.3 e 5.4, sendo que a tabela 5.3 mostra os resultados para as instâncias do conjunto  $D10000$  e a tabela 5.4 mostra os resultados para as instâncias do conjunto  $D250$ . Os mesmos resulta-

dos para a variante do VRSP correspondente ao padrão IEEE 802.11n são apresentados nas tabelas 5.5 e 5.6, sendo que a tabela 5.5 mostra os resultados para as instâncias do conjunto  $D10000$  e a tabela 5.6 apresenta os resultados para o conjunto  $D250$ . Estas quatro tabelas possuem a mesma estrutura. A coluna 1 descreve os valores de  $|L|$ , enquanto que a coluna 2 mostra o valor equivalente ao *throughput* máximo que a rede pode atingir quando todas as conexões transmitem na velocidade máxima, que por sua vez é dado por  $\overline{ub}$ . As colunas 3 a 6 apresentam respectivamente (i) o número de instâncias para as quais o Datarate PPTAS foi capaz de encontrar a solução ótima, (ii) o valor médio do limite inferior  $lb$  referente ao *throughput* das soluções encontradas para 30 instâncias, (iii) o gap de otimalidade relativa ( $\frac{ub^* - lb}{lb}$ ) entre  $lb$  e o melhor limite superior conhecido  $ub^*$  para cada instância, e (iv) o tempo médio de execução deste algoritmo. Neste caso, o melhor limite superior conhecido  $ub^*$  é dado pelo valor mínimo obtido entre  $\overline{ub}$  e os limites superiores encontrados por B&C-F1 e B&C-F2. O número de instâncias resolvidas na otimalidade por B&C-F1 em até 3600 segundos é dado na coluna 7. Já a média do limite superior  $ub$ , do limite inferior  $lb$ , e do gap de otimalidade relativa ( $\frac{ub - lb}{lb}$ ), sobre as 30 instâncias, encontradas por B&C-F1 são apresentadas, respectivamente, nas colunas 8 a 10. Além disso, a coluna 11 mostra o tempo de execução médio deste algoritmo. Os mesmos resultados são apresentados para B&C-F2 nas últimas cinco colunas, respectivamente. Quando um destes algoritmos não foi capaz de encontrar uma solução factível não vazia em até 3600 segundos, a coluna correspondente é preenchida com um '-'. Além disso, quando não há memória suficiente para executar qualquer um destes algoritmos, a coluna correspondente é preenchida com o sinal  $\times$ .

Tabela 5.3: Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto  $D10000$  referentes a variante do VRSP que corresponde ao padrão IEEE 802.11b.

L	Datarate PPTAS					B&C-F1					B&C-F2				
	$\bar{ub}$	#opt	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)
8	88,00	0	83,63	5,64	0,02	30	88,00	88,00	0,00	0,01	30	88,00	88,00	0,00	0,01
16	176,00	0	151,73	16,47	0,02	30	176,00	176,00	0,00	0,01	30	176,00	176,00	0,00	0,02
32	352,00	0	280,03	26,04	0,02	30	352,00	352,00	0,00	0,01	30	352,00	352,00	0,00	0,08
64	704,00	0	541,63	30,04	366,00	30	703,27	703,27	0,00	0,03	30	703,27	703,27	0,00	6,90
128	1408,00	0	-	-	3600,00	30	1407,82	1407,82	0,00	0,11	×	×	×	×	×
256	2816,00	0	-	-	3600,00	30	2813,08	2813,08	0,00	0,42	×	×	×	×	×
512	5632,00	0	-	-	3600,00	30	5617,95	5617,95	0,00	3,17	×	×	×	×	×
1024	11264,00	0	-	-	3600,00	30	11203,22	11203,22	0,00	25,33	×	×	×	×	×
2048	22528,00	0	-	-	3600,00	30	22286,73	22286,73	0,00	179,22	×	×	×	×	×

Tabela 5.4: Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto  $D250$  referentes a variante do VRSP que corresponde ao padrão IEEE 802.11b.

L	Datarate PPTAS					B&C-F1					B&C-F2				
	$\bar{ub}$	#opt	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)
8	88,00	0	-	-	3600,00	30	81,40	81,40	0,00	0,04	30	81,40	81,40	0,00	0,04
16	176,00	0	-	-	3600,00	30	157,40	157,40	0,00	0,05	30	157,40	157,40	0,00	0,09
32	352,00	0	-	-	3600,00	30	277,95	277,95	0,00	0,26	30	277,95	277,95	0,00	0,68
64	704,00	0	-	-	3600,00	30	438,88	438,88	0,00	7,89	30	438,88	438,88	0,00	52,82
128	1408,00	0	-	-	3600,00	13	618,72	610,37	1,38	2779,00	×	×	×	×	×
256	2816,00	0	-	-	3600,00	0	911,04	748,10	21,81	3600,00	×	×	×	×	×
512	5632,00	0	-	-	3600,00	0	1278,06	838,22	52,52	3600,00	×	×	×	×	×
1024	11264,00	0	-	-	3600,00	0	1767,34	636,92	192,72	3600,00	×	×	×	×	×
2048	22528,00	0	-	-	3600,00	0	16686,11	-	-	3600,00	×	×	×	×	×

Tabela 5.5: Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto  $D10000$  referentes a variante do VRSP que corresponde ao padrão IEEE 802.11n.

$ L $	Datarate PPTAS					B&C-F1					B&C-F2				
	$\bar{ub}$	#opt	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)
8	2400,00	0	2323,00	3,55	9,43	30	2400,00	2400,00	0,00	0,01	30	2400,00	2400,00	0,00	0,01
16	4800,00	0	4238,00	13,84	94,52	30	4798,00	4798,00	0,00	0,01	30	4798,00	4798,00	0,00	0,02
32	9600,00	0	7751,00	24,11	196,38	30	9582,00	9582,00	0,00	0,02	30	9582,00	9582,00	0,00	0,09
64	19200,00	0	-	-	3600,00	30	19106,00	19106,00	0,00	0,05	30	19106,00	19106,00	0,00	3,81
128	38400,00	0	-	-	3600,00	30	37949,00	37949,00	0,00	0,19	×	×	×	×	×
256	76800,00	0	-	-	3600,00	30	75014,00	75014,00	0,00	1,17	×	×	×	×	×
512	153600,00	0	-	-	3600,00	30	146739,00	146739,00	0,00	20,28	×	×	×	×	×
1024	307200,00	0	-	-	3600,00	30	278334,00	278334,00	0,00	991,90	×	×	×	×	×
2048	614400,00	0	-	-	3600,00	0	503874,69	203126,00	436,01	3600,00	×	×	×	×	×

Tabela 5.6: Resultados do Datarate PPTAS, B&C-F1, e B&C-F2 para o conjunto  $D250$  referentes a variante do VRSP que corresponde ao padrão IEEE 802.11n.

$ L $	Datarate PPTAS					B&C-F1					B&C-F2				
	$\bar{ub}$	#opt	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)	#opt	ub	lb	gap(%)	t(s)
8	2400,00	0	-	-	3600,00	30	1226,00	1226,00	0,00	0,25	30	1226,00	1226,00	0,00	0,33
16	4800,00	0	-	-	3600,00	30	1619,00	1619,00	0,00	2,48	30	1619,00	1619,00	0,00	5,65
32	9600,00	0	-	-	3600,00	30	1934,00	1934,00	0,00	146,66	29	1937,70	1934,00	0,19	1011,83
64	19200,00	0	-	-	3600,00	0	2675,63	2173,00	23,26	3600,00	0	3659,57	2066,00	77,41	3600,00
128	38400,00	0	-	-	3600,00	0	8330,16	2309,00	261,52	3600,00	×	×	×	×	×
256	76800,00	0	-	-	3600,00	0	19008,65	2481,00	666,65	3600,00	×	×	×	×	×
512	153600,00	0	-	-	3600,00	0	45099,86	2512,00	1700,30	3600,00	×	×	×	×	×
1024	307200,00	0	-	-	3600,00	0	116626,56	1573,00	7939,79	3600,00	×	×	×	×	×
2048	614400,00	0	-	-	3600,00	0	592757,24	-	-	3600,00	×	×	×	×	×



A respeito da variante do VRSP que corresponde ao padrão IEEE 802.11b, pode ser visto a partir das tabelas 5.3 e 5.4 que o Datarate PPTAS foi capaz de encontrar soluções viáveis, em até 3600 segundos, para instâncias com até 64 conexões do conjunto  $D10000$  (assim como ocorreu em Goussevskaia et al. [2016]), e para nenhuma instância do conjunto  $D250$ . Enquanto isso, B&C-F1 encontrou soluções ótimas para todas as 270 instâncias do conjunto  $D10000$ , e para todas as instâncias do conjunto  $D250$  com até 64 conexões, bem como 13 das instâncias com 128 conexões. Além disso, este foi capaz de encontrar soluções viáveis em que a média dos gaps de otimalidade foi de 21,81%, 52,52%, e 192,72% para as instâncias do conjunto  $D250$  com 256, 512, e 1024 conexões, respectivamente. No entanto, B&C-F2 encontrou soluções ótimas apenas para as instâncias dos conjuntos  $D10000$  e  $D250$  com até 64 conexões. B&C-F2 não foi capaz de executar instâncias com 128 ou mais conexões, devido ao grande número de variáveis  $w_{ij}^{uv}$ , para todo  $(i, j) \in L$  e  $(u, v) \in L \setminus \{(i, j)\}$ , que estão presentes apenas na formulação F2. Como o número de variáveis  $x_{ij}^c$  cresce linearmente enquanto que o número de variáveis  $w_{ij}^{uv}$  cresce quadraticamente em relação a  $|L|$ , é esperado que B&C-F2 atinja o limite máximo de memória em instâncias menores do que B&C-F1. Além disso, B&C-F1 foi mais rápido na média do que B&C-F2 quando ambos encontraram soluções ótimas. Isto se deve também ao menor número de variáveis de F1, em comparação a F2, que faz com que F1 seja mais rápida de resolver em cada nó da árvore de *branch and cut*.

A respeito da variante do VRSP que corresponde ao padrão IEEE 802.11n, pode ser visto a partir das tabelas 5.5 e 5.6 que o Datarate PPTAS foi capaz de encontrar soluções viáveis, em até 3600 segundos, apenas nas instâncias do conjunto  $D10000$  com até 32 conexões (assim como ocorreu em Goussevskaia et al. [2016]), e para nenhuma instância do conjunto  $D250$ . Enquanto isso, B&C-F1 encontrou soluções ótimas para todas as instâncias do conjunto  $D10000$  com até 1024 conexões, e para todas as instâncias do conjunto  $D250$  com até 32 conexões. Além disso, este foi capaz de encontrar soluções para todas as instâncias do conjunto  $D10000$  com 2048 conexões e para todas

as instâncias do conjunto  $D250$  com até 1024 conexões, mas nestes casos com gaps de otimalidade altos. Estes resultados mostram que estas instâncias do conjunto  $D250$  são mais difíceis já que, mesmo atingindo o tempo limite de execução, o gap de otimalidade das soluções obtidas por B&C-F1 foram altos. B&C-F2 foi capaz de executar apenas nas instâncias de ambos os conjuntos com até 64 conexões. No entanto, este foi capaz de encontrar soluções ótimas, em até 3600 segundos, para todas as 120 instâncias do conjunto  $D10000$  com até 64 conexões e para 89 (das 90) instâncias do conjunto  $D250$  com até 32 conexões.

### 5.1.2 Resultados para o VRBSP

Nesta seção são descritos os experimentos que foram realizados para analisar a capacidade de B&C-F1 e B&C-F2 para obter soluções para o VRBSP. Neste caso, os valores SINR mínimo  $q_m^b$  e sua respectiva velocidade de transmissão  $\bar{r}_m^b$ , para todo  $b \in B = \{20, 40, 80, 160\}$  e  $m \in M = \{0, \dots, 9\}$ , são definidos de acordo com o padrão IEEE 802.11ac. Portanto, os valores de  $q_m^b$  e  $\bar{r}_m^b$  são dados pela tabela 1.1.

Os resultados de B&C-F1 e B&C-F2 são apresentados nas tabelas 5.7 e 5.8, sendo que a tabela 5.7 exhibe os resultados para as instâncias do conjunto  $D10000$  e a tabela 5.8 mostra os resultados para as instâncias do conjunto  $D250$ . Estas duas tabelas possuem a mesma estrutura. A coluna 1 apresenta os valores de  $|L|$ , enquanto que a coluna 2 mostra o valor equivalente ao *throughput* máximo que a rede pode atingir quando todas as conexões transmitem na velocidade máxima, que por sua vez é dado por  $\bar{ub}$ . O número de instâncias resolvidas na otimalidade por B&C-F1 em até 3600 segundos é apresentado na coluna 3. A média do limite superior  $ub$  e do limite inferior  $lb$ , bem como a média do gap otimalidade relativa ( $\frac{ub-lb}{lb}$ ), sobre 30 instâncias, encontradas por B&C-F1 são apresentadas nas colunas 4 a 6, respectivamente. Além disso, a coluna 7 mostra o tempo de execução médio do mesmo algoritmo. Os mesmos resultados são apresentados para B&C-F2 nas últimas cinco colunas, respectivamente. Quando

um destes algoritmos não foi capaz de encontrar uma solução viável não vazia em até 3600 segundos, a coluna correspondente é preenchida com '-'. Além disso, quando a memória física não é o suficiente para executar algum destes algoritmos, a coluna correspondente é preenchida com o sinal  $\times$ .

Pode se ver a partir das tabelas 5.7 e 5.8 que B&C-F1 encontrou soluções ótimas para todas as instâncias do conjunto  $D10000$  com até 256 conexões, e para todas as instâncias do conjunto  $D250$  com até 8 conexões. Além disso, este algoritmo foi capaz de encontrar soluções viáveis para todas as instâncias com 512 conexões do conjunto  $D10000$  e para todas as instâncias do conjunto  $D250$  com até 128 conexões, mas nestes casos com gaps otimalidade altos. Enquanto isso, B&C-F2 foi capaz de executar apenas em instâncias com até 64 conexões de ambos os conjuntos. B&C-F2 encontrou soluções ótimas, em até 3600 segundos, para todas as instâncias com até 64 conexões do conjunto  $D10000$  e todas as instâncias com 8 conexões do conjunto  $D250$ . Além disso, este foi capaz de encontrar soluções viáveis para todas as instâncias do conjunto  $D250$  com até 64 conexões, mas nestes casos com gaps de otimalidade altos.

Estas tabelas mostram que as instâncias do conjunto  $D250$  são mais difíceis que as do conjunto  $D10000$  de Goussevskaia et al. [2016] como o conjecturado acima. Pode-se observar na tabela 5.8 que nestas instâncias mais difíceis, os limites superiores de B&C-F2 foram melhores do que os de B&C-F1. Isto era esperado uma vez que a formulação F2 faz uso das variáveis  $w_{ij}^{uv}$ ,  $(i, j) \in L$  e  $(u, v) \in L \setminus \{(i, j)\}$ , ao invés da abordagem de Big  $\mathcal{M}$  usada por F1, o que geralmente resulta em melhores limites superiores [Bazaraa et al., 2011; Codato & Fischetti, 2006]. Infelizmente, apesar de ter melhores limites superiores do que B&C-F1, B&C-F2 não foi capaz de executar em instâncias com mais de 64 conexões, devido ao alto consumo de memória física, o que se deve precisamente à grande quantidade de variáveis  $w_{ij}^{uv}$ .



## 5.2 Experimentos com heurísticas

Esta seção apresenta e discute os experimentos computacionais realizados para avaliar o desempenho das heurísticas CH, BRKGA e VNS para solucionar o VRBSP.

Os resultados são apresentados nas tabelas 5.9 e 5.10, sendo que a tabela 5.9 exhibe os resultados para as instâncias do conjunto *D10000* e a tabela 5.10 mostra os resultados para as instâncias do conjunto *D250*. Estas duas tabelas possuem a mesma estrutura. A coluna 1 apresenta os valores de  $|L|$ . Já as colunas 2 e 3 mostram, respectivamente, a média do *throughput* ( $lb^{ga}$ ) e do *gap* de otimalidade relativa ( $\frac{opt-lb^{ga}}{opt}$ ) das soluções encontradas pelo BRKGA, sobre 30 instâncias, onde *opt* é o valor do melhor limite superior conhecido para cada instância. Neste caso, *opt* é dado pelo valor mínimo obtido entre os limites superiores referentes as soluções ótimas encontradas por B&C-F1 e B&C-F2. Quando a solução ótima não é conhecida, esta coluna é preenchida com um '-'. Os mesmos valores são exibidos para CH nas colunas 4 e 5, respectivamente. Além disso, a coluna 6 reporta a média do desvio relativo percentual ( $\frac{lb^{ch}-lb^{ga}}{lb^{ga}}$ ) do *throughput* ( $lb^{ch}$ ) das soluções encontradas por CH em relação ao *throughput* ( $lb^{ga}$ ) das soluções obtidas pelo BRKGA. Valores negativos indicam que as soluções de BRKGA foram melhores do que aquelas de CH. Em seguida, a coluna 7 apresenta o tempo médio de execução de CH. Por fim, as colunas 8, 9 e 10 mostram, respectivamente, a média do *throughput* das soluções de VNS, do respectivo *gap* de otimalidade relativa, e do desvio relativo percentual ( $\frac{lb^{vns}-lb^{ga}}{lb^{ga}}$ ) do *throughput*  $lb^{vns}$  das soluções encontradas por VNS em relação ao *throughput* ( $lb^{ga}$ ) das soluções obtidas pelo BRKGA. Neste caso, valores positivos do %dev indicam que as soluções de VNS foram melhores que aquelas de BRKGA.

Pode-se observar nas tabelas 5.9 e 5.10 que, nas instâncias onde a solução ótima é conhecida, os gaps de otimalidade de BRKGA, CH e VNS são no máximo 5,83%, 4,95% e 0,05% nas instâncias do conjunto *D250* com 8 conexões. Já nas instâncias com 8 conexões do conjunto *D10000*, enquanto o CH, e conseqüentemente o VNS,

Tabela 5.9: Resultados do BRKGA, CH e VNS para o conjunto *D10000* referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac.

L	BRKGA		CH				VNS		
	lb	gap(%)	lb	gap(%)	%dev	t(s)	lb	gap(%)	%dev
8	4758,00	31,15	6240,00	0,00	31,15	0,01	6240,00	0,00	31,15
16	10998,00	13,48	12480,00	0,00	13,48	0,01	12480,00	0,00	13,48
32	23478,00	6,31	24960,00	0,00	6,31	0,01	24960,00	0,00	6,31
64	48438,00	3,04	49899,20	0,02	3,02	0,02	49909,60	0,00	3,04
128	98358,00	1,46	99769,80	0,02	1,44	0,04	99788,00	0,01	1,45
256	198198,00	0,61	199258,80	0,07	0,54	0,24	199390,10	0,01	0,60
512	396795,10	-	395642,00	-	-0,29	1,47	397442,50	-	0,16
1024	772125,25	-	764511,15	-	-0,99	10,95	775121,10	-	0,39
2048	1362898,58	-	1355032,61	-	-0,58	86,45	1390409,81	-	2,02

Tabela 5.10: Resultados do BRKGA, CH e VNS para o conjunto *D250* referentes a variante do VRBSP que corresponde ao padrão IEEE 802.11ac.

L	BRKGA		CH				VNS		
	lb	gap(%)	lb	gap(%)	%dev	t(s)	lb	gap(%)	%dev
8	4486,95	5,83	4529,20	4,95	0,94	0,01	4747,93	0,05	5,82
16	7448,05	-	7012,92	-	-5,84	0,01	7499,74	-	0,69
32	11366,44	-	10390,99	-	-8,58	0,01	11420,19	-	0,47
64	14866,52	-	13313,29	-	-10,45	0,01	15069,92	-	1,37
128	18622,63	-	15319,78	-	-17,74	0,02	18975,66	-	1,90
256	22742,51	-	17829,80	-	-21,60	0,04	22970,55	-	1,00
512	27356,45	-	21177,62	-	-22,59	0,15	28020,08	-	2,43
1024	32829,98	-	25101,01	-	-23,54	0,43	34097,58	-	3,86
2048	35775,13	-	29903,26	-	-16,41	1,04	38906,40	-	8,75

encontraram a solução ótima, o BRKGA obteve um gap de 31,15%. Isto se deve ao fato de que, na solução ótima para estas instâncias, todas as conexões estão alocadas a canais de 160 MHz, e neste caso em particular, o BRKGA não consegue convergir para esta solução em até 600 segundos. Para as instâncias do conjunto *D10000* com até 256 conexões, a própria heurística construtiva já foi capaz de encontrar soluções melhores que as do BRKGA, e ainda assim com um tempo computacional de no máximo 86,45 segundos nas maiores instâncias do conjunto *D10000* com 2048 conexões. Isto não ocorreu nas outras instâncias dos conjuntos *D10000* e *D250*, onde as soluções do BRKGA foram até 23,54% melhores que as do CH, nas instâncias com 1024 conexões do conjunto *D250*. Entretanto, em todas as instâncias testadas, as soluções encontradas pelo VNS foram melhores do que as do BRKGA. Nas maiores instâncias (com 2048 conexões) dos conjuntos *D10000* e *D250*, as soluções do VNS foram em média 2,02%

e 8,75% melhores que as do BRKGA, respectivamente.

De acordo com os valores referentes ao gap de otimalidade e do desvio percentual das soluções obtidas pela heurística VNS, percebe-se que ela apresentou o melhor desempenho em relação as demais heurísticas. Dentre os fatores que contribuíram para o bom desempenho do VNS destacam-se o uso da heurística CH para gerar a solução inicial utilizada por ela, bem como os métodos de perturbação e busca local para obtenção de novos ótimos locais. O uso da heurística CH pelo VNS, para gerar a solução inicial, permite que ele parta de uma solução de boa qualidade, o que contribuiu para que o VNS encontrasse soluções ainda melhores. O método de perturbação auxiliou o VNS na geração de novos vizinhos da solução corrente e também a escapar de ótimos locais, permitindo a exploração de diferentes regiões do espaço de busca. Além disso, o método de busca local do VNS permite a obtenção de soluções de boa qualidade cujo o particionamento dos canais é localmente ótimo em tempo polinomial. Isto ocorre porque tal método é capaz de obter diversas soluções por meio da exploração do espaço de soluções correspondente à atribuição de conexões a canais de 20 MHz em conjunto com o algoritmo 6, que computa o particionamento localmente ótimo dos canais, conforme o descrito na seção 4.2.3.

As figuras apresentadas a seguir, que vão de 5.1 a 5.14, mostram os gráficos com os resultados obtidos por meio dos experimentos realizados com as heurísticas CH, BRKGA e VNS, em que foram utilizadas as instâncias do conjunto *D250*. Esse conjunto de instâncias foi escolhido para a geração dos gráficos por ter se mostrado mais desafiador. Com isso, foi possível a geração de gráficos que permitissem uma melhor visualização das diferenças de desempenho entre as heurísticas, tanto em relação a qualidade das soluções obtidas por elas quanto pela forma como as mesmas escolhem os canais e larguras de banda para as conexões transmitirem.

A figura 5.1 mostra o desempenho das heurísticas CH, BRKGA e VNS em relação ao *throughput* das soluções obtidas para solucionar o VRBSP. De acordo com os resultados apresentados na figura 5.1, percebe-se que à medida que a quantidade de

conexões aumenta, o crescimento da qualidade das soluções obtidas pelas heurísticas BRKGA e VNS é superior ao da heurística CH, principalmente para redes com mais de 64 conexões. Já para redes com mais de 1024 conexões a qualidade das soluções obtidas pelo VNS é significativamente superior ao das soluções obtidas pelo BRKGA.

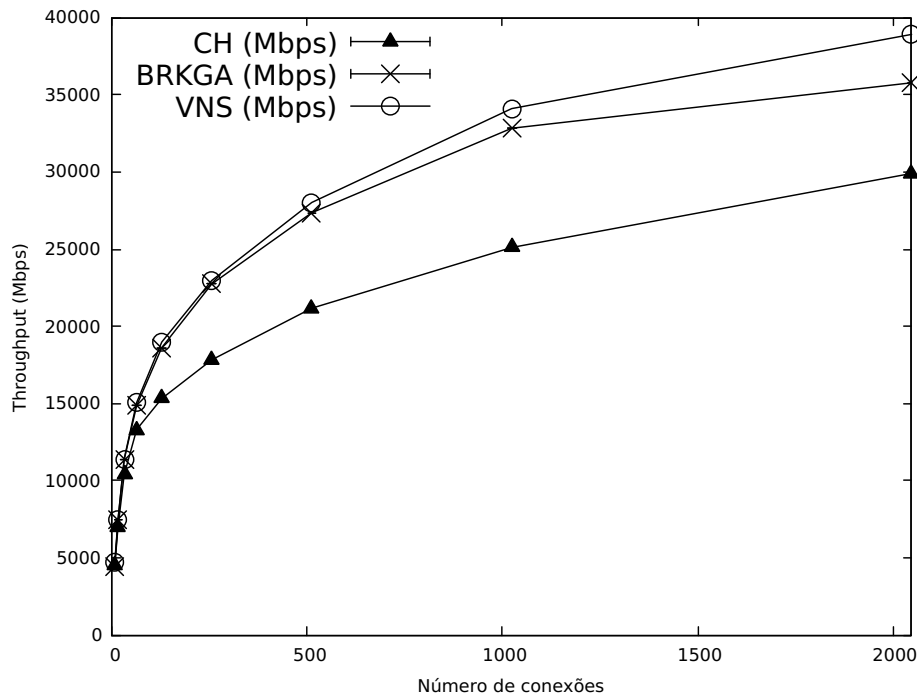


Figura 5.1: Desempenho das heurísticas em relação ao *throughput* das soluções obtidas utilizando as instâncias do conjunto *D250*.

As figuras 5.2 e 5.3 mostram a distribuição média do *throughput* referente as soluções encontradas para o VRBSP pelas heurísticas CH, BRKGA e VNS como uma função de distribuição acumulada. Percebe-se que, por meio da figura 5.2, o *throughput* de todas as soluções obtidas por BRKGA e VNS para as instâncias com 1024 conexões foram superiores aos das soluções obtidas por CH. Neste caso, o *throughput* das soluções encontradas por BRKGA e VNS foram acima de 32000 Mbps e 34000 Mbps, respectivamente. Por meio da figura 5.3 pode ser visto que o *throughput* de todas as soluções obtidas por BRKGA e VNS para as instâncias com 2048 conexões foram superiores aos das soluções obtidas por CH. O *throughput* das soluções encontradas por BRKGA e VNS foram acima de 35000 Mbps e 39000 Mbps, respectivamente. Além



disso, a distribuição média do *throughput* das soluções obtidas pelo BRKGA e o VNS mostraram uma pequena variação, ao contrário do *throughput* das soluções obtidas por CH que apresentaram uma variação maior. Por fim, percebeu-se também que o VNS apresentou um desempenho melhor do que as demais heurísticas em relação a qualidade e a consistência das soluções encontradas.

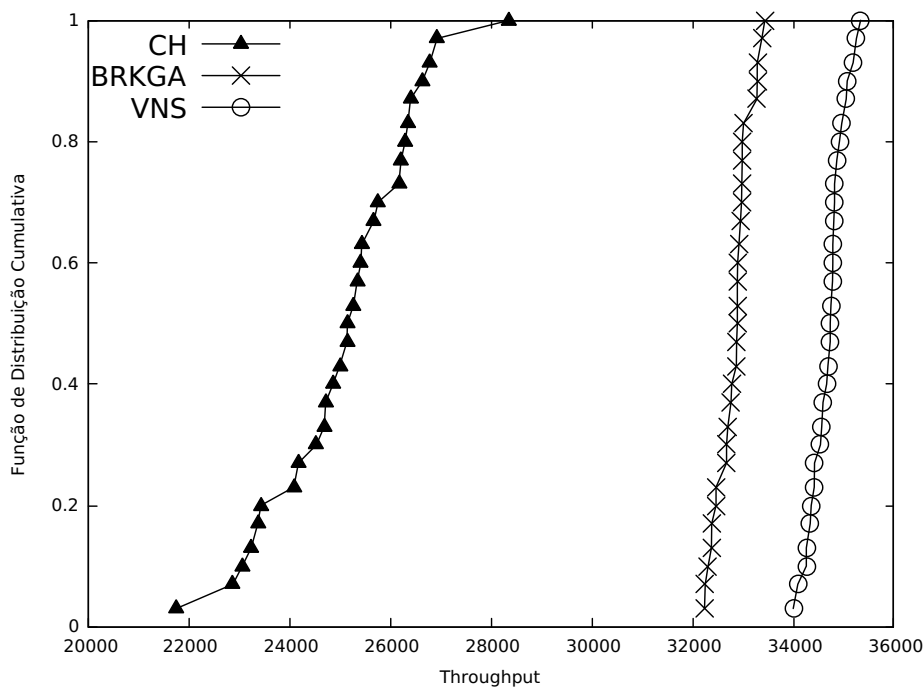


Figura 5.2: Distribuição média do *throughput* das soluções obtidas pelas heurísticas como uma função de distribuição acumulada para as instâncias do conjunto *D250* com 1024 conexões.

As figuras 5.4 e 5.5 apresentam a distribuição do *throughput* referente as soluções encontradas pelas heurísticas CH, BRKGA e VNS por meio de gráficos de caixa (do inglês *box plot*). Percebe-se que, por meio da figura 5.4, a variabilidade das soluções obtidas por BRKGA e VNS para as instâncias com 1024 conexões foi bem menor do que a das soluções obtidas por CH. E de acordo com os dados apresentados pela figura 5.5, a variabilidade das soluções obtidas pelas heurísticas BRKGA e VNS também foi inferior ao das soluções obtidas por CH, porém a variabilidade entre as soluções de CH foi maior em comparação ao da variação apresentada por CH na figura 5.4. Além disso, a distribuição das soluções obtidas pelo VNS mostram que o mesmo apresentou um

desempenho melhor do que as demais heurísticas em relação ao *throughput*.

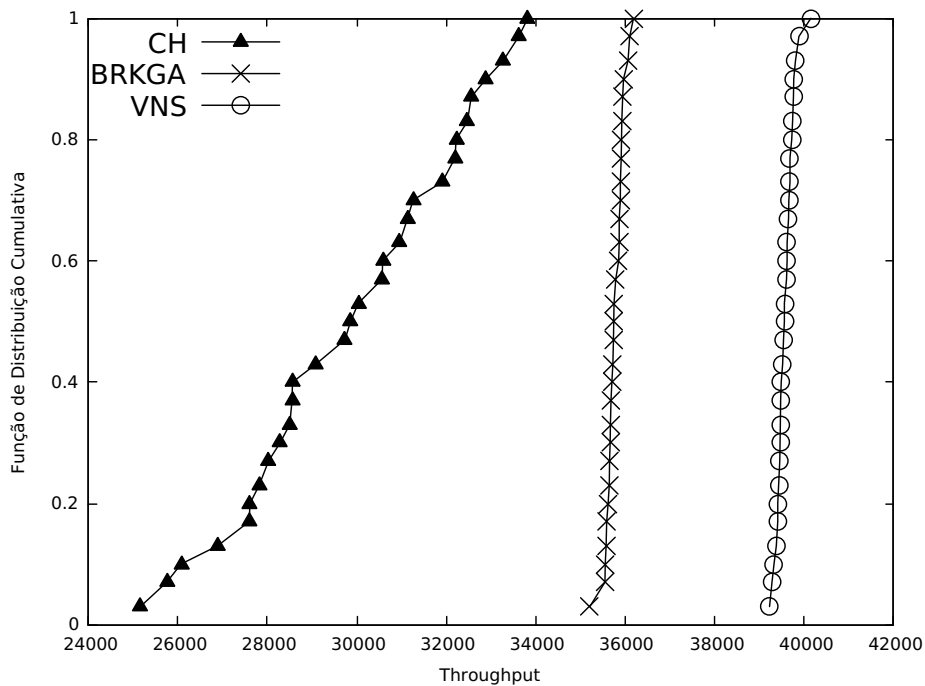


Figura 5.3: Distribuição média do *throughput* das soluções obtidas pelas heurísticas como uma função de distribuição acumulada para as instâncias do conjunto *D250* com 2048 conexões.

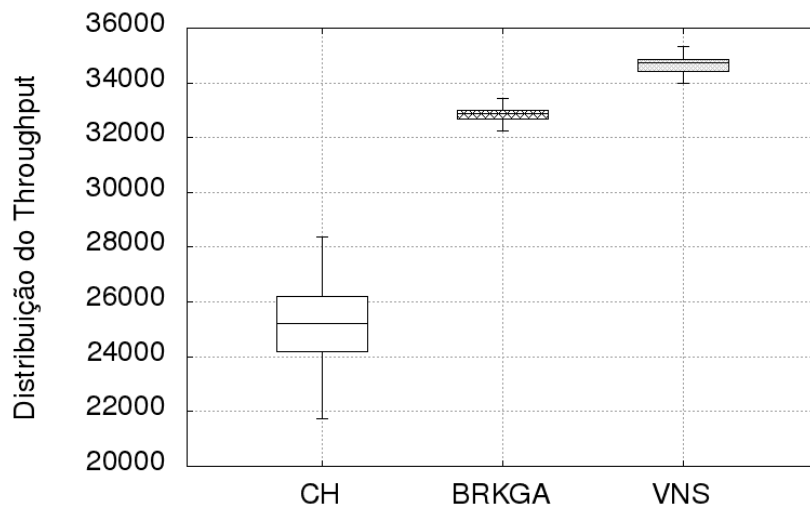


Figura 5.4: Box plot referente a distribuição do *throughput* das soluções obtidas pelas heurísticas para as instâncias do conjunto *D250* com 1024 conexões.

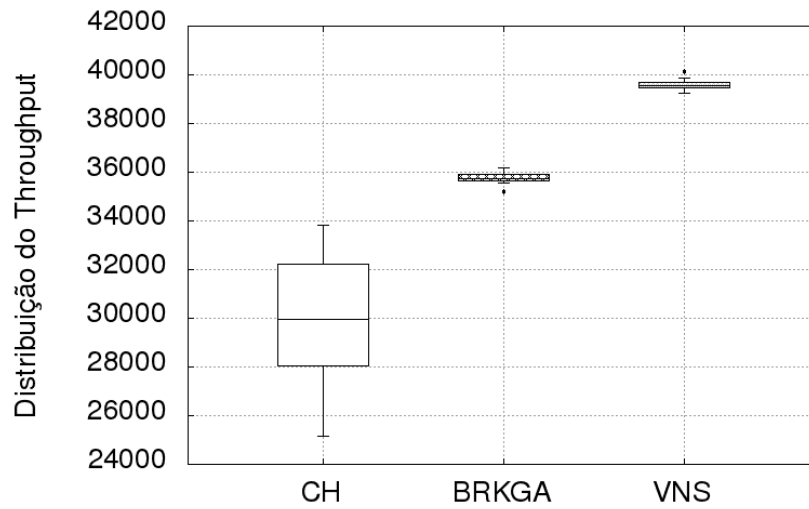


Figura 5.5: Box plot referente a distribuição do *throughput* das soluções obtidas pelas heurísticas para as instâncias do conjunto *D250* com 2048 conexões.

As figuras 5.6 e 5.7 ilustram como as melhores soluções encontradas pelo BRKGA e o VNS evoluem ao longo de 600 segundos de execução, utilizando as instâncias com 1024 e 2048 conexões. Com isso, os resultados apresentados em ambas as figuras mostram que o VNS sistematicamente encontra soluções melhores e de forma mais rápida do que o BRKGA ao longo do tempo.

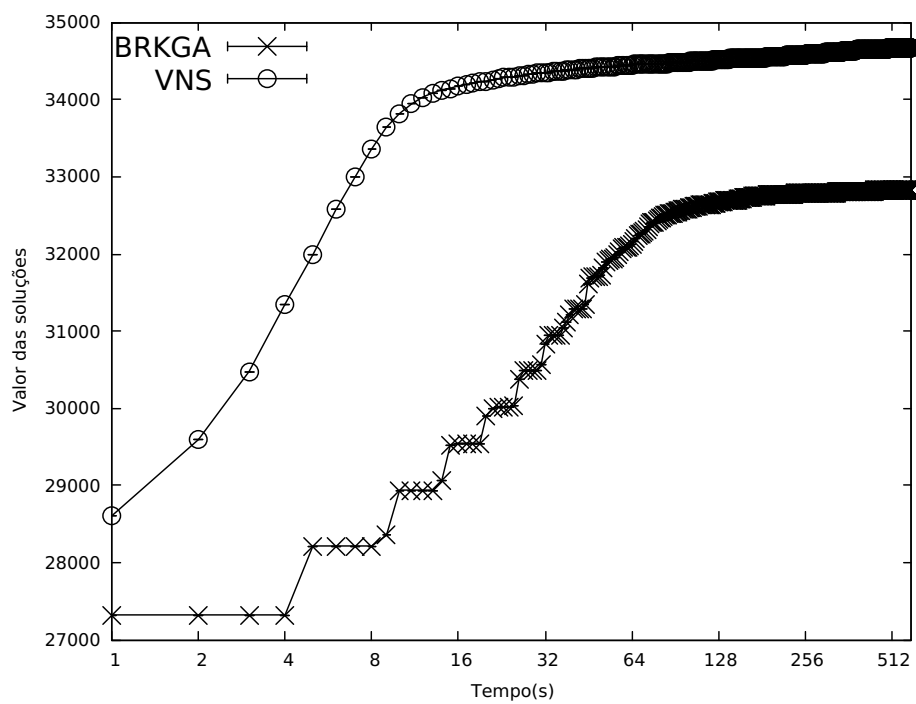


Figura 5.6: Evolução das melhores soluções encontradas pelo BRKGA e o VNS ao longo de 600 segundos de execução para as instâncias do conjunto *D250* com 1024 conexões.

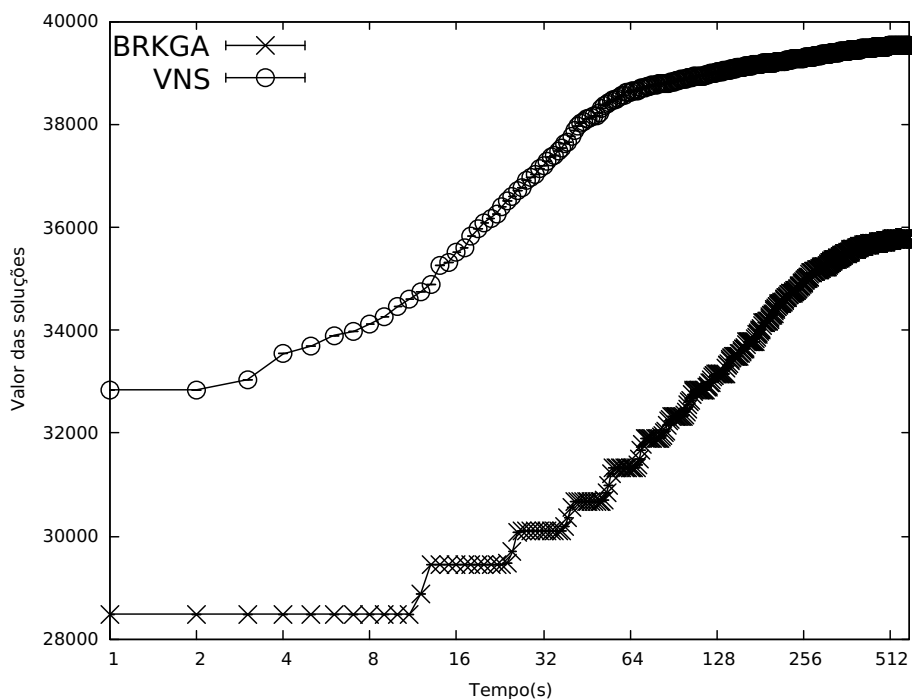


Figura 5.7: Evolução das melhores soluções encontradas pelo BRKGA e o VNS ao longo de 600 segundos de execução para as instâncias do conjunto *D250* com 2048 conexões.

Na figura 5.8 são ilustradas a distribuição média da quantidade de canais utilizados (figura 5.8a) e das conexões ativas (figura 5.8b), de acordo com a largura de banda dos canais, das soluções obtidas pelas heurísticas propostas para as instâncias com 32 conexões utilizando o padrão IEEE 802.11ac. Os mesmos resultados são apresentados pelas figuras 5.9, 5.10, 5.11, 5.12, 5.13 e 5.14 para as instâncias com 64, 128, 256, 512, 1024 e 2048 conexões, respectivamente.

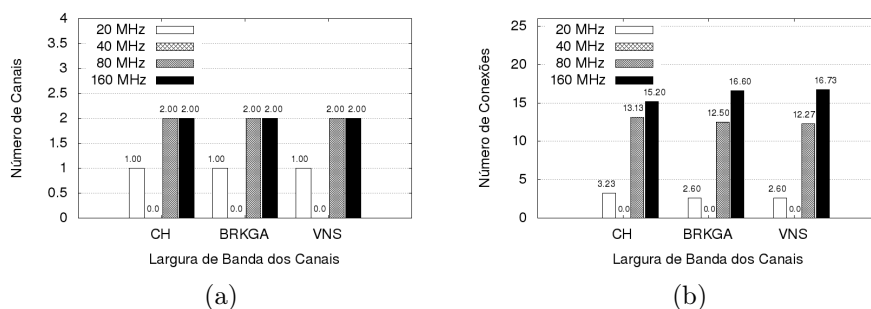


Figura 5.8: Distribuição média do número de canais usados (figura 5.8a) e das conexões ativas (figura 5.8b), de acordo com a largura de banda dos canais, das soluções com 32 conexões obtidas pelas heurísticas propostas.

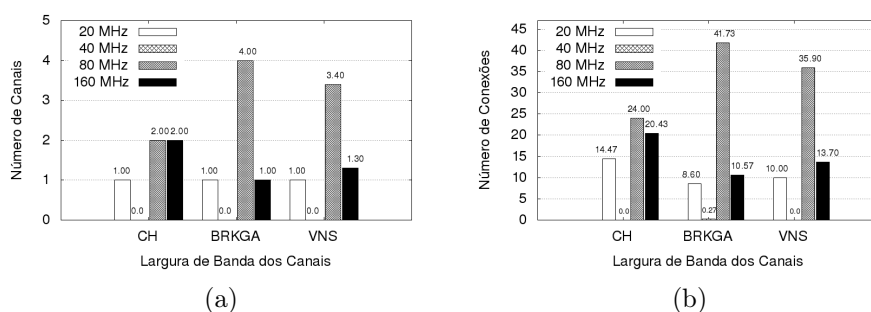


Figura 5.9: Distribuição média do número de canais usados (figura 5.9a) e das conexões ativas (figura 5.9b), de acordo com a largura de banda dos canais, das soluções com 64 conexões obtidas pelas heurísticas propostas.

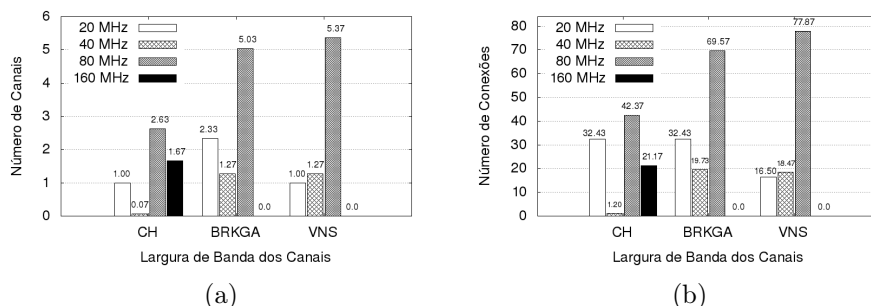


Figura 5.10: Distribuição média do número de canais usados (figura 5.10a) e das conexões ativas (figura 5.10b), de acordo com a largura de banda dos canais, das soluções com 128 conexões obtidas pelas heurísticas propostas.

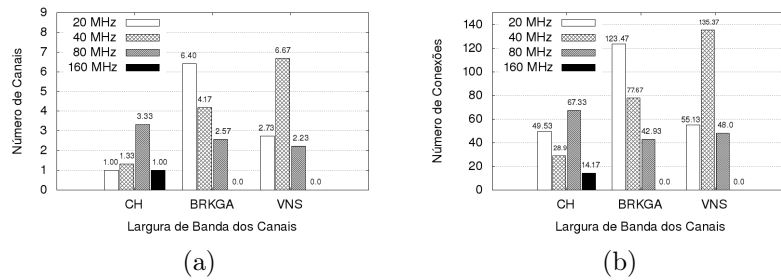


Figura 5.11: Distribuição média do número de canais usados (figura 5.11a) e das conexões ativas (figura 5.11b), de acordo com a largura de banda dos canais, das soluções com 256 conexões obtidas pelas heurísticas propostas.

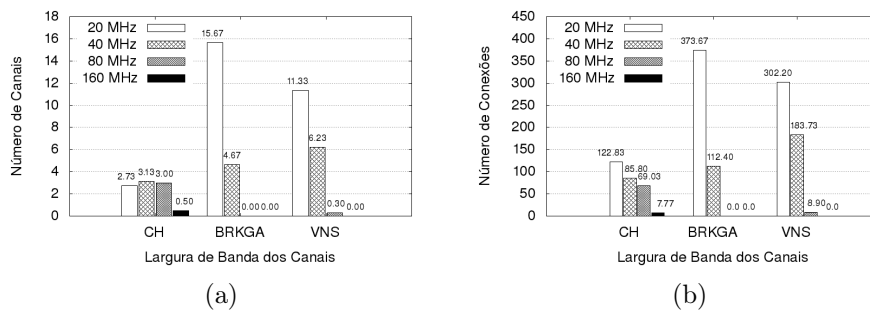


Figura 5.12: Distribuição média do número de canais usados (figura 5.12a) e das conexões ativas (figura 5.12b), de acordo com a largura de banda dos canais, das soluções com 512 conexões obtidas pelas heurísticas propostas.

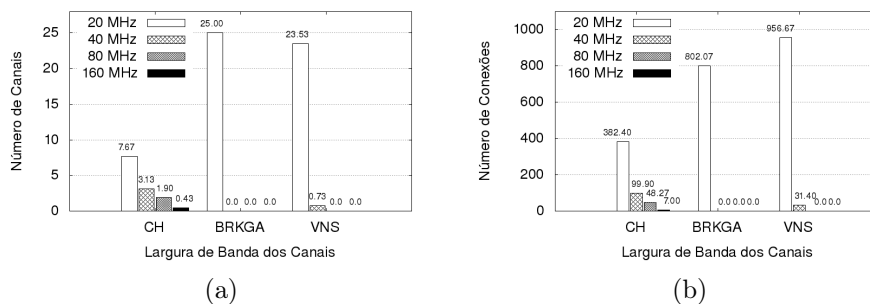


Figura 5.13: Distribuição média do número de canais usados (figura 5.13a) e das conexões ativas (figura 5.13b), de acordo com a largura de banda dos canais, das soluções com 1024 conexões obtidas pelas heurísticas propostas.

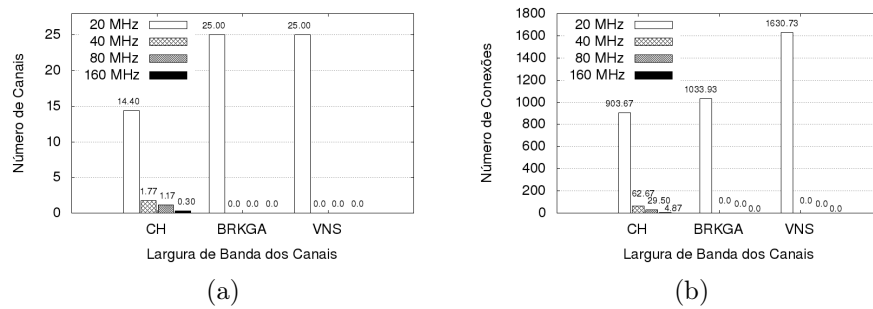


Figura 5.14: Distribuição média do número de canais usados (figura 5.14a) e das conexões ativas (figura 5.14b), de acordo com a largura de banda dos canais, das soluções com 2048 conexões obtidas pelas heurísticas propostas.

Os dados apresentados nas figuras que vão de 5.8 a 5.14 mostram que o percentual de conexões que são escalonadas pelas heurísticas para os canais com larguras de banda menores aumenta à medida que cresce a densidade da rede. No caso das redes menores que possuem 32 conexões, a maioria das conexões foram escalonadas pelas heurísticas para transmitir em canais com largura de banda iguais a 160 MHz e 80 MHz. Além disso, os canais com largura de banda iguais a 160 MHz são os que possuem mais conexões transmitindo neles. De acordo com os resultados obtidos pelas heurísticas para redes com 64 e 128 conexões, os canais com largura de banda iguais a 80 MHz possuem uma quantidade maior de conexões transmitindo neles do que os canais cuja largura de banda é de 160 MHz. Em relação as redes com 128 conexões foi visto que as heurísticas BRKGA e VNS foram capazes de escalonar mais conexões do que a heurística CH, porém não utilizaram canais com largura de banda igual a 160 MHz, enquanto que CH escalonou uma quantidade menor de conexões e utilizou canais com larguras de banda maiores. Já nas redes com 256 e 512 conexões foi visto que as heurísticas BRKGA e VNS escalonaram a maior parte das conexões para canais com larguras de banda iguais a 20 MHz e 40 MHz. Percebeu-se também que, no caso das redes com 512 conexões, os canais cuja largura de banda é igual a 20 MHz possuem uma quantidade significativamente maior de conexões que transmitem neles em comparação com os canais cuja largura de banda é igual a 40 MHz. Apesar de terem utilizado canais



com largura de banda maior, as soluções obtidas pela heurística CH apresentaram uma quantidade menor de conexões escalonadas. Por fim, os resultados obtidos pelas heurísticas, para redes com 1024 e 2048 conexões, mostram que a maioria das conexões foram atribuídas a canais menores cuja largura de banda é igual a 20 MHz.

A alta interferência que há entre as conexões nas instâncias do conjunto *D250* faz com que o receptor de cada conexão tenha mais dificuldade para decodificar os dados que são enviados para ele, principalmente nas redes com mais conexões e em canais com largura de banda maiores. Assim, à medida que o número de conexões aumenta e conseqüentemente a interferência, as heurísticas BRKGA e VNS buscam escalonar o maior número possível de conexões capazes de transmitir com o intuito de maximizar o *throughput* da rede para o canais com largura de banda igual a 20 MHz e 40 MHz também, ao invés de simplesmente escalonar uma quantidade menor de conexões para os canais de 80 MHz e 160 MHz.



# Capítulo 6

## Conclusões

Nesta tese foi estudado o Problema de Escalonamento de Conexões com Múltiplas Velocidades de Transmissão e em Canais com Largura de Banda Variável (VRBSP) em redes sem fio. Com o objetivo de oferecer algoritmos capazes de encontrar soluções para o VRBSP e o seu caso especial o VRSP, foram propostas duas formulações de MILP, denominadas F1 e F2. Também foram propostas duas heurísticas baseadas nas metaheurísticas BRKGA e VNS, além de uma heurística construtiva (CH), para encontrar soluções do VRBSP. Experimentos computacionais foram realizados para avaliar o desempenho dos algoritmos de *branch and cut* baseados nas formulações F1 e F2, denominados B&C-F1 e B&C-F2, e estes algoritmos foram comparados com o melhor algoritmo da literatura para o VRSP, denominado Datarate PPTAS [Goussevskaia et al., 2016]. Além disso, foram realizados experimentos com as heurísticas BRKGA, CH e VNS para avaliar a qualidade de suas respectivas soluções.

Os resultados para o VRSP mostraram que o Datarate PPTAS não foi computacionalmente eficiente nas instâncias testadas, já que este foi eficiente apenas para encontrar soluções (não-ótimas) em instâncias do conjunto  $D10000$  com até 64 conexões, e em nenhuma das instâncias do conjunto  $D250$ . O algoritmo B&C-F2 também mostrou tais problemas de desempenho. Devido ao grande número de variáveis desta

formulação, o mesmo foi capaz de executar apenas para instâncias com até 64 conexões. No entanto, a maior parte destas soluções encontradas foram ótimas. Os melhores resultados foram obtidos por B&C-F1, que foi capaz de encontrar soluções ótimas para todas as instâncias do conjunto  $D10000$  com até 1024 conexões e 13 instâncias do conjunto  $D250$  com até 128 conexões.

Os resultados para VRBSP mostraram que os melhores resultados foram novamente obtidos por B&C-F1, que foi capaz de encontrar soluções ótimas para todas as instâncias do conjunto  $D10000$  com até 256 conexões, enquanto que B&C-F2 foi capaz de executar apenas para instâncias com até 64 conexões. Em relação as instâncias do conjunto  $D250$ , ambos algoritmos de *branch and cut* foram capazes de encontrar soluções ótimas apenas para as instâncias com até 8 conexões. O limites superiores obtidos por B&C-F2, em até 3600 segundos foram melhores do que os obtidos por B&C-F1 para as instâncias de tal conjunto com 16, 32, e 64 conexões, que mostraram que a relaxação linear dos limites superiores de F2 são de fato melhores que as de F1. No entanto, as soluções (limites inferiores) obtidas por B&C-F1 dentro deste limite de tempo foram melhores que as encontradas por B&C-F2, o que demonstra a relação custo benefício entre o maior número de variáveis da formulação F2 e seu respectivo ganho por meio da relaxação linear não foi grande o suficiente para que F2 fosse melhor do que F1 nas instâncias testadas. Neste caso, o bom desempenho apresentado por B&C-F1 se deve ao fato de que o número de variáveis da formulação F1 na qual ele se baseia cresce linearmente e é menor, em comparação a F2. Isto faz com que B&C-F1 seja capaz de resolver F1 de forma mais rápida em cada nó da árvore de *branch and cut* e conseqüentemente ter um melhor desempenho. No caso dos resultados apresentados pelas heurísticas percebeu-se que VNS obteve os melhores resultados, principalmente para as instâncias maiores, e que ela melhora significativamente a qualidade das soluções obtidas pela heurística construtiva. O VNS apresentou soluções melhores do que as demais heurísticas pois ele parte de uma solução inicial de boa qualidade que é gerada pela heurística construtiva e realiza de forma sucessiva a perturbação e a

busca local para explorar o espaço de busca para obter novas soluções. Além disso, a perturbação e busca local foram implementadas de forma que ao trabalharem em conjunto elas fossem capazes de explorar as diferentes regiões do espaço de busca de forma eficiente. Desta forma, a perturbação e as estruturas de vizinhança usadas por ela permitiram a geração de novos vizinhos da solução corrente e ajudaram o VNS a escapar de ótimos locais. Já a busca local permitiu a obtenção novos ótimos locais com o auxílio do algoritmo 6 que computa o particionamento localmente ótimo dos canais em tempo polinomial. Assim, percebeu-se por meio dos resultados obtidos que os algoritmos exatos e heurísticos propostos foram capazes de obter soluções viáveis e de boa qualidade para esta versão do problema que considera canais com largura de banda variáveis.

Como trabalhos futuros podem ser estudadas outras abordagens para linearizar a formulação não-linear inteira mista (3.1)-(3.7). Abordagens baseadas em *lifting* [Balas, 2005] podem ser aplicadas para melhorar os limites superiores de F1, e algoritmos de Geração de Colunas [Desaulniers et al., 2006] podem ser aplicados para computar eficientemente a relaxação linear de F2 para instâncias maiores. A hibridização das formulações com heurísticas bem como o desenvolvimento de novas heurísticas e metaheurísticas [Talbi, 2009] também podem ser estudadas. Sugere-se também o estudo de variantes de VRBSP que consideram a utilização de múltiplos intervalos de tempo e o controle de potência dos transmissores para o escalonamentos de conexões. Além do mais, recomenda-se o estudo de redes sem fio multi-serviços (do inglês *Multi-Service Wireless Networks*) [Carvalho et al., 2008] que, por sua vez, suportam diversos tipos de serviços como os de multimídia, telefonia sobre IP e Internet, sendo que cada um deles possui diferentes requisitos e demandas para operarem.



# Referências Bibliográficas

- Abdelkhalek, O.; Masri, H. & Krichen, S. (2015). An adaptive variable neighborhood search for solving the multi-objective node placement problem. *Electronic Notes in Discrete Mathematics*, 47:189–196. The 3rd International Conference on Variable Neighborhood Search (VNS'14).
- Am, V.; Currie, J. & Wilson, D. I. (2017). A systematic approach to modeling organic rankine cycle systems for global optimization. Em *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pp. 487--492. IEEE.
- Andrade, C. E.; Resende, M. G.; Zhang, W.; Sinha, R. K.; Reichmann, K. C.; Doverspike, R. D. & Miyazawa, F. K. (2015). A biased random-key genetic algorithm for wireless backhaul network design. *Applied Soft Computing*, 33:150 – 169.
- Andrews, M. & Dinitz, M. (2009). Maximizing capacity in arbitrary wireless networks in the sinr model: Complexity and game theory. Em *International Conference on Computer Communications (INFOCOM)*, pp. 1332–1340. IEEE.
- Arenales, M.; Morabito, R.; Armentano, V. & Yanasse, H. (2015). *Pesquisa operacional: para cursos de engenharia*. Elsevier Brasil.
- Arora, S.; Karger, D. & Karpinski, M. (1999). Polynomial time approximation schemes for dense instances of np-hard problems. *Journal of Computer and System Sciences*, 58:193–210.

- Ásgeirsson, E. I.; Halldórsson, M. M. & Tonoyan, T. (2017). Universal Framework for Wireless Scheduling Problems. Em *International Colloquium on Automata, Languages, and Programming (ICALP)*, Leibniz International Proceedings in Informatics (LIPIcs), pp. 129:1–129:15. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Balas, E. (2005). Projection, lifting and extended formulation in integer and combinatorial optimization. *Annals of Operations Research*, 140(1):125.
- Bazaraa, M. S.; Jarvis, J. J. & Sherali, H. D. (2011). *Linear programming and network flows*. John Wiley & Sons.
- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 2:154–160.
- Bellman, R. (1958). On a routing problem. *Quarterly of applied mathematics*, 16:87–90.
- Bisschop, J. (2006). *AIMMS - Optimization Modeling*. Lulu.com.
- Brandão, J. S.; Noronha, T. F. & Ribeiro, C. C. (2016). A biased random-key genetic algorithm to maximize the number of accepted lightpaths in wdm optical networks. *Journal of Global Optimization*, 65:813–835.
- Brar, G.; Blough, D. M. & Santi, P. (2006). Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. Em *International Conference on Mobile Computing and Networking (Mobicom)*, pp. 2–13, Los Angeles, CA, USA. ACM.
- Buriol, L. S.; Resende, M. G.; Ribeiro, C. C. & Thorup, M. (2005). A hybrid genetic algorithm for the weight setting problem in ospf/is-is routing. *Networks: An International Journal*, 46(1):36–56.
- Carvalho, G. H.; Martins, V. S.; Francês, C. R.; Costa, J. C. & Carvalho, S. V. (2008). Performance analysis of multi-service wireless network: An approach integrating cac,



- scheduling, and buffer management. *Computers & Electrical Engineering*, 34(4):346-356.
- Castaño, F.; Rossi, A.; Sevaux, M. & Velasco, N. (2014). A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints. *Computers & Operations Research*, 52:220 – 230. Recent advances in Variable neighborhood search.
- Chen, Y.-Y. & Chen, C. (2015). Simulated annealing for interface-constrained channel assignment in wireless mesh networks. *Ad Hoc Networks*, 29:32–44.
- Chilukuri, S. & Sahoo, A. (2015). Delay-aware tdma scheduling for multi-hop wireless networks. Em *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, pp. 27:1–27:10. ACM.
- Cisco (2012). *802.11ac: The Fifth Generation of Wi-Fi. Technical White Paper*. Cisco and its affiliates.
- Codato, G. & Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54:756–766.
- Cormen, T. H. (2009). *Introduction to algorithms*. MIT press.
- Desaulniers, G.; Desrosiers, J. & Solomon, M. M. (2006). *Column generation*, volume 5. Springer Science & Business Media.
- Djukic, P. & Valaee, S. (2009). Delay aware link scheduling for multi-hop tdma wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 17:870–883.
- Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Ephremides, A. & Truong, T. V. (1990). Scheduling broadcasts in multihop radio networks. *IEEE Transactions on communications*, 38:456–460.

- Erzin, A.; Mladenovic, N. & Plotnikov, R. (2017). Variable neighborhood search variants for min-power symmetric connectivity problem. *Computers & Operations Research*, 78:557 – 563.
- Ford Jr, L. R. (1956). Network flow theory. Relatório técnico, The Rand Corporation.
- Gast, M. S. (2013). *802.11 ac: a survival guide: Wi-Fi at gigabit and beyond*. O'Reilly Media, Inc.
- Gonçalves, J. F. & Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Goussevskaia, O.; Halldórsson, M. M.; Wattenhofer, R. & Welzl, E. (2009). Capacity of arbitrary wireless networks. Em *International Conference on Computer Communications (INFOCOM)*, pp. 1872–1880. IEEE.
- Goussevskaia, O.; Oswald, Y. A. & Wattenhofer, R. (2007). Complexity in geometric sinr. Em *International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 100–109. ACM.
- Goussevskaia, O.; Vieira, L. F. & Vieira, M. A. (2016). Wireless scheduling with multiple data rates: From physical interference to disk graphs. *Computer Networks*, 106:64–76.
- Goussevskaia, O.; Vieira, L. F. M. & Vieira, M. A. M. (2012). Wireless multi-rate scheduling: From physical interference to disk graphs. Em *Conference on Local Computer Networks*, pp. 651–658. IEEE.
- Gupta, P. & Kumar, P. R. (2000). The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404.
- Halldórsson, M. M. & Mitra, P. (2011). Wireless capacity with oblivious power in general metrics. Em *Proceedings of the twenty-second annual ACM-SIAM symposium on*

- Discrete Algorithms*, pp. 1538–1548, San Francisco, California. Society for Industrial and Applied Mathematics.
- Halldórsson, M. M. & Mitra, P. (2012). Wireless capacity and admission control in cognitive radio. Em *International Conference on Computer Communications (INFOCOM)*, pp. 855–863. IEEE.
- Henry, D. (2010). *CCNP Wireless IUWMS Quick Reference*. Cisco Press.
- Hertz, A. & Widmer, M. (2003). Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operation Research Society*, 151:247–252.
- IEEE Computer Society (2013). *Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*. Technical Report.
- Johnson, D. S. (1973). *Near-optimal bin packing algorithms*. Tese de doutorado, Massachusetts Institute of Technology.
- Kesselheim, T. (2011). A constant-factor approximation for wireless capacity maximization with power control in the sinr model. Em *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1549–1559, San Francisco, California. Society for Industrial and Applied Mathematics.
- Kesselheim, T. (2012). Approximation algorithms for wireless link scheduling with flexible data rates. Em *European Symposium on Algorithms*, pp. 659–670. Springer.
- Lanza-Gutierrez, J. M. & Gomez-Pulido, J. A. (2016). Studying the multiobjective variable neighbourhood search algorithm when solving the relay node placement problem in wireless sensor networks. *Soft Computing*, 20(1):67--86.

- Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, 8(2):219--223.
- Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3--30.
- Mladenović, N. & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097--1100.
- Moura, H.; Bessa, G. V.; Vieira, M. A. & Macedo, D. F. (2015). Ethanol: Software defined networking for 802.11 wireless networks. Em *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pp. 388--396. IEEE.
- Network, A. (2008). Retail wireless networks validated reference design solution datasheet.
- Noronha, T. F.; Resende, M. G. & Ribeiro, C. C. (2011). A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, 50(3):503--518.
- Ong, E. H.; Knecht, J.; Alanen, O.; Chang, Z.; Huovinen, T. & Nihtilä, T. (2011). Ieee 802.11ac: Enhancements for very high throughput wlans. Em *International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 849--853, Toronto, ON, Canada. IEEE.
- Plotnikov, R.; Erzin, A. & Mladenovic, N. (2016). Variable neighborhood search-based heuristics for min-power symmetric connectivity problem in wireless networks. Em Kochetov, Y.; Khachay, M.; Beresnev, V.; Nurminski, E. & Pardalos, P., editores, *Discrete Optimization and Operations Research*, pp. 220--232, Cham. Springer International Publishing.

- Powell, S. G. & Baker, K. R. (2009). *Management science: The art of modeling with spreadsheets*. Wiley.
- Rappaport, T. S. (2002). *Wireless communications: principles and practice*. Prentice Hall, 2nd.
- Reis, R.; Ritt, M.; Buriol, L. S. & Resende, M. G. (2011). A biased random-key genetic algorithm for ospf and deft routing to minimize network congestion. *International Transactions in Operational Research*, 18(3):401--423.
- Riley, G. F. (2003). The georgia tech network simulator. Em *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp. 5–12. ACM.
- Spears, W. & deJong, K. (1991). On the virtues of parameterized uniform crossover. Em *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230--236, San Mateo.
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley Publishing.
- Tanenbaum, A. S. & Wetherall, D. (2011). *Computer networks, 5-th edition*. Pearson Education.
- Vieira, F. R.; de Rezende, J. F. & Barbosa, V. C. (2016). Scheduling wireless links by vertex multicoloring in the physical interference model. *Computer Networks*, 99:125–133.
- Wolsey, L. (1998). *Integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley.