

Universidade Federal de Ouro Preto - UFOP  
Instituto de Ciências Exatas e Biológicas - ICEB  
Departamento de Computação - DECOM

**GERENCIAMENTO DE ESCALA OPERACIONAL  
DOS CONTROLADORES DE TRÁFEGO AÉREO  
DO CINDACTA I**

Aluno: Bruno Mota Avelar Almeida  
Matrícula: 03.1.4175

Orientador: Túlio Ângelo Machado Toffolo

Ouro Preto  
10 de dezembro de 2010

Universidade Federal de Ouro Preto - UFOP  
Instituto de Ciências Exatas e Biológicas - ICEB  
Departamento de Computação - DECOM

**GERENCIAMENTO DE ESCALA OPERACIONAL  
DOS CONTROLADORES DE TRÁFEGO AÉREO  
DO CINDACTA I**

Proposta de monografia apresentada ao curso de Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, como requisito parcial para a conclusão da disciplina Monografia I (BCC390).

Aluno: Bruno Mota Avelar Almeida  
Matrícula: 03.1.4175

Orientador: Túlio Ângelo Machado Toffolo

Ouro Preto  
10 de dezembro de 2010

## **Resumo**

O transporte aéreo tornou-se uma das grandes alavancas do mundo moderno. Imaginar o planeta sem a rapidez e eficiência proporcionada por um avião é uma tarefa completamente irrealizável. O transporte de pessoas pelos céus do planeta se tornou uma premissa básica para o desenvolvimento econômico de todo e qualquer país, pois não há forma mais rápida de se ligar todos continentes do mundo globalizado nos tempos atuais. A principal engrenagem desse complexo sistema é o controlador de tráfego aéreo (CTA), o qual é encarregado de prestação de serviços a quem quer que esteja voando ou que se proponha a voar, fornecendo informações de voo às aeronaves, informações meteorológicas e também fazendo ordenação do fluxo de tráfego aéreo entre os diferentes aeroportos. Este trabalho estuda a escala de serviço do CTA e propõe algoritmos para geração da mesma.

## **Abstract**

The air transportation has become one of the biggest propellers of the modern world. Imagine the planet without quickness and efficiency provided by aircraft is a task completely impossible. People transportation through skies of the planet has become a basic premise for the economic progress of any country, because nowadays this is the fastest way to connect all continents of the globalized world. The main part of this complex system is the air traffic controller (ATC), who is in charge of everyone that is flying or intend to fly, providing them with flying and meteorological information, and also ordering the air traffic flow among different airports. This paper studs the air traffic controller job schedule and proposes algorithms to generate it.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	2
1.2.1	Objetivo geral . . . . .	2
1.2.2	Objetivos específicos . . . . .	2
1.3	Organização do trabalho . . . . .	2
<b>2</b>	<b>Descrição do Problema</b>	<b>3</b>
2.1	Estrutura do Controle de Tráfego Aéreo . . . . .	3
2.2	O Controlador de Tráfego Aéreo . . . . .	4
2.3	A Escala dos CTAs . . . . .	6
2.4	Estrutura do sistema proposto . . . . .	8
<b>3</b>	<b>Revisão Bibliográfica</b>	<b>9</b>
3.1	GRASP . . . . .	9
3.2	ILS . . . . .	11
3.3	Simulated Annealing . . . . .	12
3.4	VND . . . . .	15
<b>4</b>	<b>Conclusão</b>	<b>17</b>
	<b>Referências</b>	<b>18</b>

## Lista de Figuras

1	Modelo de escala. . . . .	8
2	Procedimento GRASP . . . . .	10
3	Procedimento Fase de Construção GRASP . . . . .	11
4	Procedimento ILS . . . . .	12
5	Algoritmo <i>Simulated Annealing</i> . . . . .	15
6	Procedimento VND . . . . .	16

# 1 Introdução

Este capítulo tem por objetivo apresentar a motivação que impulsionou a realização deste trabalho (Seção 1.1), identificar os objetivos que se pretende alcançar ao planejá-lo (Seção 1.2). Por fim, o conteúdo de cada capítulo é descrito sucintamente (Seção 1.3).

## 1.1 Motivação

A maioria dos usuários de transporte aéreo no território brasileiro desconhece a existência de uma grande infra-estrutura de segurança envolvendo aeronaves em decolagem, aterrissagem, manobras nos pátios ou estacionadas. O controlador de tráfego aéreo (CTA) é a ponta deste grande sistema de gerenciamento e controle do espaço aéreo. O CTA é responsável pela aeronave desde seu acionamento do motor até sua total parada no aeroporto de destino. Embora de grande importância, a profissão de controlador de tráfego aéreo não é regulamentada no país e é desconhecida do grande público.

Em 1968, o Sistema de Defesa aérea e Controle de Tráfego Aéreo (SISDACTA) foi criado pelo Ministério da Aeronáutica. O espaço aéreo brasileiro foi dividido em quatro áreas: Dacta I, II, III e Amazônia. Operando no Dacta I está o CINDACTA I, situado em Brasília-DF, capacitado para lidar com 4.000 planos de vôo repetitivos e 2.500 planos de vôo simultâneos. Aproximadamente 45% por cento do total dos voos do país passam pelo espaço aéreo pelo qual o CINDACTA I é responsável. O CINDACTA I foi criado em 1973 com a função de zelar pela segurança do deslocamento das aeronaves no espaço aéreo compreendendo a região da Grande São Paulo, Bauru, Ribeirão Preto, São José do Rio Preto, Cuiabá, Palmas, Brasília, Belo Horizonte, Uberlândia, Uberaba e Vitória. A Força Aérea Brasileira segue normas da Organização Internacional de Aviação Civil (OACI).

O tráfego aéreo brasileiro, bem como o de outros países, tem aumentado consideravelmente nos últimos anos. O CTA deve se adaptar ao volume de tráfego sem perder em qualidade de serviço. Não há espaço para erros humano, uma vez que um pequeno erro do CTA pode provocar um acidente, como aconteceu em 29 de setembro de 2006, quando duas aeronaves se colidiram no ar, ocasionando a morte de 154 pessoas. Assim, a profissão exige agilidade intelectual, velocidade de raciocínio, capacidade de adaptação e principalmente resistência ao estresse.

O CTA carrega nos ombros uma enorme responsabilidade: ele supervisiona uma zona do espaço aéreo, de tamanho variável, denominada setor. O espaço aéreo é cortado por diferentes aerovias, que representam corredores pelos quais as aeronaves se deslocam de um aeroporto a outro. Para todas estas rotas existem regras específicas de navegação, e o CTA deve garantir a separação mínima entre as aeronaves.

## **1.2 Objetivos**

### **1.2.1 Objetivo geral**

O presente trabalho tem por objetivo geral o estudo de técnicas heurísticas para posterior desenvolvimento de um sistema computacional que gere uma escala de alocação dos CTAs e permita maior flexibilidade nos horários de trabalho. Para tanto, serão estudados modelos de otimização baseados em técnicas heurísticas.

### **1.2.2 Objetivos específicos**

- Revisar a literatura sobre as técnicas mais utilizadas para resolução desta classe de problemas.
- Estudar a linguagem de programação C++.
- Estudar algoritmos de otimização para futura implementação.
- Analisar diversas escalas operacionais dos CTAs com o intuito de identificar as principais necessidades dos mesmos.
- Identificar e atender as restrições trabalhistas e operacionais na geração das escalas.

## **1.3 Organização do trabalho**

Este trabalho está organizado como se segue. O Capítulo 2 apresenta a situação atual dos controladores de tráfego aéreo do Brasil e detalha o problema abordado por este trabalho.

O Capítulo 3, por sua vez, apresenta uma revisão bibliográfica de heurísticas que serão implementadas no *software* de gerenciamento de escala. As metodologias GRASP, ILS, *Simulated Annealing* e VND são apresentadas e discutidas.

Por fim, no Capítulo 4 são apresentadas as conclusões a que se chegou mediante a análise das heurísticas e as principais ideias para implementação, bem como algumas considerações finais.

## 2 Descrição do Problema

### 2.1 Estrutura do Controle de Tráfego Aéreo

O controle de tráfego aéreo é um serviço prestado por profissionais que, em terra, orientam e monitoram aeronaves (geralmente, aviões) no ar e no solo, de forma a garantir um fluxo de tráfego seguro, ordenado e expedito. Estes profissionais são os Controladores de Tráfego Aéreo (CTA), e eles são responsáveis por fornecer indicações e autorizações de voo, de acordo com as características operacionais das aeronaves e as condições de tráfego em determinado momento. Estas autorizações podem incidir sobre a rota, altitude e/ou velocidade propostas pelo operador da aeronave, para determinado voo, devendo os pilotos cumprir as instruções e autorizações recebidas.

Em muitos países, os serviços de controle de tráfego aéreo são prestados em toda a extensão do espaço aéreo e estes serviços são utilizados por todos os usuários (aeronaves privadas, militares e comerciais). Os espaços aéreos são divididos em duas classes, a saber:

- Espaço aéreo controlado: é aquele em que o CTA é responsável por prover separação entre as aeronaves.
- Espaço Aéreo não controlado: em oposição ao espaço aéreo controlado, é aquele em que são os próprios pilotos das aeronaves que são os responsáveis por manter a separação entre a sua aeronave e outras.

Dependendo do tipo de voo e da classe do espaço aéreo, o controlador de tráfego aéreo pode emitir instruções que os pilotos devem seguir ou apenas informações de voo para ajudá-los a operar no espaço aéreo. Em todos os casos, entretanto, o piloto tem a responsabilidade final pela segurança da aeronave, e pode sempre optar por não cumprir as instruções do CTA numa emergência.

Os serviços de controle de tráfego aéreo são prestados basicamente por três tipos de órgãos operacionais:

- Torre de Controle de Aeródromo (abreviação TWR, vem da palavra inglesa *Tower*): é uma parte do aeródromo responsável pelo controle de tráfego aéreo nas proximidades deste aeródromo. Costuma ser a estrutura mais alta de um aeroporto, sendo que a sua altura pode variar entre alguns poucos metros a várias dezenas de metros. Ela possui esta altura de acordo com a sua necessidade de visão das áreas cujo controle aéreo ela efetua, seja em terra ou no ar.
- Controle de Aproximação (abreviação APP, vem da expressão em inglês *Approach Control*): exerce o controle do espaço aéreo em um raio de 40

milhas (ou 74,08 km) a partir do seu respectivo aeródromo. Quando uma aeronave se aproxima de um determinado aeródromo, o APP transfere o seu controle para a TWR quando a aeronave está já a poucos minutos de seu pouso. Existem 47 APPs no Brasil.

- Centro de Controle de Área (abreviação ACC, vem da expressão em inglês *Area Control Center*): sua função primordial é a prestação de serviços de supervisão de tráfego aéreo a todas as aeronaves sob sua área de jurisdição (regiões de informação de vôo ou rotas ATS (*Air Transport Service*), mais conhecidas como aerovias superiores ou inferiores) que estejam na fase de vôo de cruzeiro (momento em que a aeronave passa a seguir a rota padrão). O ACC atua em espaços aéreos controlados de áreas maiores do que os do controle de aproximação e a torre de controle, e é responsável ainda pelo espaço aéreo transoceânico. Os centros de controle de área prestam o serviço de vigilância por radar ou de informação de voo. Assim, são os responsáveis por emitir mensagens e coordenar o alerta de busca e salvamento em casos de perda de contato por rádio e radar com a aeronave além do tempo previsto pelo regulamento.

## 2.2 O Controlador de Tráfego Aéreo

O controlador de tráfego aéreo é a pessoa encarregada de separar o tráfego de aeronaves no espaço aéreo e nos aeroportos de modo seguro, ordenado e rápido. Os controladores de tráfego aéreo trabalham emitindo autorizações aos pilotos, ou seja, dando instruções e informações necessárias dentro do espaço aéreo de sua jurisdição com o objetivo de prevenir colisões entre aeronaves e entre aeronaves e obstáculos nas imediações dos aeroportos. O controlador de tráfego aéreo é o elo do Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB) que faz a interface mais próxima a um acidente aeronáutico. Por isso seu trabalho é muito específico.

No Brasil, os controladores de tráfego aéreo são treinados em dois locais: os controladores civis são formados na cidade de São José dos Campos, no DECEA - Departamento de Controle do Espaço Aéreo - e os militares são formados na cidade de Guaratinguetá na EEAR - Escola de Especialistas de Aeronáutica. Ambas as instituições são da área de ensino da Força Aérea Brasileira.

Na maioria dos países signatários da Convenção de Chicago <sup>1</sup> a atividade de controlador de tráfego aéreo é reconhecida como profissão. No Brasil, apesar da grande importância desta função, ainda não há o reconhecimento desta atividade como profissão, até porque ela é ainda desconhecida pela sociedade em geral,

---

<sup>1</sup>Convenção sobre aviação civil em que as regras atuais de tráfego aéreo foram definidas

que só veio tomar conhecimento de sua existência após o acidente envolvendo um Boeing da Gol Linhas Aéreas Inteligentes e um Embraer Legacy em 29 de setembro de 2006.

Esta atividade vem se tornando cada vez mais complexa devido ao crescente número de aeronaves que cruzam o espaço aéreo e ao surgimento de aeronaves cada vez mais modernas e rápidas, as quais voam conjuntamente com outras mais antigas e lentas. Os CTAs necessitam de diversas habilidades para exercer, de maneira eficiente, o seu trabalho. Dentre outras, eles precisam das seguintes habilidades:

- Raciocínio rápido.
- Controle emocional.
- Raciocínio espacial.
- Capacidade de rápida adaptação às mudanças operacionais.
- Capacidade de atuar em grupo.
- Capacidade física e orgânica para atuar seja dia ou noite.

Para se tornar controlador de tráfego aéreo, além das habilidades supracitadas, é necessário adquirir conhecimento, dentre outras, nas seguintes áreas:

- Meteorologia.
- Navegação aérea.
- Geografia (relevo e acidentes geográficos da região).
- Língua inglesa.
- Reconhecimento e desempenho de aeronaves.
- Normas de tráfego aéreo.

O controlador de tráfego aéreo possui enorme grau de responsabilidade. Uma falha pode significar a perda simultânea de centenas de vidas. Dentre os serviços prestados pelo controlador às aeronaves, o serviço de Vetoração Radar é o que confere ao controlador o maior nível de responsabilidade. No ato da prestação deste serviço, o controlador literalmente assume a navegação da aeronave transmitindo instruções de velocidade, proa e altitudes a serem executadas pelo piloto. Além da segurança dos passageiros e tripulantes, a atuação do controlador, seja adequada ou inadequada, pode significar, respectivamente, economia ou prejuízo

para as companhias aéreas e para a aviação geral. O avião deixou de ser um transporte somente de pessoas a passeio e transformou-se em um dos mais importantes meios de transporte. Sabe-se hoje que as crises do setor aéreo podem afetar a vida política, comercial e social de um país. Recentemente (2006), após a crise aérea e os desastres ocorridos, a sociedade tomou consciência da importância do controlador de tráfego aéreo.

Segundo uma pesquisa realizada pelo ISMA-BR (*International Stress Management Association do Brasil*), policiais e seguranças têm a profissão mais estressante do mundo. Em seguida, vem os motoristas de ônibus e os CTAs. Executivos, bancários e atendimento ao público empatam na terceira colocação.

### **2.3 A Escala dos CTAs**

A escala de serviço dos CTAs militares, é dividida atualmente em 2 (dois) tipos:

- Escala de serviço armado: Consiste em serviço de 24 horas de duração cujo objetivo é zelar pela segurança do CINDACTA I. Um CTA, ao assumir o serviço, fica responsável por tudo e todos que entram e que saem da unidade. Ele é, também, responsável pelos seus subordinados, que recebem instruções de manuseio de armamento e trabalho em equipe.
- Escala Operacional: Consiste na própria atividade de controle de tráfego aéreo. Um CTA trabalha atualmente 20 turnos de serviço por mês na escala operacional. Os turnos são divididos em Manhã (M), Tarde (T) e Pernoite (P). O turno da manhã inicia-se às 06:00, o da tarde às 13:30 e o pernoite às 22:30.

A escala do CTA é, atualmente, produzida por um militar, denominado escalante. Este escalante não dispõe de recursos computacionais que o auxiliem na geração da escala. Para produzi-la o escalante divide os turnos entre os CTAs de forma aleatória, até que, ao final da distribuição, cada controlador contabilize 20 turnos, que são dispostos de tal maneira que não infrinjam restrições operacionais. As restrições operacionais são:

1. Mínimo de 12 CTAs no turno da manhã, 13 no da tarde e 9 no pernoite;
2. Máximo de 10 manhãs por mês por CTA;
3. Máximo de 10 tardes por mês por CTA;
4. Máximo de 10 pernoites por mês por CTA;

5. Máximo de 3 pernoites consecutivos por CTA;
6. Máximo de 6 serviços consecutivos sem um dia inteiro de folga por CTA;
7. Máximo de 6 dias consecutivos de folga por CTA;
8. Período mínimo de descanso entre serviços de 1 turno por CTA.

Os itens 5, 6, 7 e 8 são cumulativos para o mês subsequente.

A escala do CTA é mensal, e deve ser disponibilizada 15 dias antes do início do mês. Após a publicação da escala, o CTA tem entre 6 e 8 dias para efetuar as trocas necessárias, que devem ser feitas num formulário de papel e submetidas ao escalante, que verifica a possibilidade da troca de acordo com as restrições operacionais. Caso seja possível, a troca é registrada e no final do dia é impressa uma nova escala contemplando todas as trocas realizadas. Esse ciclo se repete até o último dia de trocas. Após este dia, ficam proibidas toda e qualquer troca, a não ser que haja um motivo muito forte para tal. Os problemas em relação a essa forma de trabalho estão relacionadas a seguir:

- Nem sempre os CTAs estão em serviço nos dias em que são permitidas as trocas. Por isso, muitas vezes o mesmo precisa ir ao local de trabalho para efetivar as trocas.
- Algumas vezes, um CTA, por descuido, lança uma troca erroneamente. Isto implica no cancelamento da troca e gera um efeito cascata, ou seja, trocas que dependiam desta também são canceladas.
- O gasto com ligações para o colega de trabalho é alto. Em média, cada CTA efetua 20 trocas de serviço para atender suas necessidades. São, pelo menos, 60 minutos em ligações todo mês.
- Apesar de todos os esforços, nem sempre todos os CTAs conseguem fazer as trocas que necessitam.

O Centro de Controle de Brasília (ACC BS) é dividido em 3 grandes regiões: Brasília (BR), Rio de Janeiro (RJ) e São Paulo (SP). O objetivo inicial é trabalhar primeiramente com a região de São Paulo, que dispõe atualmente de 53 CTAs. Um modelo de escala é apresentado na Figura 1. Foram exibidos apenas os 15 primeiros dias para facilitar a visualização.

ESCALA DEZEMBRO 2010 CINDACTA I - REGIÃO SP															
	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SO MACHADO	M	T	P			M	T	P			M	T	P		
SO JERONIMO	T	P		T	M	T	P								P
SO DOS SANTOS	P		M	P	P		M	M	T	MP		M	M		P
1S RIVALDO	P			M	T	P			M	P	P			M	T
2S LEÃO	T	P			M	T	E				T	P			T
2S PAMPILIO		T	T	P	P		M	T	P				T	P	
2S NOGUEIRA			T	MP	T	P		T	P			M	P	P	
2S GENÉSIO				T	T					T	P	P		M	M
1S PEDRO LUIZ	T	T		M		M	T	P			M	T	P		
1S SACCILOTTO	T	P		M	M	M	P			M		P		T	M
2S FÁBIO	T	M	T	P			M	T	P			M	T	P	
2S BARSANULFO	T	P		M	M	T	P			M	T	P			
3S CRISTINA	P		P	T	P		T	P	T	P			M	T	P
3S ALINE VEIGA	M	T	M			M	T	M			M	T	T		
3S ANTÔNIO	M	T	P			M	T	P		M	M	T			T
3S DOS REIS	M	T	T	T		M	P	T		M			T		T
3S LUANA	P			T	T	P	P		M	T	P		M	M	P
3S CÁSSIO		M	M	T	MP				T	T			M	P	T
3S KAROLINE	T	P				T	P		M	M	T			M	M
3S ROGÉRIO			T	T	M	T	MP		T			M	T	P	
3S LUTHIANO	M	T	MP		M	M		P			M	T			
3S DE BARROS	T		T	MP	T		M		P			M	T	P	
3S AVELAR			P		T	T	T			P	T	T	P		
3S R. OLIVEIRA						T	M	T	P		T	M		T	T
3S RENATO		M	T	P			M	T	P		P		T	P	
3S COELHO		T	P		T	MP	T	M						T	M
3S MARTINS	T		M		M	T	MP		M	M			T		
3S H COELHO	M	T		P		M	T	P			T	T	P		
3S JULIANA COSTA		M	T	M	T		M	T	M		T		T	M	P
3S ANDREZA	MP	T	T	M		P		T	M	T	P		T	M	T
3S LEILLANE	T				T		T	T	T		M	T	MP		
3S DANIELLE		P	T		MP	T	P			T	P	P		T	P
3S MOURA	P		T	MP	T	P		T	M	T	P			M	M
3S AQSON	P		T		M	M	T	T	P			M	M	T	M

Figura 1: Modelo de escala.

## 2.4 Estrutura do sistema proposto

Como foi dito anteriormente, o objetivo deste trabalho é dar maior flexibilidade ao horário de serviço do CTA. Propõe-se um sistema que utilize técnicas heurísticas de otimização para gerar a escala de serviço mensal. Para isso, além das restrições operacionais, o sistema trabalhará também com “objetivos individuais” dos CTAs. Por exemplo, um CTA não pode ou não deseja trabalhar entre os dias 20 e 25 e não poderá trabalhar no turno da manhã entre os dias 5 e 11 do próximo mês. Para este CTA, além das restrições descritas acima, teremos um conjunto de objetivos individuais que poderão ou não ser atendidos.

Em um modelo de otimização, as restrições operacionais serão modeladas como restrições, enquanto os objetivos individuais deverão estar apenas na função

objetivo, a qual calcula o não atendimento aos objetivos individuais. Assim, o modelo de otimização deve minimizar a função objetivo, desde que atendendo às restrições operacionais.

Os valores dos custos serão definidos na segunda parte desse projeto, e levarão em conta o grau de urgência do citado “objetivo individual”. As técnicas heurísticas que serão implementadas no sistema são abordadas no capítulo seguinte.

### **3 Revisão Bibliográfica**

Metaheurísticas são procedimentos de busca destinados a resolver de forma aproximada um problema de otimização. Elas exploram o espaço de soluções tendo a capacidade de escapar das armadilhas dos ótimos locais. Elas podem ser de busca local ou de busca populacional. Neste trabalho algoritmos de busca local são apresentados.

#### **3.1 GRASP**

A metaheurística GRASP (Greedy Randomized Adaptive Search Procedure) é um algoritmo comumente aplicado a problemas de otimização combinatória. Como diversos métodos construtivos, a aplicação do GRASP consiste em criar uma solução inicial e depois efetuar uma busca local para melhorar a qualidade da solução. Seu diferencial para outros métodos está na geração dessa solução inicial, baseada nas três primeiras iniciais de sua sigla em inglês: gulosa (Greedy), aleatória (Randomized) e adaptativa (Adaptive).

A Figura 2 apresenta o pseudocódigo básico do método GRASP para um problema de minimização.

Enquanto outros algoritmos como a busca tabú e os algoritmos genéticos valem-se de estratégias com grande ênfase na busca local, o GRASP é dito construtivo por privilegiar a geração de uma solução inicial de melhor qualidade para utilizar a busca local apenas para pequenas melhorias.

A estratégia de construção de uma solução no GRASP consiste na definição de um critério de avaliação dos elementos que podem ser inseridos em um conjunto que, ao final do processo, será uma solução para o problema de otimização que se pretende resolver. Esse critério adapta-se à solução já construída, de forma que a valoração dos elementos muda durante a construção da solução. Entretanto, esse critério não é tomado como referência absoluta para a decisão do próximo elemento a ser inserido, havendo uma escolha aleatória entre os melhores elementos a cada iteração.

---

**Function GRASP**

---

**Entrada:** Inteiro  $GRASPmax$ , Função  $f(\cdot)$

**Saída:** Solução  $s^*$  melhor quanto à função  $f$  em  $GRASPmax$  iterações

$f^* \leftarrow \infty$ ;

**para cada** *uma das GRASPmax iterações faça*

    Construa uma solução  $s$  por uma heurística parcialmente gulosa;

    Submeta  $s$  a um procedimento de busca local;

**se**  $f^* < f(s)$  **então**

$s^* \leftarrow s$ ;

$f^* \leftarrow f(s)$ ;

**fim**

**fim**

Retorne  $s^*$ ;

---

Figura 2: Procedimento GRASP

Uma solução para um problema de otimização combinatória é visto no GRASP como um conjunto com elementos que atendam a todas as restrições existentes no problema. Começa-se com um conjunto vazio, e são inseridos elementos nesse conjunto até que ele represente uma solução viável para o problema.

A cada iteração, todos os elementos candidatos são avaliados segundo uma função gulosa que meça o benefício da inserção desse elemento para a construção da solução. A medida desse benefício é dita míope por ser uma avaliação imprecisa de como a inserção de um elemento pode colaborar para a obtenção de uma solução de melhor qualidade, seja em número de elementos necessários, no impacto na função objetivo do problema ou outra métrica qualquer.

Uma vez realizada essa valoração, é construída a RCL (restricted candidate list): uma lista contendo os elementos com melhor valor na função gulosa, podendo seu tamanho ser definido por um parâmetro absoluto como um número de elementos que devam existir na lista, ou por um percentual de tolerância entre o valor do melhor elemento encontrado e o valor de um elemento que pode estar na lista. Dessa lista, escolhe-se um elemento aleatoriamente para ser inserido na solução. Devido a esse misto entre a função gulosa na construção de uma lista e a decisão aleatória sobre qual elemento dessa lista utilizar, diz-se que o GRASP é semi-guloso.

O pseudocódigo da fase de construção é apresentado na Figura 3:

---

**Function** Construção GRASP

---

**Entrada:** Função  $f(.)$

**Saída:** Solução  $s$  construída de forma parcialmente gulosa quanto à função  $f$

$s \leftarrow \emptyset$

**enquanto** a solução  $s$  não estiver totalmente construída **faça**

    Classifique os elementos da solução utilizando um critério guloso ;

    Crie uma RCL, composta pelos melhores elementos

    Selecione aleatoriamente um candidato pertencente à RCL para ser incluído na solução

    Atualize a lista e RCL, considerando a remoção do elemento removido

**fim**

Retorne  $s$ ;

---

Figura 3: Procedimento Fase de Construção GRASP

Ao final de cada iteração, todos os elementos candidatos a inserção são reavaliados em decorrência do último elemento inserido, uma vez que elementos muito parecidos com aquele não serão mais tão interessantes quanto eram antes.

## 3.2 ILS

O método Iterated Local Search (ILS) é baseado na ideia de que um procedimento de busca local pode ser melhorado gerando-se novas soluções de partida, as quais são obtidas por meio de perturbações na solução ótima local.

Para aplicar um algoritmo ILS, quatro componentes tem que ser especificadas:

- Procedimento SolucaoInicial(), que gera uma solução inicial  $s_0$  para o problema;
- Procedimento BuscaLocal( $s'$ ), que retorna uma solução  $s''$  possivelmente melhorada em relação a  $s'$ ;
- Procedimento Perturbacao( $s$ ), que modifica a solução corrente  $s$  guiando a uma solução intermediária  $s'$ ;
- Procedimento CriterioAceitacao(), que decide de qual solução a próxima perturbação será aplicada.

O sucesso do ILS é centrado no conjunto de amostragem de ótimos locais, juntamente com a escolha do método de busca local, das perturbações e do critério de aceitação. Em princípio, qualquer método de busca local pode ser usado, mas

o desempenho do ILS com respeito à qualidade da solução final e a velocidade de convergência depende fortemente do método escolhido. Normalmente um método de descida é usado, mas também é possível aplicar algoritmos mais sofisticados, tais como Busca Tabu ou outras metaheurísticas.

A Figura 4 mostra o pseudocódigo do algoritmo ILS básico.

---

**Function ILS**

---

**Entrada:** Solução  $s$ , Função  $f(\cdot)$   
**Saída:** Solução  $s^*$  de qualidade superior ou igual à  $s$  de acordo com a função  $f$

$s^* \leftarrow \text{BuscaLocal}(s, f);$

**enquanto** *critério de parada não satisfeito* **faça**

$s' \leftarrow \text{Perturbação}(\text{histórico}, s^*);$

$s'' \leftarrow \text{BuscaLocal}(s', f);$

$s^* \leftarrow \text{CritérioAceitação}(\text{histórico}, s', s'', s^*);$

**fim**

Retorne  $s^*$ ;

---

Figura 4: Procedimento ILS

### 3.3 Simulated Annealing

Arrefecimento simulado ou simulated annealing é uma metaheurística para otimização que consiste numa técnica de busca local probabilística, e se fundamenta numa analogia com a termodinâmica.

Esta metaheurística é uma metáfora de um processo térmico, dito annealing ou recozimento, utilizado em metalurgia para obtenção de estados de baixa energia num sólido. O processo consiste de duas etapas: na primeira a temperatura do sólido é aumentada para um valor máximo no qual ele se funde; na segunda o resfriamento deve ser realizado lentamente até que o material se solidifique, sendo acompanhado e controlado esse arrefecimento. Nesta segunda fase, executada lentamente, os átomos que compõem o material organizam-se numa estrutura uniforme com energia mínima. Isto provoca que os átomos desse material ganhem energia para se movimentarem livremente e, ao arrefecer de forma controlada, dar-lhes uma melhor hipótese de se organizarem numa configuração com menor energia interna, para ter, como resultado prático, uma redução dos defeitos do material.

De forma análoga, o algoritmo de arrefecimento simulado substitui a solução atual por uma solução próxima (i.e., na sua vizinhança no espaço de soluções),

escolhida de acordo com uma função objetivo e com uma variável  $T$  (dita Temperatura, por analogia). Quanto maior for  $T$ , maior a componente aleatória que será incluída na próxima solução escolhida. À medida que o algoritmo progride, o valor de  $T$  é decrementado, começando o algoritmo a converter para uma solução ótima, necessariamente local.

Uma das principais vantagens deste algoritmo é permitir testar soluções mais distantes da solução atual e dar mais independência do ponto inicial da pesquisa.

Esta técnica começa sua busca a partir de uma solução inicial qualquer, o procedimento principal consiste em um loop ou laço que gera aleatoriamente, em cada iteração, um único vizinho  $s'$  da solução corrente  $s$ . A cada geração de um novo vizinho  $s'$  de  $s$ , é testada a variação  $\Delta$  do valor da função objetivo, isto é,  $\Delta = f(s') - f(s)$ , onde temos as seguintes situações:

- $\Delta < 0$ : Há uma redução de energia, a qual implica que a nova solução é melhor que a anterior. O método aceita a solução e  $s$  passa a ser a nova solução corrente;
- $\Delta = 0$ : Caso de estabilidade, não havendo redução de energia. Na verdade, situação pouco provável de acontecer na prática. A aceitação da solução é, portanto, indiferente;
- $\Delta > 0$ : Houve um aumento do estado de energia. A aceitação desse tipo de solução é mais provável a altas temperaturas e bastante improvável a temperaturas reduzidas.

Para reproduzir essas características, geralmente usa-se, para calcular a probabilidade de se aceitar a nova solução, uma função conhecida por fator de Boltzmann, que é dada por  $e^{(-\Delta/T)}$ , onde  $T$  é um parâmetro do método, chamado de temperatura e que regula a probabilidade de soluções com pior custo. Por exemplo, esta poderá ser:

- Gera-se um número aleatório retirado de uma distribuição uniforme no intervalo  $[0, 1]$ ;
- Se este número for menor ou igual a  $p$ , aceita-se a solução.
- Se for maior que  $p$ , rejeita-se a solução.

A temperatura  $T$  assume inicialmente um valor elevado,  $T_0$ . Após um número fixo de iterações (o qual representa o número de iterações para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento  $\alpha$ , tal que  $T_n \leftarrow \alpha \times T_{n-1}$ , sendo  $0 < \alpha < 1$ . Como esse procedimento se dá no início, há uma chance maior de se

escapar de mínimos locais e, à medida que  $T$  se aproxima de zero, o algoritmo se comporta como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos que possa piorar ( $T \leftarrow 0 \Rightarrow e^{-\Delta/T} \leftarrow 0$ ).

O procedimento é finalizado quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da melhor solução seja mais aceita, ou seja, quando o sistema estiver estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local.

Algoritmos baseados em Simulated Annealing geralmente incluem reaquecimento seguido de um novo processo de resfriamento, utilizado quando a quantidade de movimentos consecutivamente rejeitados é alta. É também comum trabalhar nas temperaturas mais altas com taxa de resfriamento menor e aumentá-la quando a temperatura reduzir.

Apresenta-se, pela Figura 5, o algoritmo Simulated Annealing básico aplicado a um problema de minimização.

```

procedimento SA( $f(\cdot), N(\cdot), \alpha, SAmax, T_0, s$ )
1  $s^* \leftarrow s$ ;           {Melhor solução obtida até então}
2  $IterT \leftarrow 0$ ;       {Número de iterações na temperatura T}
3  $T \leftarrow T_0$ ;        {Temperatura corrente}
4 enquanto ( $T > 0$ ) faça
5   enquanto ( $IterT < SAmax$ ) faça
6      $IterT \leftarrow IterT + 1$ ;
7     Gere um vizinho qualquer  $s' \in N(s)$ ;
8      $\Delta = f(s') - f(s)$ ;
9     se ( $\Delta < 0$ )
10      então
11         $s \leftarrow s'$ ;
12        se ( $f(s') < f(s^*)$ ) então  $s^* \leftarrow s'$ ;
13      senão
14        Tome  $x \in [0, 1]$ ;
15        se ( $x < e^{-\Delta/T}$ ) então  $s \leftarrow s'$ ;
16      fim-se;
17    fim-enquanto;
18     $T \leftarrow \alpha \times T$ ;
19     $IterT \leftarrow 0$ ;
20 fim-enquanto;
21  $s \leftarrow s^*$ ;
22 Retorne  $s$ ;
fim SA;

```

Figura 5: Algoritmo *Simulated Annealing*

### 3.4 VND

O Método de Descida em Vizinhança Variável (Variable Neighborhood Descent, VND), proposto por Mladenovic e Hansen [12], é um método de refinamento que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada. O pseudocódigo deste método, em que se considera o refinamento de uma solução  $s$  utilizando uma função de avaliação  $f$ , a ser minimizada, e um conjunto de  $r$  diferentes vizinhanças  $N = \{N^{(1)}, N^{(2)}, \dots, N^{(r)}\}$ , é apresentado pela Figura 6.

---

**Function VND**

---

**Entrada:** Solução  $s$ , Vizinhança  $N(\cdot)$ , Inteiro  $r$ , Função  $f(\cdot)$

**Saída:** Solução  $s^*$  de qualidade superior ou igual à  $s$  de acordo com a função  $f$

Seja  $r$  o número de diferentes estruturas de vizinhança;

$k \leftarrow 1$  {Tipo de estrutura de vizinhança corrente};

**enquanto**  $k < r$  **faça**

    Encontre o melhor vizinho  $s' \in N^k(s)$ ;

**se**  $s'$  for melhor que  $s$  de acordo com a função  $f$  **então**

$s \leftarrow s'$

$k \leftarrow 1$

**senão**

$k \leftarrow k + 1$

**fim**

**fim**

$s^* \leftarrow s$

Retorne  $s^*$ ;

---

Figura 6: Procedimento VND

Dependendo do problema abordado, a busca pelo melhor vizinho pode ser cara computacionalmente. Nesta situação é comum fazer a busca pela primeira solução de melhora. Outra alternativa, bastante utilizada, é aplicar explorar a vizinhança de forma aleatória, ou seja, executar a busca em ordem aleatória, ao invés de executá-la de forma sequencial.

O método VND baseia-se em três princípios básicos:

- Um ótimo local com relação a uma dada estrutura de vizinhança não corresponde necessariamente a um ótimo local com relação a uma outra estrutura de vizinhança;
- Um ótimo global corresponde a um ótimo local para todas as estruturas de vizinhança;
- Para muitos problemas, ótimos locais com relação a uma ou mais estruturas de vizinhança são relativamente próximas.

Ainda de acordo com Mladenovic e Hansen, o último princípio, de natureza empírica, indica que um ótimo local frequentemente fornece algum tipo de informação sobre o ótimo global. Este é o caso em que os ótimos local e global compartilham muitas variáveis com o mesmo valor, o que sugere uma investigação sistemática da vizinhança de um ótimo local até a obtenção de uma nova solução de melhor valor.

## 4 Conclusão

Conclui-se que, devido a importância que tem o controlador de tráfego aéreo na segurança de toda e qualquer aeronave que sobrevoa o espaço aéreo brasileiro, o sistema a ser desenvolvido será de grande utilidade. Os benefícios obtidos serão:

- Maior facilidade em adaptar o horário de serviço do CTA.
- Maior índice de satisfação no trabalho, que resulta numa melhor qualidade de controle de tráfego aéreo.
- Possível redução na carga de trabalho do CTA.
- Redução de custos em ligações telefônicas;
- Redução de consumo de papel, já que atualmente todas as trocas são feitas em formulários.
- Eliminação de erros em trocas, que são comuns e causam enormes inconvenientes.

## Referências

- [1] A. A. Andreatta, S. E. R. Carvalho, and C. C. Ribeiro. An object-oriented framework for local search heuristics. In *TOOLS '98: Proceedings of the Technology of Object-Oriented Languages and Systems*, page 33, Washington, DC, USA, 1998. IEEE Computer Society.
- [2] K.A. Dowsland. Simulated Annealing. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, Advanced Topics in Computer Science Series, chapter 2, pages 20–69. Blackwell Scientific Publications, London, 1993.
- [3] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- [4] A. Fink and S. Voß. Hotframe: a heuristic optimization framework. In S. Voß and D. L. Woodruff, editors, *Optimization Software Class Libraries*, pages 81–154. Kluwer Academic Publishers, Boston, 2002.
- [5] L. Di Gaspero and A. Schaerf. Easylocal++: an object-oriented framework for the flexible design of local-search algorithms. *Softw. Pract. Exper.*, 33(8):733–765, 2003.
- [6] F. Glover. Future paths for Integer Programming and links to Artificial Intelligence. *Computers and Operations Research*, 5:553–549, 1986.
- [7] A. Hertz and D. de Werra. The tabu search metaheuristic: how we used it. *Annals of Mathematics and Artificial Intelligence*, 1:111–121, 1990.
- [8] A. Hertz and M. Widmer. Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 151:247–252, 2003.
- [9] S. Kirkpatrick, D.C. Gellat, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.
- [10] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 11, pages 321–353. Kluwer Academic Publishers, 2003.
- [11] A. Mendes, P. França, and P. Moscato. Np-opt: an optimization framework for np problems. In *Proceedings of IV SIMPOI/POMS 2001. Agosto 11-14, 2001.*, Guarujá/SP, 2001.

- [12] N. Mladenovic and P. Hansen. A variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.
- [13] J. V. Moccellini, M. O. Santos, and M. S. Nagano. Um método heurístico Busca Tabu-Simulated Annealing para Flowshops Permutacionais . In *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*, pages 1088–1095, 2001.
- [14] A. Schaefer. Tabu search techniques for large high-school timetabling problems. In *Proceedings of the 30th National Conference on Artificial Intelligence*, pages 363–368, 1996.
- [15] M. J. F. Souza. Inteligência computacional para otimização. Notas de aula. Disponível em < [http : //www.decom.ufop.br/prof/marcone/ Disciplinas/InteligenciaComputacional/InteligenciaComputacional.htm](http://www.decom.ufop.br/prof/marcone/ Disciplinas/InteligenciaComputacional/InteligenciaComputacional.htm) >. Acesso em: 13 nov. 2010, 2010.
- [16] J.R.A. Torreão. Inteligência Computacional. Notas de aula, Universidade Federal Fluminense, Niterói, 2004, 2004.