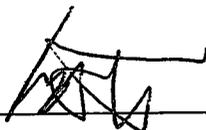


O MÉTODO DO RESFRIAMENTO SIMULADO NO PLANEJAMENTO DE
ROTAS AÉREAS

Ricardo Fernandes Ribeiro

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



Prof. Luís Alfredo Vidal de Carvalho, D.Sc.



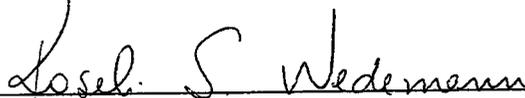
Prof. Geraldo Bonorino Xexéo, D.Sc.



Prof. Valmir Carneiro Barbosa, Ph.D.



Profª. Lúcia Maria de Assumpção Drummond, D.Sc.



Profª. Roseli Suzi Wedemann, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2008

RIBEIRO, RICARDO FERNANDES

O Método do Resfriamento Simulado no
Planejamento de Rotas Aéreas [Rio de Janeiro]
2008

IX, 55 p. 29,7cm (COPPE/UFRJ, D. Sc.,
Engenharia de Sistemas e Computação, 2008)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1. Planejamento de Rotas Aéreas
2. Resfriamento Simulado
3. Otimização Multi-objetivo

I. COPPE/UFRJ

II. Título (série)

Agradecimentos

Ao professor Luís Alfredo Vidal de Carvalho por todo o apoio dado durante este período.

Aos professores Valmir Barbosa, Geraldo Xexéo, Lúcia Drummond e Roseli Wedemann pelas contribuições fornecidas a este trabalho, assim como pela identificação de novas possibilidades associadas ao mesmo.

Aos funcionários do PESC, em particular da Secretaria e do Suporte, pela solicitude com que sempre fora atendido quando necessário.

Ao CNPq, pelo suporte financeiro proporcionado durante este período de doutorado.

Aos colegas, pela amizade, especialmente nos momentos mais difíceis.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D. Sc.)

O MÉTODO DO RESFRIAMENTO SIMULADO NO PLANEJAMENTO DE ROTAS AÉREAS

Ricardo Fernandes Ribeiro

Abril/2008

Orientador: Luís Alfredo Vidal de Carvalho

Programa: Engenharia de Sistemas e Computação

Este trabalho é voltado para o problema de particionamento de rotas aéreas em níveis de vôo, de modo a não haver cruzamentos dentro de cada nível de vôo, e minimizando o consumo de combustível e as trocas de nível dentro de cada rota. Para se obter isso, uma versão modificada do Resfriamento Simulado é utilizada. Nesta versão, as variações de todas as funções a serem otimizadas são combinadas para decidir pela aceitação de novas configurações.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D. Sc.)

THE SIMULATED ANNEALING METHOD IN AIRWAYS PLANNING

Ricardo Fernandes Ribeiro

April/2008

Advisor: Luís Alfredo Vidal de Carvalho

Department: Systems Engineering and Computer Science

This work tackles the problem of partitioning a set of routes into flight levels in such way that there will be no crossings inside each level, and minimizing fuel consumption and level changes inside each route. In order to achieve this, a modified version of Simulated Annealing is used. In this version, the variations of all functions to be optimized are combined to decide for the acceptance of new configurations.

Sumário

1	Introdução	1
2	Definições sobre Grafos	3
2.1	Conceitos	3
2.2	Problemas	4
2.2.1	<i>Maximum k-cut problem</i>	5
2.2.2	Coloração (mínima) de vértices	5
3	Resfriamento Simulado	7
3.1	Histórico	7
3.1.1	O Método de Monte Carlo	7
3.1.2	O Algoritmo de Metropolis	8
3.2	Funcionamento	10
3.2.1	O Algoritmo	11
3.3	Resfriamento Simulado Multi-objetivo	14
3.3.1	Otimização combinatória multi-objetivo	15
3.3.2	Métodos existentes	15
3.3.3	Método utilizado	16
4	Particionamento de Rotas Aéreas	18
4.1	Definição do problema	18
4.1.1	Nomenclatura	18
4.1.2	Níveis de vôo	19

4.2	Variantes básicas do problema	21
4.2.1	Primeiro Problema	21
4.2.2	Segundo Problema	25
4.2.3	Terceiro Problema	28
4.3	Estendendo o problema	35
4.3.1	Funções objetivo	35
4.3.2	Exemplo	37
5	Resultados	40
5.1	Descrição dos grafos do DIMACS	40
5.2	Resultados do Problema 2	41
6	Conclusão	43
	Referências Bibliográficas	44
A	Algoritmos Auxiliares	50
A.1	Algoritmos para o Problema 2	50
A.1.1	Particionamento dos vértices de um nível	50
A.1.2	Busca do menor nível que elimine cruzamentos	52
A.2	Algoritmos para o Problema 3	53
A.2.1	Criação do grafo de rotas pelo grafo de cruzamentos	53
A.2.2	Montagem do grafo com novos pares de cruzamentos	54

Lista de Figuras

2.1	Exemplo de grafo	4
2.2	Exemplo de coloração de grafo	6
3.1	Cálculo da área da figura usando o Método de Monte Carlo	9
3.2	Perfil de uma função objetivo com mínimo local	13
3.3	Equilíbrio Térmico e Cristalização no Resfriamento Simulado	14
4.1	Pontos cardeais em uma bússola	19
4.2	Níveis de vôo	20
4.3	Exemplo de minimização de cruzamentos	21
4.4	Grafo de interseção do conjunto de rotas	22
4.5	Uma possível solução, com três níveis e nenhum cruzamento	26
4.6	Uma 3-coloração para o grafo de interseção de rotas	27
4.7	Solução com dois pontos de mudança de nível	29
4.8	Lado de uma placa de circuito impresso correspondente ao nível 1	30
4.9	Lado de uma placa de circuito impresso correspondente ao nível 2	30
4.10	Pares de cruzamentos	32
4.11	Grafo de interseção	32
4.12	Criação de mais um par de cruzamentos	33
4.13	Melhor distribuição de níveis	39

Lista de Tabelas

4.1	Níveis de vôo segundo a direção do vôo	20
4.2	Matriz de adjacências	22
4.3	Parâmetros para o Problema 1	23
4.4	Matriz de adjacências do grafo de interseção de pares de cruzamentos	32
4.5	Consumo de combustível por altitude	36
4.6	Coefficientes para o custo unitário	37
4.7	Distâncias e frequências de vôos para o conjunto de rotas	38
5.1	Resultados do problema 2	42

Capítulo 1

Introdução

O crescente tráfego aéreo traz problemas que o Gerenciamento de Fluxo de Tráfego Aéreo (GFTA) deve tratar. A segurança é um requisito que o GFTA deve alcançar, assegurando-se de que as rotas aéreas estão separadas. Além deste requisito principal, há outros assuntos tratados pelo GFTA, como a minimização do consumo de combustível e do número de manobras, como mudanças de nível.

Estes problemas devem ser otimizados simultaneamente, gerando um problema de otimização multi-objetivo. O algoritmo de Resfriamento Simulado [26] é extensivamente aplicado como uma heurística moderna para resolver problemas de otimização com objetivo único. Neste trabalho, o algoritmo foi adaptado para suportar otimização multi-objetivo.

Aqui, apenas rotas diretas entre origem e destino foram consideradas, como em [5] e [6]. Estas rotas devem ser distribuídas em níveis de vôo pré-definidos [9, sec. 10.1.5.3] com separação de 1.000 pés, abaixo de 29.000 pés de altitude, ou de 2.000 pés, acima deste nível. Em [16] e [38], três variantes do problema de particionamento de rotas em níveis são apresentadas. Estas estão relacionadas a problemas da Teoria dos Grafos, como o *maximum k-cut* [25] e a *coloração mínima de vértices* [42], ou à minimização do número de vias em uma placa de circuito integrado [40]. Estes problemas são apresentados na seção 4.2 (p. 21).

Todavia, [5], [6], [16] e [38] tratam apenas do problema de evitar cruzamentos entre rotas no mesmo nível de cruzeiro. Em [10] existe um tratamento indireto do consumo de combustível, porém sem permitir mudanças de nível.

Dado um conjunto de vôos, a versão multi-objetivo do Resfriamento Simulado é usada para evitar cruzamentos dentro de cada nível de vôo e para minimizar o consumo global de combustível e o número de mudanças de nível em cada rota.

O capítulo 2 (p. 3) apresenta definições sobre grafos utilizadas neste trabalho, assim como problemas a estes relacionados. No capítulo 3 (p. 7) é apresentado o algoritmo de Resfriamento Simulado, seu histórico, propostas para otimização multi-objetivo ([3], [4], [11], [36]) e a versão multi-objetivo utilizada neste trabalho. O capítulo 4 (p. 18) apresenta o problema de particionamento de rotas aéreas, as versões do problema com objetivo único, e a versão estendida, multi-objetivo. Os resultados obtidos com o problema de minimização do número de níveis (sub-seção 4.2.2 (p. 25)) aplicado a grafos do DIMACS [44] são apresentados no capítulo 5 (p. 40). Finalmente, o capítulo 6 (p. 43) apresenta as conclusões deste trabalho.

Capítulo 2

Definições sobre Grafos

Este capítulo apresenta conceitos básicos sobre Teoria dos Grafos que foram utilizados neste trabalho, assim como problemas relacionados à modelagem do problema de particionamento de rotas aéreas, apresentado no capítulo 4 (p. 18). Maiores detalhes podem ser vistos em [38, cap. 2] e [15].

2.1 Conceitos

Um grafo não-orientado $G = (V, E)$ é uma estrutura na qual $V = \{v_1, v_2, \dots, v_n\}$, com $n = |V| > 0$ é o conjunto de **vértices** de G e E é o conjunto de **arestas** de G . Cada aresta $e \in E$ é um par não-ordenado (v_i, v_j) , onde v_i e v_j são elementos de V .

Uma aresta da forma (v_i, v_i) é chamada **laço**.

Duas arestas da forma $e_1 = (v_i, v_j)$ e $e_2 = (v_i, v_j)$ são chamadas **arestas paralelas**.

Um **grafo simples** é aquele que **não** contém laços nem arestas paralelas. Neste trabalho, são tratados somente grafos simples.

Se $e = (v_i, v_j) \in E$, diz-se que v_i e v_j são **adjacentes** ou **vizinhos**.

Define-se o **grau** $d_G(v_i)$, do vértice v_i em G como sendo o número de vértices adjacentes a v_i .

Um **grafo regular** é o grafo que contém todos os vértices com o mesmo grau. Um **vértice isolado** é o vértice que possui grau nulo.

A **densidade** de arestas p , $0 \leq p \leq 1$, de um grafo é delimitada por:

$$p = \frac{|E|}{\frac{|V|*(|V|-1)}{2}}$$

Um grafo G pode ser representado graficamente no plano por pontos representando os vértices e por linhas representando as arestas. Por exemplo, considere o grafo definido por $G = (V, E)$, $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1 = (v_1, v_2), e_2 = (v_1, v_3), e_3 = (v_1, v_5), e_4 = (v_3, v_5)\}$, conforme apresentado na figura 2.1. Repare que v_4 é um vértice isolado.

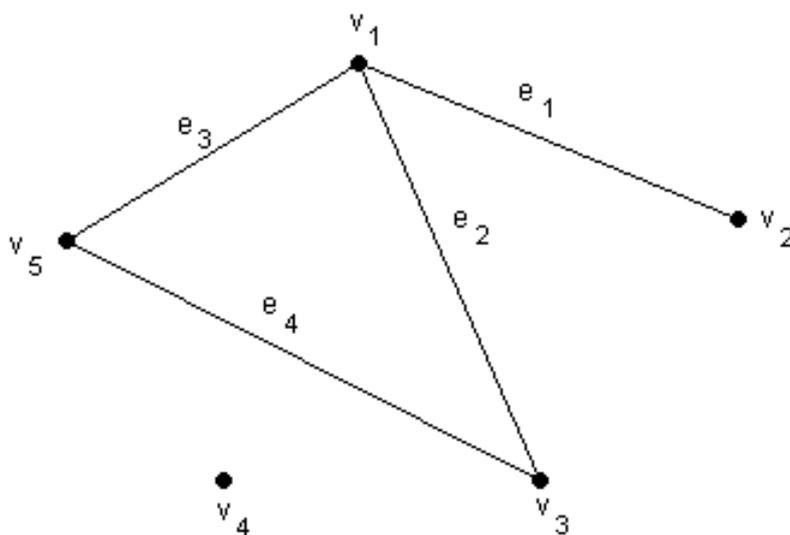


Figura 2.1: Exemplo de grafo

2.2 Problemas

Esta seção apresenta os problemas sobre grafos que estão relacionados com este trabalho.

2.2.1 *Maximum k-cut problem*

O *maximum k-cut problem* está relacionado com o problema da sub-seção 4.2.1 (p. 21). Este problema consiste em, dado um grafo $G = (V, E)$, particionar seus vértices em k conjuntos, de modo que o número de arestas entre os conjuntos seja máximo. Ou, em outras palavras, que o número de arestas dentro dos conjuntos seja mínimo.

Este é um problema APX-completo [24], o que significa que é possível resolvê-lo em tempo polinomial com um erro relativamente pequeno [12]. A obtenção de uma solução exata é um problema NP-difícil [25].

Usando o grafo da figura 2.1 como exemplo, e considerando-se $k = 2$, os possíveis particionamentos dos vértices são:

$$V_1 = \{v_1\} \text{ e } V_2 = \{v_2, v_3, v_4, v_5\}$$

$$V_1 = \{v_1, v_4\} \text{ e } V_2 = \{v_2, v_3, v_5\}$$

$$V_1 = \{v_1, v_5\} \text{ e } V_2 = \{v_2, v_3, v_4\}$$

$$V_1 = \{v_1, v_4, v_5\} \text{ e } V_2 = \{v_2, v_3\}$$

Como o vértice v_4 possui grau nulo, pode ser incluído em qualquer um dos conjuntos, de modo indiferente. Cada um destes particionamentos colocam três arestas entre os conjuntos.

2.2.2 *Coloração (mínima) de vértices*

Uma coloração de vértices de um grafo $G = (V, E)$ é uma atribuição de cores aos vértices, de forma que dois vértices adjacentes possuam sempre cores diferentes. A coloração mínima de vértices significa encontrar um menor T de forma que $G = (V, E)$ possa ser colorido com as cores $\{c_1, c_2, \dots, c_T\}$. Desta forma, diz-se que G é T -colorível. Trata-se de um problema NP-completo [42].

O problema de coloração mínima de vértices está relacionado ao problema da sub-seção 4.2.2 (p. 25).

Na figura 2.2, é apresentado o grafo da figura 2.1, o qual é um grafo 3-colorível.

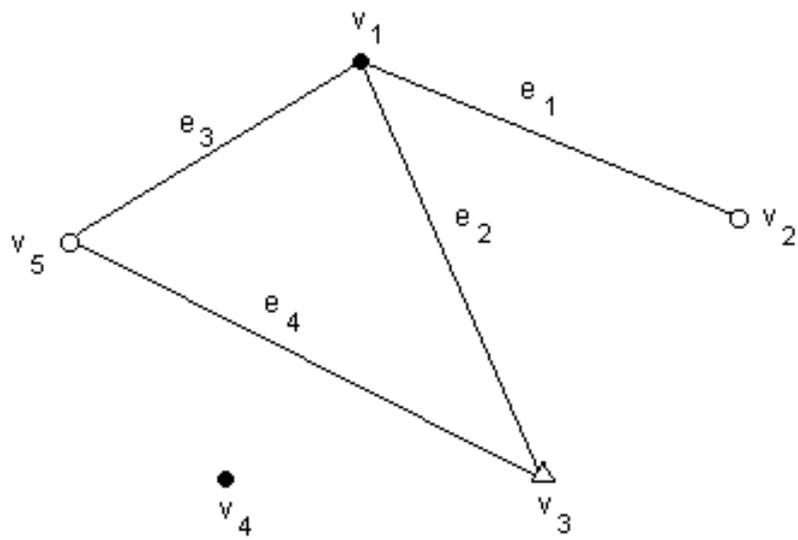


Figura 2.2: Exemplo de coloração de grafo

Capítulo 3

Resfriamento Simulado

Neste capítulo é apresentado um breve histórico sobre o Resfriamento Simulado (*Simulated Annealing*, em Inglês) e seu funcionamento. São também apresentadas propostas para a adaptação do Resfriamento Simulado para a otimização multi-objetivo ([3], [4], [11], [36]) . Maiores detalhes podem ser vistos em [7, cap. V] e [8, cap. 16].

3.1 Histórico

Nesta seção, são apresentados os métodos que precederam o Resfriamento Simulado [26]: o Método de Monte Carlo [34] e o Algoritmo de Metropolis [33].

3.1.1 O Método de Monte Carlo

O Método de Monte Carlo é um procedimento computacional bastante utilizado, cujo nome se deve ao fato de que sua operação probabilística se assemelha aos jogos de azar dos cassinos de Monte Carlo. Embora suas bases teóricas fossem conhecidas há muito tempo, este método só pôde ser aplicado após o desenvolvimento do computador digital.

A idéia básica do Método de Monte Carlo na solução de um problema com incógnita q é descobrir uma variável aleatória X , cuja média \bar{X} seja igual ao valor da referida incógnita, permitindo-se abstrair o problema em si e resolvê-lo, indiretamente, através do cálculo dessa variável.

$$q = \bar{X}$$

A média \bar{X} pode ser aproximada, tanto quanto for desejado, pelo cálculo da média de um grande número de valores x_i que, após sorteados através de um mecanismo capaz de gerar a aleatoriedade da variável X e devidamente contados, apresentam-se com frequência $f_i(1 \leq i \leq k)$, ou seja:

$$X \approx \frac{\sum_{i=1}^k x_i f_i}{\sum_{i=1}^k f_i}$$

Sendo p_i a probabilidade de ocorrência do valor x_i na variável aleatória X , a equação acima pode ser reescrita como:

$$X \approx \sum_{i=1}^k x_i p_i$$

Como exemplo, pode-se utilizar o Método de Monte Carlo para o cálculo de uma figura plana S , colocando-a dentro de um quadrado de lado l e sorteando-se dois números aleatórios (um para a ordenada e outro para a abscissa) no intervalo entre zero e l , marcando-se um ponto dentro do quadrado para cada par sorteado. Isto pode ser visto na figura 3.1.

Fazendo-se esse procedimento um grande número de vezes, teremos um total t de pontos dentro do quadrado, sendo que n pontos estarão dentro da figura. Assim, pode-se calcular a área da figura por:

$$S = l^2 * \frac{n}{t}$$

3.1.2 O Algoritmo de Metropolis

O Método de Monte Carlo também pode ser utilizado para o cálculo de propriedades de sistemas termodinâmicos como, por exemplo, um gás perfeito em equilíbrio em uma dada temperatura T , contido em um recipiente fechado.

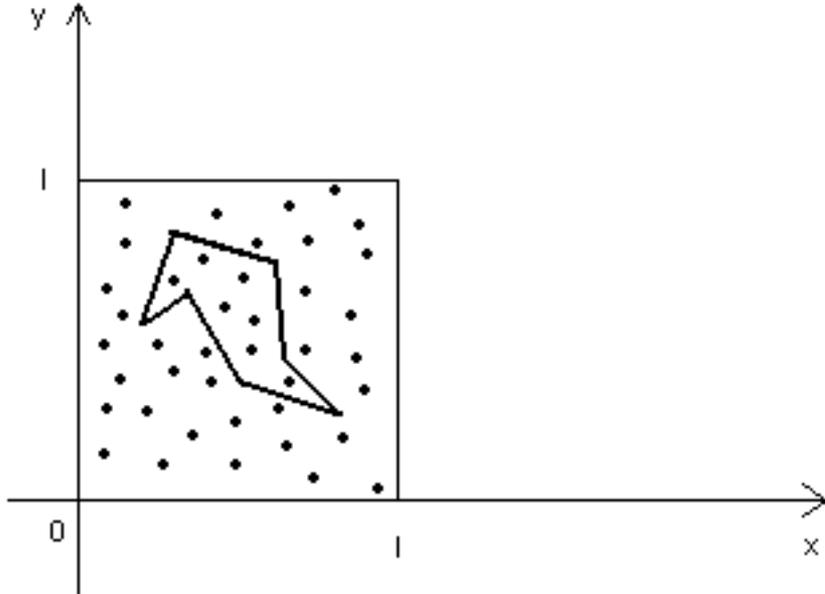


Figura 3.1: Cálculo da área da figura usando o Método de Monte Carlo

Para tal sistema, a densidade de probabilidade p relativa a uma energia E é dada pela distribuição de Boltzmann [31]:

$$p = e^{-\frac{E}{\mathcal{K}T}}$$

onde \mathcal{K} é a constante de Boltzmann [21]. A energia total E_t deste sistema estatístico é a soma das energias médias de suas \mathcal{N} moléculas.

Pela distribuição de Boltzmann, percebe-se que, para temperaturas muito altas, há uma grande probabilidade de encontrarem-se moléculas em todos os níveis energéticos, enquanto que, para baixas temperaturas, apenas moléculas com pequenos níveis de energia são detectadas com frequência. Assim sendo, para um gás em equilíbrio na temperatura T , não há necessidade de considerarmos, para efeito de cálculo da energia média, todos os valores de energia possíveis, mas apenas aqueles com probabilidade de ocorrência significativa.

No algoritmo de Metropolis [33], dada uma configuração inicial C_0 do sistema e uma temperatura de equilíbrio T_0 , o algoritmo escolhe aleatoriamente uma molécula $M_i (1 \leq i \leq \mathcal{N})$ e valores Δx e Δy através dos quais movimenta a molécula M_i de sua posição atual, gerando uma configuração diferente C_k do sistema.

A variação ΔE_t da energia total do sistema é calculada por $\Delta E_t = E_t(C_k) - E_t(C_{k-1})$, onde $E_t(C_k)$ é a energia do sistema na configuração C_k . Se a variação ΔE_t for negativa, a configuração é imediatamente aceita como a nova configuração do sistema. Caso contrário, a aceitação da nova configuração é decidida por sorteio, com probabilidade $p = e^{-\frac{\Delta E_t}{kT}}$. Em caso de rejeição, a nova configuração é igual à anteriormente existente.

Repetindo-se este procedimento um número grande de vezes, converge-se para uma posição média final que representa o equilíbrio do gás na temperatura T_0 .

3.2 Funcionamento

Utilizando-se o Algoritmo de Metropolis com temperaturas cada vez menores, pode-se simular a condensação de um gás para o estado líquido e, em seguida, emular a solidificação do mesmo. Sendo a redução da temperatura suficientemente lenta, forma-se uma estrutura cristalina representante do estado mais organizado conhecido, cuja energia é a menor que tal sistema pode atingir.

Observando-se a semelhança entre a formação de cristais a partir do vapor e a minimização global de funções matemáticas, foi formulado o método de Resfriamento Simulado [26], que é basicamente a aplicação sucessiva do método de Metropolis para uma seqüência de temperaturas decrescentes, com o objetivo de gerar uma configuração de energia mínima.

Podem-se encontrar provas formais de convergência para o algoritmo de Resfriamento Simulado em [35] e [17].

3.2.1 O Algoritmo

Dada uma função f , com um conjunto \mathcal{D} de soluções viáveis, uma solução inicial $x_0 \in \mathcal{D}$ e uma temperatura inicial T_0 , a qual na realidade é um número grande que serve como parâmetro do método, seleciona-se aleatoriamente uma variável da função e modifica-se seu valor, também de forma aleatória.

Da mesma forma que no cálculo da energia de um gás perfeito, calcula-se o valor da função com a variável contendo o novo valor e verifica-se a variação Δf , correspondente à diferença entre o valor novo e o valor anterior da função. Caso Δf seja negativo, o novo valor da função é prontamente aceito. Se Δf for positivo, calcula-se a probabilidade de Boltzmann $p = e^{-\frac{\Delta f}{T}}$ (a constante de Boltzmann é usualmente colocada como unitária, dado que não se trata mais de um gás perfeito, mas de uma analogia). Sorteia-se um número aleatório e verifica-se se o mesmo está entre 0 e p . Se isto ocorre, o novo valor é aceito, embora aumente o valor da função f . Do contrário, o valor anterior da variável é mantido.

Quando a média da função objetivo se estabiliza, ou quando a solução não é alterada, reduz-se a temperatura e repete-se o procedimento anterior. Isto é feito até que o sistema se estabilize em uma dada solução.

Esta possibilidade de se aumentar o valor da função f permite ao método escapar de mínimos locais, como exemplificado na figura 3.2, quando a temperatura é alta. Com temperaturas baixas, limita-se a busca por um mínimo apenas a vales mais profundos.

O Resfriamento Simulado está esquematizado no algoritmo 1.

Algoritmo 1: Resfriamento Simulado

Dados: f uma função a ser minimizada;

$x_0 \in \mathcal{D}$ uma solução viável inicial;

T_0 uma temperatura inicial;

$r(k)$ uma sequência decrescente de temperaturas;

início

$k := 0$;

$x_k := x_0$;

$T_k := T_0$;

enquanto *o sistema não está cristalizado* **faça**

enquanto *o sistema não atinge o equilíbrio térmico em T_k* **faça**

selecione aleatoriamente uma solução viável $x' \in \mathcal{D}$;

$\Delta f := f(x') - f(x_k)$;

se $\Delta f \leq 0$ **então** $x_k := x'$;

senão $x_k := x'$ com probabilidade $p = e^{-\frac{\Delta f}{T_k}}$;

fim

$T_{k+1} := r(k)$;

$x_{k+1} := x_k$;

$k := k + 1$;

fim

retornar x_k ;

fim

Para a melhor compreensão do algoritmo, deve-se explicitar o significado de o sistema estar em equilíbrio térmico e cristalizado, termos oriundos da Termodinâmica, porém sem análogos nos problemas de Otimização Combinatória. Estes conceitos podem ser visualizados na figura 3.3.

Equilíbrio Térmico O sistema encontra-se em equilíbrio térmico quando, para uma dada temperatura T_k , a média da função objetivo, calculada a partir das

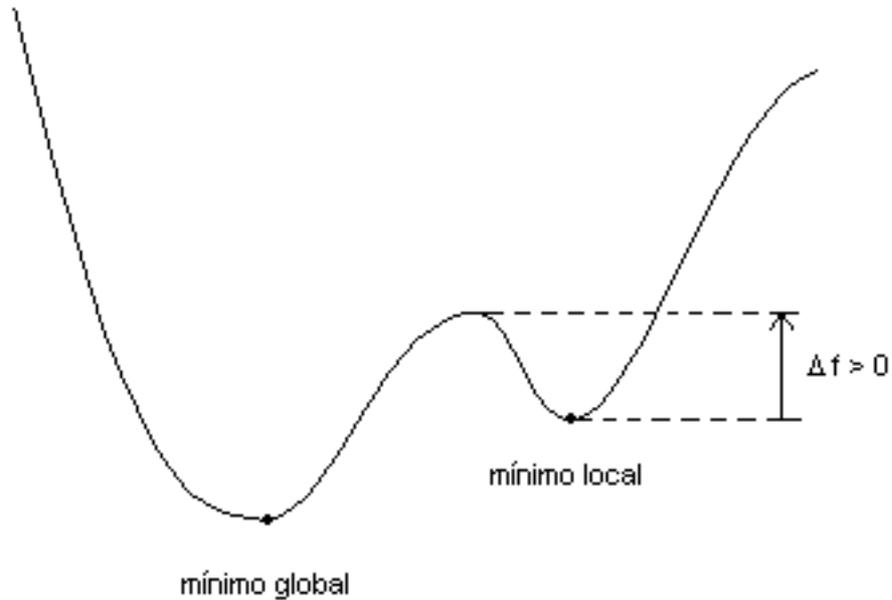


Figura 3.2: Perfil de uma função objetivo com mínimo local

soluções aceitas no decorrer do algoritmo, estabiliza-se, tendo variação inferior a ε por n iterações sucessivas, sendo ε e n fornecidos como parâmetros. Pode-se considerar equilíbrio térmico quando não há alteração da solução no decorrer de m iterações sucessivas, sendo m também definido como parâmetro. Quando se reduz a temperatura, é computada uma nova média.

Cristalização O sistema cristaliza-se quando, em analogia à formação de cristais a partir do vapor, a média da função objetivo não diminui ao se reduzir a temperatura. Na melhor hipótese, o sistema atingiu o seu mínimo global.

Definição da temperatura inicial

Para a definição de uma temperatura inicial que não seja demasiadamente alta (processamento inútil, busca randômica), nem demasiadamente baixa (convergência

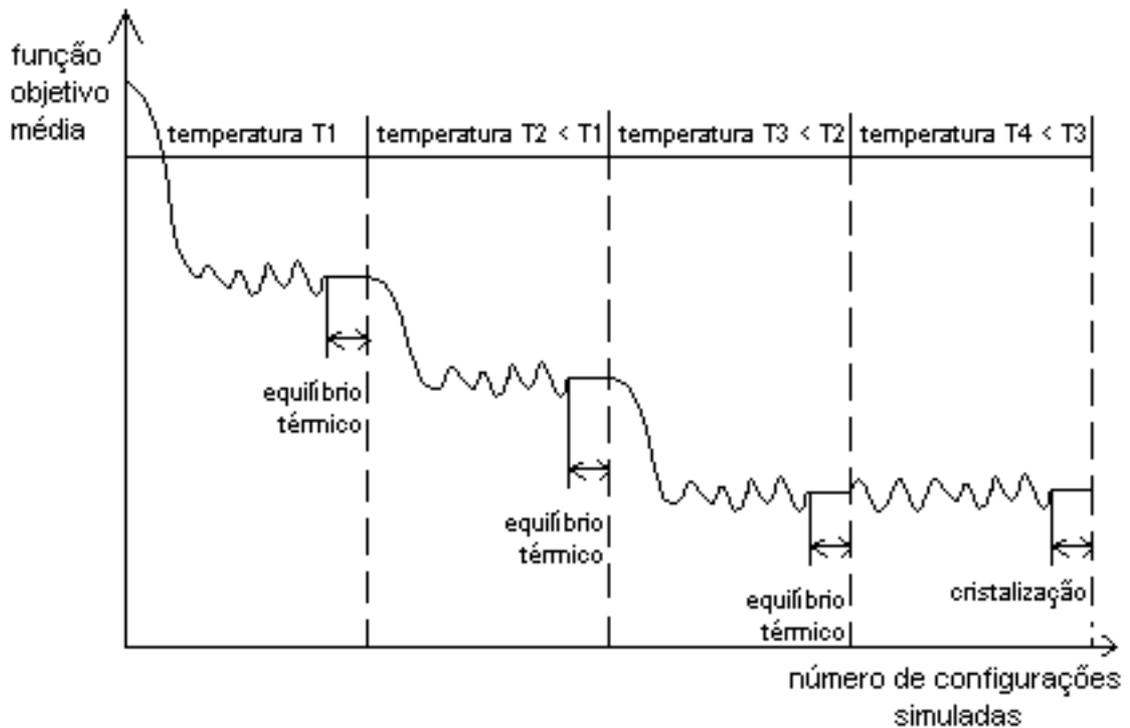


Figura 3.3: Equilíbrio Térmico e Cristalização no Resfriamento Simulado

prematura), pode-se utilizar o seguinte procedimento, baseado na distribuição de Boltzmann:

A partir da configuração inicial x_0 e de uma probabilidade p , fornecida como parâmetro, de aceitação de configurações que aumentem f , gerar um número grande de "vizinhos" de x_0 e, para os vizinhos x_i tais que $f(x_i) > f(x_0)$, calcular a média $\overline{\Delta F}$, sobre as variações da função f , e efetuar:

$$T_0 = \frac{-\overline{\Delta F}}{\ln(p)}$$

3.3 Resfriamento Simulado Multi-objetivo

O algoritmo de Resfriamento Simulado, conforme descrito anteriormente, efetua a minimização de uma única função. Neste trabalho, há a necessidade de efetuar a minimização de mais de uma função ao mesmo tempo. Esta seção apresenta a definição do problema de *otimização combinatória multi-objetivo*, um sumário de

esquemas ([3], [4], [11], [36]) existentes para adicionar o suporte à otimização multi-objetivo ao Resfriamento Simulado e o método adotado neste trabalho.

3.3.1 Otimização combinatória multi-objetivo

O problema de otimização combinatória multi-objetivo pode ser formulado [11] da seguinte forma:

$$\min\{f_1(x) = z_1, \dots, f_j(x) = z_j\} \text{ tal que } x \in \mathcal{D}$$

onde a solução $x = [x_1, \dots, x_n]$ é um vetor de variáveis discretas e \mathcal{D} é o conjunto de soluções possíveis.

Diz-se que uma solução $x \in \mathcal{D}$ é *eficiente* ou *Pareto-ótima* ([3], [11], [36]) se qualquer modificação que melhore uma função $f_i(x')$ irá degenerar uma função $f_j(x')$. O conjunto de soluções Pareto-ótimas é chamado *fronteira Pareto*. A fronteira Pareto representa todos os possíveis *trade-offs* entre as funções a serem otimizadas.

3.3.2 Métodos existentes

Aqui é apresentado um rápido resumo dos métodos existentes para a incorporação da otimização multi-objetivo ao Resfriamento Simulado.

Pareto-otimalidade Em [11] e [36], a otimização multi-objetivo é baseada no conceito de Pareto-otimalidade. [36] não utiliza diretamente as funções a serem otimizadas, mas sim uma função $c(x)$ que estima a quantidade de soluções que dominam uma solução x dada. Diz-se que uma solução y *domina* x se:

$$\forall i f_i(y) \leq f_i(x) \text{ e } \exists j f_j(y) < f_j(x)$$

Utiliza-se então a função $c(x)$ como no Resfriamento Simulado com objetivo único.

[11] efetua a verificação de dominância para definir se uma solução x' será testada pelo Resfriamento Simulado. Somente soluções não-dominadas pela solução corrente são testadas. A aceitação de uma nova solução não-dominada x' é dada por:

$$P(x', x, T, \Lambda) = \min\{1, \exp(\sum_{j=1}^n \lambda_j ((f_j(x') - f_j(x))/T))\}$$

onde $\Lambda = [\lambda_1, \dots, \lambda_n]$ é um vetor de pesos tal que $\forall i \lambda_i \geq 0$ e $\sum_{i=1}^n \lambda_i = 1$.

Otimizações *clandestinas* Em [3], apenas a primeira função $f_1(x)$ é utilizada para aceitação ou rejeição de uma solução no Resfriamento Simulado. Durante cada iteração do algoritmo, é utilizado o método de Newton [37] para otimizar as funções f_i antes de ser efetuado o teste de aceitação.

Múltiplas curvas de otimização Em vez de uma única curva de resfriamento, [4] mantém várias curvas, uma global e uma para cada função a ser otimizada. O algoritmo funciona do seguinte modo: testa-se se uma solução x' melhora todas as funções $f_i(x')$. Nesse caso, a aceitação é imediata.

Se piorar todas as funções, efetua-se o teste de aceitação para a distribuição de Boltzmann global. Se melhorar algumas funções, é selecionada uma função f_k . Se f_k melhora, x' é aceita. Caso contrário, efetua-se o teste de aceitação para a distribuição de Boltzmann referente a f_k .

3.3.3 Método utilizado

A implantação da otimização multi-objetivo no Resfriamento Simulado neste trabalho foi realizada através das seguintes mudanças:

- É mantida uma temperatura T_i para cada função f_i . Sendo a temperatura inicial $T_{i,0}$ definida a partir das diferenças do valor entre a solução inicial e soluções vizinhas, a temperatura serve como uma normalização da variação do valor de f_i no termo $e^{-\frac{\Delta f_i}{T_i}}$. Este termo pode então ser considerado uma medida da melhora ou da deterioração do valor de f_i . Esta propriedade é explorada na otimização multi-objetivo aqui implementada.

- Foi eliminado o teste de aceitação imediata. Isso pode ser feito porque, para um Δf_i negativo, ocorrerá $e^{-\frac{\Delta f_i}{T_i}} > 1$. Assim, ao se sortear um número aleatório p , certamente será menor do que o valor obtido.

O teste de aceitação passou a ser feito da seguinte forma:

$$p < \prod_{i=1}^n (e^{-\frac{\Delta f_i}{T_i}})^{k_i} \text{ com } k_i \geq 0 \text{ e } \sum_{i=1}^n k_i = n$$

Este procedimento pode ser entendido da seguinte forma: se, para uma função f_i , a nova solução x' melhora seu valor, a participação no produtório será superior a 1, aumentando a probabilidade de aceitação da mesma. Se, ao invés disso, x' piorar uma função f_j , a participação no produtório será no sentido de reduzir a probabilidade de aceitação. Com isto, as soluções que melhoram a maior quantidade de funções são prestigiadas.

Os expoentes k_i são normalmente colocados em 1. Porém, é possível colocar valores distintos, para aumentar ou diminuir a importância de uma função no processo de otimização.

Os testes de equilíbrio térmico e cristalização são efetuados apenas com a primeira função, f_1 . As temperaturas T_i decrescem segundo a mesma taxa de decaimento.

Capítulo 4

Particionamento de Rotas Aéreas

Neste capítulo, é apresentado o problema do particionamento de rotas aéreas (seção 4.1), versões simplificadas do problema (seção 4.2) e uma versão estendida do problema, na seção 4.3.

4.1 Definição do problema

O problema de particionamento de rotas aéreas consiste em, dado um conjunto de rotas aéreas e uma quantidade de níveis de vôo disponíveis, distribuir estas rotas, podendo-se efetuar mudanças de nível, de modo a não haver cruzamentos entre as mesmas. Existem restrições, como a existência de níveis ótimos, que minimizam o consumo de combustível ([1, sec 5.3], [10], [29]). Também é desejável a minimização do número de manobras como, por exemplo, mudanças de nível [2].

Neste problema, estão sendo consideradas rotas diretas entre origem e destino, como em [5], [6], [38]. Esta seção apresenta um breve sumário da nomenclatura utilizada, conforme sub-seção 4.1.1, além das normas sobre o uso dos níveis de vôo, como apresentado na sub-seção 4.1.2.

4.1.1 Nomenclatura

Na descrição de rotas aéreas, a altitude é usualmente denotada em pés (1 pé = 0,3048 metro). As distâncias são denotadas em milhas náuticas, abreviada por NM ou nmi, do inglês *nautical mile* (1 NM = 1.852 metros) [2], [9, sec. 14.4.1].

A direção do deslocamento é dado em graus, conforme indicado em uma bússola (figura 4.1). O Norte corresponde a 0° , o Leste a 90° , o Sul a 180° e o Oeste a 270° [9, sec. 14.2]. Uma viagem realizada entre as direções 0 e 179 é considerada como *direção Leste* (*eastbound*, em Inglês). Já uma viagem realizada entre as direções 180 e 359 é chamada *direção Oeste* (*westbound*).

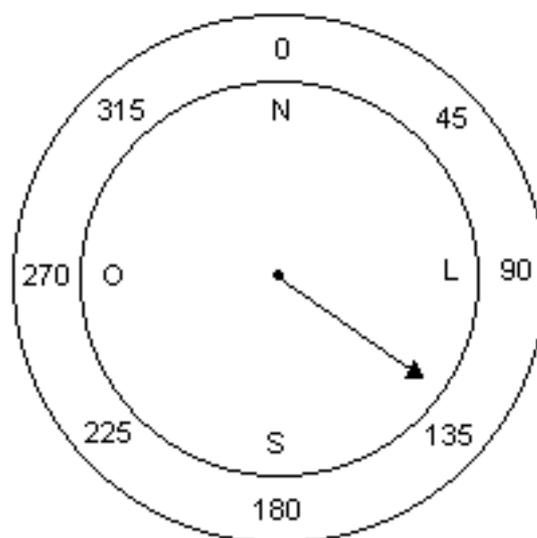


Figura 4.1: Pontos cardeais em uma bússola

Um nível de vôo (FL, de *Flight Level*) é denotado em centenas de pés. Deste modo, um avião localizado em FL 290 está a 29.000 pés de altitude.

4.1.2 Níveis de vôo

Os aviões, em suas rotas de cruzeiro, não voam em qualquer altitude, mas sim em altitudes pré-definidas, chamadas níveis de vôo. Abaixo de 29.000 pés (8.850 metros), a separação entre os níveis de vôo é de 1.000 pés. Acima dessa altitude, a separação é de 2.000 pés. Sobre o oceano, é requerida uma separação de 4.000 pés.

É também feita uma separação de acordo com a direção do vôo. Vôos realizados na direção Leste têm níveis de vôo alternados com vôos realizados na direção Oeste [9, sec. 10.1.5.3]. A tabela 4.1 apresenta os níveis de vôo mais utilizados.

Tabela 4.1: Níveis de vôo segundo a direção do vôo

	Direção Leste	Direção Oeste
Abaixo de FL 290	250, 270	260, 280
FL 290 e acima	290, 330, 370, 410	310, 350, 390

Segundo [20], é recomendado que os vôos com menos de 3 horas de duração (aqui considerado como 2.700 km ou 1.460 NM) utilizem os níveis abaixo de FL 290, reservando-se os níveis iguais ou superiores a 290 para os vôos mais longos. A figura 4.2 apresenta a separação dos níveis de vôo.

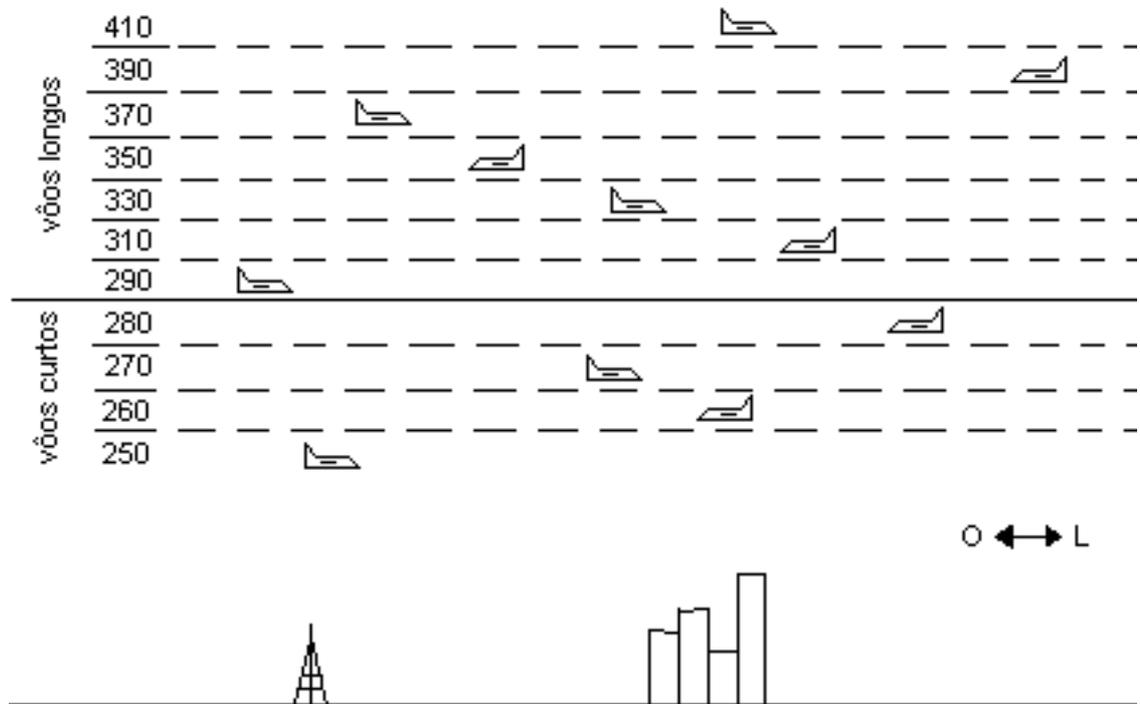


Figura 4.2: Níveis de vôo

4.2 Variantes básicas do problema

Nesta seção, são apresentadas três variantes do problema de particionamento de rotas aéreas, conforme apresentado em [16] e [38]. Estes problemas foram resolvidos com o Resfriamento Simulado, cuja implementação foi feita na linguagem de programação C ([18], [19], [39], [41]).

4.2.1 Primeiro Problema

O primeiro problema consiste em, dado um conjunto de rotas e uma quantidade fixa M de níveis de vôo, efetuar o particionamento destas rotas nos níveis disponíveis, de modo a minimizar o somatório do número de cruzamentos entre as rotas pertencentes a um mesmo nível. Um exemplo deste problema pode ser visto na figura 4.3.

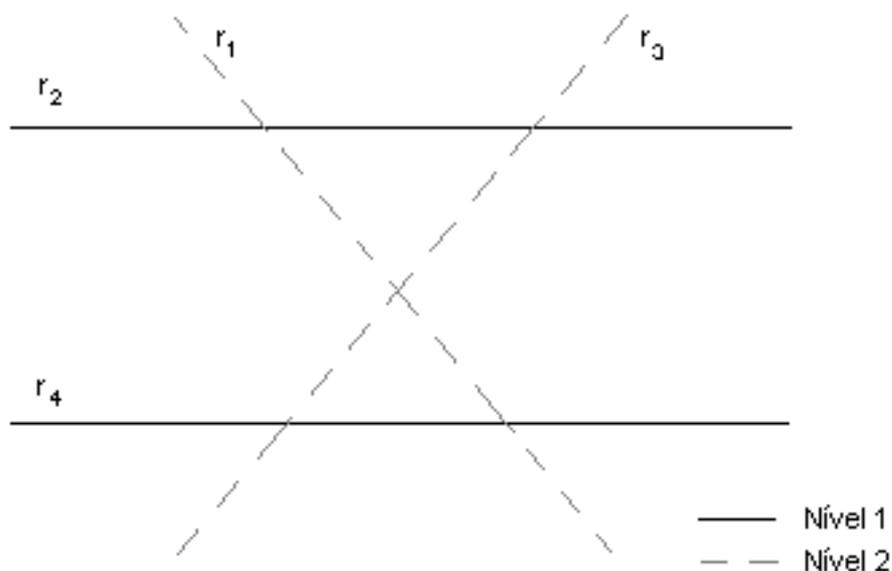


Figura 4.3: Exemplo de minimização de cruzamentos

Relação com a Teoria dos Grafos

A partir do conjunto de rotas dado, pode-se construir um grafo $G = (V, E)$ no qual cada rota é associada a um vértice. Cada aresta do grafo G representa um cruzamento entre duas rotas. O conjunto de rotas apresentado na figura 4.3 pode ser representado pelo grafo da figura 4.4, cuja matriz de adjacências é apresentada na tabela 4.2.

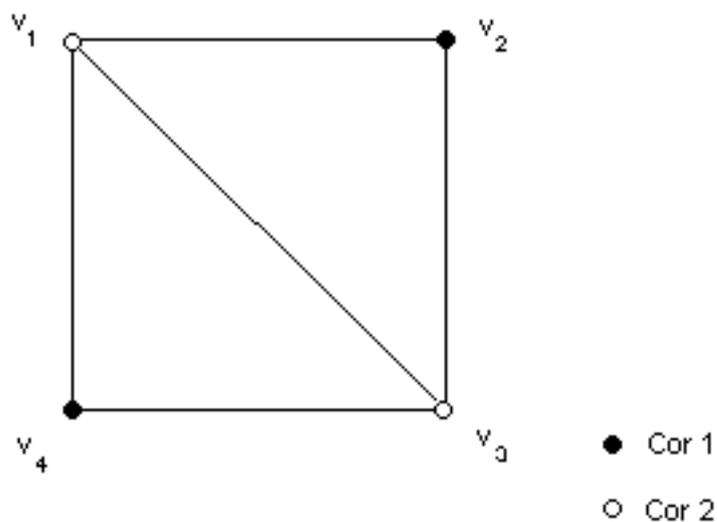


Figura 4.4: Grafo de interseção do conjunto de rotas

Tabela 4.2: Matriz de adjacências

$$A = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{vmatrix}$$

Deste modo, o problema de minimização de cruzamentos pode ser associado ao

problema de coloração dos vértices de $G = (V, E)$, chamado *grafo de interseção do conjunto de rotas*, usando M cores, de modo que o número de arestas entre vértices de mesma cor seja mínimo. Ou ainda que a quantidade de arestas entre vértices de cores diferentes seja máximo.

Este é o *maximum M-cut problem* (ver sub-seção 2.2.1 (p. 5)), ou seja, o particionamento dos vértices de $G = (V, E)$ em M conjuntos de tal forma que o número total de arestas entre os conjuntos seja máximo.

Na figura 4.4 é apresentada uma partição do conjunto de vértices que não é uma 2-coloração para o grafo de interseção do conjunto de rotas correspondente à figura 4.3. Existe uma aresta entre os vértices v_1 e v_3 , coloridos com a cor cinza, o que corresponde a um cruzamento entre as rotas r_1 e r_3 .

Considerações computacionais

Na implementação do Resfriamento Simulado (ver sub-seção 3.2.1 (p. 11)) para este problema, o conjunto de rotas foi representado por uma matriz de adjacências da forma descrita anteriormente. Os níveis ocupados pelas rotas (vértices) foram dispostos em um vetor. Por exemplo, as rotas apresentadas na figura 4.3 ficariam dispostos como $v = \{2, 1, 2, 1\}$ indicando $N_1 = \{r_2, r_4\}$ e $N_2 = \{r_1, r_3\}$. Os parâmetros fornecidos para o algoritmo estão apresentados na tabela 4.3.

Tabela 4.3: Parâmetros para o Problema 1

Decaimento	0.85
Equilíbrio	0.001
Cristalização	0.001
Mínimo no equilíbrio	10
Mínimo repetições	10
Probabilidade de aceitação	0.25

Decaimento é a taxa de resfriamento do algoritmo, após o equilíbrio térmico ser alcançado em uma dada temperatura T_k . No caso, significa que a temperatura T_{k+1} será $0.85 \times T_k$.

Equilíbrio é a variação relativa máxima que a média da função objetivo deve apresentar para que seja o processamento seja considerado como estando em equilíbrio térmico e, assim, reduzir-se a temperatura.

Cristalização é a variação relativa máxima entre as médias da função objetivo para duas temperaturas consecutivas. Se este critério é satisfeito, o processamento termina.

Mínimo no equilíbrio é a quantidade de iterações consecutivas que o algoritmo deve executar abaixo da taxa de equilíbrio, para que seja considerado o equilíbrio térmico. É feita esta exigência no algoritmo, uma vez que é possível a ocorrência de diferenças pequenas na média da função objetivo nos passos iniciais do processamento. Todavia, é extremamente difícil que isso ocorra por um número considerável de vezes consecutivas. Desse modo, previne-se uma convergência prematura do algoritmo.

Mínimo repetições se refere à outra forma de se considerar o equilíbrio térmico, que é quando não há alteração da solução por um dado número de iterações consecutivas. Este evento indica que foi atingido um ponto de mínimo, e as soluções que possuem um valor superior da função objetivo estão sendo continuamente rejeitadas. Este parâmetro foi adicionado porque a média da função objetivo não é alterada quando uma solução é rejeitada.

Probabilidade de aceitação é a probabilidade fornecida ao se calcular a temperatura inicial para uma execução do Resfriamento Simulado, usando-se a inversa da distribuição de Boltzmann.

Função objetivo

O Resfriamento Simulado necessita do valor de uma função para determinar se uma solução é "melhor" do que outra. Esta função está diretamente ligada com o problema a ser resolvido.

Para o problema de minimização de cruzamentos, sejam definidas as seguintes funções, considerando-se n como o número de rotas e r_i e r_j como sendo a i -ésima e j -ésima rotas do conjunto de rotas:

$$c(r_i, r_j) = \begin{cases} 1, & \text{se } (r_i \text{ e } r_j \text{ são adjacentes) e } (\text{nível}(r_i) = \text{nível}(r_j)) \\ 0, & \text{caso contrário} \end{cases}$$

A partir desta função auxiliar, pode-se definir a função objetivo:

$$f(s) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c(r_i, r_j)$$

Não é necessário fazer o somatório para cada nível separadamente.

4.2.2 Segundo Problema

Dado um conjunto de rotas que se interceptam, o segundo problema consiste em encontrar o menor número, M , de níveis de vôo, de tal modo que, particionando-se as rotas nesses M níveis, o somatório do número de cruzamentos seja nulo. Na figura 4.5, é apresentada uma solução com 3 níveis de vôo, sem cruzamentos entre rotas. O menor M possível para este conjunto de rotas é 3, não sendo possível uma solução sem cruzamentos com $M = 2$.

Relação com a Teoria dos Grafos

De forma análoga ao primeiro problema, o conjunto de rotas pode ser representado por um grafo, sendo as rotas representadas pelos vértices e os cruzamentos, pelas arestas. Porém, o problema de determinar o menor número de níveis que permite a eliminação de cruzamentos equivale ao problema de encontrar a coloração mínima de vértices (ver sub-seção 2.2.2 (p. 5)).

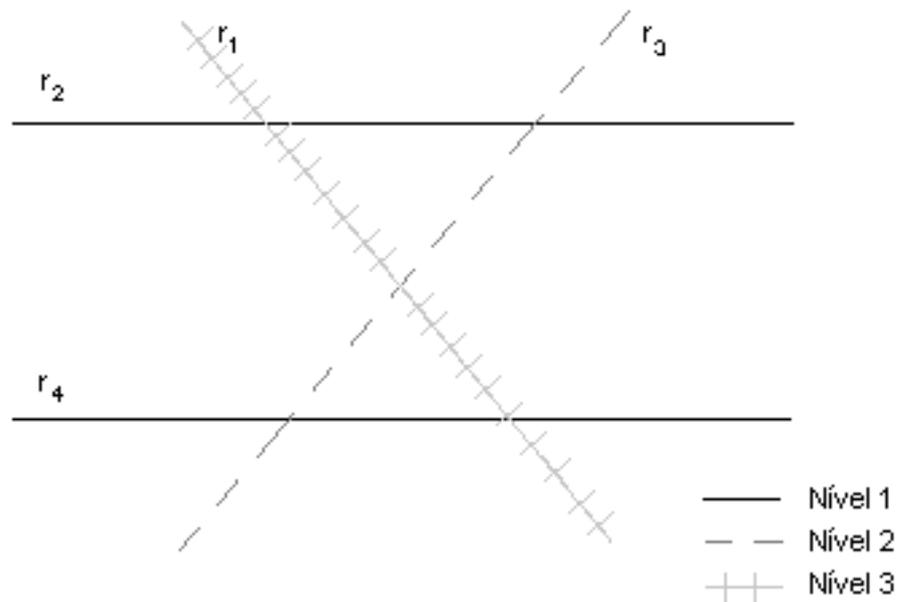


Figura 4.5: Uma possível solução, com três níveis e nenhum cruzamento

No caso, os vértices coloridos com a mesma cor correspondem aos subconjuntos de rotas e as cores representam os níveis de vôo. Na figura 4.6, é apresentada uma 3-coloração para o grafo de interseção de rotas correspondente à figura 4.5. Aqui não existe aresta entre vértices coloridos com a mesma cor, indicando a ausência de cruzamentos entre rotas na figura 4.5.

Considerações computacionais

Para este problema, foi utilizada a função objetivo da mesma forma que no primeiro problema, assim como os parâmetros fornecidos ao algoritmo de Resfriamento Simulado. Todavia, foram efetuadas algumas alterações na implementação do Resfriamento Simulado para o segundo problema, conforme segue:

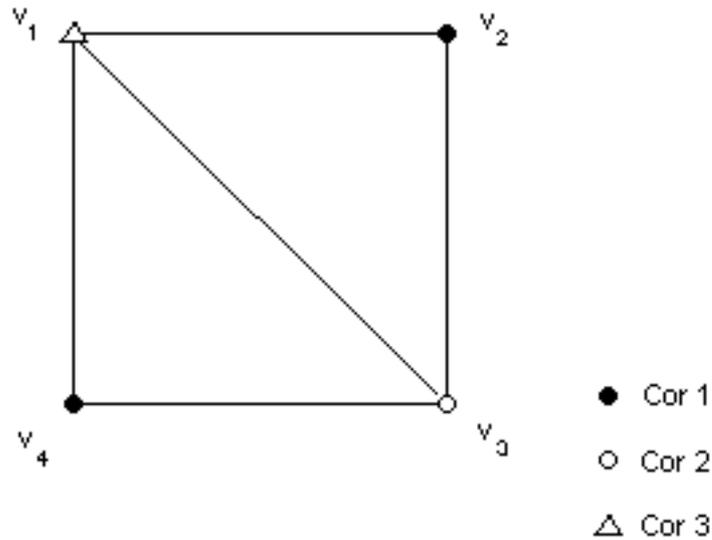


Figura 4.6: Uma 3-coloração para o grafo de interseção de rotas

Busca crescente por níveis Em [38], começa-se a busca pelo número mínimo de níveis a partir da quantidade máxima possível de níveis (ou cores, no caso da coloração de grafos), isto é, um nível para cada rota, ou uma cor para cada vértice. Então, vai-se reduzindo a quantidade de níveis (cores) até que não seja mais possível uma coloração de vértices, ou uma distribuição de rotas sem cruzamentos.

Neste trabalho foi feito o inverso, começando-se com a quantidade mínima de níveis ou cores (ou seja, $M = 2$) e, a cada execução, incrementa-se a quantidade de níveis/cores, até que seja encontrada uma solução sem conflitos (ver sub-seção A.1.2 (p. 52)). A solução encontrada para cada M tem o nível com a maior quantidade de conflitos dividido (ver sub-seção A.1.1 (p. 50)), sendo então usada como solução inicial para a execução do Resfriamento Simulado para $M + 1$ níveis.

O raciocínio sobre esta forma de resolver o problema pode ser entendido da seguinte maneira: ainda que uma solução com poucos níveis não seja satisfatória

(dado que existem conflitos), ao efetuar-se a divisão do nível mais conflituoso, esta estará mais próxima da solução satisfatória, sendo portanto uma boa instância inicial para a execução seguinte do algoritmo.

Esta abordagem trouxe resultados bastante superiores à abordagem tradicional.

Alteração seletiva da solução No algoritmo padrão do Resfriamento Simulado, a escolha da variável e do valor a ser alterado em uma dada solução é feita de forma aleatória. Neste problema, foi inserido um teste na escolha da variável, no caso uma rota (ou vértice). Se a rota não estivesse com cruzamentos com outras rotas no mesmo nível, era escolhida outra rota.

A idéia sobre este procedimento é de que, ao alterar-se o nível de uma rota que não recebe cruzamentos, o melhor que pode ocorrer é não piorar a solução. Porém, quando se altera o nível de uma rota que está recebendo cruzamentos, existe a possibilidade de melhora.

Tal alteração trouxe uma melhora significativa, tanto no tempo de execução, quanto na qualidade da solução. Posteriormente, essa alteração foi incorporada à resolução do primeiro problema.

Os resultados obtidos no segundo problema são apresentados na seção 5.2 (p. 41).

4.2.3 Terceiro Problema

Dado um conjunto de rotas que se interceptam e um número fixo de níveis de vôo M , este problema consiste em encontrar o menor número de inserções de pontos de mudança de nível NM de modo que, ao se particionarem as rotas em M níveis, o somatório do número de cruzamentos seja nulo.

De certa forma, este problema é mais realista do que os dois anteriores, dado que o primeiro problema permite a existência de cruzamentos dentro de um mesmo nível, ainda que em quantidades reduzidas, o que seria indesejável em uma situação real. O segundo problema assume que se pode utilizar quantos níveis forem desejados, o que não é verdade, uma vez que existe uma distância entre os níveis a ser respeitada

e a altitude máxima que uma aeronave pode alcançar é limitada por sua potência e pelo fato de a atmosfera ficar cada vez mais rarefeita em altitudes maiores ([13], [14, pp. 9.7–9.8]). Desse modo, o número de níveis é limitado.

Na figura 4.7 é apresentada uma solução para o conjunto de rotas das figuras 4.3 e 4.5 com dois níveis e dois pontos de mudança de nível.

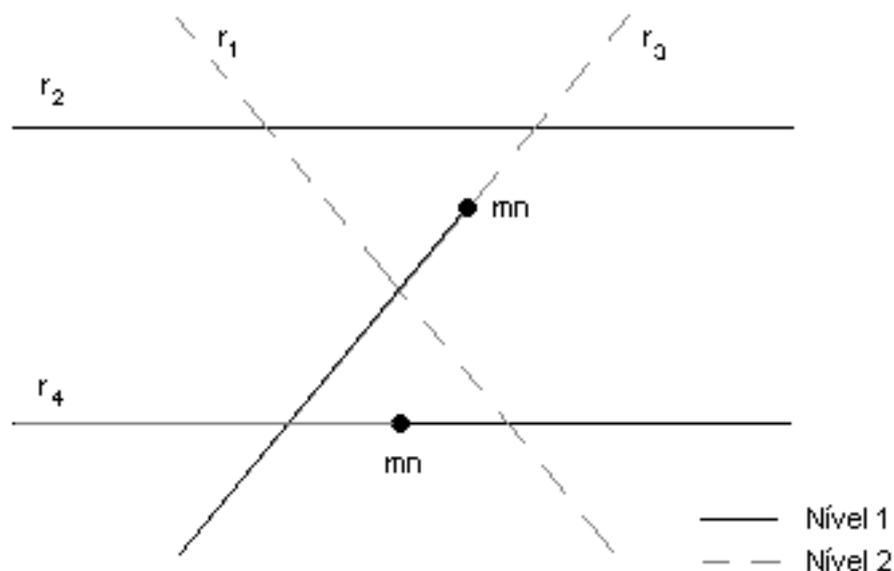


Figura 4.7: Solução com dois pontos de mudança de nível

Para $M = 2$, este problema é equivalente à minimização do número de vias em uma placa de circuito impresso [40], de forma que, duas trilhas de um mesmo lado da placa nunca se cruzem. As trilhas correspondem às rotas e os lados, aos níveis de vôo.

As figuras 4.8 e 4.9 apresentam os dois lados de uma placa de circuito impresso. Em nenhum dos casos ocorre cruzamento entre trilhas. Notar também que existem duas vias ligando um lado a outro da placa.

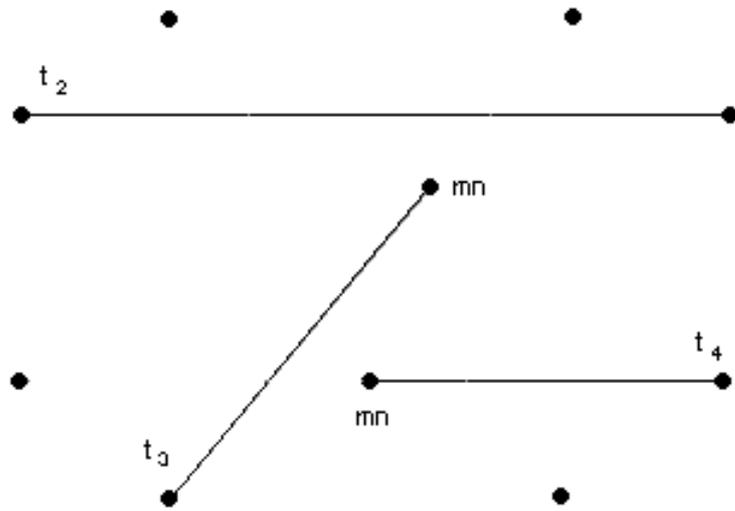


Figura 4.8: Lado de uma placa de circuito impresso correspondente ao nível 1

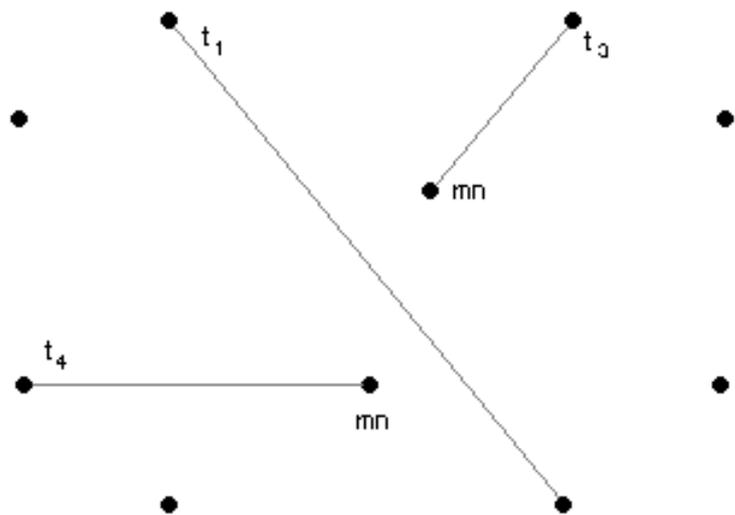


Figura 4.9: Lado de uma placa de circuito impresso correspondente ao nível 2

Representação

Como uma mudança de nível sempre ocorre entre dois cruzamentos, ou par de cruzamentos, pode-se descartar a parte da rota entre um aeroporto e um cruzamento. Desta forma, somente os segmentos de rotas limitados por cruzamentos serão relevantes neste contexto, particularmente os pares de cruzamentos.

Assim, é possível representar-se o problema por um conjunto de pares de cruzamentos, conforme exemplificado na figura 4.10. Se dois pares de cruzamentos são adjacentes e pertencem a rotas distintas, diz-se que são adjacentes tipo 1. Caso dois pares de cruzamentos sejam adjacentes e pertençam à mesma rota, diz-se que são adjacentes tipo 2.

A partir do conjunto de pares de cruzamentos, pode-se obter o *grafo de interseção de um conjunto de pares de cruzamentos* correspondente, aplicando-se as seguintes regras:

- Cada vértice corresponde a um par de cruzamentos.
- Dois vértices são adjacentes por aresta tipo 1, se os seus respectivos pares de cruzamentos são adjacentes tipo 1.
- Dois vértices são adjacentes por aresta tipo 2, se os seus respectivos pares de cruzamentos são adjacentes tipo 2.

Essa representação pode ser vista na figura 4.11. Na implementação aqui efetuada, as arestas tipo 2 foram, na verdade, colocadas na forma $4n + 2$, sendo n um número sequencial indicando a que rota a dupla de pares de cruzamentos pertence. Com esse formato, é possível saber que a aresta é tipo 2 e a qual rota pertence através de operações binárias **AND** e deslocamento. Essa representação está exemplificada na tabela 4.4.

A inserção de um ponto de mudança de nível corresponde à criação de mais um par de cruzamentos nessa rota. A figura 4.12 apresenta o conjunto de pares de

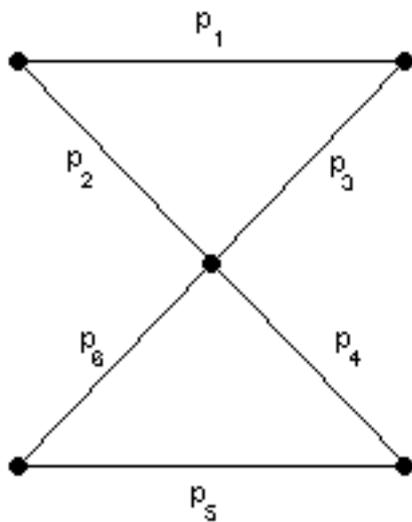


Figura 4.10: Pares de cruzamentos

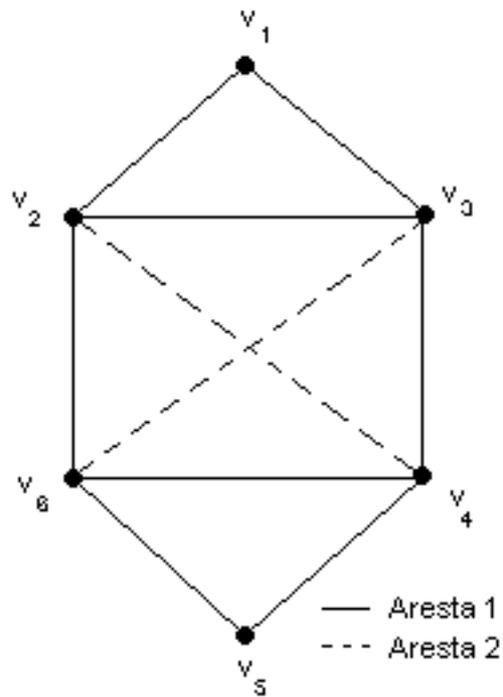


Figura 4.11: Grafo de interseção

Tabela 4.4: Matriz de adjacências do grafo de interseção de pares de cruzamentos

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 6 \\ 0 & 2 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 6 & 1 & 1 & 0 \end{pmatrix}$$

cruzamentos da figura 4.10 com a inserção de um ponto de mudança de nível no par de cruzamentos p_3 , criando um novo par de cruzamentos p_7 .

Os novos pares de cruzamentos são posicionados da seguinte forma:

- Havendo pares de cruzamentos adjacentes tipo 2, o novo par de cruzamentos será criado adjacente ao primeiro par de cruzamentos adjacente ao par que

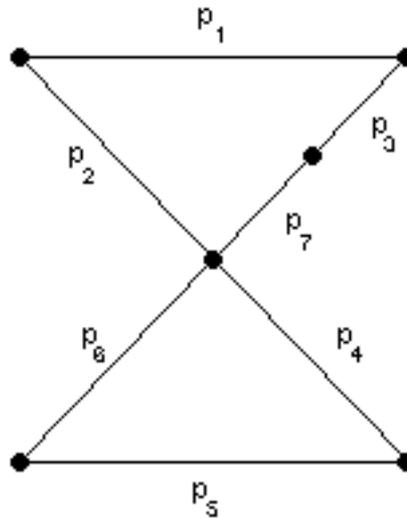


Figura 4.12: Criação de mais um par de cruzamentos

recebeu o ponto de mudança de nível (par receptor).

- Se o par receptor não possuir pares de cruzamentos adjacentes tipo 2, o novo par de cruzamentos será criado adjacente ao primeiro par de cruzamentos adjacente tipo 1 ao par receptor. Por exemplo, a recepção de um ponto de mudança de nível em p_5 criaria um par de cruzamentos p_8 adjacente a p_4 , dado que p_4 é o primeiro par de cruzamentos adjacente tipo 1 a p_5 e não há pares adjacentes tipo 2 a p_5 .

Considerações computacionais

O processamento deste problema foi dividido em duas fases. Na primeira fase, foi utilizado um grafo de rotas, obtido a partir do grafo de cruzamentos (ver sub-seção A.2.1 (p. 53)), da mesma forma que no problema 1, visando à minimização do número de cruzamentos. A distribuição de níveis obtida nesta fase é então utilizada

como solução inicial para a fase posterior.

Na segunda fase é feita a busca pela quantidade mínima de níveis necessários para a eliminação de cruzamentos. A partir da boa experiência obtida no problema 2, começa-se com a quantidade mínima de cruzamentos (ou seja, $NM = 0$) e incrementa-se NM a cada execução do Resfriamento Simulado até que seja encontrada uma solução sem cruzamentos.

Este problema é diferente dos dois anteriores porque nestes sempre era utilizado o mesmo grafo para a avaliação das soluções. Aqui, quando se insere uma mudança de nível, está-se, na verdade, acrescentando-se mais um vértice ao grafo de interseção dos pares de cruzamentos (ver sub-seção A.2.2 (p. 54)). Deste modo, é preciso anexar o grafo à solução correspondente, armazenando-se também os vértices correspondentes aos pares de cruzamentos que foram divididos.

Vale lembrar que, ao fazer-se o grafo de interseção de um conjunto de pares de cruzamentos, perde-se a informação de quais pares de cruzamentos estavam "à esquerda" e "à direita". Desse modo, é necessário manter uma estrutura à parte com esta informação. Isto é necessário para definir quais vértices serão adjacentes ao novo vértice, e quais permanecem adjacentes ao vértice antigo, correspondente ao par de cruzamentos que foi dividido.

Função objetivo

Assumindo-se que seja recebida a solução s já com as adições das mudanças de nível, juntamente com o respectivo grafo, para o problema de minimização do número de mudanças de nível, sejam definidas as seguintes funções, considerando-se n como o número de pares de cruzamentos e p_i e p_j como sendo a i -ésimo e j -ésimo pares de cruzamentos do conjunto de pares de cruzamentos:

$$c(p_i, p_j) = \begin{cases} 1, & \text{se } (p_i \text{ e } p_j \text{ são adjacentes tipo 1) e } (\text{nível}(p_i) = \text{nível}(p_j)) \\ 0, & \text{se } p_i \text{ e } p_j \text{ são adjacentes tipo 2, ou não são adjacentes} \end{cases}$$

A partir desta função auxiliar, pode-se definir a função objetivo:

$$f(s) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c(p_i, p_j)$$

Esta função efetua o somatório total do número de pares de cruzamentos adjacentes tipo 1 em todos os níveis, ou seja, o número total de cruzamentos entre rotas no mesmo nível.

4.3 Estendendo o problema

Esta seção propõe uma modelagem do problema de particionamento de rotas aéreas que incorpora restrições existentes em situações reais. Considerando-se um número fixo de níveis de vôo, como no terceiro problema (ver sub-seção 4.2.3 (p. 28)), deseja-se minimizar tanto o consumo de combustível, quanto o número de mudanças de nível por rota.

Estas restrições devem ser otimizadas simultaneamente, embora a preferência seja para a eliminação de cruzamentos dentro de cada nível de vôo, fazendo com que este problema seja de otimização multi-objetivo, conforme seção 3.3 (p. 14).

No problema 3, os trechos localizados entre o aeroporto e um cruzamento eram desconsiderados. Aqui, estes trechos deverão ser incluídos no cálculo do custo total.

4.3.1 Funções objetivo

A implementação das restrições propostas anteriormente requer a elaboração de uma função objetivo para cada uma destas. A primeira função, que conta o número de cruzamentos dentro de cada nível de vôo, é definida conforme a sub-seção 4.2.1 (p. 21) para a primeira fase e conforme a sub-seção 4.2.3 (p. 28) para a segunda fase.

A definição das funções objetivo para o consumo de combustível e para a contagem de mudanças de nível por rota são definidas a seguir.

Cálculo do custo unitário

Para o cálculo do custo total de um conjunto de rotas, é preciso fazer uma estimativa do custo para uma distância percorrida. No caso, será assumido que o custo unitário para a altitude ótima é igual a 1.

A partir dos valores de [1, sec. 5.3.2], têm-se as seguintes proporções de consumo de combustível por diferença entre a altitude utilizada e a altitude ótima, medida em níveis de vôo, descrita na tabela 4.5. Estes valores correspondem ao modelo A330-343.

Tabela 4.5: Consumo de combustível por altitude

Diferença para a altitude ótima	Custo
+20	1,03
0	1
-20	1,01
-40	1,032
-60	1,072

Com estes dados, pode-se aproximar a função de consumo de combustível pela diferença entre a altitude corrente e a altitude ótima por um polinômio de grau 4.

$$\text{custo}(d) = \sum_{i=0}^4 a_i * d^i$$

Usando-se os valores citados na tabela 4.5, os coeficientes a_i são apresentados na tabela 4.6.

Cálculo do custo total

A partir da definição do custo unitário, pode-se calcular o custo total de um conjunto de rotas, considerando-se também a distância percorrida em cada rota e a frequência de vôos. O nível ótimo foi, por simplificação, fixado em FL 330 [1, sec. 5.3.3].

Tabela 4.6: Coeficientes para o custo unitário

a_0	1
a_1	$1,25 * 10^{-4}$
a_2	$4,645833 * 10^{-5}$
a_3	$9,375 * 10^{-7}$
a_4	$8,854167 * 10^{-9}$

Quando um par de cruzamentos é dividido, considera-se a metade da distância do par original nos pares gerados.

Assim, o custo total é definido como segue, podendo-se utilizar esta fórmula na primeira e na segunda fases.

$$f_2(s) = \sum_{i=1}^n \text{custo}(\text{nível}(r_i) - 330) * \text{distância}(r_i) * \text{frequência}(r_i)$$

Contagem de mudanças de nível por rota

Como foi mencionado anteriormente, é desejável a minimização do número de manobras, como mudanças de nível, por rota. Nesta modelagem, a penalização para o excesso de mudanças de nível na mesma rota é feita contando-se o número de mudanças de nível elevado ao quadrado.

Assim, uma rota com 0 ou 1 mudança de nível participará do somatório com 0 ou 1. Uma rota com duas mudanças de nível terá peso 4, ou seja, pior do que duas rotas com uma mudança de nível cada (totalizando peso 2).

$$f_3(s) = \sum_{i=1}^n \text{mudanças}(r_i)^2$$

4.3.2 Exemplo

Tomando como exemplo o conjunto de rotas da figura 4.7 (p. 29), a serem distribuídos nos níveis FL 260 e FL 280 e considerando-se as distâncias e frequências dadas na tabela 4.7, calcular uma distribuição satisfatória de níveis.

Tabela 4.7: Distâncias e frequências de vôos para o conjunto de rotas

Par de cruzamentos	Rota	Distância (NM)	Frequência
p_1	r_2	950	15
p_2	r_1	580	25
p_3	r_3	670	20
p_4	r_1	450	25
p_5	r_4	850	10
p_6	r_3	400	20
p_7	r_1	150	25
p_8	r_2	225	15
p_9	r_3	250	20
p_{10}	r_2	100	15
p_{11}	r_4	250	10
p_{12}	r_1	200	25
p_{13}	r_3	150	20
p_{14}	r_4	100	10

A melhor distribuição de níveis encontrada é apresentada na figura 4.13, contendo duas mudanças de nível nos pares de cruzamentos p_2 e p_5 , pertencentes a rotas distintas, nenhum cruzamento entre trechos contidos no mesmo nível e custo total de 102.508.

Exemplo de transição

Para ilustração, é aqui apresentado um exemplo de transição, a qual conduziu à solução apresentada.

Temperaturas: $T_1 = 4,32809$, $T_2 = 1166$, $T_3 = 3,28403$

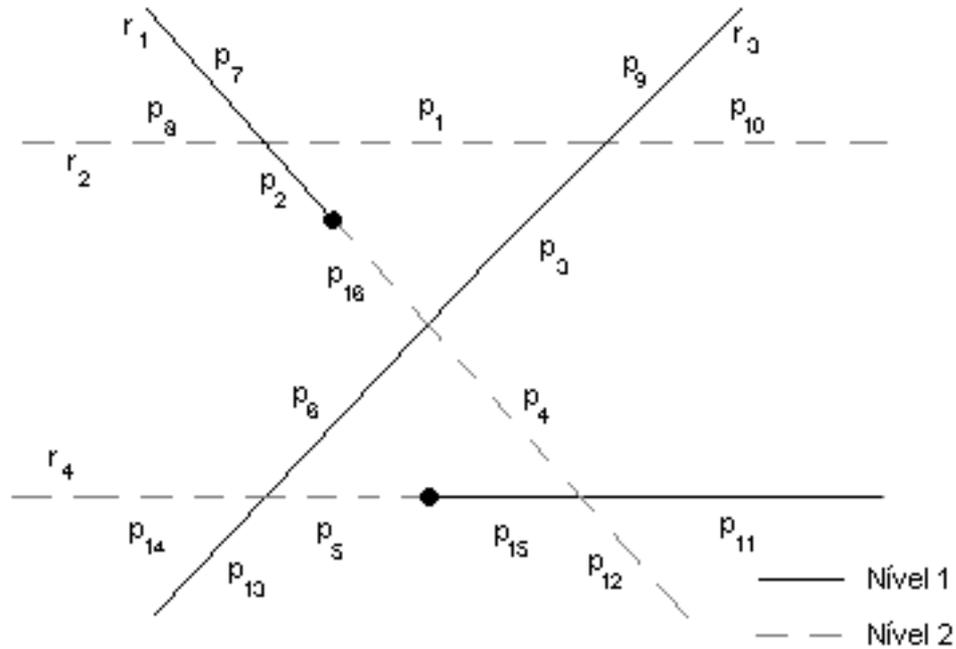


Figura 4.13: Melhor distribuição de níveis

Solução anterior: mudanças de nível em p_5 e p_{12} .

$$L_1 = \{p_3, p_6, p_9, p_{11}, p_{12}, p_{13}, p_{15}\}$$

$$L_2 = \{p_1, p_2, p_4, p_5, p_7, p_8, p_{10}, p_{14}, p_{16}\}$$

$$f_1(s) = 4, f_2(s) = 101982, f_3(s) = 2$$

Solução nova: mudanças de nível em p_5 e p_2 .

$$L_1 = \{p_2, p_3, p_6, p_9, p_{11}, p_{12}, p_{13}, p_{15}\}$$

$$L_2 = \{p_1, p_4, p_5, p_7, p_8, p_{10}, p_{14}, p_{16}\}$$

$$f_1(s) = 0, f_2(s) = 102508, f_3(s) = 2$$

$$e^{-\frac{\Delta f_1}{T_1}} = 2,51984, e^{-\frac{\Delta f_2}{T_2}} = 0,63695, e^{-\frac{\Delta f_3}{T_3}} = 1$$

O produtório tem resultado 1,605, levando à aceitação certa da solução nova.

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos com os testes realizados para o problema de minimização do número de níveis (conforme sub-seção 4.2.2 (p. 25)) aplicados aos grafos do DIMACS [44], bem como uma breve descrição destes grafos.

5.1 Descrição dos grafos do DIMACS

Esta seção apresenta uma rápida explicação sobre cada um dos grupos de grafos do DIMACS. Tratam-se de grafos bastante complexos, possuindo características próprias.

Os arquivos de entrada que formam o grafo possuem o seguinte formato: a primeira linha contém o número de vértices e o número de arestas do grafo. Cada uma das linhas subsequentes contém um par de vértices denotando uma aresta do grafo desejado. Esses grafos estão disponíveis em [44].

Neste trabalho, foi testado um subconjunto dos grafos do DIMACS, exatamente os que foram testados em [38]. Os grafos aqui citados têm seus números de vértices e arestas colocados entre parênteses.

Grupo dos Grafos Randômicos Arquivos com nome iniciado com DSJC contém grafos que pertencem à classe dos grafos randômicos utilizados por David Johnson em [23]. Os grafos com prefixo DSJR são os grafos geométricos e, se o prefixo for da forma DSJR...c, trata-se de um grafo geométrico complementar ao DSJR

correspondente.

DSJRx.1[c] é um grafo geométrico com x vértices, distribuídos aleatoriamente dentro de um quadrado unitário e, se no fim do arquivo houver um c , é um grafo complementar. Exemplos de grafos randômicos são DSJC125.5.col (125, 7782) e o DSJR500.1.col (500, 3555).

Grupo de Grafos Regulares São grafos gerados aleatoriamente com uma coloração conhecida, sendo que é possível que seu número cromático seja inferior que a coloração conhecida. A diferença entre o número cromático e a clique máxima é grande. São grafos regulares, isto é, todos os vértices possuem o mesmo grau. A sintaxe do nome pode ser resumida como FLAT x . y . z .col, sendo x o número de vértices, y o número de partições e z a regularidade. Foi testado o grafo FLAT300.20.0.col (300, 21375).

Grupo de Grafos de Gary Lewandowski Arquivos iniciados com SCH ou MUL são os da classe de grafos pertencentes ao mundo real [28]. Banco de dados: SCHOOL1.NSH.col (352, 14612), MULSOL.I.1.col (197, 3925).

Grupo de Grafos de Leighton Arquivos inicializados com LE são os grafos de Leighton [27]. Esses grafos são gerados de forma que o número cromático seja igual ao tamanho da clique máxima. Foi testado o grafo LE450.15b.col (450, 8168).

5.2 Resultados do Problema 2

Na tabela 5.1, são apresentados os resultados obtidos para o problema de obtenção do número mínimo de níveis para a não-ocorrência de cruzamentos (ver sub-seção 4.2.2 (p. 25)), comparados com os resultados de [38] e os valores ótimos, obtidos em [38] e [44].

Tabela 5.1: Resultados do problema 2

Grafo	Resultado ótimo	Resultado obtido	Resultado de [38]
MYCIEL6.col	7	7	8
DSJR500.1.col	12	12	17
MYCIEL7.col	8	8	8
GAMES120.col	9	9	9
ANNA.col	11	11	11
MULSOL.I.1.col	49	49	49
QUEEN9_9.col	10	11	11
DSJC125.5.col	17	22	25
LE450_15b.col	15	20	27
SCHOOL1_NSH.col	14	37	29
FLAT300_20_0.col	20	48	43

Capítulo 6

Conclusão

Este trabalho elabora um sistema de particionamento de rotas aéreas de forma diferente de outros sistemas nos quais somente a eliminação de cruzamentos dentro de cada nível de vôo era tratada ou, como em [10], o consumo de combustível era tratado apenas indiretamente, sendo que não era contemplada a possibilidade de mudanças de nível em uma dada rota. Aqui as restrições são explicitadas e diretamente tratadas, com o sistema efetuando a otimização simultânea das mesmas.

Para a elaboração deste sistema, foi utilizado o algoritmo de Resfriamento Simulado, o qual originalmente otimiza apenas uma única função. Neste trabalho, foi desenvolvida uma versão multi-objetivo do Resfriamento Simulado, que efetua a otimização simultânea das funções envolvidas de forma natural e direta. Essa forma apresenta uma relação com a Lógica Nebulosa ([22], [32]), para o caso em que todas as funções são pioradas.

Referências Bibliográficas

- [1] AIRBUS, *Getting to grips with Fuel Economy* [online]. Disponível na Internet via WWW. URL: <http://www.iata.org/NR/ContentConnector/CS2000/Siteinterface/sites/whatwedo/file/Airbus.Fuel.Economy.Material.pdf>. Consultado em 31 de Janeiro de 2008.
- [2] VAN DEN AKKER, J. M., VAN KEMENADE, C. H. M., KOK, J. N., "Evolutionary 3D-Air Traffic Flow Management". In: T. Bäck and D. Fogel and Z. Michalewicz (eds), *Handbook of Evolutionary Computation*, part G3.9, Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
- [3] AUBRY, J. F., BEAULIEU, F., SÉVIGNY, C., BEAULIEU, L., TREMBLAY, D., "Multiobjective Optimization with a Modified Simulated Annealing Algorithm for External Beam Radiotherapy Treatment Planning", *Medical Physics*, v. 33, n. 12, pp. 4718–4729, 2006.
- [4] AVELLO, E. A., BAESLER, F. F., MORAGA, R. J., "A Meta-heuristic Based on Simulated Annealing for Solving Multiple-objective Problems in Simulation Optimization". In: *36th Winter Simulation Conference*, pp. 508–513, Washington, DC, USA, December 2004.
- [5] BARNIER, N., BRISSET, P., "Graph Coloring for Air Traffic Flow Management". In: *CPAIOR'02: Fourth International Workshop on Integration*

- of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*, pp. 133–147, Le Croisic, France, March 2002.
- [6] BARNIER, N., BRISSET, P., "Graph Coloring for Air Traffic Flow Management", *Annals of Operations Research*, v. 130, n. 1-4, pp. 163–178, August 2004.
- [7] CARVALHO, L. A. V., *Síntese de Redes Neurais com Aplicações à Representação do Conhecimento e à Otimização*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1989.
- [8] CARVALHO, L. A. V., *Datamining - A Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração*. Rio de Janeiro, Ed. Ciência Moderna, 2005.
- [9] COMANDO DA AERONÁUTICA, *Manual do Especialista em Informação Aeronáutica* [online]. Disponível na Internet via WWW. URL: http://www.aisweb.aer.mil.br/aisweb_files/publicacoes/mca/mca_053-1_191200.pdf. Consultado em 10 de Janeiro de 2008.
- [10] CONSTANS, S., GADENNE, N., FONDACCI, R., "Applying Genetic Techniques to the Tactical Flight Level Assignment". In: *Advanced OR and AI Methods and Transportation, Proc. of 10th Jubilee Meeting of the EURO Working Group on Transportation and 16th Mini-Euro Conference*, pp. 69–75, Poznan, Poland, September 2005.
- [11] CZYZAK, P., JASZKIEWICZ, A., "Pareto Simulated Annealing – A Metaheuristic Technique for Multiple-objective Combinatorial Optimization", *Journal of Multi-Criteria Decision Analysis*, v. 7, pp. 34–47, 1998.
- [12] DEINEKO, V., JONSSON, P., KLASSON, M., KROKHIN, A., *The Approximability of MAX CSP with Fixed-value Constraints*, Report CoRR/abs/cs/0602075, Computing Research Repository, 2006.

- [13] FEDERAL AVIATION ADMINISTRATION, *Density Altitude* [online]. Disponível na Internet via WWW. URL: http://www.faa.gov/gslac/ALC/libview_normal.aspx?id=6847. Consultado em 19 de Janeiro de 2008.
- [14] FEDERAL AVIATION ADMINISTRATION, *Pilot's Handbook of Aeronautical Knowledge* [online]. Disponível na Internet via WWW. URL: http://www.faa.gov/library/manuals/aviation/pilot_handbook/media/faa-h-8083-25-2of4.pdf. Consultado em 25 de Março de 2008.
- [15] FEOFILOFF, P., KOHAYAKAWA, Y., WAKABAYASHI, Y., *Uma Introdução Sucinta à Teoria dos Grafos* [online]. Disponível na Internet via WWW. URL: <http://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf>. Consultado em 07 de Janeiro de 2008.
- [16] FONDACCI, R., GOLDSCHMIDT, O., LETROUIT, V., "Combinatorial Issues in Air Traffic Optimization", *Transportation Science*, v. 32, n. 3, pp. 256–267, 1998.
- [17] GRANVILLE, V., KRIVANEK, M., RASSON, J. P., "Simulated Annealing: A Proof of Convergence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 6, pp. 652–656, June 1994.
- [18] HUSS, E., *The C Library Reference Guide* [online]. Disponível na Internet via WWW. URL: http://www.acm.uiuc.edu/webmonkeys/book/c_guide/. Consultado em 18 de Janeiro de 2008.
- [19] IBM, *C/C++ Language Reference* [online]. Disponível na Internet via WWW. URL: http://www.nersc.gov/vendor_docs/ibm/pdf/sc094957.pdf. Consultado em 18 de Janeiro de 2008.
- [20] INTERNATIONAL CIVIL AVIATION ORGANIZATION, *Contingency Plans - SAM* [online]. Disponível na Internet via WWW. URL:

<http://www.icao.int/y2k/contingency/sam.html>. Consultado em 30 de Janeiro de 2008.

- [21] INTERNATIONAL UNION OF PURE AND APPLIED CHEMISTRY, *Boltzmann Constant* [online]. Disponível na Internet via WWW. URL: <http://www.iupac.org/goldbook/B00695.pdf>. Consultado em 25 de Março de 2008.
- [22] JANTZEN, J., *Tutorial on Fuzzy Logic* [online]. Disponível na Internet via WWW. URL: <http://www.iau.dtu.dk/~jj/pubs/logic.pdf>. Consultado em 25 de Março de 2008.
- [23] JOHNSON, D. S., ARAGON, C. R., MCGEOCH, L. A., SCHEVON, C., "Optimization by Simulated Annealing: an Experimental Evaluation; Part II, Graph Coloring and Number Partitioning", *Operations Research*, v. 39, n. 3, pp. 378–406, 1991.
- [24] KANN, V., *Maximum K-CUT* [online]. Disponível na Internet via WWW. URL: http://www.ensta.fr/~diam//ro/online/viggo_wwwcompendium/node88.html. Consultado em 07 de Janeiro de 2008.
- [25] KANN, V., KHANNA, S., LAGERGREN, J., PANCONESI, A., *On the Hardness of Approximating MAX-k-CUT and its Dual*, Technical Report CJTCS-1997-2, Chicago Journal of Theoretical Computer Science, 1997.
- [26] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P., "Optimization by Simulated Annealing", *Science*, v. 220, n. 4598, pp. 671–680, 1983.
- [27] LEIGHTON, F. T. J., "A Graph Coloring Algorithm for Large Scheduling Problems". In: *Journal of Research of the National Bureau of Standards*, v. 84, n. 6, pp. 489–506, 1979.
- [28] LEWANDOWSKI, G., CONDON, A., "Experiments with Parallel Graph Coloring Heuristics and Applications of Graph Coloring". In: *DIMACS Series in Discrete*

Mathematics and Theoretical Computer Science, American Mathematical Society, v. 26, pp. 309–334, 1996.

- [29] LIDEN, S., "Optimum Cruise Profiles in the Presence of Winds". In: *11th Digital Avionics Systems Conference*, pp. 254–261, Seattle, WA, USA, October 1992.
- [30] MARTIN, L., "Simulation and Modelling of Air Traffic over the Western Atlantic Route System". In: *Digital Avionics System, 2001. DASC. The 20th Conference*, v. 2, pp. 6E2/1–6E2/8, Daytona Beach, FL, USA, October 2001.
- [31] MCGOVERN, J., *The Boltzmann Distribution* [online]. Disponível na Internet via WWW. URL: <http://theory.phy.umist.ac.uk/~judith/stat.therm/node66.html>. Consultado em 25 de Março de 2008.
- [32] MENDEL, J. M., "Fuzzy Logic Systems for Engineers: A Tutorial". In: *Proceedings of the IEEE*, v. 83, n. 3, pp. 345–377, March 1995.
- [33] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., TELLER, E., "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics*, v. 21, n. 6, pp. 1087–1092, 1953.
- [34] METROPOLIS, N., ULAM, S., "The Monte Carlo Method". In: *Journal of American Statistical Association*, v. 44, n. 247, pp. 335–341, 1949.
- [35] MITRA, D., ROMEO, F., SANGIOVANNI-VINCENTELLI, A., "Convergence and Finite-time Behavior of Simulated Annealing". In: *24th IEEE Conference on Decision and Control*, v. 24, pp. 761–767, Fort Lauderdale, FL, USA, December 1985.
- [36] NAM, D., PARK, C. H., "Pareto-based Cost Simulated Annealing for Multiobjective Optimization". In: *Proceedings of the 4th Asia-Pacific*

Conference on Simulated Evolution and Learning (SEAL'02), v. 2, Nanyang Technical University, Singapore, November 2002.

- [37] OPTIMAL SPEKTRUM, *O Método de Newton* [online]. Disponível na Internet via WWW. URL: <http://www.simulab.uel.br/spek/pn104.htm>. Consultado em 25 de Março de 2008.
- [38] PIMENTEL, W., *Planejamento de Rotas Aéreas Utilizando Algoritmos Genéticos*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2001.
- [39] PLAUGER, P. J., BRODIE, J., *Standard C* [online]. Disponível na Internet via WWW. URL: <http://www-ccs.ucsd.edu/c/>. Consultado em 18 de Janeiro de 2008.
- [40] PRINTER, R., "Optimal Layer Assignment for Interconnect". In: *Journal VLSI Computer Systems*, v. 1, n. 2, pp. 123–137, 1984.
- [41] SCHILDT, H., *C Completo e Total*. 3ª edição, tradução e revisão técnica de C. Mayer, São Paulo, Makron Books, 1996.
- [42] SKIENA, S., *Vertex Coloring* [online]. Disponível na Internet via WWW. URL: <http://problemsolving.ui.edu/book/AlgDesignManual/BOOK/BOOK4/NODE178.HTM>. Consultado em 25 de Março de 2008.
- [43] SUBBU, R., LIZZI, J., IYER, N., JHA, P. D., SUCHKOV, A., "MONACO - Multi-Objective National Airspace Collaborative Optimization". In: *Aerospace Conference 2007, IEEE*, pp. 1–14, Big Sky, MT, USA, March 2007.
- [44] TRICK, M., *Graph Coloring Instances* [online]. Disponível na Internet via WWW. URL: <http://mat.gsia.cmu.edu/COLOR/instances.html>. Consultado em 07 de Janeiro de 2008.

Apêndice A

Algoritmos Auxiliares

Neste apêndice, são apresentados alguns algoritmos formulados para dar suporte à resolução dos problemas abordados neste trabalho.

A.1 Algoritmos para o Problema 2

A.1.1 Particionamento dos vértices de um nível

Dado que, após uma execução do Resfriamento Simulado é conhecido qual o nível que contém o maior número de conflitos, o que pode ser feito por contagem, procede-se à divisão dos vértices desse nível para dois níveis. Este algoritmo visa a minimizar o número de conflitos dentro de cada novo nível, embora não haja garantia de otimalidade.

Inicialmente, o algoritmo faz uma contagem de conflitos por vértice, ordenando-os em ordem decrescente. Em seguida, para cada vértice, é verificado quantos conflitos o vértice tem com os vértices já posicionados em cada um dos dois novos níveis. Coloca-se o vértice no nível que gere menos conflitos. Em caso de empate, coloca-se o vértice no nível em que há menos vértices.

Este procedimento está esquematizado no algoritmo 2:

Algoritmo 2: Divide Nível

Dados: Grafo é o grafo de rotas;
Instância é a solução a ter o nível dividido;
NivelAntigo é o nível a ser dividido;
NivelNovo é o nível a receber parte dos vértices;

início

para cada *vértice* \in *Instância* **pertencente a** *NivelAntigo* **faça**

inserir vértice em vertices;

fim

ordenar vertices pelo número de conflitos;

 membros[novo] := \emptyset ;

 membros[antigo] := \emptyset ;

para cada *vértice* \in *vertices* **faça**

 conflitos[novo] := contar conflitos de vértice com membros[novo];

 conflitos[antigo] := contar conflitos de vértice com membros[antigo];

se *conflitos[novo]* < *conflitos[antigo]* **então**

inserir vértice **em** membros[novo];

senão se *conflitos[antigo]* < *conflitos[novo]* **então**

inserir vértice **em** membros[antigo];

senão

se *#membros[antigo]* <= *#membros[novo]* **então**

inserir vértice **em** membros[antigo];

fim

senão inserir vértice **em** membros[novo];

fim

fim

para cada *vértice* \in *membros[novo]* **faça**

 mudar nível de vértice para NivelNovo **em** Instância;

fim

fim

A.1.2 Busca do menor nível que elimine cruzamentos

O algoritmo 3 é uma interface para o algoritmo de Resfriamento Simulado (ver algoritmo 1 (p. 12)), efetuando uma busca por quantidade crescente de níveis e efetuando a divisão do nível com mais conflitos, ao se incrementar a quantidade de níveis.

Algoritmo 3: Busca Nível

Dados: configuracoes [global] são as configurações do programa;

f é a função a ser minimizada;

Grafo é o grafo de rotas;

início

MinimoNiveis := 2;

inicial := geraInstancia(Grafo, MinimoNiveis);

resultado := NULO;

repita

se resultado ≠ *NULO* **então**

inicial **recebe** o conteúdo de resultado;

nivel := Nível de inicial com maior número de conflitos;

incrementar inicial → niveis;

divideNivel(Grafo, inicial, nivel, inicial → niveis);

fim

configuracoes → T_0 := calculaT0(Grafo, inicial, configuracoes → P);

resultado := resfriamento(f, Grafo, inicial, configuracoes → T_0);

até $f(\text{resultado}) = 0$;

fim

A.2 Algoritmos para o Problema 3

A.2.1 Criação do grafo de rotas pelo grafo de cruzamentos

Este algoritmo gera o grafo de rotas a ser utilizado na primeira fase do problema 3 a partir do grafo de cruzamentos. Basicamente, isto é feito condensando-se os vértices pertencentes a uma mesma rota em um único vértice. Isso pode ser identificado, já que as arestas tipo 2 estão na forma $4n + 2$, sendo n um número que identifica a rota à qual a aresta se refere. Em seguida, as arestas tipo 1 são adicionadas ao grafo de rotas. O procedimento está esquematizado no algoritmo 4.

Algoritmo 4: Grafo Rotas

Dados: GrafoCruzamentos é o grafo de cruzamentos;

início

para cada aresta (i, j) adjacente tipo 2 **em** GrafoCruzamentos **faça**

 GrafoRotas \rightarrow mapeamento[i] := (i, j) **deslocamento à direita** 2;

 GrafoRotas \rightarrow mapeamento[j] := (i, j) **deslocamento à direita** 2;

fim

$v :=$ maior valor presente na matriz de adjacências de GrafoCruzamentos;

$v := (v$ **deslocamento à direita** 2);

para cada vértice *em* GrafoCruzamentos **faça**

se GrafoRotas \rightarrow mapeamento[*vértice*] **não** está preenchido **então**

incrementar v ;

 GrafoRotas \rightarrow mapeamento[*vértice*] := v ;

fim

fim

para cada aresta (i, j) adjacente tipo 1 **em** GrafoCruzamentos **faça**

inserir aresta (GrafoRotas \rightarrow mapeamento[i],

 GrafoRotas \rightarrow mapeamento[j]) **em** GrafoRotas;

fim

retornar GrafoRotas;

fim

A.2.2 Montagem do grafo com novos pares de cruzamentos

Este algoritmo recebe o grafo de cruzamentos original, sem nenhuma mudança de nível, e uma solução contendo as mudanças de nível a serem efetuadas. A criação do grafo com os novos vértices, correspondentes às mudanças de nível é necessária para a avaliação da função objetivo.

O grafo original, além da matriz de adjacências, mantém uma tabela de direcionamentos para as adjacências de cada vértice. Essa informação estava presente no conjunto de pares de cruzamentos, porém é perdida ao se montar o grafo de cruzamentos. Estes direcionamentos são necessários para se efetuar o correto desmembramento de um vértice, correspondente a um par de cruzamentos que recebeu uma mudança de nível.

O algoritmo 5 inicialmente copia as arestas e direcionamentos do grafo original, colocando os direcionamentos dos novos vértices como vazios inicialmente.

Para cada vértice a ser desmembrado, é verificado qual o vértice adjacente junto ao qual o novo vértice será inserido, dando-se preferência a vértices adjacentes tipo 2. O vértice desmembrado e o novo vértice têm seus direcionamentos atualizados, assim como os vértices antes adjacentes ao vértice desmembrado.

É colocada aresta entre o vértice desmembrado e o novo vértice. Se havia adjacência tipo 2, é colocado o valor da rota correspondente. Se o vértice antes adjacente ao vértice desmembrado era por aresta tipo 1, essa nova aresta recebe o maior valor antes existente na matriz de adjacências acrescido de 4, visto estar-se especificando uma rota diferente das já existentes.

Finalmente, as arestas entre o vértice desmembrado e os vértices contidos no direcionamento repassado ao novo vértice são eliminadas e formadas novas arestas entre o novo vértices e estes vértices contidos no direcionamento.

Algoritmo 5: Monta Grafo

Dados: GrafoOriginal é o grafo sem nenhuma mudança de nível;
Instância é uma solução com as mudanças de nível;

início

novoGrafo := criaGrafoCruzamentos(Instância→numVertices);

copiar arestas de GrafoOriginal **para** novoGrafo;

copiar direcionamentos de GrafoOriginal **para** novoGrafo;

para cada *novo vértice em novoGrafo* **faça**

 novoGrafo→direcionamento[vértice, esquerda] := \emptyset ;

 novoGrafo→direcionamento[vértice, direita] := \emptyset ;

fim

se *existem somente arestas tipo 1* **então** maximaRota := -1;

senão

 maximaRota := maior valor na matriz de adjacências de novoGrafo;

 maximaRota := maximaRota **deslocamento à direita** 2;

fim

novoVertice é o primeiro vértice recém-criado;

para cada *vértice* \in *Instância*→*mudancas* **faça**

se *existir adjacente tipo 2* **então** rota := novoGrafo[vértice, adjacente];

senão se *existir adjacente tipo 1* **então**

 rota := -1;

 seja *d* o direcionamento tal que adjacente \in novoGrafo→direcionamento[vértice, *d*];

 seja *od* o direcionamento oposto a *d*;

 novoGrafo→direcionamento[novoVertice, *od*] := {vértice};

 novoGrafo→direcionamento[novoVertice, *d*] := novoGrafo→direcionamento[vértice, *d*];

 novoGrafo→direcionamento[vértice, *d*] := {novoVertice};

para cada $v \in$ *novoGrafo*→*direcionamento*[*novoVertice*, *d*] **faça**

substituir vértice **por** novoVertice **em** novoGrafo→direcionamento[*v*, *od*];

fim

se *rota = -1* **então**

incrementar maximaRota;

 rota := (maximaRota **deslocamento à esquerda** 2) + 2;

fim

inserir aresta (vértice, novoVertice) := (novoVertice, vértice) := rota;

para cada $v \in$ *novoGrafo*→*direcionamento*[*novoVertice*, *d*] **faça**

em novoGrafo **inserir** arestas (novoVertice, *v*) := (*v*, novoVertice) := (vértice, *v*);

em novoGrafo **remover** arestas (vértice, *v*) e (*v*, vértice);

fim

incrementar novoVertice;

fim

retornar novoGrafo;

fim
