

BRUNO MOTA AVELAR ALMEIDA

Orientador: Túlio Ângelo Machado Toffolo
Co-orientador: Marcone Jamilson Freitas Souza

**UM ALGORITMO HEURÍSTICO PARA O PROBLEMA DE
GERENCIAMENTO DE ESCALA OPERACIONAL DE
CONTROLADORES DE TRÁFEGO AÉREO**

Ouro Preto
Dezembro de 2011

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**UM ALGORITMO HEURÍSTICO PARA O PROBLEMA DE
GERENCIAMENTO DE ESCALA OPERACIONAL DE
CONTROLADORES DE TRÁFEGO AÉREO**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

BRUNO MOTA AVELAR ALMEIDA

Ouro Preto
Dezembro de 2011



UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

UM ALGORITMO HEURÍSTICO PARA O PROBLEMA DE
GERENCIAMENTO DE ESCALA OPERACIONAL DE
CONTROLADORES DE TRÁFEGO AÉREO

BRUNO MOTA AVELAR ALMEIDA

Monografia defendida e aprovada pela banca examinadora constituída por:

Prof. TÚLIO ÂNGELO MACHADO TOFFOLO – Orientador
Universidade Federal de Ouro Preto

Prof. MARCONE JAMILSON FREITAS SOUZA – Co-orientador
Universidade Federal de Ouro Preto

Prof. SABIR RIBAS
Universidade Federal de Ouro Preto

Ouro Preto, Dezembro de 2011

Resumo

Este trabalho apresenta um algoritmo heurístico para resolução de um problema de timetabling, o Problema de Gerenciamento de Escalas Operacionais de Controladores de Tráfego Aéreo (PGECTA). O algoritmo proposto combina as metaheurísticas *Multi-Start* e *General Variable Neighborhood Search* (GVNS). Do procedimento Multi-Start utilizou-se a fase de construção para produzir soluções viáveis e de boa qualidade rapidamente. O GVNS foi escolhido devido a sua simplicidade, eficiência e capacidade natural de sua busca local, feita pelo método *Variable Neighborhood Descent*, para lidar com diferentes vizinhanças. Resultados computacionais mostram a efetividade do algoritmo proposto.

Palavras-chave: *Timetabling*, Escalonamento dos Controladores de Tráfego Aéreo, Meta-heurísticas

Abstract

This work presents a heuristic algorithm to solve a timetabling problem, the Problem of Managing Operational Scales of Air Traffic Controllers. (PMSATC). The proposed algorithm combines two metaheuristics, Multi-Start and General Variable Neighborhood Search (GVNS). Multi-Start procedure is used in the construction phase to produce feasible and high quality solutions quickly. The GVNS was chosen because of its simplicity, efficiency and capacity of its natural local search, made by the Variable Neighborhood Descent method, to deal with different neighborhoods. Computational results show the effectiveness of the proposal.

Keywords: Timetabling, Air Traffic Controller Rostering, Metaheuristics

À minha família, aos meus amigos.

Agradecimentos

A Deus.

Aos meus pais João Luiz e Marlene, pelo amor incondicional, pelo eterno orgulho ao longo da caminhada, pelo apoio, compreensão, ajuda, e, em especial, por todo carinho.

Ao meu irmão Tiago, pelo seu exemplo de determinação, pela amizade, compreensão e pela grande ajuda.

Aos meus grandes e eternos amigos Bruno Nazário e Carol, pela oportunidade de fazer parte de suas vidas.

Aos meus amigos da República Tarja Preta, pelos grandes momentos vividos ao longo desta caminhada.

Aos meus amigos controladores de tráfego aéreo, minha segunda família, pela motivação e pela amizade.

Ao meu amigo Vítor Nazário, pela amizade e pela grande ajuda prestada nesta monografia.

Aos professores Túlio e Marcone, pela orientação deste trabalho.

Aos amigos João Machado e Gessi, pela amizade e hospitalidade.

A todos amigos de Ouro Preto, pela amizade.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	2
1.2.1	Objetivo geral	2
1.2.2	Objetivos específicos	2
1.3	Estrutura do Trabalho	2
2	Revisão Bibliográfica	4
2.1	Introdução	4
2.2	Heurísticas	4
2.3	Metaheurísticas	5
2.3.1	<i>Multi-Start</i>	5
2.3.2	Busca Geral em Vizinhança Variável	7
2.3.3	Descida em Vizinhança Variável	7
3	Descrição do Problema Abordado	9
3.1	O Problema de Gerenciamento de Escala dos Controladores de Tráfego Aéreo	9
4	Metodologia	12
4.1	Representação de uma solução	12
4.2	Estruturas de Vizinhança	13
4.2.1	Movimento Adiciona e Remove CTA - $N^{AR}(s)$	13
4.2.2	Movimento Troca CTA - $N^T(s)$	14
4.3	Avaliação de uma solução	14
4.4	Geração de uma solução inicial	15
4.5	Algoritmo Proposto	17
4.6	<i>Framework</i> OptFrame	18
5	Experimentos	20
6	Conclusões e Trabalhos Futuros	24

Referências Bibliográficas	25
Anexos	26

Lista de Figuras

3.1	Exemplo de uma escala de trabalho retirada da internet	11
4.1	Exemplo de Solução para o PGECTA	12
4.2	Exemplo dos vetores de pertinência	13
4.3	Exemplo aplicação do movimento $N^{AR}(s)$	14
4.4	Exemplo aplicação do movimento $N^T(s)$	14
5.1	Instância r55_1	22
6.1	Instância v53_1	28
6.2	Instância r55_1	29
6.3	Instância r55_2	30
6.4	Instância v53_1	31
6.5	Instância v51_1	32

Lista de Tabelas

5.1	Característica das Instâncias	20
5.2	Resultados <i>MSGVNS</i>	21
5.3	Características das melhores soluções	23

Lista de Algoritmos

2.1	<i>Multi-Start</i>	6
2.2	ConstroiSolucaoGulosa	6
2.3	<i>GVNS</i>	7
2.4	<i>Troca Vizinhaça</i>	7
2.5	Descida em Vizinhaça Variável	8
4.1	ConstróiSoluçãoInicial	15
4.2	GereRankMelhoresCTAs	16
4.3	<i>MSGVNS</i>	17
4.4	Refinamento	18
4.5	VND	18

Capítulo 1

Introdução

O presente trabalho trata do Problema de Gerenciamento de Escalas Operacionais de Controladores de Tráfego Aéreo (PGECTA). Neste problema, deve-se determinar uma escala operacional válida para os Controladores de Tráfego Aéreo (*Air Traffic Controller*) atendendo as restrições trabalhistas e operacionais e, se possível, as restrições pessoais.

1.1 Motivação

A maioria dos usuários de transporte aéreo no território brasileiro desconhece a existência de uma grande infraestrutura de segurança envolvendo aeronaves em decolagem, aterrissagem, manobras nos pátios ou estacionadas. O controlador de tráfego aéreo (CTA) é a chave deste grande sistema de gerenciamento e controle do espaço aéreo. O CTA é responsável pela aeronave desde o acionamento do motor até a 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.

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 humanos, uma vez que um pequeno erro do CTA pode provocar um acidente, como aquele acontecido em 29 de setembro de 2006, http://pt.wikipedia.org/wiki/Voo_Gol_1907, 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 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.

Para uma boa execução de suas atividades, é importante que os CTAs tenham uma escala de trabalho equilibrada ao longo do mês, que respeite todas as normas trabalhistas e que, se

possível, atendam às requisições pessoais. Como o PGECTA é uma variante de um problema de *timetabling*, é um problema da classe NP-difícil (Cabral et al., 2000). Como tal, ainda não existem algoritmos que o resolvam em tempo polinomial; sendo portanto indicados para sua resolução procedimentos heurísticos.

1.2 Objetivos

1.2.1 Objetivo geral

- Desenvolver um algoritmo heurístico para gerar soluções viáveis e de boa qualidade para o Problema de Gerenciamento de Escala dos Controladores de Tráfego Aéreo (PGECTA)

1.2.2 Objetivos específicos

- Revisar a literatura sobre técnicas mais utilizadas para busca de soluções de problemas de escalonamento pessoal;
- Estudar a linguagem de programação C++;
- Gerar um algoritmo que permita a geração automática da escala operacional, vistas as restrições operacionais e individuais dos controladores de tráfego aéreo;
- Reduzir, quando possível, o número de dias de trabalho do controlador de tráfego aéreo.
- Utilizar, no mínimo dois algoritmos de otimização e comparar os resultados.
- Contribuir com a formação de recursos humanos especializados nessa área do conhecimento.
- Contribuir para a consolidação da linha de pesquisa “Otimização Combinatória” do grupo de Logística e Pesquisa Operacional da UFOP.
- Se possível, produzir um artigo a ser submetido a um periódico nacional.

1.3 Estrutura do Trabalho

O presente trabalho está dividido em cinco capítulos, incluindo esta introdução.

No Capítulo 2 é apresentada uma revisão bibliográfica sobre os diversos métodos utilizados na resolução de problemas de *timetabling*, além de uma breve revisão sobre os métodos utilizados neste trabalho. Já no Capítulo 3 é apresentada a definição do Problema de Gerenciamento de Escala dos Controladores de Tráfego Aéreo (PGECTA).

No Capítulo 4 é apresentado o algoritmo proposto, denominado MSGVNS, para resolver o PGECTA. Para detalhar esse algoritmo, o qual é baseado nas metaheurísticas *Multi-Start*

e *General Variable Neighborhood Search*, é mostrada como uma solução é representada, como são geradas as soluções iniciais, as estruturas de vizinhança utilizadas, a função de avaliação e a estrutura geral do algoritmo proposto.

No Capítulo 5 são apresentados e analisados os resultados computacionais e no Capítulo 6 são apresentadas as conclusões e apontadas as sugestões para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

É comum encontrarmos na literatura trabalhos da área de Pesquisa Operacional relacionados à resolução de problemas como: escalonamento de horários (Mariano, 2007; Freitas, 2007), escalonamento de salas de aulas (Silva et al., 2005), escalonamento de motoristas (Méllo et al., 2006), escalonamento de enfermeiros (Camilo e Stelle, 2008), entre outros.

A literatura tem mostrado que algoritmos heurísticos têm resolvido com eficiência vários problemas *timetabling* (Loo et al., 1986; D. e de Werra, 1997; Santiago-Mozos et al., 2005; Lü e Hao, 2010). De modo geral, cada um desses trabalhos propõe uma solução boa, quando não ótima, para o problema abordado. Essa solução visa ao atendimento de todas as restrições, maximizando o aproveitamento de recursos envolvidos no problema, além de tentar atender ao máximo as preferências das pessoas envolvidas. A diferença entre esses problemas reside, sobretudo, nas regras trabalhistas existentes e em algumas restrições operacionais.

2.2 Heurísticas

As heurísticas são técnicas que visam a obtenção de soluções de boa qualidade em um tempo computacional aceitável. Essas técnicas, no entanto, não garantem a obtenção da solução ótima para o problema nem são capazes de garantir o quão próximo a solução obtida está da ótima.

As heurísticas podem ser construtivas ou de refinamento. As construtivas têm por objetivo construir uma solução, usualmente, elemento a elemento. A escolha de cada elemento está, geralmente, relacionada a uma determinada função que o avalia de acordo com sua contribuição para a solução. Tal função é bastante relativa, pois varia conforme o tipo de problema abordado.

As heurísticas de refinamento, também chamadas de mecanismos de busca local, são técnicas baseadas na noção de vizinhança. Para definirmos o que é uma vizinhança, seja S o

espaço de busca de um problema de otimização e f a função objetivo a minimizar. O conjunto $N(s) \subseteq S$, o qual depende da estrutura do problema tratado, reúne um número determinado de soluções s' , denominado vizinhança de s . Cada solução $s' \in N(s)$ é chamada de vizinho de s e é obtida a partir de uma operação chamada de movimento.

Em linhas gerais, esses métodos partem de uma solução inicial s_0 , percorrem o espaço de busca por meio de movimentos, passando de uma solução para outra que seja sua vizinha.

2.3 Metaheurísticas

Metaheurísticas são procedimentos de busca local destinados a resolver aproximadamente um problema de otimização, tendo a capacidade de escapar das armadilhas dos ótimos locais, ainda distantes de um ótimo global. Elas podem ser de busca local ou populacional. Na primeira, a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança. Já na segunda, trabalha-se com um conjunto de soluções, re combinando-as com o intuito de aprimorá-las.

Neste trabalho, foi desenvolvido um método inspirado nas metaheurísticas *Multi-Start* - MS (Martí, 2003) e Busca Geral em Vizinhança Variável (GVNS), que utiliza como busca local o método Descida em Vizinhança Variável (VND), descritos nas Seções 2.3.1 2.3.2 e 2.3.3, respectivamente.

2.3.1 *Multi-Start*

A metaheurística *Multi-Start* consiste em fazer amostragens do espaço de soluções, aplicando a cada solução gerada um procedimento de refinamento. As amostras são obtidas por meio da geração de soluções aleatórias. Com este procedimento, há uma diversificação no espaço de busca, possibilitando escapar dos ótimos locais. A grande vantagem do método é que ele é de fácil implementação. No Algoritmo 2.1 apresenta-se o pseudocódigo de um procedimento *Multi-Start* básico para um problema de minimização. Um número máximo de iterações ou um tempo máximo de processamento é normalmente utilizado como critério de parada.

Algoritmo 2.1: *Multi-Start*

Entrada: Função $f(\cdot)$, $N(\cdot)$

```

1  $f^* \leftarrow \infty$ 
2 enquanto Critério de parada não atendido faça
3    $s \leftarrow \text{ConstroiSolucaoGulosa}()$ 
4    $s \leftarrow \text{BuscaLocal}()$ 
5   se  $f(s) < f(s^*)$  então
6      $s^* \leftarrow s$ 
7      $f^* \leftarrow f(s)$ 
8   fim
9 fim
10  $s \leftarrow s^*$ 
11 retorna  $s$ 

```

Uma variação comum no procedimento *Multi-Start* consiste em partir de uma solução inicial gerada por um procedimento construtivo guloso. Assim, no Algoritmo 2.1 cria-se uma linha 0 e substitui-se a linha 1, tal como se segue:

0 – $s^* \leftarrow \text{ConstroiSolucaoGulosa}()$; Melhor solução até então 1 – $f \leftarrow f(s^*)$; Valor associado a s^*

O procedimento $\text{ConstroiSolucaoGulosa}()$ encontra-se descrito no Algoritmo 2.2.

Algoritmo 2.2: $\text{ConstroiSolucaoGulosa}$

Entrada: Função $g(\cdot)$

```

1  $s \leftarrow \emptyset$ 
2 Inicialize o conjunto  $C$  de elementos candidatos
3 enquanto  $C \neq \emptyset$  faça
4    $g(t_{\text{melhor}}) = \text{melhor}\{g(t) \mid t \in C\}$ 
5    $s \leftarrow s \cup \{t_{\text{melhor}}\}$ 
6   Atualize o conjunto  $C$  de elementos candidatos
7 fim
8 retorna  $s$ 

```

O Algoritmo 2.2 mostra a construção de uma solução inicial para um problema de otimização que utiliza uma função gulosa $g(\cdot)$. Neste Algoritmo, t_{melhor} indica o membro do conjunto de elementos candidatos com o valor mais favorável da função de avaliação g , isto é, aquele que possui o menor valor de g no caso de o problema ser de minimização ou o maior valor de g no caso de o problema ser de maximização.

2.3.2 Busca Geral em Vizinhaça Variável

A Busca Geral em Vizinhaça Variável, ou *General Variable Neighborhood Search* (GVNS), é uma metaheurística inspirada na ideia de buscar boas soluções por meio de estruturas de vizinhaça diferentes. Referências ao GVNS podem ser encontradas em Hansen et al. (2008b), Hansen et al. (2008a), Hansen e Mladenovic (2001), Mladenovic e Hansen (1997). A metaheurística GVNS se baseia no fato de que um *ótimo global* é, consequentemente, um *ótimo local* para todas as possíveis vizinhanças de uma solução.

O Algoritmo 2.3, que faz uso do método Troca Vizinhaça (Algoritmo 2.4), mostra o pseudocódigo do GVNS.

Algoritmo 2.3: *GVNS*

Entrada: Função $f(\cdot)$, $kMax$, $tMax$, Nível p

Saída: Solução s^* de qualidade possivelmente superior à s de acordo com a função f

```

1 enquanto  $k = kMax$  faça
2    $k \leftarrow 1$ 
3   enquanto  $t < tMax$  faça
4      $s' \leftarrow \text{Perturbação}(s, k)$ 
5      $s'' \leftarrow \text{VND}(s', f)$ 
6     TrocaVizinhaça( $f(\cdot), s, s'', k$ )
7   fim
8    $t \leftarrow \text{TempoCPU}$ 
9 fim
10 retorna  $s$ 
```

Algoritmo 2.4: *Troca Vizinhaça*

Entrada: Função $f(\cdot)$

```

1 se  $f(s'') < f(s)$  então
2    $s \leftarrow s''$ 
3    $k \leftarrow 1$ 
4 fim
5 senão
6    $k \leftarrow k + 1$ 
7 fim
```

2.3.3 Descida em Vizinhaça Variável

A Descida em Vizinhaça Variável (Hansen e Mladenovic, 2001), conhecida na literatura inglesa como *Variable Neighborhood Descent* - VND, é uma metaheurística que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhaça.

Seja s a solução corrente e N a vizinhança de uma solução estruturada em r vizinhanças distintas, isto é, $N = N^{(1)} \cup N^{(2)} \cup \dots \cup N^{(r)}$. O VND inicia-se analisando a primeira estrutura de vizinhança $N^{(1)}$. A cada iteração, o método gera o melhor vizinho s' da solução corrente s na vizinhança $N^{(k)}$. Caso s' seja melhor que s , então s' passa a ser a nova solução corrente e retorna-se à vizinhança $N^{(1)}$. Caso contrário, passa-se para a próxima estrutura de vizinhança $N^{(k+1)}$. O método termina quando não é possível encontrar uma solução $s' \in N^{(r)}$ melhor que a solução corrente.

O Algoritmo 2.5 mostra o pseudocódigo do VND.

Algoritmo 2.5: Descida em Vizinhança Variável

Entrada: Indivíduo s e Função de Avaliação f

Entrada: r vizinhanças distintas

Saída: Indivíduo s^* de qualidade possivelmente superior à s de acordo com a função f

```

1  $k \leftarrow 1$ 
2 enquanto  $k \leq r$  faça
3   Encontre o melhor vizinho  $s' \in N^{(k)}(s)$ 
4   se  $f(s') < f(s)$  então
5      $s \leftarrow s'$ ;  $k \leftarrow 1$ 
6   fim
7   senão
8      $k \leftarrow k + 1$ 
9   fim
10 fim
11 retorna  $s$ 
```

Capítulo 3

Descrição do Problema Abordado

3.1 O Problema de Gerenciamento de Escala dos Controladores de Tráfego Aéreo

O Problema de Gerenciamento de Escala dos Controladores de Tráfego Aéreo (PGECTA) tem algumas características desses problemas apontados no Capítulo 2, mas difere em algumas, como as descritas a seguir.

Cada CTA deve contabilizar ao máximo 20 turnos ao final de cada mês, sendo que as restrições operacionais não devem ser infringidas. As restrições operacionais são:

- Mínimo de 12 CTAs no turno da manhã, 13 no da tarde e 9 no pernoite;
- Máximo de 10 manhãs por mês por CTA;
- Máximo de 10 tardes por mês por CTA;
- Máximo de 10 pernoites por mês por CTA;
- Máximo de 3 pernoites consecutivos por CTA;
- Máximo de 6 serviços consecutivos sem um dia inteiro de folga por CTA;
- Máximo de 6 dias consecutivos de folga por CTA;
- Período mínimo de descanso entre serviços de 1 turno por CTA.

Além disso, o problema torna-se mais complexo no fato de o mesmo ter que atender, além das restrições operacionais, restrições como preferências de horários por parte dos controladores.

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 anteriormente, teremos um conjunto de objetivos individuais que poderão ser ou não atendidos.

Vale ressaltar que, este trabalho não tem como objetivo diminuir a escala de trabalho dos controladores, ou seja, reduzir o número médio de turnos trabalhos por CTA, ao invés disso, o trabalho tem como foco, gerar uma escala operacional válida e que atenda, ao máximo, todas as restrições individuais.

A Figura 3.1 mostra uma escala de trabalho, retirada da internet, para um mês com 31 dias de trabalho.

	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S	D	S	T	Q	Q	S	S
CTA1	1	T	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CTA2		M	T	MP			M	T	P			M	T		MP	T	M	P	P	M	M	P	P	T		M	T	P	T	T	M
CTA3		T		I	T		M	T		M	T	P				M	T	P	P	M	M	MP	P	T	T	T	M	I	P	T	MP
CTA4		P				P			M	T	P			T	T	MP		M	T	P	P		T	MP	T	P	M	M	M	M	T
CTA5			MP		P			M	T	P			M	MP	P			MP	T	P	P		T	T	P	M	P		T	P	P
CTA6					P								M	MP	P				T	P	P		M	T	P	T	M		M	T	P
CTA7		T		P		MP			M	T				M	T	T		MP	T	P	P		T	T	T	P	T	T	T	T	P
CTA8		MP			MP	P		P			M	P		P	T	T		M	M	T	T		M	T	T	P	T	T	T	T	M
CTA9		T	M		T			MP		T	M				T	M	T	MP		P		T	P		M	T	T	P	T	T	M
CTA10		MP	T			T			M		T	MP	T			M	M	P				T	T		P	T	T	T	T	M	M
CTA11		T	MP											MP		MP					T					MP	MP	T	T	MP	MP
CTA12			P	T		T					T	MP		P	M	T															
CTA13		T	M	T		M								T	T	T															
CTA14		T	M	T	P		M	MP	T	M							M														
CTA15			T	T	T	M		T	M	T																					
CTA16		T		T		T	M	T		P																					
CTA17		P		MP	T		M	T																							
CTA18		M	MP	T	M				T																						
CTA19		T		MP					T	P	T		MP																		
CTA20		M		MP	T																										
CTA21		M	T			T																									
CTA22		T			M	M	M																								
CTA23			T	T		M																									
CTA24		P		T	T																										
CTA25		T				MP	T		MP	T																					
CTA26					P				M	T																					
CTA27		M	MP						M	M	T																				
CTA28		T							M	M	T																				

Figura 3.1: Exemplo de uma escala de trabalho retirada da internet

Capítulo 4

Metodologia

Neste capítulo é apresentada a metodologia proposta para resolver o PGECTA. Na Seção 4.1 mostra-se como uma solução do problema é representada, enquanto na Seção 4.2 são apresentados os movimentos utilizados para explorar o espaço de soluções do problema. Na Seção 4.3 mostra-se como uma solução é avaliada. Os métodos de geração de uma solução inicial são apresentados na Seção 4.4. Na Seção 4.5 é descrito o algoritmo proposto para resolver o PGECTA, o qual é inspirado nas metaheurísticas Multi-Start e Busca Geral em Vizinhança Variável (GVNS). Por fim, na seção 4.6 são apresentados os principais aspectos do *framework* OptFrame, que auxiliou no desenvolvimento deste trabalho.

4.1 Representação de uma solução

Uma solução é representada por uma matriz $E_{|T| \times |D|}$, em que:

- $|T| = 3$, são os três turnos possíveis: manhã, tarde, pernoite.
- $|D|$, são o número de dias do mês em que o escalonamento está sendo executado.

A Figura 4.1 exemplifica uma possível solução para o PGECTA em um mês hipotético de 4 dias.

	<i>Dia1</i>	<i>Dia2</i>	<i>Dia3</i>	<i>Dia4</i>
<i>Manha</i>	$e_{0,1}$	$e_{0,2}$	$e_{0,3}$	$e_{0,4}$
<i>Tarde</i>	$e_{1,1}$	$e_{1,2}$	$e_{1,3}$	$e_{1,4}$
<i>Pernoite</i>	$e_{2,1}$	$e_{2,2}$	$e_{2,3}$	$e_{2,4}$

Figura 4.1: Exemplo de Solução para o PGECTA

Cada célula $e_{i,j}$ na matriz da Figura 4.1 é um vetor de pertinência de tamanho igual ao número de controladores $nCTA$, isto é:

$$e_{i,j} = [cta_0, cta_1, cta_2, \dots, cta_{nCTA}]$$

Cada índice deste vetor pode ter o valor 0 ou 1, sendo que, o valor 1 indica que o controlador está trabalhando no turno i do dia j , já o valor 0 indica que o CTA não está trabalhando no turno i do dia j .

Desconsiderando-se as restrições operacionais, podemos exemplificar parte de uma possível escala dos controladores para três turnos do mês hipotético exposto na solução da Figura 4.1 por meio dos vetores apresentados na Figura 4.2.

$$e_{0,1} = [0, 1, 1, 0, 1]$$

$$e_{1,3} = [1, 1, 1, 1, 1]$$

$$e_{2,4} = [1, 0, 0, 0, 1]$$

Figura 4.2: Exemplo dos vetores de pertinência

Percebe-se que, no primeiro vetor do exemplo os CTAs 2, 3 e 5 estão trabalhando no turno da manhã do dia 1. Para o segundo vetor, notamos que todos os CTAs estão trabalhando no turno da tarde do dia 3. Por último, os CTAs 1 e 5 estão trabalhando no turno da noite do dia 4.

4.2 Estruturas de Vizinhaça

Para explorar o espaço de soluções do problema foram propostas duas estruturas de vizinhaça, $N^{AR}(\cdot)$ e $N^T(\cdot)$, as quais usam dois tipos de movimento a seguir descritos.

4.2.1 Movimento Adiciona e Remove CTA - $N^{AR}(s)$

Movimento Adiciona e Remove CTA - $N^{AR}(s)$: Este movimento consiste em dar folga a um dado CTA em um turno i do dia j e colocar um outro CTA para trabalhar no lugar deste. Desta maneira, neste movimento uma célula do vetor $e_{i,j}$ muda seu valor de 1 para 0, enquanto uma outra célula deste mesmo vetor mudar seu valor de 0 para 1.

Na Figura 4.3 têm-se um exemplo de aplicação do movimento $N^{AR}(s)$, considerando a solução exemplo da Figura 4.1.

Na Figura 4.3, nota-se que o CTA 1 deixou trabalhar no turno da noite do dia 4, sendo substituído pelo CTA 3.

$$\begin{aligned} e_{inicial(2,4)} &= [1, 0, 0, 0, 1] \\ e_{final(2,4)} &= [0, 0, 1, 0, 1] \end{aligned}$$

Figura 4.3: Exemplo aplicação do movimento $N^{AR}(s)$

4.2.2 Movimento Troca CTA - $N^T(s)$

Movimento Troca CTA - $N^T(s)$: Consiste em trocar duas células distintas, ou seja, diferentemente do movimento $N^{AR}(s)$, neste movimento podemos selecionar um CTA de um vetor $e_{i,j}$ e outro CTA de um vetor $e_{n,m}$, desde que se os dias forem iguais ($m == j$), os turnos devem ser diferentes ($n \neq i$).

$$\begin{aligned} e_{inicial(1,2)} &= [1, 0, 0, 1, 1] \\ e_{inicial(2,3)} &= [1, 0, 1, 0, 1] \\ e_{final(1,2)} &= [1, 0, 1, 0, 1] \\ e_{final(2,3)} &= [1, 0, 0, 1, 1] \end{aligned}$$

Figura 4.4: Exemplo aplicação do movimento $N^T(s)$

Na Figura 4.4, nota-se que o CTA 4 deixou trabalhar no turno da tarde do dia 2, sendo substituído pelo CTA 3. Além disso, o CTA 4 passou a trabalhar no turno da noite do dia 3, substituindo o CTA 3.

4.3 Avaliação de uma solução

No modelo heurístico adotado, uma solução s é avaliada por uma função f , a ser minimizada, dada pela Eq. (4.1). Esta função é baseada em penalidades, sendo composta de duas parcelas: f^{obj} e f^{inv} .

$$f(s) = f^{obj}(s) + f^{inv}(s) \quad (4.1)$$

A parcela f^{obj} , dada pela Eq. (4.2), refere-se à função objetivo do problema, sendo: CTA o conjunto de controladores; D o conjunto de dias trabalhados; T o conjunto de turnos por dia trabalhado; $x_{c,t,d}$ uma matriz binária que vale 1 se o controlador c trabalha no turno t do dia d , e 0, caso contrário; e $P_{c,t,d}$ refere-se ao valor positivo da *não preferência* do controlador c em trabalhar no turno t do dia d ; caso este controlador não tenha objeção em trabalhar no turno t do dia d , o valor será 0.

$$f^{obj}(s) = \sum_{c \in CTA} P_{c,t,d} \times x_{c,t,d} \quad \forall t \in T; d \in D \quad (4.2)$$

Já a segunda parcela f^{inv} , dada pela Eq. (4.3), refere-se às restrições operacionais do problema, descritas no Capítulo 3. Caso alguma dessas restrições seja violada, é atribuído um valor alto à solução s , pois esta solução é inviável na prática.

$$f^{inv}(s) = \sum_{c \in CTA} \left(f_c^T(s) + f_c^{T_{M,T,N}}(s) + f_c^{PC}(s) + f_c^{SF}(s) + f_c^F(s) + f_c^{MD}(s) \right) \quad (4.3)$$

Na Eq. (4.3), a parcela $f_c^T(s)$ avalia s quanto à inviabilidade do CTA c trabalhar mais que 20 turnos totais, enquanto que a parcela $f_c^{T_{M,T,N}}(s)$ avalia o CTA c trabalhar mais que 10 turnos da manhã, da tarde ou da noite; $f_c^{PC}(s)$ avalia s quanto ao desrespeito do CTA c trabalhar mais que três pernoites consecutivos; $f_c^{SF}(s)$ avalia s quanto ao fato de o CTA c trabalhar mais que seis dias consecutivos sem folga; $f_c^F(s)$ avalia s quanto à situação de o CTA c folgar mais que seis dias consecutivos e $f_c^{MD}(s)$ avalia s quanto ao desrespeito ao fato de o CTA c trabalhar dois turnos consecutivos.

4.4 Geração de uma solução inicial

Para geração de uma solução inicial para o problema utiliza-se uma heurística gulosa, cujo pseudocódigo está descrito no Algoritmo 4.1.

Algoritmo 4.1: ConstróiSoluçãoInicial

Entrada: Conjunto de dias D a serem alocados

Saída: Solução de inicial S

```

1  enquanto  $D \neq \emptyset$  faça
2      Escolha um dia aleatório  $d \in D$ 
3      Remova o dia  $d$  do conjunto  $D$ 
4       $t \leftarrow$  manhã
5       $Rank_M \leftarrow$  GereRankMelhoresCTAs( $t, d$ )
6       $S(t, d) \leftarrow Rank_M$ 
7       $t \leftarrow$  tarde
8       $Rank_T \leftarrow$  GereRankMelhoresCTAs( $t, d$ )
9       $S(t, d) \leftarrow Rank_T$ 
10      $t \leftarrow$  noite
11      $Rank_N \leftarrow$  GereRankMelhoresCTAs( $t, d$ )
12      $S(t, d) \leftarrow Rank_N$ 
13 fim
14 Retorne  $S$ 
```

Na linha 2 do Algoritmo 4.1 escolhe-se aleatoriamente um dia d para realizar a alocação dos

CTAs. Nas linhas 6, 9 e 12 insere-se na solução inicial S do problema os *melhores* CTAs (escolhidos pelo procedimento GereRankMelhoresCTAs, descrito no Algoritmo 4.2) para trabalhar no dia d nos turnos da manhã, tarde e noite, respectivamente.

Algoritmo 4.2: GereRankMelhoresCTAs

Entrada: Turno t , Dia d , Número de CTAs disponíveis $nCTA$
Saída: Vetor v dos *melhores* CTAs para trabalhar no turno t do dia d

```

1   $Rank[.] \leftarrow 0$ 
2  para  $c \leftarrow 1$  até  $nCTA$  faça
3       $Rank[c] \leftarrow foPenalizacao(c, t, d)$ 
4       $Rank[c] \leftarrow Rank[c] + foDescanso(c, t, d)$ 
5       $Rank[c] \leftarrow Rank[c] + foMaxDiasSemFolga(c, t, d)$ 
6       $Rank[c] \leftarrow Rank[c] + foMaxDiasFolga(c, t, d)$ 
7       $Rank[c] \leftarrow Rank[c] + foMaxTurnoFixoTrab(c, t, d)$ 
8       $Rank[c] \leftarrow Rank[c] + foMaxTurnosTrab(c, t, d)$ 
9       $Rank[c] \leftarrow Rank[c] + foMaxNoitesConsecutivas(c, t, d)$ 
10      $Rank[c] \leftarrow Rank[c] + foMinTurnosTrab(c, t, d)$ 
11 fim
12  $v \leftarrow$  CTAs (necessários para atender o Turno  $t$ ) com menor valor referente ao vetor  $Rank$ 
13 Retorne  $v$ 
```

As penalizações das linhas 3, 4, 5, 7, 8 e 9 do Algoritmo 4.2 são positivas; já as penalizações das linhas 6 e 10 são negativas.

Na linha 3 do Algoritmo 4.2, o CTA c é penalizado caso ele não possa ou não queira trabalhar do turno t do dia d . O valor dessa penalização é apresentado na seção de resultados. Na linha 4, o CTA c é penalizado caso ele já esteja alocado para trabalhar no turno anterior ao turno t do dia d . A linha 5 penaliza o CTA c por trabalhar mais que seis dias seguidos sem folga antes do dia d . Na linha 6 o CTA c é penalizado caso ele esteja folgando os seis dias anteriores ao dia d . Esta penalização deve ser negativa pois o CTA deve ser alocado o mais rápido possível. A linha 7 penaliza o CTA c caso ele já tenha trabalhado 10 turnos t no mês da alocação, já a linha 8 penaliza o CTA c caso ele já esteja trabalhando 20 turnos totais no mês. A linha 9 penaliza o CTA c caso ele esteja trabalhando os últimos três turnos da noite anteriores ao dia d . Por último, a linha 10 bonifica (isto é, atribui uma penalização negativa) cada CTA c por cada turno trabalhado. Esta bonificação faz com que os CTAs que estejam trabalhando menos sejam alocados; desta forma, a solução tende a distribuir o número de turnos trabalhados igualmente entre os controladores.

Uma solução inicial para o problema é feita aplicando *MultiStartmax* vezes a fase de construção do procedimento *Multi-Start* (Procedimento *ConstróiSoluçãoInicial()*) descrito no Algoritmo 4.1) e retorna a melhor das soluções construídas.

4.5 Algoritmo Proposto

O algoritmo proposto, denominado *MSGVNS*, combina os procedimentos MS e GVNS, descritos na seção 2.3.2. Do procedimento *Multi-Start* utilizou-se a fase de construção para produzir soluções viáveis e de boa qualidade rapidamente. O GVNS foi escolhido devido a sua simplicidade, eficiência e capacidade natural de sua busca local, feita pelo método *Variable Neighborhood Descent* – VND, para lidar com diferentes vizinhanças, como relatado em Souza et al. (2010).

O pseudocódigo do algoritmo *MSGVNS* está esquematizado no Algoritmo 4.3. Neste algoritmo, *MultiStartmax* representa a quantidade de iterações em que a fase de construção do procedimento *Multi-Start* é aplicada e *IterMax* indica o número máximo de iterações realizadas em um dado nível de perturbação.

Algoritmo 4.3: *MSGVNS*

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

```

1   $s_0 \leftarrow s$ 
2   $s_0 \leftarrow$  Melhor Solução em MultiStartmax iterações do procedimento ConstróiSoluçãoInicial()
3   $s^* \leftarrow \text{VND}(s_0, f)$ 
4   $p \leftarrow 0$ 
5  enquanto critério de parada não satisfeito faça
6       $iter \leftarrow 0$ 
7      enquanto  $iter < \text{IterMax}$  e critério de parada não satisfeito faça
8           $s' \leftarrow \text{Refinamento}(s^*, p, f)$ 
9          se  $s'$  for melhor que  $s^*$  de acordo com a função  $f$  então
10              $s^* \leftarrow s'$ ;
11              $p \leftarrow 0$ ;
12              $iter \leftarrow 0$ 
13         fim
14         senão
15              $iter \leftarrow iter + 1$ 
16         fim
17     fim
18      $p \leftarrow p + 1$ 
19 fim
20 retorna  $s^*$ 

```

A solução inicial s_0 (linha 2 do Algoritmo 4.3) é gerada aplicando-se *MultiStartmax* a fase de construção do procedimento *Multi-Start*, isto é, o procedimento *ConstróiSoluçãoInicial* descrito na seção 4.4.

A busca local é feita pelo procedimento VND, Algoritmo 4.5, usando-se os dois movimentos descritos na seção 4.2, ou seja, os movimentos das vizinhanças N^{AR} , N^T . Para a perturbação, esses dois movimentos são aplicados de forma aleatória de acordo com o procedimento *SelecionaVizinhança* (linha 2 do Algoritmo 4.4).

Algoritmo 4.4: Refinamento

Entrada: r vizinhanças: N^{AR}, N^T
Entrada: Solução Inicial s , Nível p e Função de Avaliação f
Saída: Solução s

```

1 para  $i \leftarrow 1$  até  $p + 2$  faça
2    $k \leftarrow \text{SelecioneVizinhança}(r)$ 
3    $s' \leftarrow \text{Perturbação}(s, k)$ 
4 fim
5  $s \leftarrow \text{VND}(s', f)$ 
6 retorna  $s$ 

```

Algoritmo 4.5: VND

Entrada: r vizinhanças na ordem aleatória: N^{AR}, N^T
Entrada: Solução Inicial s e Função de Avaliação f
Saída: Solução s

```

1  $k \leftarrow 1$ 
2 enquanto  $k \leq r$  faça
3   Encontre o melhor vizinho  $s' \in N^{(k)}(s)$ 
4   se  $f(s') < f(s)$  então
5      $s \leftarrow s'$ ;
6      $k \leftarrow 1$ 
7   fim
8   senão
9      $k \leftarrow k + 1$ 
10  fim
11 fim
12 retorna  $s$ 

```

4.6 *Framework* OptFrame

O algoritmo foi implementando com auxílio do *framework* OptFrame¹ (Coelho et al., 2011, 2010). Essa escolha, foi devido a seu arcabouço de fácil utilização, que inclui:

- Representações de soluções e populações;
- Arcabouços de métodos heurísticos e metaheurísticos;
- Interface para análise dos resultados;
- Fórum de discussão com os colaboradores do projeto.

O *framework* OptFrame é, basicamente, uma estrutura computacional que agiliza o desenvolvimento de algoritmos heurísticos. O objetivo do *framework* é fornecer uma simples

¹disponível em <http://sourceforge.net/projects/optframe/>

interface em C++ para componentes comuns de metaheurísticas baseadas em trajetória e população, aplicadas a problemas de otimização combinatória. Uma vez que muitos métodos são comuns na literatura, o OptFrame fornece uma implementação eficiente para versões simples de algumas heurísticas e metaheurísticas, todavia, o usuário pode desenvolver versões "mais inteligentes" e mais robustas, aplicadas ao seu problema específico. Além disso, o OptFrame possui suporte ao desenvolvimento de sistemas em paralelo, tanto para memória compartilhada quanto para memória distribuída. OptFrame tem sido aplicado com sucesso para modelar e resolver vários problemas combinatórios, mostrando um bom equilíbrio entre a flexibilidade e eficiência.

Capítulo 5

Experimentos

O algoritmo proposto *MSGVNS* foi implementado em C++ usando o *framework* de otimização OptFrame e compilado pelo g++ 4.0. Os testes foram executados em um microcomputador Pentium Core 2 Quad(Q6600), 2.4 GHZ e 8 GB de RAM.

Para testar o algoritmo proposto, foi usado um conjunto de duas instâncias, denominadas de r55_1 e r55_2, do PGECTA associadas à operação de um centro de controladores de tráfego aéreo brasileiro, considerando dados disponibilizados na Internet. Além disso, foram geradas duas instâncias, v53_1 e v51_1. A Tabela 5.1 mostra as características de cada uma delas.

Tabela 5.1: Característica das Instâncias

Instância	$nCTA$	$nDias$	$nManha$	$nTarde$	$nPernoite$	$nRestricoesCTA$
r55_1	55	30	12	13	9	43,85
r55_2	55	30	12	13	9	45,72
v53_1	53	30	12	13	9	48
v51_1	51	30	12	13	9	50

Na Tabela 5.1, a coluna “Instância” indica o problema-teste. A coluna $nCTA$ indica o número de controladores disponíveis para alocação, $nDias$ indica o número de dias do mês de alocação. As colunas $nManha$, $nTarde$ e $nPernoite$ indicam, respectivamente, o número de CTA’s necessários para os turnos da manhã, tarde e pernoite, para o mês em questão. Já a $nRestricoesCTA$ indica a média de restrições por CTA, ou seja, o número total de restrições pessoais dividido por $nCTA$.

As instâncias geradas neste trabalho possuem um menor número de CTA’s disponíveis e um maior número de restrições por CTA se comparadas as instâncias reais, ou seja, foram geradas instâncias mais difíceis, de forma a testar com uma maior precisão a eficiência o algoritmo proposto.

Foram utilizados os seguintes parâmetros. Para a fase de construção da solução

inicial: $foDescanso = 5000$, $foMaxDiasSemFolga = 5000$, $foMaxDiasFolga = -200$, $foMaxTurnoFixoTrab = 5000$, $foMaxTurnosTrab = 5000$, $foMaxNoitesConsecutivas = 5000$ e $foMinTurnosTrab = 200$. Para a função objetivo os pesos de inviabilidade das parcelas $f_c^T(s)$, $f_c^{T_{M,T,N}}(s)$, $f_c^{PC}(s)$, $f_c^{SF}(s)$ e $f_c^{MD}(s)$ foram 10000 cada uma. Por último, os parâmetros do *Multi-Start* e do GVNS foram: $MultiStartmax = 300000$, $IterMax = 50$.

Durante a implementação do algoritmo e alguns testes empíricos preliminares, os parâmetros descritos acima foram calibrados. É notório que o valor das inviabilidades $f_c^{T_{M,T,N}}(s)$, $f_c^{PC}(s)$, $f_c^{SF}(s)$, $f_c^F(s)$ e $f_c^{MD}(s)$ são altos se comparados às restrições pessoais, como exemplo, vide figura 5.1, esses valores são de fácil calibragem e um valor maior não tem grande influência no algoritmo. Deste modo, caso uma das leis trabalhistas não seja atendida, a solução é facilmente descartada. Para os parâmetros do procedimento GereRankMelhoresCTAs (Algoritmo 4.2, descrito no Capítulo 4), $foDescanso$, $foMaxDiasSemFolga$, $foMaxDiasFolga$, $foMaxTurnoFixoTrab$, $foMaxTurnosTrab$, $foMaxNoitesConsecutivas$ e $foMinTurnosTrab$, foi realizada uma bateria de experimentos junto a um controlador de tráfego aéreo, este CTA analisou as soluções obtidas e calibrou os parâmetros de forma a obter uma solução mais adequada à realidade. Para o parâmetro $MultiStartmax$ foi dado um valor alto, sendo que valores maiores que este não tiveram uma influência significativa na solução final do procedimento *Multi-Start*, além de consumirem um alto tempo computacional. Por fim, o parâmetro $IterMax$ é típico da literatura, como relatado em Souza et al. (2010).

Na Figura 5.1, a coluna “Pesos” indica a penalização na função objetivo, para cada restrição pessoal que não for atendida do CTA em questão. Já a coluna “Restrições” indica as restrições pessoais de cada CTA, por exemplo, o CTA 6 não possui nenhuma restrição pessoal, já o CTA 22 não deseja trabalhar em nenhum dos turnos entre os dias 22 a 27.

Foram realizadas 100 execuções para cada uma das instâncias com tempo limite de 15s.

A Tabela 5.2 mostra os resultados obtidos. Nessa tabela, a coluna “Instância” indica o problema-teste utilizado. A coluna “Melhor” indica o melhor valor encontrado pelo Algoritmo *MSGVNS*. Já a coluna “Média” indica a média do *MSGVNS* em 100 execuções.

Tabela 5.2: Resultados *MSGVNS*

Instância	Tempo (s)	MSGVNS	
		Melhor	Média
r55_1	15	0	4,47
r55_2	15	0	10,85
v53_1	15	176	195,99
v51_1	15	463	501,13

Como podemos observar na Tabela 5.2, o algoritmo *MSGVNS* foi capaz de gerar soluções de boa qualidade e baixa variabilidade. Percebe-se que o algoritmo foi capaz de encontrar soluções competitivas em todas as instâncias, encontrando a solução ótima para as instâncias r55_1 e r55_2, já que a penalização total foi nula. Já para as instâncias v53_1 e v51_1, nota-

CTA	Pesos	Restrições
0	275	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
1	270	1M1T1N2M2T2N3M3T3N4M4T4N5M5T5N6M6T6N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N30M30T30N
2	265	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
3	260	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
4	255	1M2M3M4M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
5	250	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
6	245	-
7	240	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
8	235	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
9	230	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
10	225	-
11	220	-
12	215	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
13	210	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
14	205	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
15	200	-
16	195	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
17	190	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
18	185	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
19	180	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
20	175	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
21	170	-
22	165	22M22T22N23M23T23N24M24T24N25M25T25N26M26T26N27M27T27N
23	160	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
24	155	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
25	150	1M4M4T5M5T6M6T7M7M8M11M11T12M12T13M13T14M15M18M18T19M19T20M20T20N21M22M25M25T26M26T27M27T27N28M29M
26	145	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
27	140	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
28	135	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
29	130	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
30	125	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
31	120	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
32	115	5M5T5N6M6T6N7M7T7N8M8T8N9M9T9N10M10T10N
33	110	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
34	105	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
35	100	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
36	95	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
37	90	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
38	85	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
39	80	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
40	75	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
41	70	-
42	65	5M5T5N6M6T6N7M7T7N8M8T8N15M15T15N16M16T16N17M17T17N18M18T18N
43	60	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
44	55	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
45	50	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
46	45	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
47	40	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
48	35	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
49	30	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
50	25	10M10T10N11M11T11N12M12T12N13M13T13N14M14T14N15M15T15N
51	20	12M12T12N13M13T13N14M14T14N15M15T15N16M16T16N17M17T17N
52	15	17M17T17N18M18T18N19M19T19N20M20T20N21M21T21N22M22T22N
53	10	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
54	5	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M

Figura 5.1: Instância r55_1

se que o algoritmo *MSGVNS* conseguiu obter soluções factíveis e satisfazendo as restrições pessoais de 95% dos controladores.

A Tabela 5.3 mostra as características da melhor solução obtida em cada uma das quatro instâncias utilizadas. Nessa tabela, a coluna *nMaxTurnos* indica o número máximo de turnos que um CTA *c* teve que trabalhar durante o mês, a coluna *nMinTurnos* indica o mínimo de turnos trabalhados. A coluna *nRestricoesVioladas* mensura o número total de restrições pessoais violadas.

Tabela 5.3: Características das melhores soluções

Instância	<i>nMaxTurnos</i>	<i>nMinTurnos</i>	<i>nRestricoesVioladas</i>
r55_1	19	18	0
r55_2	19	18	0
v53_1	20	19	4
v51_1	20	20	10

Analisando a Tabela 5.3, é notório que, na melhor solução das instâncias r55_1 e r55_2, os CTAs trabalhavam no máximo 19 turnos por mês e no mínimo 18 turnos, além disso, em ambas as soluções nenhum CTA teve suas restrições não atendidas, ou seja, todas as 2412 restrições da instância r55_1 e todas as 2515 restrições da instância r55_2 foram atendidas. Já para a instância v53_1, foi relatado um máximo de 20 turnos, um mínimo de 19 turnos e apenas 4 das 2640 restrições pessoais não foram atendidas. Por fim, para a instância v51_1, todos os CTAs trabalhavam 20 turnos e das 2550 restrições pessoais, apenas 10 não foram atendidas.

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho teve seu foco no Problema de Gerenciamento de Escalas Operacionais dos Controladores de Tráfego Aéreo (PGECTA). Em virtude de sua dificuldade de solução, foi proposto um algoritmo heurístico, denominado *MSGVNS*, que combina os procedimentos *Multi-Start* e *GVNS*.

Utilizando dois problemas-teste envolvendo dados reais e dois gerados neste trabalho, o algoritmo foi capaz de encontrar soluções de boa qualidade rapidamente e com baixa variabilidade das soluções finais. Além disso, o algoritmo proposto conseguiu provar a otimalidade de duas soluções.

A aplicação em questão mostra-se de extrema importância para a manutenção do sistema aéreo brasileiro, visto que em muitos órgãos de controle essa alocação era feita por um militar, denominado escalante. Como as restrições consideradas pelo escalante eram apenas as operacionais, as trocas de serviços tornavam-se muito onerosas, e na maioria das vezes, ineficientes. Os resultados obtidos são aplicáveis e resultariam em uma maior satisfação dos controladores de tráfego aéreo.

Para trabalhos futuros é proposta a concessão de novos tipos de perturbações e ampliação da busca local, realizada pelo procedimento VND. Propõe-se a criação de um conjunto mais amplo de problemas-testes, contribuição de suma importância para a literatura. Sugere-se, também, implementação de um modelo matemático para o problema, buscando, além de uma comparação entre o algoritmo proposto *MSGVNS* e um *solver* comercial, uma implementação de um algoritmo híbrido, que combinaria a flexibilidade das metaheurísticas com o poderio da programação matemática.

Referências Bibliográficas

- Cabral, L. A. F.; Pontes, R. C. V.; Souza, M. J. F. e Maculan, N. (2000). An heuristic approach for large scale crew scheduling problems at rio sul airlines. In *Proceedings of the 40th International Symposium of the AGIFORS*, pp. 1–16, Istambul.
- Camilo, C. e Stelle, D. (2008). Aplicando algoritmos genéticos ao problema de definição de escala de trabalho do corpo de enfermagem de um hospital universitário. In *XL Simpósio Brasileiro de Pesquisa Operacional*, pp. 1216–1224, João Pessoa, PB.
- Coelho, I.; Munhoz, P. L. A.; Haddad, M. N.; Coelho, V. N.; Silva, M. M.; Souza, M. J. F. e Ochi, L. S. (2011). A computational framework for combinatorial optimization problems. In *VII ALIO/EURO Workshop on Applied Combinatorial Optimization*, pp. 51–54, Porto.
- Coelho, I.; Ribas, S.; Perche, M. H. P.; Munhoz, P. L. A.; Souza, M. F. e Ochi, L. S. (2010). Optframe: a computational framework for combinatorial optimization problems. In *XLIII Simpósio Brasileiro de Pesquisa Operacional*, pp. 1–12, Bento Gonçalves, RS.
- D. e de Werra (1997). The combinatorics of timetabling. *European Journal of Operational Research*, 96(3):504 – 513.
- Freitas, C. C. (2007). Uma ferramenta baseada em algoritmos geneticos para a geracao de tabela de horário escolar. In *Sétima Escola Reginal de Computação BAHIA-SERGIPE*, Vitória da Conquista/BA.
- Hansen, P. e Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467.
- Hansen, P.; Mladenovic, N. e Pérez, J. A. M. (2008a). Variable neighborhood search. *European Journal of Operational Research*, 191:593–595.
- Hansen, P.; Mladenovic, N. e Pérez, J. A. M. (2008b). Variable neighborhood search: methods and applications. *4OR: Quarterly journal of the Belgian, French and Italian operations research societies*, 6:319–360.
- Loo, E.; Goh, T. e Ong, H. (1986). A heuristic approach to scheduling university timetables. *Computers & Education*, 10(3):379 – 388.

- Lü, Z. e Hao, J.-K. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235 – 244.
- Mariano, A. S. (2007). Geração automática de grade horária para a faculdade de engenharia elétrica da ufu usando algoritmo genéticos. Mestrado em engenharia elétrica, DEE/UFU, Uberlândia.
- Martí, R. (2003). Multi-start methods. In Glover, F. e Kochenberger, G., editores, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pp. 355–368. Springer New York.
- Mladenovic, N. e Hansen, P. (1997). A variable neighborhood search. *Computers and Operations Research*, 24:1097–1100.
- Méllo, F. G. A.; Senne, E. L. F. e Lorena, L. A. N. (2006). Uma abordagem para o problema de escalonamento de motoristas. In *XXVI ENEGEP*, pp. 1606–1617, Fortaleza, CE.
- Santiago-Mozos, R.; Salcedo-Sanz, S.; DePrado-Cumplido, M. e Bousoño-Calzón, C. (2005). A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a spanish university. *Computers & Operations Research*, 32(7):1761 – 1776.
- Silva, A. S. N.; Sampaio, R. M. e Alvarenga, G. B. (2005). Uma aplicação de simulated annealing para o problema de alocação de salas. *Journal of Computer Science - InfoComp*, 4:59–66.
- Souza, M. J. F.; Coelho, I. M.; Ribas, S.; Santos, H. G. e Merschmann, L. H. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research, EJOR*, 207:1041–1051.

Anexos

Instâncias Utilizadas

As figuras 6.2, 6.3, 6.4 e 6.5 apresentam, respectivamente, as instâncias r55_1, r55_2, v53_1 e v51_1.

CTA	Pesos	Restrições
0	275	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
1	270	1M1T1N2M2T2N3M3T3N4M4T4N5M5T5N6M6T6N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N30M30T30N
2	265	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
3	260	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
4	255	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
5	250	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
6	245	-
7	240	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
8	235	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
9	230	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
10	225	-
11	220	-
12	215	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
13	210	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
14	205	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
15	200	-
16	195	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
17	190	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
18	185	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
19	180	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
20	175	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
21	170	-
22	165	22M22T22N23M23T23N24M24T24N25M25T25N26M26T26N27M27T27N
23	160	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
24	155	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
25	150	1M4M4T5M5T6M6T7M7M8M11M11T12M12T13M13T14M15M18M18T19M19T20M20T20N21M22M25M25T26M26T27M27T27N28M29M
26	145	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
27	140	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
28	135	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
29	130	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
30	125	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
31	120	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
32	115	5M5T5N6M6T6N7M7T7N8M8T8N9M9T9N10M10T10N
33	110	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
34	105	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
35	100	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
36	95	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
37	90	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
38	85	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
39	80	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
40	75	1T2T3T4T5T6T7T8T9T10T11T12T13T14T15T16T17T18T19T20T21T22T23T24T25T26T27T28T29T30T
41	70	-
42	65	5M5T5N6M6T6N7M7T7N8M8T8N15M15T15N16M16T16N17M17T17N18M18T18N
43	60	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
44	55	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M
45	50	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
46	45	1N2N3N4N5N6N7N8N9N10N11N12N13N14N15N16N17N18N19N20N21N22N23N24N25N26N27N28N29N30N
47	40	1T4T5T6T7T8T11T12T13T14T15T18T19T20T21T22T25T26T27T28T29T
48	35	1M1T1N2M2T2N3M3T3N8M8T8N9M9T9N10M10T10N15M15T15N16M16T16N17M17T17N22M22T22N23M23T23N24M24T24N29M29T29N30M30T
49	30	1N4N5N6N7N8N11N12N13N14N15N18N19N20N21N22N25N26N27N28N29N
50	25	10M10T10N11M11T11N12M12T12N13M13T13N14M14T14N15M15T15N
51	20	12M12T12N13M13T13N14M14T14N15M15T15N16M16T16N17M17T17N
52	15	17M17T17N18M18T18N19M19T19N20M20T20N21M21T21N22M22T22N
53	10	1M4M5M6M7M8M11M12M13M14M15M18M19M20M21M22M25M26M27M28M29M
54	5	1M2M3M4M5M6M7M8M9M10M11M12M13M14M15M16M17M18M19M20M21M22M23M24M25M26M27M28M29M30M

Figura 6.1: Instância r55_1

CTA	Pesos	Restrições
0	39	26M13N24T4M4T4N20N27T18M
1	30	25M12M12T12N16N2T26M30N14T18N11M3T19M29T
2	40	28T14M14T14N
3	28	13M24N25M30T3M3T3N23N5T11M17T20T15T22T
4	38	15T24M24T24N20N28T23N5M5T5N11M14N30T6M6T6N17M
5	45	
6	20	26M
7	24	8N27M5M5T5N14M14T14N7N11M16N28M4M4T4N20M20T20N
8	30	26M26T26N8N16M16T16N14T17N27N4T2M2T2N10M10T10N3M9T
9	37	5M24M27M18M18T18N26M26T26N30M2T25M25T25N3N22N16N
10	41	8M8T8N12M12T12N19M19T19N21M28M7N29M29T29N6N9T15M13T4T3M
11	36	1T9T3T2N27M27T27N28M28T28N10M18T4M4T4N21N25M25T25N
12	20	7M16M21T6M29N23N12T2M2T2N18M8T10T5T
13	27	12N
14	28	21M21T21N16M22T19M4T24M12M12T12N28M8M
15	26	8M8T8N
16	40	14T24M24T24N22T4N11M11T11N7T18M28N30T20T13M13T13N21M21T21N
17	29	28T2N30M24M24T24N16M23T21T6N12N9M9T9N18M18T18N
18	20	12M3N9M22M22T22N25T29M29T29N
19	44	29N21M21T21N14M14T14N13T8N22T3M3T3N10M
20	27	12T
21	34	20M4M4T4N13T30M14M14T14N17N26T8N5N2M2T2N21T7T
22	30	14T25M6N10N1T2N24M24T24N28N21T19M30T22N
23	49	11M16M16T16N29M4T9M21N25N26T
24	45	13M13T13N7M2M6T30M20M20T20N9N29N
25	22	17M19N15N16N30M30T30N
26	45	13T20N29M29T29N22M22T22N21M28M28T28N6M6T6N19M19T19N12N11T17M8M
27	32	5T25N30M30T30N11N
28	37	17N12M12T12N20M18T6M6T6N14M22M22T22N
29	23	
30	48	5N22M22T22N19M19T19N29N17M17T17N25N11T12N15N26M26T26N
31	44	3M3T3N10N15T26T19M19T19N2M28M28T28N21T7N
32	49	21T3M3T3N18M20T6N
33	42	24T10N11T9T23N13T5M5T5N12N3M3T3N17M
34	26	5M5T5N17M23N
35	29	16M20T
36	24	22M22T22N11T25T
37	21	5M29M29T29N21M
38	31	30M3T21M21T21N17M11M20M7M7T7N8N10T
39	24	7T21N27T20N14N3N28M4N24N
40	46	14M2M24T9M9T9N28M15T
41	37	24M24T24N27N28M
42	43	4M4T4N23N18M18T18N13M7T25M10T6T17M
43	28	18M18T18N19T7M7T7N9M15T22M22T22N17N
44	36	29T17N24M2N6M14N
45	22	21T
46	31	27T4M22M22T22N24N30N2M23N8M8T8N18N15N29T
47	37	4N3N12N26M26T26N6N8M8T8N
48	49	21N16N4M30M22M26M26T26N25N23M23T23N
49	32	3N19N22T14M14T14N1N21T16N
50	21	28M28T28N6M6T6N17N7M24M
51	46	1N16M5M5T5N19T24M20M20T20N22N3M3T3N
52	33	6T15M15T15N24N10N14N17M17T17N7T8T
53	38	21T11M11T11N27T5M1M1T1N15M17M17T17N28T19M14N
54	48	27M27T27N2M9M20T26M6M6T6N23T11N10T3M5T

Figura 6.2: Instância r55_2

CTA	Pesos	Restrições
0	39	26M13N24T4M4T4N20N27T18M
1	30	25M12M12T12N16N2T26M30N14T18N11M3T19M29T
2	40	28T14M14T14N
3	28	13M24N25M30T3M3T3N23N5T11M17T20T15T22T
4	38	15T24M24T24N20N28T23N5M5T5N11M14N30T6M6T6N17M
5	45	
6	20	26M
7	24	8N27M5M5T5N14M14T14N7N11M16N28M4M4T4N20M20T20N
8	30	26M26T26N8N16M16T16N14T17N27N4T2M2T2N10M10T10N3M9T
9	37	5M24M27M18M18T18N26M26T26N30M21T25M25T25N3N22N16N
10	41	8M8T8N12M12T12N19M19T19N21M28M7N29M29T29N20M20T20N6N9T15M13T4T3M
11	36	1T9T13T2N27M27T27N28M28T28N10M18T4M4T4N21N25M25T25N
12	20	7M16M21T6M29N23N12T2M2T2N18M8T10T5T
13	27	12N
14	28	21M21T21N16M22T19M4T24M12M12T12N28M8M
15	26	8M8T8N
16	40	14T24M24T24N22T4N11M11T11N7T18M28N30T20T13M13T13N21M21T21N
17	29	28T2N30M24M24T24N16M23T21T6N12N9M9T9N18M18T18N
18	20	12M3N9M22M22T22N25T29M29T29N
19	44	29N21M21T21N14M14T14N13T8N22T3M3T3N10M
20	27	12T
21	34	20M4M4T4N13T30M14M14T14N17N26T8N5N2M2T2N21T7T
22	30	14T25M6N10N1T2N24M24T24N28N21T19M30T22N
23	49	11M16M16T16N29M4T9M21N25N26T
24	45	13M13T13N7M2M6T30M20M20T20N9N29N
25	22	17M19N15N16N30M30T30N
26	45	13T20N29M29T29N22M22T22N21M28M28T28N6M6T6N19M19T19N12N11T17M8M
27	32	5T25N30M30T30N11N
28	37	17N12M12T12N20M18T6M6T6N14M22M22T22N
29	23	
30	48	5N22M22T22N19M19T19N29N17M17T17N25N11T12N15N26M26T26N
31	44	3M3T3N10N15T26T19M19T19N2M28M28T28N21T7N
32	49	21T3M3T3N18M20T6N
33	42	24T10N11T9T23N13T5M5T5N12N3M3T3N17M
34	26	5M5T5N17M23N
35	29	16M20T
36	24	22M22T22N11T25T
37	21	5M29M29T29N21M
38	31	30M3T21M21T21N17M11M20M7M7T7N8N10T
39	24	7T21N27T20N14N3N28M4N24N
40	46	14M2M24T9M9T9N28M15T
41	37	24M24T24N27N28M
42	43	4M4T4N23N18M18T18N13M7T25M10T6T17M
43	28	18M18T18N19T17M7T7N9M15T22M22T22N17N
44	36	29T17N24M2N6M14N
45	38	21T11M11T11N27T5M1M1T1N15M17M17T17N28T19M14N
46	31	27T4M22M22T22N24N30N23N8M8T8N18N15N29T
47	37	4N3N12N26M26T26N6N8M8T8N
48	49	21N16N4M30M22M26M26T26N25N23M23T23N
49	32	3N19N22T14M14T14N1N21T16N
50	48	27M27T27N2M9M20T26M6M6T6N23T11N10T3M5T
51	46	1N16M5M5T5N19T24M20M20T20N22N3M3T3N
52	33	6T15M15T15N24N10N14N17M17T17N7T8T

Figura 6.3: Instância v53_1

CTA	Pesos	Restrições
0	33	28M28T28N24M24T24N17M10T3M3T3N21M21T21N1N23M12M14M14T14N26T13N19T2N4T30M
1	27	15M15T15N1T117M17T17N21M4N2T16M28M6N10T18M13T8M7N9M11N20M27M23N25M25T25N11N29M29T29N22T3T30M24M24T24N19M26N
2	41	6M6T6N29M29T21T11N5M18M4M4T4N26M26T26N17M12T
3	48	18N2N26T8M8T8N14N9M9T9N22M10M5M5T5N24N27M29M19T28T7M4N21M21T21N23T1T30N17T3N15T16M12N
4	38	4N9M23M23T23N11T21N6M27M7N17N15T13M3M3T3N10N12T22N14N20T30M19N26N8T18M29M21N25N5N16N24N
5	40	21T22M22T22N17M26N5M5T5N30T27N11N14T12M12T12N15T2N7M7T7N23N
6	46	2N19M20M7M14M4T9N25M25T25N10M18T11M11T11N24M16M7M7T17N
7	28	10M1N22M15M8N9M3N28M28T28N14M4T14N19M19T19N21T18N20M20T20N6N16M16T16N25T24M24T24N11N12T29M13M27M4T17M7N26M5M
8	47	11T18T20T19T4T19T10M10T10N8N22T11M1T1N29T25T2T30M30T30N23N
9	33	22N13N3M3T3N18M18T18N12T5M5T5N8M8T8N2T21N9T17M23M23T23N10M11N1N28T27M
10	26	18M18T18N28M10M10T10N11M19T4T15M15T15N6N2M2T2N25T29T9T23N
11	42	21T15N14N4M4T4N25N30T13M13T13N9M9T9N3N18N6T22T24T16M16T16N19T7M7T7N11N1M12T20T2T27M27T27N8N28M26M17M17T17N23M23T23N
12	44	4M29N24M24T24N14N20N28M13M13T13N5N9N
13	40	20N13M13T13N12M14M23N25M25T25N22T16M16T16N2M10N24M24T24N19N8M29M29T29N7N6M30N1M5T4T18M18T18N3N21N28N17T26T9M9T9N11N
14	41	17T11N4N13T19M19T19N6N27M6N9T9N18M22N26M28T3M3T3N7T2M2T2N
15	29	19M19T19N9M11T28N7M7T7N23M23T23N18T30T17M8T26M15M15T15N6N29N24N25T21T14M13T5M5T5N
16	24	30T28N12N23T17M11N3M3T3N25T16N22M21M21T21N
17	21	18T30M22M28M11T1M26M26T26N8T24M24T24N3M27N25T23N15M15T15N10T9T17N5M6N7N20M20T20N13T
18	33	30T2T15M15T15N29M29T29N18T24N10M10T10N6T27M4M4T4N17M17T17N11T22T21M21T21N16M23M20M20T20N5T9M9T9N7M7T7N
19	49	24M17T16N12M12T12N23M30N9M9T9N3N18N7M7T7N8M14T28M19T27N1T26M19T27N1T26M5M5T5N13N11N20M
20	28	9T24M24T24N1M6N2N11T28T12M4M4T4N20M19N7T13T
21	30	1T24M24T24N10N30M30T30N12M12T12N15N11N26T7T22N4M4T4N3M3T3N23N25M20M20T20N14N13T6M8T2M21T21N2T17M19N
22	47	6N8M8T8N22T27T9N7M7T7N3T15T12M12N
23	43	24M28M12T30T27M20T6N3M3T3N11T21T17N13N15T14N10N29M29T29N23M23T23N18M18T18N21M21T21N19T5M5T5N16M16T16N26M7M7T7N4T1M1T1N22T
24	40	1M12M12T12N3T27M30M30T30N16M24N5M5T5N2T22N25M26T6M6T6N18M18T18N15M15T15N10M10T10N14M28N9M9T9N4N25M21N7N23M23T23N19M19T19N29T
25	32	14N28M8T8N13M13T13N18M4T3M3T3N26M26T26N29T5T17N16M9T24M24T24N11T25T21T30T20N15T6N27M10M10T10N23M23T23N12M12M12T1T2N
26	47	22N12M29M29T29N11T14M19M19T19N13N21M7M7T7N26M26T26N25M1T20M5T3M6M6T6N16M
27	21	14M14T14N8M8T8N25M20N28M24M4T4N17T30T21N12N7M15M15T15N5M5T5N24N13T10M10T10N3M3T3N26N9M9T9N6M6T6N16M11M
28	40	5M5T5N4M22N18T19M30N1M21T21N2M14M8M8T8N3M16T24M29M29T29N11M7T28T10M13M23T26M26T26N27N12N20N15M15T15N6N9M
29	30	27N14T6M22M20M20T20N1M17M29M8N19M19T19N
30	49	2N5N6N22T10M10T10N21N13N30T11N8M9T28M28T28N20N19M19T19N29N16M15T24M
31	30	26M1N21N18T24T25N11M11T11N4M15N23M5T
32	48	21M5N6M23M23T23N16T26N29M7N10M10T10N17T15M13T19M18T11M11T11N25N24T28M14M14T14N27N3N22T12T1N
33	35	7T2M2T2N19T16T12N29T8N27T5T18N30T11M11T11N1M1T1N28T17M26T20M20T20N
34	22	15M8M8T8N25N22N9N12T23N28T7N30M30T30N24M24T24N2M2T2N1M18M29T17N19M13N26N4N14T11M11T11N16M10M10T10N20T27N
35	34	20M7M22T29M15M15T15N11M11T11N26N2N1N9N30N4M5T8T28M23N3T6N13T27N21M21T21N10M10T10N24M24T24N18T
36	23	14M17M17T17N27N10T2M2T2N1T15T9N25N26T28M22M
37	40	9M9T9N14N17T26T15N27T28M28T28N25M25T25N21M3M3T3N24M6N16N5N
38	38	9T16T26T22M1M1T1N2M6M6T6N29M29T29N7M10M10T10N13N30M5M14M4N17M27N11M8M8T8N15M
39	30	26T20M20T20N6M25M25T25N17M17T17N4N7N11N29M9N16T5N22M22T22N8N21N
40	39	9T20M20T20N13M13T13N21N25M1N23N26N7M7T7N22T14N29M11M11T11N2M18M19N6N
41	31	13M13T13N28T2N21M19T17N9N22M6N26M26T26N4M29M
42	33	11T10M10T10N5N23M23T23N24N21N1T29N30M18N22N15N9M25T20M14N4N16N21T7T6M6T6N28T12M12T12N27M13M
43	47	12N6T22M22T22N1N15N8M8T8N24N23T14M14T14N4M2N5M5T5N19M19T19N10T17M17T17N7T30N13N3M20M16M25N26M26T26N11N18M
44	20	17M17T17N20M30T10M13N14N9M9T9N3N28T16M16T16N6M6T6N8T11M11T11N22M21N7T5M19M29M25M24T
45	28	11M21M21T21N10N20M20T20N29N28N9N14M14T14N8M8T8N30N24M22M17N27M2M2T2N19N16M16T16N15M15T15N18N23M13N25N13N
46	46	20N19T7M19N15N30M16M16T16N18M26M26T26N5M5T5N6M6T6N4N29M29N27N10T22M22T22N24M24T24N2N8N23N21T28T25T17T11M11T11N14T3M
47	45	24M3M18M12M2N9M9T9N11N30M26M26T26N7M23M13T22M8M
48	33	11T15M11T11N19T24N8T25M23N27N10N3T28N18M5M16N14N29M30T6N2M4N
49	46	28N7T27N20T6M6T6N8T11N26M14T14N15T13N6M3M3T3N21M21T21N18M4M4T4N19M30T23N29N1M1T1N12M17T22M2M25T16M16T
50	22	4T9N18M6N17M17T17N12T29M8T15M15T15N11N13M13T13N27N21N25N24T16N2N19M

Figura 6.4: Instância v51_1