



**UNIVERSIDADE FEDERAL DO AMAZONAS**  
**INSTITUTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**MODELOS TEÓRICOS E ALGORITMOS PARA A OTIMIZAÇÃO DA ALOCAÇÃO  
DE CANAIS EM REDES MÓVEIS SEM FIO**

**BRUNO RAPHAEL CARDOSO DIAS**

Março de 2014

Manaus - AM

BRUNO RAPHAEL CARDOSO DIAS

MODELOS TEÓRICOS E ALGORITMOS PARA A OTIMIZAÇÃO DA ALOCAÇÃO  
DE CANAIS EM REDES MÓVEIS SEM FIO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientadora: Rosiane de Freitas Rodrigues, Dr<sup>a</sup>.

Co-orientador: Nelson Maculan Filho, Dr.

Março de 2014

Manaus - AM

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

D541m	Dias, Bruno Raphael Cardoso Modelos teóricos e algoritmos para a otimização da alocação de canais em redes móveis sem fio / Bruno Raphael Cardoso Dias. 2014 102 f.: il. color; 29,7 cm.  Orientadora: Rosiane de Freitas Rodrigues Coorientador: Nelson Maculan Filho Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.  1. algoritmos. 2. telecomunicações. 3. otimização combinatória. 4. colorações em grafos. I. Rodrigues, Rosiane de Freitas II. Universidade Federal do Amazonas III. Título
-------	---



**PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



# **FOLHA DE APROVAÇÃO**

**"Modelos teóricos e algoritmos para a otimização da alocação de canais em redes móveis sem fio"**

**BRUNO RAPHAEL CARDOSO DIAS**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Professores:

PROFA. ROSIANE DE FREITAS RODRIGUES – PRESIDENTE

PROFA. FABÍOLA GUERRA NAKAMURA – MEMBRO

PROF. HORÁCIO ANTONIO BRAGA FERNANDES DE OLIVEIRA – MEMBRO

PROF. LUIS SATORU OCHI – MEMBRO

Manaus, 20 de março de 2014

*A minha família  
e aos meus verdadeiros amigos.*

# Agradecimentos

Agradeço, inicialmente, aos meus pais, Maria da Conceição Cardoso Dias e Tomaz da Silva Dias, por todo o esforço e dedicação dispensados na minha criação. Sem o apoio integral deles em toda a minha vida, eu não teria chegado até aqui.

A minha irmã, Bianca Gabriela Cardoso Dias, que apesar das nossas divergências ideológicas, temos, em primeiro lugar, o amor fraterno mútuo.

Aos meus amigos, que me ajudaram nos momentos de alegria e dor, e permitiram que eu pudesse aproveitar a vida de diferentes formas que eu sozinho não conseguiria. Em especial, deixo o agradecimento ao amigo Ítalo Veloso, que está sempre ao meu lado em tudo que preciso, mas amo de coração todos os quais chamo de amigos verdadeiros.

A minha orientadora, Rosiane de Freitas Rodrigues, pelo trabalho dedicado a minha formação acadêmica desde que fui seu aluno de graduação e pelo apoio no caminho trilhado rumo ao meu sonho profissional.

Aos meus colegas do grupo de Otimização, Algoritmos e Complexidade Computacional do IComp/UFAM, que me fizeram companhia nos momentos de trabalho e descontração em vários momentos dos dois anos de estudo e pesquisa, mesmo que por um curto período de participação no grupo.

À secretaria do Instituto de Computação da UFAM, que sempre foi prestativa nas mais diversas solicitações feitas durante este mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo indispensável suporte financeiro por meio de bolsa de estudo.

Resumo da Dissertação apresentada ao PPGI/IComp/UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática.

MODELOS TEÓRICOS E ALGORITMOS PARA A OTIMIZAÇÃO DA ALOCAÇÃO  
DE CANAIS EM REDES MÓVEIS SEM FIO

Bruno Raphael Cardoso Dias

Março/2014

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Rosiane de Freitas Rodrigues

Co-orientador: Prof. Dr. Nelson Maculan Filho

O problema de alocação de canais é abordado, onde, dada uma rede móvel sem fio com antenas de transmissão distribuídas na região de interesse e dada uma ou mais faixa de frequência limitada discretizada em canais de transmissão, consiste em promover uma alocação de tais canais pelas antenas de tal modo a atender as chamadas em demanda otimizando o uso dos recursos, que neste caso priorizou-se a otimização do uso dos canais alocados, em um problema de otimização Min-Max da distribuição dos canais - o *span* -, onde o maior canal alocado deve ser o menor possível. Tal problema possui uma importância cada vez maior dado o grande crescimento da demanda e a limitação dos recursos tecnológicos de comunicação envolvidos. A abordagem ao problema é de Otimização Combinatória e áreas afins. Sendo assim, é apresentado um estudo da literatura sobre o tema, com enfoque em redes celulares e redes baseadas em rádios cognitivos. A partir disto, propõe-se novos modelos teóricos para representação do problema utilizando colorações especiais em grafos, escalonamento de tarefas em máquinas paralelas com restrições de recursos e geometria de distâncias com programação por restrições, sendo possível identificar características específicas de alguns cenários de aplicação do problema geral. Com base em tais modelos, são apresentados os algoritmos desenvolvidos e implementados, sendo métodos aproximados, baseados em busca local com ênfase na meta-heurística *simulated annealing*, e métodos exatos, envolvendo *branch-and-cut* com a ferramenta IBM/ILOG CPLEX e, por fim, métodos híbridos, *branch-prune-and-bound*.

Os experimentos computacionais realizados são apresentados com uma análise comparativa de desempenho, usando tanto instâncias clássicas da literatura, como o conjunto *Philadelphia* e suas variantes, como também instâncias artificiais propostas para contemplar variantes abordadas, bem como de maior tamanho, envolvendo redes entre 70 a 150 estações. Os resultados obtidos validam os modelos teóricos propostos e os algoritmos desenvolvidos e implementados, uma vez que, resultados iguais ou melhores aos da literatura foram obtidos, com várias soluções ótimas comprovadas, além da discussão teórica e variantes propostas que se acredita robustecer o entendimento do problema e a literatura relacionada.

**Palavras-chave:** algoritmos, geometria de distâncias, coloração em grafos, escalonamento de tarefas, otimização combinatória, telecomunicações.



Abstract of Dissertation presented to PPGI/IComp/UFAM as a partial fulfillment of the requirements for the degree of Master in Informatics.

THEORETICAL MODELS AND ALGORITHMS FOR OPTIMIZING CHANNEL  
ASSIGNMENT IN WIRELESS MOBILE NETWORKS

Bruno Raphael Cardoso Dias

March/2014

Advisor: Prof. Dr. Rosiane de Freitas Rodrigues

Co-advisor: Prof. Dr. Nelson Maculan Filho

The channel assignment problem is explored, where, given a wireless mobile network with transmitters distributed in a region of interest and one or more limited frequency bands discretized into channels, the objective is to assign these channels to the transmitters in order to satisfy demands from calls, optimizing resource usage, which was considered in this work as assigned channels in a Min-Max optimization problem for channel distribution - the span, where the maximum assigned channel must be the lowest possible. This problem has great importance since demand for wireless networks is growing, but the technological resources are limited. The approach to the problem is from Combinatorial Optimization and other related areas. A literature study about the subject is made, with emphasis on cellular and cognitive radio networks. New theoretical models are proposed to represent the problem using special graph colorings, resource constrained scheduling on parallel machines and distance geometry with constraint programming, where some specific characteristics from some application scenarios were identified. Based on these models, algorithms were developed and implemented such as local search based methods, with emphasis on simulated annealing, and exact methods, involving branch-and-cut with IBM/ILOG CPLEX, and hybrid methods, branch-prune-and-bound. Computational experiments are presented with a comparative performance analysis, using both classic instances from the literature, such as Philadelphia set and its variants, and new artificial instances proposed to explore variations of the

problem, and larger sets of instances with networks using 70 to 150 transmitters. The obtained results validate the theoretical models proposed and implemented algorithms, since results equal to or better than the ones from the literature were obtained, with many proven optimal solutions, and also the theoretical discussion and proposed variants that should make the understanding of the problem and related literature more robust.

**Keywords:** combinatorial optimization, distance geometry, graph coloring, scheduling, telecommunications.

# Sumário

<b>Lista de Figuras</b>	<b>xii</b>
<b>Lista de Tabelas</b>	<b>xiv</b>
<b>Lista de Algoritmos</b>	<b>xvi</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Fundamentos teóricos</b>	<b>3</b>
2.1 Otimização combinatória . . . . .	3
2.2 Teoria dos grafos . . . . .	10
2.3 Escalonamento de tarefas . . . . .	14
2.4 Geometria de distâncias . . . . .	16
2.5 Notas do capítulo . . . . .	17
<b>3 Alocação de canais em redes móveis sem fio</b>	<b>18</b>
3.1 Redes móveis celulares . . . . .	18
3.1.1 Estrutura de uma rede celular . . . . .	19
3.1.1.1 Tecnologias de telefonia celular . . . . .	23
3.1.2 Problemas de alocação de canais em redes celulares . . . . .	27
3.1.2.1 Critérios de otimização . . . . .	29
3.1.2.2 Ambientes de teste . . . . .	32
3.2 Redes baseadas em rádios cognitivos . . . . .	39
3.2.1 Estrutura de uma rede cognitiva . . . . .	40
3.2.1.1 Sensoriamento espectral . . . . .	40
3.2.2 Problemas de alocação de canais em redes de rádios cognitivos . .	43
3.2.2.1 Critérios de otimização . . . . .	45

3.2.2.2	Ambientes de teste . . . . .	48
3.3	Notas do capítulo . . . . .	50
<b>4</b>	<b>Problema da alocação de canais Min-Max</b>	<b>51</b>
4.1	Modelagem como colorações especiais de vértices em grafos . . . . .	52
4.2	Modelagem em escalonamento de tarefas em máquinas paralelas com res- trições de recursos . . . . .	53
4.3	Modelagem em geometria de distâncias . . . . .	55
4.4	Notas do capítulo . . . . .	57
<b>5</b>	<b>Algoritmos desenvolvidos e implementados</b>	<b>59</b>
5.1	Meta-heurísticas baseadas em busca local . . . . .	59
5.1.1	Busca local iterada . . . . .	62
5.1.2	<i>Simulated annealing</i> . . . . .	64
5.2	Métodos enumerativos exatos e aproximados . . . . .	66
5.2.1	<i>Branch-and-cut</i> usando CPLEX . . . . .	66
5.2.2	<i>Branch-prune-and-bound</i> . . . . .	67
5.3	Notas do capítulo . . . . .	74
<b>6</b>	<b>Experimentos computacionais</b>	<b>77</b>
6.1	<i>Simulated annealing</i> e <i>branch-and-cut</i> via CPLEX . . . . .	77
6.1.1	Instâncias adicionais . . . . .	79
6.2	<i>Branch-prune-and-bound</i> . . . . .	82
6.3	Notas do capítulo . . . . .	85
<b>7</b>	<b>Considerações Finais</b>	<b>87</b>
	<b>Referências</b>	<b>90</b>
<b>A</b>	<b>Lista de publicações e eventos</b>	<b>101</b>

# Lista de Figuras

2.1	Exemplo de aplicação do <i>branch-and-bound</i> . . . . .	7
2.2	Exemplo de aplicação de planos de corte. . . . .	7
2.3	Exemplo de aplicação do <i>branch-and-cut</i> , onde a cada subproblema com solução contínua, é aplicado um corte de Gomory. . . . .	8
2.4	Ilustração do processo da busca local iterada. . . . .	10
2.5	Processo metalúrgico de recozimento ( <i>annealing</i> ). . . . .	10
2.6	Representação gráfica de um grafo não-direcionado planar com 7 vértices e 10 arestas. . . . .	11
2.7	Exemplo de grafo completo com 4 vértices. . . . .	12
2.8	Coloração dos vértices da Figura 2.6. . . . .	12
2.9	Coloração dos vértices da Figura 2.6 com modelos mais gerais. . . . .	13
3.1	Topo de uma torre de rádio, onde os painéis funcionam como antenas e um dispositivo de ERB modelo Ericsson RBS6000. . . . .	20
3.2	Exemplo de rede de telefonia móvel celular. . . . .	20
3.3	Tipos de padrões de células. . . . .	21
3.4	Conjunto de células hexagonais. . . . .	21
3.5	Divisão da faixa GSM-850 em 125 canais. . . . .	21
3.6	Exemplos do uso de protocolos de acesso ao meio. . . . .	23
3.7	Exemplo de um problema de alocação de canais em uma rede com 6 células. . . . .	28
3.8	Rede de 21 células utilizada nas instâncias <i>Philadelphia</i> . . . . .	33
3.9	Rede de 55 células usada nas instâncias artificiais. . . . .	38
3.10	Utilização da faixa de 0 a 6 GHz do espectro eletromagnético no Centro de Berkeley, Estados Unidos. Adaptado de Yang, 2005 [93]. . . . .	39
3.11	Exemplos dos cenários nos quais a detecção de transmissores pode falhar. . . . .	42

3.12	Modelo de temperatura de interferência. Adaptado de Akyildiz <i>et al.</i> , 2006 [3]. . . . .	43
3.13	Esquema do conceito de buracos espectrais. Adaptado de Akyildiz <i>et al.</i> , 2008 [4]. . . . .	44
3.14	Esquema de alocação <i>overlay</i> . . . . .	45
3.15	Esquema de alocação <i>underlay</i> . . . . .	46
3.16	Uma placa para rádio definido por software. . . . .	49
4.1	Problema de alocação de canais da figura 3.7 modelado como $(\gamma, \mu)$ - <i>channel-muticoloring</i> . . . . .	53
4.2	Exemplo de correlação entre escalonamento de tarefas e coloração em grafos. . . . .	54
4.3	Problema de alocação de canais da figura 3.7 modelado como o problema escalonamento de tarefas $P \mid p_k = 1, r_k, \bar{d}_k, s_{km} \mid C_{max}$ . . . . .	55
4.4	Exemplo de alocação de canais respeitando restrições de distâncias . . . . .	56
5.1	Exemplo de sequência de chamadas e a alocação de canais correspondente obtida pelo algoritmo FEA. . . . .	62
5.2	Operação de troca em uma sequência de chamadas . . . . .	64
5.3	Grafo a ser usado como exemplo para o <i>branch-prune-and-bound</i> . . . . .	69
5.4	Aplicação do <i>branch-prune-and-bound</i> para o grafo da Figura 5.3. . . . .	73
5.5	Esquema de seleção de cores baseado na verificação de todas as distâncias aos vizinhos. . . . .	74
6.1	Número de vértices $\times$ tempo de execução do algoritmo B&P&B nos problemas usados. . . . .	85

# Lista de Tabelas

3.1	Presença do serviço móvel celular em diferentes partes do mundo. . . . .	19
3.2	Estatísticas de usuários de redes celulares no Brasil. . . . .	19
3.3	Instâncias do projeto CALMA . . . . .	34
3.4	Distâncias de separação na matriz de interferências entre células para cada instância do conjunto <i>Philadelphia</i> . . . . .	36
3.5	Vetores de demanda para as instâncias <i>Philadelphia</i> . . . . .	36
3.6	Estações interferentes para cada célula na rede das instâncias <i>Helsinki</i> . . .	37
3.7	Distâncias de separação na matriz de interferências entre células para as instâncias <i>Helsinki</i> . . . . .	38
3.8	Vetores de demanda para as instâncias <i>Helsinki</i> . . . . .	38
3.9	Distâncias de separação na matriz de interferências entre células para as instâncias artificiais. . . . .	38
3.10	Vetores de demanda para as instâncias artificiais. . . . .	39
5.1	Instância de 6 células a ser usada como exemplo para a aplicação do <i>simulated annealing</i> . . . . .	62
6.1	Resultados obtidos com o <i>simulated annealing</i> e o <i>branch-and-cut</i> via CPLEX para instâncias da literatura. . . . .	78
6.2	Dados comparativos dos métodos utilizados com outros da literatura. . . .	79
6.3	Resultados obtidos com o <i>simulated annealing</i> para as instâncias adicionais do conjunto padrão. . . . .	81
6.4	Resultados obtidos com o <i>simulated annealing</i> para as instâncias adicionais do conjunto secundário com interferências mais fortes. . . . .	81
6.5	Resultados obtidos com o <i>simulated annealing</i> e as duas versões do <i>branch-prune-and-bound</i> para as instâncias de colorações com demandas únicas e distâncias variáveis. . . . .	84

6.6	Resultados da execução da terceira versão do algoritmo <i>branch-prune-and-bound</i> aplicado a instâncias dos problemas CMDGP-List e MCDGP-List (listas distintas e iguais). . . . .	84
6.7	Resultados da execução da terceira versão do algoritmo <i>branch-prune-and-bound</i> aplicado a instâncias dos problemas CMDGP-Multi-List e MCDGP-Multi-List. . . . .	85



# Lista de Algoritmos

5.1	Geração da sequência inicial de chamadas. . . . .	60
5.2	<b>FEA:</b> Alocação exaustiva por frequência . . . . .	61
5.3	Busca local iterada para alocação de canais . . . . .	63
5.4	<i>Simulated annealing</i> para alocação de canais . . . . .	65
5.5	<i>Branch-Prune-and-Bound</i> - primeira versão . . . . .	70
5.6	Teste de factibilidade de uma coloração (parcial ou completa) a partir de um vértice. . . . .	71
5.7	<i>Branch-Prune-and-Bound</i> - segunda versão . . . . .	72
5.8	Seleção de cor na segunda versão do <i>branch-prune-and-bound</i> . . . . .	75
5.9	<i>Branch-Prune-and-Bound</i> - terceira versão para (MCDGP-Multi-List) . . . . .	76
6.10	Gerador de instâncias para alocação de canais . . . . .	82

# Capítulo 1

## Introdução

Os problemas de alocação de canais (*channel assignment problems* - CAP) constituem uma importante classe de problemas, tanto por questões teóricas quanto práticas, o que motivou muitos estudos a seu respeito. Em tais problemas, o objetivo é atribuir porções do espectro eletromagnético de acordo com as demandas dos usuários da rede sem fio, de forma a minimizar ou eliminar possíveis interferências que degradam a qualidade do serviço prestado. No entanto, nem sempre houve uma visão de otimização combinatoria sobre o problema, sendo que um dos primeiros trabalhos com essa característica foi publicado em 1980 [43] e desde então esta área vem sendo muito pesquisada [51, 1, 65, 49, 60, 25, 47, 26, 50, 10, 67].

O aumento na quantidade de usuários e assinaturas requer um uso mais eficiente do espectro eletromagnético, de forma que todos consigam acessar o serviço de telefonia móvel celular com qualidade. Além disso, os tipos de tráfego em redes celulares estão se tornando cada vez mais diversificados, indo além da voz e incluindo vídeos, fotos, músicas e Internet, que demandam maiores taxas de transferência de dados. Assim, é de suma importância determinar alocações de canais que otimizem o serviço aos usuários, respeitando demandas e recursos disponíveis.

Além da importância prática, tal cenário é uma importante fonte de pesquisa científica, onde problemas clássicos são usados para a modelagem teórica de variações do CAP, como a classe de problemas de colorações de vértices em grafos, bem como podem ser aplicados métodos exatos e aproximados para a resolução cada vez mais eficiente desta classe de problemas. A questão é justamente o fato de que ainda há poucos trabalhos com enfoque mais teórico sobre tais problemas.

Sendo assim, o principal objetivo da presente pesquisa é fornecer contribuições científicas para a resolução mais eficiente do problema da alocação de canais em redes móveis sem fio em diferentes cenários de aplicação, com ênfase no problema da minimização do maior canal alocado - Min-Max CAP, de tal forma a adaptar e propor novos modelos teóricos de problemas computacionais clássicos, estender e aplicar formulações matemáticas. bem como propor estratégias algorítmicas aplicando o ferramental das áreas de Otimização Combinatória, Teoria dos Grafos e Geometria de Distâncias.

O restante desta dissertação está organizado como descrito a seguir. No Capítulo 2, são introduzidos alguns conceitos teóricos utilizados para entendimento dos problemas pesquisados. No Capítulo 3, são dadas as definições a respeito das redes celulares e cognitivas e os problemas de alocação de canais nas mesmas. No Capítulo 4, são mostrados os modelos teóricos propostos para alocação de canais, por meio de correlações com outros problemas de otimização combinatória. No Capítulo 5, são apresentados os algoritmos desenvolvidos e implementados para solucionar o problema de alocação de canais. No Capítulo 6, são mostrados os resultados empíricos obtidos com a execução dos métodos propostos em instâncias da literatura e adicionais geradas para este trabalho. Por fim, no Capítulo 7, são feitas as considerações finais, sendo sugeridas possíveis linhas de investigação futura na área.

# Capítulo 2

## Fundamentos teóricos

Neste capítulo, serão abordados alguns conceitos necessários para compreensão do trabalho elaborado, envolvendo otimização combinatória, teoria dos grafos e escalonamento de tarefas. Alguns conceitos adicionais, como os presentes nas redes de computadores, serão abordados nos capítulos correspondentes a cada problema estudado.

### 2.1 Otimização combinatória

Um problema de otimização consiste em encontrar a melhor solução dentre um conjunto de candidatos. A qualidade da solução é medida por meio de uma **função objetivo**, sendo que em um problema de minimização, a melhor solução será aquela de menor valor, e em um problema de maximização, a solução será a de maior valor [69]. O formato básico de um problema de otimização é:

$$\text{Minimizar / maximizar } f(x) \tag{2.1}$$

$$\text{Sujeito a } x \in P \tag{2.2}$$

Onde  $x$  é uma solução,  $f$  é a função objetivo e  $P$  é o conjunto de soluções factíveis (viáveis). Este conjunto é definido pelas características e restrições cada problema. O tamanho deste conjunto pode ser contável ou não, sendo que no primeiro caso, o problema é dito ser de otimização combinatória. Sem perda de generalidade, serão considerados problemas de minimização, uma vez que  $\max f(x) = -\min -f(x)$  [41].

Uma ferramenta bastante utilizada para modelagem de problemas de otimização é a

programação matemática, na qual o problema a ser resolvido é enunciado por um conjunto de expressões matemáticas, sendo uma a função objetivo, consistindo no critério a ser otimizado, e as demais as restrições do problema. Uma das principais classes de modelos programação matemática é a programação linear. Nela, a função objetivo é linear (ou seja, um polinômio de grau 1), e o conjunto de soluções é definido por restrições também lineares. Sendo  $n$  e  $m$  dois inteiros positivos, onde  $n$  é o número de variáveis e  $m$  o de restrições;  $A = [a_{ij}]$  uma matriz de dimensão  $n \times m$  (os coeficientes das restrições); um vetor  $c = [c_i] \in \mathbb{R}^n$  (os custos das variáveis em um problema de minimização) e um vetor  $b = [b_i] \in \mathbb{R}^m$  (as demandas), a seguinte formulação é um problema de programação linear:

$$\text{Minimizar } \sum_{i=1}^n c_i x_i \quad (2.1)$$

$$\text{Sujeito a } \sum_{i=1}^n a_{ij} x_i = b_j \quad (\forall j = 1, \dots, m) \quad (2.2)$$

$$x_i \geq 0 \quad (\forall i = 1, \dots, n) \quad (2.3)$$

Neste caso, as variáveis podem assumir valores reais, o que torna o conjunto de soluções que atendem às restrições (chamadas de soluções factíveis) incontável. Porém, existe um método, denominado de simplex, que na prática é bastante eficaz na resolução de problemas de programação linear com todas as variáveis contínuas. Apesar disso, a complexidade de pior caso do mesmo é exponencial, mas na grande maioria dos casos, o algoritmo é o mais rápido empiricamente. Outros métodos incluem o das elipsóides e o de pontos interiores [69, 59].

Nos problemas de otimização combinatória, entretanto, as variáveis devem assumir valores inteiros, uma vez que as soluções possuem elementos discretos. A programação inteira é uma variação da programação linear com essa restrição adicional. Quando apenas algumas variáveis devem ser inteiras, diz-se que o problema é de programação inteira mista. Sendo  $p$  o número de variáveis inteiras e  $q$  o de variáveis não inteiras (onde  $p + q = n$ ), com as variáveis inteiras indexadas por  $l$  e as contínuas por  $k$ , a formulação a seguir é de programação inteira mista:

$$\text{Minimizar } \sum_{l=1}^p c_l x_l + \sum_{k=1}^q c_k x_k \quad (2.1)$$

$$\text{Sujeito a } \sum_{l=1}^p a_{lj} x_l + \sum_{k=1}^q a_{kj} x_k = b_j \quad (\forall j = 1, \dots, m) \quad (2.2)$$

$$x_l \geq 0 \quad (\forall l = 1, \dots, n) \quad (2.3)$$

$$x_k \geq 0 \quad (\forall k = 1, \dots, q) \quad (2.4)$$

Outra técnica de modelagem utilizada é a programação por restrições, cujo foco é encontrar uma solução factível para um conjunto de restrições, e não a solução ótima. Nela, ao invés de uma função objetivo, são considerados os domínios de valores para as variáveis, além das restrições. [92].

Uma de suas principais vantagens é a facilidade de modelagem. As restrições podem assumir várias formas, como aritméticas (sem a necessidade de serem lineares) e lógicas (booleanas), além de globais e simbólicas para subestruturas naturais dos problemas [16, 91]. Restrições deste último tipo costumam já ser implementadas em ferramentas computacionais de programação por restrições, utilizando algoritmos eficientes especializados na satisfação das mesmas. O Catálogo de Restrições Globais (*Global Constraint Catalog*) [15] conta com 354 restrições desse tipo já conhecidas e estabelecidas na literatura e em ferramentas.

Uma das restrições mais utilizadas é conhecida como *AllDifferent*(variáveis), que exige que todos os valores das variáveis consideradas sejam diferentes uns dos outros. Tal restrição exige uma certa expansão do modelo quando traduzida para a programação matemática. Para duas variáveis  $x_i$  e  $x_j$ , a restrição *AllDifferent*( $x_i, x_j$ ) (ou seja,  $x_i \neq x_j$ ) fica do seguinte modo:

$$x_i - x_j + 1 \leq m y_i$$

$$x_j - x_i + 1 \leq m y_j$$

$$y_i + y_j = 1$$

$$y_i, y_j \in \{0, 1\}$$

$$0 \leq x_i, x_j \leq m - 1$$

Apesar da equivalência, além de ser mais simples de ser enunciada, a restrição *AllDifferent*(variáveis) é bastante estudada na literatura e, por isso, possui muitos algoritmos

eficientes disponíveis para seu tratamento específico.

Uma grande parte dos problemas de otimização combinatória pertence à classe **NP-difícil**, o que significa que provavelmente não exista algoritmo eficiente (ou seja, de complexidade polinomial) para resolvê-los. Além disso, o conjunto de soluções é exponencial, o que torna uma busca exaustiva inviável na maioria dos casos. Para resolvê-los, existem duas vertentes de pesquisa: os métodos exatos, que fornecem soluções garantidamente ótimas em um tempo computacional que pode ser grande, e os métodos aproximados, onde as soluções podem não ser ótimas, mas são de boa qualidade e obtidas em um tempo mais satisfatório.

Apesar de o tempo computacional dos métodos exatos ser inviável em alguns casos, há grande interesse em algoritmos desse para problemas NP-difíceis. O uso dos mesmos pode ser útil para verificação de limites inferiores ou mesmo valores de soluções ótimas em instâncias de *benchmark* e também para aplicação em cenários onde a solução desejada deve, obrigatoriamente, ser a melhor possível.

Alguns métodos exatos são:

- **Branch-and-bound:** envolve a utilização de algoritmos para programação linear contínua (como o simplex) no problema com variáveis inteiras, removendo temporariamente as restrições de integralidade (processo chamado de relaxação). Se a solução obtida para o problema relaxado satisfizer as restrições de integralidade já neste momento, a solução é ótima também para o problema original. Caso contrário, sendo  $x_i$  uma variável do problema relaxado que deveria ser inteira no problema original e cujo valor é contínuo (denotado por  $x_i^R$ ), são criados dois subproblemas (passo de ramificação), onde cada um equivale ao problema relaxado original, e cada um recebe uma restrição adicional:

– Problema 1:  $x_i \leq \lfloor x_i^R \rfloor$ .

– Problema 2:  $x_i \leq \lceil x_i^R \rceil$ .

Cada subproblema é resolvido de forma recursiva, até que a melhor solução inteira seja encontrada. Para evitar a enumeração completa dos subproblemas, são usados limites inferior (LI) e superior (LS) em cada subproblema. Se o problema for de minimização, a solução relaxada de um problema será um LI e a melhor solução inteira será um LS (caso o problema seja de maximização, será o contrário). Como

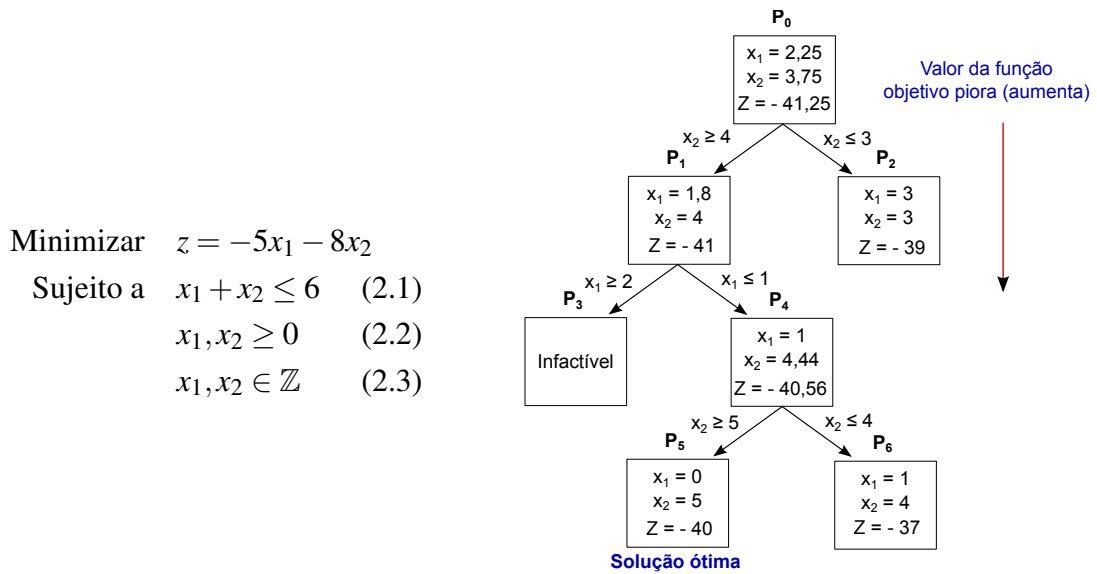


Figura 2.1: Exemplo de aplicação do *branch-and-bound*.

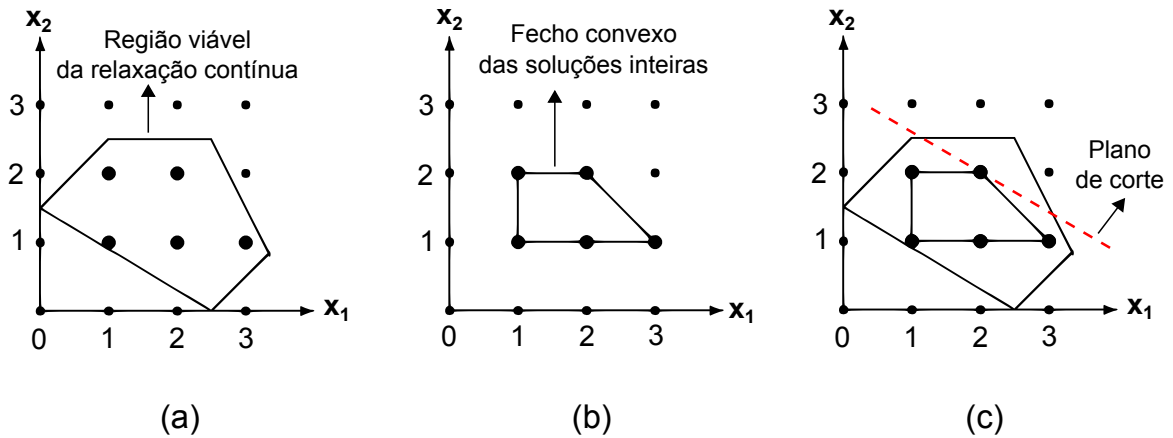
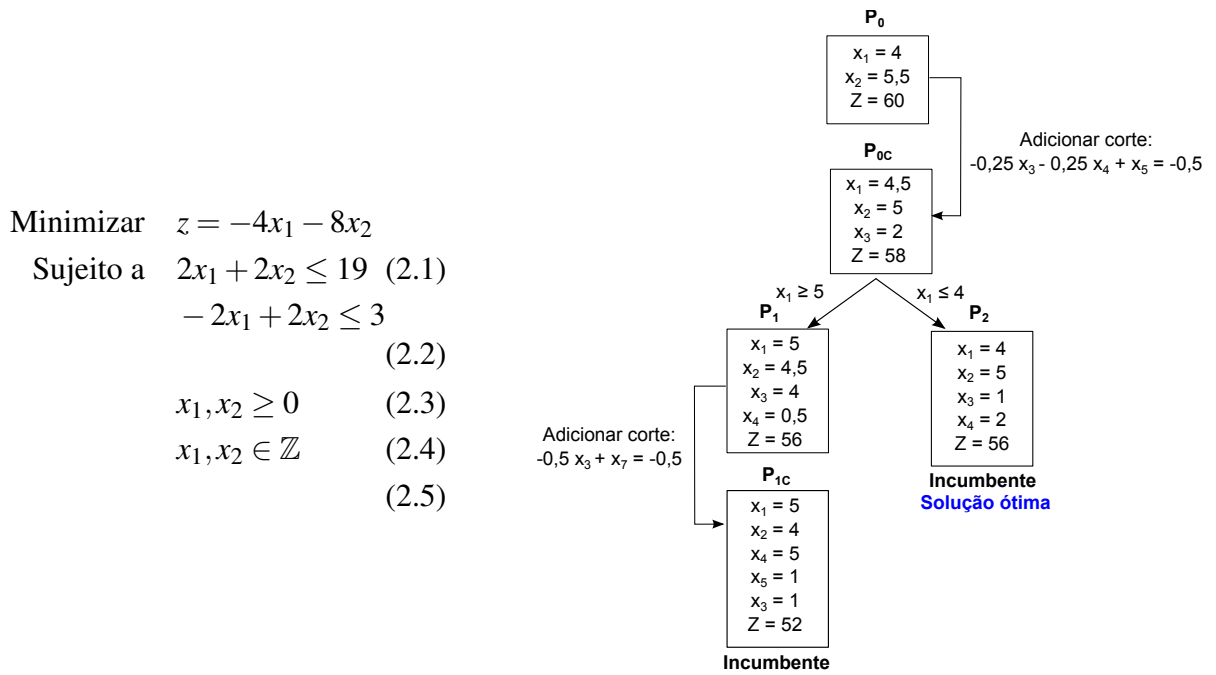


Figura 2.2: Exemplo de aplicação de planos de corte. Em (a), a região viável para a relaxação contínua do problema é marcada. Em (b), o polítopo cujos pontos extremos são apenas soluções inteiras, isto é, o fecho convexo das mesmas, é marcado. Por fim, em (c), um plano de corte (denotado pela linha pontilhada) é aplicado, permitindo que o ponto ótimo inteiro seja atingido. Adaptado de Letchford, 2011 [55].

a adição de restrições tende a piorar a qualidade das soluções, o valor do LI tenderá a aumentar e o do LS a diminuir. Quando  $LI \geq LS$ , não há necessidade de ramificar o subproblema, uma vez que não haverá como obter uma solução melhor. A Figura 2.1 exemplifica o método.

- Planos de corte:** a ideia consiste em adicionar novas restrições ao problema relaxado, de forma que a solução ótima contínua seja removida do conjunto de soluções factíveis, sem remover pontos inteiros do mesmo. Aplicando sucessivos cortes, a solução ótima inteira será encontrada diretamente por métodos de programação linear contínua. A Figura 2.2 exemplifica esse esquema.





**Figura 2.3:** Exemplo de aplicação do *branch-and-cut*, onde a cada subproblema com solução contínua, é aplicado um corte de Gomory.

Existem duas categorias de cortes: os genéricos e os específicos de um problema. Os genéricos são aplicáveis em qualquer problema de programação inteira, como os cortes de Gomory e *zero-half*, no entanto, podem ter convergência lenta por não haver exploração inteligente da região factível do problema. Já os específicos são baseados em características do problema a ser resolvido, por exemplo, garantindo que os vértices de uma clique em um grafo utilizem cores diferentes.

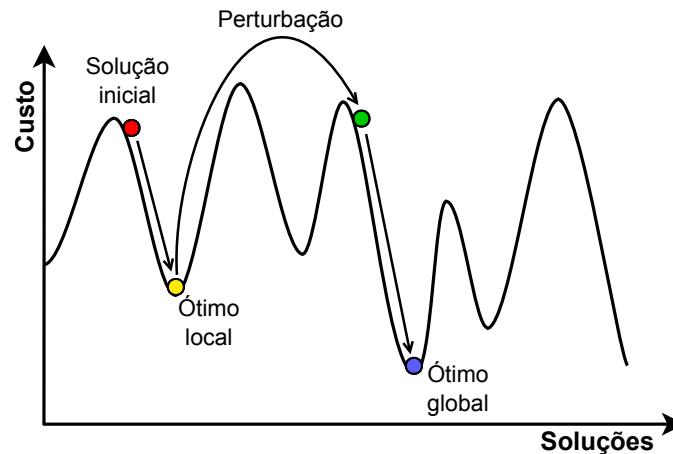
- Branch-and-cut:** combina características do *branch-and-bound* com planos de corte. A cada subproblema gerado no *branch-and-bound*, se a solução relaxada não for inteira, são encontrados cortes que eliminem tal solução, na esperança de que seja encontrada uma solução inteira. A geração de cortes pode seguir diferentes critérios, por exemplo, inserindo uma quantidade máxima de restrições genéricas ou específicas ou efetuando a separação de todos os cortes que podem ser obtidos a partir de uma mesma solução relaxada, ainda que a solução obtida após tal processo ainda não seja inteira, situação na qual será realizada uma ramificação da mesma forma que ocorre no *branch-and-bound* tradicional.
- Branch-and-prune:** este método é utilizado em modelos de programação por restrições. Consiste em atribuir um valor a uma variável de acordo com seu domínio. Em seguida, realiza-se a propagação das restrições (processo também conhecido como

*pruning*), onde valores dos domínios das outras variáveis que violem restrições em relação ao valor atribuído à variável atual são removidos. Se um domínio se tornar vazio, a solução parcial é infactível, assim, é feito um *backtrack* a um passo anterior e um outro valor é testado. Caso o domínio tenha apenas um elemento, ele será o valor da variável. Quando todos os domínios têm um único elemento, tem-se uma solução factível completa. Para que o método seja aplicado à otimização, sendo  $z_0$  o custo da solução obtida, deve-se adicionar uma restrição  $z < z_0$  ao problema [92].

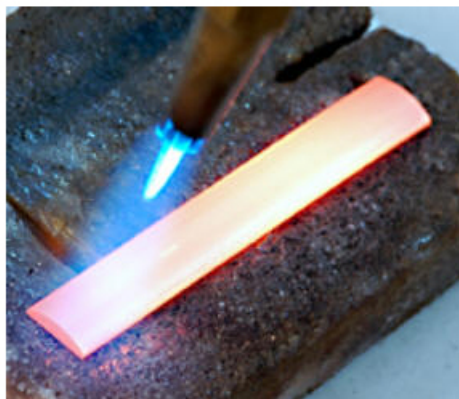
Em muitos casos práticos, mesmo bons métodos exatos podem consumir muito tempo para obter soluções ótimas para os problemas de otimização combinatória. Sendo assim, muitos pesquisadores investem nas abordagens aproximadas, que fornecem boas soluções (mas não necessariamente as melhores possíveis) em um tempo computacional razoável. Uma abordagem aproximada bastante utilizada consiste em algoritmos heurísticos, que exploram o espaço de busca utilizando características desejadas nas melhores soluções para o problema, sem garantias de quão perto da melhor solução o algoritmo chegará.

As meta-heurísticas são abordagens aproximadas mais genéricas, consistindo em *frameworks* algorítmicos que podem ser adaptados à maioria dos problemas de otimização combinatória [58]. Em geral, as meta-heurísticas possuem um passo estocástico, que permite efetuar uma exploração mais eficaz do espaço de soluções. Algumas das principais meta-heurísticas bastante utilizadas são [58, 69]:

- **Busca local iterada** (*iterated local search* - ILS): consiste em encontrar, a partir de uma dada solução, outra solução vizinha (ou seja, acessível por meio de uma perturbação que compõe a vizinhança da solução) de custo melhor. Tal solução é denominada de ótimo local. Este processo é repetido partindo de soluções em porções diferentes do espaço de busca, de forma a obter diferentes ótimos locais, que pode ser a melhor solução de todas, uma vez que todo ótimo global é um ótimo local, como ilustrado na Figura 2.4 [70].
- **Simulated annealing**: meta-heurística inspirada na metalurgia, onde o processo de recozimento (*annealing*), consiste em obter uma estrutura cristalina de um sólido, como na Figura 2.5. O processo é iniciado com uma temperatura alta que será lentamente diminuída até a temperatura final desejada, que será menor que a inicial. Tal método é baseado no algoritmo de Metropolis [62] envolvendo mecânica estatística,



**Figura 2.4:** Ilustração do processo da busca local iterada.



**Figura 2.5:** Processo metalúrgico de recozimento (*annealing*). Fonte: <http://www.rugusavay.com/annealing-process-for-metals-and-glass/>

que foi aplicado a problemas de otimização posteriormente [48], mais especificamente, problemas de particionamento em grafos e desenvolvimento de circuitos digitais. No *simulated annealing*, ao buscar por outras soluções, é possível aceitar soluções piores com probabilidade proporcional à temperatura atual.

Outros métodos meta-heurísticos bastante populares na literatura incluem os algoritmos genéticos, busca tabu, nuvem de partículas, reconexão de caminhos, GRASP (*Greedy Randomized Adaptive Search Procedures*, ou Procedimentos Adaptativos de Busca Gulosa Aleatória), nuvem de partículas, *scatter search*, dentre outros [58, 88, 34].

## 2.2 Teoria dos grafos

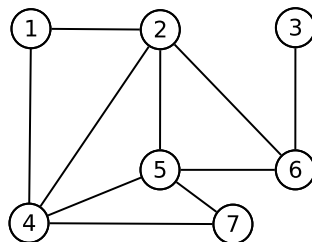
Um grafo é um par ordenado  $G = (V, E)$ , onde  $V$  é um conjunto de vértices e  $E$  é um conjunto de arestas, onde  $n = |V|$  e  $m = |E|$ . Cada elemento  $e$  no conjunto  $E$  é um par

$(i, j)$  que indica que o vértice  $i$  é ligado ao vértice  $j$  (ou seja, são adjacentes, e a aresta  $e$  incide em  $i$  e  $j$ ). O grafo é dito não-direcionado quando os pares que representam as arestas não são ordenados, isto é,  $(i, j) = (j, i)$ . A representação gráfica de um grafo consiste em pontos distintos do plano associados a cada vértice e, para cada aresta  $(i, j)$ , uma linha conectando os pontos correspondentes aos vértices  $i$  e  $j$ . Se for possível efetuar uma representação gráfica de um grafo  $G$  sem que as arestas se cruzem, diz-se que  $G$  é planar [87]. Um exemplo é dado na Figura 2.6.

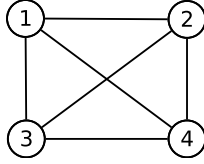
O grau de um vértice  $v$  do grafo equivale à quantidade de arestas que incidem em  $v$ . O grau máximo do grafo, denotado por  $\Delta(G)$ , é o valor do maior grau dentre todos os vértices de  $G$ . De maneira similar, o grau mínimo, denotado por  $\delta(G)$ , é definido como o valor do menor grau de  $G$ . Um grafo completo é aquele no qual cada vértice é vizinho de todos os demais, ou seja, o grau de todos os vértices é exatamente igual a  $n - 1$ . A quantidade de arestas é dada então por  $m = \frac{n(n-1)}{2}$ . Um exemplo de grafo completo é dado na Figura 2.7.

Um problema clássico em grafos é a coloração de vértices, onde tem-se a seguinte situação [17]: dado um grafo não direcionado  $G = (V, E)$ , onde  $V$  é o conjunto de vértices e  $E$  é o conjunto de arestas, deseja-se colorir cada vértice de forma que vértices adjacentes tenham cores diferentes. Uma  $k$ -coloração do grafo é uma coloração no qual são usadas, no máximo,  $k$  cores. O objetivo é minimizar a quantidade de cores usadas, que equivale a determinar o número cromático do grafo (ou seja, o menor valor possível de  $k$ , denotado por  $\chi_G$ ), um problema NP-difícil ([40]). Uma coloração para o grafo da Figura 2.6 é dada na Figura 2.8.

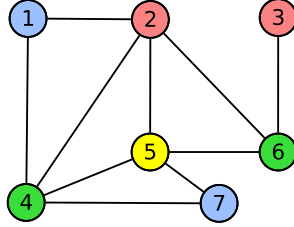
Existem inúmeras variações do problema. Em um dos primeiros trabalhos envolvendo alocação de canais, foram propostos modelos para o problema levando em conta diferentes tipos de distâncias [43], que foram modelados também como colorações em grafos, enunciados a seguir com sua relação aos grafos. O primeiro deles, denominado de **Pro-**



**Figura 2.6:** Representação gráfica de um grafo não-direcionado planar com 7 vértices e 10 arestas.



**Figura 2.7:** Exemplo de grafo completo com 4 vértices.



**Figura 2.8:** Coloração dos vértices da Figura 2.6.

**Problema de Alocação Generalizada de Canais de Mínima Extensão** (*General Minimum Span Channel Assignment Problem - F\*D-CAP*) envolve em um conjunto  $V$  de pontos no plano e um conjunto  $R = (t(r), d(r))$  de restrições. Cada restrição é um par ordenado, onde  $0 = t(0) \subset t(1) \subset \dots \subset t(m)$  e  $d(0) > d(1) > \dots > d(m)$ , e a  $k$ -ésima restrição é o par  $(t(r), d(r))$  tal que  $k \geq 0$  e  $k \in t(r)$  mas  $k \notin t(r-1)$ . A distância a ser respeitada entre canais será  $d(r)$ , e a separação *co-site* é a 0-ésima restrição. Logo, deve-se encontrar um mapeamento  $x : V \rightarrow \mathbb{Z}^+$  onde  $|x_i - x_j| \notin t(r)$  quando  $i \neq j$  e a distância entre  $i$  e  $j$  for menor ou igual a  $d(r)$  para  $r = 0, 1, \dots, m$ .

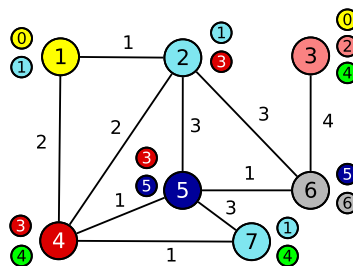
Um outro modelo proposto no mesmo trabalho é denominado de **Problema de Alocação Restrita de Canais de Mínima Extensão** (*Frequency Constrained Minimum Span Assignment Problem - F-CAP*), onde apenas a separação entre canais é considerada. Nele,  $V$  passa a ser um conjunto de rótulos que representam as ERBs, e é fornecida uma matriz de restrições  $T$ , onde cada elemento  $T(i, j)$  consiste em um conjunto de inteiros positivos tal que  $i, j \in V$ ,  $T(i, j) = T(j, i)$  e  $T(i, i) = \emptyset$ . O objetivo consiste em encontrar um mapeamento  $x : V \rightarrow \mathbb{Z}^+$  tal que  $|x_i - x_j| \notin T(i, j)$  cujo maior  $x_i$  para todo  $i \in V$  seja o menor possível.

Uma característica interessante nos dois problemas acima é que ambos são equivalentes, apesar de utilizarem restrições diferentes. Para transformar uma instância do F\*D-CAP (com o conjunto  $V$  de pontos e  $R = (t(r), d(r))$  de restrições) em F-CAP (com um conjunto  $V'$  de rótulos e matriz de restrições  $T'$ ), deve-se, primeiramente, considerar os pontos no plano como rótulos, ou seja,  $V' = V$ . A matriz  $T'$  é construída da seguinte forma. Seja  $r = \min\{r' \in [0, m] \mid D(i, j) \leq d(r')\}$ . Tem-se então que  $T'(i, j) = t(r)$  e

$$T(i, i) = \emptyset.$$

O problema F-CAP é modelado como uma variação de coloração em grafos. No **Problema de Coloração Generalizada em Grafos** (*Generalized Graph Coloring Problem - GCP*), tem-se um grafo não-direcionado  $G = (V, E)$  e, para cada aresta  $(i, j) \in E$ , um conjunto  $T(i, j)$  de valores (chamado de conjunto proibido) que determinam uma restrição de aresta (caso não haja uma aresta entre dois vértices  $i$  e  $j$ , tem-se que  $T(i, j) = \emptyset$ ). Uma coloração válida será determinada por um mapeamento  $x : V \rightarrow \mathbb{Z}^+$  onde  $|x_i - x_j| \notin T(i, j)$ . O GCP também é conhecido na literatura como **set T-coloring** devido à notação usada nos conjuntos proibidos. Quando todos os conjuntos  $T(i, j)$  para todas as arestas  $(i, j) \in E$  são iguais, tem-se o problema de **T-coloring**. Caso todos os conjuntos proibidos consistam de números inteiros consecutivos partindo de 1, pode-se simplesmente ponderar a aresta (valor  $d_{ij} = \max\{T(i, j)\}$ ) e exigir que  $|x_i - x_j| \geq d_{ij}$ , onde o problema é conhecido como **channel assignment coloring**.

Várias outras variações são amplamente exploradas na literatura. Alguns vértices podem necessitar de mais de uma cor, sendo que os mesmos possuem um peso  $w_i$  que equivale à quantidade de cores necessárias do vértice  $i$ . Esta variação é conhecida como **multicoloração** ([61]). É interessante notar que um problema de multicoloração pode ser transformado em um problema de  $k$ -coloração replicando-se o vértice  $i$  no grafo  $w_i$  vezes. Os vértices também podem ter restrições de *quais* cores os mesmos podem assumir. Na **coloração com listas**, cada vértice possui uma lista de cores possíveis que podem ser usadas. Uma simplificação desta ideia é considerar cada vértice  $i$  tendo apenas um limite superior  $\mu_i$  para a sua cor, ou seja, o vértice pode usar qualquer cor em  $[0, \mu_i]$ . Esta variação é conhecida por  **$\mu$ -coloração** ([18]). Os vértices também podem possuir limites inferiores, de forma que cada vértice  $i$  possa usar apenas cores no intervalo  $[\gamma_i, \mu_i]$ , onde  $\gamma_i$  é o limite inferior. Esta variação é conhecida como  **$(\gamma, \mu)$ -coloração** ([19]).



**Figura 2.9:** Coloração dos vértices da Figura 2.6 com modelos mais gerais.

## 2.3 Escalonamento de tarefas

Um problema de escalonamento envolve, em sua forma mais comum, um conjunto de  $n$  tarefas, chamado  $J$  (ou seja, temos  $|J| = n$ ) que devem ser executadas em um conjunto de  $m$  processadores [21, 75]. Tanto as tarefas quanto os processadores podem ter diferentes variações que caracterizam os inúmeros problemas de escalonamento conhecidos [79, 71, 27, 86]. As tarefas podem ter:

- **Tempo de processamento** ( $p_j$ ): Indica o tempo que a tarefa deve permanecer sendo executada nas máquinas (geralmente, uma) para que a mesma seja cumprida;
- **Data de Disponibilidade** (*release date*) ( $r_j$ ): É o menor instante no qual a tarefa pode começar a ser executada;
- **Prazo Recomendado de Término** (*due date*) ( $d_j$ ): Tempo máximo recomendado de completude da tarefa. Se a mesma terminar depois deste prazo, haverá uma penalidade;
- **Prazo Máximo de Término** (*deadline*) ( $\bar{d}_j$ ): Prazo máximo para a completude da tarefa (não pode ser ultrapassado);
- **Peso** ( $w_j$ ): Um valor que pode ser aplicado como uma penalização da tarefa em cima de algum critério (como, por exemplo, violação de *due date*).

Para facilitar a referência a um tipo específico de problema, foi introduzida a **notação de três campos** [42], cujo formato é  $\alpha \mid \beta \mid \gamma$ , onde:

- $\alpha$  é o ambiente de processamento, sendo alguns valores possíveis para este campo:
  - 1: Há apenas um processador disponível para todas as tarefas;
  - $P$ : Existem  $m$  processadores paralelos idênticos disponíveis, onde  $m$  é um parâmetro do problema;
  - $Q$ : Há  $m$  processadores paralelos uniformes de velocidades diferentes, sendo  $q_i$  a velocidade do  $i$ -ésimo processador. Devido às velocidades diferentes, tem-se que o tempo que a  $j$ -ésima tarefa passa no  $i$ -ésimo processador (considerando que  $j$  será executada por completo em  $i$ ) é  $\frac{p_j}{q_i}$

- $R$ : Existem  $m$  processadores paralelos não relacionados, sendo  $q_{ij}$  a velocidade que o  $i$ -ésimo processador pode executar a  $j$ -ésima tarefa, o que fornece um tempo de execução de  $\frac{p_j}{q_{ij}}$
- $\beta$  é um conjunto de características das tarefas e restrições, sendo algumas as listadas a seguir:
  - **prec**: Existe uma relação de precedência entre algumas tarefas, ou seja, certas tarefas devem ser completadas antes das tarefas que dependam destas;
  - **r<sub>j</sub>**: Se este valor estiver presente, as tarefas possuem datas de disponibilidade;
  - **p<sub>j</sub>**: Os tempos de processamentos das tarefas são restritos, sendo que  $p_j = p$  indica que todas as tarefas tem tempo de processamento  $p$ ;
  - **d<sub>j</sub>**: As tarefas possuem *due dates*;
  - **$\bar{d}_j$** : As tarefas possuem *deadlines*.
- $\gamma$  é a função objetivo a ser otimizada. Alguns valores são usados por tais funções nesta classe de problemas, tais como:
  - **Tempo Total de Completude (total completion time -  $C_j$ )**: O instante no qual a tarefa terminou sua execução, sendo dado por  $C_j = s_j + p_j$ , onde  $s_j$  é o instante de início da tarefa;
  - **Latência (lateness -  $L_j$ )**: A distância temporal entre o instante em que a tarefa foi completada e o seu *due date*. Assim,  $L_j = C_j - d_j$ ;
  - **Atraso (tardiness -  $T_j$ )**: Quanto tempo a tarefa foi executada após o seu *due date*. Temos então  $T_j = \max\{0, L_j\}$ ;

Tem-se então algumas das principais funções a serem minimizadas:

- **Makespan -  $C_{max}$** : Minimização do maior tempo de completude dentre todas as tarefas.  $C_{max}$ , logo, é a extensão do tempo usado para executar todas as tarefas;
- **Latência máxima -  $L_{max}$** : Minimização da maior latência dentre todas as tarefas. Deseja-se, então, terminar as tarefas o mais cedo possível;



- **Tempo de Completude Ponderado** -  $\sum w_j C_j$ : Minimização da soma dos tempos de completude das tarefas, sendo cada tempo multiplicado pelo peso  $w_j$  da tarefa (se  $w_j = 1$ , temos a versão não ponderada);
- **Atraso Ponderado** -  $\sum w_j T_j$ : Minimização da soma dos atrasos das tarefas com os pesos  $w_j$  das mesmas aplicados (novamente, (se  $w_j = 1$ , temos a versão não ponderada).

Um tipo de problema de escalonamento mais geral é o **Resource Constrained Project Scheduling Problem** (RCPSP) [11, 28]. Neste problema, tem-se um conjunto de tarefas a serem executadas e um conjunto de  $b$  recursos finitos ( $\mathfrak{R} = \{\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_b\}$ ) que são usados em quantidades distintas, sendo que, a cada instante, só podem ser usadas no máximo  $r_x$  unidades do recurso  $\mathfrak{R}_x$ . Existem precedências nas tarefas, e entre duas tarefas  $i$  e  $j$ , existe uma quantidade de tempo mínima que deve ser respeitada entre as mesmas, denominada de **time lag** e representada por  $\ell_{ij}$ . No RCPSP, conforme [11], é possível a existência de ciclos no grafo de precedências, de forma que os *lags*, dependendo da ordem das tarefas, não sejam simétricos.

## 2.4 Geometria de distâncias

A geometria de distâncias (GD) consiste em determinar localizações de pontos no espaço respeitando distâncias entre pares conhecidas. O problema fundamental de GD pode ser formalizado como se segue [57].

**Problema de Geometria de Distâncias** (*Distance Geometry Problem - DGP*): dado um inteiro  $K > 0$  e um grafo não-direcionado simples  $G = (V, E)$ , cujas arestas são ponderadas por uma função  $d : E \rightarrow \mathbb{R}^+$ , existe uma função  $x : V \rightarrow \mathbb{R}^n$  tal que  $\forall (i, j) \in E, \|x(i) - x(j)\| = d((i, j))$ ?

Tal problema pode ser visto, então, como a projeção de um grafo no espaço euclidiano  $n$ -dimensional. O mesmo é NP-completo para  $n = 1$  [83] e fortemente NP-difícil para  $n > 1$ . Uma tentativa de prova de que o problema é NP para  $n = 2$  (o que, por consequência, provaria sua NP-completude para tal caso) foi realizada, mas sem sucesso [14].

Existem várias aplicações conhecidas deste problema, todas envolvendo algum tipo de distância, como localização de sensores sem fio e trilateração [68], posicionamento global

(GPS), reconhecimento de padrões, estática, dentre outras. Uma das principais, sendo amplamente estudada na literatura recente, é a determinação da estrutura tridimensional de proteínas, onde o espaço no qual o grafo de entrada deve ser projetado é o  $\mathbb{R}^3$  [53, 57].

Os principais métodos para problemas de geometria de distâncias, em especial em sua aplicação na determinação da estrutura de proteínas [81], incluem um algoritmo de tempo linear que pode ser usado quando as distâncias entre todos os pares de vértices são conhecidas (ou seja, o grafo de entrada é completo) [33] e o algoritmo *branch-and-prune*, usado quando nem todas as distâncias são conhecidas, onde os vértices são ordenados em cliques de quatro elementos e, a cada passo, a posição de um vértice é determinada de acordo com as posições determinadas anteriormente, eliminando soluções infactíveis [53]

## 2.5 Notas do capítulo

Este capítulo apresentou brevemente alguns conceitos teóricos utilizados durante a pesquisa da dissertação. No entanto, existem muitos outros conceitos, por exemplo, outras classes de problemas de otimização, técnicas para resolução dos mesmos e avaliação da eficiência dos algoritmos. Mais sobre este assunto pode ser visto em [69, 54]. Uma obra sobre programação matemática [20] e livros específicos sobre meta-heurísticas [58, 88] também estão disponíveis na literatura. Para programação por restrições, uma boa fonte introdutória é o manual das ferramentas *or-tools* disponibilizadas pelo Google [92].

Em teoria dos grafos, há classes de grafos nas quais os diferentes problemas de coloração admitem algoritmos polinomiais, além de outros problemas nos quais o estudo das propriedades do grafo é crucial. Boas referências do assunto são [87, 17].

Já para a classe de problemas de escalonamento, há uma grande quantidade de problemas com impacto tanto prático quanto teórico, e novos trabalhos sobre os mesmos continuam a ser publicados. Referências clássicas para este assunto incluem [21, 73, 56], e mais sobre a complexidade de alguns problemas dessa classe pode ser vista em [22].

Por fim, existem muitas outras aplicações de geometria de distâncias. Um conjunto de trabalhos da área pode ser visto em [64]. Um *survey* da teoria e aplicações de geometria de distâncias é fornecido em [57]. Há também uma obra que explora relações entre otimização convexa e geometria de distâncias euclidianas, disponível abertamente [29].

## Capítulo 3

# Alocação de canais em redes móveis sem fio

Neste capítulo, será abordada a arquitetura das redes celulares e as de rádios cognitivos utilizadas amplamente para comunicação móvel, as quais foram exploradas nos problemas pesquisados. Além dos conceitos das redes, são definidos alguns modelos matemáticos e abordagens propostas na literatura para alocação de canais em tais ambientes.

### 3.1 Redes móveis celulares

Nos últimos anos, o número de usuários de sistemas móveis tem aumentado de maneira bastante expressiva, tornando necessária a expansão das redes de serviços de conexão sem fio. O aumento do poder computacional dos dispositivos móveis, como celulares e *tablets*, além da maior concorrência no mercado dos mesmos, o que leva a preços mais baixos, tem levado cada vez mais pessoas a adquirirem estes equipamentos e, com isso, a contratarem planos de dados para conexão com a Internet.

Estima-se que até o fim de 2013 havia aproximadamente 4,5 bilhões de assinantes de telefonia móvel celular no mundo, o que corresponde a aproximadamente 62% da população mundial atual. No entanto, como muitos usuários possuem mais de uma linha (por exemplo, em diferentes operadoras), o total de assinaturas é ainda maior, estimado em 6,7 bilhões. Tal diferença é notável, uma vez que a presença global do serviço chega a 92%, mas em alguns continentes, a presença do serviço passa dos 100%, o que significa mais de uma assinatura por habitante em tais lugares. A Tabela 3.1 mostra a presença do

**Tabela 3.1:** Presença do serviço móvel celular em diferentes partes do mundo. Adaptado de Ericsson, 2014 [38].

<b>Região</b>	<b>Presença do serviço</b>
Europa Central e Oriental	143%
Europa Ocidental	127%
América Latina	115%
Oriente Médio	104%
Ásia <sup>1</sup> e Pacífico	105%
América do Norte	102%
China	90%
África	72%
Índia	59%
<b>Global</b>	<b>92%</b>

**Tabela 3.2:** Estatísticas de usuários de redes celulares no Brasil. Adaptado de Teleco, 2014 [90].

	<b>Janeiro/2013</b>	<b>Dezembro/2013</b>	<b>Janeiro/2014</b>
<b>Assinaturas</b>	262.257.578	271.099.799	272.353.241
<b>Pré-pago</b>	80,37%	78,05%	77,95%
<b>Presença</b>	132,93%	136,45%	137%
<b>Crescimento mensal</b>	449.675	580.924	1.253.442
	0,2%	0,2%	0,5%
<b>Crescimento anual</b>	17.078.538	9.291.896	10.095.663
	7,0%	3,5%	3,8%

serviço em diferentes partes do mundo [38].

A situação específica do Brasil não é diferente. Estima-se que o número total de assinaturas no país, em janeiro de 2014, tenha chegado a 272,4 milhões, com uma presença do serviço de 137%. Um fato interessante a ser visto é que, com o aumento do uso e a popularização dos serviços, a quantidade de usuários pós-pagos tem aumentado (apesar de ainda ser minoria), uma vez que o serviço costuma ter menos restrições e menor preço para quem faz grande utilização do mesmo. A Tabela 3.2 resume as estatísticas do uso de celulares no Brasil.

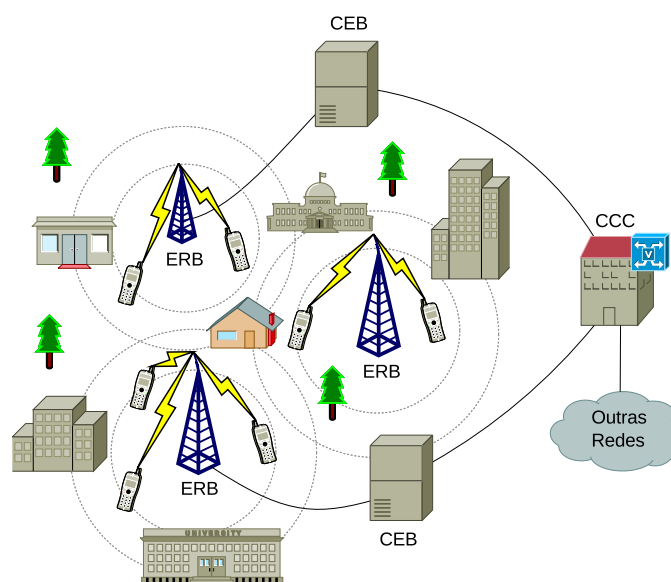
### 3.1.1 Estrutura de uma rede celular

Em uma rede móvel celular, existe um conjunto de estações de rádio-base (ERBs), que consistem em dispositivos que realizam o interfaceamento dos usuários móveis com o

<sup>1</sup>Exceto Índia e China.



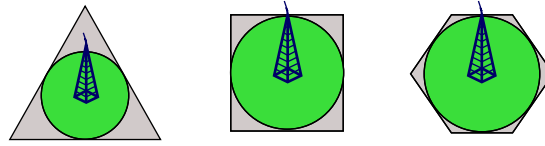
**Figura 3.1:** Topo de uma torre de rádio, onde os painéis funcionam como antenas (esquerda - fonte: <http://www.radioinnovation.se/en/documents/>), e um dispositivo de ERB modelo Ericsson RBS6000 (direita - fonte: <http://comm.ae/ericsson-radio-base-station-milestone-achieved/>).



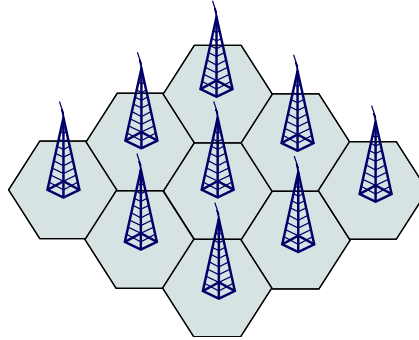
**Figura 3.2:** Exemplo de rede de telefonia móvel celular.

restante da rede, sendo ligadas à antenas com as quais os usuários se comunicam. A Figura As ERBs, por sua vez, são ligadas aos controladores de estações base (CEB) [10, 49]. Estes controladores são responsáveis pela comunicação com as centrais de comutação e controle (CCC), que possuem conexões com as redes terrestres de telefonia fixa. Cada ERB define uma célula, que consiste em uma região geográfica coberta pela antena. Os sinais propagados pelo equipamento seguem um padrão omnidirecional, ou seja, seguem em todas as direções. A Figura 3.2 exemplifica esta arquitetura.

Pra representar a área coberta por uma ERB através de um polígono regular, são usadas células hexagonais, que fornecem uma boa aproximação da região atendida. As Figu-



**Figura 3.3:** Tipos de padrões de células.

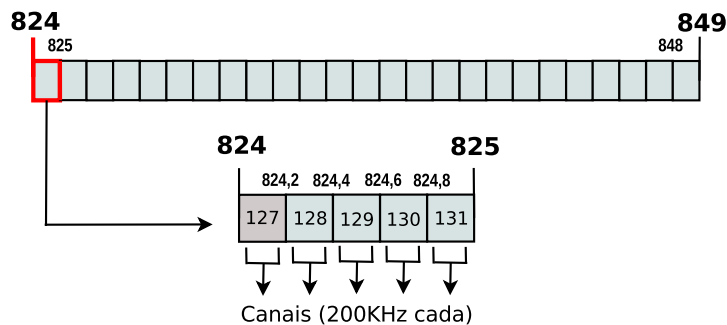


**Figura 3.4:** Conjunto de células hexagonais.

ras 3.4 mostram como o padrão hexagonal se adequa a este cenário e um exemplo de rede celular, respectivamente.

Cada chamada na rede está associada a uma célula de acordo com a sua localização - a chamada efetuará a comunicação com a ERB que cobre a área na qual o dispositivo móvel se encontrar. Tal comunicação é efetuada por meio de um canal, que consiste em uma porção discretizada do espectro eletromagnético. Um exemplo deste conceito pode ser visto na Figura 3.5, que mostra a divisão do espectro na faixa GSM-850.

Além da escassez de canais, os mesmos não podem ser atribuídos às chamadas de qualquer modo, pois existe o problema de interferência entre dispositivos localizados próximos entre si e/ou utilizando canais vizinhos no espectro eletromagnético. Existem três principais tipos de interferência:



**Figura 3.5:** Divisão da faixa GSM-850 do espectro eletromagnético em 125 canais. Os canais nas bordas da faixa não são utilizados para diminuir interferências com outras faixas.

- **Co-canal:** interferência entre chamadas de duas ERBs diferentes utilizando o mesmo canal;
- **Canal adjacente:** interferência entre chamadas de duas ERBs diferentes utilizando canais adjacentes;
- **Co-site:** interferência entre chamadas na mesma célula.

O acesso ao espectro eletromagnético é feito de forma compartilhada, de forma que é necessário utilizar um esquema para controle do fluxo de dados de cada chamada. Os três principais protocolos de acesso ao meio são [52]:

- **Acesso múltiplo por divisão de frequência** (*frequency division multiplexing access* - FDMA): neste protocolo, a largura de banda é dividida em porções menores que são reservadas para as estações da rede. Cada estação é associada permanentemente à porção correspondente, o que permite um fluxo constante de dados para a estação sem necessidade de empacotamento. Para evitar interferências entre porções próximas dentro de uma mesma banda de frequências, são utilizadas pequenas bandas de guarda.
- **Acesso múltiplo por divisão temporal** (*time division multiplexing access* - TDMA): nesta técnica, ao invés de dividir a banda, a estação pode usar toda a largura de banda disponível, no entanto, para permitir o acesso múltiplo, cada uma delas possui uma fatia de tempo na qual a estação pode efetuar suas transferências de dados. De maneira similar ao FDMA, utiliza-se tempos de guarda para garantir a sincronização efetiva das fatias de tempo das estações.
- **Acesso múltiplo por divisão de código** (*code division multiplexing access* - CDMA): neste cenário, toda a largura de banda está disponível para todas as estações móveis. Para permitir o acesso simultâneo, é atribuída uma sequência de bits para cada estação, de forma que quando receptor e transmissor utilizam a mesma sequência, é possível recuperar os dados trafegados, sendo que dados codificados com uma sequência diferente de bits são considerados ruídos.

A Figura 3.6 exemplifica tais protocolos de acesso ao meio em relação à ocupação nos domínios de tempo e frequência.

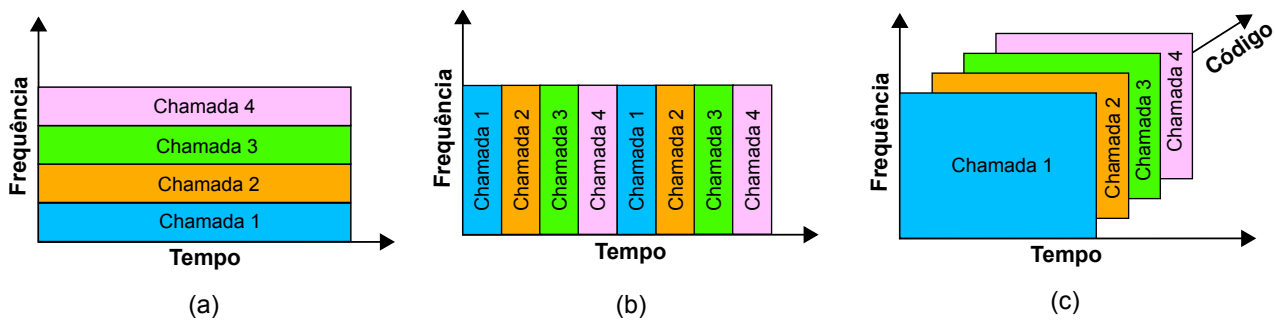


Figura 3.6: Exemplos do uso de protocolos de acesso ao meio: (a) FDMA, (b) FDMA e (c) CDMA.

### 3.1.1.1 Tecnologias de telefonia celular

Historicamente, as redes celulares foram evoluindo de acordo com o uso e as necessidades dos usuários. Atualmente, há quatro gerações de telefonia móvel celular desenvolvidas, descritas a seguir [52, 89, 39].

- **Primeira geração (1G): telefonia móvel analógica**

O primeiro sistema de comunicação móvel com voz surgiu em 1946, o qual utilizava apenas um canal para envio e recebimento de dados e havia o uso de um único transmissor. Posteriormente, uma versão melhorada do sistema foi desenvolvida, chamada de Sistema Melhorado de Telefonia Móvel (*Improved Mobile Telephony System - IMTS*), onde os canais de envio e recebimento eram separados. No entanto, a quantidade de canais disponíveis era bastante limitada (23 canais entre 150 MHz e 450 MHz), o que impedia o uso por muitos usuários.

Para sanar o problema, surgiu o Sistema Avançado de Telefonia Móvel (*Advanced Mobile Phone System - AMPS*), cuja implementação inicial ocorreu nos Estados Unidos. A divisão dos canais era feita utilizando o FDMA, sendo que eram usadas duas faixas de frequência: entre 824 e 849 MHz para a comunicação do dispositivo móvel à estação, e de 869 a 894 MHz para o inverso (da estação para o dispositivo móvel). Cada canal possuía largura de banda de 30 KHz, o que fornecia 832 canais em cada faixa de frequência, ou seja, 832 canais *duplex*. No entanto, o número de canais realmente disponíveis para voz na célula era de aproximadamente 56, o que já era bem mais que a tecnologia anterior.

- **Segunda geração (2G): telefonia móvel digital**

Ao contrário da primeira geração, todos os sinais passam a ser digitais, o que permite uma qualidade maior para as comunicações de voz, além de maior capacidade de ser-



viço em cada célula e mais segurança. Um dos primeiros sistemas 2G consiste em uma modificação do AMPS para utilização de modulação digital, o D-AMPS, também conhecido como Padrão Provisório TDMA 136 (*TDMA Interim Standard 136- IS-136*), que possui compatibilidade retrógrada com o AMPS, além de utilizar as mesmas bandas de frequência. A voz é digitalizada a uma taxa de 7,95 kbps.

Posteriormente, surgiu o Sistema Móvel para Comunicações Móveis (*Global System for Mobile Communications - GSM*), desenvolvido na Europa e amplamente usado em vários países. O sistema simula um modem entre as duas pontas da comunicação, usando, na prática, comutação por circuitos. O acesso múltiplo é gerenciado por uma combinação dos esquemas FDMA e TDMA: cada banda de frequências é dividida em 125 canais cuja largura de banda é de 200 KHz (como mostrado anteriormente na Figura 3.5), porém, os 100 KHz iniciais e os 100 KHz finais da faixa são usados como bandas de guarda, o que totaliza 124 canais úteis. Cada um desses canais suporta até oito conexões diferentes por meio da divisão do tempo de uso do canal, e a taxa de codificação da voz digitalizada é de 13 kbps.

Outro sistema de segunda geração é o Padrão Provisório CDMA 95 (*CDMA Interim Standard 95 - IS-95*). Ao contrário dos outros sistemas, o acesso à interface é gerenciado por códigos. Cada banda deste padrão possui 20 canais com largura de 1,228 MHz cada e digitalização de voz a 9,6 Kbps. Inicialmente, o padrão não foi visto como aceitável, mas a empresa Qualcomm conseguiu transformá-lo em uma opção viável na década de 1980, levando a sua ampla adoção em países da América do Norte.

Todos os sistemas de telefonia móvel digital apresentados até o momento foram desenvolvidos para transmissão de voz. Com o avanço da tecnologia de dispositivos móveis e a explosão da Internet, surgiu uma grande demanda para transmissão de dados usando as redes celulares, o que exige uma taxa de transmissão bem maior para um uso aceitável. Padrões provisórios baseados nas tecnologias de segunda geração foram desenvolvidos para sanar esta deficiência, sendo conhecidos como sistemas 2,5G.

Uma alternativa 2,5G é o Serviço Geral de Rádio por Pacote (*General Packet Radio Service*), uma evolução do sistema GSM onde é utilizada comutação de pacotes, o que aumenta a taxa de transmissão de dados para 40 kbps a 60 kbps. Também baseado no GSM é o sistema como Melhores Taxas de Dados para Evolução Global (*Enhanced Data Rates for Global Evolution - EDGE*), apelidado de 2,75G, que aumenta a veloci-

dade do GSM/GPRS para 384 kbps nominais. O CDMA também possui uma versão com taxas de transferência maiores, de até 144,4 kbps.

- **Terceira geração (3G): transmissão de dados diversos**

As tecnologias móveis para transmissão de dados, além de voz, foram consolidadas a partir desta geração. As características obrigatórias dos sistemas 3G são:

- 144 kbps em um veículo em movimento.
- 384 kbps em uso estacionário ou em mobilidade pedestre.
- 2 Mbps em ambientes internos.
- Suporte a comutação de pacotes e de circuitos.
- Bandas na faixa de 2 GHz.
- Larguras de banda de 2 MHz.
- Acesso à Internet.

A ideia original era desenvolver um único sistema de terceira geração, que seria denominado de Comunicação Móvel para Internet 2000 (*Internet Mobile Communication 2000 - IMT-2000*), porém, duas tecnologias se tornaram difundidas nas redes celulares pelo mundo.

A primeira proposta foi criada pela Ericsson e denominada de CDMA de Banda Larga (*Wideband CDMA - W-CDMA*), posteriormente também conhecida como Sistema Universal de Telecomunicações Móveis (*Universal Mobile Telecommunications System - UMTS*). As larguras de banda são de 5 MHz e o acesso ao meio é realizado por meio de códigos, no entanto, a tecnologia interage com as redes GSM existentes, apesar de não possuir compatibilidade retrógrada direta com as mesmas. A velocidade máxima nominal chega a 2 Mbps, porém, é típico que o máximo chegue a somente 384 kbps, o que é bastante aceitável e atende às normas da terceira geração. Tal sistema é amplamente usado na Europa.

O UMTS passou por evoluções posteriores de forma a tornar ainda melhor a taxa de transmissão de dados, sendo apelidadas de 3,5G. A tecnologia de Acesso de Pacotes em Alta Velocidade (*High Speed Packet Access - HSPA*) possui padrões específicos para *download* (HSDPA) e *upload* (HSUPA), sendo que as taxas nominais chegam a, respectivamente, 42,3 Mbps e 5,76Mbps, utilizando os mesmos canais de largura de 5

MHz. O HSPA foi posteriormente incrementado (HSPA+) de forma a suportar taxas de *download* de 168 Mbps e de *upload* de 22 Mbps por meio do uso dos mesmos canais e de múltiplas antenas (*Multiple Input, Multiple Output* - MIMO).

Outra proposta 3G é a CDMA-2000, que possui compatibilidade retrógrada com o IS-95. Semelhante ao UMTS, também usa códigos para acesso ao meio e larguras de banda de 5 MHz, porém, não há possibilidade de interfaceamento com redes GSM e nem D-AMPS. Esta tecnologia é mais empregada em países da América do Norte e fornece velocidades de até 144 Kbps, motivo pelo qual ela não é unanimemente considerada de terceira geração. A empresa Qualcomm melhorou então a tecnologia de forma a suportar melhor transmissões de dados (*Data Only* - CDMA-2000 1xEV-DO) ou mistas de dados e voz (*Data and Voice* - CDMA-2000 1xEV-DV), com taxas de *download* de até 3,1 Mbps e de *upload* de até 1,8 Mbps.

- **Quarta geração (4G): maiores velocidades**

A quarta geração das redes móveis celulares já é usada amplamente nos países desenvolvidos, sendo que sua implementação ainda está sendo realizada em outros, como o Brasil. Algumas características desejadas para as redes 4G incluem:

- 100 Mbps em grande mobilidade.
- 1 Gbps em alta mobilidade.
- Utilização de comutação de pacotes utilizando IP.
- Largura de banda dos canais escalável, entre 5 e 20 MHz.
- Alta qualidade para serviços multimídia.

O principal padrão 4G é conhecido como Evolução de Longo Prazo (*Long Term Evolution* - LTE), o qual oferece velocidades nominais de 300 Mbps para *download* e 75,4 Mbps para *upload* com uso de canais de 20 MHz e quatro antenas para cada sentido da comunicação (4x4 MIMO). O LTE utiliza esquemas baseados em FDMA para o acesso ao meio e suporta o uso de canais com larguras de banda entre 1,4 MHz e 20 MHz.

Uma desvantagem do LTE diz respeito à sua regulamentação. Existem diversas faixas de frequência utilizáveis para o sistema, ocorrendo diferenças no uso de continente para continente. Assim, alguns aparelhos móveis com tecnologia 4G podem não funcionar fora do país onde foi adquirido.

- **Quinta geração (5G): o futuro**

Apesar de a tecnologia 4G ainda não ser totalmente implementada mundialmente, já há esforços para o desenvolvimento da quinta geração das redes móveis (5G), sendo que as implementações experimentais são esperadas para 2017, com comercialização do serviço a partir de 2020.

Atualmente, a Coreia do Sul está fazendo grandes investimentos para o desenvolvimento do 5G, cuja velocidade poderá chegar a mil vezes maior que o 4G LTE, permitindo o uso de conteúdo multimídia de alta definição com total mobilidade [24].

A empresa Samsung já vem realizando testes com possíveis sinais 5G, tendo conseguido um sinal de 1 Gbps a uma distância de 2 km. Para atingir a alta velocidade, foram usadas 64 antenas e uma troca rápida de transmissores e receptores de forma a obter o sinal de maior qualidade possível.

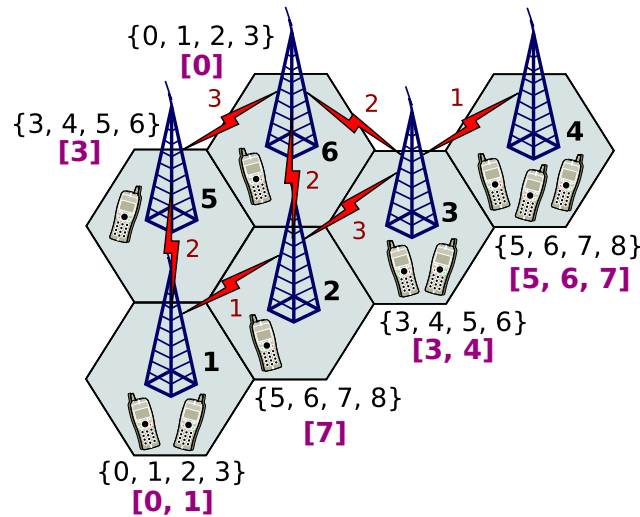
Outra empresa, a Huawei, por sua vez, visualiza redes de quinta geração com velocidades de até 10 Gbps, com implementação e comercialização entre 2020 e 2030. Tal velocidade se adequaria também às novas tecnologias de ultra alta definição (4K), que exigem uma largura de banda muito grande para transferência de conteúdo com qualidade [45].

### 3.1.2 Problemas de alocação de canais em redes celulares

Os problemas de alocação de canais em redes celulares (*channel assignment problems* - CAPs) podem ser enunciados como se segue ([49, 10]). Existe um conjunto de  $n$  antenas (ERBs), onde cada ERB  $i$  tem uma demanda de  $q_i$  canais, que corresponde à quantidade de chamadas daquela estação. Cada chamada necessita de um canal para sua realização. O  $k$ -ésimo canal (ou seja, o canal da  $k$ -ésima chamada) alocado para a  $i$ -ésima ERB é denotado por  $f_{ki}$ . Com isto, tem-se que para todo par  $(i, j)$  de ERBs que possam interferir entre si, existe um valor  $d_{ij}$  (a distância de separação) para o qual a condição:

$$|f_{ki} - f_{mj}| \geq d_{ij}$$

Deve ser respeitada. Quando as ERBs  $i$  e  $j$  não interferem entre si, tem-se que  $d_{ij} = 0$ . Tais valores formam uma **matriz de interferência**  $D_{n \times n} = [d_{ij}]$ . As estações não podem assumir qualquer canal dentre os disponíveis: para cada ERB  $i$ , existe um conjunto



**Figura 3.7:** Exemplo de alocação de canais em uma rede com 6 células. O número abaixo de cada estação é o seu identificador e o conjunto de números entre chaves é o domínio de frequência da estação. Os raios entre estações indicam possíveis interferências e o valor próximo às mesmas indica a distância mínima de separação entre as ERBs. Os pequenos aparelhos celulares indicam a quantidade de chamadas, sendo que os valores entre colchetes são os canais alocados para as mesmas.

denominado de **domínio de frequência** ( $F_i$ ) que contém os possíveis canais que as chamadas de  $i$  podem receber. Estes conjuntos representam as restrições de estações que, por estarem em localizações com alto uso de certa parte do espectro eletromagnético, devem evitar o uso desta parte para não haver interferência. O conjunto de todos os canais disponíveis para a rede, denotado por  $F$ , é a união de todos os domínios de frequência ( $F = F_1 \cup F_2 \cup \dots \cup F_n$ ). Um exemplo é dado na figura 3.7, juntamente com uma alocação factível para a instância.

Para efetuar a atribuição dos canais disponíveis para as chamadas existentes, existem três esquemas. O primeiro, chamado de alocação **fixa** de canais (*fixed channel assignment* - FCA) atribui permanentemente os canais demandados por uma célula para sua ERB correspondente, de forma que apenas aquele número de chamadas poderá ser atendido em um mesmo instante. No esquema **dinâmico** (*dynamic channel assignment* - DCA), inicialmente, todos os canais estão disponíveis para todas as estações e são alocados de acordo com as chamadas que vão surgindo (a instância muda com o tempo). O esquema **híbrido** (*hybrid channel assignment* - HCA) combina os esquemas FCA e DCA de forma que cada ERB possui alguns canais pré-alocados permanentemente e quando a mesma tem mais uma chamada, um novo canal é alocado dinamicamente.

### 3.1.2.1 Critérios de otimização

Existem três principais critérios de otimização que podem ser considerados ao alocar canais em uma rede celular: a minimização da ordem, a minimização da extensão de canais usada e a minimização da interferência, que serão mostrados a seguir.

- **Minimização da ordem**

Neste problema, dadas as  $n$  ERBs, o vetor de demandas  $c$  e o domínio da frequência  $F_i$  de cada ERB  $i$  e a matriz de distâncias  $D$ , deseja-se alocar, para cada antena  $i$ ,  $q_i$  canais de forma que as distâncias de frequências sejam satisfeitas e que a **quantidade de canais usados** (conhecida como **ordem**) seja a menor possível.

Uma formulação de programação linear inteira para este problema pode ser dada como segue. Tem-se os seguintes conjuntos de variáveis:

$$x_{if} = \begin{cases} 1 & \text{se o canal } f \in F_i \text{ for atribuído à estação } i; \\ 0 & \text{caso contrário} \end{cases}$$

$$y_f = \begin{cases} 1 & \text{se o canal } f \in F \text{ foi alocado a alguma estação;} \\ 0 & \text{caso contrário} \end{cases}$$

O modelo então é dado por:

$$\text{Minimizar } \sum_{f \in F} y_f \quad (3.1)$$

$$\text{Sujeito a } \sum_{f \in F_i} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \quad (3.2)$$

$$x_{if} + x_{jg} \leq 1 \quad (\forall i, j = 1, 2, \dots, n; f \in F_i; g \in F_j : |f - g| < d_{ij}) \quad (3.3)$$

$$x_{if} \leq y_f \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.4)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.5)$$

$$y_f \in \{0, 1\} \quad (\forall f \in F) \quad (3.6)$$

O conjunto de restrições 3.2 faz com que a demanda de canais de cada ERB seja satisfeita. O conjunto 3.3 exige que os canais atribuídos para cada estação respeitem as distâncias de separação com outras ERBs que causem interferência. As restrições 3.4

garantem que, se um canal foi alocado para alguma ERB, o mesmo será marcado como usado e contabilizado na função objetivo. Por fim, as restrições de 3.5 e 3.6 garantem que todas as variáveis são binárias.

- **Minimização da extensão usada (*span*) de canais**

A extensão usada (*span*) do espectro eletromagnético é definida como a diferença entre o maior e o menor canal dentre todos os que foram alocados para as estações.

Uma formulação de programação linear inteira para este problema pode ser dada como segue. Partindo do modelo para a minimização da ordem, adiciona-se duas variáveis inteiras  $z_{max}$  e  $z_{min}$ , onde  $z_{max}$  cujo valor será o do maior canal usado e  $z_{min}$  o do menor canal usado. Mantendo-se os conjuntos de variáveis  $x_{if}$  e  $y_f$  e denotando-se por  $f_{max}$  o maior canal disponível dentre todos os domínios de frequência, tem-se a seguinte formulação de programação inteira:

$$\text{Minimizar } z_{max} - z_{min} \quad (3.7)$$

$$\text{Sujeito a } \sum_{f \in F_i} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \quad (3.8)$$

$$x_{if} + x_{jg} \leq 1 \quad (\forall i, j = 1, 2, \dots, n; f \in F_i; g \in F_j : |f - g| < d_{ij}) \quad (3.9)$$

$$x_{if} \leq y_f \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.10)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.11)$$

$$y_f \in \{0, 1\} \quad (\forall f \in F) \quad (3.12)$$

$$z_{max} \geq f y_f \quad (\forall f \in F) \quad (3.13)$$

$$z_{min} \leq f_{max} - (f_{max} - f) y_f \quad (\forall f \in F) \quad (3.14)$$

$$z_{max}, z_{min} \in \mathbb{Z}_+ \quad (3.15)$$

Em relação ao modelo de minimização da ordem, a função objetivo é modificada para a minimização da extensão usada do espectro (conforme definido) anteriormente, e são adicionados mais 3 conjuntos de restrições: o conjunto 3.13 garante que a variável  $z_{max}$  tenha o valor do maior canal usado, enquanto o conjunto 3.14 faz com que a variável  $z_{min}$  receba o valor do menor canal utilizado. As restrições 3.15 garantem a integralidade das novas variáveis introduzidas.

Quando todos os domínios de frequência são iguais ( $F = F_1 = F_2 = \dots = F_n$ ) e os

mesmos consistem de números consecutivos (ou seja, um intervalo de inteiros positivos), a minimização do *span* equivale à minimização do maior canal utilizado (já que o menor será sempre o primeiro valor dos domínios de frequência). Além disso, convém notar que, quando todos os domínios de frequência equivalem a  $\mathbb{Z}_+$  e, entre ERBs vizinhas, basta que os canais alocados para as mesmas sejam diferentes (ou seja, todas as distâncias de separação iguais a 1), a minimização do *span* equivale à minimização da ordem.

- **Minimização da interferência**

Em alguns casos, é possível que o conjunto de canais disponíveis para a rede não seja suficiente para que a alocação seja feita respeitando todas as separações. Assim, pode-se permitir que os pesos das arestas sejam violados, sendo que neste cenário haverá interferência proporcional à quanto cada aresta foi violada.

Considere duas ERBs  $i$  e  $j$ , cujos canais são  $f$  e  $g$  respectivamente, ou seja,  $x_{if} = x_{jg} = 1$ . Uma penalidade  $p_{ijfg}$  será aplicada neste caso, que poderá ser 0 se não houver interferência. A partir do modelo de minimização da ordem, tem-se o modelo para minimização da interferência modificando-se a função objetivo para a seguinte equação:

$$\sum_{\substack{i=1,2,\dots,n; \\ j=1,2,\dots,n}} \sum_{\substack{f \in F_i; \\ g \in F_j}} p_{ijfg} x_{if} x_{jg}$$

No entanto, tal equação não é linear. Para obter uma formulação de programação linear inteira equivalente, deve-se adicionar um novo conjunto de variáveis  $z_{ijfg}$ , onde:

$$z_{ijfg} = \begin{cases} 1 & \text{se o canal } f \text{ está alocado para a ERB } i \text{ e o canal } g \text{ está alocado para a ERB } j; \\ 0 & \text{caso contrário} \end{cases}$$

Tem-se então o seguinte modelo:



$$\text{Minimizar } \sum_{\substack{i=1,2,\dots,n; \\ j=1,2,\dots,n}} \sum_{\substack{f \in F_i; \\ g \in F_j}} p_{ijfg} z_{ijfg} \quad (3.16)$$

$$\text{Sujeito a } \sum_{f \in F_i} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \quad (3.17)$$

$$x_{if} + x_{jg} \leq 1 + z_{ijfg} \quad (\forall i, j = 1, 2, \dots, n; f \in F_i; g \in F_j : |f - g| < d_{ij}) \quad (3.18)$$

$$x_{if} \leq y_f \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.19)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (3.20)$$

$$y_f \in \{0, 1\} \quad (\forall f \in F) \quad (3.21)$$

$$z_{ijfg} \in \{0, 1\} \quad (\forall i, j = 1, 2, \dots, n; f \in F_i; g \in F_j) \quad (3.22)$$

Além da mudança na função objetivo, é importante verificar que o lado direito do conjunto de restrições 3.18 é modificado para comportar o novo conjunto de variáveis, ou seja, se  $x_{if} = x_{jg} = 1$ , então  $z_{ijfg} = 1$  e, portanto,  $z_{ijfg} + 1 = 2$ . Logo, o modelo é linear.

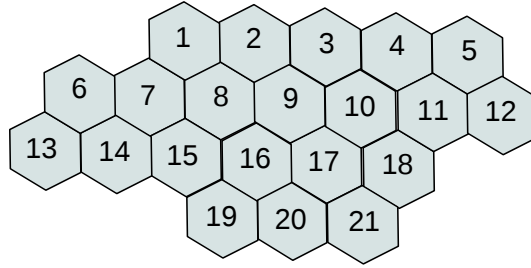
### 3.1.2.2 Ambientes de teste

Conforme modelos e algoritmos foram desenvolvidos para os problemas de alocação de canais, vários *benchmarks* foram criados de forma a analisar a eficácia dos métodos de solução em diversos cenários. Os principais conjuntos de instâncias serão mostrados a seguir.

- **CALMA - *Combinatorial Algorithms for Military Applications***

O projeto CALMA teve como objetivo analisar diversos métodos para problemas de otimização combinatória, de forma a identificar vantagens e desvantagens das diferentes abordagens [2]. Para a sua aplicação, os problemas de alocação de canais foram escolhidos e pesquisadores da França, Holanda e Reino Unido desenvolveram algoritmos para os mesmos.

Para a aplicação dos métodos, foram desenvolvidos dois conjuntos de instâncias, sendo um baseado em redes reais e outro gerado artificialmente. O primeiro conjunto, CELAR, foi fornecido pelo *Centre d'Électronique de L'Armement France* e contém 11



**Figura 3.8:** Rede de 21 células utilizada nas instâncias *Philadelphia*.

problemas para os três critérios de otimização. O segundo conjunto, GRAPH (*Generating Radio Link frequency Assignment Problems Heuristically*), contém 14 instâncias pseudoaleatórias, também para as três funções objetivos supracitadas.

Em todas as instâncias, cada ERB  $i$  tem demanda unitária e deve receber um canal a partir de seu domínio de frequência  $D_i$ . Entre duas estações  $i$  e  $j$  que podem interferir entre si, existe uma distância de separação  $c_{ij}$  que deve ser respeitada, sendo que existem dois cenários possíveis: sendo  $f$  o canal atribuído a  $i$  e  $g$  o canal de  $j$ , ou  $|f - g| \geq c_{ij}$  ou  $|f - g| = c_{ij}$ . Este segundo cenário ocorre em comunicações **duplex**, onde cada dispositivo é tanto receptor quanto transmissor e, para cada sentido de comunicação, um canal deve ser alocado e a distância entre os mesmos deve ser exatamente igual à distância de separação [65]. Os dados sobre as instâncias CALMA (critério de otimização e quantidade de estações e de restrições de separação) são dados na Tabela 3.3.

- ***Philadelphia***

Este conjunto de instâncias foi um dos primeiros presentes na literatura a serem modelados a partir de cenários reais utilizando células hexagonais [8]. As instâncias foram definidas com base em uma rede celular existente na cidade da Filadélfia, nos Estados Unidos, composta por 21 células, conforme mostrado na Figura 3.8. A partir desta rede, foram definidas diferentes demandas e restrições para cada instância.

Ao contrário das instâncias do projeto CALMA, as células das instâncias *Philadelphia* possuem demandas distintas de canais (ao invés de todas unitárias) e os domínios de frequência não são definidos, ou seja, para cada célula  $i$ , tem-se que  $D_i = D = \mathbb{Z}_+$ . As interferências entre duas células  $i$  e  $j$  são dadas de acordo com a distância geográfica  $w_{ij}$  entre as mesmas, porém, as instâncias são definidas em um sistema *two-band buffering*, de forma que só haverá interferências entre células cuja distância seja 1 ou 2 (e entre os diferentes canais para a mesma célula).

**Tabela 3.3:** Instâncias do projeto CALMA

<b>Conjunto</b>	<b>Núm. Inst.</b>	<b>Critério</b>	<b>ERBs</b>	<b>Restrições</b>
CELAR	01	Min. Ordem	916	5548
	02	Min. Ordem	200	1235
	03	Min. Ordem	400	2760
	04	Min. Ordem	680	3967
	05	Min. <i>Span</i>	400	2598
	06	Min. Interf.	200	1322
	07	Min. Interf.	400	2865
	08	Min. Interf.	916	5744
	09	Min. Interf.	680	4103
	10	Min. Interf.	680	4103
	11	Min. Ordem	680	4103
GRAPH	01	Min. Ordem	200	1134
	02	Min. Ordem	400	2245
	03	Min. <i>Span</i>	200	1134
	04	Min. <i>Span</i>	400	2244
	05	Min. Interf.	200	1134
	06	Min. Interf.	400	2170
	07	Min. Interf.	400	2170
	08	Min. Ordem	680	3757
	09	Min. Ordem	916	5246
	10	Min. <i>Span</i>	680	3907
	11	Min. Interf.	680	3757
	12	Min. Interf.	680	4017
	13	Min. Interf.	916	5273
	14	Min. Ordem	916	4683

Para determinar as distâncias de separação entre canais a serem alocados para as diferentes células, utiliza-se uma **matriz de interferência**  $C$  de dimensão  $n \times n$  onde cada elemento  $c_{ij}$  de  $C$  consiste na separação mínima a ser respeitada entre todos os canais alocados para  $i$  e todos os alocados para  $j$ . O elemento  $c_{ii}$  é a distância *co-site* a ser respeitada entre os diferentes canais de uma mesma célula.

O conjunto de dados (matriz de interferência e demandas de canais das células) para as instâncias *Philadelphia* mais utilizado na literatura é dado em [25]. As Tabelas 3.4 e 3.5 mostram as distâncias de separação de acordo com a distância geográfica e as demandas, respectivamente.

- **Helsinki**

Este conjunto de instâncias também foi gerado a partir de um cenário real. A rede utilizada, baseada em uma da cidade de Helsinque, Finlândia, possui 25 células [51]. Os domínios de frequência são, assim como o conjunto *Philadelphia*, equivalentes ao conjunto dos inteiros positivos. As demandas de cada célula são distintas.

A matriz de interferências para estas instâncias é dada da seguinte forma: a separação *co-site* da célula  $i$ , isto é, o elemento  $c_{ii}$  da matriz, tem valor 5. Para cada célula  $j$  que possa interferir com  $i$ , tem-se que  $c_{ij} = c_{ji} = 1$ . Todos os outros elementos da matriz são 0.

A lista de estações que interferem com cada uma das 25 células e os vetores de demanda utilizados é dada na Tabela 3.6. As Tabelas 3.7 e 3.8 mostram as distâncias de separação de acordo com a distância geográfica e os vetores de demanda usados, respectivamente.

- **Instâncias artificiais de 55 células**

Em [25], para realizar experimentos com o algoritmo proposto pelo autor, além do uso das instâncias *Philadelphia* e *Helsinki*, foram criadas outras duas instâncias artificiais envolvendo uma rede de 55 células, que pode ser vista na Figura 3.9.

Da mesma forma que as instâncias *Philadelphia*, as artificiais usam um sistema de *two-band buffering* e demandas distintas. A correlação entre as distâncias geográficas e as distâncias de separação de canais para estas instâncias são dadas na Tabela 3.9, enquanto as demandas são dadas na Tabela 3.10.

**Tabela 3.4:** Distâncias de separação na matriz de interferências entre células para cada instância do conjunto *Philadelphia*.

Nome da matriz	Distância entre células $i$ e $j$	Valor do elemento $c_{ij}$	Distância <i>co-site</i> ( $c_{ii}$ )
$C_{21}^1$	1	1	7
	2	1	
	> 2	0	
$C_{21}^2$	1	2	7
	2	1	
	> 2	0	
$C_{21}^3$	1	1	6
	2	1	
	> 2	0	
$C_{21}^4$	1	2	6
	2	1	
	> 2	0	
$C_{21}^5$	1	1	5
	2	1	
	> 2	0	
$C_{21}^6$	1	2	5
	2	1	
	> 2	0	
$C_{21}^7$	1	1	4
	2	1	
	> 2	0	
$C_{21}^8$	1	2	4
	2	1	
	> 2	0	

**Tabela 3.5:** Vetores de demanda para as instâncias *Philadelphia*.

Nome do vetor	Demandas por célula
$D_{21}^1$	8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8
$D_{21}^2$	5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25

**Tabela 3.6:** Estações interferentes para cada célula na rede das instâncias *Helsinki*.

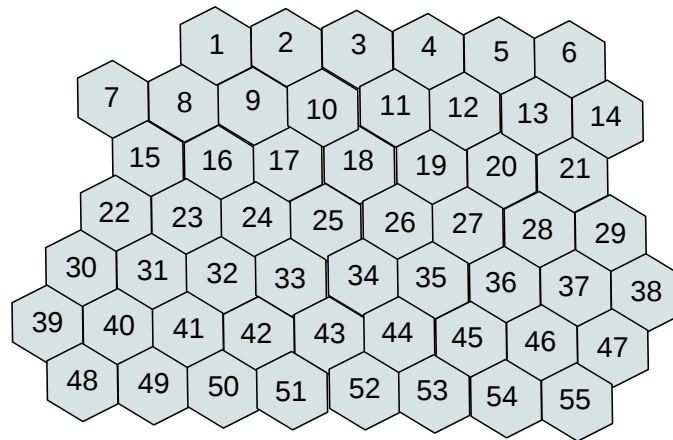
<b>Célula</b>	<b>Núm. Interferentes</b>	<b>Estações</b>
1	11	2, 3, 5, 7, 8, 9, 10, 12, 13, 14, 15
2	10	1, 3, 5, 7, 8, 10, 12, 13, 14, 15
3	13	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
4	11	3, 7, 8, 9, 10, 11, 12, 13, 23, 24, 25
5	10	1, 2, 3, 10, 11, 12, 13, 14, 15, 16
6	5	3, 7, 8, 9, 10
7	11	1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13
8	12	1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 24
9	11	1, 3, 4, 6, 7, 8, 10, 11, 12, 24, 25
10	17	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 22, 24
11	19	3, 4, 5, 7, 8, 9, 10, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23
12	11	1, 2, 3, 4, 5, 7, 8, 9, 10, 13, 14
13	17	1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20
14	14	1, 2, 3, 5, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20
15	14	1, 2, 5, 10, 11, 13, 14, 16, 17, 18, 19, 20, 21, 22
16	10	5, 10, 11, 13, 14, 15, 17, 18, 19, 20
17	6	13, 14, 15, 16, 18, 19
18	8	11, 13, 14, 15, 16, 17, 19, 20
19	11	11, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23
20	10	11, 13, 14, 15, 16, 18, 19, 21, 22, 23
21	6	11, 15, 19, 20, 22, 23
22	9	10, 11, 15, 19, 20, 21, 23, 24, 25
23	8	4, 11, 19, 20, 21, 22, 24, 25
24	8	4, 8, 9, 10, 11, 22, 23, 25
25	6	4, 9, 11, 22, 23, 24

**Tabela 3.7:** Distâncias de separação na matriz de interferências entre células para as instâncias *Helsinki*.

Nome da matriz	Distância entre células $i$ e $j$	Valor do elemento $c_{ij}$	Distância <i>co-site</i> ( $c_{ii}$ )
$C_{25}^1$	1	1	2
	2	1	
	> 2	0	

**Tabela 3.8:** Vetores de demanda para as instâncias *Helsinki*

Nome do vetor	Demandas por célula
$D_{25}^3$	10, 11, 9, 5, 9, 4, 5, 7, 4, 8, 8, 9, 10, 7, 7, 6, 4, 5, 5, 7, 6, 4, 5, 7, 5
$D_{25}^4$	5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25, 8, 5, 5, 5



**Figura 3.9:** Rede de 55 células usada nas instâncias artificiais.

**Tabela 3.9:** Distâncias de separação na matriz de interferências entre células para as instâncias artificiais.

Nome da matriz	Distância entre células $i$ e $j$	Valor do elemento $c_{ij}$	Distância <i>co-site</i> ( $c_{ii}$ )
$C_{55}^1$	1	1	7
	2	1	
	> 2	0	

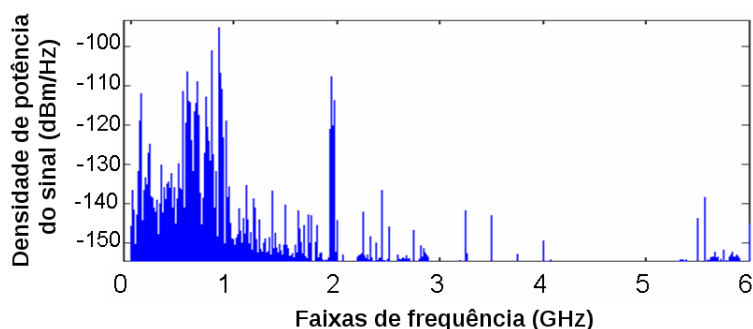
**Tabela 3.10:** Vetores de demanda para as instâncias artificiais.

Nome do vetor	Demandas por célula
$D_{55}^5$	5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25, 8, 5, 5, 5, 5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25, 8, 5, 5, 5, 25, 8, 5, 5, 5
$D_{55}^6$	10, 11, 9, 5, 9, 4, 5, 7, 4, 8, 8, 9, 10, 7, 7, 6, 4, 5, 5, 7, 6, 4, 5, 7, 5, 10, 11, 9, 5, 9, 4, 5, 7, 4, 8, 8, 9, 10, 7, 7, 6, 4, 5, 5, 7, 6, 4, 5, 7, 5, 6, 4, 5, 7, 5

## 3.2 Redes baseadas em rádios cognitivos

Como discutido anteriormente, o número de usuários dos sistemas de comunicação sem fio tem aumentado bastante, porém, a oferta do espectro eletromagnético disponível não tem acompanhado tal aumento. No entanto, nem todas as faixas de frequência estão saturadas: muitas delas, apesar de licenciadas para usos específicos, possuem sobras na capacidade de usuários. A Figura 3.10 ilustra tal situação.

Em determinadas faixas de frequência, existem muitas variações (tanto temporais quanto geográficas) da taxa de utilização das mesmas. Uma forma de mitigar a escassez do espectro eletromagnético envolve a alocação dos canais em função do tempo, de forma que um dispositivo ocupe faixas que estão vagas em determinados instantes do tempo. No entanto, esta abordagem exigiria informação antecipada e exata sobre os instantes das variações, o que não ocorre na prática [23].



Frequências (GHz)	0 a 1	1 a 2	2 a 3	3 a 4	4 a 5	5 a 6
Uso (%)	54,4	35,1	7,6	0,25	0,128	4,6

**Figura 3.10:** Utilização da faixa de 0 a 6 GHz do espectro eletromagnético no Centro de Berkeley, Estados Unidos. Adaptado de Yang, 2005 [93].



### 3.2.1 Estrutura de uma rede cognitiva

Em uma rede com rádios cognitivos, existem dois tipos de usuários: primários e secundários. Os usuários primários são os que estão devidamente licenciados para acessar uma determinada faixa de frequências, enquanto os secundários são os usuários não licenciados, caso dos dispositivos de rádio cognitivo. Quando alguma faixa não está sendo usada por usuários primários, os secundários podem utilizá-las, no entanto, não podem causar interferências excessivas com os primários, que tem prioridade na rede.

Uma abordagem para permitir o uso mais eficaz do espectro consiste em rádios capazes de detectar as faixas subutilizadas de frequências licenciadas e efetuar suas comunicações através delas, liberando as mesmas quando um usuário devidamente licenciado precisa utilizá-las. Esta característica é fundamental aos chamados rádios cognitivos, que fazem uso oportunista dos canais não usados em determinados instantes, maximizando a eficácia da alocação [85].

#### 3.2.1.1 Sensoriamento espectral

O sensoriamento espectral consiste em técnicas utilizadas pelos rádios cognitivos para a detecção da presença de usuários primários em determinadas faixas de frequência. Existem três principais abordagens para isso: a detecção de transmissores, a detecção cooperativa e a detecção por interferências ([3, 10]), que serão vistas a seguir.

- **Detecção de transmissores**

Nesta abordagem, o rádio cognitivo tenta detectar algum sinal enviado por um usuário primário na faixa desejada. O modelo tradicional consiste em um teste de hipótese para um canal em um instante de tempo  $t$  formulado da seguinte forma :

$$x(t) = \begin{cases} n(t) & H_0 \\ h \times s(t) + n(t) & H_1 \end{cases}$$

Onde  $x(t)$  é o sinal recebido pelo rádio cognitivo,  $s(t)$  é o sinal transmitido pelo usuário primário,  $n(t)$  é o ruído branco gaussiano aditivo (RBGA) e  $h$  é o ganho de amplitude do canal. A hipótese nula ( $H_0$ ) é o cenário sem usuários primários, enquanto a hipótese alternativa ( $H_1$ ) tem a presença dos mesmos.

Existem três principais esquemas para a detecção de transmissores. A detecção

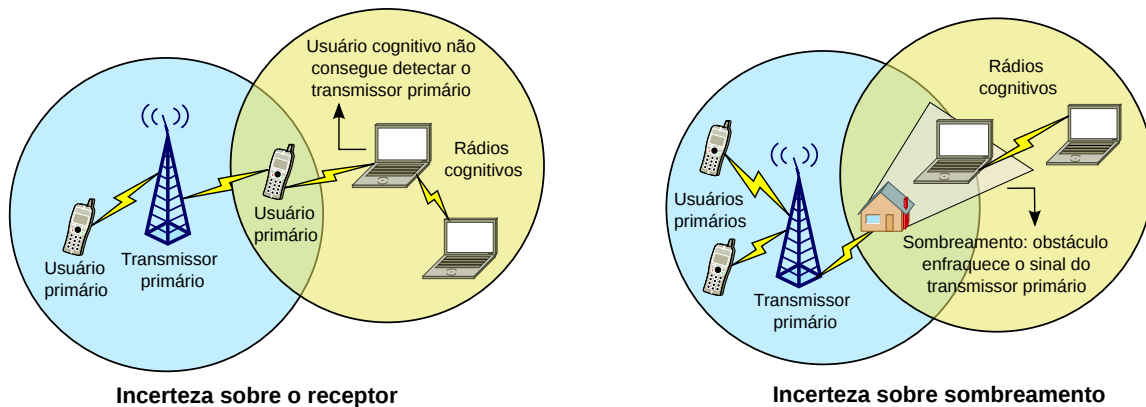
baseada em filtro casado (*matched filter*) é utilizada quando toda a natureza do sinal do usuário primário é conhecida pelo rádio cognitivo, sendo o melhor esquema nessa situação. A detecção baseada em energia usada quando não há informação completa sobre o sinal do usuário primário, sendo necessária apenas a potência do RBGA. Um filtro passa-banda é aplicado ao sinal recebido pelo rádio cognitivo, que posteriormente é elevado ao quadrado e integrado no intervalo de observação. O resultado é então comparado com um limiar para detectar a presença de um usuário primário. Por fim, a detecção de característica cicloestacionária toma proveito do fato de que a maioria dos sinais modulados são acompanhados de ondas senóides, trens de pulso e prefixos cíclicos, o que cria uma periodicidade no sinal recebido. Tal característica é detectada por meio de uma função de correlação espectral, capaz de diferenciar entre ruído e sinal modulado, dada a periodicidade do sinal que não ocorre no ruído puro.

- **Detecção cooperativa**

Em alguns cenários, a abordagem de detecção de transmissores pode não encontrar corretamente a fonte de sinal de um transmissor primário. A rede de rádio cognitivo e a rede de dispositivos licenciados podem estar fisicamente separadas (ou seja, o sinal do rádio cognitivo não alcança o transmissor primário e vice-versa), mas havendo uma intersecção nas áreas de cobertura de cada rede, um receptor primário pode estar localizado em tal intersecção, cenário conhecido como incerteza sobre o receptor. Se o rádio cognitivo, não encontrando o transmissor primário, usar a faixa do espectro desejada, a recepção do usuário primário na intersecção das redes sofrerá interferência indesejada. Outra situação pode ocorrer quando o rádio cognitivo está na área de cobertura do transmissor (o que deveria possibilitar a detecção normal), porém, existe um obstáculo físico que atenua o sinal que chega ao rádio cognitivo, o que pode impossibilitar que o uso do espectro por usuários primários seja verificado corretamente, cenário conhecido como incerteza sobre sombreamento. A Figura 3.11 exemplifica estes cenários.

Para contornar essas situações, a abordagem de detecção cooperativa utiliza informações dos vários rádios cognitivos na rede para detectar vagas na faixa desejada. Cada dispositivo cognitivo aplica a detecção de transmissores ao seu redor e comunica-se com outros rádios cognitivos que necessitem de tais informações.

Existem três principais esquemas de detecção cooperativa [5]. Na detecção centralizada, um dos dispositivos da rede cognitiva que esteja apto a comunicar-se com todos



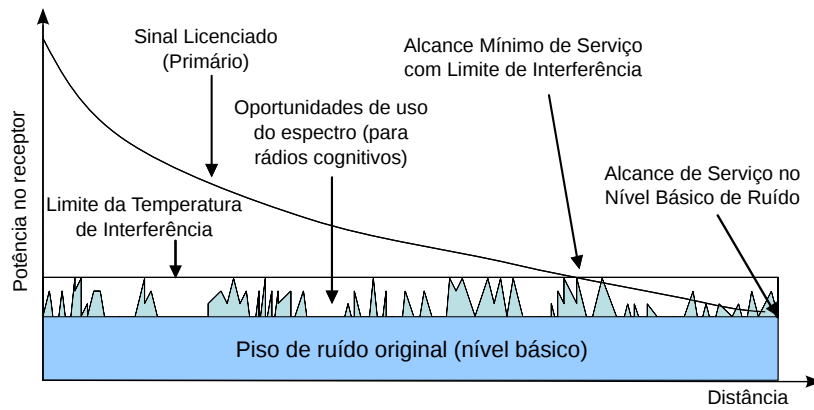
**Figura 3.11:** Exemplos dos cenários nos quais a detecção de transmissores pode falhar.

os outros rádios cognitivos é eleito como nó principal (chamado de centro de fusão). Este nó será responsável por verificar transmissores primários com base nas informações colhidas pelos outros rádios cognitivos, repassando aos mesmos quais faixas estão disponíveis em um determinado instante. Na detecção distribuída, cada dispositivo envia suas informações sobre detecção de transmissores primários a todos os outros rádios cognitivos. Cada um destes deverá decidir com base em um critério local se existe ou não algum usuário primário ocupando a faixa desejada. Se o critério não for satisfeito, os resultados são enviados para os outros rádios cognitivos até que uma decisão unânime seja alcançada. Já na detecção baseada em repasses, a comunicação de informações de um rádio cognitivo a outro em algum dos esquemas acima pode se tornar difícil se o canal usado causar muita degradação de sinal, seja por questões de distância ou de interferência. Neste esquema, alguns rádios são usados como intermediários para repasse das informações entre outros dois rádios, onde cada ponta tem um canal forte com o rádio intermediário, mas um fraco entre as mesmas diretamente.

- **Detecção baseada em interferências**

As interferências em uma rede sem fio são causadas pelos dispositivos transmissores, podendo ser reguladas pelos mesmos através do ajuste de parâmetros da transmissão, como a potência do sinal emitido. No entanto, são os dispositivos receptores os mais afetados com a interferência, o que torna difícil aos transmissores alterarem seus parâmetros de acordo com os níveis de interferência dos receptores a cada instante de tempo.

Para que o próprio receptor consiga regular os níveis de interferência, a Comissão Federal de Comunicações (*Federal Communications Commission - FCC*) dos Estados



**Figura 3.12:** Modelo de temperatura de interferência. Adaptado de Akyildiz *et al.*, 2006 [3].

Unidos propôs um novo modelo de medição de interferência chamado de temperatura de interferência, apropriado para utilização dinâmica do espectro eletromagnético. Nele, os sinais de rádio tem um alcance no qual os mesmos aproximam-se do nível básico de ruído. À medida que mais sinais são transmitidos na mesma faixa de frequência, o piso de ruído aceitável é incrementado. Os receptores possuem um nível denominado de limite de temperatura de interferência o qual o piso de ruído não deve exceder para que não haja interferências excessivas nos dispositivos.

A partir disso, os rádios cognitivos podem utilizar as faixas de frequência desejadas cujos pisos de ruído não excedam os limites de temperatura. A Figura 3.12 ilustra o modelo.

### 3.2.2 Problemas de alocação de canais em redes de rádios cognitivos

Após a detecção das oportunidades de uso do espectro eletromagnético licenciado, os rádios cognitivos podem passar a ocupá-lo para efetuar suas comunicações. Os mesmos devem coexistir com os usuários primários que estejam presentes na rede (que têm prioridade no uso das faixas). Para evitar interferências nos usuários primários, os rádios cognitivos devem usar faixas que estejam desocupadas (os chamados “buracos espectrais”) no instante desejado, migrando para outras quando um usuário primário passa a usá-las. A Figura 3.13 ilustra este conceito.

O problema de alocação de canais em redes de rádios cognitivos pode ser enunciado como se segue. Existem  $n$  dispositivos de rádio cognitivo e  $m$  usuários primários presentes na rede. O espectro eletromagnético disponível é dividido em  $q$  canais a serem alocados para os usuários. O ganho de canal na ocasião de que o  $i$ -ésimo rádio cognitivo utilize o

$k$ -ésimo canal é denotado por  $a_{ik}$ , e potência alocada para tal par é denotado por  $P_{ik}$ . A relação sinal-ruído (*signal-noise ratio* - SNR) no  $i$ -ésimo rádio cognitivo, denotada por  $\gamma_i$  é dada por:

$$\gamma_i = \frac{a_{ik}P_{ik}}{\sum_{\substack{j=1 \\ j \neq i}}^n a_{jk}P_{jk} + N_0}$$

Onde  $N_0$  é o nível básico de ruído. O cálculo da razão sinal-ruído utilizado nesta equação leva em conta vários rádios cognitivos que possam estar utilizando o mesmo canal: para um dado rádio  $i$  utilizando o canal  $k$ , a comunicação realizada por algum outro rádio  $j$  no mesmo canal será considerada como interferência, juntamente com o piso  $N_0$ . Considerando apenas um usuário por canal, tem-se:

$$\gamma_i = \frac{a_{ik}P_{ik}}{N_0}$$

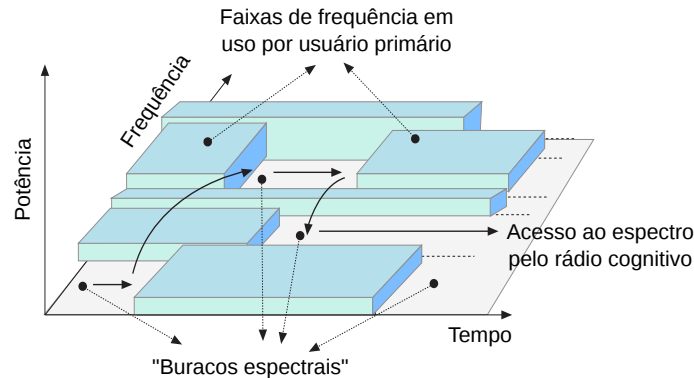
A taxa máxima possível de transferência para o rádio  $i$ , denotada por  $\bar{R}_i$  é dada pela Capacidade de Shannon, calculada da seguinte forma:

$$R_i = \sum_{k=1}^q \log_2(1 + \gamma_i) \Rightarrow R_i = \sum_{k=1}^q \log_2 \left( 1 + \frac{a_{ik}P_{ik}}{\sum_{j=1, j \neq i}^n a_{jk}P_{jk} + N_0} \right)$$

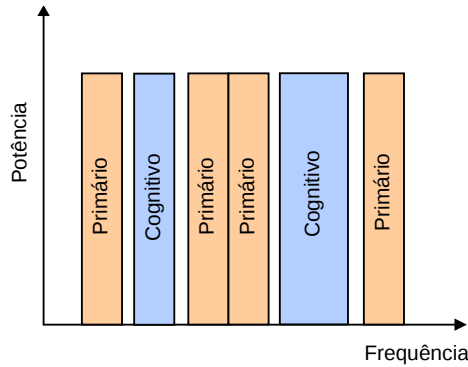
No caso de apenas um usuário por canal:

$$R_i = \sum_{k=1}^q \log_2 \left( 1 + \frac{a_{ik}P_{ik}}{N_0} \right)$$

Mantendo a relação sinal-ruído a um nível baixo, pode-se aproximar o cálculo de  $R_i$



**Figura 3.13:** Esquema do conceito de buracos espectrais. Adaptado de Akyildiz *et al.*, 2008 [4].



**Figura 3.14:** Esquema de alocação *overlay*.

da seguinte forma:

$$R_i = \frac{\log e}{N_0} \sum_{k=1}^q a_{ik} P_{ik}$$

Cada um dos rádios cognitivos possui um valor mínimo de capacidade de transmissão  $\underline{R}_i$  a ser atendido pela alocação, ou seja, deve-se garantir que  $R_i \geq \underline{R}_i$ .

Duas importantes características devem ser levadas em consideração ao efetuar a alocação de canais para os rádios cognitivos, da mesma forma que ocorre em redes celulares: o esquema de uso do espectro e o objetivo a ser alcançado pela alocação.

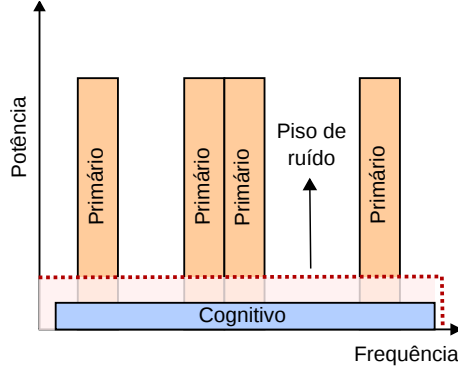
Existem dois principais esquemas de alocação de canais para os rádios cognitivos. Na alocação *overlay*, os rádios cognitivos deverão, em todo instante, utilizar apenas as faixas que constituem buracos espectrais, causando o mínimo de interferência aos usuários primários, como ilustrado pela Figura 3.14.

Já o esquema *underlay* permite que os rádios cognitivos utilizem a mesma faixa de frequências dos usuários primários mesmo que estes estejam ocupando os canais desejados. Para isso, é utilizada a técnica de *spread spectrum*, na qual a transmissão é codificada, sendo que na rede cognitiva ela é realizada a uma potência abaixo do nível de ruído, que não causará interferências aos usuários primários e só poderá ser interpretada pelos rádios cognitivos. Para compensar a baixa potência, é utilizada uma largura de banda maior que na alocação *overlay*. A Figura 3.15 mostra este conceito.

### 3.2.2.1 Critérios de otimização

- **Minimização da potência total dos rádios cognitivos**

Como já foi visto, os rádios cognitivos devem efetuar suas comunicações causando o



**Figura 3.15:** Esquema de alocação *underlay*.

mínimo possível de interferência aos usuários primários. Considerando os seguintes conjuntos de variáveis:

$$\ell_{ik} = \begin{cases} 1 & \text{se o canal } k \text{ for atribuído ao rádio cognitivo } i; \\ 0 & \text{caso contrário.} \end{cases}$$

$P_{ik}$  = potência atribuída ao canal  $k$  no rádio  $i$ .

Tem-se a seguinte formulação matemática [63]:

$$\text{Minimizar } \sum_{i=1}^n \sum_{k=1}^q \ell_{ik} P_{ik} \quad (3.23)$$

$$\text{Sujeito a } \sum_{i=1}^n \ell_{ik} \leq 1 \quad (\forall k = 1, 2, \dots, q) \quad (3.24)$$

$$\sum_{k=1}^q \log_2 \left( 1 + \frac{a_{ik} P_{ik}}{\sum_{j=1, j \neq i}^n a_{jk} P_{jk} + N_0} \right) \geq \underline{R}_i \quad (\forall i = 1, 2, \dots, n) \quad (3.25)$$

$$\ell_{ik} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; \forall k = 1, 2, \dots, q) \quad (3.26)$$

$$P_{ik} \in \mathbb{R} \quad (\forall i = 1, 2, \dots, n; \forall k = 1, 2, \dots, q) \quad (3.27)$$

O conjunto de restrições 3.24 garante que cada rádio cognitivo deve receber no máximo um canal. As restrições 3.25 faz com que os canais e as potências alocados para os rádios cognitivos satisfaçam a demanda da taxa de transferência. Por fim, as restrições 3.26 exigem que as variáveis  $\ell_{ik}$  assumam apenas os valores 0 ou 1 (as variáveis  $P_{ik}$  são irrestritas).

Um caso especial envolve limites para as potências que podem ser usadas em cada canal. Sendo  $\overline{P}_k$  a potência máxima a ser usada no canal  $k$ , pode-se adicionar a seguinte restrição para limitar a potência:

$$\sum_{i=1}^n \ell_{ik} P_{ik} \leq \overline{P}_k \quad (\forall k = 1, 2, \dots, q)$$

- **Maximização da utilização espectral**

Nesta abordagem [44], os usuários primários podem estar ocupando as faixas desejadas, logo, os rádios cognitivos devem minimizar a interferência causada aos primários. Para diferenciar a influência de um usuário primário e a de um rádio cognitivo, o ganho de canal será denotado por  $a_{ik}^j$ , consistindo no ganho de canal no  $i$ -ésimo dispositivo obtido pelo  $j$ -ésimo equipamento, ambos utilizando o canal  $k$ . O conjunto dos  $m$  usuários primários utilizando um canal  $k$  será denotado por  $(\Theta^k)$ .

Uma forma de garantir a minimização do impacto dos rádios cognitivos nos usuários primários é impondo um limite para a potência de todas as conexões oportunistas, denotado por  $\overline{\zeta}$ , de forma que:

$$\sum_{i=1}^n a_{ik}^j P_{ik} \leq \overline{\zeta} \quad (\forall j \in \Theta^k, \forall k = 1, 2, \dots, n)$$

Apesar da restrição de potência máxima, os rádios cognitivos não podem transmitir a uma potência muito baixa, pois a comunicação seria bastante degradada nesta situação. Assim, para cada um deles, a relação sinal-ruído deve ser maior que um limite inferior  $\overline{\gamma}$  para que as comunicações sejam satisfatórias, ou seja:

$$\gamma_i \geq \overline{\gamma} \quad (\forall i = 1, 2, \dots, n)$$

A partir disso, tem-se o seguinte modelo de programação inteira mista, utilizando as mesmas variáveis  $\ell_{ik}$ :



$$\text{Maximizar } \sum_{k=1}^q \sum_{i=1}^n \ell_{ik} \quad (3.28)$$

$$\text{Sujeito a } 0 \leq \sum_{i=1}^n \ell_{ik} \leq 1 \quad (\forall k = 1, 2, \dots, q) \quad (3.29)$$

$$a_{ik}^i P_{ik} - \bar{\gamma} \sum_{j=1, j \neq i}^n a_{ik}^j P_{jk} - \bar{\gamma} N_0 \geq (\ell_{ik} - 1) \delta \quad (3.30)$$

$$\sum_{i=1}^n a_{ik}^j P_{ik} \leq \bar{\zeta} \quad (\forall j \in \Theta^k, \forall k = 1, 2, \dots, n) \quad (3.31)$$

$$0 \leq \sum_{i=1}^n \ell_{ik} P_{ik} \leq \bar{P}_k \quad (\forall k = 1, 2, \dots, q) \quad (3.32)$$

$$\ell_{ik} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; \forall k = 1, 2, \dots, q) \quad (3.33)$$

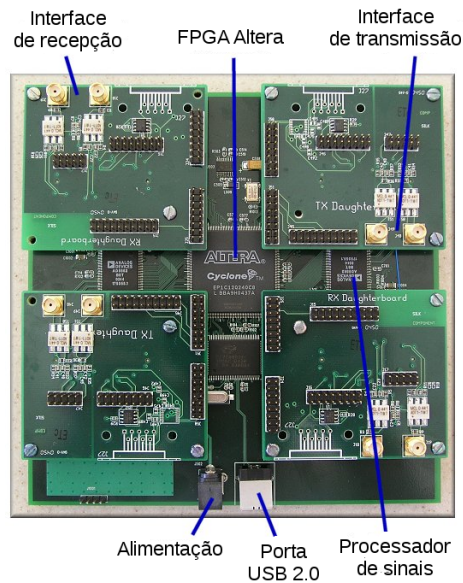
$$P_{ik} \in \mathbb{R} \quad (\forall i = 1, 2, \dots, n; \forall k = 1, 2, \dots, q) \quad (3.34)$$

Onde  $\delta$  é uma constante suficientemente grande. O conjunto de restrições 3.29 faz com que cada rádio cognitivo receba no máximo um canal. A restrição 3.30 garantem que o canal alocado a cada rádio cognitivo satisfaça o mínimo exigido para a relação sinal-ruído. O conjunto 3.31 limita a interferência causada pelos rádios cognitivos aos usuários primários. As restrições 3.32 garantem que a potência do sinal alocado para cada canal e rádio cognitivo não passe de um determinado limite superior. As restrições 3.33 fazem com que as variáveis  $\ell_{ik}$  sejam binárias. Por fim, o conjunto 3.34 permite que as variáveis  $P_{ik}$  sejam irrestritas.

### 3.2.2.2 Ambientes de teste

A principal abordagem utilizada na literatura para avaliação de algoritmos e métodos propostos para o problema de alocação de canais em redes cognitivas consiste na simulação de redes com quantidade variada de dispositivos, sendo alguns deles usuários primários para avaliação do impacto na rede licenciada já existente, além dos rádios cognitivos em si.

Alguns parâmetros utilizados na simulação são:



**Figura 3.16:** Uma placa para rádio definido por software.

- Número de rádios cognitivos ( $n$ ) e de usuários primários ( $m$ );
- Número de canais considerados ( $q$ );
- Área considerada contendo os dispositivos cognitivos;
- Raio da área de cobertura de cada dispositivo;
- Função de distribuição dos equipamentos na área considerada (geralmente uniforme);
- Fator de perda de potência em função da distância;
- Níveis máximos de interferência (em dBm) para os rádios cognitivos e usuários primários;
- Potência máxima a ser emitida em cada canal (em dBm).

Os rádios cognitivos são baseados em *software*, de forma que os algoritmos e modelos podem ser implementados diretamente nos dispositivos, sendo que alguns deles permitem também a simulação de uma rede cognitiva sem a necessidade da aquisição dos *hardwares* de rádio (ou seja, totalmente via *software*). Um exemplo de dispositivo de rádio definido por software é dado na Figura 3.16.

Alguns dos principais *frameworks* de *software* para uso em simulação e implementação de redes cognitivas incluem:

1. **GNU Radio:** De código aberto, é compatível com vários dispositivos e também pode ser utilizado para fazer uma simulação totalmente por *software* [74];
2. **Cognitive Radio Cognitive Network Simulator (CRCN):** Baseado no simulador NS-2, é uma ferramenta facilitadora de pesquisas de rádios cognitivos. Possui interface gráfica para definição de parâmetros e execução da simulação [94].

### 3.3 Notas do capítulo

Devido à grande importância prática, muitos estudos são realizados no âmbito da alocação de canais em redes sem fio. Um dos mais completos sobre redes celulares inclui uma tese [49] que foi posteriormente incorporada em *site* [37] e livro [50] mais atualizados. Há também um *survey* [10] envolvendo ambos os tipos de redes citados neste capítulo, com ênfase nos diferentes tipos de ambientes e interferências.

Em rádios cognitivos, devido à recente atenção de pesquisadores em torno do mesmo, há muitos estudos sobre aspectos tecnológicos dos mesmos. Um resumo sobre tomadas de decisões em redes cognitivas durante 10 anos pode ser visto em [46].

# Capítulo 4

## Problema da alocação de canais

### Min-Max

Neste capítulo, serão apresentados os modelos utilizados na literatura para representação de instâncias do problema, bem como as modificações propostas aos mesmos.

O problema da alocação de canais Min-Max consiste em atribuir canais para as diferentes chamadas da rede, respeitando restrições impostas na mesma e minimizando o valor do maior canal usado. Diversas variações podem ser definidas de acordo com as características do problema específico a ser resolvido, sendo as mesmas estudadas nos modelos teóricos deste capítulo.

A característica mais importante em todos os problemas de alocação de canais, incluindo o Min-Max, é a presença de interferência entre chamadas cujas ERBs estejam próximas umas das outras ou que usem canais próximos no espectro eletromagnético, de acordo com seu tipo (*co-site*, *co-canal* ou canal adjacente). Tal interferência é bastante prejudicial à qualidade do serviço móvel, portanto, deve ser evitada. Para isso, os canais alocados às chamadas devem respeitar uma distância de separação entre si, isto é, a diferença absoluta entre canais de duas chamadas distintas deve ser maior ou igual a essa distância.

Em alguns casos, há uma lista de canais associada à cada estação, de forma que suas chamadas devem utilizar canais pertencentes a essa lista, chamada de domínio de frequência da estação. Esta situação ocorre por diversos motivos que impeçam o uso de todos os canais em uma determinada região, por exemplo, em conexões próximas a limites entre países [49], onde as regras de disponibilidade de frequências pode diferir entre os terri-

tórios distintos, por razões legais ou tecnológicas. Quando todas as estações podem usar qualquer canal, diz-se que todos os domínios de frequência são iguais ao conjunto de canais disponíveis.

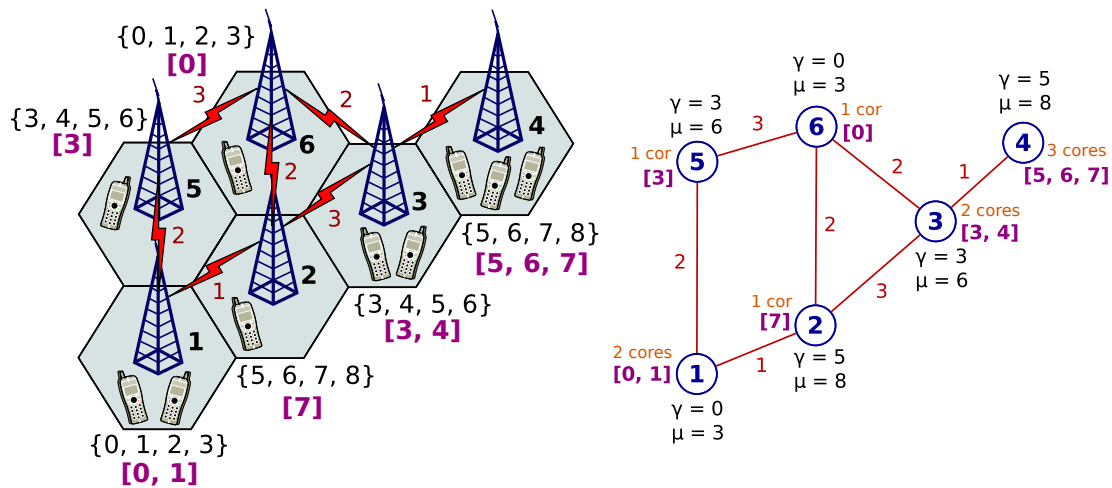
Dependendo do modelo utilizado, uma mesma estação pode ser responsável por uma ou diversas chamadas. Em alguns dispositivos de comunicação sem fio, a própria estação transmissora é a que necessita do tráfego, sem estar conectada a chamadas de outros dispositivos. Nesse caso, a estação é responsável somente por uma chamada, que é a sua própria. Mais comum em redes celulares é o cenário no qual, na região geográfica da célula definida por uma ERB, existem diversos usuários, cada um com um dispositivo móvel, os quais se comunicam com a ERB e necessitam de canais distintos para suas chamadas, respeitando suas distâncias de separação.

Muitas vezes é necessário combinar características de cada cenário de acordo com a aplicação. Por exemplo, as instâncias *Philadelphia* e suas variações contêm interferências entre chamadas na mesma ERB e em pares de ERBs cujas distâncias geográficas sejam de até duas unidades. Além disso, cada ERB é responsável por diversas chamadas em sua célula. Já nas instâncias CALMA, que também contêm interferências entre certos pares de ERBs, todas as estações possuem apenas uma chamada, porém, cada uma possui uma lista restrita de canais que podem ser alocados às mesmas.

## **4.1 Modelagem como colorações especiais de vértices em grafos**

A correlação da alocação de canais com os problemas de coloração de vértices em grafos são bem estabelecidos na literatura, inclusive com algumas generalizações do problema. Deste modo, identificou-se certas características nos problemas que pudessem ser exploradas em novos modelos.

Para modelar o problema de alocação de canais como coloração, inicialmente, são feitas as seguintes correlações: cada estação da rede de telefonia celular corresponde a um vértice do grafo; cada canal do espectro eletromagnético equivale a uma cor; os domínios de frequência serão listas de cores dos vértices. Como os domínios considerados são contíguos, ou seja, conjuntos com todos os naturais entre o menor e maior canal, tem-se limites inferior e superior; a quantidade de chamadas será o número de cores a serem



**Figura 4.1:** Problema de alocação de canais da figura 3.7 modelado como  $(\gamma, \mu)$ -channel-muticoloring. Os números entre colchetes à esquerda de cada vértice indicam as cores atribuídas ao mesmo. Abaixo de cada vértice, tem-se os limites inferior ( $\gamma$ ) e superior ( $\mu$ ).

atribuídas para o vértice; as distâncias de separação entre canais das estações serão os pesos das arestas.

Com base nas mesmas, pode-se enunciar o modelo de multicoloração de canais baseada em listas limitadas, ou seja,  $(\gamma, \mu)$ -channel-multicoloring. Retomando o exemplo da figura 3.7, tem-se que o mesmo pode ser modelado como esta coloração, pois os domínios de frequência são contíguos. O critério de otimização, que no Min-Max-CAP é a minimização do maior canal usado, passa a ser a minimização do **número cromático ponderado** do grafo, isto é, deseja-se colorir todos os vértices de acordo com suas demandas respeitando todas as restrições. A figura 4.1 mostra a modelagem do exemplo anterior como coloração em grafos, com número cromático ponderado igual a 7.

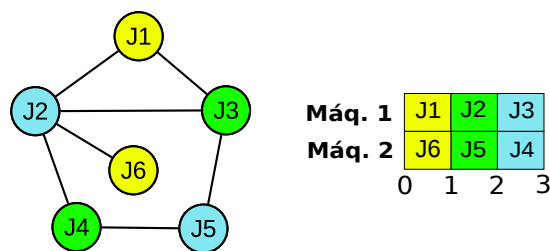
## 4.2 Modelagem em escalonamento de tarefas em máquinas paralelas com restrições de recursos

Apesar de ser comum na literatura correlacionar as variações do CAP com colorações em grafos, durante a pesquisa realizada não se encontrou uma correlação direta do CAP com escalonamento. No entanto, este tipo de associação não é algo totalmente novo, uma vez que é comum associar problemas de escalonamento com colorações em grafos, como exemplificado na Figura 4.2.

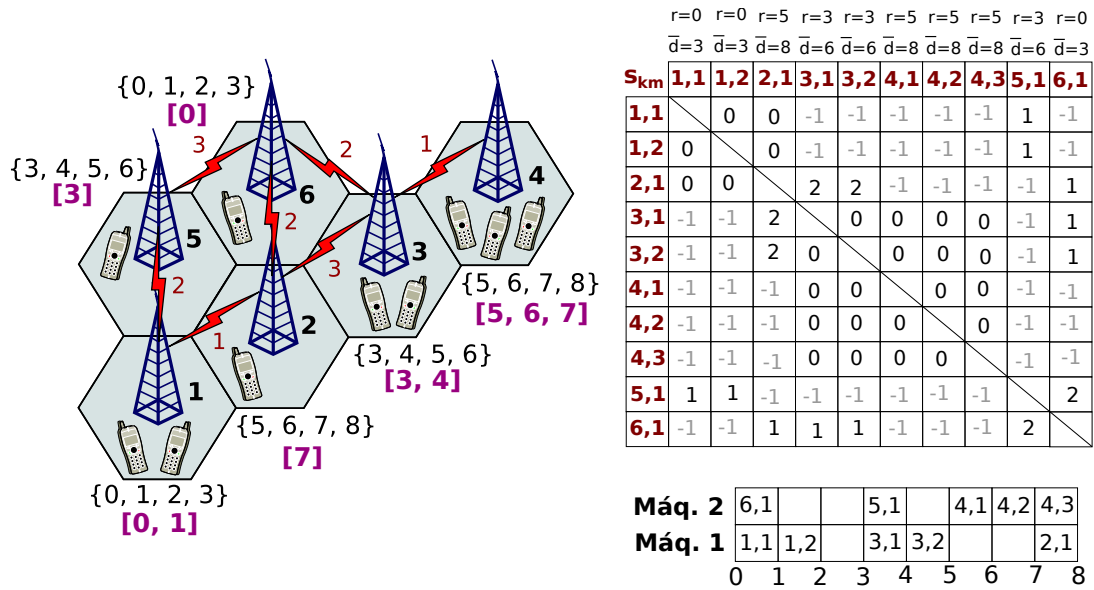
Conforme enunciado anteriormente, cada **chamada** das estações necessita de um canal a ser alocado para a mesma. Para efetuar a correlação com escalonamento de tarefas,

deve-se considerar, como tarefas, cada chamada do problema original de alocação de canais, ao invés de somente as estações. Os canais podem (e devem) ser reutilizados, de forma a fazer um uso mais eficiente do espectro eletromagnético, logo, algumas tarefas serão executadas ao mesmo tempo, ou seja, **em paralelo** [36], o que nos dá, como ambiente de execução,  $P$  (máquinas paralelas idênticas).

Pode-se relacionar o tempo em problemas de escalonamento com canal no problema de alocação de canais. Pode-se considerar os canais como os instantes de tempo do escalonamento e, por consequência, cada chamada deverá ter um instante de tempo alocado para a mesma em alguma máquina. Considera-se cada canal como uma unidade de tempo, assim, o tempo de processamento da tarefa associada a qualquer chamada será  $p_k = 1$ . No caso dos domínios de frequência, considerando-se o caso de intervalos contíguos, tem-se uma **janela de tempo** para a execução da tarefa associada [35], ou seja, existe um intervalo de tempo limitado dentro do qual a tarefa deve ser executada. Tem-se então que, para uma chamada  $k$ , a mesma terá um *release date* equivalente ao limite inferior do domínio de frequência de sua estação e um *deadline* equivalente ao limite superior do domínio (usando o exemplo de coloração, tem-se que  $r_k = \gamma_i$ , onde  $i$  é a estação da chamada  $k$ , e  $\bar{d}_j = \mu_i$ ). Como deve existir uma distância entre canais de estações adjacentes, deve-se adicionar uma restrição de separação entre tarefas associadas às chamadas que possam interferir entre si. Isso é feito com o uso de **setup dependente da sequência**, no qual, para cada par de tarefas  $(i, j)$ , existe um tempo mínimo  $s_{km}$  entre o tempo de completude de  $k$  e o de início de  $m$ . Como as distâncias de canais são entre os valores alocados para os canais (ou seja, os tempos de início das tarefas associadas às chamadas) e o tempo de processamento das tarefas é 1, tem-se que  $s_{km} = d_{ij} - 1$ . Por fim, a minimização da faixa utilizada do espectro será a minimização da faixa de tempo usada ( $C_{max}$ ) menos 1 (devido ao fato de que  $C_{max}$  conta o tempo de completude da última tarefa - como todos os tempos de processamento são 1, o maior canal usado será  $C_{max} - 1$ ). Dadas estas características,



**Figura 4.2:** Exemplo de correlação entre escalonamento de tarefas e coloração em grafos.



**Figura 4.3:** Problema de alocação de canais da figura 3.7 modelado como escalonamento de tarefas  $P \mid p_k = 1, r_k, \bar{d}_k, s_{km} \mid C_{max}$ . Cada chamada é tratada como uma tarefa distinta. As datas de disponibilidade, os prazos máximos de término e os tempos de *setup* são dados pela tabela. O gráfico de Gantt para o escalonamento correspondente à alocação de canais é dado abaixo da tabela, onde os números  $x, y$  dentro dos slots de tempo indicam, respectivamente, a estação e qual chamada recebeu o instante de tempo associado ao canal.

pode-se propor o modelo de escalonamento  $P \mid p_k = 1, r_k, \bar{d}_k, s_{km} \mid C_{max}$ .

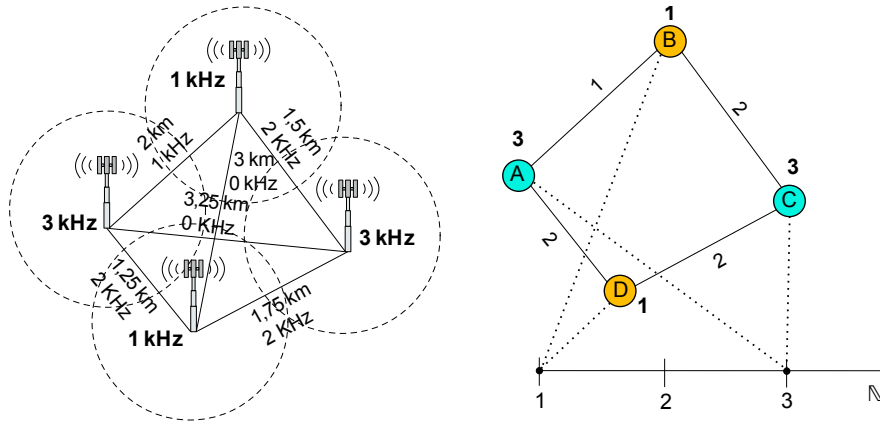
Este problema também pode ser modelado como um RCPS, considerando as mesmas características, no entanto, no lugar de *setup*, tem-se *time lags*. Como existem precedências, e o grafo pode ser cíclico, para modelar as distâncias de reuso, cada aresta será duplicada, sendo uma em cada direção, e o peso de cada uma (que serão os *time lags*  $\ell_{km}$  e  $\ell_{mk}$ ) será igual à distância de reuso do Min-Max-CAP.

Pode-se visualizar graficamente a solução para o modelo de escalonamento por meio de um **Gráfico de Gantt**, que consiste em um gráfico bidimensional com um eixo representando as máquinas e o outro representando o tempo. Os valores dentro das células (os *slots* de tempo) indicam a tarefa (ou seja, uma chamada) que estará ocupando tal máquina naquele instante. Para o exemplo da figura 3.7, o gráfico de Gantt associado ao modelo de escalonamento pode ser visto na figura 4.3.

### 4.3 Modelagem em geometria de distâncias

Uma característica encontrada nas instâncias dos problemas de alocação de canais foi que a separação exigida entre canais alocados a ERBs distintas é inversamente proporcional à distância geográfica entre as mesmas. Isso motivou o estudo da correlação das colorações





**Figura 4.4:** Exemplo de alocação de canais respeitando restrições de distâncias. Na imagem à esquerda, entre duas ERBs, se a distância geográfica entre as mesmas estiver entre 0 e 2km, os canais (em kHz) atribuídos a elas devem ter uma diferença absoluta de pelo menos 2. Se a distância geográfica estiver entre 2 e 3km, os canais devem apenas ser diferentes e se a distância for maior que 3km, as ERBs podem usar os mesmos canais. As distâncias (geográfica e de separação de canais) são dadas em cada aresta. O número em negrito é o canal alocado para a ERB. A imagem à direita mostra a rede como um grafo não direcionado.

especiais em grafos com o problema de **geometria de distâncias** (GD), no qual são dadas as distâncias entre os pontos e deve-se encontrar as posições dos mesmos no espaço, sendo uma aplicação comum a determinação de estrutura de proteínas. Um estudo da relação entre distâncias geográficas e separação de canais já existia na literatura [43], porém, sem a caracterização com geometria de distâncias. Um exemplo de relação entre distância geográfica e separação de canais é dado na Figura 4.4.

O ponto de partida dos modelos desenvolvidos é o Problema de Coloração Generalizada em Grafos (GCP ou  $T$ -coloring), restringindo os conjuntos proibidos a valores consecutivos. Uma forma de visualizar uma coloração em grafos consiste em considerar as cores a serem atribuídas aos vértices como pontos na reta que serão posicionados nos vértices do grafo. Desta forma, baseando-se na definição do problema básico de GD, define-se o **Problema de Coloração em Geometria de Distâncias** (*Coloring Distance Geometry Problem* - CDGP): dado um grafo não direcionado  $G = (V, E, d)$ , onde  $d : E \rightarrow \mathbb{Z}_+$ , deve-se encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que  $\|x_i - x_j\| = d_{ij}$  para cada  $(i, j) \in E$ . O objetivo será a minimização da maior cor utilizada, ou seja,  $\max_{i \in V} x_i$ . No caso de  $d_{ij} = 1$  ( $\forall (i, j) \in E$ ), tem-se o problema de coloração clássico. É interessante verificar que, se o grafo for completo, a coloração clássica é resolvida em tempo linear, já que todos os vértices terão cores distintas. O problema de GD envolvendo um grafo completo também é resolvido em tempo linear [33], o que reforça a correlação.

Como já citado em capítulos anteriores, no contexto de alocação de canais, a rela-

ção de igualdade nas distâncias indica que a comunicação realizada é bidirecional. No caso de comunicação unidirecional, as distâncias dadas serão limites inferiores para a diferença entre canais (ou seja, cores no problema de coloração correspondente) alocados para vértices vizinhos. No **Problema de Coloração em Geometria de Distâncias Mí-nimas** (*Coloring Min-Distance Geometry Problem - CMDGP*), é fornecido um grafo não direcionado  $G = (V, E, d)$ , onde  $d : E \rightarrow \mathbb{Z}_+$ , e deve-se encontrar a projeção  $x : V \rightarrow \mathbb{N}$  (tal que  $|x_i - x_j| \geq d_{ij}$  para cada  $(i, j) \in E$ ) cujo valor da maior cor seja o menor possível.

Dado que em uma rede sem fio podem existir comunicações tanto unidirecionais quanto bidirecionais, pode-se definir um modelo mais geral para este cenário. No **Problema de Coloração em Geometria de Distâncias Mistas** (*Mixed Coloring Distance Geometry Problem - MCDGP*), tem-se um grafo não direcionado  $G = (V, E, d, f)$ , onde  $d : E \rightarrow \mathbb{Z}_+$  e  $f : E \rightarrow \{0, 1\}$ , deve-se encontrar uma projeção  $x : V \rightarrow \mathbb{N}$  tal que, para cada  $(i, j) \in E$ , se, e somente se,  $f(i, j) = 0$ , então  $|x_i - x_j| \geq d_{ij}$ , e se, e somente se,  $f(i, j) = 1$ , então  $|x_i - x_j| = d_{ij}$ , onde o valor da maior cor utilizada deve ser o menor possível.

É possível adicionar restrições de listas de cores e demandas ao modelo MCDGP de forma a permitir seu uso em cenários de alocação de canais com domínios de frequência e múltiplas chamadas por célula. Nesta situação, denominada de **Problema de Lista-Multicoloração em Geometria de Distâncias Mistas** (*Mixed List-Multicoloring Distance Geometry Problem - MCDGP-Multi-List*), é dado um grafo não direcionado  $G = (V, E, d, f, c, l)$ , onde  $d : E \rightarrow \mathbb{Z}_+$ ,  $f : E \rightarrow \{0, 1\}$ ,  $c$  é o conjunto de demandas de cores dos vértices e  $l$  é a lista de cores possíveis para cada vértice. Deve-se encontrar uma projeção  $x$  do grafo em  $\mathbb{N}$  onde, para cada vértice  $i$ , são atribuídos  $c_i$  pontos de cores (ou seja,  $x_i$  é um conjunto de  $c_i$  elementos) tais que, para todo  $k = 1, \dots, c_i$ , tem-se que  $x_{ik} \in l(i)$  e para cada  $(i, j) \in E$ , tem-se que para todo  $k = 1, \dots, c_i$  e  $m = 1, \dots, c_j$ , se, e somente se,  $f(i, j) = 0$ , então  $|x_{ik} - x_{jm}| \geq d_{ij}$ , e se, e somente se,  $f(i, j) = 1$ , então  $|x_{ik} - x_{jm}| = d_{ij}$ , onde o valor da maior cor utilizada deve ser o menor possível.

## 4.4 Notas do capítulo

Com os modelos propostos para representação dos problemas estudados, foi possível analisar o impacto de características de diferentes cenários de alocação de canais na modelagem e solução dos problemas.

Para as colorações especiais em grafos, o uso de modelos contendo limites inferior e superior das cores dos vértices para a alocação de canais não é comum, ao contrário do uso de listas para representação do domínio da frequência. Uma vez que a complexidade computacional do problema original envolvendo limites ( $(\gamma, \mu)$ -coloração) ainda está em aberto para alguns tipos de grafos [19], existe a possibilidade de obter algoritmos eficientes para certos casos específicos de problemas de alocação de canais a partir dos modelos propostos. Com o uso de modelos com multicoloração, torna-se importante o uso de métodos que evitem a replicação explícita dos vértices do grafo em pequenas cliques, o que aumenta consideravelmente o tempo de execução dos mesmos. Técnicas de programação inteira já foram aplicadas com sucesso nesse tipo de modelo [61].

Quanto aos modelos em escalonamento de tarefas, conforme já citado, não foram encontrados trabalhos envolvendo correlações diretas dessa classe de problemas com alocação de canais. Com os modelos propostos, existe a possibilidade de adaptação do vasto ferramental de escalonamento nos problemas estudados nesta dissertação, em especial, envolvendo características específicas como tempos de preparação e tempos de processamento iguais. Também foram identificados certos casos cuja complexidade é polinomial em escalonamento, por exemplo, quando todas as separações entre canais são iguais, que equivalem ao modelo de escalonamento em máquinas paralelas idênticas com tempos de processamento iguais e minimização do *makespan* [12].

Os modelos em geometria de distâncias foram motivados pelo fato de que a separação entre canais de diferentes chamadas também é uma distância, cuja métrica é bem definida. Apesar da relação entre as distância geográficas entre ERBs e separações entre canais já ser conhecida na literatura, não foram identificados trabalhos que explorassem tais características, além utilizando geometria de distâncias, o que foi também uma forma de mostrar uma aplicação adicional dessa classe de problemas (alocação de canais e colorações em grafos).

# Capítulo 5

## Algoritmos desenvolvidos e implementados

Neste capítulo, serão mostrados os métodos algorítmicos utilizados para a solução do problema, sendo dois meta-heurísticos e dois enumerativos, seguindo diferentes abordagens.

### 5.1 Meta-heurísticas baseadas em busca local

Algoritmos baseados em técnicas de busca local foram desenvolvidos para a alocação de canais, mais especificamente os problemas das instâncias *Philadelphia*, *Helsinki* e as artificiais de 55 células, ou seja, as células terão demandas distintas e maiores que 1. Os métodos foram baseados no modelo de coloração proposto considerando pesos nas arestas e, em escalonamento, envolvendo tempos de preparação distintos.

Para o problema de alocação de canais, uma solução pode ser codificada como uma sequência de pares  $(v, k)$ , onde  $v$  é uma célula e  $k$  é o índice da chamada. Os canais serão alocados de acordo com a ordem de ocorrência na sequência. Uma sequência inicial de chamadas para a inicialização é obtida ordenando as células em ordem decrescente de demandas de canais, onde as chamadas das células serão todas adicionadas na frente da sequência. O Algoritmo 5.1 fornece o pseudocódigo para isso.

Para alocar os canais de acordo com a sequência de chamadas, foi utilizada a regra de **alocação exaustiva por frequência** (*frequency exhaustive allocation* - FEA) [84, 13], que consiste em atribuir o menor canal que não viole restrições de interferência para a chamada e cujo pseudocódigo é dado pelo Algoritmo 5.2. Tal estratégia funciona como

exposto a seguir. Inicialmente, nas linhas 2 a 4, são criadas estruturas auxiliares para a alocação de canais propriamente dita, sendo elas, respectivamente, um vetor que conterá o número de canais já alocados para cada estação, a matriz de canais alocados e uma variável que será o *span* da alocação.

O canal candidato é obtido da seguinte maneira:

1. O primeiro candidato é o menor canal possível, de valor 1;
2. Se a ERB usada pela chamada possuir outras chamadas com canais já alocados, o menor canal possível é incrementado em  $\lambda d_{ii}$ , onde  $\lambda$  é o número de chamadas já com canais e  $d_{ii}$  é a separação de canais entre chamadas na mesma célula;
3. Finalmente, é feita uma verificação para garantir que o canal candidato não viole restrições de interferência com os canais atribuídos a células adjacentes. Se alguma violação for encontrada, o canal candidato é incrementado em 1 e a verificação é repetida. Quando nenhuma violação for encontrada, o canal candidato é o mínimo alocável para a chamada.

O pseudocódigo para a regra FEA é dado no Algoritmo 5.2. Um exemplo de sequência de chamadas e a alocação de canais correspondente utilizando a instância fornecida pela Tabela 5.1 (matriz de interferências e demandas) é mostrado na Figura 5.1.

Para modificar a solução, uma vizinhança  $N : S \rightarrow 2^S$  com movimentos de troca entre pares é usada. Dada uma sequência  $S$  definindo uma solução, uma nova sequência  $S'$  é derivada efetuando a troca entre duas posições de chamadas em  $S$ . A Figura 5.2 exemplifica o processo, onde a sequência da Figura 5.1 é modificada trocando as posições das duas

---

**Algoritmo 5.1** Geração da sequência inicial de chamadas.

---

**Entrada:** conjunto de ERBs  $V$ , conjunto de arestas  $E$ , matriz de distâncias entre estações  $d$  e conjunto de demandas de canais  $q$ .

- 1: **Função** GERA INICIAL( $rede = (V, E, d, q)$ )
  - 2:      $S \leftarrow$  sequência de chamadas (inicialmente vazia)
  - 3:      $V' \leftarrow V$
  - 4:     **enquanto**  $V' \neq \emptyset$  **faça**
  - 5:          $i \leftarrow$  elemento de  $Q$  de maior demanda
  - 6:         **para**  $k \leftarrow 1$  **até**  $i.numChamadas$  **faça**
  - 7:             Adicionar a  $k$ -ésima chamada de  $i$  à frente de  $S$
  - 8:          $V' \leftarrow V' - \{i\}$
  - 9:     **retorne**  $S$
-

---

**Algoritmo 5.2 FEA: Alocação exaustiva por frequência**

---

**Entrada:** conjunto de ERBs  $V$ , conjunto de arestas  $E$ , matriz de distâncias entre estações  $d$ , conjunto de demandas de canais  $q$  e sequência  $S$  de chamadas.

```
1: Procedimento ALOCACAOFEA( $rede = (V, E, d, q), S$ )
2:    $numCanaisAlocados \leftarrow$  vetor de  $numCelulas$  elementos inicializados com 0
3:    $canaisAlocados \leftarrow$  matriz com  $i$  linhas, cada uma com  $q_i$  elementos
4:    $maiorCanal(S) \leftarrow 0$ 
5:   para  $c$  de 1 até  $numChamadas$  faça
6:      $i \leftarrow celula(S[c])$ 
7:      $canalCandidato \leftarrow ((q_i - numCanaisAlocados[i]) * d_{ii}) + 1$ 
8:      $flagOK \leftarrow$  falso
9:     enquanto não  $flagOK$  faça
10:       $flagOK \leftarrow$  verdadeiro
11:      para cada célula  $j$  adjacente a  $i$  faça
12:        para  $k$  from 1 até  $numCanaisAlocados[k]$  faça
13:          se  $|canalCandidato - canaisAlocados[j][k]| < d_{ij}$  então
14:             $flagOK \leftarrow$  verdadeiro
15:            interrompa
16:          se não  $flagOK$  então
17:             $canalCandidato \leftarrow canalCandidato + 1$ 
18:            interrompa
19:       $numCanaisAlocados[i] \leftarrow numCanaisAlocados[i] + 1$ 
20:       $canaisAlocados[i][numCanaisAlocados[i]] \leftarrow canalCandidato$ 
21:      se  $canalCandidato > maiorCanal(S)$  então
22:         $maiorCanal(S) \leftarrow canalCandidato$ 
```

---

**Tabela 5.1:** Instância de 6 células a ser usada como exemplo para a aplicação do *simulated annealing*.

Matriz de interferências	Demandas
5 2 1 1 0 0	1, 1, 2, 1, 3, 2
2 5 2 2 1 0	
1 2 5 2 2 1	
1 2 2 5 2 1	
0 1 2 2 5 2	
0 0 1 1 2 5	

chamadas marcadas, o que, após o uso da regra FEA, fornece uma alocação de menor custo em relação à sequência original.

### 5.1.1 Busca local iterada

O algoritmo de busca local iterada parte da solução inicial fornecida pelo Algoritmo 5.1. É feita uma busca local em tal solução utilizando a vizinhança de trocas entre pares na sequência de chamadas, sendo que, a cada troca realizada, o algoritmo de decodificação FEA é aplicado para determinar o custo da solução vizinha. Para determinação do ótimo local, foi utilizada a estratégia de melhor melhoria (*best improvement*), consistindo em determinar a melhor solução na vizinhança. Para efetuar a perturbação de um ótimo local, é utilizada uma quantidade de trocas entre pares na solução, sem levar em conta o impacto na função objetivo, sendo que o número de trocas pode ser variável de acordo com o tamanho da instância. O Algoritmo 5.3 fornece o pseudocódigo do processo.

5,1	3,1	1,1	3,2	6,2	6,1	4,1	5,2	2,1	5,3
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Célula	Canal(is)
1	1
2	10
3	3, 8
4	5
5	1, 11, 16
6	4, 9

**Figura 5.1:** Exemplo de sequência de chamadas e a alocação de canais correspondente obtida pelo algoritmo FEA.

---

**Algoritmo 5.3** Busca local iterada para alocação de canais

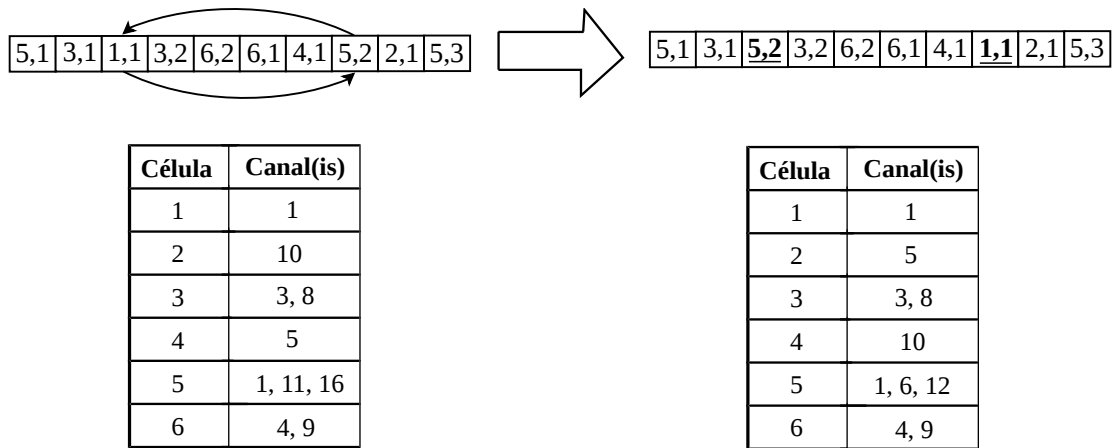
---

**Entrada:** conjunto de ERBs  $V$ , conjunto de arestas  $E$ , matriz de distâncias entre estações  $d$ , conjunto de demandas de canais  $q$ , sequência inicial  $S$  de chamadas, número máximo de iterações  $maxIter$  e número de perturbações  $numPert$ .

```
1: Função BUSCALOCALITERADA( $rede = (V, E, d, q)$ ,  $S$ ,  $maxIter$ ,  $numPert$ )
2:   para  $k$  de 1 até  $maxIter$  faça
3:      $R \leftarrow S$ 
4:      $melhorou \leftarrow$  verdadeiro
5:     // Busca local na vizinhança de trocas entre pares
6:     enquanto  $melhorou =$  verdadeiro faça
7:        $melhorou \leftarrow$  falso
8:       para  $i$  de 1 até  $|V|$  faça
9:         para  $j$  de  $i + 1$  até  $|V|$  faça
10:           Trocar as chamadas das posições  $i$  e  $j$  em  $S$ 
11:           ALOCACAOFEA( $rede, R$ )  $\triangleright$  Alocar canais usando a regra FEA
12:           se  $maiorCanal(R) < maiorCanal(S)$  então
13:              $S \leftarrow R$ 
14:              $melhorou \leftarrow$  verdadeiro
15:           Trocar as chamadas das posições  $j$  e  $i$  em  $S$ 
16:         para  $p$  de 1 até  $numPert$  faça
17:            $i \leftarrow$  posição aleatória das chamadas
18:            $j \leftarrow$  posição aleatória das chamadas
19:           Trocar as chamadas das posições  $i$  e  $j$  em  $S$ 
20:   retorne  $S$ 
```

---





**Figura 5.2:** Operação de troca em uma sequência de chamadas. A sequência original possui uma alocação de custo 16. Após trocar os elementos (5,2) and (1,1), é obtida uma nova alocação de custo 12, que é melhor.

### 5.1.2 *Simulated annealing*

O algoritmo *simulated annealing* proposto funciona da seguinte forma: partindo de uma sequência inicial  $S$  e uma temperatura inicial  $t_0$ , é feita uma cópia de  $S$  para uma outra sequência  $R$  que será modificada (e  $S$  será considerada a melhor solução obtida na temperatura). Enquanto o equilíbrio termodinâmico não for atingido para a temperatura, duas posições aleatórias na sequência  $R$  são escolhidas e trocadas de posição entre si. O algoritmo FEA é aplicado e, então:

- Se o maior canal utilizado para a nova sequência for menor que o usado na sequência antiga, então  $R$  é copiado para  $S$ ;
- Caso contrário, com probabilidade  $e^{(-\Delta/t)}$  (onde  $\Delta$  é a diferença entre o maior canal utilizado em  $R$  e em  $S$ ),  $R$  é copiado para  $S$ .

Quando o equilíbrio termodinâmico é atingido, a solução  $S$  é comparada à melhor obtida durante todo o processo até o momento. Se o custo de  $S$  for menor, então ela se tornará a nova melhor solução corrente. A temperatura então é decrementada. No algoritmo proposto, foi utilizado o esquema de resfriamento **linear**. Além disso, foi considerado que o equilíbrio termodinâmico é atingido após um número fixo  $L$  de iterações a cada temperatura. Esta abordagem, apesar de simples, teve bons resultados. O pseudocódigo desta rotina é dado pelo Algoritmo 5.4.

---

**Algoritmo 5.4** *Simulated annealing* para alocação de canais

---

**Entrada:** conjunto de ERBs  $V$ , conjunto de arestas  $E$ , matriz de distâncias entre estações  $d$ , conjunto de demandas de canais  $q$ , sequência inicial  $S$  de chamadas, temperatura inicial  $t_0$ , temperatura final  $t_F$ , fator de resfriamento  $\alpha$  e número de iterações  $L$  para o equilíbrio termodinâmico.

```
1: Função SIMULATEDANNEALING( $rede = (V, E, d, q)$ ,  $S$ ,  $t_0$ ,  $t_f$ ,  $\alpha$ ,  $L$ )
2:    $Melhor \leftarrow S$ 
3:    $R \leftarrow S$ 
4:    $t \leftarrow t_0$ 
5:   enquanto  $t > t_f$  faça
6:     para  $p$  de 1 até  $L$  faça
7:        $R \leftarrow R.NovoVizinho()$             $\triangleright$  Gerar novo vizinho aleatório
8:        $ALOCACAOFEA(rede, R)$             $\triangleright$  Alocar canais usando a regra FEA
9:        $\Delta \leftarrow R.maiorCanal - S.maiorCanal$ 
10:      se  $\Delta < 0$  então
11:         $S \leftarrow R$ 
12:      senão
13:         $r \leftarrow$  número real aleatório entre 0 e 1
14:        se  $r < e^{(-\Delta/t)}$  então
15:           $S \leftarrow R$ 
16:        se  $S.maiorCanal < Melhor.maiorCanal$  então
17:           $Melhor \leftarrow S$ 
18:       $t \leftarrow t - \alpha$ 
19:  retorne  $Melhor$ 
```

---

## 5.2 Métodos enumerativos exatos e aproximados

De forma a verificar a qualidade da solução obtida pelas meta-heurísticas baseadas em busca local, foi implementado um modelo matemático modificado do problema para obter soluções ótimas para as instâncias consideradas. Tal implementação aproveita os resultados obtidos pelos algoritmos anteriores e tenta melhorá-los utilizando recursos da ferramenta CPLEX.

Além disso, baseando-se na modelagem utilizando geometria de distâncias, foram desenvolvidas duas versões do método denominado de *branch-prune-and-bound*, que combina o método *branch-and-prune* utilizado em modelos de programação por restrições (onde a base, mais especificamente, foi o algoritmo proposto para o problema de geometria de distâncias moleculares [53]) com a técnica *branch-and-bound* para efetuar a otimização do objetivo considerado.

### 5.2.1 Branch-and-cut usando CPLEX

Para comparação dos resultados obtidos, foi implementado um modelo de programação inteira para o problema usando a ferramenta IBM/ILOG CPLEX.

O modelo utilizado foi baseado no de [49] para o critério de minimização da extensão usada, mostrado no Capítulo 3 e reproduzido abaixo:

$$\text{Minimizar } z_{max} - z_{min}$$

$$\text{Sujeito a } \sum_{f \in F_i} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \quad (5.1)$$

$$x_{if} + x_{jg} \leq 1 \quad (\forall i, j = 1, 2, \dots, n; f \in F_i; g \in F_j : |f - g| < d_{ij}) \quad (5.2)$$

$$x_{if} \leq y_f \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (5.3)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F_i) \quad (5.4)$$

$$y_f \in \{0, 1\} \quad (\forall f \in F) \quad (5.5)$$

$$z_{max} \geq f y_f \quad (\forall f \in F) \quad (5.6)$$

$$z_{min} \leq f_{max} - (f_{max} - f) y_f \quad (\forall f \in F) \quad (5.7)$$

$$z_{max}, z_{min} \in \mathbb{Z}_+ \quad (5.8)$$

Para o problema específico a ser resolvido, vale observar que, como os domínios de frequência são todos iguais (ou seja,  $F_1 = F_2 = \dots = F$ ), o primeiro canal a ser usado será sempre 1 (o primeiro disponível). Assim, o critério de otimização é equivalente à minimização do maior canal dentre todos os usados, permitindo a remoção da variável  $z_{min}$  e do conjunto de restrições (7). Outra observação que pode ser feita é que o conjunto de variáveis  $y_f$  é redundante, uma vez que é possível descobrir qual o maior canal usado a partir das variáveis de valor 1 do tipo  $x_{if}$ . Dessa forma, pode-se eliminar as variáveis  $y_f$  e também o conjunto de restrições (3), sendo necessário alterar as restrições (6) para uso das variáveis  $x_{if}$ .

O modelo modificado, utilizado na implementação, fica como mostrado a seguir:

$$\begin{aligned} &\text{Minimizar} && z_{max} \\ &\text{Sujeito a} && \sum_{f \in F} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \end{aligned} \quad (5.1)$$

$$x_{if} + x_{jg} \leq 1 \quad (\forall i, j = 1, 2, \dots, n; f, g \in F : |f - g| < d_{ij}) \quad (5.2)$$

$$z_{max} \geq f x_{if} \quad (\forall i = 1, 2, \dots, n; f \in F) \quad (5.3)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F) \quad (5.4)$$

$$z_{max} \in \mathbb{Z}_+ \quad (5.5)$$

Para fazer uso do modelo, é necessário ter um limite superior para o valor do maior canal a ser usado, uma vez que não há um conjunto pré-definido de listas de canais para as ERBs. Para isso, o método foi alimentado com soluções obtidas pelo *simulated annealing*, utilizando os *MIP starts* do CPLEX, de forma que o processo de *branch-and-cut* é iniciado já com soluções incumbentes válidas, diminuindo o tempo gasto no processo de busca das soluções.

### 5.2.2 *Branch-prune-and-bound*

A partir dos estudos das correlações dos problemas de alocações de canais com colorações em grafos e também destes últimos com geometria de distâncias, vistos no capítulo anterior, foi desenvolvido um algoritmo combinando o método *branch-and-prune* utilizadas no Problema de Geometria de Distâncias Moleculares Discretizáveis [53] e técnicas clássicas de *branch-and-bound*. O método proposto consiste em uma enumeração implí-

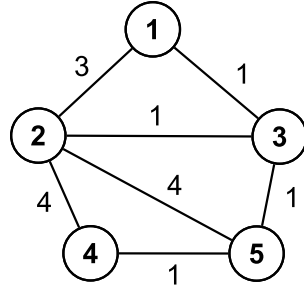
cita do espaço de busca de soluções, colorindo cada vértice considerando as distâncias aos seus vizinhos, atribuindo a menor cor possível que respeite tais distâncias, onde uma solução parcial pode ser descartada se for inactível para o problema ou se não puder ser melhor que a atual melhor solução.

O algoritmo funciona como enunciado a seguir. Primeiro, um vértice  $i$  cujas demandas não foram satisfeitas é selecionado como ponto de partida. Ele recebe a cor 1, a menor disponível. Em seguida, escolhe-se um vizinho  $j$  de  $i$  (podendo ser o próprio  $i$  com demandas maiores que um) e o mesmo terá até duas possibilidades de cores:  $cor_{if} + d_{ij}$  e  $cor_{if} - d_{ij}$  (onde  $f$  é o índice da última cor atribuída a  $i$ ), sendo esta última utilizada apenas se for maior ou igual a 1 (ou seja,  $cor_{if} \geq d_{ij} + 1$ ). O método é então recursivamente aplicado aos vizinhos de  $j$ . Se, em um determinado momento, todos os vizinhos estão coloridos, um outro vértice dentre todos não coloridos é selecionado e pintado com a cor 1, e daí o processo recursivo continua. Quando todos os vértices estiverem coloridos, tem-se uma solução completa válida para o problema. Para garantir a total exploração do espaço de soluções, após exaurir as soluções possíveis de serem obtidas partindo do vértice  $i$ , um outro vértice é selecionado como inicial e o processo é repetido.

A cada vez que um vértice é colorido, são feitos dois testes. O primeiro, chamado de teste de factibilidade, consiste em verificar se, para cada par de vértices  $i$  e  $j$ , a condição  $||cor_{if} - cor_{jg}| - d_{ij}| \geq 0$  (equivalentemente,  $|cor_{if} - cor_{jg}| \geq d_{ij}$ , pois todas as distâncias são inteiros não negativos) seja satisfeita. Caso não seja, a solução corrente é descartada por **pruning**. O segundo consiste em verificar se a maior cor usada (a função objetivo do problema, que deve ser minimizada) da coloração parcial é maior ou igual a um limite superior (equivalente a maior cor usada da melhor solução já encontrada). Em caso positivo, a solução é descartada por **bounding**. Limites inferiores são gerados de acordo com a árvore geradora mínima (AGM) do grafo de entrada, onde a coloração parcial é aplicada à AGM e estendida de forma a colorir os vértices restantes do grafo. O Teorema 5.1 a seguir demonstra a validade desta abordagem.

**Teorema 5.1.** *A maior cor usada na coloração da árvore geradora mínima de um grafo  $G$  fornece um limite inferior para a coloração de  $G$ .*

*Prova.* Para o problema de coloração de vértices considerando distâncias, tem-se a seguinte formulação de programação inteira (a mesma utilizada no *branch-and-cut* via



**Figura 5.3:** Grafo a ser usado como exemplo para o *branch-prune-and-bound*.

CPLEX mostrada anteriormente):

Minimizar  $z_{max}$

$$\text{Sujeito a } \sum_{f \in F} x_{if} = q_i \quad (\forall i = 1, 2, \dots, n) \quad (5.1)$$

$$x_{if} + x_{jg} \leq 1 \quad (\forall i, j = 1, 2, \dots, n; f, g \in F : |f - g| < d_{ij}) \quad (5.2)$$

$$z_{max} \geq f x_{if} \quad (\forall i = 1, 2, \dots, n; f \in F) \quad (5.3)$$

$$x_{if} \in \{0, 1\} \quad (\forall i = 1, 2, \dots, n; f \in F) \quad (5.4)$$

$$z_{max} \in \mathbb{Z}_+ \quad (5.5)$$

Onde  $F$  é o conjunto de cores disponíveis, a variável  $x_{if}$  tem valor 1 se  $i$  recebeu a cor  $f$  e 0 caso contrário, e a variável  $z_{max}$  terá o valor da maior cor usada (ou seja, o *span*). O conjunto de restrições (2) define pelo menos uma restrição para cada aresta do grafo (será exatamente uma se  $d_{ij} = 1$  - mas, nos problemas considerados,  $d_{ij}$  pode ser maior que 1, portanto, haverá mais restrições definidas). A quantidade de arestas da AGM de  $G$  é menor ou igual à quantidade de arestas de  $G$ , portanto, o modelo para coloração da AGM terá no máximo a quantidade de restrições do modelo para  $G$ . Assim, o modelo da AGM é uma relaxação em relação a  $G$ , e como o problema é de minimização, tem-se então que o valor da maior cor usada na AGM fornece um limite inferior para a coloração de  $G$ .  $\square$

Os Algoritmos 5.5 e 5.6 fornecem, respectivamente o pseudocódigo do método *branch-prune-and-bound* e de seu teste de factibilidade. A Figura 5.4 exemplifica o algoritmo aplicado ao grafo da Figura 5.3.

Uma segunda versão do algoritmo também foi implementada, uma vez que, em alguns casos, o método não atingiu valores ótimos. Esta versão envolve considerar distâncias

---

**Algoritmo 5.5** *Branch-Prune-and-Bound* - primeira versão

---

**Entrada:** grafo  $G$  (com conjunto de vértices  $V$ , conjunto de arestas  $E$  e matriz de distâncias entre vértices  $d$ ), vértice atual  $i$  a ser colorido, vértice anterior  $j$ , coloração atual  $col$ , melhor coloração já encontrada  $colMelhor$  e limite superior  $ls$ .

```
1: Função BPB-1( $G = (V, E, d), i, j, col, colMelhor, ls$ )
2:    $li \leftarrow \text{LIMITEINFERIORAGM}(\text{AGM}(G), col)$ 
3:   se  $li > ls$  então
4:     retorne ▷ Poda por bounding
5:    $numCoresTestadas = 0$ 
6:   enquanto  $numCoresTestadas < 2$  faça
7:     se  $j = -1$  então
8:        $cor \leftarrow 1$ 
9:        $numCoresTestadas = 2$ 
10:    senão
11:      se  $numCoresTestadas = 0$  então
12:         $cor \leftarrow col[j] + d_{ij}$ 
13:      senão
14:         $cor \leftarrow col[j] - d_{ij}$ 
15:        se  $cor < 1$  então
16:          retorne ▷ Cor inválida, deve ser desconsiderada
17:           $numCoresTestadas = numCoresTestadas + 1$ 
18:           $col[i] \leftarrow cor$ 
19:          se maior cor usada (span) em  $col$  for menor que  $cor$  então
20:            Span de  $col$  passa a ser  $cor$ 
21:          se  $cor \geq ls$  então
22:            Remove cor de  $i$ 
23:            retorne ▷ Poda por bounding
24:          se  $\text{TesteFactibilidade}(G, col, i) = \text{falso}$  então
25:            Remove cor de  $i$ 
26:            retorne ▷ Viola distâncias, poda por pruning
27:          se todos os vértices foram completamente coloridos então
28:            se Span de  $col < span$  de  $colMelhor$  então
29:               $colMelhor \leftarrow col$ 
30:            senão
31:               $temVizinho \leftarrow \text{falso}$ 
32:              para cada vértice  $k$  vizinho de  $i$  faça
33:                se  $k$  não estiver colorido então
34:                   $temVizinho \leftarrow \text{verdadeiro}$ 
35:                  BPB-1( $G, k, i, col, colMelhor, ls$ )
36:                se  $temVizinho = \text{falso}$  então
37:                  para cada vértice  $k$  de  $G$  faça
38:                    se  $k$  não estiver colorido então
39:                      BPB-1( $G, k, -1, col, colMelhor, ls$ )
40:                  Remove cor de  $i$ 
41:            retorne  $colMelhor$ 
```

---

---

**Algoritmo 5.6** Teste de factibilidade de uma coloração (parcial ou completa) a partir de um vértice.

---

**Entrada:** grafo  $G$  (com conjunto de vértices  $V$ , conjunto de arestas  $E$  e matriz de distâncias entre vértices  $d$ ), vértice  $i$  a ser considerado e coloração atual  $col$ .

```
1: Função TESTEFACTIBILIDADE( $G = (V, E, d), i, col$ )
2:   para cada vértice  $j$  vizinho de  $i$  faça
3:     se  $j$  estiver colorido então
4:       se  $|col[i] - col[j]| < d_{ij}$  então
5:         retorne falso
6:   retorne verdadeiro
```

---

para múltiplos vértices a cada vez ao invés de apenas entre o último vértice colorido e o atual. Nesse caso, a cor do vértice  $i$  é determinada da seguinte forma: um vetor  $somaCores$  de  $limiteSuperior$  elementos (todos inicializados com 0, e onde os índices são de 1 a  $limiteSuperior$ ), onde  $limiteSuperior$  é o valor da maior cor usada na melhor solução encontrada. Então, para cada vizinho  $j$  de  $i$  que já está colorido, deve-se resolver a inequação de valor absoluto  $|cor_j - cor_i| \geq d_{ij}$ . A inequação deve ser desmembrada em duas:

$$cor_j - cor_i \geq d_{ij} \quad \mathbf{e} \quad cor_j - cor_i \leq -d_{ij}$$

As soluções são dadas então por:

$$cor_i \geq cor_j + d_{ij} \quad \mathbf{e} \quad cor_i \leq cor_j - d_{ij}$$

Assim, no vetor  $somaCores$ , os elementos de 1 a  $d_{ij} - cor_j$  são incrementados caso  $d_{ij} - cor_j \geq 1$  e os elementos de  $d_{ij} + cor_j$  até  $limiteSuperior$  idem. A cor atribuída a  $i$  será então o menor índice  $k$  cujo valor  $somaCores[k]$  seja igual ao número de vizinhos coloridos de  $i$ . Se nenhum elemento de  $somaCores$  tiver esse valor, significa que, para colorir  $i$ , deve-se usar uma cor adicional, portanto, a solução parcial pode ser descartada por *bounding*. A Figura 5.5 exemplifica esse esquema.

Para o modelo mais geral de coloração com distâncias, o MCDGP-Multi-List, foi desenvolvida uma terceira versão do *branch-prune-and-bound*, cujo funcionamento é descrito a seguir. Primeiramente, é escolhido um vértice  $i$  cujas demandas ainda não foram completamente satisfeitas. Então, uma cor da lista  $\ell_i$  do vértice é escolhida e, se a mesma não violar nenhuma restrição de distância (ou seja, não estiver bloqueada no vértice), ela é adicionada à alocação atual. É realizada então uma propagação de restrições, de forma



---

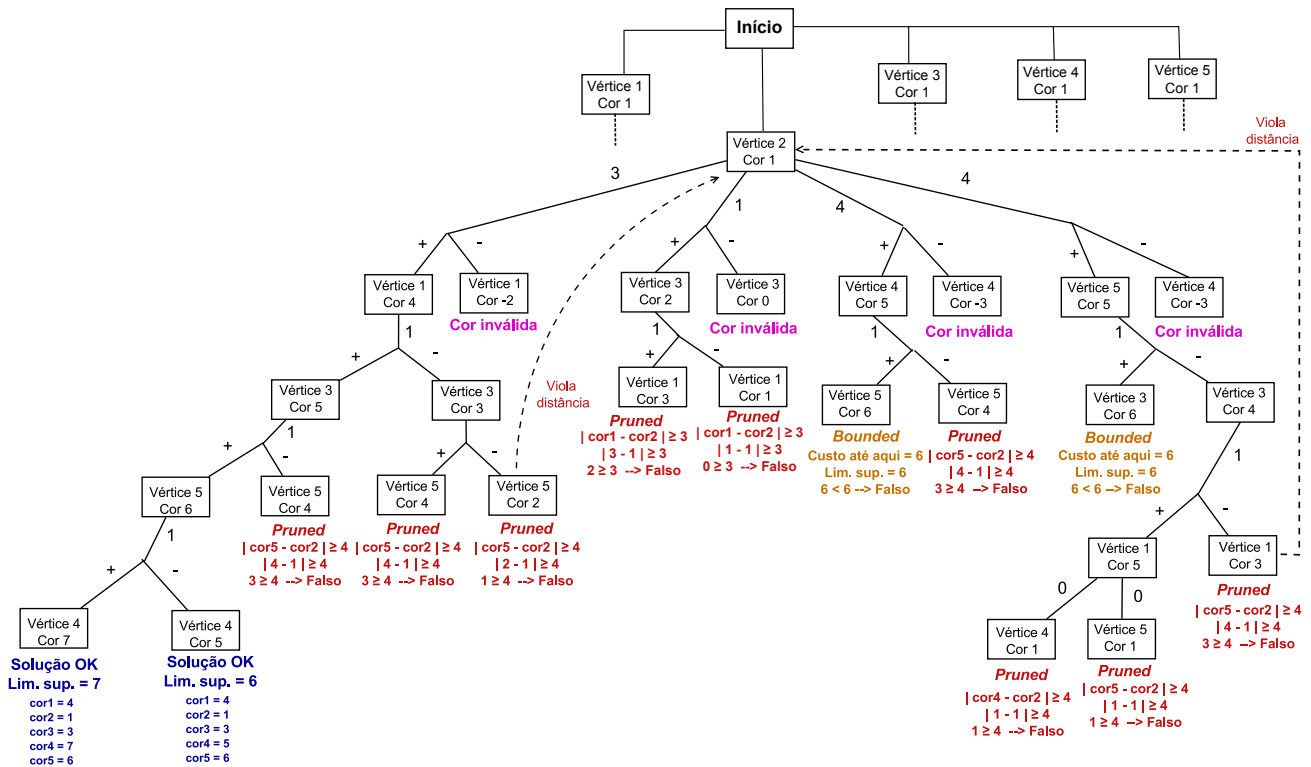
**Algoritmo 5.7** *Branch-Prune-and-Bound* - segunda versão

---

**Entrada:** grafo  $G$  (com conjunto de vértices  $V$ , conjunto de arestas  $E$  e matriz de distâncias entre vértices  $d$ ), vértice atual  $i$  a ser colorido, vértice anterior  $j$ , coloração atual  $col$ , melhor coloração já encontrada  $colMelhor$  e limite superior  $ls$ .

```
1: Função BPB-2( $G = (V, E, d), i, j, col, colMelhor, ls$ )
2:    $li \leftarrow \text{LIMITEINFERIORAGM}(\text{AGM}(G), col)$ 
3:   se  $li > ls$  então
4:     retorne ▷ Poda por bounding
5:    $cor \leftarrow \text{SELECIONARCOR}(G, i, coloracao, limiteSuperior)$ 
6:   se  $cor = -1$  então
7:     retorne ▷ Poda por bounding
8:    $col[i] \leftarrow cor$ 
9:   se maior cor usada (span) em  $col$  for menor que  $cor$  então
10:     $Span$  de  $col$  passa a ser  $cor$ 
11:   se  $cor \geq ls$  então
12:     Remover cor de  $i$ 
13:     retorne ▷ Poda por bounding
14:   se  $\text{TESTEFACTIBILIDADE}(G, col, i) = \text{falso}$  então
15:     Remover cor de  $i$ 
16:     retorne ▷ Viola distâncias, poda por pruning
17:   se todos os vértices foram completamente coloridos então
18:     se  $Span$  de  $col < span$  de  $colMelhor$  então
19:        $colMelhor \leftarrow col$ 
20:   senão
21:     para cada vértice  $k$  de  $G$  tal que  $k \neq i$  faça
22:       se  $k$  não estiver colorido então
23:         BPB-2( $G, k, -1, col, colMelhor, ls$ )
24:       Remover cor de  $i$ 
25:   retorne  $colMelhor$ 
```

---



**Figura 5.4:** Aplicação do *branch-prune-and-bound* para o grafo da Figura 5.3, partindo do vértice 2. A cada nó colorido, se o próximo vértice for adjacente, são consideradas duas possibilidades (somando ou subtraindo a distância do último vértice colorido), caso contrário, o próximo vértice deverá usar a cor 1. A ordem de visitação dos nós segue a busca em profundidade, onde são explorados primeiro os vizinhos de menor índice (ou vértices não-adjacentes de menor índice) e a primeira cor testada envolve a soma da distância, seguida da subtração da distância (ou apenas a cor 1, caso sejam não-adjacentes).

que, nos vértices vizinhos, as cores que violam distâncias são bloqueadas. Se todas as cores da lista estiverem bloqueadas, o nó é podado (caso denominado de *pruning*), caso contrário, o valor da maior cor utilizada na alocação atual é verificado. Se for maior ou igual ao custo da melhor solução já encontrada (um limite superior para a exploração, denotado por  $U$ , inicialmente  $\infty$ , e vai decrescendo a medida que soluções melhores são encontradas), o nó é podado (situação de *bounding*), caso contrário, o método é recursivamente aplicado. Quanto todas as demandas são satisfeitas, tem-se uma solução viável completa. O algoritmo então remove a última cor utilizada na alocação e explora outras opções.

Para verificar quais cores estão bloqueadas em um vértice, utiliza-se uma matriz  $Blq$ , com todos os elementos inicializados com 0, onde  $Blq_{ik} > 0$  se a cor  $k$  estiver bloqueada em  $i$  e  $Blq_{ik} = 0$  caso contrário. Quanto a execução do algoritmo é terminada, tem-se a solução ótima da instância. O pseudocódigo do procedimento é dado no Algoritmo 5.9.

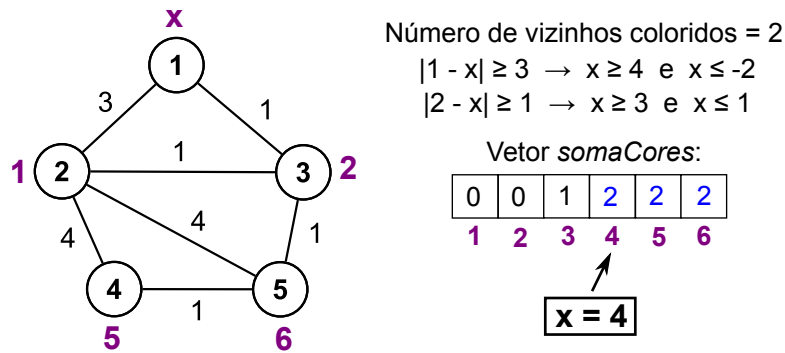


Figura 5.5: Esquema de seleção de cores baseado na verificação de todas as distâncias aos vizinhos.

### 5.3 Notas do capítulo

Os algoritmos desenvolvidos neste trabalho foram baseados em técnicas clássicas de otimização, além de métodos eficazes em outros problemas cujas correlações foram discutidas no capítulo anterior.

Dentre as meta-heurísticas baseadas em busca local, durante testes preliminares com instâncias da literatura, constatou-se que o tempo computacional da busca local iterada utilizada não era competitivo, sendo alto para instâncias de médio porte. Tal característica é justificada pelo fato de que, apesar de o número de células nas redes consideradas ser baixo (21 células nas instâncias mais simples), as demandas de canais em cada uma tendem a ser grandes, sendo que a representação utilizada é impactada pela quantidade total de chamadas da rede. Como a busca local iterada efetua todas as trocas entre pares de chamadas, o processo tende a usar bastante tempo computacional. Em contrapartida, o *simulated annealing* não apresentou essa desvantagem, uma vez que não há uma exploração exaustiva de vizinhança, mas apenas a geração de um vizinho aleatório. Devido a isso, não foram realizados experimentos massivos com a busca local iterada.

Quanto aos métodos enumerativos, o *branch-and-cut* via CPLEX não foi utilizado apenas para resolver o problema de maneira exata, mas também para verificar o impacto dos algoritmos internos da ferramenta partindo a partir das soluções obtidas pelo *simulated annealing*, de forma a melhorá-las quando possível, o que gerou também a possibilidade de explorar mais características do resolvidor, tais como os *MIP starts*. Já as duas versões do método *branch-prune-and-bound* possibilitaram analisar o impacto da aplicação das distâncias entre vértices na solução dos problemas. Levando em conta apenas a distância de um vértice a ser colorido ao seu antecessor, o algoritmo não apresenta comportamento exato em todos os casos, como ocorre também com o *branch-and-prune*

---

**Algoritmo 5.8** Seleção de cor na segunda versão do *branch-prune-and-bound*

---

**Entrada:** grafo  $G$  (com conjunto de vértices  $V$ , conjunto de arestas  $E$  e matriz de distâncias entre vértices  $d$ ), vértice  $i$  a ser considerado, coloração atual  $col$  e limite superior  $ls$ .

```
1: Função SELECIONARCOR( $G = (V, E, d), i, col, ls$ )
2:    $somaCores \leftarrow$  vetor de  $ls$  elementos (todos inicializados com 0)
3:    $numVizinhosColoridos \leftarrow 0$ 
4:   para cada vizinho  $j$  de  $i$  faça
5:     se  $j$  estiver colorido então
6:        $numVizinhosColoridos \leftarrow numVizinhosColoridos + 1$ 
7:        $resultPrimIneq \leftarrow coloracao[j] - d_{ij}$ 
8:       se  $resultPrimIneq \geq 1$  então
9:         para  $k \leftarrow 1$  até  $resultPrimIneq$  faça
10:           $somaCores[k] \leftarrow somaCores[k] + 1$ 
11:           $resultSegIneq \leftarrow coloracao[j] + d_{ij}$ 
12:          se  $resultSegIneq \leq limiteSuperior$  então
13:            para  $k \leftarrow resultSegIneq$  até  $limiteSuperior$  faça
14:               $somaCores[k] \leftarrow somaCores[k] + 1$ 
15:          para  $k \leftarrow resultSegIneq$  até  $limiteSuperior$  faça
16:            se  $somaCores[k] = numVizinhosColoridos$  então
17:              retorne  $k$ 
18:   retorne -1 ▷ Causará uma poda por bouding
```

---

puro para o problema de geometria de distâncias moleculares [53]. A segunda versão já é um algoritmo exato, porém, devido às verificações de todas as distâncias de vizinhos, se torna mais lento.

Resultados de experimentos computacionais com todos os métodos desenvolvidos serão detalhados no Capítulo 6, a seguir.

---

**Algoritmo 5.9** *Branch-Prune-and-Bound* - terceira versão para (MCDGP-Multi-List)

---

**Entrada:** grafo  $G$  (com conjunto de vértices  $V$ , conjunto de arestas  $E$  e matriz de distâncias entre vértices  $d$ ), função binária de arestas  $f$ , demandas de cores  $q$ , vértice atual  $i$  a ser colorido, vértice anterior  $j$ , coloração atual  $col$ , melhor coloração já encontrada  $colMelhor$  e limite superior  $ls$ .

```
1: Função BPB-3( $G = (V, E, d), f, q, i, j, col, colMelhor, ls$ )
2:   se todas as demandas de cores de todos os vértices foram satisfeitas então
3:     se  $\max_{\substack{i \in V \\ 1 \leq f \leq q_i}} col(i, f) < \max_{\substack{i \in V \\ 1 \leq f \leq q_i}} colMelhor(i, f)$  então
4:        $colMelhor \leftarrow col$ 
5:        $ls \leftarrow \max_{\substack{i \in V \\ 1 \leq f \leq q_i}} col(i, f)$ 
6:     senão
7:       para all  $i \in V$  faça
8:         se  $i$  não foi totalmente colorido então
9:           para todo  $k \in \ell(i)$  faça
10:            se  $Blq_{i,k} = 0$  então
11:               $col(i) \leftarrow col(i) \cup \{k\}$        $\triangleright$  Atribuir cor  $k$  a  $i$  (e decrementar
12:              demanda atual de  $i$ )
13:            se  $\max_{\substack{i \in V \\ 1 \leq f \leq q_i}} col(i, f) < ls$  então
14:              para cada vizinho  $j$  de  $i$  faça
15:                para todo  $m \in \ell(j)$  faça
16:                  se  $f((i, j)) = 0$  e  $|k - m| < d_{i,j}$  então
17:                     $Blq_{j,m} \leftarrow Block_{j,m} + 1$ 
18:                  senão
19:                    se  $f((i, j)) = 1$  e  $|k - m| \neq d_{i,j}$  então
20:                       $Blq_{j,m} \leftarrow Blq_{j,m} + 1$ 
21:                BPB-3( $G, f, q, i, j, col, colMelhor, ls$ )
22:              para cada vizinho  $j$  de  $i$  faça
23:                para todo  $m \in \ell(j)$  faça
24:                  se  $f((i, j)) = 0$  e  $|k - m| < d_{i,j}$  então
25:                     $Blq_{j,m} \leftarrow Blq_{j,m} - 1$ 
26:                  senão
27:                    se  $f((i, j)) = 1$  e  $|k - m| \neq d_{i,j}$  então
28:                       $Blq_{j,m} \leftarrow Blq_{j,m} - 1$ 
29:                 $col(i) \leftarrow col(i) - \{k\}$   $\triangleright$  Remover cor  $k$  de  $i$  (e incrementar
30:                demanda atual de  $i$ )
31:            retorne  $colMelhor$ 
```

---

# Capítulo 6

## Experimentos computacionais

Neste capítulo, serão apresentados os resultados obtidos com os algoritmos propostos, incluindo suas parametrizações, utilizando instâncias da literatura e outras adicionais geradas heurísticamente para este trabalho.

### 6.1 *Simulated annealing e branch-and-cut via CPLEX*

Para avaliar empiricamente os métodos desenvolvidos, foram utilizadas os conjuntos *Philadelphia*, *Helsinki* e as instâncias artificiais de 55 células. As implementações foram feitas nas linguagens C e C++ e os programas resultantes executados em um computador com processador Intel Core i7-3770 (4 núcleos a 3,4GHz), 12GB de memória e sistema operacional Ubuntu 12.04.1 (64 bits).

A parametrização dos algoritmos foi feita como se segue:

#### 1. *Simulated annealing*:

- 20 execuções para cada instância.
- Temperatura inicial  $t_0 = 100$ .
- Temperatura inicial  $t_f = 0$ .
- Número de iterações para o equilíbrio termodinâmico = 40.

#### 2. **B&C via CPLEX:**

- A solução final de cada execução do *simulated annealing* foi repassada como *MIP start* ao método (ou seja, 20 *MIP starts*).

**Tabela 6.1:** Resultados obtidos com o *simulated annealing* e o *branch-and-cut* via CPLEX para instâncias da literatura.

Células	#Inst.	Matr. Interf.	Demandas	Limite Inferior	Simulated Annealing [31]								B&C via CPLEX	
					Melhor Solução	Média Soluções	Pior Solução	Desvio Padrão	Melhor Tempo	Média Tempo	Pior Tempo	Média Tempo Melh. Sol.	Solução Obtida	Tempo
21	1	$C_{21}^1$	$D_{21}^1$	533	533	533.00	533	0.00	3.39	3.44	3.48	0.00	533	3.37
	2	$C_{21}^1$	$D_{21}^2$	309	309	309.00	309	0.00	2.98	3.02	3.07	0.00	309	1.92
	3	$C_{21}^2$	$D_{21}^1$	533	533	533.00	533	0.00	7.01	7.10	7.23	0.02	533	68.74
	4	$C_{21}^2$	$D_{21}^2$	309	309	309.00	309	0.00	5.47	5.53	5.96	0.09	309	25.32
	5	$C_{21}^3$	$D_{21}^1$	457	457	457.00	457	0.00	4.56	4.65	4.75	0.00	457	2.86
	6	$C_{21}^3$	$D_{21}^2$	265	265	265.00	265	0.00	3.70	3.73	3.79	0.00	265	3.35
	7	$C_{21}^4$	$D_{21}^1$	457	457	457.00	457	0.00	8.61	8.74	8.91	0.65	457	31.60
	8	$C_{21}^4$	$D_{21}^2$	265	265	268.70	272	1.35	6.32	6.42	6.50	1.89	265	18.68
	9	$C_{21}^5$	$D_{21}^1$	381	381	381.00	381	0.00	6.34	6.48	6.60	0.00	381	2.10
	10	$C_{21}^5$	$D_{21}^2$	221	221	221.00	221	0.00	4.63	4.68	4.73	0.02	221	1.90
	11	$C_{21}^6$	$D_{21}^1$	381	440	445.10	447	1.66	9.84	9.99	10.17	5.48	427	6897.88
	12	$C_{21}^6$	$D_{21}^2$	221	264	267.23	269	1.41	7.25	7.33	7.43	4.13	253	389718.40
	13	$C_{21}^7$	$D_{21}^1$	305	305	305.00	305	0.00	8.53	8.67	8.88	0.01	305	1.93
	14	$C_{21}^7$	$D_{21}^2$	177	182	183.33	185	0.79	5.75	5.84	5.92	1.71	180	47.70
	15	$C_{21}^8$	$D_{21}^1$	305	442	444.23	447	1.26	11.62	11.77	11.91	5.96	427	350.32
	16	$C_{21}^8$	$D_{21}^2$	177	270	271.87	274	1.23	8.40	8.56	8.69	4.22	253	5702.32
25	17	$C_{25}^1$	$D_{25}^3$	21	73	73.00	73	0.00	1.86	1.87	1.89	0.00	73	0.29
	18	$C_{25}^1$	$D_{25}^4$	89	200	200.00	200	0.00	11.88	12.02	12.17	0.00	200	1.70
55	19	$C_{55}^1$	$D_{55}^5$	309	309	309.00	309	0.00	30.81	31.26	31.66	0.00	309	47.99
	20	$C_{55}^1$	$D_{55}^6$	71	71	71.77	73	0.50	7.68	7.72	7.77	1.79	71	7.64

- Biblioteca utilizada: *Callable Library* (linguagem C).
- Número de *threads* = 1.

Um limite inferior para o valor do maior canal usado em cada instância pode ser definido como o maior produto da distância co-site pela demanda de canais dentre todos os vértices, isto é [25]:

$$\text{Limite inferior} = \max_{i \in V} [(c_{ii} \times (d_i - 1)) + 1]$$

Onde  $V$  é o conjunto de todas as ERBs da rede. A Tabela 6.1 mostra os resultados obtidos, com todos os tempos fornecidos medidos em segundos de CPU.

Os resultados obtidos foram comparados com os obtidos por Chakraborty [25], Kendall e Mohamad [47] e Chaves-González *et al.*, uma vez que as instâncias utilizadas foram as mesmas. A Tabela 6.2 resume os dados comparativos.

Os resultados obtidos são bastante competitivos em relação aos reportados na literatura, sendo melhores na maioria dos casos, apesar da simplicidade do algoritmo proposto. Nas instâncias onde os outros trabalhos já haviam chegado ao limite inferior, o SA obteve os mesmos resultados. Já em instâncias com valores ótimos não atingidos, o SA foi capaz de melhorar os resultados já existentes em praticamente todas elas, podendo-se des-

**Tabela 6.2:** Dados comparativos dos métodos utilizados com outros da literatura. O asterisco (\*) indica resultado que pode estar incorreto no trabalho correspondente.

Células	#Inst.	Matr. Interf.	Demandas	Limite Inferior	B&C via CPLEX (ótimo)	Simulated Annealing		Chakraborty [25]		Kendall e Mohamad [47]		Chaves-González et al. [26]	
						Melhor Solução	Gap (%)	Melhor Solução	Gap (%)	Melhor Solução	Gap (%)	Melhor Solução	Gap (%)
21	1	$C_{21}^1$	$D_{21}^1$	533	533	533	0	533	0	533	0	533	0
	2	$C_{21}^1$	$D_{21}^2$	309	309	309	0	309	0	309	0	309	0
	3	$C_{21}^2$	$D_{21}^1$	533	533	533	0	533	0	533	0	-	-
	4	$C_{21}^2$	$D_{21}^2$	309	309	309	0	309	0	309	0	-	-
	5	$C_{21}^3$	$D_{21}^1$	457	457	457	0	457	0	457	0	-	-
	6	$C_{21}^3$	$D_{21}^2$	265	265	265	0	265	0	265	0	-	-
	7	$C_{21}^4$	$D_{21}^1$	457	457	457	0	457	0	457	0	-	-
	8	$C_{21}^4$	$D_{21}^2$	265	265	265	0	280	5.67	273	3.00	-	-
	9	$C_{21}^5$	$D_{21}^1$	381	381	381	0	381	0	381	0	383	0.52
	10	$C_{21}^5$	$D_{21}^2$	221	221	221	0	221	0	221	0	223	0.90
	11	$C_{21}^6$	$D_{21}^1$	381	427	440	6.42	463	10.23	435	3.57	-	-
	12	$C_{21}^6$	$D_{21}^2$	221	253	264	4.34	273	7.90	268	5.92	-	-
	13	$C_{21}^7$	$D_{21}^1$	305	305	305	0	305	0	305	0	-	-
	14	$C_{21}^7$	$D_{21}^2$	177	180	182	1.11	197	9.44	185	2.77	-	-
	15	$C_{21}^8$	$D_{21}^1$	305	427	442	3.51	465	8.89	444	3.98	-	-
	16	$C_{21}^8$	$D_{21}^2$	177	253	270	6.71	278	9.88	271	7.11	-	-
25	17	$C_{25}^1$	$D_{25}^3$	21	73	73	0	73	0	73	0	-	-
	18	$C_{25}^1$	$D_{25}^4$	89	200	200	0	-121*	-39.50	200	0	-	-
55	19	$C_{55}^1$	$D_{51}^5$	309	309	309	0	309	0	309	0	-	-
	20	$C_{55}^1$	$D_{21}^6$	71	71	71	0	79	11.26	72	1.40	-	-

tacar a instância 20, onde o algoritmo atingiu o limite inferior, chegando a uma solução comprovadamente ótima sem necessidade de verificação por métodos exatos.

Uma situação peculiar ocorreu com a instância 18. O SA obteve a mesma solução dos outros trabalhos, menos a obtida por Chakraborty. Porém, ao executar o *branch-and-cut* via CPLEX, um método exato, verificou-se que o valor ótimo obtido foi equivalente ao SA. Isto pode indicar que houve um erro nos resultados reportados por Chakraborty, uma vez que todos os dados da instância foram amplamente verificados de forma a evitar inconsistências.

### 6.1.1 Instâncias adicionais

Com a finalidade de verificar o comportamento do algoritmo em outros cenários, foram gerados conjuntos adicionais de instâncias de alocação de canais. Para isso, foi desenvolvido um algoritmo para geração dos dados de cada instância, consistindo em três etapas: geração de uma rede, cálculo de distâncias e geração de demandas.

No passo de geração de uma rede com  $n$  ERBs, um grafo não-direcionado desconexo com  $n$  vértices e nenhuma aresta é inicialmente criado. Então, são criadas arestas alea-



tórias, de forma que cada vértice tenha grau mínimo 1 e grau máximo 6, de forma que a rede resultante siga o formato hexagonal das células.

Para determinar a matriz de interferência da rede, calcula-se distância entre cada par de ERBs, ou seja, o tamanho do menor caminho entre as duas ERBs do par. Para isso, aplica-se o algoritmo de Floyd-Warshall no grafo gerado no passo anterior, sem a necessidade de montar os menores caminhos entre pares, mas apenas calcular o tamanho dos mesmos. A partir desses cálculos, a matriz  $C = [c_{ij}]$  é montada. Foram gerados dois conjuntos de instâncias, cada um com parâmetros distintos, vistos a seguir.

1. **Conjunto padrão** (dados baseados nas instâncias da literatura).

- $c_{ii} = 5$  para toda ERB  $i$ .
- Se a distância entre as ERBs  $i$  e  $j$  for 1,  $c_{ij} = c_{ji} = 2$ .
- Se a distância entre  $i$  e  $j$  for 2,  $c_{ij} = c_{ji} = 1$ .
- Caso contrário, ou seja, distância entre  $i$  e  $j$  maior que 2,  $c_{ij} = c_{ji} = 0$ .

2. **Conjunto secundário** (interferências mais fortes).

- $c_{ii} = 7$  para toda ERB  $i$ .
- Se a distância entre as ERBs  $i$  e  $j$  for menor que 4,  $c_{ij} = c_{ji} = \max(\alpha - (d_{ij} - 1), 1)$ , onde  $\alpha = 3$ .
- Caso contrário, ou seja, distância entre  $i$  e  $j$  maior ou igual a 4,  $c_{ij} = c_{ji} = 0$ .

Por fim, as demandas de canais para cada ERB são geradas aleatoriamente, sendo que dois conjuntos de demandas são gerados para cada tamanho da rede: um com as demandas de canais entre 1 e 15 e outro com demandas entre 1 e 35. O Algoritmo 6.10 resume o processo de geração das instâncias, e as Tabela 6.3 e 6.3 mostra, os resultados obtidos com a aplicação dos algoritmos propostos nas instâncias adicionais dos conjuntos padrão e secundário, respectivamente, onde o CPLEX foi limitado a, aproximadamente, 10800 segundos de CPU (equivalente a 3 horas) por instância.

Uma característica notável do modelo de programação inteira para o problema é que, para essas instâncias, que são bem maiores que as da literatura, houve bastante dificuldade do resolvedor em solucionar a relaxação contínua do problema no nó raiz, o que reforça a necessidade do uso de heurísticas eficazes em aplicações reais.

**Tabela 6.3:** Resultados obtidos com o *simulated annealing* para as instâncias adicionais do conjunto padrão. Na coluna de resultados do CPLEX, um asterisco (\*) indica que o mesmo conseguiu melhorar a solução do SA passada como entrada.

Células	Matr. Interf.	Demandas	Total Chamadas	Limite Inferior	Simulated Annealing [31]							B&C via CPLEX		
					Melhor Solução	Média Soluções	Pior Solução	Desvio Padrão	Melhor Tempo	Média Tempo	Pior Tempo	Média Tempo Melh. Sol.	Solução Obtida	Tempo
70	$C_{70}^1$	$D_{70}^1$	639	71	134	135.05	136	0.67	53.38	60.86	80.68	16.65	130*	10801.00
	$C_{70}^2$	$D_{70}^2$	1254	166	260	264.05	266	1.56	145.01	170.53	205.85	92.40	260	10802.01
80	$C_{80}^1$	$D_{80}^1$	659	71	126	126.60	128	0.58	69.31	82.87	102.55	28.59	126	10802.14
	$C_{80}^2$	$D_{80}^2$	1454	171	279	281.10	283	1.14	234.18	260.28	341.45	97.71	279	10802.35
90	$C_{90}^1$	$D_{90}^1$	734	71	128	129.70	132	0.95	89.72	108.03	129.86	30.64	128	10800.89
	$C_{90}^2$	$D_{90}^2$	1634	171	289	290.25	293	1.18	297.71	321.65	382.11	95.19	289	10801.55
100	$C_{100}^1$	$D_{100}^1$	797	71	113	116.05	117	0.92	92.87	106.07	116.12	43.57	111*	10801.74
	$C_{100}^2$	$D_{100}^2$	1787	171	265	267.55	269	1.16	299.02	335.21	447.38	132.20	265	10801.54
110	$C_{110}^1$	$D_{110}^1$	852	71	118	119.00	120	0.55	157.52	176.59	202.39	83.30	118	10801.44
	$C_{110}^2$	$D_{110}^2$	1967	166	276	277.60	279	1.16	578.33	616.20	653.18	292.74	276	10802.55
120	$C_{120}^1$	$D_{120}^1$	973	71	128	129.10	130	0.70	196.84	236.79	267.72	77.67	125*	10801.26
	$C_{120}^2$	$D_{120}^2$	2143	171	267	268.35	270	0.96	615.57	712.41	771.17	340.97	267	10802.21
130	$C_{130}^1$	$D_{130}^1$	1051	71	123	123.85	125	0.48	218.93	243.49	280.34	43.00	121*	10801.30
	$C_{130}^2$	$D_{130}^2$	2311	171	257	259.75	262	1.37	572.78	750.95	826.02	229.01	257	10802.20
140	$C_{140}^1$	$D_{140}^1$	1098	71	122	123.50	125	0.74	294.89	324.06	340.70	153.09	122	10801.11
	$C_{140}^2$	$D_{140}^2$	2543	171	282	284.20	288	1.36	783.73	942.30	1165.50	435.39	282	10802.21
150	$C_{150}^1$	$D_{150}^1$	1212	71	128	129.65	131	0.79	354.40	376.91	414.58	153.90	128	10801.01
	$C_{150}^2$	$D_{150}^2$	2839	171	283	285.40	287	1.20	939.33	1112.79	1419.76	309.93	283	10801.60

**Tabela 6.4:** Resultados obtidos com o *simulated annealing* para as instâncias adicionais do conjunto secundário com interferências mais fortes.

Células	Matr. Interf.	Demandas	Total Chamadas	Limite Inferior	Simulated Annealing [31]							B&C via CPLEX		
					Melhor Solução	Média Soluções	Pior Solução	Desvio Padrão	Melhor Tempo	Média Tempo	Pior Tempo	Média Tempo Melh. Sol.	Solução Obtida	Tempo
70	$C_{70}^3$	$D_{70}^3$	477	99	303	304.70	306	0.90	234.36	254.09	269.15	81.85	303	10802.10
	$C_{70}^4$	$D_{70}^4$	1282	239	781	785.50	789	2.69	1465.08	1496.67	1550.43	506.45	781	10812.68
80	$C_{80}^3$	$D_{80}^3$	625	99	371	372.40	374	0.80	477.18	489.09	500.93	193.03	371	10808.29
	$C_{80}^4$	$D_{80}^4$	1540	239	860	864.80	867	2.09	2575.54	2617.37	2657.12	1231.43	860	10853.89
90	$C_{90}^3$	$D_{90}^3$	669	99	332	334.30	335	1.19	516.85	543.24	569.51	349.94	332	10804.53
	$C_{90}^4$	$D_{90}^4$	1729	239	809	813.20	817	2.23	3016.21	3051.65	3082.22	1544.29	809	10848.03
100	$C_{100}^3$	$D_{100}^3$	795	99	388	389.50	391	0.92	903.02	952.07	986.92	399.30	388	10821.81
	$C_{100}^4$	$D_{100}^4$	1845	239	887	890.20	893	1.83	4620.82	4709.77	4786.80	1496.26	887	10857.49
110	$C_{110}^3$	$D_{110}^3$	860	99	358	359.00	361	0.77	976.64	1025.80	1065.07	386.31	358	10823.85
	$C_{110}^4$	$D_{110}^4$	2040	239	828	831.60	834	1.80	4677.56	4786.17	4872.57	1224.42	828	10900.42
120	$C_{120}^3$	$D_{120}^3$	873	99	339	339.20	340	0.40	1134.75	1171.61	1223.16	495.27	339	10823.32
	$C_{120}^4$	$D_{120}^4$	2223	239	877	879.50	883	1.69	6964.13	7093.11	7186.02	2197.32	877	10878.44
130	$C_{130}^3$	$D_{130}^3$	1093	99	415	416.40	417	0.66	2076.24	2143.25	2194.09	832.11	415	10841.50
	$C_{130}^4$	$D_{130}^4$	2378	239	887	891.60	894	2.24	7880.72	8734.53	9248.03	4486.48	887	10897.88
140	$C_{140}^3$	$D_{140}^3$	1123	99	381	383.00	385	1.18	2138.92	2157.81	2201.75	847.69	381	10843.90
	$C_{140}^4$	$D_{140}^4$	2558	239	851	853.50	856	1.50	7284.59	8508.40	9606.98	4025.51	851	10894.22
150	$C_{150}^3$	$D_{150}^3$	1283	99	427	428.30	430	1.10	3041.92	3118.01	3161.13	1057.45	427	10858.02
	$C_{150}^4$	$D_{150}^4$	2688	239	875	878.30	881	1.55	7480.58	8935.76	11168.39	4118.39	875	10874.84

---

**Algoritmo 6.10** Gerador de instâncias para alocação de canais

---

**Entrada:** Número  $n$  de ERBs da rede.

```
1: Função GERAINSTANCIAS-ALOCCANAIS( $n$ )
2:    $A \leftarrow$  matriz de adjacências  $n \times n$  (inicialmente com zeros)
3:   para  $i \leftarrow 1$  até  $n$  faça
4:      $grauGerado \leftarrow$  número aleatório em  $U[1, 6]$ 
5:     se grau de  $i < grauGerado$  então
6:       para  $k \leftarrow$  grau de  $i$  até  $grauGerado$  faça
7:          $j \leftarrow$  número aleatório em  $U[1, n]$  tal que  $j \neq i$ 
8:          $A[i, j] \leftarrow 1$ 
9:          $A[j, i] \leftarrow 1$ 
10:        Incrementar graus de  $i$  e  $j$ 
11:    $Dist \leftarrow$  matriz de distâncias  $n \times n$ 
12:    $Dist \leftarrow$  FLOYD-WARSHALL( $A, n$ )
13:    $C \leftarrow$  matriz de interferências  $n \times n$ 
14:   para  $i \leftarrow 1$  até  $n$  faça
15:      $C[i, i] \leftarrow 5$ 
16:     para  $j \leftarrow i + 1$  até  $n$  faça
17:       se  $Dist[i, j] = 1$  então
18:          $C[i, j] \leftarrow 2$ 
19:          $C[j, i] \leftarrow 2$ 
20:       senão
21:         se  $Dist[i, j] = 2$  então
22:            $C[i, j] \leftarrow 1$ 
23:            $C[j, i] \leftarrow 1$ 
24:         senão
25:            $C[i, j] \leftarrow 0$ 
26:            $C[j, i] \leftarrow 0$ 
27:    $q \leftarrow$  vetor de demandas
28:   para  $i \leftarrow 1$  até  $n$  faça
29:      $q[i] \leftarrow$  número aleatório em  $U[1, maxDem]$ , onde  $maxDem = 15$  ou  $35$ 
30:   retorne matriz  $C$  (equivalente ao conjunto de distâncias  $d$ ) e vetor  $q$ 
```

---

## 6.2 *Branch-prune-and-bound*

Para efetuar uma análise de ambas as versões do *branch-prune-and-bound*, foram geradas instâncias de grafos com distâncias nas arestas e demandas unitárias de cores em todos os vértices. As quantidades de vértices consideradas nas instâncias foram 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18 e 20. As duas versões do B&P&N foram implementadas em linguagens C e executadas no mesmo equipamento dos testes anteriores (máquina com processador Intel Core i7-3770 (4 núcleos a 3,4GHz), 12GB de memória e sistema operacional Ubuntu 12.04.1 (64 bits)).

De forma a comparar os resultados obtidos, o algoritmo *simulated annealing* proposto

anteriormente foi adaptado para essas instâncias. A solução inicial obtida antes da meta-heurística em si foi usada como limite superior inicial em ambas as versões do B&P&B, de forma a comparar a convergência das mesmas com a do SA partindo de um mesmo ponto inicial. Para cada execução de cada versão do B&P&B, foram medidas as quantidades de podas por *pruning* e *bounding*, além da quantidade de soluções completas geradas pelas enumerações. Ambas as versões foram limitadas a 3 horas (10800 segundos) de tempo de CPU para sua execução. A Tabela 6.5 fornece os dados resultantes dos testes.

A primeira versão do *branch-prune-and-bound* se mostrou muito mais rápida, uma vez que leva em conta apenas a distância de no máximo um vértice vizinho a cada vértice a ser colorido, ao contrário da segunda versão que considera as distâncias de todos os vizinhos coloridos. No entanto, em algumas instâncias (com 14 e 16 vértices), a solução obtida não foi ótima, mesmo sem consumir todo o tempo disponível para execução. Particularmente, para a instância de 16 vértices, a segunda versão foi capaz de obter um resultado melhor mesmo atingindo o limite de tempo. A instância de 18 vértices registrou uma situação contrária, onde a primeira versão obteve um resultado melhor.

Outra propriedade verificada diz respeito ao número de podas de cada tipo em cada versão do algoritmo. A segunda versão do algoritmo efetua muito mais podas por *bounding*. Como o processo de seleção de cores de um vértice efetua uma poda desse tipo caso não consiga encontrar uma cor válida entre 1 e o limite superior atual no momento de colorir o vértice considerado, o que não ocorre na primeira versão. Em contrapartida, a segunda versão não realizou nenhuma poda por *pruning*, o que indica que todas as colorações geradas (parciais e completas) não violaram distâncias do grafo, o que também é explicado pelo fato de levar em conta as distâncias de todos os vértices a cada passo. Como a primeira versão considera apenas um vizinho, em alguns casos, a cor pode violar a distância de outro vizinho, o que será detectado pelo teste de factibilidade.

Para a terceira versão do *branch-prune-and-bound*, experimentos computacionais foram realizados utilizando algumas instâncias geradas aleatoriamente com número de vértices variando entre 4 e 10. Foram derivados cenários para os problemas CMDGP-List, MCDGP-List (com listas distintas e iguais), CMDGP-Multi-List e MCDGP-Multi-List. Os resultados dos experimentos são dados na Tabela 6.6 e na Figura 6.1. As instâncias envolvendo multicoloração são mais difíceis de resolver, uma vez que, ao incrementar a demanda de um vértice, o mesmo está basicamente sendo duplicado no grafo. O tempo

**Tabela 6.5:** Resultados obtidos com o *simulated annealing* e as duas versões do *branch-prune-and-bound* para as instâncias de colorações com demandas únicas e distâncias variáveis.

# Vért.	Sol. inicial	Sim. Anneal.		BPB-Prev					BPB-select				
		Sol. Obt.	Tempo	Sol. Obt.	# Bounds	# Prunes	# Sol.	Tempo	Sol. Obt.	# Bounds	# Prunes	# Sol.	Tempo
3	8	6	0.000	6	1	2	2	0.000	6	2	0	2	0.000
4	8	6	0.000	6	8	9	2	0.000	6	20	0	2	0.000
5	6	6	0.000	6	20	10	4	0.000	6	77	0	1	0.000
6	7	7	0.000	7	59	53	4	0.000	7	634	0	1	0.000
7	14	12	0.000	12	2003	4489	5	0.000	12	4132	0	3	0.020
8	6	6	0.000	6	62	5	8	0.000	6	1878	0	1	0.000
9	20	14	0.000	14	8145	21108	7	0.020	14	228856	0	5	0.580
10	15	13	0.000	13	19251	43944	11	0.030	13	1658851	0	3	2.590
12	23	14	0.020	14	146056	378135	13	0.630	14	84727255	0	5	189.810
14	12	8	0.010	9	24749	64137	8	0.230	8	3353637389	0	5	7948.410
16	18	13	0.040	14	4016068	12314917	21	23.720	13	4191668753	0	5	10800.030
18	33	25	0.080	25	1779401118	5012294261	25	10800.080	31	1603800110+	0	3	10800.080
20	38	34	0.140	34	1956130621	4201328391	26	10800.140	33	2059037363+	0	6	10800.140

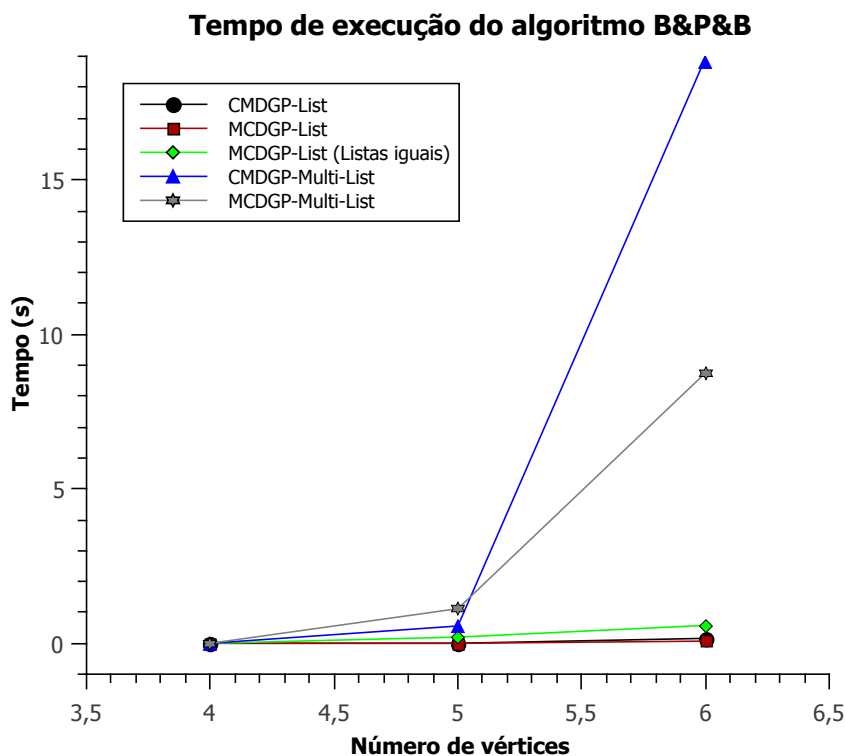
**Tabela 6.6:** Resultados da execução da terceira versão do algoritmo *branch-prune-and-bound* aplicado a instâncias dos problemas CMDGP-List e MCDGP-List (listas distintas e iguais).

#Vert	CMDGP-List					MCDGP-List					MCDGP-List (listas iguais)				
	Sol. Obt.	#Bounds	#Prunes	#Sol	Tempo (s)	Sol. Obt.	#Bounds	#Prunes	#Sol	Tempo (s)	Sol. Obt.	#Bounds	#Prunes	#Sol	Tempo (s)
4	7	1165	280	1	0.000	7	1165	280	1	0.000	4	6066	2301	2	0.000
5	8	10174	6385	1	0.000	8	10174	6385	1	0.000	5	147910	81783	2	0.020
6	10	305652	2875327	2	0.160	10	72589	109671	2	0.070	6	2175875	1731638	5	0.580
7	13	3759459	84146237	3	4.630	15	352338	4421436	1	2.390	9	97921297	164055357	4	42.950
8	13	73343425	1708180121	3	112.420	13	73343425	1708180121	3	86.890	9	535223979	915288021	4	301.170
9	13	557401766	450263944	3	877.640	-	-	-	-	-	-	-	-	-	-

de solução para quantidade de vértices maior que 6 neste caso foi muito alto. Quando distâncias iguais são permitidas no problema, o tempo de execução tende a ser menor, já que há menos opções de cores que podem ser usadas sem violar restrições de igualdade nas distâncias. Entretanto, ao introduzir listas iguais, o tempo aumenta, já que cada vértice passa a ter mais opções.

**Tabela 6.7:** Resultados da execução da terceira versão do algoritmo *branch-prune-and-bound* aplicado a instâncias dos problemas CMDGP-Multi-List e MCDGP-Multi-List.

#Vert	CMDGP-Multi-List					MCDGP-Multi-List				
	Sol. Obt.	#Bounds	#Prunes	#Sol	Tempo (s)	Sol. Obt.	#Bounds	#Prunes	#Sol	Tempo (s)
4	7	4151	1191	1	0.000	7	1907	42714	1	0.000
5	10	2722793	3400106	3	0.560	10	508274	11309799	3	1.210
6	10	18391637	136594416	4	18.940	10	3575337	91939693	4	8.750



**Figura 6.1:** Número de vértices  $\times$  tempo de execução do algoritmo B&P&B nos problemas usados.

### 6.3 Notas do capítulo

Os resultados da heurística *simulated annealing* mostraram-se satisfatórios em comparação aos outros trabalhos da literatura, obtendo soluções de custo mais baixo. Além disso, por meio da aplicação do resolvidor CPLEX a uma formulação de programação inteira, foi possível determinar os custos ótimos para todas as instâncias da literatura. Como uma contribuição adicional, novas instâncias foram criadas e o uso dos algoritmos nas mesmas foi avaliado empiricamente, sendo que, por serem muito maiores que as da literatura, o uso do CPLEX foi bastante limitado, sendo obtidos apenas limites superiores para algumas dessas instâncias.

Testes com os algoritmos *branch-prune-and-bound* também foram realizados com um conjunto diferente de instâncias cujo foco foi explorar distâncias em grafos. Foi possível

verificar que a técnica de determinação de cor por meio da distância de um vizinho anterior se mostrou bem mais rápida, no entanto, em algumas instâncias verificou-se que, mesmo não consumindo todo o tempo alocado, o método não chegou à otimalidade verificada na segunda versão do método. Em contrapartida, a segunda versão consegue atingir os ótimos se for executado até sua finalização e dispensa testes de factibilidade, mas é mais lenta.

# Capítulo 7

## Considerações Finais

Nesta dissertação, foram apresentados os problemas de alocação de canais em redes sem fio, que possuem grande importância no cenário atual de comunicação móvel.

Diversas contribuições teóricas e práticas foram feitas com este trabalho. Foram propostos novos modelos que incorporam características importantes ainda não modeladas, através de correlações com outros problemas de otimização combinatória. Utilizando modelos em grafos, foi possível estabelecer correlações com colorações restritas, incluindo modelos recentes da literatura, como a  $(\gamma, \mu)$ -coloração.

Não foram encontrados, durante a pesquisa, estudos envolvendo a correlação direta entre escalonamento de tarefas e alocação de canais. Assim, foram propostos modelos em máquinas paralelas com restrições de recursos, tarefas de tempo unitário e tempos de preparação distintos para efetuar tal correlação.

Devido à presença das interferências em redes sem fio, que exige a separação entre canais, identificou-se a possibilidade da aplicação de geometria de distâncias em alocação de canais, baseando-se também nas correlações com colorações em grafos, de forma que cada canal é considerado um ponto inteiro na reta dos números positivos e as distâncias consideradas são as separações entre canais.

A partir dessas correlações com outros problemas de otimização combinatória, é possível explorar todo o ferramental de uma classe de problemas aplicando-o a outras classes, com as devidas adaptações, o que permite o surgimento de novos algoritmos e mesmo a determinação de propriedades para soluções dos mesmos.

Para solução do problema, foram implementados algoritmos aproximados e exatos. Como métodos aproximados, foram desenvolvidos uma busca local iterada e um *simu-*



*lated annealing* onde a solução é codificada como uma lista de chamadas da rede e a vizinhança de uma solução é definida por trocas entre pares de chamadas na sequência. Em testes preliminares, a busca local iterada não apresentou bom desempenho, portanto, o foco foi no *simulated annealing*.

Como método exato, foi implementado um modelo matemático de programação inteira com a ferramenta CPLEX, onde as melhores soluções obtidas pelos métodos aproximados são utilizadas como limites superiores e soluções iniciais do *branch-and-cut* da ferramenta.

Com os métodos utilizados, foi possível obter resultados iguais ou melhores que os existentes na literatura para a maioria das instâncias usadas, com soluções ótimas para instâncias do problema de alocação de canais em redes celulares cujos valores ótimos não foram encontrados na literatura. Como contribuição adicional, foram gerados conjuntos adicionais de instâncias, com maiores redes, demandas de canais e interferências, cuja solução é mais difícil que as da literatura, o que permite uma avaliação mais completa do desempenho de algoritmos propostos para o problema.

Outro algoritmo desenvolvido foi o *branch-prune-and-bound*, combinando técnicas consolidadas de geometria de distâncias moleculares com um método mais geral de otimização combinatória, em duas versões que exploram diferentes maneiras para tratamento das distâncias nas arestas dos grafos, sendo possível verificar um comportamento aproximado em uma versão e exato em outra. Ambas as versões foram testadas utilizando instâncias artificiais cujo foco foram as distâncias entre os vértices, possibilitando a exploração prática da correlação entre colorações e geometria de distâncias, em particular, a avaliação de casos específicos que não são amplamente discutidos na literatura, por exemplo, envolvendo distâncias inteiras na reta ( $\mathbb{R}^1$ ). Apesar de o algoritmo ainda não apresentar desempenho competitivo, um estudo mais aprofundado pode ser feito de forma a aproveitar melhor a estrutura dos problemas propostos no método.

Como trabalhos futuros, existe a possibilidade de desenvolver algoritmos ainda mais eficazes que obtenham soluções melhores para as instâncias adicionais, uma vez que a tendência nas redes reais é aumentar cada vez mais para comportar a maior demanda dos usuários. Além disso, com base nos modelos matemáticos citados, incluindo a modificação utilizada na implementação do *branch-and-cut* via CPLEX, podem ser identificados cortes específicos do problema para implementação de um algoritmo mais eficaz, além da

aplicação de geração de colunas ou mesmo a combinação de ambos os métodos em um *branch-cut-and-price* robusto. Por fim, para as correlações com geometria de distâncias e os algoritmos *branch-prune-and-bound*, é interessante identificar formas adicionais de ordenar os vértices antes da execução do método, uma vez que isso tem forte impacto na velocidade de obtenção de soluções de custo menor que a melhor já encontrada, além de ajudar na exploração do espaço de soluções do problema. Além disso, é importante encontrar um equilíbrio nas duas formas de determinação de cores com base nas distâncias (uma de cada versão do método), uma vez que, quanto mais distâncias são consideradas a cada passo, a confiabilidade em relação ao valor ótimo é maior, mas passa a ser exigido mais tempo computacional.

# Referências

- [1] AARDAL, K. I., HIPOLITO, A., VAN HOESEL, C. P. M., AND JANSEN, B. A branch-and-cut algorithm for the frequency assignment problem. *Research Memorandum 96/011* (1996).
- [2] AARDAL, K. I., HURKENS, C., LENSTRA, J. K., AND TIOURINE, S. Algorithms for Radio Link Frequency Assignment: the CALMA Project. *Operations Research* 50, 6 (2002), 968–980.
- [3] AKYILDIZ, I. F., LEE, W.-Y., VURAN, M. C., AND MOHANTY, S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks* 50, 13 (2006), 2127–2159.
- [4] AKYILDIZ, I. F., LEE, W.-Y., VURAN, M. C., AND MOHANTY, S. A Survey on Spectrum Management in Cognitive Radio Networks. *IEEE Communications Magazine* 46, 4 (2008), 40–48.
- [5] AKYILDIZ, I. F., LO, B. F., AND BALAKRISHNA, R. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication* 4, 1 (2011), 40–62.
- [6] AMORIM, R. X., DIAS, B. R. C., AND RODRIGUES, R. F. ILS com reconexão de caminhos entre ótimos locais para um problema clássico de escalonamento com antecipação e atraso. In *Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional* (2012), Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Disponível em <http://www2.claiosbpo2012.iltc.br/pdf/102263.pdf>.
- [7] AMORIM, R. X., DIAS, B. R. C., RODRIGUES, R. F., AND UCHOA, E. A hybrid genetic algorithm with local search approach for e/t scheduling problems

on identical parallel machines. In *GECCO '13 Companion - Proceedings of 15th Annual Conference Companion on Genetic and Evolutionary Computation* (2013), ACM.

- [8] ANDERSON, L. A Simulation Study of Some Dynamic Channel Assignment Algorithms in a High Capacity Mobile Telecommunications System. *IEEE Transactions on Communications* 21, 11 (1973), 1294–1301.
- [9] ARAÚJO, A., PEÑARANDA, A., AND NAKAMURA, F. G. Avaliando o Impacto da Representação da Solução no Problema de Cobertura em Redes de Sensores sem Fio. In *Anais Eletrônicos do XLIV Simpósio Brasileiro de Pesquisa Operacional* (2012).
- [10] AUDHYA, G. K., SINHA, K., GHOSH, S. C., AND SINHA, B. P. A survey on the channel assignment problem in wireless networks. *Wireless Communications and Mobile Computing* 11 (2011), 583–609.
- [11] AZEVEDO, G. H. I. Escalonamento de projetos com restrição de recursos e com *timelags* máximos e mínimos: um limite inferior por propagação de restrições. Master's thesis, Mestrado em Engenharia de Produção – Universidade Federal Fluminense, 2012.
- [12] BAPTISTE, P., AND BRUCKER, P. Scheduling equal processing time jobs: a survey. In *Handbook of Scheduling*, J. Y.-T. Leung, Ed. Chapman and Hall/CRC, 2004, pp. 14.1 – 14.37.
- [13] BECKMANN, D., AND KILLAT, U. A new strategy for the application of genetic algorithms to the channel assignment problem. *IEEE Transactions on Vehicular Technology* 48, 4 (1999), 1261–1269.
- [14] BEEKER, N., GAUBERT, S., GLUSA, C., , AND LIBERTI, L. Is the Distance Geometry Problem in NP? In *Distance Geometry*, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, Eds. Springer, 2013, pp. 85–93.
- [15] BELDICEANU, N., AND BELDICEANU, N. Global Constraints Catalog. <http://www.emn.fr/z-info/sdemasse/gccat/>, 2014.

- [16] BOCKMAYR, A. Integer Programming, Constraint Programming, and their Combination. <http://www.win.tue.nl/diamant/ipday060127/bockmayr.pdf>, 2006.
- [17] BONDY, J. A., AND MURTY, U. S. R. *Graph Theory with Applications*. Elsevier Science Publishing, 1982.
- [18] BONOMO, F., AND CECOWSKI, M. Between coloring and list-coloring:  $\mu$ -coloring. *Electronic Notes in Discrete Mathematics* 19 (2005), 117–123.
- [19] BONOMO, F., DURÁN, G., AND MARENCO, J. Exploring the complexity boundary between coloring and list-coloring. *Electronic Notes in Discrete Mathematics* 25 (2006), 41–47.
- [20] BRADLEY, S. P., HAX, A. C., AND MAGNANTI, T. L. *Applied Mathematical Programming*. Addison-Wesley, 1977. Reeditado digitalmente e disponível em <http://web.mit.edu/15.053/www/>.
- [21] BRUCKER, P. *Scheduling Algorithms*, 5th ed. Springer, 2006.
- [22] BRUCKER, P., AND KNUST, S. Complexity results for scheduling problems. <http://www.informatik.uni-osnabrueck.de/knust/class/>, 2009.
- [23] CABRIC, D. B., AND BRODERSEN, R. W. *Cognitive Radios: System Design Perspective*. PhD thesis, EECS Department, University of California, Berkeley, 2007.
- [24] CAMPI, M. Coreia do Sul desenvolve rede 5G mil vezes mais veloz que a 4G. INFO Online, disponível em <http://info.abril.com.br/noticias/mercado/2014/02/coreia-do-sul-desenvolve-rede-5g-mil-vezes-mais-veloz-que-a-4g.shtml>, 2014. Acessado em 05 de fevereiro de 2014.
- [25] CHAKRABORTY, G. An Efficient Heuristic Algorithm for Channel Assignment Problem in Cellular Radio Networks. *IEEE Transactions on Vehicular Technology* 50, 6 (2001), 1528–1539.

- [26] CHAVES-GONZÁLEZ, J. M., VEGA-RODRÍGUEZ, M. A., DOMÍNGUEZ-GONZÁLEZ, D., AND JUAN A. GÓMEZ-PULIDO, J. M. S.-P. Population-Based Incremental Learning to Solve the FAP Problem. In *Proceedings of the 2th International Conference on Advanced Engineering Computing and Applications in Sciences* (2008).
- [27] COELHO, I. M., HADDAD, M. N., OCHI, L. S., SOUZA, M. J. F., AND FARIAS, R. A Hybrid CPU-GPU Local Search Heuristic for the Unrelated Parallel Machine Scheduling Problem. In *Proceedings of 3rd Workshop on Applications for Multi-Core Architecture* (2013), vol. 19, pp. 201–232.
- [28] DA SILVA, A. R. V., AND OCHI, L. S. A dynamic resource constrained project scheduling problem. *International Journal of Data Mining, Modelling and Management* 5, 4 (2013), 370–379.
- [29] DATORRO, J. *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2013. Disponível em <https://ccrma.stanford.edu/~dattorro/mybook.html>.
- [30] DIAS, B. R. C., RODRIGUES, R. F., AND MACULAN, N. Alocação de canais em redes celulares sem fio: algoritmos e modelos teóricos em grafos e escalonamento. In *Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional* (2012), Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Disponível em <http://www2.claiosbpo2012.iltc.br/pdf/102176.pdf>.
- [31] DIAS, B. R. C., RODRIGUES, R. F., AND MACULAN, N. *Simulated annealing* para a alocação de canais em redes móveis celulares. In *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional* (2013), Sociedade Brasileira de Pesquisa Operacional (SOBRAPO).
- [32] DIAS, B. R. C., RODRIGUES, R. F., AND SZWARCFITER, J. On graph coloring problems with distance constraints. In *Proceedings of I Workshop on Distance Geometry and Applications (DGA 2013)* (2013).
- [33] DONG, Q., AND WU, Z. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization* 22, 1–4 (2002), 365–375.

- [34] DORNE, R., AND HAO, J. Tabu search for graph coloring, t-coloring and set t-colorings. In *Metaheuristics: advances and trends in local search paradigms for optimization*, S. Voss, S. Martello, I. Osman, and C. Roucairol, Eds. Kluwer Academic Publishers, 1998, pp. 77–92.
- [35] DOURADO, M. C., DE F. RODRIGUES, R., AND SZWARCFITER, J. L. Scheduling unit time jobs with integer release dates to minimize the weighted number of tardy jobs. *Annals of Operations Research* 169, 1 (2009), 81–91.
- [36] DOURADO, M. C., DE F. RODRIGUES, R., AND SZWARCFITER, J. L. Scheduling on parallel machines considering job-machine dependency constraints. *Electronic Notes in Discrete Mathematics*, 36 (2010), 431–438.
- [37] EISENBLÄTTER, A., AND KOSTER, A. FAP web - a website about frequency assignment problems. <http://fap.zib.de>, 2007.
- [38] ERICSSON. Mobility Report - February 2014. <http://www.ericsson.com/traffic-market-report>, Fevereiro 2014.
- [39] FOROUZAN, B. A. *Data Communications and Networking*, 4th ed. McGraw-Hill, 2007.
- [40] GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [41] GOLDBARG, M. C., AND LUNA, H. P. L. *Otimização Combinatória e Programação Linear*, 2<sup>a</sup> ed. Elsevier, 2005.
- [42] GRAHAM, R. L., LAWLER, E. L., LENSTRA, J. K., AND KAN, A. H. G. R. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* (1979).
- [43] HALE, W. K. Frequency assignment: Theory and applications. *Proceedings of the IEEE* 25 (1980), 1497–1514.
- [44] HOANG, A. T., AND LIANG, Y.-C. Maximizing Spectrum Utilization of Cognitive Radio Networks Using Channel Allocation and Power Control. In *64th Vehicular Technology Conference* (2006), IEEE.

- [45] HUAWEI. 5G: A Technology Vision. Disponível em [http://www.huawei.com/ilink/en/download/HW\\_314849](http://www.huawei.com/ilink/en/download/HW_314849), 2013. Acessado em 05 de fevereiro de 2014.
- [46] JOUINI, W., MOY, C., AND PALICOT, J. Decision making for cognitive radio equipment: analysis of the first 10 years of exploration. *EURASIP Journal on Wireless Communications and Networking 2012* (2012), 26.
- [47] KENDALL, G., AND MOHAMAD, M. Solving the Fixed Channel Assignment Problem in Cellular Communications Using an Adaptive Local Search. In *5th International Conference for the Practice and Theory of Automated Timetabling (PATAT 2004)* (2005), E. Burke and M. Trick, Eds., Lecture Notes in Computer Science, vol. 3616, Springer, Heidelberg.
- [48] KIRKPATRICK, S., JR., C. D. G., AND VECCHI, M. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680.
- [49] KOSTER, A. M. C. A. *Frequency assignment: models and algorithms*. Universiteit Maastricht, 1999.
- [50] KOSTER, A. M. C. A., AND MUÑOZ, X. *Graphs and Algorithms in Communication Networks: Studies in Broadband, Optical, Wireless and Ad Hoc Networks*. Texts in Theoretical Computer Science. Springer, 2010.
- [51] KUNZ, D. Channel Assignment for Cellular Radio Using Neural Networks. *IEEE Transactions on Vehicular Technology* 40, 1 (1991), 188–193.
- [52] KUROSE, J. F., AND ROSS, K. W. *Redes de computadores e a Internet: Uma abordagem top-down*, 3rd ed. Pearson Addison-Wesley, 2006.
- [53] LAVOR, C., LIBERTI, L., MACULAN, N., AND MUCHERINO, A. The discretizable molecular distance geometry problem. *European Journal of Operational Research* 52, 1 (2012), 115–146.
- [54] LAWLER, E. *Combinatorial Optimization: networks and matroids*. Dover Publications, 1976.



- [55] LETCHFORD, A. N. *An Introduction to Branch-and-Cut - Part I: Polyhedral Theory*, 2011.
- [56] LEUNG, J. Y.-T. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman and Hall/CRC, 2004.
- [57] LIBERTI, L., LAVOR, C., MACULAN, N., AND MUCHERINO, A. Euclidean Distance Geometry and Applications. *SIAM Review* 56, 1 (2014), 3–69.
- [58] LUKE, S. *Essentials of Metaheuristics*. Lulu, 2009. Disponível em <http://cs.gmu.edu/~sean/book/metaheuristics/>.
- [59] MACULAN, N., AND FAMPA, M. *Otimização Linear*. Editora da Universidade de Brasília (UnB), 2006.
- [60] MCDIARMID, C., AND REED, B. Channel assignment and weighted coloring. *Networks* 36 (2000), 114–117.
- [61] MEHROTRA, A., AND TRICK, M. A. A Branch-And-Price Approach for Graph Multi-Coloring. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, E. Baker, A. Joseph, A. Mehrotra, and M. A. Trick, Eds., vol. 37 of *Operations Research/Computer Science Interfaces Series*. Springer, 2007, pp. 15–29.
- [62] METROPOLIS, N., ROSENBLUTH, A., AND ROSENBLUTH, M. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* 21, 6 (1953), 1087–1092.
- [63] MISHRA, S. M. Channel allocation in cognitive radios. Tech. rep., EECS Department, University of California, Berkeley, 2005. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.7476>.
- [64] MUCHERINO, A., LAVOR, C., LIBERTI, L., AND MACULAN, N. *Distance Geometry*. Springer, 2013.
- [65] MURPHEY, R. A., PARDALOS, P. M., AND RESENDE, M. G. C. Frequency Assignment Problems. In *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, 1999, pp. 295–377.

- [66] NAKAMURA, F. G. *Algoritmos para Controle de Densidade em Redes de Sensores sem Fio*. PhD thesis, Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Minas Gerais, 2010.
- [67] NARAYANASWAMY, K. Analysis and Optimization of Channel Allocation Strategies in Cellular Network. *International Journal of Computer Networks and Wireless Communications* 2, 6 (2012), 739–743.
- [68] OLIVEIRA, H. A. B. F. *Localização no Tempo e no Espaço em Redes de Sensores sem Fio*. PhD thesis, Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Minas Gerais, 2008.
- [69] PAPADIMITRIOU, C., AND STEIGLITZ, K. *Combinatorial Optimization: algorithms and complexity*. Dover Publications, 1998.
- [70] PENNA, P. H. V., SUBRAMANIAN, A., AND OCHI, L. S. An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *Journal of Heuristics* 19 (2013), 201–232.
- [71] PESSOA, A., UCHOA, E., DE ARAGÃO, M. P., AND DE F. RODRIGUES, R. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation* 2 (2010), 259–290.
- [72] PEÑARANDA, A., ARAÚJO, A., NAKAMURA, F. G., AND RODRIGUES, R. F. Um Algoritmo GRASP Para O Problema De K-cobertura Em Redes De Sensores Sem Fio. In *Anais Eletrônicos do XLIII Simpósio Brasileiro de Pesquisa Operacional* (2011).
- [73] PINEDO, M. *Scheduling: Theory, Algorithms and Systems*, 2<sup>a</sup> ed. Prentice Hall, 2002.
- [74] PROJETO GNU. GNU Radio. <http://www.gnuradio.org>, 2001–2012.
- [75] RODRIGUES, R. F. *Caracterizações e Algoritmos para Problemas Clássicos de Escalonamento*. PhD thesis, Programa de Engenharia de Sistemas e Computação – COPPE – Universidade Federal do Rio de Janeiro, 2009.

- [76] RODRIGUES, R. F., AMORIM, R. X., AND DIAS, B. R. C. MIP models and algorithms for earliness/tardiness scheduling problems on parallel machines. In *25th European Conference on Operational Research* (2012).
- [77] RODRIGUES, R. F., DIAS, B. R. C., , OLIVEIRA, A. M., AND MACULAN, N. Global and local optimization approaches for channel assignment in wireless networks. In *Proceedings of the Global Optimization Workshop 2012* (2012).
- [78] RODRIGUES, R. F., DIAS, B. R. C., COELHO, F. J. M., OLIVEIRA, A. M., AND MACULAN, N. Special colorings for channel assignment in cellular mobile networks. In *Annals of 25th European Chapter on Combinatorial Optimization* (2012).
- [79] RODRIGUES, R. F., PESSOA, A., UCHOA, E., AND POGGI DE ARAGÃO, M. Heuristic algorithm for the parallel machine total weighted tardiness scheduling problem. *Relatórios de Pesquisa em Engenharia de Produção* (2008).
- [80] RODRIGUES, R. F., SANTOS, C., DIAS, B. R. C., LAVOR, C., MACULAN, N., AND SZWARCFITER, J. Some notes on Euclidean distance geometry and graph theory involving telecom, computer networks and molecular biology applications. In *20th Conference of the International Federation of Operational Research Societies (IFORS 2014)* (2014).
- [81] SALLAUME, S., MARTINS, S. L., OCHI, L. S., SILVA, W., LAVOR, C., AND LIBERTI, L. A discrete search algorithm for finding the structure of protein backbones and side chains. *International Journal of Bioinformatics Research and Applications* 9, 3 (2013), 261–270.
- [82] SANTOS, J. T. A. N., DIAS, B. R. C., AND RODRIGUES, R. F. Alocação de berços em terminais portuários: proposta de abordagem em teoria de escalonamento considerando restrições tarefa-máquina. In *Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional* (2012), Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Disponível em <http://www2.claiosbpo2012.iltc.br/pdf/105918.pdf>.

- [83] SAXE, J. B. Embeddability of weighted graphs in  $k$ -space is strongly NP-hard. In *Proceedings of 17th Allerton Conference in Communications, Control and Computing* (1979), pp. 480–489.
- [84] SIVARAJAN, K. N., MCELIECE, R. . J., AND KETCHUM, J. W. Channel assignment in cellular radio. In *IEEE 39th Vehicular Technology Conference* (1989), vol. 2, pp. 846–850.
- [85] SOUSA, M. P., LOPES, R. F., LOPES, W. T. A., AND ALENCAR, M. S. Redes Cognitivas: Um Novo Paradigma para as Comunicações Sem Fio. In *Minicursos do XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2010), Sociedade Brasileira de Computação (SBC), pp. 153–197.
- [86] SOUZA, M. J. F., OCHI, L. S., AND MACULAN, N. Minimizing earliness and tardiness penalties on a single machine scheduling problem with distinct due windows and sequence-dependent setup times. In *Proceedings of VI ALI-O/EURO Conference on Applied Combinatorial Optimization* (2008).
- [87] SZWARCFITER, J. L. *Grafos e Algoritmos Computacionais*. Elsevier, 1986.
- [88] TALBI, E.-G. *Metaheuristics: From Design to Implementation*. John Wiley & Sons, 2009.
- [89] TANENBAUM, A. S. *Computer Networks*, 4th ed. Prentice Hall, 2003.
- [90] TELECO. Estatísticas de Celulares no Brasil. <http://www.teleco.com.br/ncel.asp>, Janeiro 2014.
- [91] TRICK, M. Constraint Programming: A Tutorial. <http://mat.gsia.cmu.edu/trick/cp.ppt>.
- [92] VAN OMME, N., PERRON, L., AND FURNON, V. or-tools user’s manual. Tech. rep., Google, 2014.
- [93] YANG, J. Spatial channel characterization for cognitive radios. Tech. Rep. UCB/ERL M05/8, EECS Department, University of California, Berkeley,

2005. Disponível em <http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/4293.html>.

[94] ZHONG, J., AND LI, J. **Cognitive Radio Cognitive Network Simulator**. <http://stuweb.ee.mtu.edu/~ljialian>, 2009.

# Apêndice A

## Lista de publicações e eventos

1. “*Graph coloring problems with distance constraints: branch-prune-and-bound approach*”, em submissão para o periódico *Discrete Applied Mathematics (DAM)* - Edição Especial DGA 2013.
2. “*Simulated annealing for channel assignment in wireless mobile networks*”, em submissão para o *Journal of Global Optimization (JOGO)*.
3. “*Some notes on Euclidean distance geometry and graph theory involving telecom, computer networks and molecular biology applications*” [80], aceito para a *20th Conference of the International Federation of Operational Research Societies (IFORS 2014)*, a ser realizada de 13 a 18 de julho de 2014 em Barcelona, Espanha.
4. “*Simulated annealing para a alocação de canais em redes móveis celulares*” [31], trabalho completo submetido ao XLV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2013), realizado de 16 e 19 de setembro de 2013 em Natal - RN.
5. “*On graph coloring problems with distance constraints*” [32], submetido ao *Workshop on Distance Geometry and Applications (DGA 2013)*, realizado de 24 a 27 de junho de 2013 em Manaus - AM, e premiado como um dos cinco melhores do evento e apresentado na forma oral e de pôster.
6. “*A hybrid genetic algorithm with local search approach for E/T scheduling problems on identical parallel machines*” [7] (co-autor), pôster submetido a *15th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO 2013)*.

7. “Alocação de canais em redes celulares sem fio: algoritmos e modelos teóricos em grafos e escalonamento” [30], submetido para o XLIV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2012), realizado de 24 e 28 de setembro de 2012 no Rio de Janeiro - RJ. O trabalho foi escolhido como um dos quatro melhores submetidos para o Prêmio de Iniciação Científica do evento e selecionado para apresentação oral.
8. “Alocação de berços em terminais portuários: proposta de abordagem em teoria de escalonamento considerando restrições tarefa-máquina”[82] (co-autor), pôster submetido para o SBPO 2012.
9. “ILS com reconexão de caminhos entre ótimos locais para um problema clássico de escalonamento com antecipação e atraso”[6] (co-autor), trabalho completo submetido para o SBPO 2012.
10. “*Global and local optimization approaches for channel assignment in wireless networks*” [77], submetido para o *Global Optimization Workshop (GOW 2012)*, realizado de 26 a 29 de junho de 2012 em Natal - RN.
11. “*Global and local combinatorial optimization approaches for channel assignment in wireless networks*”, em submissão ao *Journal of Global Optimization (JOGO)*, Edição Especial do GOW 2012.
12. “*MIP models and algorithms for earliness/tardiness scheduling problems on parallel machines*” (co-autor), submetido a *25th European Conference on Operational Research [76]*.
13. “*Special colorings for channel assignment in cellular mobile networks*”, submetido ao *25th European Chapter on Combinatorial Optimization [78]*.
14. “Alguns Algoritmos para Escalonamento de Tarefas com Restrições”, submetido para a *XVI Escuela Latinoamericana de Verano de Investigación Operativa (ELAVIO 2012)*, realizada de 6 e 10 de fevereiro de 2012 em Bento Gonçalves - RS, como requisito para participação no evento.