

# Alocação de Freqüências para Telefonia Celular: Uma Solução Através do Procedimento de Busca Guloso Adaptativo Aleatório

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Carlos Alberto de Sousa Oliveira e aprovada pela Banca Examinadora.

Fortaleza, 24 de Março de 2000.

Prof. Dr. Fernando de Carvalho Gomes  
(Orientador)

Dissertação apresentada ao Mestrado em Ciência da Computação, UFC, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

**Alocação de Frequências para Telefonia Celular:  
Uma Solução Através do Procedimento de Busca  
Guloso Adaptativo Aleatório**

**Carlos Alberto de Sousa Oliveira<sup>1</sup>**

24 de Março de 2000

**Banca Examinadora:**

- Prof. Dr. Fernando de Carvalho Gomes (Orientador)
- Prof. Dr. Luiz Satoru Ochi
- Prof. Dr. Ricardo Cordeiro Corrêa

---

<sup>1</sup>Bolsista da FUNCAP (Fundação Cearense de Amparo à Pesquisa)

---

# Agradecimentos

- Aos meus pais e familiares.
- À minha noiva, Janaina.
- Ao meu orientador.
- Ao pesquisador PhD Maurício G.C. Resende (AT&T) por suas importantes sugestões para este trabalho.
- Ao Prof. Dr. Luiz Satoru Ochi que foi um dos participantes da banca e colaborou com importantes sugestões para a versão final do trabalho.
- Ao Prof. Dr. Ricardo Correa por sua ajuda tanto na fase inicial deste trabalho quanto em sua versão final.
- Aos meus colegas de mestrado.
- Aos professores do departamento de computação.
- Aos funcionários da UFC.
- Aos criadores do software gratuito usado para a confecção desta dissertação (emacs,  $\text{\LaTeX}$ , xfig, etc).

---

# Sumário

Agradecimentos	iii
Sumário	iii
Lista de Tabelas	vii
Lista de Figuras	viii
Lista de Algoritmos	xi
Abreviaturas	xii
<b>1 Introdução</b>	<b>2</b>
1.1 Otimização combinatória . . . . .	4
1.1.1 Problema do Caixeiro Viajante . . . . .	6
1.1.2 Problema da mochila . . . . .	7
1.1.3 Conjunto independente máximo e clique máxima . . . . .	8
1.2 Problemas em telefonia móvel . . . . .	10
1.2.1 Exemplos . . . . .	10
1.2.2 Histórico . . . . .	12
1.2.3 Definições . . . . .	13
1.2.4 Tecnologias . . . . .	20
1.3 Metaheurísticas . . . . .	23
1.3.1 Busca Tabu . . . . .	25
1.3.2 Têmpera Simulada . . . . .	27
1.3.3 Algoritmos Genéticos . . . . .	29
1.4 Organização da Dissertação . . . . .	33

<b>2</b>	<b>O Problema da Alocação de Frequências</b>	<b>34</b>
2.1	Introdução . . . . .	34
2.2	Definição do problema . . . . .	35
2.3	Complexidade . . . . .	37
2.4	PAF e coloração de vértices . . . . .	38
2.4.1	Definições de teoria dos grafos . . . . .	38
2.4.2	Heurísticas para coloração de grafos . . . . .	39
2.5	Trabalhos relacionados . . . . .	40
2.5.1	Algoritmos Exatos . . . . .	42
2.6	Algoritmos aproximativos . . . . .	45
2.6.1	Heurísticas com garantia de aproximação . . . . .	45
2.6.2	Heurísticas sem garantia de aproximação . . . . .	46
2.6.3	Metaheurísticas . . . . .	49
2.7	Formulações alternativas . . . . .	51
2.8	Conclusão . . . . .	51
<b>3</b>	<b>A Metaheurística GRASP</b>	<b>52</b>
3.1	Introdução . . . . .	52
3.2	Descrição . . . . .	53
3.2.1	Principais características . . . . .	53
3.2.2	Intensificação e diversificação no GRASP . . . . .	53
3.2.3	Construtor . . . . .	55
3.2.4	Busca Local . . . . .	56
<b>4</b>	<b>Proposta de um GRASP para o PAF</b>	<b>59</b>
4.1	Introdução . . . . .	59
4.2	Descrição . . . . .	60
4.2.1	Instância e função objetivo . . . . .	60
4.2.2	Construtor . . . . .	61
4.2.3	Busca local . . . . .	62
4.3	GRASP reativo . . . . .	64
4.4	GRASP e Religação de Caminhos . . . . .	66
<b>5</b>	<b>Resultados Computacionais</b>	<b>70</b>
5.1	Instâncias . . . . .	70

5.1.1	Geração de instâncias aleatórias . . . . .	71
5.1.2	Geração de instâncias com ótimo conhecido . . . . .	72
5.1.3	Instância Philadelphia . . . . .	75
5.2	Resultados . . . . .	76
5.2.1	Instâncias com ótimo conhecido . . . . .	77
5.2.2	Instâncias aleatórias e philadelphia . . . . .	78
<b>6</b>	<b>Conclusão</b>	<b>82</b>
<b>A</b>	<b>Listas de resultados para o PAF</b>	<b>84</b>
A.1	Introdução . . . . .	84
A.2	Grafos de reticulado . . . . .	85
A.3	GRASP . . . . .	87
	<b>Índice Remissivo</b>	<b>99</b>

---

# Lista de Tabelas

1.1	Alguns dos serviços reconhecidos pela ITU. . . . .	13
5.1	Parâmetros e valores usados no gerador de instancias para o PAF . . . .	72
5.2	Conjunto de instâncias do PAF usadas durante os experimentos. As colunas denominadas e, n e m representam, respectivamente, o número de estações base, o número total de antenas e o número de arestas do grafo de interferência. As outras colunas representam a densidade percentual do grafo (relação entre o número total de arestas e o número máximo de arestas) e o número de arestas com $d(i, j) = 1$ , $d(i, j) = 2$ , $c^{co}(i, j) > 0$ e $c^{ad}(i, j) > 0$ , respectivamente. . . . .	73
A.1	Resultados para grafos de reticulado do tipo 1. . . . .	85
A.2	Resultados para grafos de reticulado do tipo 2. . . . .	86
A.3	Resultados para grafos de reticulado do tipo 3. . . . .	86
A.4	Comparação de resultados computacionais para instancias do PAF. . . .	88
A.5	Comparação de resultados computacionais para instancias do PAF (con- tinuação). . . . .	89
A.6	Comparação de resultados computacionais para instancias do PAF (con- tinuação). . . . .	90
A.7	Comparação de resultados computacionais para instancias do PAF (con- tinuação). . . . .	91

---

# Lista de Figuras

1.1	Instância do problema do caixeiro viajante com 10 pontos. Duas possíveis soluções são apresentadas em 1) e 2). A distância total percorrida no caminho 1 é de 182 unidades, enquanto no caminho 2 é de 195 unidades.	6
1.2	Instância do problema do máximo conjunto independente. . . . .	9
1.3	Instância do problema da clique máxima. Neste caso, a clique máxima é subgrafo induzido pelos vértices $c, d, e$ e $f$ . . . . .	9
1.4	Um aparelho móvel desloca-se da célula $a$ para a célula $b$ , provocando um “handoff”. . . . .	12
1.5	Representação de uma região dividida em células. . . . .	14
1.6	O espectro de frequências e o exemplo de uma região, compreendida entre as frequências de 890MHz e 915MHz, dividida em canais $c_1, c_2, \dots, c_{20}$ . . . . .	15
1.7	Os transmissores $a$ e $b$ , que usam o mesmo canal, provocam interferência no aparelho móvel situado na região de influencia dos dois. . . . .	16
1.8	Interferência co-canal e adjacente-canal. Os transmissores $A$ e $B$ usam o canal $f_1$ . Os transmissores $C$ e $D$ usam, respectivamente, os canais $f_2$ e $f_3$ , que são adjacentes no espectro de frequências. . . . .	17
1.9	Grafo de interferência para um conjunto de 5 células. . . . .	18
1.10	Divisão de um canal em slots de tempo. . . . .	21
1.11	Contraste entre um mínimo local e o mínimo global de uma função. . . . .	25
1.12	Exemplo de cromossomo para o problema da mochila. Cada posição representa se o objeto correspondente será incluído (1) ou não (0). O tamanho do cromossomo representa o número de objetos disponíveis. . . . .	30
1.13	Metodo de seleção “roleta viciada”. Cada cromossomo tem probabilidade de ser escolhido proporcional a seu valor de adequação. . . . .	31

1.14	Operador de crossover em atuação para o problema da mochila. Do lado esquerdo, dois cromossomos. A linha pontilhada indica o ponto de crossover. Do lado direito o cromossomo resultante. . . . .	32
1.15	Exemplo de alteração realizada pelo operador de mutação para o problema da mochila. . . . .	32
2.1	Um exemplo de alocação de frequências . . . . .	37
2.2	Um grafo de reticulado triangular. . . . .	46
3.1	Lista de possíveis elementos de uma solução $(x_1, x_2, \dots, x_n)$ , ordenados de acordo com um critério de adequação. Os elementos da RCL $(x_1, \dots, x_4)$ são os que satisfazem o parâmetro $\alpha$ . . . . .	56
3.2	Exemplo de vizinhança para uma solução qualquer $s$ , durante a execução do algoritmo de busca descida da montanha. $s'$ é a solução vizinha de $s$ com menor valor de função objetivo. . . . .	57
4.1	Exemplo para o construtor do GRASP . . . . .	62
4.2	Exemplo para busca local. Os canais selecionados pelo algoritmo são marcado como $f_i, f_i + 4, f_i + 8$ e $f_i + 12$ e $f_i + 16$ . A separação entre canais neste caso é de 4. . . . .	64
4.3	Exemplo de resultado do algoritmo busca local. A estação base B foi selecionada pelo algoritmo e uma alocação em bloco é realizada. . . . .	65
4.4	Exemplo de Religação de Caminhos. A linha cheia representa o caminho original no espaço de busca. A linha pontilhada representa o novo caminho de religação das duas soluções. Eventualmente o novo caminho pode passar por uma solução melhor que a final. . . . .	68
5.1	Grafos de reticulado. Esses grafos podem ser coloridos, respectivamente, com apenas duas, três, ou quatro cores. . . . .	74
5.2	Possível solução para o primeiro grafo de reticulado da Figura 5.1. . . .	75
5.3	Matriz de distância mínima para a instância “philadelphia”. . . . .	76
5.4	Instância “Philadelphia”. . . . .	77
5.5	Gráfico da variação da função objetivo <i>versus</i> tempo para GRASP Reativo mais Religação de Caminhos para instâncias com 50, 60, 70, 80, 90, 100, 110 e 120 transmissores. . . . .	80

5.6	Comparação de valores de função objetivo para uma instância com 600 transmissores, usando diferentes métodos, <i>versus</i> tempo de execução. .	81
-----	--	----

---

# Lista de Algoritmos

1.1	Metaheurística Busca Tabu . . . . .	26
1.2	Metaheurística Têmpera Simulada . . . . .	28
1.3	Metaheurística Algoritmo Genético . . . . .	33
2.1	Algoritmo RLF . . . . .	40
2.2	Heurística T-Coloração . . . . .	47
2.3	Heurística Dual Guloso . . . . .	48
3.1	Metaheurística GRASP . . . . .	53
3.2	Procedimento de construção do GRASP . . . . .	55
3.3	Método descida da montanha. . . . .	58
4.1	Criação da RCL . . . . .	61
4.2	Metaheurística GRASP com Religação de Caminhos . . . . .	68
4.3	Religação de Caminhos . . . . .	69

---

# Abreviaturas

**GRASP** Greedy Randomized Adaptive Search Procedure. Metaheurística para problemas de otimização combinatória.

**AG** Algoritmos Genéticos.

**AMPS** Advanced Mobile Phone System. Sistema Avançado de Telefonia Móvel.

**BCA** Borrowing Channel Assignment. Método de alocação de frequências híbrido.

**BT** Busca Tabu.

**CDMA** Code Division Multiple Access. Tecnologia de transmissão celular que divide os canais por códigos em vez de frequências.

**D-AMPS** Digital Advanced Mobile Phone System. Sistema Avançado Digital de Telefonia Móvel, uma expansão do AMPS.

**FDMA** Frequency Division Multiple Access. Técnica de divisão do espectro de frequências em canais.

**ITU** International Telecommunications Unit. Unidade Internacional de Telecomunicações, um mundial órgão de regulamentação da área de telecomunicações.

**GSM** Global System for Mobile communications. Padrão europeu para comunicação móvel.

**TDMA** Time Division Multiple Access. Tecnologia que divide os canais de frequências em slots de tempo.

**TS** Têmpera Simulada.

## Abstract

The Frequency Assignment Problem (FAP) is NP-Hard, and arises from radio transmission networks. It plays an important role in cellular mobile systems. The FAP consists of a set  $V$  of antennas distributed in the area covered by the system, and a number  $n$  of frequencies. Since in real-world applications  $n < |V|$ , the system operator is obliged to reuse frequencies. This produces interference. As the number of users increases, the same happens to the total interference. Our aim is to globally minimize the total system interference. In this work we used a GRASP (Greedy Randomized Adaptive Search Procedure) to solve the problem. Moreover, Reactive GRASP, and Path Relinking were used respectively to tune the diversification phase, and to incorporate elements from good solutions. The proposed GRASP was compared against a greedy heuristic based on T-coloring, and a downhill local search. The experimental results indicate that the proposed GRASP performs well when applied to real-world and artificial instances.

## Resumo

O Problema da Alocação de Frequências (PAF) é um problema NP-Difícil que ocorre em sistemas de transmissão de informações através de ondas de rádio. Ele é importante na área de telefonia celular, onde o objetivo é reduzir ao mínimo a interferência total do sistema. O PAF consiste de um conjunto  $V$  de antenas, distribuídas pela área de cobertura do sistema, e um número  $n$  de frequências disponíveis. Como em instâncias do mundo real  $n < |V|$ , torna-se necessário o reuso de frequências em diversas antenas, produzindo interferência. O objetivo deste trabalho é minimizar a interferência total do sistema através do uso de um GRASP (Greedy Randomized Adaptive Search Procedure). Além disso, as metodologias GRASP Reativo e Religação de Caminhos são empregadas para, respectivamente, realizar a sintonia durante a fase de diversificação e para incorporar elementos de boas soluções já encontradas. O GRASP proposto foi comparado com uma heurística gulosa baseada no método da T-Coloração e com um algoritmo de Descida da Montanha. Os resultados experimentais indicam que o GRASP proposto oferece boas soluções quando aplicado a instâncias reais e artificiais do problema.

# Capítulo 1

---

## Introdução

*“It would be a mistake to think that solving problems is a purely ‘intellectual affair’; determination and emotions play an important role.”*  
(Polya, 1988)

Nos últimos anos tem-se visto, no decorrer de pouco tempo, o surgimento de várias tecnologias importantes. Entre elas podem ser citadas as telecomunicações móveis, a microinformática e a Internet. O principal ponto em comum entre estas tecnologias e o que as diferencia de tendências tecnológicas anteriores, é que elas estão permitindo uma revolução na forma como os seres humanos se comunicam e colaboram entre si. Com ferramentas tão poderosas e simples, pode-se realizar trabalhos que exigem altos níveis de colaboração sem a necessidade de interação física.

O resultado destas inovações tecnológicas é o surgimento de um paradigma diferente para o desenvolvimento econômico. Elas estão criando novas perspectivas para a economia, uma nova organização social baseada na comunicação globalizada e até mesmo uma mudança nos costumes de diversão e consumo.

No caso das telecomunicações móveis, o padrão de crescimento experimentado tem sido sem precedentes. Iniciando como um método especial de transmissão e restrito a um pequeno número de usuários nos anos 80, a telefonia móvel já é uma realidade em quase todo o mundo. Nos países desenvolvidos, o elevado número de usuários tem impulsionado o surgimento de grandes companhias que exploram tanto o fabrico de aparelhos quanto o provimento de serviços.

Como consequência das mudanças surgidas na área de telecomunicações, alguns recursos, que antes eram de uso restrito, têm-se tornado indispensáveis, sendo até mesmo um fator limitante para o desenvolvimento tecnológico. Em sistemas de telefonia con-

vencionais fixos, voz e informações trafegam através de fios de cobre. Em contraste, a telefonia móvel usa ondas de rádio para a transmissão de informações. O fato de usar um meio de transmissão naturalmente disponível, permite que seja implementada de forma rápida e a um preço competitivo em comparação com a telefonia convencional fixa. Entretanto, o espectro de frequências é um recurso limitado e dependendo do modo como é usado, pode rapidamente se esgotar. Dado que ocupa uma posição tão central nesta tecnologia, percebe-se a importância do gerenciamento adequado do espectro de frequências.

Alguns fatores têm contribuído para amenizar o problema do aumento de demanda do espectro de frequências. Um desses fatores foi a transmissão digital de dados. Esta nova tecnologia permitiu um aumento substancial na capacidade, qualidade e eficiência dos serviços, através da codificação binária das informações e da melhoria dos equipamentos usados. Mesmo assim, a demanda crescente torna o problema do uso racional do espectro um dos mais importantes para a garantia da expansão futura do sistema.

O uso do espectro de frequências pode tornar-se crítico para aplicações futuras. Como exemplo, podem ser citadas as aplicações de multimídia e teleconferência. Pelo fato de fazerem amplo emprego de som, vídeo e outros conteúdos, estas aplicações precisam transmitir rapidamente grandes quantidades de informação. Portanto, para que funcionem adequadamente, precisarão usar o espectro de uma forma muito mais otimizada. Ademais, mesmo o uso de telefonia celular somente para a transmissão de voz, quando se trata de cobertura de grandes áreas, para muitos usuários, se constitui importante problema para as operadoras.

Nesta dissertação, são tratados problemas ligados à otimização do uso do espectro de frequências em telefonia móvel. Mais especificamente, será discutido o problema de alocação de frequências. Este capítulo tem o objetivo de fazer uma introdução geral ao assunto e encontra-se organizado como segue. Na Seção 1.2 far-se-á uma exposição dos problemas encontrados na área de telefonia móvel. Logo após, a Seção 1.1 trata-se de uma introdução aos problemas de otimização combinatória. Depois, será examinado de forma introdutória, na Seção 1.3, o método de solução para problemas combinatórios que será proposto neste trabalho. Finalmente, na Seção 1.4, será apresentado um plano geral da dissertação.

## 1.1 Otimização combinatória

Em certos problemas que ocorrem em vários setores, como indústria, transportes, administração, etc., uma série de recursos precisam ser usados, de forma a cumprir um objetivo pré-especificado. Por exemplo, pode-se querer produzir um bem usando a menor quantidade de matéria prima. Pode-se estar interessado em oferecer produtos com a melhor qualidade por um preço fixo. Como último exemplo, pode-se querer determinar a localização de hospitais munidos de ambulâncias, de forma a atender a população no menor tempo possível. Todos estes problemas tem em comum o fato de possuir objetivos que estão sujeitos a restrições. Problemas com estas características são classificados como *problemas de otimização*.

Um problema de otimização em sua forma mais genérica pode ser estabelecido como um conjunto de equações e inequações da forma:

$$\begin{aligned} &\text{otimizar } f(x) \\ &\text{sujeito a: } g_i(x) \geq 0, \quad i = 1, \dots, m \\ &\quad h_j(x) = 0, \quad j = 1, \dots, n \end{aligned}$$

Nesta série de fórmulas,  $f(x)$  é chamada de *função objetivo*, e representa o valor que deve ser *otimizado* (maximizado ou minimizado). Já as funções  $g_i(x)$ ,  $i = 1, \dots, m$  e  $h_j(x)$ ,  $j = 1, \dots, n$  são chamadas *restrições do problema* e especificam que propriedades deve possuir uma solução aceitável. Define-se o conceito de *solução viável* para um problema de otimização como um conjunto de valores  $a_1, a_2, a_3, \dots$  para as variáveis  $x_1, x_2, x_3, \dots$  que aparecem nas fórmulas, tal que  $g_i(x)$ ,  $i = 1, \dots, m$  e  $h_j(x)$ ,  $j = 1, \dots, n$  sejam satisfeitas.

Uma instância de um problema de otimização  $\Pi$  é um par  $(C, f)$ , onde  $C$  é o conjunto de todas as soluções viáveis para  $\Pi$  e  $f: C \rightarrow \mathbb{R}$  é a função objetivo. Então um problema de otimização pode ser definido alternativamente como encontrar um elemento  $x \in C$  tal que  $f(x)$  seja o menor (ou maior) possível.

Dependendo da forma das funções  $f$ ,  $g_i$  e  $h_j$ , pode-se ter vários tipos de problemas de otimização. Os casos mais importantes ocorrem quando a função objetivo e as restrições são

1. *Convexas*, ou seja, possuem apenas um mínimo (ou máximo) local;
2. *Lineares*, ou seja, são formadas apenas pela soma de múltiplos de variáveis.

As variáveis que aparecem nas funções  $f$ ,  $g_i$  e  $h_j$  são também chamadas de *variáveis de decisão*, pois definem o valor da função objetivo.

Os problemas de otimização que possuem apenas funções lineares também são conhecidos como problemas de *programação linear*. Eles podem ser representados da seguinte forma:

$$\begin{aligned} & \text{minimizar} && \sum_{i=1}^n c_i x_i \\ & \text{sujeito a:} && \sum_{i=1}^n A_{j,i} x_i = b_j, \text{ para } j = 1, \dots, m \\ & && x_i \geq 0, \text{ para } i = 1, \dots, n \end{aligned}$$

onde  $A_{j,i}$  representa o elemento de uma matriz de pesos,  $x_i$  é uma variável de decisão,  $b_j$  e  $c_i$  são elementos de vetores com  $m$  e  $n$  constantes, respectivamente.

Um importante tipo de problema de otimização ocorre quando as variáveis de decisão são restritas a valores inteiros. Um problema deste tipo envolve uma quantidade inteira (finita ou infinita) de objetos. Neste caso o problema é chamado de *otimização combinatória* ou *discreta*. Estes problemas têm sido amplamente estudados devido à sua aplicabilidade em situações reais.

Os problemas de otimização combinatória variam em dificuldade e podem ser classificados de acordo com o número de instruções necessárias para sua resolução. Alguns destes problemas, como por exemplo, os de fluxo em rede, possuem procedimentos de solução que podem ser executados em tempo polinomial em relação ao tamanho da entrada. Em outras palavras, o tempo de execução pode ser expresso como um polinômio, onde a variável representa o número de bits necessários para a codificação dos dados do problema (Papadimitriou e Steiglitz, 1982). Entretanto, para uma classe grande de problemas de otimização combinatória, nenhum algoritmo com eficiência polinomial foi encontrado até hoje. Esta classe de problemas é conhecida como NP-Difícil e encontrar soluções ótimas para seus elementos representa um dos maiores desafios teóricos para a Ciência da Computação nos nossos dias (veja (Garey e Johnson, 1979)).

Com a finalidade de melhor ilustrar os problemas de otimização combinatória, serão apresentados a seguir alguns exemplos clássicos. São eles o Problema do Caixeiro Viajante (PCV – Seção 1.1.1), o problema da mochila (Seção 1.1.2) e os problemas do máximo conjunto independente e da clique máxima (Seção 1.1.3).

### 1.1.1 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) é um dos problemas de otimização combinatória mais estudados até hoje. Informalmente, ele pode ser enunciado da seguinte forma: dado um conjunto de  $P$  de pontos num plano, encontrar uma seqüência  $\{p_1, p_2, \dots, p_n\} \subset P$  de pontos que serão visitados uma única vez retornando ao ponto inicial  $p_1$ , de forma que a distância total percorrida seja mínima. Se encararmos os pontos como cidades, a solução para este problema indica o melhor caminho que um vendedor poderia utilizar para visitar todas as cidades (veja Figura 1.1), sendo este o contexto original do problema. Na Figura 1.1 são apresentados dois possíveis caminhos (1 e 2), sendo que 1 possui distância total menor que 2.

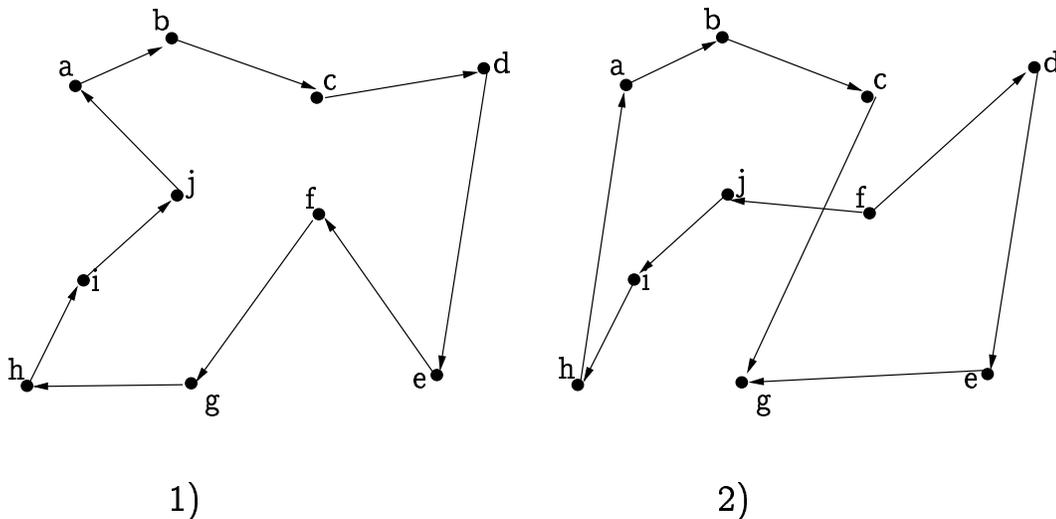


Figura 1.1: Instância do problema do caixeiro viajante com 10 pontos. Duas possíveis soluções são apresentadas em 1) e 2). A distância total percorrida no caminho 1 é de 182 unidades, enquanto no caminho 2 é de 195 unidades.

Uma definição mais formal pode ser encontrada em (Dantzig *et al.*, 1954). São dadas as distâncias entre cada par de pontos  $i$  e  $j$ , representados por  $c_{i,j} \in \mathbb{R}$ . Para modelar o problema, são necessárias variáveis de decisão  $x_{i,j}$  que indicam se o ponto  $j$  é visitado imediatamente após o ponto  $i$  ( $x_{i,j} = 1$ ) ou não ( $x_{i,j} = 0$ ), na seqüência de visitação. Então tem-se a seguinte formulação:

$$\text{minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j} \quad (1.1)$$

$$\text{sujeito a } \sum_{j=1}^n x_{i,j} = 1, \text{ para } i = 1, \dots, m \quad (1.2)$$

$$\sum_{i=1}^n x_{i,j} = 1, \text{ para } j = 1, \dots, m \quad (1.3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{i,j} \leq |S| - 1, \text{ para todo } S \subset V \quad (1.4)$$

$$x_{i,j} \in \{0, 1\} \quad (1.5)$$

onde  $n = |P|$  é o número de pontos e  $S$  é um subconjunto próprio de  $P = \{1, \dots, n\}$ . A função 1.1 estabelece o objetivo, que é o de minimizar a distância total. As equações 1.2 e 1.3 garantem que cada ponto possui exatamente uma conexão chegando e uma saindo. Finalmente, a equação 1.4 garante que a solução não possui ciclos desconexos.

As aplicações nas quais aparece o PCV são bastante variadas. Entre elas, podem ser citadas a construção de circuitos integrados, seqüenciamento de tarefas (Laporte, 1992) e roteamento de veículos (Christofides, 1985).

Apesar de parecer um problema simples à primeira vista, nenhum algoritmo eficiente foi criado até hoje para o PCV. O algoritmo mais simples para o problema seria verificar cada uma das possíveis soluções e usar a de menor tamanho. Entretanto, partindo de qualquer ponto, existem  $\frac{(n-1)!}{2}$  possíveis caminhos. Cada novo ponto acrescentado representa um aumento enorme a esta cifra. Por exemplo, mesmo um problema com apenas 10 pontos possui mais de 180 mil opções que precisam ser computadas, tornando este algoritmo inadequado na prática.

### 1.1.2 Problema da mochila

No problema da mochila, dispõe-se de um volume de  $c$  unidades, que corresponde à capacidade da mochila. Dentro desse volume pode-se colocar qualquer objeto de um conjunto  $X = \{x_1, x_2, \dots, x_n\}$  com  $n$  objetos. Cada objeto possui volume  $t_i$ ,  $i = 1, \dots, n$  e valor  $p_i$ ,  $i = 1, \dots, n$ . O objetivo do problema é achar uma configuração de objetos cujo volume total seja menor ou igual a  $c$  e tenha valor máximo. Matematicamente pode-se enunciar o problema como:

$$\text{maximizar } \sum_{i=1}^n p_i x_i \quad (1.6)$$

$$\text{sujeito a: } \sum_{i=1}^n t_i x_i \leq c, \text{ para } j = 1, \dots, m \quad (1.7)$$

$$x_i \in \{0, 1\}, \text{ para } i = 1, \dots, n \quad (1.8)$$

onde cada  $x_i$  é uma variável de decisão que tem valor 1 se o objeto  $i$  estiver na mochila ou 0 caso contrário.

As aplicações práticas em que o problema da mochila aparece encontram-se principalmente na indústria, em diversos setores de manufatura e em problemas de tomada de decisão. Por exemplo, na administração de uma empresa, pode-se querer modelar uma situação em que dispõe-se de um valor fixo para executar melhorias em uma indústria. Cada uma destas melhorias terá um custo definido. Entretanto deseja-se maximizar o retorno que a empresa terá com estes investimentos. Neste caso os investimentos farão o papel do conjunto  $\{x_1, \dots, x_n\}$  de objetos, cada um possuindo um custo e um valor de retorno. Ao final o custo total dos investimentos não pode exceder o montante disponível, que corresponderá ao valor máximo  $c$ .

### 1.1.3 Conjunto independente máximo e clique máxima

O problema do conjunto independente máximo (PCIM) é típico dos que ocorrem em teoria dos grafos (como será visto na Seção 2.4). Um *grafo*  $G(V, E)$  é formado por um conjunto de  $V = \{v_1, \dots, v_n\}$  de *vértices* e um conjunto  $E = \{\{v_{i_1}, v_{j_1}\}, \dots, \{v_{i_m}, v_{j_m}\}\}$  de pares de vértices de  $V$ . Os pares não ordenados de vértices em  $\{v_i, v_j\} \in E$  também são chamados de *arestas*. Dado um grafo  $G(V, E)$ , deseja-se encontrar o subconjunto máximo de vértices  $V' \subset V$  tal que para cada par de vértices  $(x, y) \in V'$ , não existe aresta entre  $x$  e  $y$ , ou seja,  $(x, y) \notin E$ . Na Figura 1.2 tem-se uma instância do problema. Os vértices na cor preta representam um conjunto independente de vértices.

O PCIM é usado na modelagem de situações em que deseja-se encontrar um conjunto máximo de elementos que não possuam certa relação entre si. Por exemplo, achar, em um grupo qualquer de pessoas, o máximo conjunto daquelas que não se conhecem é um caso do PCIM.

Um problema estreitamente relacionado ao PCIM é o conhecido problema da clique máxima (PCM) (Garey e Johnson, 1979). O PCM também é NP-Difícil e seu objetivo é, dado um grafo  $G(V, E)$ , encontrar o maior subconjunto  $V' \in V$  totalmente conexo. Ou seja, para cada par de vértices  $(x, y) \in V'$ , existe uma aresta  $(x, y) \in E$  no grafo  $G$ . Na Figura 1.3 vê-se um exemplo de clique, formada pelos vértices  $c, d, e$  e  $f$ .

As aplicações do PCM pode ser encontradas nas mais diversas situações. Um dos muitos exemplos que podem ser citados seria, em uma empresa de telefonia, encontrar

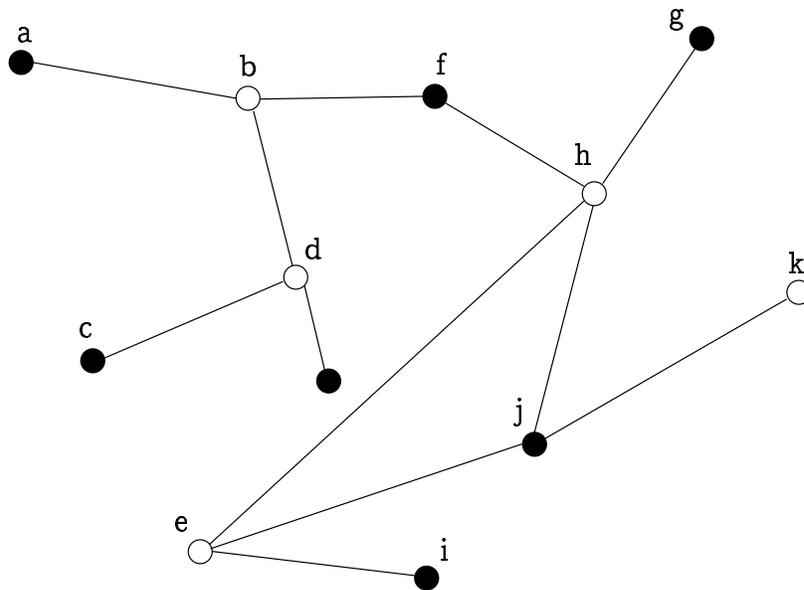


Figura 1.2: Instância do problema do máximo conjunto independente.

o maior grupo de usuários que fizeram ligações entre si no último mês. Neste caso, cada usuário é representado por um vértice do grafo. As arestas, por sua vez, existiriam entre cada par de usuários toda vez que houvesse uma ligação registrada entre eles no período especificado.

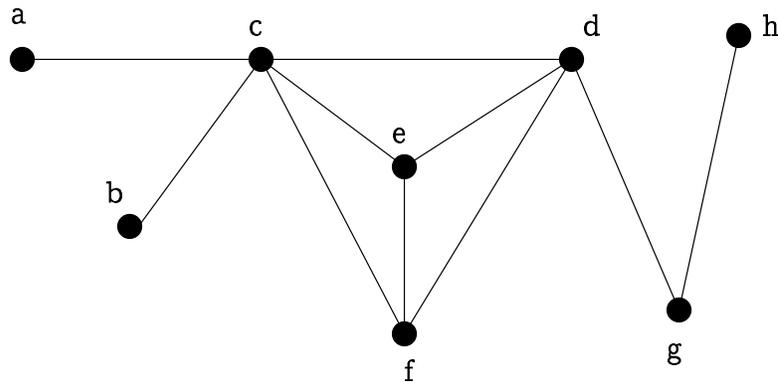


Figura 1.3: Instância do problema da clique máxima. Neste caso, a clique máxima é subgrafo induzido pelos vértices c, d, e e f.

O que relaciona os problemas da clique máxima e do conjunto independente máximo é o fato de que eles possuem características complementares. Dado o grafo  $G(V, E)$ , define-se o complementar de  $G$ , denominado  $\overline{G}(V, E')$ , como sendo um grafo que possui

o mesmo conjunto de vértices  $V$ , mas com arestas  $(x, y) \in E'$  se e somente se  $(x, y) \notin E$ . Com esta definição, pode-se observar que, achando a clique máxima de um grafo  $G$ , também será encontrado o conjunto máximo independente do grafo  $\overline{G}$ , o que implica que os dois problemas possuem a mesma complexidade (Garey e Johnson, 1979).

## 1.2 Problemas em telefonia móvel

Nesta seção serão discutidas várias noções relacionadas a problemas de telefonia móvel. Elas serão apresentadas com o objetivo de permitir o correto entendimento dos problemas que serão tratados nesta dissertação. Os temas que serão tratados são os seguintes. Em primeiro lugar, na Seção 1.2.1 serão apresentados exemplos de alguns problemas que ocorrem na área de telefonia móvel. Na Seção 1.2.2, encontra-se um breve resumo sobre a história do desenvolvimento dos sistemas de telefonia móvel celular. A Seção 1.2.4 tem como objetivo oferecer algumas informações sobre as principais tecnologias existentes e suas características. Finalmente, na Seção 1.2.3 serão mostrados conceitos de telefonia celular que serão usados nesta dissertação.

### 1.2.1 Exemplos

Muitas das dificuldades associadas à telefonia móvel permanecem como desafios à pesquisa. Em sua maior parte, eles estão relacionados a problemas de combinatória e implicam no que é conhecido como explosão de possibilidades. Para muitos desses problemas, não existem algoritmos eficientes e que dêem uma resposta exata e portanto novos métodos para o tratamento de instâncias grandes do problema devem ser pesquisados.

Na área de telefonia móvel, problemas NP-Difíceis podem ser identificados em todas as fases: projeto, implementação e gerenciamento. A seguir, serão apresentados alguns exemplos destes problemas. São eles os problemas de localização de estações base, gerenciamento de potência e deslocamento de aparelhos móveis<sup>1</sup> entre diferentes células (veja (Mateus e Loureiro, 1998)).

---

<sup>1</sup> “hand-off” em inglês.

### Localização de estações base

O problema de *localização de estações base* consiste em determinar onde elas devem ser construídas de forma a otimizar vários objetivos, como o custo do terreno onde a EB será instalada, a potência usada, o número de usuários atendidos e custos de manutenção. Também devem ser levadas em conta várias restrições, como por exemplo, a distância máxima e mínima entre duas estações base, a potência mínima aceitável, etc. Além disso, a localização das estações base tem influência em um grande número de atividades, como por exemplo, o cálculo de sinais em diferentes pontos da área de estudo. Esse cálculo deve considerar obstáculos existentes, como o relevo da região, construções, etc.

### Gerenciamento de Potência

O problema de *gerenciamento de potência* é outro que faz parte do projeto e implementação de qualquer sistema de telefonia móvel. Dado um conjunto de transmissores, deve-se decidir a potência que deve ser usada de modo a reduzir interferências e permitir uma qualidade de serviço satisfatória. Se a potência usada for muito baixa, os aparelhos móveis podem sofrer falhas de sinal. Caso a potência seja muito alta, os transmissores passarão a sofrer interferências indesejáveis. Para alcançar o valor ideal, vários parâmetros, como distância entre transmissores, frequências usadas, etc., precisam ser levados em consideração.

### Deslocamento entre células

Outro problema bastante difícil nesta área é relacionado com o deslocamento de aparelhos entre células, também conhecido pelo termo “handoff”. Em um sistema de telefonia móvel, uma célula corresponde à região servida por um conjunto de antenas. Na Figura 1.4, o aparelho móvel passa da célula a, para a célula b. Quando isso acontece, é preciso que uma série de medidas sejam tomadas, envolvendo todo o sistema. Seria necessário, por exemplo, alocar uma frequência livre na antena vizinha, fazer com que o tráfego de informações flua para a nova antena, etc. Pode-se definir uma série de objetivos para este problema, como minimizar o tempo de total para que a comunicação possa ser restabelecida, minimizar a interferência gerada pela mudança, etc. Fazer com que essa mudança aconteça da forma mais transparente para o usuário, sem perda de informações e de uma forma suave para o sistema como um todo, é um grande desafio

para projetistas de sistemas celulares.

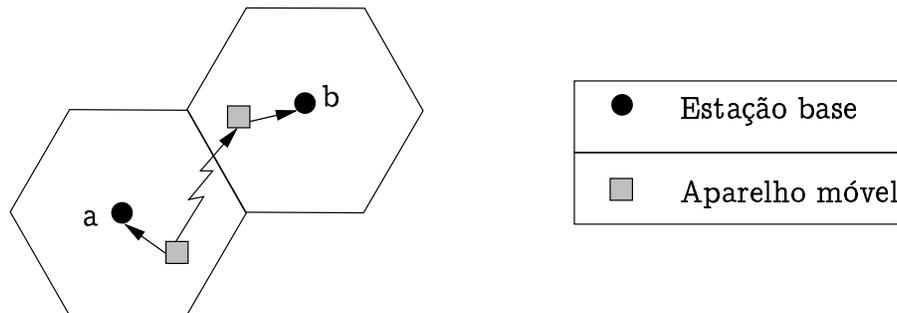


Figura 1.4: Um aparelho móvel desloca-se da célula a para a célula b, provocando um “handoff”.

### 1.2.2 Histórico

A história do uso de ondas eletromagnéticas para a transmissão de informações teve início com os estudos do físico alemão Heinrich R. Hertz (1857-1894). Baseando-se na teoria eletromagnética criada por James Clerk Maxwell (1831-1879), Hertz foi o primeiro a criar ondas eletromagnéticas de forma artificial. Os primeiros serviços que se utilizaram da propagação de ondas eletromagnéticas foram os de telégrafo. A partir disso, os serviços de rádio, televisão, comunicação marítima, radioamadorismo, comunicação das forças armadas, satélites, entre outros, têm feito uso do espectro de frequências para a realização de suas transmissões.

Para racionalizar o uso do espectro de frequências, a comunidade internacional criou a ITU (International Telecommunications Unit – Unidade Internacional de Telecomunicações) em 1934. Derivada da antiga International Telegraph Union (União Telegráfica Internacional) fundada em 1865, a ITU é desde 1947 uma agência das Nações Unidas. Seu principal objetivo é cooperar com as diversas nações e estados-membros para a alocação de frequências a determinados serviços. Isso evita que países diferentes usem legislações distintas o que causaria muita confusão, pois o espectro de frequências não é restrito a fronteiras geográficas. Alguns dos serviços definidos pela ITU encontram-se relacionados na Tabela 1.1.

A telefonia celular enquadra-se, de acordo com a ITU, no serviço móvel terrestre, que engloba toda a comunicação feita em terra através de aparelhos móveis. Outras

Fixo Aeronáutico	Móvel Aeronáutico
Satélite-móvel Aeronáutico	Radionavegação Aeronáutica
Radionavegação Aeronáutica – satélite	Rádio Amador
Rádio Amador – satélite	Transmissão de TV e rádio
Transmissão de TV e rádio – satélite	Móvel Terrestre
Satélite Móvel Terrestre	Satélite Meteorológico
Radioastronomia	Radionavegação
Operações Espaciais	Tempo Padrão
Telecomando	Telemetria

Tabela 1.1: Alguns dos serviços reconhecidos pela ITU.

faixas de espectro são alocadas para a comunicação marítima e aérea, bem como para todos os outros serviços estabelecidos, como radioamadorismo, transmissão de TV, etc.

A telefonia celular surgiu como um serviço de comunicação exclusivo para automóveis, no fim da década de 70. Além do alto custo, os aparelhos não tinham autonomia, dependendo da bateria dos automóveis para funcionar. A primeira geração usava uma tecnologia analógica, conhecida por AMPS (*Advanced Mobile Phone System*).

No final dos anos 80 houve um grande impulso para a difusão dos serviços de telefonia celular com o desenvolvimento de novos tipos de bateria, o que propiciou o surgimento de aparelhos menores e com maior autonomia. O desenvolvimento do GSM (*Global System for Mobile Communications*) também teve conseqüências positivas, permitindo a criação de uma nova geração de tecnologias digitais.

Atualmente, a área de telefonia móvel celular encontra-se numa fase de rápido crescimento, marcada pela disputa de grandes empresas pelo controle do mercado. Isso tem gerado um grande número de inovações nos serviços e na capacidade do sistema. A tendência atual é a integração entre a comunicação de voz e dados, criando uma estrutura que interligará computação móvel e Internet (Mateus e Loureiro, 1998).

### 1.2.3 Definições

#### Células e estações-base

Uma rede de telefonia celular é formada por dois elementos principais: antenas receptoras e aparelhos móveis. As antenas receptoras são localizadas em pontos estratégicos conhecidos como *estações base*.

A área de cobertura do sistema telefônico é dividida em várias regiões denominadas células. O formato e a extensão das células são definidos de acordo com a geografia da região e características da tecnologia que deverá ser empregada. O método mais usado para a divisão de uma região é a construção de células hexagonais. Entretanto outras formas de divisão em células podem ser adotadas. Por exemplo, em cidades com grandes edifícios, a divisão em quadras é mais efetiva, devido ao grande número de construções altas, organizadas em bloco. Já para a cobertura celular de estradas, uma distribuição linear das estações base é suficiente. Pode-se também, de forma equivalente, caracterizar uma célula como a área de cobertura de uma estação base. Na Figura 1.5 tem-se uma representação esquemática de células e suas respectivas áreas de cobertura. Note que existe interseção entre a área de cobertura de células adjacentes.

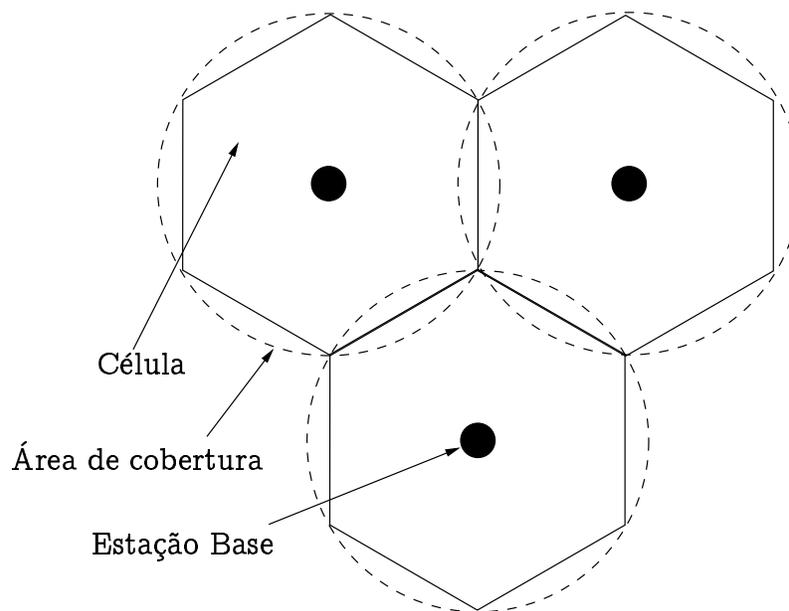


Figura 1.5: Representação de uma região dividida em células.

### Espectro de frequências

As ondas de rádio são parte do espectro de ondas eletromagnéticas. O espectro eletromagnético estende-se desde as ondas de baixa frequência até os raios X, passando pela luz visível. As ondas de rádio formam a região normalmente reservada para fins de comunicação.

A faixa do espectro regulamentada para a telefonia móvel, também é conhecida

como *banda*. Por exemplo, na Grã-Bretanha, a faixa de espectro dedicada à telefonia celular vai de 890 a 915MHz e de 935 a 960MHz (Periannan, 1996). Esta banda é dividida ainda em regiões menores, chamadas canais de frequências, ou simplesmente *canais* (veja Figura 1.6). Um canal pode ser identificado por sua frequência central. Portanto, é comum referir-se na literatura especializada ao “canal de 600 MHz”, quando na verdade refere-se ao canal em que esta é a frequência central.

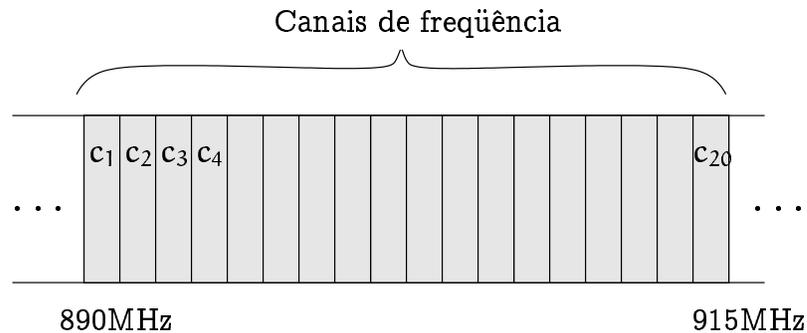


Figura 1.6: O espectro de frequências e o exemplo de uma região, compreendida entre as frequências de 890MHz e 915MHz, dividida em canais  $c_1, c_2, \dots, c_{20}$

Como a quantidade de canais disponíveis é limitada, pode-se alocar a cada canal um único número  $c_i \in \mathbb{N}$ . Pode-se dizer então que o espectro de canais disponíveis é um conjunto  $C \subset \mathbb{N} = \{c_1, \dots, c_n\}$ . Deste modo, ao fazer a alocação de canais a antenas, não é preciso referir-se diretamente aos canais e sim a números inteiros, o que facilitará em muito a formalização do problema nas seções seguintes.

Cada estação base no sistema de telefonia celular é responsável pelo atendimento de todos os usuários que se encontram na região alcançada pela célula correspondente. Como a demografia de cada região é diferente, as estações base também possuem demandas que variam.

Define-se *demanda* de uma estação-base como o número de canais de frequência necessários para o serviço de cobertura de uma célula. Quanto maior o número de usuários que se deseja atender em uma célula, maior a demanda.

### Tipos de interferência

Interferência é um fenômeno que ocorre em sistemas de transmissão via ondas de rádio. Ela existe quando um receptor está exposto a transmissões de duas fontes operando em frequências iguais ou muito próximas (veja Figura 1.7). Em geral a inter-

ferência impede a comunicação adequada entre transmissores e receptores ou faz com que ruídos indesejáveis prejudiquem esta comunicação. Uma medida da qualidade de uma ligação via ondas de rádio é o nível de interferência existente entre seus elementos. Deve-se salientar que a interferência ocorre apenas nos receptores de uma transmissão e é desse ponto de vista que deve ser mensurada.

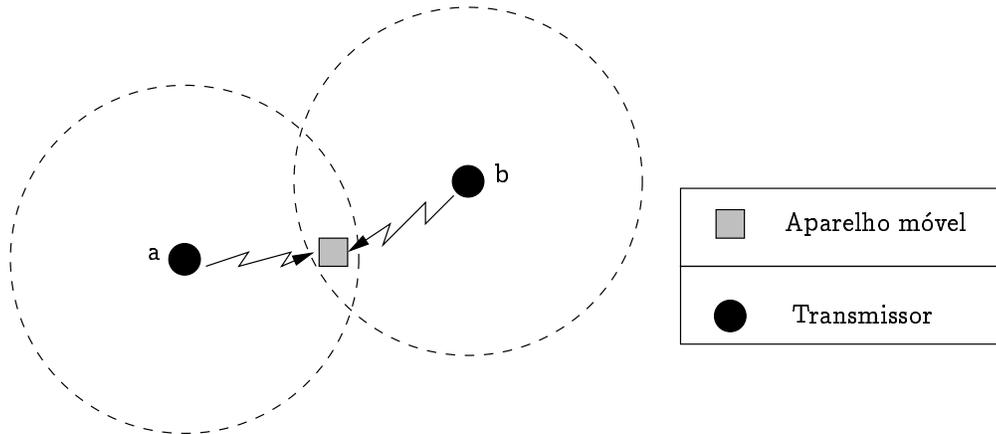


Figura 1.7: Os transmissores a e b, que usam o mesmo canal, provocam interferência no aparelho móvel situado na região de influência dos dois.

Na telefonia móvel, o problema da interferência ocorre principalmente devido à limitação do espectro disponível. Por exemplo, quando o número de antenas é maior que o número de canais, o que ocorre na maioria dos casos, alguns canais terão que ser usados em mais de uma antena. Entretanto, é preciso que este reuso seja feito de modo que a interferência resultante seja aceitável.

Os tipos de interferência são classificados de acordo com a proximidade dos transmissores. *Interferência local* (co-site) é a que ocorre entre transmissores que estão situados no mesmo lugar. Caso contrário a interferência é denominada *remota* (far-site). Entre os tipos de interferência remota, os mais importantes são a *interferência co-canal* e a *adjacente-canal*.

**Interferência local** Este tipo de interferência ocorre apenas com transmissores que situam-se na mesma posição geográfica. No caso da telefonia celular, existe interferência local entre cada par de antenas transmissoras de uma estação base.

**Interferência remota** Este tipo de interferência pode existir entre dois quaisquer transmissores que estejam geograficamente separados. Pode-se classificar interferências remotas de acordo com as frequências usadas pelos transmissores. Por exemplo, quando dois transmissores usam o mesmo canal, isso pode resultar em uma forte interferência, sentida por seus receptores. Este tipo de interferência é denominada *co-canal* (Figura 1.7). Uma interferência em menor intensidade ocorre quando dois transmissores usam canais adjacentes no espectro, ou seja, quando a diferença  $|y(a) - y(b)|$  entre os canais atribuídos a  $a$  ( $y(a)$ ) e  $b$  ( $y(b)$ ) é igual a 1. Os valores destas interferências são encontrados de forma empírica para cada par de transmissores do sistema e tabelados na matriz  $C_{i,j}$  para a interferência co-canal e  $A_{i,j}$  para adjacente-canal.

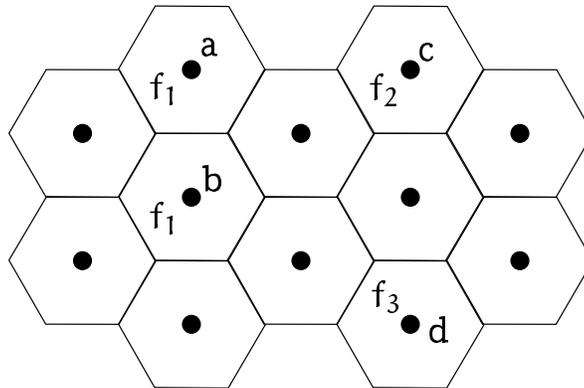


Figura 1.8: Interferência co-canal e adjacente-canal. Os transmissores A e B usam o canal  $f_1$ . Os transmissores C e D usam, respectivamente, os canais  $f_2$  e  $f_3$ , que são adjacentes no espectro de frequências.

Na Figura 1.8 vê-se um exemplo em que ocorrem interferências co-canal e adjacente-canal. As antenas a e b possuem o mesmo canal, já as antenas b e c possuem canais adjacentes. A interferência entre a e b é dada por  $C_{a,b}$  e entre b e c por  $A_{b,c}$ .

### Separação mínima entre canais

A quantidade de interferência gerada das diversas formas mostradas acima podem ter diferentes impactos no funcionamento dos sistemas de telefonia. A interferência pode gerar simplesmente ruídos na transmissão. Entretanto, quando ultrapassa um certo limite, toda a transmissão pode ser prejudicada. Alguns fatores que influenciam o limite de interferência aceitável são: a sensibilidade dos aparelhos, a capacidade de

filtrar frequências indesejáveis, a distância entre os transmissores, etc. Tais limites são definidos de forma empírica de modo a garantir o correto funcionamento do sistema de telefonia móvel.

Para evitar que a interferência ultrapasse o limite aceitável, define-se entre cada par de antenas  $a, b$  um valor  $d(a, b)$  que representa a distância mínima entre os canais atribuídos a estas antenas. Por exemplo, na Figura 1.9 é apresentado um conjunto de células, com restrições de separação entre as estações base representadas por arestas. Quando não existe arestas entre duas estações base isso significa que não há restrição alguma. Portanto, os transmissores situados em  $b$  e  $c$  poderiam receber quaisquer canais disponíveis. O grafo cujas arestas representam a separação entre canais é chamado de *grafo de interferência*.

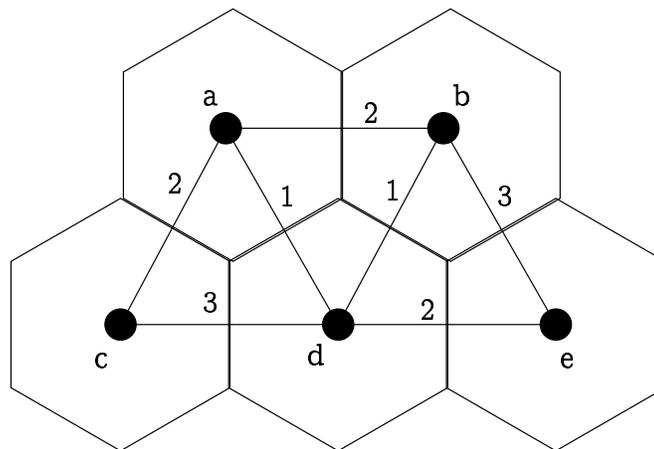


Figura 1.9: Grafo de interferência para um conjunto de 5 células.

### Tipos de alocações de frequências

Uma *alocação* é a especificação de canais de frequências para cada um dos transmissores de um sistema de telefonia celular. Dependendo da forma como é realizada, uma alocação de frequências pode ser de dois tipos: estática ou dinâmica.

Uma *alocação estática* de frequências é aquela realizada apenas quando a rede é projetada ou modificada. Os algoritmos de alocação estática podem ser, portanto, mais elaborados além de poderem ser executados durante um tempo relativamente longo, pois só serão usados poucas vezes durante a vida útil de um sistema de telefonia móvel. Entretanto, devido ao tamanho das instâncias e à complexidade do problema, algoritmos

aproximativos rápidos devem ser usados a fim de oferecer boas soluções em um tempo aceitável.

Uma *alocação dinâmica* é realizada à medida que novos canais são necessários às estações-base. Isso ocorre quando o número de usuários acessando a estação-base ultrapassa sua capacidade de serviço. Caso haja canais disponíveis em outras estações, pode-se tentar aproveitá-los para servir uma demanda extra. Os algoritmos de alocação dinâmica precisam trabalhar de forma distribuída e em tempo real. Além disso, por terem que levar em consideração diversas características da tecnologia celular em que estão sendo empregados, tornam-se muito específicos àquela tecnologia. Muito tem sido publicado nesta área nos últimos anos (veja (Singh e Bertsekas, 1996; Cao e Singhal, 1997)). Em Katzela e Naghshineh (Katzela e Naghshineh, 1996), há uma discussão detalhada sobre os métodos de alocação estática e dinâmica.

Além desta classificação tradicional dos métodos de alocação de frequências, alguns trabalhos tentam explorar métodos intermediários entre eles. Um exemplo é o esquema de *empréstimo de canais* (*Borrowing Channel Assignment – BCA*) (Lee, 1989). De acordo com este método, as células possuem uma alocação fixa. No momento em que o número de pedidos é maior que a capacidade da célula, ela pode pedir canais “emprestados” a outras células. No artigo de Sandalidis *et. al.* (Sandalidis *et al.*, 1999), BCA é modelado como um problema de otimização e três heurísticas baseados em algoritmos genéticos e redes neurais são apresentados.

Apesar da conveniência da alocação dinâmica de canais, deve-se notar que na maioria das vezes, ela é usada tendo como base uma alocação estática. Portanto, percebe-se o papel importante que esta última exerce na definição de uma rede de telefonia móvel. Nesta dissertação, trata-se exclusivamente de alocações estáticas de canais. A menos que indicado o contrário, deve-se entender por todo o resto do trabalho que é a este tipo de alocação de frequências a que se está referindo.

Devido à separação mínima que deve existir entre cada par de transmissores de uma rede de telefonia móvel, conclui-se que todas as alocações de canais que não respeitam estes limites serão consideradas inválidas. De modo mais formal, uma alocação de frequências é *válida*, se para todo par  $(a, b)$  de transmissores do sistema, a diferença entre as frequências alocadas a  $a$  e  $b$  é menor que  $d(a, b)$ .

### 1.2.4 Tecnologias

No desenvolvimento da telefonia móvel, diversas tecnologias têm sido utilizadas. O motivo para esta diversidade é o rápido avanço realizado nesta área, o grande número de empresas concorrentes no mercado e a falta de uma padronização para o setor. Nesta seção serão mostradas algumas das tecnologias mais importantes.

#### TDMA

Na utilização de um canal de frequências para transmissão de voz, pode-se perceber que, durante alguns períodos de tempo, ele não estará sendo totalmente aproveitado. Por exemplo, quando um dos participantes de uma conversação telefônica fala, o outro recebe informações. Isto significa que, durante a maior parte do tempo, o canal está sendo usado de forma unidirecional. Além disso, toda conversação é formada por pausas, durante as quais nenhuma informação é transmitida. Essas características são usadas pelos serviços de telefonia para tentar otimizar a transmissão de informações em uma conversa telefônica.

Uma característica da percepção do ouvido humano é usada como base para a criação do método TDMA. Ela diz respeito ao fato de que não se consegue captar alterações sonoras que ocorrem muito rapidamente. Isto indica que em vez de transmitir toda e qualquer variação sonora, é suficiente transmitir informações em pequenos intervalos de tempo, como por exemplo, a cada 10 ms.

Observando essas características da transmissão de voz, uma série de estratégias foram desenvolvidas. Estas estratégias visam a redução da quantidade de informações que precisam ser transmitidas em uma ligação telefônica. Aplicados à telefonia celular, estes métodos permitem que um canal seja compartilhado por diversos usuários, sem perda de qualidade de serviço.

O TDMA (Time Division Multiple Access – acesso múltiplo por divisão no tempo) é uma tecnologia que divide o uso de um canal de frequências em diversos “slots”, ou seja, em períodos de tempo em que diferentes transmissões estarão sendo realizadas. Na Figura 1.10 tem-se um exemplo de compartilhamento de canal através do tempo. Existem três usuários, representados por  $u_1$ ,  $u_2$  e  $u_3$  que transmitem informações de forma alternada através dos slots  $s_1$ ,  $s_2$ ,  $s_3$ , etc.

O TDMA é um dos métodos de compartilhamento de canal mais usados em telefonia celular, sendo implementado pela maioria das tecnologias. Uma das poucas exceções é

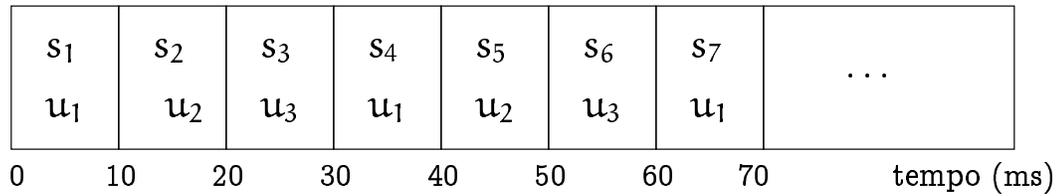


Figura 1.10: Divisão de um canal em slots de tempo.

o CDMA, que será discutido na Seção 1.2.4.

## AMPS

AMPS (Advanced Mobile Phone System – sistema avançado de telefonia móvel) é um sistema de comunicação móvel para telefonia celular. Ele é um dos mais antigos e foi o primeiro a ser usado no Brasil. Apareceu em 1983 nos Estados Unidos e rapidamente se popularizou. O principal problema do sistema AMPS é o uso de tecnologia analógica, que está sendo gradualmente substituída por métodos digitais. Outras desvantagens do sistema são: o reduzido número de usuários por célula, limitações no uso do espectro, problemas de segurança na transmissão de dados (que pode ser facilmente quebrada) e dificuldades com o *handoff*, que provoca interrupções momentâneas. Apesar disso, o AMPS tem como vantagem a simplicidade e a facilidade de implantação, em relação a tecnologias mais complexas. Isso fez com que desfrutasse de ampla aceitação e ainda corresponde hoje a uma grande parcela dos sistemas de telefonia móvel em funcionamento.

Como uma forma de reabilitar o AMPS, foi desenvolvida uma versão digital sua, o D-AMPS. Esta versão suporta as tecnologias FDMA (*Frequency Division Multiple Access* – acesso múltiplo por divisão de frequências) e TDMA. Esta versão é voltada principalmente para sistemas em que se deseja uma transição mais suave entre a tecnologia analógica e a digital.

## GSM

O GSM (Global Systems for Mobile Communications – sistema global para comunicações móveis) foi criado para ser o padrão europeu de telefonia celular (Fahham, 1996). As principais características de projeto do GSM são:

- Boa qualidade de transmissão de voz

- Baixo custo de serviço
- Suporte interurbano<sup>2</sup>.
- Suporte a terminais de mão<sup>3</sup>.
- Uso eficiente do espectro
- Compatibilidade com ISDN (Integrated Services Digital Network – rede digital de serviços integrados)

Os estudos para a definição do GSM tiveram início em 1982. Tendo sido proposto por uma comissão da *Conference of European Posts and Telegraphs* (Conferência de Correios e Telégrafos Europeus), o padrão GSM foi adotado pelo European Telecommunication Standards Institute (Instituto Europeu de Padrões de Telecomunicação) em 1989. O método mais usado pelo GSM para codificação de sinais é o TDMA, sendo também possível o uso de CDMA. Atualmente o GSM é um padrão firmemente estabelecido na Europa, sendo também utilizado em outros lugares do mundo.

## CDMA

O CDMA (*Code Division Multiple Access* – acesso múltiplo por divisão de código) é um sistema de telefonia celular desenvolvido nos EUA, tendo encontrado lá seu maior número de usuários. O CDMA oferece muitas vantagens sobre outros sistemas. Sua principal característica é o uso total da banda disponível para a transmissão, sem a necessidade de uma divisão prévia em canais. Isto permite a multiplexação e o uso de códigos para o envio simultâneo, tornando o sistema mais robusto e aumentando a segurança e o sigilo mesmo sem o uso de criptografia. Outras vantagens que podem ser apontadas no CDMA são: controle de potência de transmissão, permitindo a redução de interferência; redução do ruído através da exploração dos momentos de silêncio da conversação; maior número de canais disponíveis, etc.

Apesar das vantagens, o CDMA é um sistema complexo e caro de se implementar, restringindo seu uso. Entretanto, devido às vantagens apresentadas, espera-se que seus custos tornem-se menores à medida que a tecnologia se desenvolve.

---

<sup>2</sup> “roaming” em inglês.

<sup>3</sup> “hand-held” em inglês.

## 1.3 Metaheurísticas

O projeto de algoritmos exatos e que executem em tempo polinomial para problemas pertencentes à classe NP-Difícil tem sido um dos principais problemas abertos da Ciência da Computação. Por este motivo, quando se trabalha com instâncias suficientemente grandes destes problemas, torna-se necessário lançar mão de métodos aproximativos.

Os algoritmos aproximativos podem ser de dois tipos: com garantia de aproximação ou sem garantia de aproximação. Os primeiros fornecem uma resposta que encontra-se, no máximo, a uma distância fixa de um ótimo global. Em geral, dado um valor  $\varepsilon > 1$ , um algoritmo  $\varepsilon$ -aproximativo  $A$  para um problema  $\Pi$  é aquele em que, para cada instância  $I_\Pi$  de  $\Pi$  retorna um solução  $s$  tal que

$$f(s) < \varepsilon \cdot f(s^*),$$

onde  $s^*$  é a solução ótima para a instância  $I_\Pi$ . Os algoritmos com garantia de aproximação normalmente exploram propriedades específicas do problema, que fornecem a garantia desejada. Portanto, tais esquemas não podem ser facilmente generalizados. De acordo com resultados da teoria da complexidade, sabe-se que alguns problemas não admitem nem mesmo algoritmos com garantias de aproximação, a menos que  $P = NP$  (Garey e Johnson, 1979).

Entre os algoritmos aproximativos sem garantia de performance, estão as *heurísticas*. Heurísticas são algoritmos que usam conhecimento acerca do problema que está sendo resolvido de modo a encontrar rapidamente e com alta probabilidade uma solução satisfatória. O desenvolvimento de heurísticas para problemas de otimização combinatória tem sido uma área de pesquisa bastante ativa nos últimos anos.

Apesar das vantagens dos procedimentos heurísticos em termos de velocidade e qualidade das soluções, certos problemas tornam difícil o seu uso em algumas situações. Por exemplo, sabe-se que a qualidade da solução de um algoritmo heurístico de otimização pode variar de acordo com as características da instância a ser otimizada. Ou seja, uma heurística pode ser eficiente para algumas instâncias; por outro lado pode ter comportamento insatisfatório para outras instâncias do problema. Algumas características das instâncias que podem prejudicar o desempenho de uma heurística são: a topologia da instância, sua estrutura, etc.

Outro problema grave que ocorre na utilização de heurísticas para problemas combinatórios é relacionado à tendência que elas possuem a ficarem presas em ótimos locais.

Ótimos locais são pontos  $s^*$  do espaço de soluções com a seguinte propriedade. Para cada ponto  $s'$  pertencente a uma vizinhança de  $s^*$ , tem-se que  $f(s') \geq f(s^*)$ . Ou seja, qualquer elemento pertencente à vizinhança de  $s^*$  possui valor de função objetivo igual ou pior que o de  $s^*$ . Na Figura 1.11 vê-se um exemplo de função contínua, não convexa e com vários mínimos locais. As heurísticas, que procuram sempre melhorar o valor da função objetivo, podem facilmente ficar estagnadas nestes pontos sem possibilidades de alcançar o ótimo global.

Como uma forma de generalizar o procedimento de busca heurístico, foram propostas uma série de heurísticas gerais, conhecidas como *metaheurísticas*. Algumas características diferenciam as metaheurísticas de heurísticas comuns. Uma delas é a sua generalidade. Metaheurísticas são esquemas gerais que podem ser implementados para diversos problemas. Uma metaheurística não especifica detalhes, que podem ser adaptados ao problema em questão, mas indica apenas os elementos importantes que devem estar presentes em qualquer implementação. Normalmente, uma metaheurística pode ser reutilizada em diversos domínios de problemas diferentes, variando-se apenas alguns parâmetros de funcionamento. Este parâmetros também são conhecidos como *parâmetros de calibragem* e permitem a adequação da metaheurística a um problema específico.

Outra característica importante das metaheurísticas é o uso de procedimentos aleatórios. Através do uso adequado de escolhas aleatórias a metaheurística reduz o risco de estagnação em ótimos locais durante a busca. Deste modo tenta-se eliminar o principal problema existente nas heurísticas de busca local. O uso de procedimentos aleatórios também favorece uma melhor exploração do espaço de soluções, permitindo que, em alguns casos, as soluções possam ser encontradas mais rapidamente.

Uma vantagem importante das metaheurísticas diz respeito a seu comportamento para grandes instâncias de problemas combinatoriais. Devido às características apontadas, as metaheurísticas são capazes de encontrar soluções satisfatórias mesmo para instâncias muito grandes de problemas NP-Difíceis, que não podem ser abordados com métodos exatos. Além disso, para alguns problemas, para os quais boas heurísticas não foram ainda desenvolvidas, as metaheurísticas permanecem como umas das poucas alternativas de solução.

Diversas metaheurísticas foram propostas, tentando explorar aspectos diferentes do procedimento de busca por soluções. Entre aquelas que têm se mostrado mais promissoras para o uso geral, pode-se citar a Busca Tabu (Glover, 1989a; Glover, 1989b),

Têmpera Simulada (Kirkpatrick *et al.*, 1983), os Algoritmos Genéticos (Holland, 1975) e GRASP (Greedy Randomized Adaptive Search Procedure)(Feo e Resende, 1989). Nas próximas seções serão discutidas três destas metaheurísticas. Na Seção 1.3.1 será exposto o método de Busca Tabu. A Seção 1.3.2 será dedicada à metaheurística de têmpera simulada. Por fim, será apresentado o método de Algoritmos Genéticos na Seção 1.3.3. O GRASP, sendo objeto da proposta apresentada nesta dissertação, será descrito detalhadamente no Capítulo 3.

### 1.3.1 Busca Tabu

Busca Tabu (BT) é um método de busca global em otimização combinatória introduzido por Fred Glover (Glover, 1989a; Glover, 1989b). O processo usado por BT procura evitar a tendência das heurísticas a ficarem presas em mínimos locais. Para escapar de mínimos locais, a metaheurística BT usa um conjunto de operações, tais como guardar as soluções que já foram encontradas em uma lista de elementos proibidos, denominada *lista tabu*. A lista tabu é uma estrutura de dados com  $t$  posições que permite “lembrar” características das últimas  $t$  soluções encontradas. Isso evita que o algoritmo retorne inadvertidamente a soluções que já foram exploradas. O tamanho da lista tabu é um dos parâmetros de ajuste do algoritmo e deve ser definido empiricamente.

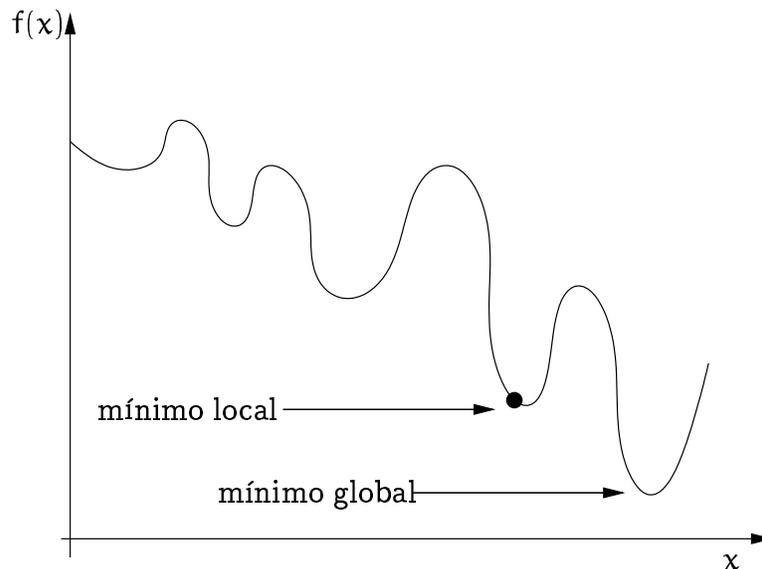


Figura 1.11: Contraste entre um mínimo local e o mínimo global de uma função.

Segundo Glover (Glover, 1989a), BT faz uma exploração agressiva do espaço de

soluções, buscando sempre a melhor configuração em cada movimento. Partindo de uma solução qualquer  $s$ , o algoritmo pesquisa a sua vizinhança procurando o elemento com melhor função objetivo  $s'$ . Caso esse elemento seja melhor que a solução atual ( $f(s') \leq f(s)$ ), passa-se a ele, adicionando-o também à lista tabu. Caso contrário, esta solução será aceita somente se ela não estiver ainda na lista tabu. Feita a escolha, o algoritmo reinicia a busca.

---

**Algoritmo 1.1** Metaheurística Busca Tabu
 

---

**Require:**  $s$  é a solução inicial

```

1: {Inicialize a lista tabu}
2:  $L \leftarrow \emptyset$ .
3: while iter < max_iter do
4:   Seja  $s$  a solução atual. Encontre o vizinho  $s'$  com menor função objetivo.
5:   if  $s' \notin L$  and  $f(s') \leq f(s)$  then
6:     Faça  $L \leftarrow L \cup \{s\}$ 
7:     Faça  $s \leftarrow s'$ 
8:   else
9:     if  $f(s') \leq \text{Asp}(f(s))$  then
10:      {Atualiza critério de aspiração}
11:       $\text{Asp}(f(s')) \leftarrow f(s) - \Delta f$ 
12:      Faça  $L \leftarrow L \cup \{s\}$ 
13:      Faça  $s \leftarrow s'$ 
14:    end if
15:  end if
16:  retirar elemento mais antigo de  $L$ .
17:  max_iter  $\leftarrow$  max_iter + 1
18: end while

```

---

Outro componente importante da metaheurística Busca Tabu é o uso de um *critério de aspiração*. No momento em que todas as possíveis escolhas para a nova solução já encontram-se na lista tabu, o critério de aspiração oferece uma maneira de retornar a uma delas. Existem diversas formas de definir um critério de aspiração. Uma possível maneira seria exigir que uma solução pertencente à lista tabu só possa ser revisitada caso leve a uma solução melhor que a solução atual. Um pseudo-código mostrando todas as características da metaheurística BT é mostrado no Algoritmo 1.1.

A metaheurística Tabu Search têm sido usada para a resolução de um grande número de problemas de otimização combinatória. Uma lista exhaustiva é apresentada por Glover e Laguna (Glover e Laguna, 1997). Alguns trabalhos recentes na área têm como objetivo

paralelizar a metaheurística. Um exemplo é o trabalho de Porto *et al.* (Porto *et al.*, 1999).

### 1.3.2 Têmpera Simulada

A metaheurística Têmpera Simulada (TS) é um método de otimização global proposto por Kirkpatrick *et al.* (Kirkpatrick *et al.*, 1983). TS foi desenvolvida a partir da simulação do processo de resfriamento em sistemas físicos. Um metal em estado de fusão, quando resfriado rapidamente, tende a tornar-se quebradiço. Isso acontece porque ele não consegue encontrar um estado de energia mínima que garanta sua estabilidade. Para que o resfriamento ocorra de maneira uniforme, é necessário que a temperatura seja diminuída lentamente, com variações gradativas.

Em problemas de otimização, pode-se fazer uma analogia entre a valor da função objetivo e o estado de energia do sólido conforme descrito acima. O objetivo final do processo de busca é encontrar a configuração com menor valor de função objetivo, assim como no processo de resfriamento o objetivo é encontrar o estado de menor energia interna.

Para alcançar este objetivo, Têmpera Simulada é executada em várias etapas. Cada etapa é realizada em uma temperatura menor, de modo a simular o processo de resfriamento. Em cada etapa do processo de resfriamento, várias iterações são realizadas tentando melhorar a solução. Por exemplo, partindo da configuração  $s_1$ , o algoritmo pode, através de uma pesquisa na vizinhança de  $s_1$ , encontrar uma solução  $s_2$ . Se  $f(s_2) \leq f(s_1)$  então aceita-se  $s_2$  como a solução atual e continua-se o processo. Neste aspecto, TS comporta-se de forma idêntica a uma heurística comum de busca local, como a “descida da montanha”<sup>4</sup>. Por outro lado, se  $f(s_2) > f(s_1)$ , então  $s_2$  pode ainda ser aceita, com probabilidade igual a

$$e^{-\Delta F/T},$$

onde  $\Delta F = f(s_1) - f(s_2)$  representa a diferença entre valores de função objetivo e  $T$  é a temperatura atual, que pode ser livremente especificada pelo projetista do algoritmo para refletir o processo de resfriamento. O valor  $e^{-\Delta F/T}$  é conhecido como *critério Metrópolis* de aceitação. Ele é baseado na distribuição de Boltzmann, que modela o fenômeno físico de resfriamento.

---

<sup>4</sup>do inglês “down hill”

Uma descrição em pseudo-código para a metaheurística Têmpera Simulada é mostrado no Algoritmo 1.2. No início do processamento, a temperatura é inicializada para um valor alto. Logo após, inicia-se a busca, usando o método apresentado no parágrafo anterior. A metaheurística continua suas iterações até que a solução alcance equilíbrio. Isto pode ser indicado, por exemplo, pela execução de um número fixo de iterações (MAX\_TRY) sem que haja qualquer melhoria da função objetivo. Após o término de um destes passos, a temperatura é reduzida por um valor  $\Delta T$  e a computação reinicia. Chega-se ao final do algoritmo quando a temperatura alcança o valor zero. O resultado da computação é simplesmente o valor da solução corrente neste momento.

---

**Algoritmo 1.2** Metaheurística Têmpera Simulada
 

---

```

1: Faça  $T \leftarrow T_0$ 
2: Inicialize a solução inicial  $s$ 
3: repeat
4:    $M \leftarrow 0$ 
5:   repeat
6:     Seja  $s$  a solução atual. Encontre o vizinho  $s'$  com menor função objetivo
7:     Faça  $\Delta F \leftarrow f(s) - f(s')$ 
8:     if  $s' \leq s$  then
9:       Faça  $s \leftarrow s'$ 
10:    else
11:      if  $\text{random}[0,1] < \exp(-\Delta F/T)$  then
12:        Faça  $s \leftarrow s'$ 
13:      end if
14:    end if
15:    Faça  $M \leftarrow M + 1$ 
16:  until  $M \geq \text{MAX\_TRY}$ 
17:  Faça  $T \leftarrow T - \Delta T$ 
18: until  $T \leq 0$ 

```

---

Percebe-se que a metaheurística Têmpera Simulada passa por várias fases, onde em cada uma delas a liberdade para incorporar soluções piores é diminuída. Isso faz com que a metaheurística possa escapar de ótimos locais no início do processo. Por outro lado, a redução gradual de temperatura faz com que a probabilidade  $e^{-\Delta/T}$  torne-se cada vez menor. Portanto a cada nova fase de resfriamento torna-se mais difícil deslocamentos para soluções piores, convergindo ao final para um valor único.

Pode-se provar que este algoritmo aproxima-se assintoticamente do ótimo global com probabilidade igual a 1 (van Laarhoven e Aarts, 1987). Apesar disso, assim como ocorre

com outras metaheurísticas, o comportamento da BT depende do ajuste de parâmetros dependentes da aplicação, como por exemplo,  $T$  e  $\Delta T$ , do estabelecimento de um critério de convergência, etc.

Nos últimos anos, TS tem sido extensivamente usada para resolução de vários problemas. Ela tem sido usada tanto com problemas clássicos, como apresentado na série de artigos de Johnson *et al.* (Johnson *et al.*, 1992a; Johnson *et al.*, 1992b; Johnson *et al.*, 1992c), quanto em outros problemas como processamento de imagens (Carnevali *et al.*, 1985), mapeamento de DNA (Goldstein e Waterman, 1987) e projeto de experimentos estatísticos (Meyer e Nachtsheim, 1988). Outra tendência na área é o desenvolvimento de implementações paralelas, como realizado, por exemplo, por Abramson (Abramson, 1991). Além disso, pode-se citar trabalhos de referência sobre o assunto, como por exemplo o que foi produzido por Collins *et al.* (Collins *et al.*, 1988).

### 1.3.3 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são uma classe de algoritmos que tentam simular o processo de evolução dos seres vivos, através da seleção natural. Os primeiros trabalhos na área são devidos a Rechenberg (Rechenberg, 1973), Holland (Holland, 1975) e Schwefel (Schwefel, 1981). Os AGs têm sido usados com sucesso nos últimos anos em diversos problemas de otimização combinatória e, ao contrário de Tabu Search e GRASP, as aplicações aparecem também em outras áreas, como, por exemplo, problemas de reconhecimento de padrões. Em sua forma mais simples, um AG codifica a solução de um problema em uma representação binária, à qual é dada o nome de *cromossomo*. Um cromossomo representa um elemento do conjunto de possíveis soluções.

Define-se uma função  $f$ , chamada função de adequação ou função objetivo, que avalia as características desejáveis de um cromossomo. Por exemplo, pode-se estabelecer que quanto menor o valor de  $f(x)$ , mais adaptado será o cromossomo  $x$ . A idéia central de um AG é criar uma população inicial de cromossomos e a partir de algumas operações (conhecidas como operadores genéticos) derivar novas gerações. Os operadores genéticos têm características que permitem que os cromossomos por eles gerados sejam melhor adaptados à função objetivo.

Como exemplo, pode-se verificar a aplicação de um Algoritmo Genético ao problema da mochila (veja Seção 1.1.2). Em primeiro lugar, é preciso definir a codificação da solução no formato de um cromossomo. Seja  $X = \{x_1, x_2, \dots, x_n\}$  o conjunto de objetos, com seus respectivos volumes e valores. Uma representação de cromossomo para

o problema da mochila pode ser formada por uma seqüência de bits  $b_1 b_2 \dots b_n$  que indicarão se  $x_i$ ,  $i = 1, \dots, n$  será incluído na mochila (1) ou não (0). Na Figura 1.12 vê-se um exemplo de cromossomo para esse problema. A função de adequação para ele será exatamente a função objetivo, conforme apresentado na equação (1.6).

0	1	1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---

Figura 1.12: Exemplo de cromossomo para o problema da mochila. Cada posição representa se o objeto correspondente será incluído (1) ou não (0). O tamanho do cromossomo representa o número de objetos disponíveis.

Em cada uma das gerações faz-se uma seleção dos cromossomos que deverão reproduzir-se. Esta seleção ocorre de forma probabilística, mas com uma tendência maior para a escolha dos melhores elementos. Para implementar a seleção usa-se um método conhecido como “roleta viciada” (veja Figura 1.13). Este método baseia-se em um sorteio onde a probabilidade de ser escolhido é diretamente proporcional à função de adaptação do cromossomo. Isto pode ser enunciado como

$$p_i = \frac{f(x_i)}{\sum_{j=1}^m f(x_j)},$$

onde  $m$  é o número de cromossomos. Desta forma, procura-se simular o mecanismo de seleção natural, em que os mais aptos possuem maiores chances de sobreviver e reproduzir-se.

Os operadores genéticos permitem que se realizem modificações nos cromossomos selecionados de forma a derivar novos elementos da população. Eles são o método pelo qual uma nova geração de cromossomos é formada a partir da população antiga. Diversos operadores podem ser especificados de acordo com as características do problema, mas os operadores clássicos usados com AGs são os de *cruzamento* e *mutação*.

No operador clássico de cruzamento, dois cromossomos são escolhidos inicialmente através do algoritmo de seleção. Em seguida, é necessário definir, de forma aleatória, uma posição na representação binária que será chamada *ponto de cruzamento*. Um novo cromossomo será criado da seguinte forma. Copia-se a primeira parte do cromossomo 1 (até o ponto de cruzamento) para o início do novo cromossomo. Depois, copia-se a parte final do segundo cromossomo, que vai do ponto de cruzamento até o final. O

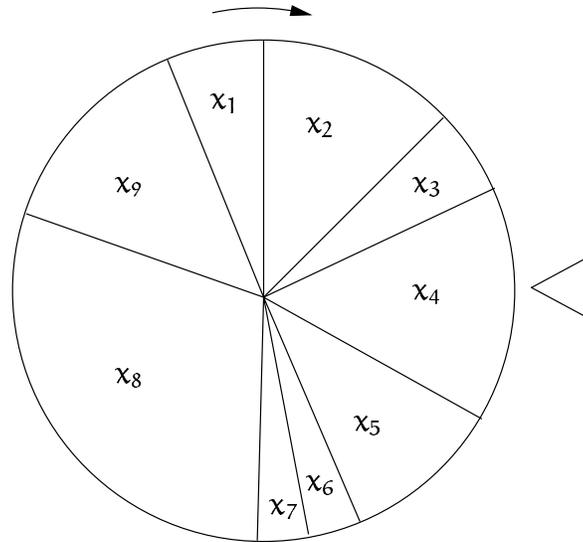


Figura 1.13: Método de seleção “roleta viciada”. Cada cromossomo tem probabilidade de ser escolhido proporcional a seu valor de adequação.

novo cromossomo será, portanto, um cruzamento das características dos cromossomos 1 e 2. Veja um exemplo de aplicação desse operador na Figura 1.14.

A partir deste método básico de cruzamento, muitas variações podem ser adotadas. Como exemplo, pode-se definir operadores que usem mais de um ponto de cruzamento, selecionando diferentes partes de cada um dos cromossomos geradores. Pode-se também definir operadores de cruzamento que levem em consideração características específicas do problema. Por exemplo, uma forma de codificar o problema do caixeiro viajante como um cromossomo seria defini-lo como uma seqüência de cidades numeradas a serem visitadas. Um operador de cruzamento eficiente deve levar em consideração esse conhecimento acerca da representação da solução.

O segundo operador que é usado com freqüência no contexto de Algoritmos Genéticos é o *operador de mutação*. Ele é usado como uma maneira de diversificar as soluções existentes na memória e fugir de ótimos locais. Seu objetivo principal é evitar a estagnação da população, que acontece quando todas as soluções na memória são muito parecidas, ou mesmo idênticas. O funcionamento do operador de mutação é simples, consistindo em selecionar um ou mais bits da representação binária e alterá-la para o valor inverso (Figura 1.15).

No problema da mochila, que está sendo usado como exemplo, uma execução do operador de mutação consiste em mudar aleatoriamente o valor de um dos objetos que

estão representados no cromossomo para um valor diferente. O operador de mutação deve ser usado poucas vezes, pois tem o potencial de alterar radicalmente o valor das soluções existentes na memória, o que seria contraproducente para o AG. No Algoritmo 1.3 tem-se uma descrição em pseudo-código de um Algoritmo Genético.

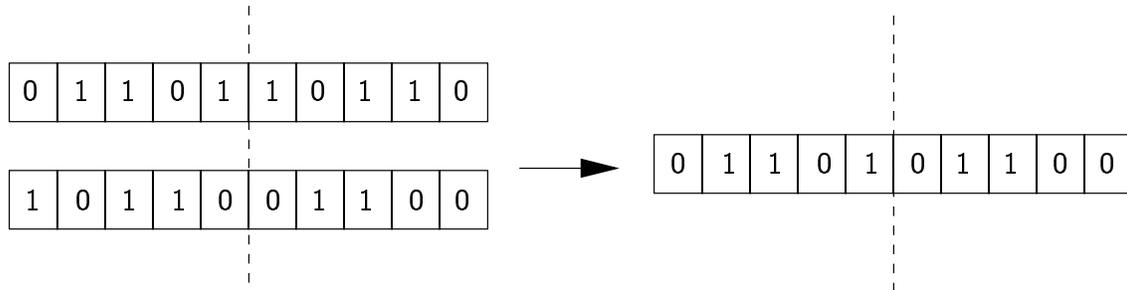


Figura 1.14: Operador de crossover em atuação para o problema da mochila. Do lado esquerdo, dois cromossomos. A linha pontilhada indica o ponto de crossover. Do lado direito o cromossomo resultante.

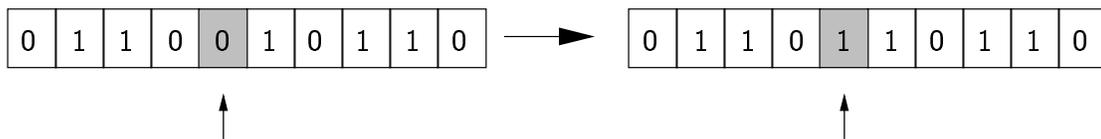


Figura 1.15: Exemplo de alteração realizada pelo operador de mutação para o problema da mochila.

Na prática, para a maioria das aplicações, esses dois operadores são insuficientes para a geração de boas soluções. O principal problema com os operadores clássicos é que as alterações realizadas criam soluções inviáveis, na maioria das vezes. Isto ocorre pois a solução é encarada pelos operadores apenas como um conjunto de bits, sem um conhecimento maior sobre a estrutura do problema. Muito da pesquisa atual em AGs tem como objetivo a solução destas dificuldades. Como exemplo, podem ser citados os Algoritmos Meméticos (Moscato, 1989) que buscam integrar AGs com busca local. Outro método recentemente proposto é conhecido como Scatter Search e tenta melhorar os algoritmos genéticos através de um melhor uso do determinismo (Glover, 1998).

---

**Algoritmo 1.3** Metaheurística Algoritmo Genético

---

```
1: {Inicializa População}
2: Faça população  $P \leftarrow \emptyset$ 
3: Faça  $i \leftarrow 0$ 
4: while  $i < \text{TAM\_POP}$  do
5:   Criar nova solução  $s$ 
6:   Faça  $P \leftarrow P \cup \{s\}$ 
7:   Faça  $i \leftarrow i + 1$ 
8: end while
9: Faça  $i \leftarrow 0$ 
10: while  $i < \text{MAX\_GEN}$  do
11:   Seleção
12:   Cruzamento
13:   Mutação
14:   Atualizar melhor solução  $s^*$ 
15:   Faça  $i \leftarrow i + 1$ 
16: end while
```

---

## 1.4 Organização da Dissertação

Os próximos capítulos desta dissertação encontram-se divididos como segue. No Capítulo 2 será feito um estudo detalhado do Problema da Alocação de Freqüências, que será o principal assunto desta dissertação. GRASP (Greed Randomized Adaptive Search Procedure) é uma metaheurística para resolução de problemas de otimização combinatória que será de interesse central para este trabalho, portanto, no Capítulo 3, esta metaheurística será discutida em profundidade. No Capítulo 4 será apresentada a proposta de solução do PAF, que envolve o uso de um GRASP. Logo após, o Capítulo 5 será dedicado à uma exposição dos resultados computacionais gerados pela abordagem proposta. Finalmente, no Capítulo 6 serão apresentadas conclusões e trabalhos futuros.

# Capítulo 2

---

## O Problema da Alocação de Frequências

*“There is an even more important reason for skepticism: there exists no unifying theory which demonstrates that formal models are a viable approach to the wide range of problems which arise in the real world. The purpose of this paper is to provide such an unifying theory for a wide variety of real world problems.”*

(Hale, 1980)

### 2.1 Introdução

O Problema da Alocação de Frequências (PAF) é um problema prático que ocorre em muitas situações na área de telecomunicações. Por este motivo tem sido objeto de estudos sob vários aspectos, que variam de acordo com a aplicação. Neste capítulo, será apresentada uma definição formal do Problema da Alocação de Frequências. Além disto, será feito um estudo do problema, analisando as diversas soluções propostas na literatura.

Este capítulo encontra-se dividido como segue. Na Seção 2.2 será feita uma definição formal do problema que será estudado neste trabalho. A Seção 2.3 é dedicada a um estudo da complexidade do problema. Um dos pontos importantes para a compreensão adequada do PAF é o seu relacionamento com o problema da coloração. Isto será discutido na Seção 2.4. Na Seção 2.5 serão apresentados os trabalhos publicados sobre o PAF, cobrindo vários aspectos que serão relevantes para o estudo do problema. Finalmente, na Seção 2.7, serão discutidas algumas formulações alternativas propostas para o PAF.

## 2.2 Definição do problema

De um modo geral, o nome Problema da Alocação de Freqüências tem sido usado para designar uma série de problemas relacionados, que ocorrem na área de transmissão de informações via radiofreqüências. Todos eles têm em comum o fato de que procura-se alocar freqüências a transmissores de uma forma ótima. Entretanto, as restrições impostas ao problema e a função objetivo podem variar bastante, dependendo da arquitetura do sistema, dos tipos de interferência que estão sendo levado em consideração, etc.

Uma forma de classificar estes problemas foi proposta por Hale (Hale, 1980). Ele propôs uma formalização do problema usando teoria dos grafos. Tal formalização pode ser vista como uma forma estendida do problema de coloração. Este é um problema clássico, e bastante conhecido por pertencer à classe NP-Difícil, além das diversas aplicações que podem ser modeladas como um problema de coloração (veja, por exemplo, (Matula *et al.*, 1972)).

O Problema da Alocação de Freqüências pode contemplar diversos objetivos (veja (Hale, 1980)). Um destes objetivos seria minimizar o número de canais usados pela alocação, ou seja minimizar  $|\mathcal{X}|$ , onde  $\mathcal{X} = \{x | y(c) = x \text{ para todo } c \in C\}$ . Este objetivo também é chamado de *minimizar a ordem*. Um segundo objetivo possível seria minimizar a diferença  $|y(b) - y(a)|$  entre o valor do maior canal alocado ( $y(b)$ ) e o menor ( $y(a)$ ). Esta diferença é também chamada de *span* e representa a área do espectro que está sendo ocupada pelo sistema.

Por último e de maior interesse para este trabalho, há o objetivo de minimizar a interferência total do sistema. Os tipos de interferência considerados em um sistema de telefonia celular são os seguintes: local, que ocorre entre antenas que encontram-se na mesma estação base; co-canal que ocorre entre antenas usando o mesmo canal, mesmo estando em estações base diferentes; e adjacente-canal, que ocorre entre antenas que usam canais adjacentes no espectro.

O foco do trabalho será concentrado nas características do PAF aplicado a sistemas de telefonia celular. As diferenças principais do PAF que ocorrem neste caso são:

- O número de freqüências disponíveis para o uso em telefonia celular é fixo. Portanto não faz sentido (a não ser para instâncias com número reduzido de antenas) otimizar o número de canais (ordem) ou a faixa de espectro (*span*) usados. Com o aumento do número de usuários nas redes de telefonia móvel, o Problema da Alocação de Freqüências tem se tornado principalmente o de garantir a existência de

alocações válidas, com o mínimo de interferência. Portanto, o foco da dissertação estará sobre a otimização da interferência total.

- Os vários transmissores do sistema estão agrupados em algumas estações-base, com interferências locais prefixadas. Isso permite tratar a alocação de canais dentro de uma estação base de uma forma diferenciada.

Inicialmente, serão apresentadas algumas definições de teoria dos grafos necessárias para a compreensão do problema.

Um *grafo*  $G(V, E)$  é formado por um conjunto de  $V = \{v_1, \dots, v_n\}$  de *vértices* e um conjunto  $E = \{\{v_{i_1}, v_{j_1}\}, \dots, \{v_{i_m}, v_{j_m}\}\}$  de pares de vértices de  $V$ . Os pares não ordenados de vértices em  $\{v_i, v_j\} \in E$  também são chamados de *arestas*.

Seja  $G(V, E)$  um grafo onde  $V$  é o conjunto transmissores. Seja  $C \subset Z_+$  o conjunto de canais disponíveis. Serão definidas duas funções:  $c^{co} : E \rightarrow [0, 1]$  que indica o nível de interferência co-canal e  $c^{ad} : E \rightarrow [0, 1]$  para interferência adjacente-canal, tal que  $c^{co}(u, v) \geq c^{ad}(u, v)$ ,  $\forall u, v \in V$ . Uma alocação de frequências é um mapeamento  $y$  de  $V$  em  $C$ . Então o PAF consiste em minimizar a função

$$f(x) = \sum_{y(v)=y(w)} c^{co}(v, w) + \sum_{|y(v)-y(w)|=1} c^{ad}(v, w) \quad (2.1)$$

tal que

$$|y(v) - y(w)| \geq d(v, w) \quad (2.2)$$

para cada par de antenas  $v, w \in V$ ,  $v \neq w$ .

Na Figura 2.1 tem-se um exemplo de instância do Problema da Alocação de Frequências formada por quatro células, nomeadas  $c_1, c_2, c_3$  e  $c_4$ . Em cada célula tem-se uma estação base, com um número variável de transmissores. o vetor demanda para as estações base de  $c_1, c_2, C$  e  $c_4$ , respectivamente, é  $[4, 3, 3, 2]$ . Nessa figura pode-se observar o grafo de interferências com arestas entre  $(c_1, c_2)$ ,  $(c_1, c_4)$ ,  $(c_2, c_3)$ ,  $(c_3, c_4)$  e por fim  $(c_2, c_4)$ . As separações mínimas entre canais para cada par de estações base é mostrado no rótulo da aresta correspondente. Finalmente, no interior das estações base e próximo a cada transmissor, vê-se o número do canal alocado para este transmissor. A alocação de frequências apresentada usa 11 canais diferentes, mas ocupa uma faixa de 13 canais.

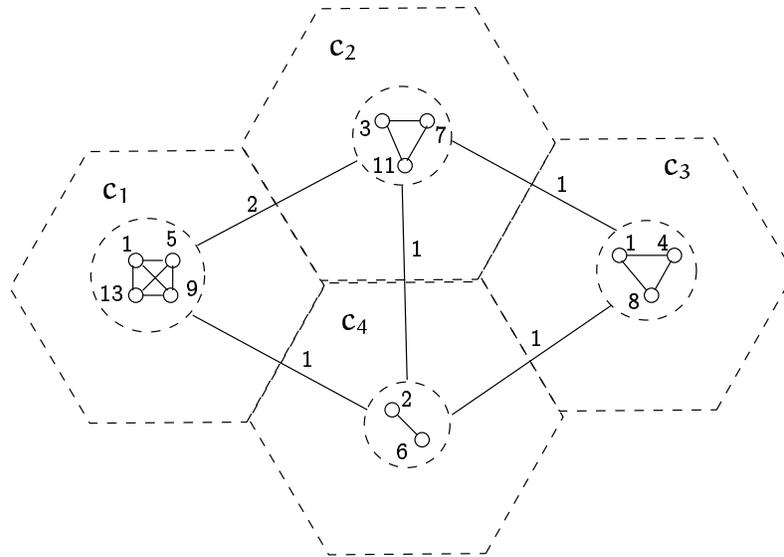


Figura 2.1: Um exemplo de alocação de frequências

## 2.3 Complexidade

O primeiro resultado importante em termos de complexidade para o Problema da Alocação de Frequências é definir em que classe ele se enquadra. Isto é enunciado no seguinte teorema:

**Teorema 1 (Hale, 1980):** *O Problema da Alocação de Frequências pertence à classe NP-Difícil.*

Para mostrar que o PAF é um problema NP-Difícil, será provado que o problema de decisão DPAF associado pertence à classe NP-Completo. Assume-se que todas as interferências associadas ao problema possuem valores racionais. Então, DPAF, o problema de decisão associado ao PAF, é definido da seguinte forma: Dados como entrada a função distância mínima  $d : E \rightarrow \mathbb{N}$  e as funções de interferência co-canal e adjacente-canal ( $c^{co}$  e  $c^{ad}$ ), o DPAF consiste em responder se existe uma alocação de canais que produza interferência menor ou igual a  $\chi \in \mathbb{Q}$ .

Com este problema, para cada algoritmo que forneça como resposta uma alocação, é fácil verificar se ele respondeu corretamente à questão. Basta calcular o somatório das interferências  $S$  e verificar se  $S \leq \chi$ . Para isto basta verificar cada uma das arestas do grafo de interferência, o que pode ser feito no máximo em  $O(|E|) \leq O(|V|^2)$ .

O último passo da prova consiste em reduzir algum problema  $\Pi'$ , reconhecidamente pertencente à classe NP-Completo, ao PAF. Este passo foi realizado, no trabalho de Hale (Hale, 1980), a partir do problema da  $k$ -coloração de grafos, que é reconhecidamente NP-Completo (veja (Garey e Johnson, 1979)).

Uma vez que o problema de decisão associado ao PAF pertence à classe NP-Completo, pode-se afirmar que o próprio PAF é um problema NP-Difícil.

## 2.4 PAF e coloração de vértices

Uma das maneiras de formalizar o PAF é tratar os transmissores como vértices de um grafo com arestas para cada par de transmissores em que ocorram restrições. A aplicação de teoria dos grafos torna-se portanto essencial para o correto entendimento do problema. Além disso, resultados já existentes para problemas relacionados permitem chegar a novos algoritmos que exploram as propriedades do problema.

Nesta seção mostra-se como o PAF pode ser formulado em termos do problema da coloração de grafos (PCG). Além disso, serão apresentados alguns dos mais importantes algoritmos para coloração de grafos. Estes algoritmos formam a base dos métodos usados para resolução do PAF. Portanto, torna-se necessário um conhecimento destes algoritmos para que possa-se discutir a metodologia de solução para o PAF.

### 2.4.1 Definições de teoria dos grafos

Nesta seção serão definidos vários problemas importantes em teoria dos grafos que serão necessários durante o trabalho para melhor compreensão do PAF.

**Coloração própria de vértices.** Dados um grafo  $G(V, E)$  e uma função  $w : V^2 \rightarrow \{0, 1\}$  definida por  $w(a, b) = 1$  sse  $(a, b) \in E$ , uma coloração própria de  $G$  é um mapeamento  $\phi : V \rightarrow \mathbb{N}$  tal que  $|\phi(a) - \phi(b)| \geq w(a, b)$ .

**Coloração por listas.** Seja  $G(V, E)$  um grafo com conjunto de vértices  $V = \{v_1, \dots, v_k\}$ . Então denota-se por  $L_i$  o conjunto de cores admissíveis associadas a  $v_i$ . Denota-se a união dos  $L_i$  por:

$$L = \bigcup_{i \in V} L_i.$$

Um mapeamento  $\phi : V \rightarrow L$  é uma coloração por listas, ou  $L$ -coloração, se  $\phi$  for uma coloração para  $G$  e  $\phi(v_i) \in L_i$  para todo  $i \in \{1, \dots, k\}$ . Em algumas publicações, as listas  $L_i$  referem-se a cores proibidas. Neste caso, pode-se considerar  $L'_i$  como sendo o

complemento em relação ao conjunto  $L_i$ , o que permitirá tratar o problema de forma semelhante.

**T-Coloração.** Uma T-Coloração é um tipo especial de coloração em que evitam-se cores que estejam em um determinado conjunto. Dado um grafo e um conjunto finito  $T \subset \mathbb{N}$ , uma T-coloração é uma alocação  $\gamma : V \rightarrow \mathbb{N}$  tal que, para toda aresta  $(i, j) \in E$ ,  $|\gamma(i) - \gamma(j)| \notin T$ . Pode-se provar que o Problema da Alocação de Frequências reduz-se a um problema de T-Coloração, como mostrado em (Hale, 1980). Justifica-se portanto, o interesse teórico neste tipo especial do problema da coloração.

### 2.4.2 Heurísticas para coloração de grafos

Várias heurísticas foram desenvolvidas para resolução do problema da coloração. Elas são importantes neste estudo porque servem como um ponto de partida para heurísticas mais elaboradas que serão propostas para o PAF. Nesta seção serão abordadas as heurísticas mais conhecidas para esse problema. A maior parte das informações desta seção foram encontradas no trabalho de (Laguna e Martí, 1998).

Os métodos mais simples verificam os vértices do grafo em seqüência e atribuem à primeira cor disponível. Esse algoritmos são portanto conhecidos como algoritmos seqüenciais. Entretanto, podem haver muitas variações na forma como os vértices são considerados, levando a versões diferentes do algoritmo. A metodologia conhecida como *Maior Primeiro* (MP) empregada em (Welsh e Powell, 1967) ordena os vértices em ordem crescente de seus graus. O método *Menor por Último*, usado em (Matula *et al.*, 1972) assume uma ordenação similar, com a seguinte diferença: pegue o vértice com menor grau e coloque no final da lista. Depois, dos vértices que sobraram, pegue aquele com menor grau no grafo induzido por eles e coloque na penúltima posição da lista. O algoritmo continua dessa forma até que a lista esteja completa.

Uma melhoria que pode ser obtida nos métodos acima apresentados, é a realização de trocas nos vértices já coloridos, de forma que o novo vértice possa ser colorido com uma cor já usada. Os métodos que resultam dessa mudança são chamados *Maior Primeiro com trocas* e *Menor por Último com trocas*.

Em (Brélaz, 1979) foi introduzido o método chamado DSATUR, em que os vértices são escolhidos de acordo com o seu *grau de saturação*. O grau de saturação de um vértice é uma quantidade que se altera no decorrer do algoritmo e representa o número de cores diferentes já atribuídas a vizinhos daquele vértice. Em cada passo o vértice com maior grau de saturação é selecionado e uma cor válida é atribuída a ele. A idéia

subjacente a esse algoritmo é a de que devem ser selecionados primeiro vértices com poucas opções para alocação de cores.

Enquanto as heurísticas anteriores atribuem cores um vértice de cada vez, outras preocupam-se em encontrar todos o vértices que podem receber uma determinada cor. O conjunto de vértices que possuem a mesma cor é também chamado *classe de cor*. Um algoritmo baseado nessa idéia é o *Maior Primeiro Recursivo*, devido a Leighton (Leighton, 1979). Para criar uma nova classe de cor, o algoritmo procede como mostrado no Algoritmo 2.1.

---

**Algoritmo 2.1** Algoritmo RLF
 

---

- 1: Seja  $V$  o conjunto de vértices que já foram coloridos.
  - 2: Seja  $\bar{V}$  o conjunto de vértices que não foram coloridos.
  - 3: Selecione o vértice  $v \in \bar{V}$  com maior grau no grafo induzido por  $\bar{V}$ .
  - 4: Seja  $U$  o conjunto de vértices adjacentes a  $v$ .
  - 5: **while**  $\bar{V} \neq \emptyset$  **do**
  - 6:   Selecione o vértice  $v \in \bar{V}$  com maior grau no grafo induzido por  $U$ .
  - 7: **end while**
- 

Uma modificação no algoritmo anterior denominada *RLF com troca* (XRLF) foi introduzida em (Jonhson *et al.*, 1991). Seu método consiste em criar diversas classes candidatas para uma mesma cor. Para cada uma dessas classes, calcula-se o número de arestas induzidas no grafo formado pelos vértices restantes. XRLF é a heurística base para uma implementação de GRASP para o problema da coloração (Laguna e Martí, 1998).

## 2.5 Trabalhos relacionados

O Problema da Alocação de Freqüências (PAF) tem sido estudado já há algumas décadas e vários trabalhos têm sido realizados com o objetivo de fornecer uma solução. Os primeiros métodos usados para a resolução desse problema foram informais e executados manualmente (Jansky, 1977). A principal técnica para evitar a interferência causada por múltiplos transmissores era a alocação de freqüências suficientemente distantes no espectro (Hale, 1980). Entretanto, esse método tem o efeito indesejado de comprometer uma parte preciosa do espectro disponível, tornando-se inviável à medida que o número de transmissores aumenta.

Depois da Segunda Guerra houve um crescimento acelerado do uso do espectro de

freqüências, devido em grande parte ao uso militar mas também ao uso comercial das telecomunicações. Nesta época começou-se a discutir métodos para o uso adequado do espectro (JTAC, 1968). Aplicações como a transmissão de TV e outros conteúdos transmitidos via satélite, comunicação militar, radioamadorismo, comunicação naval, etc., passaram a ocupar freqüências de rádio em quantidades cada vez maiores. Métodos automatizados passaram então a ser pesquisados através de um estudo algorítmico do problema. Entretanto, modelos matemáticos para o estudo da interferência em transmissões de rádio só passaram a ser desenvolvidos a partir da década de 60 (Eden *et al.*, 1960; Fastert, 1960).

Somente em 1980, William Hale, em um importante trabalho (Hale, 1980), propôs uma modelagem matemática baseada em teoria dos grafos para uma larga classe de problemas de alocação de freqüência. Entre suas contribuições para esse campo de estudos, foi proposta uma terminologia padronizada para referir-se aos diferentes tipos de problemas associados. Provou-se, por redução ao problema de coloração em grafos, que o PAF pertence à classe computacional NP-Difícil. Provou-se que problemas com restrições de distância e freqüência podem ser reduzidos a problemas com restrições apenas nas freqüências usadas. Demonstrou-se a existência de subproblemas que possuem complexidade polinomial.

No mesmo trabalho, introduziu-se um novo tipo de problema de coloração generalizado, hoje conhecido como problema da T-coloração (PTC), adequado à modelagem do PAF. A partir de 1980, parte dos esforços para a resolução do PAF têm se concentrado no estudo do problema da T-coloração. Vários pesquisadores têm desenvolvido trabalhos nesta área, como por exemplo, Cozzens e Roberts (Cozzens e Roberts, 1982; Cozzens e Roberts, 1991; Roberts, 1991), D. Liu (Liu, 1992). O Problema da Alocação de Freqüências também pode ser visto como uma coloração onde, para cada vértice do grafo existe uma lista de freqüências disponíveis.

Com o grande desenvolvimento dos sistemas de telefonia móvel celular, iniciaram-se estudos para a resolução de problemas de alocação de freqüências associados a esta aplicação. Uma característica peculiar aos sistemas celulares é o número predefinido de canais disponíveis. Portanto, em vez de tentar reduzir o número de canais usados, o objetivo torna-se o de produzir alocações válidas e reduzir a interferência associada à alocação.

Neste capítulo trata-se dos diferentes métodos algorítmicos apresentados na literatura para a resolução do PAF. Entretanto, dar-se-á maior ênfase àqueles que apresen-

tam interesse para o trabalho. Para informações mais detalhadas acerca dos trabalhos existentes, deve-se consultar o Capítulo *Frequency assignment problems*, escrito por Murphey, Pardalos e Resende. Este é uma coletânea onde pode-se encontrar referências sobre as várias técnicas usadas para a resolução do PAF nos últimos anos.

Outro trabalho de referência muito bem elaborado é o de Katzela e Naghshineh (Katzela e Naghshineh, 1996), onde os diversos esquemas de alocação de frequências em telefonia celular são analisados. Por fim, pode-se citar como leitura recomendada o trabalho do projeto CALMA (Combinatorial ALgorithms for Military Applications). Este projeto foi criado pelo EUCLID (EUropean Cooperation for the Long term In Defence – Cooperação europeia para defesa a longo prazo) para estudar problemas de alocação de frequências. Este projeto contou com a participação de grupos de várias universidades europeias, que estudaram o problema usando técnicas diferentes, como por exemplo Busca Tabu (Bouju *et al.*, 1995), Algoritmos Genéticos (Kapsalis *et al.*, 1995) e Satisfação de Restrições (Schiex *et al.*, 1995; vom Scheid, 1997).

Sob o aspecto teórico, os métodos de resolução podem ser divididos em dois tipos: exatos e aproximativos. A Seção 2.5.1 será dedicada à apresentação de métodos exatos divisados para o problema. Na Seção 2.6 serão apresentados os diversos algoritmos aproximativos propostos para o PAF.

### 2.5.1 Algoritmos Exatos

Em otimização combinatória, métodos que buscam soluções exatas estão frequentemente associados à programação matemática com variáveis inteiras, ou simplesmente *Programação Inteira* (PI). Através da PI, pode-se modelar qualquer problema de otimização combinatória, em particular os de otimização em grafos (Campelo e Maculan, 1994).

Como a maioria dos problemas em PI envolvem complexidade exponencial, alguns métodos de resolução têm sido desenvolvidos como: branch-and-cut, branch-and-bound, geração de colunas, métodos poliédricos, etc. Deve-se notar entretanto que para boa parte dos problemas, tais métodos podem levar no pior caso a uma enumeração completa das soluções do problema. Portanto na maioria dos casos e principalmente para grandes instâncias do problema, as soluções via PI não tem uso prático. Algumas das aplicações de métodos exatos ao PAF serão vistas nos próximos parágrafos.

### Formulações em Programação Inteira

Numa formulação em PI, as relações entre valores do problema são modeladas por variáveis que devem receber somente valores inteiros. Um importante caso especial ocorre quando as variáveis do problema podem receber somente valores 0 ou 1. Neste caso chama-se o modelo de programação inteira 0-1 e as variáveis assim usadas são chamadas de *variáveis de decisão*. Suponha que o conjunto de antenas seja dado por  $V = \{v_1, \dots, v_n\}$  e o conjunto de canais disponíveis por  $C = \{1, \dots, n\}$ . Então pode-se dividir uma variável de decisão  $x_{if}$  tal que seu valor seja 1 quando a antena  $i$  receba o canal  $f$  e 0 caso contrário. Ou seja:

$$x_{if} = \begin{cases} 1 & \text{se } \text{freq}(i) = f \\ 0 & \text{caso contrário.} \end{cases}$$

Uma outra variável de decisão pode ser criada para representar se uma frequência está sendo usada ou não:

$$y_f = \begin{cases} 1 & \text{se a frequência } f \text{ estiver sendo usada} \\ 0 & \text{caso contrário.} \end{cases}$$

Aardal *et al.* (Aardal *et al.*, 1998) propuseram uma formulação inteira para o PAF usando estas variáveis de decisão. Sua formulação é apresentada a seguir:

(PAFVP)

$$\min \quad \sum_{f \in C} y_f \quad (2.3)$$

$$\text{t.q.} \quad \sum_f x_{i,f} = 1 \quad \forall i \in V \quad (2.4)$$

$$x_{i,f} + x_{j,g} \leq 1 \quad \forall i, j \in V, \forall f, g \in C, |f - g| < d(i, j) \quad (2.5)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i \in V \quad (2.6)$$

Na formulação PAFVP, a função objetivo 2.3 representa a soma das frequências usadas. Isto permite minimizar o número total de frequências da alocação (*span*). As equações 2.4 garantem que cada antena receberá um e apenas um canal. A restrição evita portanto a existência de soluções inviáveis. O conjunto de inequações 2.5 representa a restrição de que, para duas antenas que causem interferência, só uma delas pode receber a frequência especificada. Este grupo de inequações, portanto, modela as restrições de interferência. Por último, no conjunto de equações 2.6 tem-se as restrições de integralidade das variáveis de decisão.

Branch and Cut (BC) é uma técnica muito usada em conjunção com PI. A idéia básica do BC é resolver formas mais simples do problema original e usá-las para derivar a solução do problema. Sua principal dificuldade consiste em degenerar-se para uma enumeração completa em alguns casos. Em Aardal *et al.* (Aardal *et al.*, 1998), esse método é usado para fornecer soluções para formulação PAFVP mostrada acima. O primeiro passo para a aplicação do branch-and-cut no algoritmo proposto por Aardal *et al.* é realizar um pré-processamento da instância retirando redundâncias existentes na formulação original. Com isto o número de variáveis é bastante reduzido. Outro fator importante para que se encontrem boas soluções é o estabelecimento de limites inferiores adequados para o problema. Isto é feito usando o tamanho máximo das cliques encontradas no grafo de interferências. Com bons limites inferiores, o branch-and-cut consegue provar mais facilmente se uma solução é ou não ótima para o problema.

Baseado na formulação anterior que permite minimizar o número total de canais usados, foi proposta uma versão onde a função objetivo é a interferência total do sistema. Para isto, será necessário acrescentar as constantes  $c_{i,j} \in [0, 1] \quad \forall i, j \in V$  representando o valor da interferência co-canal entre cada par de antenas e  $a_{i,j} \in [0, 1] \quad \forall i, j \in V$  representando a interferência adjacente-canal. Então tem-se a seguinte formulação:

(PAFC)

$$\begin{aligned} \min \quad & \sum_{i,j,f} x_{i,f} x_{j,f} c_{i,j} + \sum_{i,j,f} x_{i,f} x_{j,(f+1)} a_{i,j} \\ & + \sum_{i,j,f} x_{i,f} x_{j,(f-1)} a_{i,j} \end{aligned} \quad (2.7)$$

tal que:

$$\sum_f x_{i,f} = 1 \quad \forall i \in V \quad (2.8)$$

$$x_{i,f} + x_{j,g} \leq 1 \quad \forall i, j \in V, \forall f, g \in C, |f - g| < d(i, j) \quad (2.9)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i \in V \quad (2.10)$$

Com esta formulação, a função objetivo representa o somatório da interferência co-canal e adjacente canal. A multiplicação das variáveis no somatório garante que os fatores só serão considerados quando as duas variáveis forem iguais a 1. Entretanto, com esta alteração a função objetivo não mantém a linearidade, tornando-se uma soma de termos quadráticos, o que torna mais difícil sua solução por métodos convencionais, como o uso de relaxações lineares.

## 2.6 Algoritmos aproximativos

Dada a dificuldade inerente do Problema da Alocação de Frequências, diversos algoritmos aproximativos para problemas NP-Difíceis passaram a ser propostos. O objetivo desses algoritmos é fornecer uma solução próxima do ótimo com um esforço computacional relativamente baixo. Alguns algoritmos aproximativos oferecem garantias quanto à qualidade do valor encontrado. Na Seção 2.6.1 será apresentado um algoritmo com essas características. Muitas heurísticas para o PAF são baseadas em heurísticas existentes para o problema da coloração de grafos. Nesta seção serão apresentadas algumas das que se mostraram mais efetivas na solução do problema.

### 2.6.1 Heurísticas com garantia de aproximação

Alguns algoritmos aproximativos oferecem uma garantia de aproximação com relação ao ótimo do problema. Mais formalmente, dado um algoritmo aproximativo  $A$  para um problema de minimização  $\Pi$  e um número racional  $\epsilon$ , diz-se que o algoritmo oferece uma aproximação de  $\epsilon$  se para cada solução  $s \in A(\Pi)$  retornada por ele, tem-se que

$$f(s) \leq \epsilon \cdot f(s^*)$$

onde  $s^* \in A(\Pi)$  representa a solução ótima para o problema. No caso de problemas de maximização, o mesmo é verdadeiro com a inversão da inequação. Ou seja:

$$f(s) \geq \epsilon \cdot f(s^*)$$

Pode-se provar que, a menos que  $P = NP$ , não existem algoritmos aproximativos para o PAF (Eisenblätter, 1997). Apesar deste resultado, nada impede que algoritmos aproximativos sejam desenvolvidos para versões simplificadas do problema. Esse foi o caminho seguido por McDiarmid e Reed (McDiarmid e Reed, 1997). Em seu trabalho, eles desenvolveram um algoritmo aproximativo para coloração de grafos de reticulado triangular (Figura 2.2), onde cada vértice possui uma demanda. Outra restrição imposta é a de que a função separação mínima seja da seguinte forma:

$$d(i, j) = \begin{cases} 1 & \text{se } (i, j) \in E(G) \\ 0 & \text{caso contrário.} \end{cases}$$

Para esse problema restrito, pode-se provar o seguinte teorema:

**Teorema 2** (McDiarmid e Reed) *Dado um grafo de reticulado triangular  $G$  e um vetor de demanda  $w$  para os vértices de  $G$ , existe um algoritmo  $\varepsilon$ -aproximativo polinomial  $\Pi$ , tal que*

$$f(\Pi(G, w)) \leq \varepsilon \cdot f(s^*)$$

onde  $s^*$  é uma solução ótima do problema, e  $\varepsilon \geq 4/3$ .

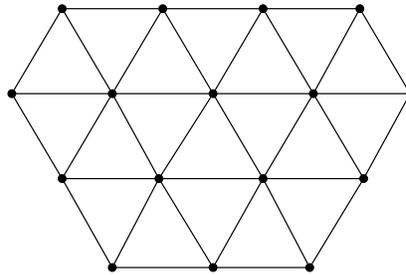


Figura 2.2: Um grafo de reticulado triangular.

### 2.6.2 Heurísticas sem garantia de aproximação

Uma segunda classe de algoritmos aproximativos é composta por heurísticas que não oferecem garantias de aproximação. Apresentaremos a seguir alguns dos algoritmos aproximativos mais importantes entre aqueles propostos para o PAF.

#### T-Coloração

A heurística T-Coloração recebeu esse nome por ter sido inicialmente usada no contexto do problema da T-coloração (Cozzens e Roberts, 1982). Seu objetivo principal é encontrar uma solução viável, através do uso da noção de “saturação” e “grau de espaçamento”.

*Saturação* de um vértice é o número de cores disponíveis para ele no momento. Esta idéia foi usada pela primeira vez com o algoritmo DSATUR, proposto por Brélaz (Brélaz, 1979; Peemöller, 1983) (como visto na Seção 2.4.2). O *Grau de espaçamento* representa o número de vértices vizinhos ao vértice atual com cores ainda não atribuídas. Usando estas duas medidas, o algoritmo da T-Coloração busca encontrar o próximo vértice ao qual deve-se alocar uma cor válida.

Usando esse algoritmo para resolução do PAF, é necessário calcular a cada iteração os valores para saturação e grau de espaçamento. O Algoritmo 2.2 mostra em pseudocódigo o funcionamento da heurística.

---

**Algoritmo 2.2** Heurística T-Coloração
 

---

```

1: Inicialize  $\text{sat}[i]$  para 0,  $\forall i \in \{1, \dots, |V|\}$ .
2: Inicialize  $\text{esp}[i]$  para 0,  $\forall i \in \{1, \dots, |V|\}$ .
3: Faça  $U = V$ .
4: while  $U \neq \emptyset$  do
5:   Selecione  $v_i \in U$  com maior valor combinado de  $\text{sat}[i]$  e  $\text{esp}[i]$ .
6:   Faça  $U = U - \{v_i\}$ 
7:   if Existem canais disponíveis para  $v_i$  then
8:     Atribua um canal aleatoriamente a  $v_i$ , entre aqueles disponíveis.
9:   else
10:    Atribua um canal aleatoriamente a  $v_i$ .
11:   end if
12:   {Atualização de valores}
13:   for cada  $j \in \{1, \dots, |V|\}$  do
14:     Faça  $\text{sat}[j]$  igual ao número de canais que o  $v_j$  não pode usar.
15:     Faça  $\text{esp}[j]$  igual ao número de vizinhos de  $v_j$  ainda não atribuídos.
16:   end for
17: end while

```

---

### Dual Guloso

O algoritmo dual guloso recebeu esse nome porque, apesar de usar uma estratégia gulosa, trabalha fazendo as escolhas inversas e evitando uma tomada de decisão entre diversas opções. A idéia do algoritmo foi usada para a solução de outros problemas como, por exemplo, o de localização em circuitos eletrônicos (Jünger *et al.*, 1994). Em primeiro lugar o algoritmo procura caracterizar o que seria uma má escolha para o par (antena, frequência). Para isto ele procura entre todas as combinações válidas de (antena, frequência) aquela que viola o maior número de restrições e que possui maior valor de interferência total, descartando-a em seguida. Esse processo é repetido até que sobre apenas um par de antena com sua frequência associada. Esse passo repetido até que todas as antenas tenham recebido uma frequência válida. No Algoritmo 2.3 vê-se um pseudo código da heurística, tal como apresentada por Borndörfer *et al.* (Borndörfer *et al.*, 1997).

**Algoritmo 2.3** Heurística Dual Guloso

---

```

1: Inicialize U para o conjunto de todas os pares (v, f) tal que v ∈ V e f ∈ C.
2: Faça pena[i,j]=0 ∀i ∈ {1, ..., |V|} e ∀j ∈ {1, ..., |C|}
3: while U ≠ ∅ do
4:   for Todos os pares (v, f) ∈ U do
5:     if (v, f) viola alguma restrição or (v, f) provoca interferência co-canal ou
       adjacente-canal then
6:       pena[v,f] = pena[v,f]+1
7:     end if
8:   end for
9:   Selecione o par (v, f) com menor valor de pena[v,f] (desempates são feitos alea-
       toriamente)
10:  Atribua o canal f à antena v
11:  Faça U = U − (v, f)
12: end while

```

---

É fácil perceber que o algoritmo dual guloso tem uma performance inferior à heurística T-Coloring: a cada iteração, todos os pares antena-freqüência válidos precisam ser analisados para a alocação de um novo vértice. Isto faz com que este algoritmo tenha um desempenho fraco em comparação a outros. Segundo Borndörfer *et al.* (Borndörfer *et al.*, 1997), o fator mais importante para o sucesso deste algoritmo é a escolha adequada da função que irá medir o quão má é uma combinação antena-freqüência. Com o uso de Fibonacci Heaps (Cormen *et al.*, 1990) para manter o conjunto de pares, pode-se obter uma complexidade de tempo igual a  $O(|C|^2|V| \log(|C||V|) + |C|^2|E|)$  e com espaço igual a  $O(|C||V| + |E|)$ .

**Fluxo de custo mínimo**

A heurística *Fluxo de custo mínimo* MCF, discutida em (Borndörfer *et al.*, 1997), foi criada com o intuito de equalizar o uso dos diferentes canais em uma alocação. Um modelo baseado no problema de fluxos com custo mínimo é usado no algoritmo.

MFC tenta modificar a alocação de freqüências  $y$  para uma alocação  $y'$ , mantendo inalterada a seguinte propriedade:  $\forall (v, w) \in E, v < w$ , uma das condições a seguir é válida

$$\left\{ \begin{array}{l} y(v) \leq y(w) \Rightarrow y'(v) \leq y'(w) \\ \quad \quad \quad \wedge \\ y(v) > y(w) \Rightarrow y'(v) \geq y'(w) \end{array} \right.$$

O objetivo dessas condições é produzir uma alocação que utilize o maior número de canais disponíveis, fazendo com que as contagens de uso dos diferentes canais fiquem mais próximas entre si. Um método usado para satisfazer as condições consiste em resolver um problema de fluxo de custo mínimo, tendo como base o grafo de interferência  $G$ . As funções  $d$ ,  $c^{co}$  e  $c^{ad}$  são usados para computar os coeficientes de custo e as capacidades dos arcos. Para usar esta heurística, é necessário resolver o problema de custo mínimo, o que pode ser feito através de métodos como o simplex. O resultado obtido, é uma alocação em que há uma distribuição melhor de canais entre os transmissores, o que provavelmente levará a uma menor interferência.

De acordo com resultados reportados (Borndörfer *et al.*, 1997), esta heurística não oferece ganhos substanciais em relação a outras metodologias discutidas, além do tempo elevado de execução, devido ao problema de programação linear envolvido. Por este motivo, o MCF não será usado como base de estudo nesta dissertação.

### 2.6.3 Metaheurísticas

Nesta seção, serão apresentados os principais resultados para o PAF desenvolvidos através de metaheurísticas.

#### Busca Tabu

BT tem sido usado por alguns pesquisadores para encontrar boas soluções para o PAF. Em um artigo de 1997, Adjakpl'e e Jumard (Adjakpl'e e Jumard, 1997) propuseram um algoritmo de BT para o PAF aplicado à telefonia celular. A sua metodologia usa um processo de alocação de frequências em blocos, de forma a manter as antenas pertencentes a uma mesma estação base com uma distância mínima entre si. Os movimentos busca na vizinhança consistem em trocar um bloco de canais por outro, a fim de minimizar a interferência. Outro tipo de movimento usado nesse algoritmo é a troca de todos os canais usados em uma célula pelos canais que geram menor interferência no momento até que a demanda da célula seja satisfeita. Os experimentos realizados com esse algoritmo usaram duas instâncias de planejamento de frequências da empresa canadense Bell Mobilité. Entretanto os testes realizados envolveram apenas uma comparação com um algoritmo de busca local, o que não permite avaliar sua qualidade em relação a outros métodos.

## Algoritmos Genéticos

Os AGs têm sido bastante usados como um método para resolução de problemas de alocação de frequências. Crompton *et al.* (Crompton *et al.*, 1994) apresenta um AG para o PAF que leva em consideração múltiplos objetivos, que são codificados através de fatores multiplicativos aplicados à função objetivo. Na sua implementação, duas representações do problema foram usadas. Na representação R1 o tamanho de cada cromossomo é  $n$ , o número de transmissores. Cada elemento é o número do canal alocado àquele transmissor. Na representação R2, os transmissores que receberam o mesmo canal são agrupados seqüencialmente, em ordem crescente dos canais. Segundo experimentos realizados, a representação R2 apresentou desempenho superior. No mesmo trabalho, uma forma simples de paralelização do AG é apresentada, onde vários processadores executam o AG, havendo troca ocasional dos cromossomos com melhor função objetivo entre as populações.

## GRASP

A metaheurística GRASP (General Randomized Adaptive Search Procedure) também foi usada para dar um solução para o PAF. Em seu artigo, Liu *et al.* (Liu *et al.*, 2000) propõem um GRASP para problema da alocação de frequências. Entretanto este trabalho apresenta algumas diferenças com relação ao que foi desenvolvido nesta dissertação. As mais importantes são:

- A função objetivo minimizada pelo GRASP de Liu *et al.* diz respeito apenas às violações da separação mínima entre canais. Neste trabalho, além de levar isto em consideração, objetiva-se minimizar a interferência total do sistema (co-canal e adjacente-canal), conforme proposto em (Borndörfer *et al.*, 1997).
- O método usado no referido artigo não leva em consideração a estrutura diferenciada do problema, aplicado à telefonia celular. Nos algoritmos apresentados neste trabalho, por outro lado, considera-se os transmissores agrupados em uma conjunto de estações base, o que permite criar alocações mais eficientes.
- Por fim, o algoritmo de implementação do GRASP apresentado por Liu *et al.* é diferente daquele proposto nesta dissertação. Por exemplo, para realização de busca local no referido artigo é empregado o método de têmpera simulada, enquanto neste trabalho uma heurística gulosa é usada.

Devido às diferenças na função objetivo empregada nas duas propostas, não pôde-se comparar os resultados encontradas pelo GRASP de Liu *et al.* e o GRASP apresentado neste trabalho.

## 2.7 Formulações alternativas

Uma tendência atual vista em alguns trabalhos acerca do PAF é a definição de novas formas de modelagem, nas quais estratégias diferentes de alocação possam ser usadas com melhores resultados. Pode ser citado como exemplo o trabalho de Capone e Trubian (Capone e Trubian, 1999), onde um novo modelo matemático, que incorpora a idéia de interferência cumulativa é usada. Um segundo exemplo pode ser visto no trabalho de N. Dunkin *et al.* (Dunkin *et al.*, 1998). Nele as restrições existentes no PAF são divididas em dois tipos: restrições fortes (hard restrictions) e fracas (soft restrictions). Isto permite aos autores a utilização do método de satisfação de restrições (constraint satisfaction), também usado em (Ottosson e Carlsson, 1997). Em (Borndörfer *et al.*, 1999) uma modelagem alternativa para o PAF usando programação inteira é apresentada. Tal modelagem leva em consideração a potência dos transmissores como um fator determinante da interferência.

## 2.8 Conclusão

Neste capítulo foram apresentados vários métodos computacionais usados para resolução do PAF. Todos esses métodos foram desenvolvidos para a versão mais geral do problema e não levam em consideração características do PAF para telefonia celular, apesar de alguns deles já terem sido usados com esse fim. Mesmo tratando-se de métodos bastante diferentes entre si, pode-se dizer que sua principal desvantagem é não serem adequados a algumas características importantes de sistemas de telefonia celular já discutidos, como a localização de transmissores em estações base, interferência local fixa, etc.

Neste trabalho será apresentada uma solução melhorada para o PAF celular, usando a metodologia GRASP, que será discutida no próximo capítulo.

# Capítulo 3

---

## A Metaheurística GRASP

*“Em geral, o progresso parece ser maior do que realmente é.”*

Nestroy

### 3.1 Introdução

GRASP (Greedy Randomized Adaptive Search Procedure) é uma metaheurística proposta por Feo e Resende (Feo e Resende, 1995), baseada em dois processos principais: um construtor de soluções com a característica de gerar boas soluções de forma aleatória e uma heurística de busca local que atua na melhoria destas soluções. GRASP tem sido aplicado com sucesso em um grande número de problemas de otimização combinatória, como os problemas de agendamento (Feo *et al.*, 1991), roteamento (Hjorring, 1995), localização de fábricas (Delmaire *et al.*, 1997), máximo conjunto independente (Feo *et al.*, 1994), entre outros.

Este capítulo tem como objetivo realizar um estudo detalhado da metaheurística GRASP. O capítulo acha-se dividido como segue: Na Seção 3.2, será descrito o funcionamento de um GRASP e apresentados seus principais componentes. O construtor do GRASP será discutido na Seção 3.2.3. Na Seção 3.2.4 será examinado o procedimento de busca local e sua importância.

## 3.2 Descrição

### 3.2.1 Principais características

Uma execução do GRASP é constituída de uma série de iterações. Em cada iteração, o construtor é executado, gerando uma solução inicial, que é passada ao procedimento de busca local. O resultado final do GRASP será a melhor das soluções encontradas em cada uma das iterações. No Algoritmo 3.1, tem-se uma descrição em pseudo-código da metaheurística.

---

**Algoritmo 3.1** Metaheurística GRASP

---

- 1: Entrada de dados
  - 2: **while** Critério não satisfeito **do**
  - 3:   Construtor guloso aleatório
  - 4:   Busca local
  - 5:   Atualizar melhor solução
  - 6: **end while**
- 

Depois da entrada de dados na linha 1, o GRASP entra em repetição enquanto o critério de parada não for alcançado. O critério de parada pode ser tempo de processamento, número de iterações ou qualquer outro que faça sentido para a aplicação. Na linha 3, tem-se o procedimento de construção. Na linha 4 entra em cena o procedimento de busca local, que procura melhorar a solução encontrada através de uma busca apenas em sua vizinhança. Finalmente, na linha 5 ocorre a atualização da melhor solução encontrada até o momento, caso a solução atual tenha sido melhor que as anteriores. Nas próximas seções, será feita uma descrição de cada um dos componentes desta metaheurística.

### 3.2.2 Intensificação e diversificação no GRASP

No projeto de metaheurísticas para otimização combinatória, duas estratégias são freqüentemente utilizadas para a obtenção de soluções globais:

1. *diversificação*, usada como um método para evitar que a metaheurística fique presa em mínimos locais e
2. *intensificação*, que permite garantir que um ótimo global será encontrado, desde que este encontre-se na vizinhança analisada.

Diversificação é a estratégia usada em uma metaheurística para explorar o espaço de soluções de forma variada, a fim de escapar de mínimos locais. A diversificação do espaço de busca normalmente implica em aceitar novas soluções, mesmo que elas sejam piores do que as existentes na memória. Pode-se citar exemplos de como isso é realizado em outras metaheurísticas. Um Algoritmo Genético (AG), por exemplo, promove a diversificação através da criação de populações com um número considerável de indivíduos diferentes. Outro método usados em AGs com esse fim é o operador de mutação, que cria indivíduos aleatoriamente diferentes dos que existem na memória. Na metaheurística Busca Tabu, o uso da lista de soluções proibidas também visa a diversificação do espaço busca, reduzindo o risco de que ela retorne a pontos da vizinhança que já foram explorados. Finalmente, na metaheurística de Têmpera Simulada, a variação de temperatura é o elemento que introduz diversificação nas soluções encontradas. Isto ocorre porque quanto maior a temperatura, maior a probabilidade de que uma solução pior que a atual seja aceita pelo algoritmo.

A intensificação é a estratégia oposta da diversificação e tem como objetivo encontrar a melhor solução à qual uma vizinhança pode levar. Desta forma, o processo de intensificação tende a usar algoritmos gulosos e míopes (algoritmos que só enxergam uma pequena parte do espaço de busca). Como exemplos de uso efetivo da diversificação, pode-se citar a metaheurística de Busca Tabu, que normalmente usa movimentos que levam para a melhor solução vizinha. Algoritmos Genéticos empregam a seleção e o operador de cruzamento com o objetivo de intensificar as características dos melhores indivíduos de uma população. Por fim, pode-se dizer que na metaheurística de têmpera simulada, a redução de temperatura induz a uma convergência gradual do algoritmo.

GRASP foi projetado tendo como objetivo uma convergência rápida. Sua idéia central é usar intensificação e diversificação da forma mais direta possível. Seu algoritmo usa um laço em que, a cada iteração, um construtor é executado, seguido imediatamente de uma busca local para melhoria da solução gerada. O construtor funciona no GRASP como o componente de diversificação, permitindo que a busca seja reiniciada em cada iteração, sempre a partir de pontos diferentes. Uma outra característica associada ao construtor GRASP é a de gerar soluções que, na maioria dos casos, possuem boa qualidade. Isto evita que o procedimento de busca precise realizar grande esforço computacional para melhorar a solução inicial e, conseqüentemente, melhora a eficiência da metaheurística.

### 3.2.3 Construtor

No GRASP, o construtor de soluções é guloso, aleatório e adaptativo (Greedy Randomized Adaptive). É guloso, pois privilegia as “melhores” soluções, de acordo com algum critério previamente estabelecido. É aleatório, pois a escolha dos elementos que formam uma nova solução é feita tendo como base um critério estocástico, com uma tendência para os melhores elementos. Isso permite uma diversificação nas características das soluções geradas. Por fim, o construtor é adaptativo, pois, durante sua execução, o critério de classificação dos elementos que irão compor a solução é atualizado de forma dinâmica.

Para a grande maioria dos problemas de otimização combinatória, uma solução é formada por uma série de elementos discretos. No caso do PAF, uma solução é um conjunto de canais atribuídos a transmissores. Portanto, os elementos dos quais a solução para o PAF é constituída são os transmissores e os canais.

O algoritmo de construção do GRASP cria a solução desejada de forma incremental. A cada iteração da construção, o algoritmo usa um dos melhores elementos disponíveis no momento. Para evitar o determinismo, o construtor usa uma lista onde serão armazenados os elementos que satisfazem a um parâmetro de adequação  $\alpha$ . Esta lista é chamada de *lista restrita de candidatos* (Restricted Candidate List – RCL). Na Figura 3.1 vê-se uma representação da RCL com os elementos que satisfazem o parâmetro  $\alpha$ .

O Algoritmo 3.2 mostra um pseudocódigo do construtor. Após a fase de inicialização da RCL, o construtor escolhe de forma aleatória um dos elementos da lista ( $x$ ). O próximo passo consiste em reordenar a lista de elementos restantes de acordo com as alterações de prioridades causadas pela escolha de  $x$ . Esta fase caracteriza o GRASP como um processo adaptativo pois a ordenação da RCL é constantemente atualizada. Estes passos são repetidos enquanto a solução estiver incompleta.

---

#### Algoritmo 3.2 Procedimento de construção do GRASP

---

**Require:** Parâmetro  $\alpha$ .

- 1: Faça solução  $S = \emptyset$ .
  - 2: **while** Solução incompleta **do**
  - 3:   Ordene os candidatos em uma lista (RCL)
  - 4:   Selecione o candidato  $x$  que satisfaça ao critério de seleção  $\alpha$ .
  - 5:   Faça  $S = S \cup \{x\}$
  - 6: **end while**
-

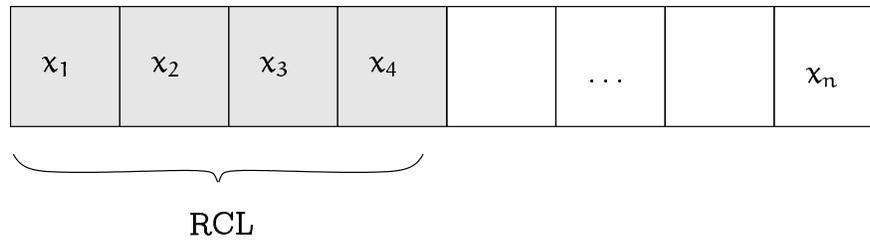


Figura 3.1: Lista de possíveis elementos de uma solução  $(x_1, x_2, \dots, x_n)$ , ordenados de acordo com um critério de adequação. Os elementos da RCL  $(x_1, \dots, x_4)$  são os que satisfazem o parâmetro  $\alpha$ .

### 3.2.4 Busca Local

Após o algoritmo construtor ser executado, o próximo passo do GRASP é usar a solução gerada como ponto de partida para uma busca local. Esta busca tem como objetivo permitir que a vizinhança da solução atual seja explorada, de forma a encontrar um ótimo, possivelmente local, a partir daquela configuração.

Existe um grau de liberdade muito grande para a definição da metodologia de busca local no GRASP. É necessário apenas que, dada uma solução válida para o problema, o algoritmo consiga retornar uma solução melhorada, ou a mesma, caso não seja possível realizar nenhuma melhoria.

Vários esquemas de busca local têm sido idealizados. Entre eles, um dos mais usados é o método de “descida da montanha”. Esse algoritmo funciona realizando “perturbações” na solução que permitam uma melhoria. Uma *perturbação* é qualquer alteração pontual na solução que altere sua função objetivo. Partindo de um solução específica, o conjunto de soluções que podem ser encontradas realizando apenas uma perturbação é chamado de *vizinhança da solução*.

Uma descrição em linguagem natural do método de descida da montanha pode ser vista no Algoritmo 3.3. O primeiro passo do algoritmo é inicializar a solução e o contador de tentativas. Logo após o algoritmo entra em um laço, onde serão feitas uma série de tentativas para melhorar o valor de função objetivo da solução. Isso é realizado gerando todos os elementos da vizinhança da solução atual e selecionando aquele com menor valor objetivo ( $s'$ ). A Figura 3.2 mostra um exemplo de vizinhança de uma solução. Caso  $f(s') \leq f(s)$ , pode-se fazer com que  $s$  seja igual a  $s'$  e continuar no laço. Caso contrário, houve uma falha na tentativa de melhorar a solução atual. Isto é registrado através

do incremento do contador de tentativas. Quando o número de tentativas frustradas de melhoria ultrapassam o parâmetro MAX\_TRY, o laço é terminado. Este algoritmo tem como pontos fortes a simplicidade de implementação e eficiência.

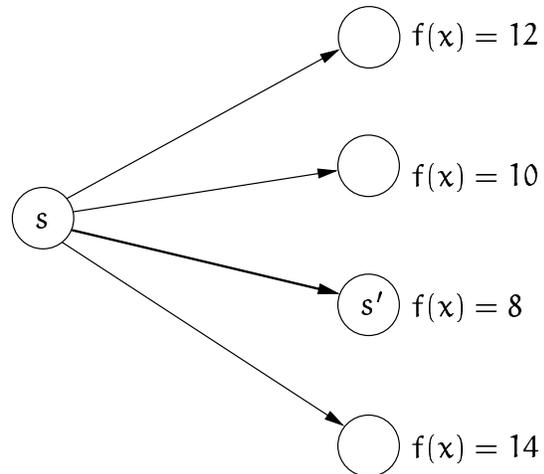


Figura 3.2: Exemplo de vizinhança para uma solução qualquer  $s$ , durante a execução do algoritmo de busca descida da montanha.  $s'$  é a solução vizinha de  $s$  com menor valor de função objetivo.

---

**Algoritmo 3.3** Método descida da montanha.

---

**Require:** Solução inicial  $s_i$ , número de tentativas MAX\_TRY.

```
1: Faça  $i = 0$ 
2: Faça  $s = s_i$ 
3: while  $i < \text{MAX\_TRY}$  do
4:   Faça  $V$  igual à vizinhança de  $s$ 
5:   Selecione o elemento  $s'$  de  $V$  com menor valor de função objetivo.
6:   if  $f(s') \leq f(s)$  then
7:     Faça  $s = s'$ 
8:     Faça  $i = i + 1$ 
9:   else
10:    Faça  $i = 0$ 
11:   end if
12: end while
```

---

# Capítulo 4

---

## Proposta de um GRASP para o PAF

*“Some people thinks that mathematics is a serious business that must always be could and dry; but we think mathematics is fun, and we aren't ashamed to admit the fact”*

(Graham *et al.*, 1989)

### 4.1 Introdução

Este capítulo tem como objetivo expor uma proposta de GRASP para o Problema da Alocação de Frequências (PAF). O PAF, definido no Capítulo 2, visa minimizar a interferência associada ao uso do espectro de frequências, para um conjunto de estações base, em um sistema de telefonia celular. O objetivo do problema é especificar uma alocação  $y : C \rightarrow \mathbb{N}$  de transmissores para canais, de modo que transmissores adjacentes no grafo de interferência recebam frequências separadas por uma certa distância. Esta separação mínima é indicada pela função  $d : C \times C \rightarrow \mathbb{N}$ .

A função objetivo para este problema visa minimizar o valor da interferência total. Os tipos de interferência que serão levados em consideração para o PAF são: interferência local, co-canal e adjacente-canal (veja Seção 1.2.3). Esse problema não possui até o momento nenhum algoritmo eficiente que encontre a solução exata em tempo polinomial. No decorrer deste capítulo, será abordado o método proposto de solução aproximada para o PAF. A abordagem, baseada na metaheurística GRASP, apresenta boas soluções em comparação com outros métodos existentes, conforme será visto no Capítulo 5.

## 4.2 Descrição

### 4.2.1 Instância e função objetivo

A fase inicial do algoritmo GRASP constitui-se da entrada de dados. Nesta fase, as informações sobre o que será resolvido é lida, através de um arquivo de descrição da instância.

Uma instância do PAF é constituída pelos seguintes dados:

- O número de transmissores ( $n \in \mathbb{N}$ );
- O número de estações base ( $m \in \mathbb{N}$ );
- Um *vetor de demanda*, que informa quantos transmissores possui a estação base  $i$ , para  $i \in \{1, \dots, m\}$  ( $m_i \in \mathbb{N}$ );
- Uma matriz  $m \times m$  com os valores de separação mínima de canais entre cada par de transmissores ( $D_{i,j} \in \mathbb{N}$ );
- Uma matriz  $m \times m$  com valores reais entre 0 e 1, representando a interferência co-canal entre cada par de transmissores ( $C_{i,j} \in [0, 1]$ );
- Uma matriz  $m \times m$  com valores reais entre 0 e 1, representando a interferência adjacente-canal entre cada par de transmissores ( $A_{i,j} \in [0, 1]$ );

De um modo geral, é difícil encontrar algoritmos que gerem soluções viáveis para o PAF, já que mesmo o problema de achar estas soluções já é NP-Completo (Borndörfer *et al.*, 1997). Portanto, optou-se por usar uma função objetivo que leva em consideração o número de restrições violadas pela solução. O número de restrições violadas é uma medida de quanto uma solução é inviável. A função objetivo  $f$  que o programa tenta minimizar é uma soma da interferência total do sistema (2.1) e do número de restrições violadas (2.2), multiplicada por um peso  $\rho$ . Desta forma, um custo maior é sempre atribuído a uma solução com mais restrições violadas. Portanto, tem-se:

$$f(x) = \rho \cdot \sum_{|y(v)-y(w)| < d(v,w)} 1 + \sum_{y(v)=y(w)} c^{co}(v,w) + \sum_{|y(v)-y(w)|=1} c^{ad}(v,w) \quad (4.1)$$

### 4.2.2 Construtor

O construtor usado pelo GRASP é baseado em uma modificação do algoritmo de T-Coloração, apresentado na Seção 2.6.2. A principal diferença, entretanto, entre este algoritmo de construção e o da T-Coloração, é que em vez de ser usado para criar uma solução completa, o algoritmo apenas ordena os elementos da RCL em ordem decrescente. O algoritmo da T-Coloração foi usado como base para o construtor do GRASP pelo fato de ser este um método relativamente simples, em comparação com outras metodologias existentes. Além disso esse algoritmo é considerado como um dos que oferecem boas soluções para o problema (veja (Borndörfer *et al.*, 1997)). Em último lugar, verificou-se que o algoritmo se adapta com facilidade à tarefa alternativa de construção da RCL, passo importante para a geração de soluções no GRASP.

Seguindo a metodologia usada por construtores GRASP, a solução completa é criada em várias iterações. Em cada iteração, o algoritmo escolhe o próximo transmissor ao qual será atribuída uma frequência. O critério usado para ordenação dos transmissores é o número de frequências disponíveis. Quanto menor o número de frequências livres para alocação em um transmissor, maior deve ser sua prioridade.

Como explicado na Seção 3.2.3 o construtor evita o determinismo através do uso da lista restrita de candidatos (RCL). Na RCL proposta, os transmissores são ordenados em ordem crescente do número de canais disponíveis para alocação. No Algoritmo 4.1 vê-se uma descrição de como é construída a RCL.

---

#### Algoritmo 4.1 Criação da RCL

---

**Require:** RCL é uma lista vazia

- 1: Inicialize  $\text{sat}[i]$  para 0,  $\forall i \in \{1, \dots, |V|\}$ .
  - 2: Inicialize  $\text{esp}[i]$  para 0,  $\forall i \in \{1, \dots, |V|\}$ .
  - 3: Faça  $U \leftarrow V$ .
  - 4: **while**  $U \neq \emptyset$  **do**
  - 5:   Selecione  $v_i \in U$  com maior valor combinado de  $\text{sat}[i]$  e  $\text{esp}[i]$ .
  - 6:   Coloque  $v_i$  no fim da RCL
  - 7:   Faça  $U \leftarrow U - \{v_i\}$
  - 8:   Atualize os valores de  $\text{sat}[i]$  e  $\text{esp}[i]$ ,  $\forall i \in \{1, \dots, |V|\}$ .
  - 9: **end while**
- 

Esse algoritmo é bastante parecido com o Algoritmo 2.2 para a heurística de T-Coloração, apresentada na Seção 2.6.2.

Na Figura 4.1, vê-se um exemplo que esclarece a política de ordenação da RCL. As

antenas 1 e 2 precisam receber um canal. A antena 1 possui três vizinhos com canais já alocados, enquanto a antena 2 possui apenas dois vizinhos nessa situação. A antena 1 terá prioridade maior na RCL, pois ela possui menos canais disponíveis (os três canais vizinhos já estão ocupados). A idéia do algoritmo é garantir que as antenas com poucas opções de alocação sejam processadas primeiro.

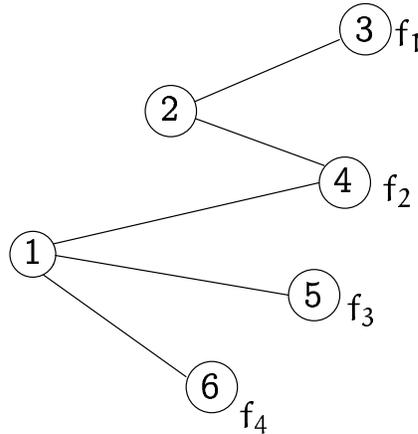


Figura 4.1: Exemplo para o construtor do GRASP

Após a criação da RCL, o transmissor a ser usado é selecionado de forma aleatória entre seus  $\alpha$  primeiros elementos. Assim, mesmo os elementos que não são o melhor possuem uma probabilidade uniforme de serem sorteados. Finalmente, depois de escolhido o transmissor, um canal é atribuído de forma aleatória entre aqueles que estão disponíveis no momento.

### 4.2.3 Busca local

Verificou-se durante a pesquisa realizada que, utilizando um método diferente de explorar a estrutura das instâncias do PAF para telefonia celular, o algoritmo de busca local seria mais eficiente. Por exemplo, em vez fazer uma perturbação em apenas um transmissor, o que muitas vezes provoca uma violação das restrições dentro da estação base, seria melhor considerar perturbações que abrangessem todos os transmissores localizados na estação base.

Portanto, o algoritmo usado para a fase de busca local do GRASP foi criado com a intenção deliberada de explorar a estrutura em células do PAF. Como se sabe, os transmissores em um sistema de telefonia móvel encontram-se agrupados em estações

base. Entre eles, ocorre interferências locais constantes, o que faz com que o valor da separação mínima entre canais  $d(i, j)$  seja fixo. Usando este conhecimento, pode-se fazer um algoritmo de busca local mais eficiente, alocando canais para toda uma estação base, em vez de fazer isso para cada transmissor.

O algoritmo é baseado no método de descida da montanha para busca local, que foi descrito na Seção 3.2.4. Uma característica importante para o correto funcionamento desse algoritmo é a definição do conceito de perturbação e vizinhança. Uma perturbação é uma alteração local na solução, que irá provavelmente mudar seu valor de função objetivo. Vizinhança de uma solução é o conjunto de soluções que podem ser derivadas da primeira a partir de uma perturbação. Na maioria dos trabalhos sobre o PAF que envolvem a definição do conceito de vizinhança, ela constitui-se na alteração do canal alocado a um único transmissor. Entretanto, o tipo de perturbação usada no GRASP proposto aloca frequências para uma estação base completa, em vez de alterar apenas a frequência de um transmissor.

A idéia básica do algoritmo consiste em, a cada iteração, selecionar uma estação base aleatória. Em seguida, escolhe-se uma faixa do espectro, também de modo aleatório, que fornecerá os canais para a alocação de frequências àquela estação base. Na Figura 4.2 ilustra-se como se dá a seleção dos canais que serão usados. Por exemplo, para uma estação base que possua 5 transmissores, serão usados 5 canais, separados por uma distância adequada, igual à separação mínima para interferência local. Se esta distância mínima for igual a 4 e selecionarmos inicialmente o canal 40, os próximos a serem usados serão os canais 44, 48, 52 e 56. No caso em que não haja canais adicionais disponíveis para alocar a toda a estação base, novos canais são usados começando do canal 1. Por exemplo, supondo que no caso acima o número de canais disponíveis fosse de apenas 50, a seqüência resultante de canais seria 40, 44, 48, 2 e 6. Desta forma é feita a alocação de canais para toda a estação base, ao mesmo tempo garantindo a viabilidade desta escolha, pelo menos dentro da EB.

Um exemplo de como opera o algoritmo de busca local é mostrado na Figura 4.3. Nesta figura vê-se uma instância simples do problema, com apenas quatro estações base. Para realizar uma perturbação na solução, é necessário selecionar uma das estações base aleatoriamente, que será, neste exemplo, a estação base B. Logo após, o algoritmo seleciona um canal inicial para a alocação, que na figura é o canal 25. Supondo-se que a separação entre canais seja igual a 4, pode-se fazer a alocação dos três transmissores que pertencem a esta estação base selecionando-se canais consecutivos separados por

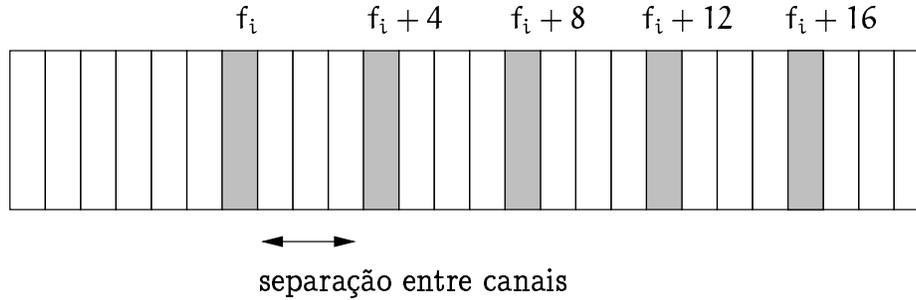


Figura 4.2: Exemplo para busca local. Os canais selecionados pelo algoritmo são marcado como  $f_i$ ,  $f_i + 4$ ,  $f_i + 8$  e  $f_i + 12$  e  $f_i + 16$ . A separação entre canais neste caso é de 4.

esta distância. Os canais resultantes da alocação exemplo são, portanto, os de número 25, 29 e 33.

### 4.3 GRASP reativo

No algoritmo do GRASP, o único parâmetro que precisa ser escolhido de forma arbitrária é o valor de  $\alpha$ , usado na fase de construção. O valor de  $\alpha$  representa o número de elementos da lista de candidatos (RCL) que será considerada para a escolha de um novo elemento. Normalmente, escolher o melhor valor para  $\alpha$  envolve a realização de uma série de testes para a calibragem do algoritmo, a fim de definir qual valor é o mais efetivo para o problema em questão. Entretanto, pode acontecer que o valor ideal para  $\alpha$  seja específico não somente para o problema, mas para cada instância que está sendo considerada. Neste caso, a dificuldade para determinação do  $\alpha$  torna-se ainda maior.

Para que o GRASP tivesse uma flexibilidade maior e com o objetivo de facilitar a calibragem do algoritmo, o ideal seria poder obter de forma automática o valor do parâmetro  $\alpha$ , tendo como entrada uma instância do problema que se deseja resolver. A fim de alcançar esse objetivo optou-se nesta dissertação pelo uso do método proposto por Prais e Ribeiro (Prais e Ribeiro, 2000), denominado *GRASP Reativo*. A essência do GRASP Reativo é a manutenção de um conjunto de valores candidatos  $\mathcal{A} = \{\alpha_1, \dots, \alpha_n\}$  para o parâmetro  $\alpha$ . Cada item de  $\mathcal{A}$  possui uma probabilidade  $p_i$  associada. A cada iteração do GRASP Reativo, um valor  $\alpha_i$  é escolhido a partir de  $\mathcal{A}$ , com probabilidade  $p_i$ , para ser usado pelo construtor. As probabilidades  $p_i$  são atualizadas depois de um

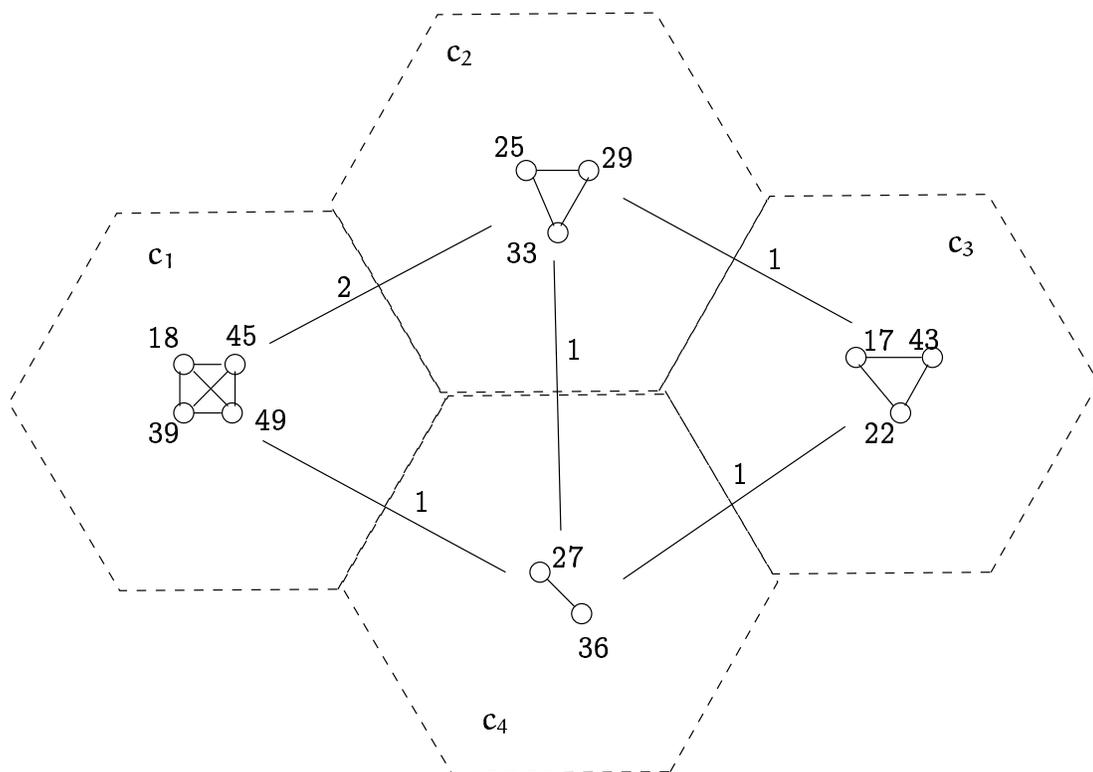


Figura 4.3: Exemplo de resultado do algoritmo busca local. A estação base B foi selecionada pelo algoritmo e uma alocação em bloco é realizada.

certo número de iterações, fazendo com que o algoritmo encontre automaticamente os melhores valores para  $\alpha$ .

A forma de atualização das probabilidades  $p_i$  de escolha dos  $\alpha_i$  pode variar de acordo com a aplicação. Uma função descrita por Prais e Ribeiro (Prais e Ribeiro, 2000) para esse fim usa um conjunto  $\bar{A} = \{a_1, \dots, a_n\}$ , onde cada elemento  $a_i$  corresponde ao valor médio das soluções encontradas pelo construtor, usando o valor  $\alpha_i$ , durante um certo número de iterações. Ao final do número pré-definido de iterações, calcula-se  $q_i, i \in \{1, \dots, n\}$ , que representa a percentagem do melhor alcançada por  $\alpha_i$ :

$$q_i = \left( \frac{F(S^*)}{A_i} \right)^\delta$$

e finalmente atualiza-se as probabilidades  $p_i, i \in \{1, \dots, n\}$ :

$$p_i = q_i / \sum_{j=1}^m q_j.$$

O método GRASP Reativo foi por implementado e aplicado ao PAF. São usadas as mesmas indicações mostradas no artigo acerca dos parâmetros de calibragem. O resultados obtidos são apresentados no Capítulo 5.

## 4.4 GRASP e Religação de Caminhos

*Religação de Caminhos* (Path Relinking – RC) é uma metodologia de busca local aplicada a metaheurísticas. Ela foi sugerida pela primeira vez por Glover e Laguna (Glover e Laguna, 1997), em um trabalho acerca da metaheurística Busca Tabu. Seu objetivo principal é fornecer um método de integração das estratégias de diversificação e intensificação. A metodologia explora o “caminho”, através do espaço de busca, que leva a soluções de alta qualidade, também chamadas de *soluções guia*. Fluerent e Glover (Fluerent e Glover, 1997) propuseram o uso desta técnica como uma forma de melhorar os resultados obtidos pelo GRASP. Finalmente, Religação de Caminhos foi usada com sucesso por Laguna e Martí (Laguna e Martí, 1999) em um GRASP para o problema de redução de cruzamentos no desenho de grafos.

O método de Religação de Caminhos consiste no uso de boas soluções já conhecidas, como se fossem “soluções guia”. Essas soluções terão um papel de facilitadoras na procura por novas melhorias. Religação de Caminhos busca incorporar elementos existentes em soluções de alta qualidade. Como diz o nome, o funcionamento da heurística

consiste em encontrar um “caminho” através do espaço de soluções que permita ligar a solução atual à solução de alta qualidade que está sendo considerada. Nesse processo, existe uma boa probabilidade de encontrar-se soluções melhores que a solução anterior. Na Figura 4.4 vê-se um exemplo de como isso pode ocorrer. Nesta figura, a linha cheia representa o caminho original, através do espaço de busca, para que se chegasse à solução guia. A linha pontilhada representa o novo caminho seguido quando se tenta achar uma solução que possua os mesmos atributos da solução guia. Entre os pontos inicial e final o processo de busca pode, em algum momento, passar por uma solução com função objetivo melhor que a solução guia original.

Uma outra forma de ver o algoritmo de Religação de Caminhos é como um método que faz com que a solução inicial torne-se cada vez mais parecida com a solução guia. Mesmo que elas sejam totalmente diferentes no início, ao término do processo a solução atual terá, na pior das hipóteses, função objetivo igual à da solução guia. Entretanto, existe uma boa possibilidade de que antes que isso aconteça possa-se encontrar uma solução com função objetivo ainda menor.

Optou-se pelo emprego do método Religação de Caminhos no GRASP proposto como uma forma de melhorar a qualidade das soluções. O método RC permite que isso aconteça através de uma busca adicional, direcionada por boas soluções já encontradas pelo GRASP. A seguir será feita uma descrição detalhada do funcionamento deste método.

A execução do método Religação de Caminhos se dá após a fase de busca local do GRASP. Nesse momento, pode-se usar a solução existente como ponto de partida. No Algoritmo 4.2 vê-se a alteração realizada no fluxo de execução padrão do GRASP. Um dos requisitos para a execução do RC é a criação de um conjunto de soluções guia, que será denotado por  $S_g$ . O número de soluções  $n_g = |S_g|$  existentes nesse conjunto pode entrar como um parâmetro do programa, mas geralmente um número fixo de soluções seria adequado. Durante as  $n_g$  primeiras iterações do GRASP, nenhuma atividade é realizada pelo método Religação de Caminhos, pois neste momento o conteúdo inicial  $S_g$  estará sendo criado. Percebe-se que quanto maior o valor de  $n_g$ , maior o tempo em que o método não realizará nenhum trabalho.

A atualização de  $S_g$  se dá à medida que novas soluções com boa solução objetivo são encontradas pelo GRASP. Como mostrado na linha 13 do Algoritmo 4.2, para cada nova solução encontrada pelo GRASP, verifica-se se ela tem melhor valor de função objetivo que o pior dos elementos de  $S_g$  (que será chamado de  $\bar{s}$ ). Caso isso seja verdade, troca-se

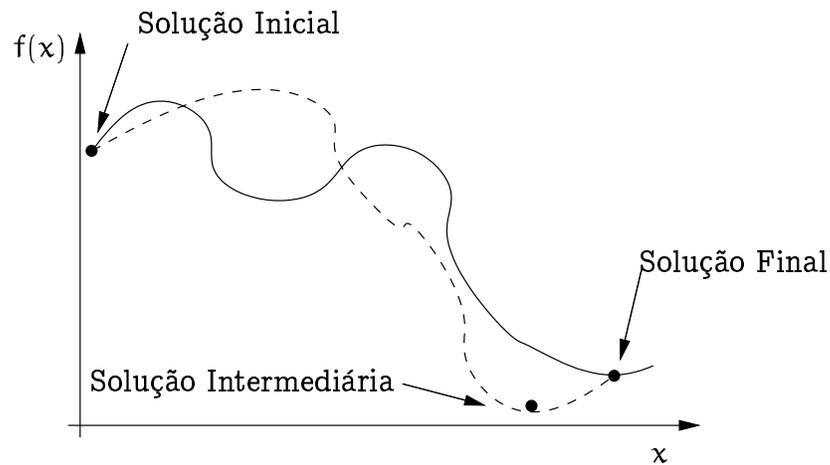


Figura 4.4: Exemplo de Religação de Caminhos. A linha cheia representa o caminho original no espaço de busca. A linha pontilhada representa o novo caminho de religação das duas soluções. Eventualmente o novo caminho pode passar por uma solução melhor que a final.

---

#### Algoritmo 4.2 Metaheurística GRASP com Religação de Caminhos

---

```

1: Entrada de dados
2: Faça  $i \leftarrow 1$ 
3: while Critério não satisfeito do
4:   Construtor guloso aleatório
5:   Busca local
6:   if  $i > \text{TAM\_PR}$  then
7:     Religação de Caminhos
8:   else
9:     inclua solução atual no conjunto  $S_g$ 
10:  end if
11:  Atualizar melhor solução
12:  Faça  $\bar{s} \leftarrow$  pior solução em  $S_g$ 
13:  if  $f(s) < f(\bar{s})$  then
14:    Faça  $S_g \leftarrow S_g \bar{s}$ 
15:    Faça  $S_g \leftarrow S_g \cup s$ 
16:  end if
17:  Faça  $i \leftarrow i + 1$ 
18: end while

```

---

o elemento  $\bar{s}$  por  $s$  em  $S_g$ .

Resta ainda analisar em detalhes como é realizado o procedimento de Religação de Caminhos. Os passos necessários para a sua execução podem ser resumidos como segue. Em primeiro lugar, é escolhido um elemento de  $S_g$  que passará a ser a solução guia. Esse elemento será chamado  $s'$ . Logo após, o algoritmo entra em um laço que é executado para cada transmissor da solução. Ele consiste em alterar o canal alocado ao transmissor  $i$  na solução original para o mesmo canal do transmissor  $i$  na solução guia. Fazendo isto, permite-se que a solução  $s$  torne-se a cada passo mais parecida com  $s'$ . Após essa alocação, o resultado pode ser melhorado através da aplicação de uma rápida busca local. É isso o que se vê na linha 6 do algoritmo. Note que existe a possibilidade de  $s$  ter piorado seu valor de função objetivo após essa troca, portanto a busca local é um passo importante para que esse método possa encontrar boas soluções. Em seguida, o algoritmo compara o valor encontrado com o melhor valor até o momento e faz as atualizações necessárias. Finalmente o algoritmo volta ao início do laço.

Os valores para a função objetivo da solução encontrada podem variar entre as diversas execuções deste laço. Sabe-se entretanto que a cada passo a solução  $s$  torna-se-á mais parecida com  $s'$ , o que acontecerá ao final do algoritmo. No Algoritmo 4.3 tem-se uma descrição em pseudocódigo do método Religação de Caminhos.

---

#### Algoritmo 4.3 Religação de Caminhos

---

**Require:** Solução inicial  $s$ , com transmissores  $\{t_1, t_2, \dots, t_n\}$ .

- 1: Escolha aleatoriamente  $s' \in S_g$ .
  - 2: Sejam  $\{t'_1, t'_2, \dots, t'_n\}$  os transmissores de  $s'$ .
  - 3: Faça  $i \leftarrow 1$
  - 4: **while**  $i \leq |V|$  **do**
  - 5:   Faça  $y(t_i) \leftarrow y(t'_i)$ .
  - 6:   Execute uma busca local em  $s$
  - 7:   Seja  $s^*$  a melhor solução até o momento
  - 8:   **if**  $f(s) < f(s^*)$  **then**
  - 9:     Faça  $s^* \leftarrow s$
  - 10:   **end if**
  - 11:   Faça  $i \leftarrow i + 1$
  - 12: **end while**
-

# Capítulo 5

---

## Resultados Computacionais

*“Non sunt multiplicanda entia praeter necessitarem.”*

William of Occam

Nesta dissertação, algoritmos heurísticos são propostos para o Problema da Alocação de Freqüências. Pela impossibilidade de garantir matematicamente a qualidade das soluções geradas por essas metaheurística, a experimentação torna-se uma parte importante do trabalho. Neste capítulo, serão apresentados resultados obtidos com o GRASP proposto.

Em primeiro lugar, será apresentado na Seção 5.1.1 o conjunto de instâncias usadas para teste dos algoritmos. Uma parte dessas instâncias serviu como um método de validação do algoritmo, como será mostrado na Seção 5.1.2. Na Seção 5.1.1 será descrito como foi desenvolvido o gerador de instâncias para o PAF. Finalmente, serão apresentados os resultados dos testes com o GRASP na Seção 5.2.

### 5.1 Instâncias

Uma das preocupações durante o desenvolvimento desta dissertação foi a criação de um conjunto padrão de instâncias que fossem representativas do problema e que ao mesmo tempo pudessem servir como referencial para os testes realizados. Nesta Seção, serão descritas as instâncias que foram usadas para realização dos testes apresentados neste capítulo.

Com o propósito de realização de testes para este trabalho, foram utilizadas várias instâncias com características distintas. Os tipos de instâncias que foram considerados para os teste da metodologia proposta foram os seguintes:

- Em primeiro lugar, foi criado um conjunto de *instâncias aleatórias*. Este conjunto foi criado usando uma série de parâmetros pertencentes a soluções reais conhecidas do PAF. As instâncias aleatórias e o método usado para sua geração serão descritos na Seção 5.1.1.
- Um segundo conjunto de instâncias, onde se conhece a priori a solução ótima, foi usado apenas com o objetivo de validação dos algoritmos. Estas instâncias, que serão chamadas *instâncias de reticulado* serão descritas na Seção 5.1.2.
- “Philadelphia” é uma instância do PAF discutida em vários artigos acerca do problema e será usada como uma das instâncias de teste. Ela será discutida na Seção 5.1.3.

### 5.1.1 Geração de instâncias aleatórias

Uma das dificuldades enfrentadas no início do trabalho com o PAF foi o pequeno número de instâncias do mundo real disponíveis para realização de testes. O conjunto de instâncias mais amplamente difundido e usado em diversos trabalhos, é o fornecido pelo projeto CALMA (Aardal *et al.*, 1995). Apesar disso, o número de instâncias é pequeno (apenas 11). Um problema ainda maior para as heurísticas, é que essas instâncias não são de telefonia celular, e portanto não possuem características importantes para o modelo adotado, como por exemplo, a propriedade das antenas de estarem reunidas em algumas estações base.

A grande maioria dos trabalhos pesquisados oferecem soluções para instâncias privadas, normalmente através de convênios com empresas (Eisenblätter, 1997; Borndörfer *et al.*, 1997; Jumard e Vovor, 1998). Deste modo, um das tarefas realizadas no decorrer da pesquisa foi a geração de instâncias que estivessem o mais próximas possível daquelas do mundo real.

Para que isso fosse possível, foram usadas informações encontradas no trabalho de Borndörfer *et al.* (Borndörfer *et al.*, 1997). Nesse artigo, as instâncias são privadas, pertencentes à empresa alemã E-Plus. Entretanto, uma série de informações relevantes sobre elas são fornecidas, como mostrado na Tabela 5.1. Baseados nestes valores, foi implementado um gerador, onde os parâmetros de saída do grafo encontram-se dentro da faixa de valores apresentados no artigo.

---

<sup>0</sup>Somente uma instância teve valor superior, igual a 57

Parâmetro	Mínimo	Máximo
Número de vértices	267	4.240
Número de arestas	20.164	529.000
Densidade do grafo (Grau mínimo dos vértices)	0	11
Grau médio dos vértices	66	249
Grau máximo dos vértices	174	561
Número de componentes conexos	1	58
Tamanho máximo das cliques no grafo	14	130
Número de arestas $(i, j)$ com $d(i, j) \neq 0$	1.053	29.524
Número de arestas $(i, j)$ com $d(i, j) = 1$	9	9.470
Número de arestas $(i, j)$ com $d(i, j) = 2$	1.046	17.934
Número de arestas $(i, j)$ com $d(i, j) = 3$	3	2.120
Número de arestas com $c^{co}(i, j) > 0$	19.111	499.476
Número de arestas com $c^{ad}(i, j) > 0$	233	103.290

Tabela 5.1: Parâmetros e valores usados no gerador de instancias para o PAF

Dos parâmetros mostrados na Tabela 5.1, apenas alguns foram considerados para a geração de instâncias. Os parâmetros usados foram: Densidade do grafo (relação entre o número total de arestas e o número máximo de arestas, igual a  $n(n-1)/2$ ) e a percentagem de arestas  $(i, j)$  com  $d(i, j) = 1$ , com  $d(i, j) = 2$  e  $d(i, j) = 3$ , respectivamente. Por último, foram também usadas informações acerca da percentagem de arestas com  $c^{co}(i, j) > 0$  e com  $c^{ad}(i, j) > 0$ .

Com esses dados, preparou-se um algoritmo capaz de gerar instâncias que tivessem características parecidas com as que foram apresentadas na Tabela 5.1. O algoritmo funciona fazendo escolhas aleatórias, mas fazendo uma verificação para que a instância não ultrapasse os limites especificados. As instâncias que foram geradas dessa forma são apresentadas na Tabela 5.1.1. Nessa tabela, as colunas representam, respectivamente: o número de estações base e o de transmissores, a densidade percentual do grafo (relação entre o número total de arestas e o número máximo de arestas) e o número de arestas com  $d(i, j) = 1$ ,  $d(i, j) = 2$ ,  $c^{co}(i, j) > 0$  e  $c^{ad}(i, j) > 0$ , respectivamente.

### 5.1.2 Geração de instâncias com ótimo conhecido

Durante o curso do trabalho, foram usadas uma série de instâncias de validação para verificar a corretude e a convergência dos algoritmos propostos. Essas instâncias

e/n	Parâmetros de Geração				
	dens(%)	$d(ij) = 1$	$d(ij) = 2$	$c^{co}(ij) > 0$	$c^{ad}(ij) > 0$
20/100	7	10	5	190	185
25/125	6	12	9	300	294
30/150	10	23	23	435	423
35/175	13	55	25	595	582
40/200	9	43	30	780	757
45/225	7	43	34	990	954
50/250	11	79	60	1225	1194
55/275	9	98	53	1485	1437
60/300	11	125	76	1770	1709
65/325	10	139	90	2080	2026
70/350	10	166	90	2415	2340
75/375	9	154	116	2775	2685
80/400	9	181	117	3160	3063
85/425	8	186	130	3570	3463
90/450	9	240	156	4005	3897
95/475	10	287	188	4465	4350
100/500	9	298	184	4950	4830
105/525	10	342	224	5460	5318
110/550	9	354	234	5995	5824
120/600	9	423	273	7140	6944

Tabela 5.2: Conjunto de instâncias do PAF usadas durante os experimentos. As colunas denominadas e, n e m representam, respectivamente, o número de estações base, o número total de antenas e o número de arestas do grafo de interferência. As outras colunas representam a densidade percentual do grafo (relação entre o número total de arestas e o número máximo de arestas) e o número de arestas com  $d(i, j) = 1$ ,  $d(i, j) = 2$ ,  $c^{co}(i, j) > 0$  e  $c^{ad}(i, j) > 0$ , respectivamente.

possuem algumas propriedades que permitem conhecer o valor ótimo para a função objetivo. Construindo instâncias dessa forma pode-se verificar se o algoritmo apresenta uma convergência mínima, capaz de chegar à solução já conhecida.

Para garantir isso, foram criados grafos de interferência para os quais já se sabe o valor ótimo da coloração. Por exemplo, no primeiro grafo da Figura 5.1, sabe-se, usando conhecimentos de teoria dos grafos, que duas cores são suficientes para colorir todos os vértices. Esta é uma propriedade comum a todos os grafos que possuem apenas ciclos de tamanho par. No grafo da Figura 5.1.a), todos os ciclos são de tamanho par, portanto essa propriedade é verdadeira. Da mesma forma, com três cores pode-se colorir todos os vértices do segundo grafo e quatro cores são suficientes para colorir o último. Os grafos de reticulados, como mostrados na Figura 5.1, serão chamados de grafos *tipo1*, *tipo2* e *tipo3*.

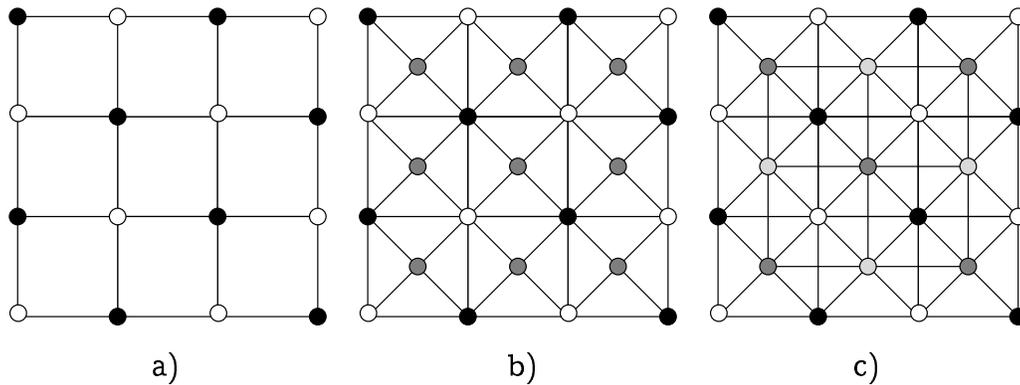


Figura 5.1: Grafos de reticulado. Esses grafos podem ser coloridos, respectivamente, com apenas duas, três, ou quatro cores.

Pode-se usar este conhecimento para criar uma solução ótima para os grafos mostrados, supondo que cada vértice corresponde a uma estação base. Como exemplo, suponha que existem 80 canais disponíveis no sistema. Pode-se gerar uma instância do PAF onde as estações base sejam grafos do tipo 1. Como precisarmos de apenas duas cores para colorir o grafo, pode-se assumir que o espectro será dividido em duas partes iguais, como mostrado na Figura 5.2. Cada uma destas partes será denominada região do espectro. Assume-se também que as interferências local, co-canal e adjacente canal só atuam ao longo das arestas do grafo de interferência. Se colocarmos um total de 5 transmissores em cada estação base, elas poderão ser livremente atribuídas com cores provenientes de uma das regiões. Uma solução com interferência zero para esse grafo simples pode

ser encontrada usando dois conjuntos de cores: um deles confinado à primeira região do espectro (1...40) e outro à segunda região (41...80). No exemplo da Figura 5.2, o conjunto de cores correspondente à cor 1 é {1,5,10,15,20} e o correspondente à cor 2 é {45,50,55,60,65}. Neste exemplo, existe grande liberdade para a escolha dos grupos de cores, desde que as restrições de interferência local sejam obedecidas. Entretanto, a medida que o espectro disponível é reduzido, o número de possibilidades é menor. O mesmo método pode ser usado para gerar instâncias com função objetivo zero para grafos para tipo 2 e 3.

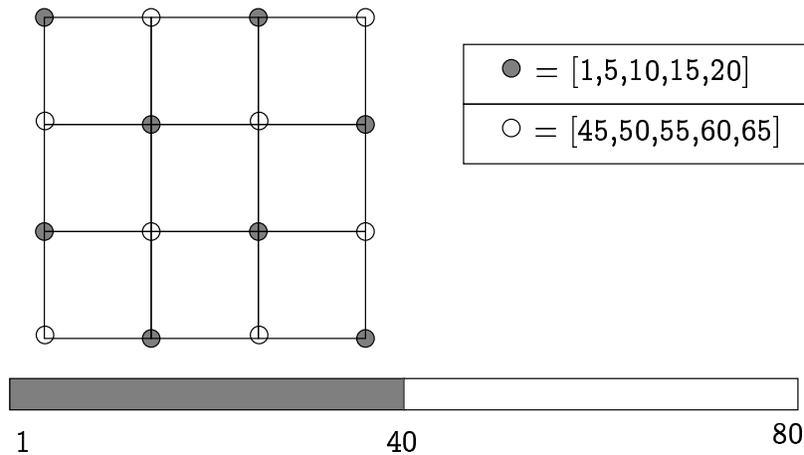


Figura 5.2: Possível solução para o primeiro grafo de reticulado da Figura 5.1.

### 5.1.3 Instância Philadelphia

A instância conhecida por “Philadelphia” refere-se a uma alocação de frequências para uma região em volta da conhecida cidade com o mesmo nome, nos Estados Unidos. Essa instância é formada apenas por 21 células hexagonais, com o formato mostrado na Figura 5.4. O vetor de demanda é:

$$m = (8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8).$$

As restrições de distância mínima entre canais  $d(i, j)$  são dadas pelas equações

$$d(i, j) = \begin{cases} 5, & \text{se } i = j \\ 2, & \text{se } 0 < \text{dist}(v_i, v_j) \leq 1 \\ 1, & \text{se } 0 < \text{dist}(v_i, v_j) \leq 2\sqrt{3} \end{cases}$$

onde  $\text{dist}(x, y)$  é a distância euclidiana entre  $x$  e  $y$ . As fórmulas resultam na matriz apresentada na Figura 5.3.

$$(d_{ij}) = \begin{pmatrix} 5 & 2 & 1 & 1 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 5 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 5 & 2 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 5 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 & 5 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ 2 & 2 & 1 & 1 & 0 & 1 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 5 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 5 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 2 & 5 & 2 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 2 & 2 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 1 & 1 & 2 & 5 & 1 & 1 & 2 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 5 & 2 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 5 & 2 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 5 \end{pmatrix}$$

Figura 5.3: Matriz de distância mínima para a instância “philadelphia”.

Esta instância já foi estudada por diversos autores (Anderson, 1973; Box, 1978; Leese, 1995; Sivarajan *et al.*, 1989; Wang e Rushford, 1996), que abordaram problemas de alocação de frequências. Portanto é de interesse usá-la como uma das instâncias de teste para os algoritmos propostos neste trabalho. Por outro lado, esta é uma das poucas instâncias reais do problema publicamente acessíveis.

## 5.2 Resultados

Nesta Seção serão apresentados os resultados dos experimentos computacionais realizados durante o desenvolvimento da dissertação. Os equipamentos usados para estes

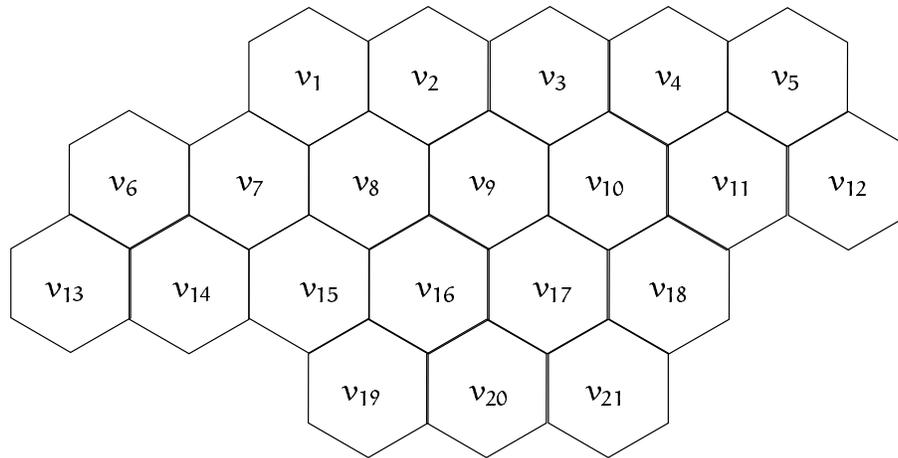


Figura 5.4: Instância “Philadelphia”.

testes foram duas estações de trabalho IBM RISC/6000 com 128MB de memória cada, com sistema operacional AIX 4.2. As heurísticas e metaheurísticas discutidas foram codificadas na linguagem C ANSI. O compilador usado foi o gcc 2.95.1 com parâmetro de otimização -O2. Todos os testes apresentados nesta seção foram executados por dez vezes, sendo cada execução realizada durante o período de uma hora. As tabelas com resultados para os experimentos aqui discutidos encontram-se reunidas no Apêndice A.

### 5.2.1 Instâncias com ótimo conhecido

Usando as instâncias que foram geradas de acordo o método descrito na Seção 5.1.2, os algoritmos propostos durante a pesquisa foram testados. O principal objetivo dos testes era verificar o tempo necessário para que se encontrasse o valor ótimo. Vale ressaltar que para cada instância de reticulado, o valor ótimo esperado é conhecido e sempre igual a zero. Isto serviu como uma forma de validar as heurísticas e verificar a velocidade de cada uma em problemas simples. Foram geradas séries de instâncias correspondentes ao conjunto padrão de testes. A primeira série constitui-se de grafos de reticulado do tipo 1, com número de estações base correspondente aos grafos do conjunto padrão. Da mesma forma, a segunda série constitui-se apenas de grafos do tipo 2 e a terceira de grafos de reticulado do tipo 3.

Os resultados destes testes encontram-se nas tabelas A.1, A.2 e A.3, apresentadas no Apêndice A. A Tabela A.1 refere-se apenas a grafos de reticulado do tipo 1. Da mesma forma as tabelas A.2 e A.3 são referentes a grafos do tipo 2 e do tipo 3, respectivamente.

Em cada uma das tabelas tem-se a média de resultados para 10 execuções dos métodos de Busca Local (BL), GRASP, GRASP Reativo (GR) e GRASP Reativo mais Religação de Caminhos (RC), além da média dos tempos.

Pode-se observar, a partir destes resultados, que o GRASP consegue encontrar as soluções ótimas em um tempo menor. Além disto, a heurística de busca local não consegue encontrar a solução ótima para instâncias maiores do problema (a partir de 400 transmissores), por ficar presa em ótimos locais. Já o GRASP com Religação de Caminhos conseguiu as mesmas soluções em um tempo menor, o que mostra que houve aumento de eficiência graças a esta estratégia. Também foram realizados testes com estratégia de GRASP reativo. Apesar dos resultados serem idênticos ao do GRASP, houve uma degradação do tempo de execução, já que este método precisa de um tempo adicional para calibragem interna do parâmetro  $\alpha$ . Os ganhos com GRASP reativo só são bons para instâncias que exigem maior tempo de execução, o que não é o caso destas.

### 5.2.2 Instâncias aleatórias e philadelphia

O resultados encontrados para as instâncias aleatórias e a instância philadelphia são resumidos nas Tabelas A.4 a A.7. O número de canais usados durante os testes foi de 80 canais, que é um valor típico. As colunas dessa tabela representam, respectivamente, o número total de transmissores (antenas), o valor médio de função objetivo encontrado nas 10 execuções, o tempo (em segundos) em que o melhor valor foi encontrado, o número de iterações necessárias para encontrar o melhor valor e o valor de  $\alpha$  necessário para encontrar o melhor valor (quando aplicável). As primeiras vinte entradas de cada tabela referem-se às instâncias aleatórias enquanto a última entrada corresponde à instância philadelphia.

Cada entrada para uma instância específica é dividida em cinco linhas, uma para cada um dos métodos testados. Em primeiro lugar, na entrada GRASP, tem-se os valores encontrados pelo GRASP descrito na Seção 4.2. Em seguida, vê-se na entrada T-Col resultados computacionais encontrados usando apenas a heurística gulosa de T-Coloração. A terceira linha refere-se aos resultados produzidos somente pela Busca Local (BL). A próxima entrada apresenta valores encontrados pelo método GRASP Reativo. Finalmente, a quinta linha mostra resultados obtidos pela combinação de GRASP Reativo e Religação de Caminhos.

Os testes com a heurística gulosa T-Coloração foram realizados por dois motivos

principais:

1. Uma versão modificada desta heurística já faz do parte do GRASP, sendo usada pelo construtor para gerar a lista restrita de candidatos
2. A heurística já foi usada em (Borndörfer *et al.*, 1997) e demonstrou ser uma das melhores heurísticas para o PAF. Com este teste, o objetivo foi verificar o quanto a implementação do GRASP poderia ser melhor que a heurística sozinha.

Percebe-se que os resultados foram amplamente favoráveis ao GRASP, o que aliás era de se esperar, visto que a heurística T-Coloração participa apenas da primeira fase do GRASP. Além disso, percebe-se que esta heurística produz resultados muito inferiores aos dos outros métodos apresentados. Provavelmente isto deve-se ao fato de que ela não está adaptada à estrutura do problema como é o caso das outras heurísticas. Convém lembrar que na função objetiva usada o número de restrições violadas possui um peso muito maior que o valor da interferência. Portanto, os valores maiores mostrados pelo método da T-Coloração indicam que esse método não foi tão eficiente na eliminação de violações.

Com relação ao algoritmo de busca local, vê-se que, para as instâncias pequenas do problema, os resultados retornados pelo método são comparáveis ao produzido pelo GRASP. Entretanto, à medida que o número de transmissores aumenta, também aumenta a diferença entre os resultados dos dois métodos. Sendo um algoritmo guloso, esta heurística tem uma dificuldade maior de escapar de mínimos locais, o que faz com que ela não tenha bom desempenho.

De acordo com os valores encontrados pelo método GRASP Reativo, percebe-se que houve melhorias com relação aos resultados dos GRASP isolado. Além disso, pode-se ver na última coluna os valores do parâmetro  $\alpha$  usado no momento em que a melhor solução foi encontrada. Percebe-se que, por ter maior liberdade para definir o valor deste parâmetro, o método GRASP Reativo é capaz de fazer uma melhor exploração do espaço de soluções do problema. Além disso, o GR fornece uma mais automatizada de determinação do parâmetro  $\alpha$ . Vê-se que o valor 0.2, entre aqueles testados (que vão de 0.1 a 1.0, com incrementos de 0.1), é o que fornece os melhores resultados na maioria das instâncias.

A adição do método Religação de Caminhos foi feita com o objetivo de melhorar a qualidade das soluções geradas pela metodologia. Os resultados encontrados com a aplicação do GRASP Reativo e Religação de Caminhos mostram que o método produziu

uma melhoria em 14 das 20 instâncias aleatórias testadas. Isso traduz a contribuição trazida pelo passo adicional realizado pelo Religação de Caminhos. Apesar disso, notou-se também um aumento perceptível no tempo em que a melhor solução foi encontrada. Isso se deve provavelmente ao esforço adicional necessário para a execução do método.

Na Figura 5.5 vê-se um gráfico onde o comportamento dinâmico do GRASP mais RC e GR é apresentado. Este gráfico relaciona o tempo de execução e o valor de função objetivo para as instâncias aleatórias com 50, 60, 70, 80, 90, 100, 110 e 120 transmissores. Percebe-se que a convergência principal do algoritmo ocorre rapidamente, em torno dos 30 segundos, mesmo para instâncias maiores do problema. A partir daí as melhorias vão ocorrendo de forma mais lenta. Entretanto nota-se que as melhores soluções, como apresentado nas tabelas de A.4 a A.7, são encontradas em após um tempo relativamente longo em comparação ao tempo de convergência.

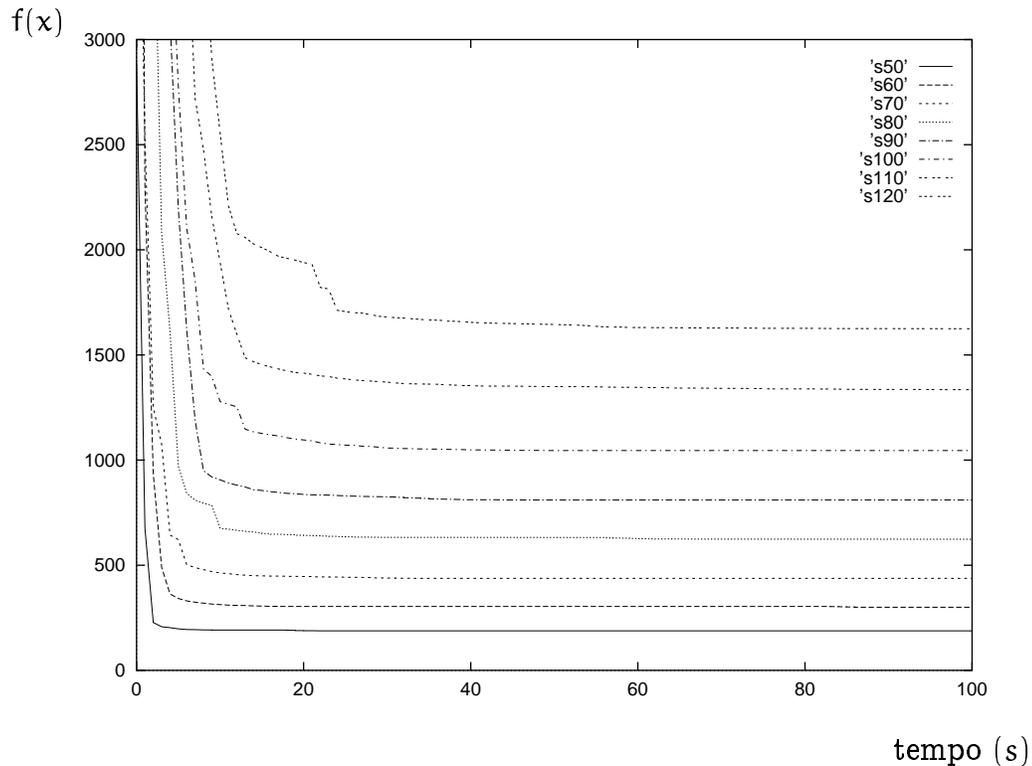


Figura 5.5: Gráfico da variação da função objetivo *versus* tempo para GRASP Reativo mais Religação de Caminhos para instâncias com 50, 60, 70, 80, 90, 100, 110 e 120 transmissores.

Na Figura 5.6 pode-se ver a evolução do valor de função objetivo com relação ao

tempo para diversos métodos. Esses resultados foram obtidos para a instância aleatória com 600 transmissores. Observa-se que a busca local é o método que apresenta convergência mais lenta, além de encontrar soluções com qualidade bem inferior, em comparação com os outros métodos. Os métodos GRASP Reativo e Religação de Caminhos também são reportados, apresentando melhorias gradativas. Os valores encontrados pela heurística T-Coloração não foram mostrados por terem sido muito piores que os dos outros métodos, prejudicando a visualização do gráfico.

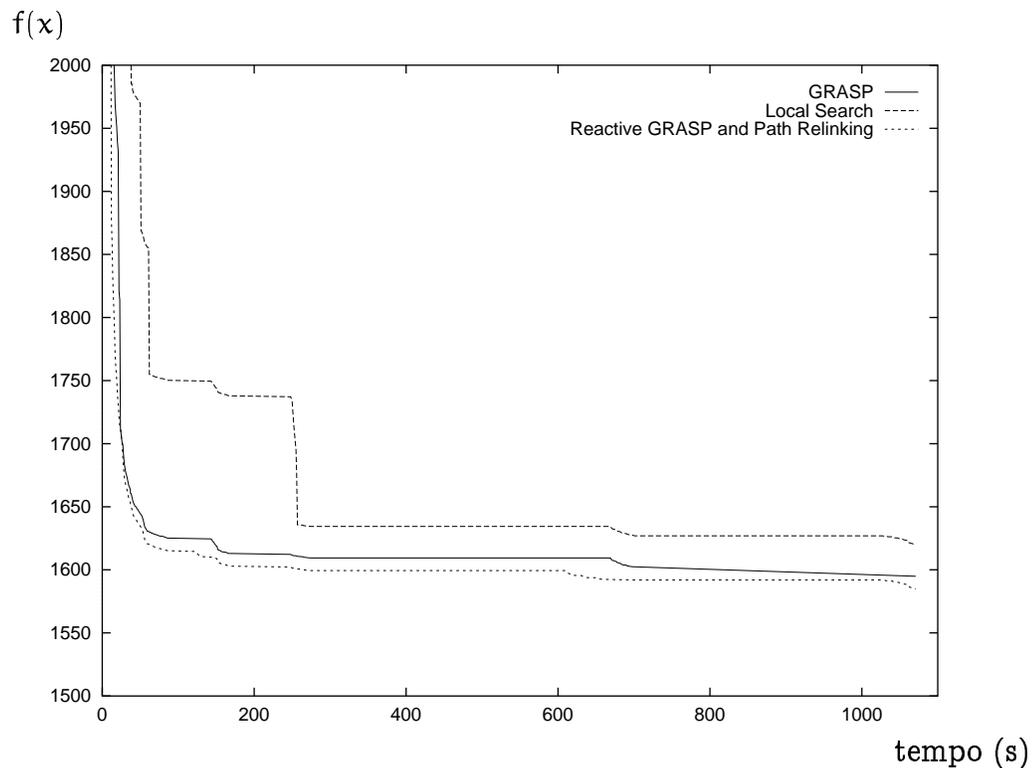


Figura 5.6: Comparação de valores de função objetivo para uma instância com 600 transmissores, usando diferentes métodos, *versus* tempo de execução.

# Capítulo 6

---

## Conclusão

*“O que não se pode falar, deve-se calar.”*

L. Wittgenstein

Neste trabalho foi apresentada uma solução metaheurística para a minimização da interferência total no problema da alocação de frequências. Este é um problema prático, que ocorre freqüentemente em sistemas de telefonia celular. A solução proposta foi baseada num GRASP (Greedy Randomized Adaptive Search Procedure), com várias modificações realizadas com o objetivo de obter soluções com melhor qualidade.

Pode-se citar algumas contribuições à teoria e à prática do problema de alocação de frequências introduzidas nesta dissertação:

1. Foi desenvolvido um GRASP para o problema, sendo ele capaz de produzir soluções de alta qualidade em comparação a outras heurísticas apresentadas na literatura.
2. Forneceu-se uma heurística de busca local simples que pode ser usada isoladamente e com sucesso para alocações de frequência que sejam necessárias com menor custo computacional, mas com boa qualidade. Isso é freqüentemente necessário com algoritmos de alocação dinâmica.
3. Foram usadas algumas técnicas adicionais, a saber, GRASP Reativo e Religação de Caminhos. Estas técnicas foram recentemente desenvolvidas para o GRASP e se mostraram capazes de melhorar a qualidade das soluções para o problema.
4. Foi realizada uma comparação entre os algoritmos propostos e alguns daqueles apresentados na literatura.

5. Finalmente, fez-se na dissertação um apanhado geral de heurísticas e metaheurísticas aplicadas ao problema, o que pode ser de grande interesse para futuros pesquisadores do assunto.

Apesar dos muitos avanços realizados durante a execução deste trabalho, podem ser apontados vários tópicos que continuam em aberto e que portanto, constituem-se em direções para trabalhos futuros:

1. *Mais instâncias do mundo real.* Um dos problemas encontrados no trabalho foi a falta de instâncias reais do problema. Um tópico para pesquisa futura seria a aplicação e avaliação do método apresentado, usando dados reais, provavelmente através de uma cooperação com empresas do setor.
2. *Alocação dinâmica.* Neste trabalho, restringiu-se à aplicação de metaheurísticas em problemas de alocação estática. Entretanto, o GRASP possui um potencial inato para o paralelismo, devido à sua estrutura de iterações independentes. Como trabalho futuro, pode-se aproveitar este potencial e aproveitá-lo também para a tarefa de alocação dinâmica.

# Apêndice A

---

## Listas de resultados para o PAF

### A.1 Introdução

Neste apêndice são apresentadas uma série de tabelas com resultados para o PAF. Estas tabelas foram criadas a partir dos experimentos discutidos no capítulo 5. Para maiores informações sobre os algoritmos usados para geração de soluções deve-se reportar ao capítulo 3.

## A.2 Grafos de reticulado

Os resultados para grafos de reticulado são apresentados nas próximas três tabelas. O valor ótimo da interferência para estes grafos é conhecido, e igual a zero. As colunas BL, GRASP, RC e GR referem-se, respectivamente, às soluções encontradas pelos métodos de Busca Local, GRASP, Path Relink e GRASP Reativo.  $f$  representa valor médio de função objetivo e  $tmp$  o tempo médio, em segundos.

Instâncias		BL		GRASP		RC		GR	
EBs	antenas	f	tmp	f	tmp	f	tmp	f	tmp
20	100	0	18	0	6	0	6	0	10
30	150	0	49	0	33	0	20	0	31
40	200	0	59	0	49	0	29	0	42
50	250	0	94	0	68	0	79	0	84
60	300	0	138	0	116	0	100	0	128
70	350	0	223	0	146	0	92	0	195
80	400	3	237	0	234	0	38	0	254
90	450	0	306	0	303	0	232	0	312
100	500	5	458	0	285	0	231	0	327
110	550	7	557	0	451	0	288	0	508
120	600	7	592	0	465	0	293	0	561
200	1000	14	713	0	517	0	326	0	682

Tabela A.1: Resultados para grafos de reticulado do tipo 1.

Instâncias		BL		GRASP		RC		GR	
EBs	antenas	f	tmp	f	tmp	f	tmp	f	tmp
23	115	0	92	0	38	0	26	0	41
32	160	0	103	0	57	0	32	0	56
41	205	0	112	0	71	0	62	0	75
50	250	0	159	0	133	0	116	0	139
61	305	6	241	0	157	0	118	0	183
72	360	8	265	0	212	0	105	0	235
83	415	7	327	0	289	0	246	0	302
98	490	12	508	0	315	0	287	0	398
113	565	9	915	0	522	0	416	0	585
128	640	15	875	0	579	0	442	0	721
145	725	19	894	0	603	0	395	0	812
200	1000	24	1025	0	582	0	451	0	877

Tabela A.2: Resultados para grafos de reticulado do tipo 2.

Instâncias		BL		GRASP		RC		GR	
EBs	antenas	f	tmp	f	tmp	f	tmp	f	tmp
23	115	0	104	0	56	0	38	0	72
32	160	0	117	0	79	0	51	0	80
41	205	0	115	0	86	0	64	0	91
50	250	0	178	0	143	0	118	0	157
61	305	0	249	0	213	0	171	0	210
72	360	3	281	0	223	0	207	0	231
83	415	17	342	0	308	0	291	0	322
98	490	23	628	0	519	0	486	0	594
113	565	19	1193	0	817	0	762	0	824
128	640	22	973	0	848	0	765	0	890
145	725	36	1374	0	972	0	849	0	1178
200	1000	49	1562	0	1218	0	826	0	1239

Tabela A.3: Resultados para grafos de reticulado do tipo 3.

## A.3 GRASP

As próximas tabelas referem-se aos experimentos realizados com as instâncias aleatórias e a instância philadelphia. As colunas representam, respectivamente, o número de antenas, o método de solução usado (GRASP, T-Coloração (T-Col), Busca Local (BL), GRASP Reativo (GR) e Religação de Caminhos (RC)), a média dos valores de função objetivo em todas as execuções, o valor da melhor função objetivo encontrada, o tempo (em segundos) em que a melhor solução foi encontrada, o número de iterações em que a melhor solução foi encontrada (aplicável apenas a GRASP, GR e RC) e finalmente o valor do parâmetro  $\alpha$  no momento em que a melhor solução foi encontrada (aplicável somente a GRASP Reativo).

inst.	método	média $f(x)$	melhor $f(x)$	tempo(s)	n. iter	$\alpha$
100	GRASP	12	12	2071	982	
	T-Col	368	12	1		
	BL	12	12	2845		
	GR	12	12	2757	814	0.1
	RC	12	12	3183	603	
125	GRASP	22	22	1210	900	
	T-Col	1806	22	1		
	BL	24	22	3173		
	GR	24	22	2353	763	0.1
	RC	22	22	2924	581	
150	GRASP	44	43	3025	837	
	T-Col	2538	44	2		
	BL	44	44	3264		
	GR	45	43	3253	812	0.2
	RC	44	43	3117	529	
175	GRASP	65	65	3546	661	
	T-Col	3282	68	2		
	BL	68	68	2061		
	GR	65	65	3418	724	0.1
	RC	65	63	3418	422	
200	GRASP	100	98	1120	296	
	T-Col	6281	101	3		
	BL	103	101	829		
	GR	99	98	2361	416	0.2
	RC	98	97	3345	391	
225	GRASP	139	139	1149	123	
	T-Col	9803	141	3		
	BL	141	141	2523		
	GR	139	137	2428	385	0.2
	RC	138	137	2853	296	

Tabela A.4: Comparação de resultados computacionais para instancias do PAF.

inst.	método	média $f(x)$	melhor $f(x)$	tempo(s)	n. iter	$\alpha$
250	GRASP	183	183	1333	111	
	T-Col	11305	185	4		
	BL	188	185	1075		
	GR	183	182	1626	103	0.2
	RC	183	181	2583	188	
275	GRASP	228	228	2361	154	
	T-Col	12870	235	5		
	BL	239	235	1947		
	GR	228	226	2253	173	0.2
	RC	227	224	3073	191	
300	GRASP	292	292	253	117	
	T-Col	13760	295	8		
	BL	298	295	1856		
	GR	291	289	2745	182	0.2
	RC	292	288	2915	197	
325	GRASP	345	345	3232	140	
	T-Col	15877	357	11		
	BL	359	357	2584		
	GR	344	342	2851	152	0.2
	RC	344	339	2934	161	
350	GRASP	422	422	3390	131	
	T-Col	16609	442	12		
	BL	446	442	1838		
	GR	422	419	3292	149	0.2
	RC	421	418	3063	131	
375	GRASP	493	493	1928	62	
	T-Col	18801	502	16		
	BL	507	502	2169		
	GR	493	492	2395	107	0.2
	RC	492	490	2174	104	

Tabela A.5: Comparação de resultados computacionais para instancias do PAF (continuação).

inst.	método	média $f(x)$	melhor $f(x)$	tempo(s)	n. iter	$\alpha$
400	GRASP	606	606	629	17	
	T-Col	23797	613	19		
	BL	613	613	1523		
	GR	605	601	1536	62	0.2
	RC	603	598	2976	96	
425	GRASP	686	686	1377	29	
	T-Col	25256	698	23		
	BL	699	698	2575		
	GR	685	682	2421	78	0.2
	RC	684	679	2613	77	
450	GRASP	781	778	1752	43	
	T-Col	18842	793	31		
	BL	795	793	2812		
	GR	780	775	1838	54	0.2
	RC	778	773	2723	71	
475	GRASP	903	902	2085	39	
	T-Col	29487	913	39		
	BL	915	913	2163		
	GR	903	899	1953	48	0.2
	RC	903	899	2530	53	
500	GRASP	1028	1025	3238	41	
	T-Col	32322	2303	46		
	BL	1036	1035	2956		
	GR	1026	1020	2597	51	0.2
	RC	1025	1017	2317	49	
525	GRASP	1169	1162	2851	38	
	T-Col	49320	1177	58		
	BL	1177	1177	1825		
	GR	1167	1158	2263	44	0.3
	RC	1165	1156	2941	47	

Tabela A.6: Comparação de resultados computacionais para instancias do PAF (continuação).

inst.	método	média $f(x)$	melhor $f(x)$	tempo(s)	n. iter	$\alpha$
550	GRASP	1308	1298	3129	36	
	T-Col	61302	1319	69		
	BL	1321	1319	2531		
	GR	1305	1301	3015	43	0.2
	RC	1303	1297	3384	34	
600	GRASP	1594	1589	2071	19	
	T-Col	11651	1606	81		
	BL	1612	1606	3287		
	GR	1593	1584	2834	35	0.2
	RC	1594	1580	3186	28	
phil.	GRASP	27	26	1852	513	
	T-Col	2184	2173	81		
	BL	28	27	1627		
	GR	26	26	1427	612	0.2
	RC	26	25	2431	416	

Tabela A.7: Comparação de resultados computacionais para instancias do PAF (continuação).

---

# Referências Bibliográficas

- Aardal, K., Hurkens, C. A. J., Lenstra, J. K., Tiourine, S. R. An overview of algorithmic approaches to frequency assignment problems, CALMA project. Relatório técnico, Dep. of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, Holanda, 1995.
- Aardal, K. I., Hipolito, A., Hoesel, C. P. M. Van, Jansen, B., Roos, C., Terlaky, T. A branch-and-cut algorithm for the frequency assignment problem. Relatório técnico, Eindhoven University, 1998.
- Abramson, D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Man. Sci.*, 37:98–113, 1991.
- Adjakpl'e, P. M., Jumard, B. Greedy and tabu search heuristics for channel block assignment in cellular systems. *Les Cahiers du GERAD*, 1997.
- Anderson, L. G. A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system. *IEEE transactions communications*, COM-21:1294–1301, 1973.
- Borndörfer, R., Eisenblätter, A., Grötschel, M., Martin, A. Frequency assignment in cellular phone networks. Relatório técnico, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1997.
- Borndörfer, R., Eisenblätter, A., Grötschel, M., Martin, A. The orientation model for frequency assignment problems. Relatório técnico, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1999.
- Bouju, A., Boyce, J., Dimitropoulos, C., vom Scheidt, G., Taylor, J. Tabu search for the radio links frequency assignment problem. Em *Applied Decision Technologies*. UNICOM, London, UK, 1995.

- Box, F. A heuristic technique for assigning frequencies to mobile radio nets. *IEEE Transactions on Vehicular Technology*, 27(2):57–64, 1978.
- Brélaz, D. New methods to color the vertices of a graph. *Communications of ACM*, 22(4):251–256, 1979.
- Campelo, R. E., Maculan, N. *Algoritmos e heurísticas: desenvolvimento e análise de performance*. Eduff/Furnas centrais elétricas, 1994.
- Cao, G., Singhal, M. Efficient distributed channel allocation for mobile cellular networks. 1997. Dep. of Computer and Information Science, The Ohio State University.
- Capone, A., Trubian, M. Channel assignment problem in cellular systems: A new model and a tabu search algorithm. *IEEE Transactions on Vehicular Technology*, 48(4):1252–1260, 1999.
- Carnevali, P., Coletti, L., Patarnello, S. Image processing by simulated annealing. *IBM Journal of Research and Development*, 29:569–579, 1985.
- Christofides, S. Vehicle routing. Em E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, editores, *Traveling salesman problem*. John Wiley, 1985.
- Collins, N. E., Eglese, R. W., Golden, B. L. Simulated annealing – an annotated bibliography. *AJMMS*, 8:209–307, 1988.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. *Introduction to algorithms*. MIT press, 1990.
- Cozzens, M. B., Roberts, F. S. T-colorings of graphs and the channel assignment problem. *Congressum Numerantium*, 35:191–208, 1982.
- Cozzens, M. B., Roberts, F. S. Greedy algorithms for T-colorings of complete graphs and the meaningfulness of conclusions about them. *Journal of combinatorics, information and system sciences*, 16(4):286–299, 1991.
- Crompton, W., Hurley, S., Stephens, N. M. A parallel genetic algorithm for frequency assignment problems. Em *Proceedings Int. Symp. on Signal Processing, Robotics and Neural Networks*, páginas 81–84. IMACS/IEEE, Lille, France, 1994.

- Dantzig, G. B., Fulkerson, D. R., Johnson, S. M. solution of a large scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.
- Delmaire, H., Díaz, J. A., Fernández, E., Ortega, M. Comparing new heuristics for the pure integer capacitated plant location problem. Relatório técnico DR97/10, Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Spain, 1997.
- Dunkin, N., Bate, J., J., P., Cohen, D. Towards high order constraint representations for the frequency assignment problem. Relatório técnico, Dep. of Computer Science Royal Holloway College, University of London, 1998.
- Eden, H., Fastert, H. W., Kaltbeitzler, K. H. More recent methods of television network planning and the results obtained. *E.B.U. Rev*, 60-A:54–59, 1960.
- Eisenblätter, A. A frequency assignment problem in cellular phone networks (extended abstract). Em *Network Design, Connectivity and Facility Location*, volume 35 de *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.
- Fahham, F. J. Cellular communication networks - GSM. Surveys and Presentations in Information Systems Engineering, Department of Electrical and Electronic Engineering, Imperial College of Science Technology and Medicine, UK, 1996. URL [http://www-dse.doc.ic.ac.uk/~nd/surprise\\_96](http://www-dse.doc.ic.ac.uk/~nd/surprise_96).
- Fastert, H. W. The mathematical theory underlying the planning of transmitter networks. *E.B.U. Rev*, 60-A:60–69, 1960.
- Feo, T. A., Resende, M. G. C. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.
- Feo, T. A., Resende, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995. URL <http://www.research.att.com/~mgcr/doc/gtut.ps.Z>.
- Feo, T. A., Resende, M. G. C., Smith, S. H. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994. URL <http://www.research.att.com/~mgcr/doc/gmis.ps.Z>.

- Feo, T. A., Venkatraman, K., Bard, J. F. A GRASP for a difficult single machine scheduling problem. *Computers & Operations Research*, 18:635–643, 1991.
- Fluerent, C., Glover, F. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. Relatório técnico, University of Colorado, 1997. Citado em (Laguna e Martí, 1998).
- Garey, M. R., Johnson, D. S. *Computers and Intractability – a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- Glover, F. Tabu search - part I. *ORSA journal of Computing*, 1:190–206, 1989a.
- Glover, F. Tabu search - part II. *ORSA journal of Computing*, 2:4–32, 1989b.
- Glover, F. A template for scatter search and path relinking. *Artificial Evolution, Lecture Notes in Computer Science*, (1363):13–54, 1998.
- Glover, F., Laguna, M. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.
- Goldstein, L., Waterman, M. S. Mapping dna by stochastic relaxation. *Advances in Applied Mathematics*, 8:194–207, 1987.
- Graham, R. L., Knuth, D. E., Patashnik, O. *Concrete Mathematics*. Addison Wesley, 1989.
- Hale, W. K. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1513, 1980.
- Hjorring, C. A. *The vehicle routing problem and local search metaheuristics*. Tese de doutorado, University of Auckland, Auckland, New Zealand, 1995.
- Holland, H. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, 1975.
- Jansky, D. M. Spectrum management techniques. Don White Consultants, 1977. Citado em (Hale, 1980).
- Johnson, D. S., Aragon, C. R., McGeoch, L. A., Schevon, C. Optimization by simulated annealing: and experimental evaluation; part I, graph partitioning. *Operations Research*, 1992a.

- Johnson, D. S., Aragon, C. R., McGeoch, L. A., Schevon, C. Optimization by simulated annealing: and experimental evaluation; part II, graph coloring and number partitioning. *Operations Research*, 1992b.
- Johnson, D. S., Aragon, C. R., McGeoch, L. A., Schevon, C. Optimization by simulated annealing: and experimental evaluation; part III, the travelling salesman problem. *Operations Research*, 1992c.
- Jonhson, D. S., Aragon, C. A., Mcgeoch, L. A., Schevon, C. Optimization by simulated annealing: an experimental evaluation – Part II (graph coloring and number partition). *Operations research*, 31:378–406, 1991.
- JTAC. Spectrum engineering – The key to progress. New York:IEEE, 1968. Citado em (Hale, 1980).
- Jumard, B., Vovor, T. A column generation approach for the exact solution of channel assignment problems. *Les Cahiers du GERAD*, 1998.
- Jünger, M., Martin, A., Reinelt, G., Weismantel, R. Quadratic 0/1 optimization and a decomposition approach for the placement of eletronic circuits. *Mathematical Programming*, 63:257–279, 1994.
- Kapsalis, A., Rayward-Smith, V. J., Smith, G. D. Using genetic algorithms to solve the radio link frequency assignment problem. Em D.W. Pearson, N.C. Steele, R.F. Albrecht, editores, *Proceedings 2nd Int. Conf. on Artificial Neural Networks and Genetic Algorithms*, páginas 37–40. Springer-Verlag, Ales, France, 1995.
- Katzela, I., Naghshineh, M. Channel assignment schemes for cellular mobile telecommunication systems. *IEEE Personal Communications Magazine*, 1996.
- Kirkpatrick, S., Jr., C. D. Gellat, P.Vicchi, M. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- Laguna, M., Martí, R. A GRASP for coloring sparse graphs. Relatório técnico, Graduate School of Business, University of Colorado, Boulder, CO 80309-0419 USA, 1998.
- Laguna, M., Martí, R. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11:44–52, 1999.

- Laporte, G. The traveling salesman problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, (59), 1992.
- Lee, C. Y. *Mobile Cellular Telecommunications Systems*. McGraw-Hill, 1989.
- Leese, R. A. Tiling methods for channel assignment in radio communication networks. Em *3rd int. congress on industrial and applied mathematics*. Hamburg, Germany, 1995.
- Leighton, F. T. A graph coloring algorithm for large scheduling problems. *J. Res. Nat. Bur. Standard*, 84(6):489–506, 1979.
- Liu, D. D. T-colorings of graphs. *Discrete mathematics*, 101:203–212, 1992.
- Liu, X., Pardalos, P. M., Rajasekaran, S., Resende, M. G. C. A GRASP for frequency assignment in mobile radio networks. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 52:195–201, 2000.
- Mateus, G. R., Loureiro, A. F. *Introdução à Computação Móvel*. 11<sup>a</sup> Escola de Computação, Rio de Janeiro, 1998.
- Matula, D. W., Marble, G., Isaacson, J. D. *Graph theory and computing*, capítulo Graph Coloring Algorithms, páginas 104–122. R.C. Read, ed. Academic press, New York, 1972.
- McDiarmid, C., Reed, B. Channel assignment and weighted colouring, 1997. Manuscrito.
- Meyer, K. K., Nachtsheim, C. J. Constructing exact d-optimal experimental designs by simulated annealing. *AJMMS*, 8:329–359, 1988.
- Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts. Relatório técnico, California Institute of Technology, 1989.
- Ottosson, G., Carlsson, M. Using global constraints for frequency allocation. Relatório técnico TR-97-07, ASTEC, 1997. Citado em (Dunkin *et al.*, 1998).
- Papadimitriou, C. H., Steiglitz, K. *Combinatorial Optimization: algorithms and complexity*. Dover Publications, Inc., New York, 1982.

- Peemöller, J. A correction to Brelaz's modification of Brown's coloring algorithm. *Communications of ACM*, 26(8):595–599, 1983.
- Periannan, R. Existing technology of cellular networks – mobile phones. Surveys and Presentations in Information Systems Engineering, Department of Electrical and Electronic Engineering, Imperial College of Science Technology and Medicine, UK, 1996. URL [\url{http://www-dse.doc.ic.ac.uk/~nd/surprise\\_96}](http://www-dse.doc.ic.ac.uk/~nd/surprise_96).
- Polya, G. *How to Solve It; A New Aspect of Mathematical Method*. Princeton Science Library. Princeton Univ Pr., 1988.
- Porto, S., Kitajima, J. P., Ribeiro, C. C. Performance evaluation of a parallel tabu search task scheduling algorithm. *Parallel Computing*, 1999. Special Issue on High-Performance Computing for Operational Research.
- Prais, M., Ribeiro, C. C. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing*, 12(3):164–176, 2000.
- Rechenberg, I. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.
- Roberts, F. S. T-colorings and graphs – recent results and open problems. *Discrete mathematics*, 93:229–245, 1991.
- Sandalidis, H. G., Stavroulakis, P. P., Rodriguez-Tellez, J. Borrowing channel assignment strategies based on heuristic techniques for cellular systems. *IEEE transactions on neural networks*, 10(1):176–181, 1999.
- Schiex, T., Fargier, H., Verfaillie, G. Valuated constraint satisfaction problems; hard and easy problems. Em *14th International Joint Conference on Artificial Intelligence*. 1995.
- Schwefel, H. P. *Numerical Optimization of Computer Models*. Wiley, Chichester, 1981.
- Singh, S., Bertsekas, D. Reinforcement learning for dynamic channel allocation in cellular telephone systems. Em *NIPS96*. 1996.

- Sivarajan, K. N., McElice, R. J., W.Ketchum, J. Channel assignment in cellular radio. Em *Proc. 39th conference IEEE Veh. Technol. Soc.*, páginas 846–850. 1989.
- van Laarhoven, P. J. M, Aarts, E. H. L. *Simulated Annealing: theory and applications*. D. Reidel Publishing Co., Holanda, 1987.
- vom Scheid, G. *Extension of GENET to n-ary Partial Constraint Satisfaction Problems and Constraint Satisfaction Optimisation Problems*. Tese de mestrado, King's College London, 1997.
- Wang, W., Rushford, C. K. An adaptive local-search algorithm for the channel assignment problem (CAP). *IEEE Transactions on Vehicular Technology*, 45:459–466, 1996.
- Welsh, D. J. A, Powell, M. B. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Comput. Journal*, 10:85–86, 1967.