

**UNIVERSIDADE FEDERAL DA PARAÍBA PROGRAMA DE
PÓS-GRADUAÇÃO EM INFORMÁTICA**

Alexander de Almeida Pinto

**UTILIZAÇÃO DAS METAHEURÍSTICAS GRASP E ILS
ADITIVADO COM PROGRAMAÇÃO LINEAR PARA
RESOLUÇÃO DO PROBLEMA DE CONSTRUÇÃO DE
TRILHOS DE AERONAVES**

Dissertação submetida ao Programa
de Pós-Graduação em Informática da
Universidade Federal da Paraíba para
a obtenção do grau de Mestre em
Informática.

Orientador: Prof. Dr. Lucídio dos
Anjos Formiga Cabral

João Pessoa

2011

Alexander de Almeida Pinto

**UTILIZAÇÃO DAS METAHEURÍSTICAS GRASP E ILS
ADITIVADO COM PROGRAMAÇÃO LINEAR PARA
RESOLUÇÃO DO PROBLEMA DE CONSTRUÇÃO DE
TRILHOS DE AERONAVES**

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Informática e aprovada em sua forma final pelo Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba. João Pessoa, 04/08/2011.

Tatiana Aires Tavares, Dra.
Coordenadora do Curso

Banca Examinadora:

Lucídio dos Anjos Formiga Cabral, Dr.
Orientador

Tatiana Aires Tavares, Dra.

Antonio Carlos Cavalcanti, Dr., UFPB

*Dedico esse trabalho a minha família que me
ajudou em todos os momentos que precisei.*

Acreditar é mais fácil do que pensar. Daí existem muito mais crentes do que pensadores.

Bruce Calvert

AGRADECIMENTOS

Aqui entra os agradecimentos.

RESUMO

Os problemas operacionais cresceram muito em complexidade nos últimos tempos, o que tem tornado necessário o desenvolvimento de técnicas que possam agilizar a tomada de decisão. Empresas que não utilizam sistemas computadorizados com essa finalidade tem perdido espaço entre seus concorrentes.

A construção de trilhos de aeronaves é considerado um dos principais problemas da indústria aeronáutica e se refere ao sequenciamento dos voos de uma companhia aérea de forma que o menor número de aeronaves seja necessário para operá-las. Esse problema possui uma característica combinatória e a sua resolução fica mais difícil a medida que a quantidade de voos envolvidos cresce. Entretanto pequenas modificações nos horários de partida desses voos, ou o acréscimo de algum voo de reposicionamento entre dois aeroportos próximos podem gerar soluções de baixo custo.

Apresentamos um algoritmo híbrido baseado na metaheurística GRASP, com a utilização do ILS e de programação inteira na busca local. Esse algoritmo é indicado para resolução de problemas de larga escala, pois nesse caso fica inviável a aplicação de um algoritmo puramente exato que poderia levar anos para realizar a tarefa. Os resultados preliminares tem mostrado agilidade na obtenção de boas soluções. **Palavras-chave:** Transporte,

PCTA, Metaheurística, Método Exato, GRASP, Rotas e Aeronaves.

ABSTRACT

Write here the English version of your Resumo...

Keywords: Transportation, ARP, metaheuristic, Exact Method, GRASP, Aircraft Routing.

LISTA DE FIGURAS

2.1	Heurística de construção gulosa de uma solução inicial. Fonte: (SOUZA, 2009)	18
2.2	Heurística de construção aleatória de uma solução inicial. Fonte: (SOUZA, 2009)	18
2.3	Procedimento GRASP. Fonte: (FEO; RESENDE, 1995)	20
2.4	Procedimento de construção do GRASP. Fonte: (FEO; RESENDE, 1995)	21
2.5	Procedimento de busca local do GRASP. Fonte: (FEO; RESENDE, 1995)	22
2.6	Procedimento Iterated Local Search. Fonte: (SOUZA, 2009) .	23
2.7	Representação esquemática do funcionamento do ILS. Fonte: (SOUZA, 2009)	24
3.1	Malha hub-and-spoke. Fonte: (Própria)	29
4.1	Representação esquemática do arco do tipo 1. Fonte: (Própria)	31
4.2	Representação esquemática do arco do tipo 2. Fonte: (Própria)	31
4.3	Representação esquemática do arco do tipo 3. Fonte: (Própria)	31
4.4	Representação esquemática do arco do tipo 4. Fonte: (Própria)	32
4.5	Construção de Trilhos de Aeronaves. Fonte: (Própria)	32
5.1	Arcos necessários para ligar dois voos. Fonte: (Própria)	34
5.2	Conversão de um voo para ser utilizado no solver. Fonte: (Própria)	36

LISTA DE TABELAS

7.1	Parametrização dos cenários	43
7.2	Resultados do cenário 1	43
7.3	Resultados do cenário 2	44

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS DO TRABALHO	15
1.2 ORGANIZAÇÃO DA PROPOSTA	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 HEURÍSTICAS CONSTRUTIVAS	17
2.2 METAHEURÍSTICA	18
2.2.1 GRASP	19
2.2.2 ILS	23
2.3 PROGRAMAÇÃO LINEAR.....	25
2.4 TRABALHOS CORRELATOS	26
3 GERAÇÃO DAS INSTÂNCIAS	28
4 DESCRIÇÃO DO PROBLEMA	30
5 MODELO MATEMÁTICO	34
5.1 FUNÇÃO OBJETIVO	35
5.2 RESTRIÇÕES	35
6 MÉTODO PROPOSTO	37
6.1 FASE DE CONSTRUÇÃO	37
6.1.1 Formação dos trilhos de forma sequencial	38
6.1.2 Formação dos trilhos de forma paralela	38
6.1.3 Escolha dos voos de um trilho	38
6.2 FASE DE BUSCA LOCAL	39
6.2.1 Vizinhaça	39
Swap-X	40
Cross-Over	40
Compactação	40
6.2.2 Perturbação usando o método exato	40
7 RESULTADOS PRELIMINARES E DISCUSSÕES	42
Referências bibliográficas	46
Anexo A – Rede de voos da Rio-Sul	48
Anexo B – Tempo dos voos da Rio-Sul	50
Anexo C – Resultado ótimo da instância Rio-Sul	51
Anexo D – Rede de voos da TAM	53
Anexo E – Tempo dos voos da TAM	56
Anexo F – Resultado ótimo da instância TAM	57

1 INTRODUÇÃO

A aviação é o principal meio de transporte capaz atravessar grandes distâncias, a sua utilização é fundamental para o crescimento das empresas e também para o desenvolvimento do turismo mundial. O transporte aéreo é um serviço que oferece benefícios de ordem social e de ordem econômica. Por ser utilizado no turismo e no comércio, ele contribui para o crescimento da economia mundial e é essencial para o rápido transporte de pessoas e de mercadorias ao redor do mundo. Finalmente, o transporte aéreo melhora a qualidade de vida das pessoas proporcionando lazer e experiências com outras culturas.

O uso da aviação comercial tem crescido significativamente nas últimas décadas. Esse rápido crescimento pode ser atribuído a um grande número de fatores. Primeiro, o aumento da renda e da qualidade de vida em muitas partes do mundo que tem incentivado as pessoas a viajar para outras áreas e explorar novas oportunidades. Segundo, a demanda tem aumentado junto com a confiança na aviação como um meio seguro de viajar. Terceiro, o aumento da eficiência nas operações tem acirrado a competitividade e reduzido as taxas e os custos das passagens. Finalmente a globalização tem forçado a viagem de pessoas para fazer negócios em outros países e também para aperfeiçoar as relações políticas e sociais. Espera-se que os impactos desses fatores continuem a acontecer porém com intensidades diferentes.

O aumento do número de companhias aéreas fez crescer a necessidade de obtenção de melhores benefícios, como a redução dos custos e aumento das receitas. Porém os problemas presentes na indústria aeronáutica são complexos envolvendo múltiplas decisões conflitantes que precisam ser otimizadas de uma só vez. Diversas técnicas são desenvolvidas e usadas para tentar melhorar o planejamento e a operação das empresas aéreas. Muitas dessas técnicas estão disponíveis na literatura científica, nos campos da pesquisa operacional e da matemática e normalmente são modelados para funcionar em sistemas computadorizados de alta capacidade com a finalidade de automatizar, ou pelo menos auxiliar a tomada de decisões. Essas técnicas se tornam mais necessárias a medida que a empresa aérea cresce e a tomada de decisão, baseada nos julgamentos individuais e nas experiências, se tornam mais difíceis (ABDELGHANY; ABDELGHANY, 2009).

Os principais problemas relacionados dizem respeito ao planejamento, envolvendo a criação de linhas de trabalho tanto para as aeronaves quanto para a tripulação. O objetivo costuma ser a minimização dos custos operacionais ou a maximização dos rendimentos. Custos operacionais consiste,

por exemplo, nos custos envolvidos com combustíveis, óleo, taxas de aterragem. Também pode ser levado em consideração a perda de rendimentos com a utilização de aeronaves com menos assentos do que a demanda de passageiros, porém fatores como bem estar dos passageiros também podem ser levados em consideração.

A modelagem de mercado é o problema que inicia essa cadeia e é a parte responsável pelo planejamento dos voos de cada região baseado na demanda de passageiros.

Os problemas de planejamento que envolvem as aeronaves mais estudados na literatura são a Atribuição de Frota (*Fleet Assignment*) e a Construção de Trilhos de Aeronaves (*Aircraft Rotation*). E os que envolvem a tripulação são conhecidos como Construção dos Dias de Trabalhos da Tripulação (*Crew Pairing*) e a Escala de Tripulantes (*Crew Scheduling*).

O problema de Atribuição de Frota determina o tipo de equipamento a ser utilizado em cada voo (PIMENTEL, 2005). O problema de construção de trilhos de aeronaves (PCTA) é o foco desse trabalho e será descrito mais adiante. O problema de construção dos dias de trabalho da tripulação visa obter o melhor conjunto de pairings¹ tal que cada voo seja coberto por pelo menos um pairing. Gastos com alojamentos, alimentação, transporte em terra e deadheads² devem ser levados em consideração. O problema de construção das escalas dos tripulantes tem o objetivo de atribuir os pairings a tripulação disponível na companhia aérea, acrescentando as atividades de solo tais como call³, Stand-by duties⁴ e dias de descanso. O objetivo dessa etapa é fazer uma distribuição da forma mais justa possível, tentando balancear a quantidade de trabalho (horas a serem voadas) entre os tripulantes, e também tentar cumprir todas as solicitações da tripulação em relação a preferência dos dias de descanso e das tarefas a serem realizadas, sem violar nenhuma restrição da legislação trabalhista em vigor.

Após a designação da frota de aeronaves ao conjunto de voos existentes segue-se o problema de construção de trilhos de aeronaves (PCTA) Problema de Construção de Trilhos de Aeronaves que também é conhecido

¹ Pairing é o conjunto de voos que pode ser guiados por uma tripulação sem que seja violadas quaisquer regras da legislação vigente e que ao final do último voo o tripulante esteja de volta a sua cidade base.

² Deadhead é o voo que o tripulante viaja sem trabalhar, com a finalidade de transporte para outra localidade normalmente para sua base ou para suprir uma nova demanda.

³ Call é o tempo que a tripulação tem para se apresentar a companhia aérea antes de iniciar de fato seu turno de trabalho.

⁴ Stand-by duties são turnos em que o tripulante fica a disposição da companhia aérea afim de suprir possíveis eventualidades.

na literatura como Aircraft Rotation Problem (ARP) ARPAircraft Rotation Problem. O PCTA é um dos principais problemas presentes na indústria da aviação. No PCTA o objetivo é a construção, para cada uma das frotas da companhia (e para os voos a elas alocados), de sequências encadeadas de voos que possam ser operados por uma única aeronave abiliolivro. Cada uma dessas sequências recebe o nome de trilho e o conjunto desses trilhos é chamado de malha aérea.

Para resolver o PCTA, devemos estar cientes de algumas restrições que envolvem tempo e espaço. Por exemplo, um voo não pode iniciar antes da chegada do voo que lhe antecede, nem de um local diferente da cidade de destino deste voo antecessor, esses exemplos são denominados respectivamente de restrições temporais e geográficas do problema. Há também a restrição de que um voo deve permanecer em solo, entre conexões, por um período de tempo que seja suficiente para fazer a troca de passageiros e abastecimento da aeronave e quando for o caso para a mudança da tripulação, esse tempo varia de acordo com o aeroporto e com o tipo de aeronave.

Existe também a restrição de consistência que está presente em instâncias que possuem frequência. Nessas instâncias um voo pode aparecer em diversos dias. Dessa forma deve-se garantir que o horário de partida desse voo seja o mesmo em todos os dias que ele ocorrer, ou seja, caso alguma modificação seja feita no horário de partida sugerido desse voo, em um dos dias, então todos os outros dias da frequência também devem ser modificados.

Outro aspecto importante diz respeito às restrições de manutenção. Sabe-se que um avião deve ter checagens periódicas. Oportunidades de realizar essas tarefas ocorrem apenas em algumas conexões potencialmente disponíveis. Como consequência, uma sequência de voos deve ser construída de forma que essas restrições não sejam violadas. A fim de incorporar essas restrições facilmente ao nosso framework, assumimos que as rotações são designadas a tipos não específicos de aeronave. Dessa forma, se uma aeronave tem necessidade de manutenção, um voo especial é criado com origem e destino na base de manutenção escolhida e com a sua duração exatamente igual ao tempo de manutenção (PONTES *et al.*, 2002).

Vale ressaltar ainda que na resolução do PCTA deve-se levar em consideração as particularidades específicas de cada companhia aérea como o número de aviões disponíveis na frota, o atraso máximo permitido nos voos, a quantidade máxima de voos que podem sofrer atraso, o número máximo de voos que podem ser cancelados, o número máximo de voos de reposicionamento que podem ser criados entre outros.

De uma maneira geral, o principal objetivo do PCTA é a minimização do número de trilhos seguido da minimização do custo total dos trilhos gerados. Esse custo pode envolver diversos componentes, sendo o tempo médio diário de utilização das aeronaves um dos mais importantes abiliolivro.

Nos dias de hoje, com o avanço da tecnologia e o aumento da competitividade desenvolver soluções com melhor qualidade acaba se tornando um fator decisivo para a permanência no mercado, tornando-se então necessário a obtenção de soluções de forma mais rápida, mais barata e utilizando menos recursos. Por causa da dificuldade que é inerente a essa classe de problemas a qualidade das soluções obtidas manualmente são muito aquém da melhor solução possível. Outra característica que reforça a necessidade da obtenção de melhores soluções é o aumento do tamanho e da complexidade das instâncias trabalhadas. A partir desse cenário pode-se perceber a necessidade de utilização de técnicas de otimização. Na literatura tem-se observado um crescimento no número de trabalhos que se utilizam de metaheurísticas como método de resolução de problemas complexos de otimização combinatoria.

As metaheurísticas podem ser definidas como um conjunto de procedimentos de caráter geral, com capacidade de guiar o procedimento de busca, tornando-o capaz de escapar de ótimos locais. Elas têm como objetivo, encontrar uma solução tão próxima quanto possível da solução ótima do problema com baixo esforço computacional. Em geral, as metaheurísticas são bastante utilizadas na resolução de problemas de otimização. Esses problemas, também conhecidos como problemas NP-difíceis NPNNon-Polinomial, podem ser definidos como um conjunto de problemas para os quais ainda não existe um algoritmo que os resolvam de forma exata e em tempo polinomial maritan2009. Para esses tipos de problemas o uso de métodos exatos é bastante restrito, uma vez que o esforço computacional para encontrar uma solução exata em instâncias reais é consideravelmente alto. Na prática, no entanto, é suficiente encontrar uma solução melhor que a utilizada no ambiente operacional.

A literatura pesquisada (ABDELGHANY; ABDELGHANY, 2009) (??) (CORDEAU *et al.*, 2001) (HAOUARI *et al.*, 2011) (LUCENA; PONTES, 2007) mostra uma grande quantidade de tentativas de resolver o problema utilizando modelagens matemáticas, que apesar de garantir a solução ótima não se mostra viável para resolver grandes instâncias. Alguns trabalhos mostram a similaridade desse problema com o problema do caixeiro viajante assimétrico (CLARKEA *et al.*, 1997). E outros resolvem uma parte do problema utilizando metaheurísticas (??).

Nesse trabalho é proposto o desenvolvimento de um método híbrido baseado em metaheurísticas e em programação linear inteira para resolução do PCTA. O método proposto procura combinar a eficiência computacional das metaheurísticas com a rápida convergência dos métodos exatos. Além disso, a pesquisa constatou que a literatura acerca do PCTA não disponibiliza as instâncias que foram trabalhadas dificultando assim a comparação dos resultados obtidos com esses trabalhos, logo também propomos a criação de um conjunto de instâncias baseado em uma malha real.

1.1 OBJETIVOS DO TRABALHO

Tendo em vista os aspectos apresentados, o objetivo principal dessa proposta de trabalho consiste no desenvolvimento de um método híbrido baseado nas metaheurísticas GRASP e ILS e em programação linear inteira para a resolução do problema construção de trilhos de aeronaves (PCTA). O método proposto tem a finalidade de explorar a eficiência computacional, e irá ser combinada com etapas de refinamentos composta por métodos exatos para acelerar a convergência e adicionalmente fugir de mínimos locais.

Além disso irá ser proposto um conjunto de instâncias baseados em uma instância reais variando em complexidade e tamanho, essas instâncias irão permitir uma melhor comparação desse trabalho com outros.

1.2 ORGANIZAÇÃO DA PROPOSTA

O trabalho está estruturada da seguinte forma:

- Capítulo 1: Apresenta a motivação e as vantagens de resolver o PCTA utilizando metaheurísticas e programação linear e enfatiza a importância desse problema na indústria aeronáutica. Ao final os objetivos do trabalho são descritos.
- Capítulo 2: Apresenta a fundamentação sobre a otimização, metaheurísticas e programação linear. Na seção referente à otimização além da descrição serão discutidos heurísticas construtivas. A seção referente às metaheurísticas irá iniciar com uma descrição seguida do detalhamento das metaheurísticas utilizadas no trabalho, como o GRASP e o ILS. Ao final da fundamentação teórica será feita uma revisão dos principais trabalhos relacionados ao presente trabalho que

estão presentes na literatura.

- Capítulo 3: Mostra como foi obtida a malha da companhia de transporte aéreo TAM e qual a estratégia será utilizada para geração das novas instâncias.
- Capítulo 4: Descreve o problema, explicando os conceitos que serão utilizados no trabalho.
- Capítulo 5: Introduz o modelo matemático que foi desenvolvido.
- Capítulo 6: Descreve o método proposto nesse trabalho, mostrando como foi feita a integração das metaheurísticas e da programação linear inteira e também descreve os parâmetros e as restrições que foram utilizadas.
- Capítulo 7: Apresenta alguns resultados preliminares que já foram obtidos com o solver, dá diretrizes de como utilizar o método proposto e indica o que se espera ter para finalizar o trabalho.
- No final é apresentado a bibliografia estudada, o cronograma de trabalho proposto durante os 24 meses de mestrado e os anexos que contém um maior detalhamento das instâncias e dos resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo é feita a fundamentação dos principais assuntos presentes nesse trabalho: a heurística construtiva, as metaheurísticas e a programação linear. Nas seções seguintes são descritas os aspectos teóricos e os principais métodos relacionados a esse trabalho.

2.1 HEURÍSTICAS CONSTRUTIVAS

As técnicas de resolução heurísticas se utilizam de processos intuitivos com a finalidade de obter uma boa solução, a um custo computacional aceitável, ou seja não garante a otimalidade de um problema. O objetivo é obter em um tempo reduzido uma solução tão próxima quanto possível do ótimo global.

Uma heurística é dita construtiva quando a construção da solução se dá elemento por elemento. A forma de escolha dos elementos variam de acordo com a estratégia e a função de avaliação adotada, essa escolha deve levar em consideração o benefício da inserção de cada elemento para a solução final, escolhendo sempre o *melhor* elemento em cada passo.

O Algoritmo 2.1 mostra o pseudocódigo para a construção de uma solução inicial para um problema de otimização que utiliza uma função gulosa $g(.)$. Nesta figura, 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.

Uma outra forma de obter uma solução inicial é escolhendo os elementos candidatos aleatoriamente. Isto é, a cada passo, o elemento a ser inserido na solução é aleatoriamente selecionado dentre o conjunto de elementos candidatos ainda não selecionados. A grande vantagem desta metodologia reside na simplicidade de implementação. Segundo testes empíricos, a desvantagem é a baixa qualidade, em média, da solução final. Essa técnica é recomendada quando a característica do problema torna mais fácil o refinamento do que a construção de uma solução notasmarcone.

O Algoritmo 2.2 mostra o pseudocódigo para a construção de uma solução inicial aleatória para um problema de otimização.

Para melhores resultados essa etapa deve ser seguida de um refinamento, pois a solução, quando gerada aleatoriamente, não costuma ser de boa qualidade.

Figura 2.1: Heurística de construção gulosa de uma solução inicial.
Fonte: (SOUZA, 2009)

Algoritmo *ConstruçãoGulosa*($g(\cdot), s$)

```

1  $s \leftarrow \emptyset$ ;
2 Inicialize o conjunto  $C$  de 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 enquanto
8 devolva  $s$ ;
fim

```

Figura 2.2: Heurística de construção aleatória de uma solução inicial.
Fonte: (SOUZA, 2009)

Algoritmo *ConstruçãoAleatória*($g(\cdot), s$)

```

1  $s \leftarrow \emptyset$ ;
2 Inicialize o conjunto  $C$  de candidatos;
3 enquanto  $C \neq \emptyset$  faça
4   Escolha aleatoriamente  $t_{\text{escolhido}} \in C$ ;
5    $s \leftarrow s \cup \{t_{\text{escolhido}}\}$ ;
6   Atualize o conjunto  $C$  de elementos candidatos;
7 fim enquanto
8 devolva  $s$ ;
fim

```

2.2 METAHEURÍSTICA

A utilização de métodos exatos para a resolução de problemas reais envolvendo otimização combinatória é restrito. Isso acontece pois com o aumento das instâncias envolvidas, o número de soluções possíveis cresce exponencialmente, fazendo com que as operações necessárias para a sua resolução não possa feita em tempo viável com os computadores atuais.

Para contornar essa limitação e obter soluções para esses tipos de problemas, os pesquisadores desenvolveram técnicas que são capazes de guiar o procedimento de busca e assim encontrar boas soluções (ARAÚJO, 2009). Esses algoritmos, denominados heurísticas, encontram essas soluções utili-

zando pouco recursos computacionais, porém não garantem a solução ótima do problema (DIAS, 2006). Na prática, geralmente, uma boa solução é suficiente, já que a tomada de decisão tem que acontecer em um curto espaço de tempo.

As heurísticas só se aplicam a uma classe restrita de problemas. Para contornar essa restrição, foram desenvolvidas técnicas mais generalistas que foram denominadas de metaheurísticas. As metaheurísticas podem ser definidas como sendo um método heurístico para resolver de forma genérica problemas de otimização com a capacidade de escapar de ótimos locais. A idéia utilizada, normalmente, é obtida de algum evento natural como sistemas biológicos, da física, da inteligência artificial entre outros.

As metaheurísticas podem explorar o espaço de soluções basicamente de duas formas: as metaheurísticas de busca local e as metaheurísticas de busca populacional. Nas metaheurísticas de busca local, o procedimento de busca utiliza uma solução como ponto de partida em cada iteração. As metaheurísticas GRASP, Arrefecimento simulado (*Simulated Annealing*), Busca Tabu e ILS podem ser citadas como exemplos de metaheurísticas ponto-a-ponto. Nas metaheurísticas de busca populacionais, soluções de boa qualidade são combinadas com o intuito de produzir soluções melhores. Podemos citar como exemplo de métodos populacionais, os Algoritmos Genéticos, Colônia de Formigas (*Ant Colony System*), Núvem de Partículas (*Particle Swarm Optimization*) e etc (ARAÚJO, 2009).

Nesse trabalho foram utilizados as metaheurísticas de busca local GRASP e ILS de forma híbrida. As próximas seções descrevem essas metaheurísticas.

2.2.1 GRASP

Essa seção descreve a metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure* - Procedimento de busca adaptativa gulosa e randômica), que foi proposto por Feo e Rezende (FEO; RESENDE, 1995), e cujos conceitos serão utilizados na metodologia proposta para resolução do PCTA. A metaheurística GRASP é um método iterativo do tipo *multi-start* formado por duas fases: uma fase de construção de uma solução e outra de busca local. A fase de construção objetiva gerar uma solução viável para o problema proposto. E a fase de busca local na qual um ótimo local na vizinhança da solução construída é pesquisado. A melhor solução encontrada, ao longo de todas as iterações GRASP realizadas, é retornada.

O pseudo-código descrito no Algoritmo 2.3 ilustra um procedimento GRASP para um problema de minimização. Na linha 1 o custo da função objetivo da melhor solução encontrada é inicializada com ∞ . A linha 2 repete o procedimento de construção e refinamento *GRASPM* vezes, por causa dessa etapa que o GRASP é considerado *multi-start*.

Na linha 3 e 4 são feitas respectivamente a construção e a busca local que são representadas nos Algoritmos 2.4 e 2.5 e serão detalhadas mais adiante.

Nas linhas 5 à 8, se a solução obtida na busca local for melhor que a melhor solução obtida até o momento ($f(s) < f^*$) então são atualizadas respectivamente a solução e o custo relativo a função objetivo dessa solução. A linha 9 encerra as iterações do GRASP e a linha 10 retorna a melhor solução obtida.

Figura 2.3: Procedimento GRASP. Fonte: (FEO; RESENDE, 1995)

Algoritmo *GRASP*($f(\cdot), g(\cdot), N(\cdot), GRASPM$, s)

```

1   $f^* \leftarrow \infty$ ;
2  para 1, 2, ..., GRASPM faça
3      Construção( $g(\cdot), \alpha, s$ );
4      BuscaLocal( $f(\cdot), N(\cdot), s$ );
5      se  $f(s) < f^*$  então
6           $s^* \leftarrow s$ ;
7           $f^* \leftarrow f(s)$ ;
8      fim se
9  fim para
10 devolva  $s^*$ ;
fim
```

Na fase de construção uma solução é iterativamente construída, elemento por elemento. A parte gulosa da função visa gerar uma solução factível de melhor custo. O componente aleatório é incluído para explorar regiões diversas do espaço de soluções e é uma das chaves da efetividade do GRASP.

A fase de construção do GRASP é baseada na construção de uma lista restrita de candidatos (LCR). Essa lista contém os melhores candidatos que podem ser adicionados a solução em um dado momento, a quantidade de elementos dessa lista é regulada pelo α que é um dos parâmetros do GRASP. O α é definido como sendo o nível de aleatoriedade da solução.

Em cada iteração dessa são selecionados todos os elementos que podem ser inseridos na solução e então é formada uma lista de candidatos que é

Figura 2.4: Procedimento de construção do GRASP.
Fonte: (FEO; RESENDE, 1995)

Algoritmo *Construção*($g(\cdot), \alpha, s$)

```

1   $s \leftarrow \emptyset$ ;
2  Inicialize o conjunto  $C$  de candidatos;
3  enquanto  $C \neq \emptyset$  faça
4       $g(t_{min}) \leftarrow \min\{g(t) \mid t \in C\}$ ;
5       $g(t_{max}) \leftarrow \max\{g(t) \mid t \in C\}$ ;
6       $LCR \leftarrow \{t \in C \mid g(t) \leq g(t_{min}) + \alpha(g(t_{max}) - g(t_{min}))\}$ ;
7      Selecione aleatoriamente um elemento  $t \in LCR$ ;
8       $s \leftarrow s \cup \{t\}$ ;
9      Atualize conjunto de candidatos;
10 fim enquanto
11 devolva  $s$ ;
fim

```

ordenada segundo algum critério de ordenação pré-determinado, no caso de um problema de minimização a lista normalmente é ordenada de acordo com o acréscimo na função objetivo que esse elemento acarretaria se fosse escolhido de forma gulosa. A heurística é dita adaptativa porque os benefícios associados com a escolha de cada elemento são atualizados em cada iteração da fase de construção para refletir as mudanças oriundas da seleção do elemento anterior. A componente probabilística do procedimento reside no fato de que cada elemento é selecionado de forma aleatória a partir de um subconjunto restrito formado pelos melhores elementos que compõem a lista de candidatos. Este subconjunto recebe o nome de lista de candidatos restrita (LCR). Esta técnica de escolha permite que diferentes soluções sejam geradas em cada iteração GRASP (SOUZA, 2009). O valor do grau de aleatoriedade α se encontra entre $[0,1]$.

Um valor de $\alpha = 0$ faz com que o algoritmo gere soluções puramente gulosas enquanto a escolha de um $\alpha = 1$ faz com que o algoritmo gere soluções puramente aleatórias.

A construção do GRASP difere do Algoritmo 2.1 por causa das linhas 4 à 7. A linha 4 obtém o valor mínimo que será acrescentado a solução final, dentre os candidatos possíveis e a linha 5 obtém o valor máximo. A linha 6 forma a LCR com os elementos que tiverem o valor entre $g(t_{min}) + \alpha(g(t_{max}) - g(t_{min}))$. Por fim a linha 7 seleciona aleatoriamente um elemento da LCR.

Com isso a quantidade de soluções possíveis é ampliada porém somente soluções promissoras são geradas.

As soluções geradas pela fase de construção do GRASP normalmente não são localmente ótimas com relação à definição de vizinhança adotada. Surge então a necessidade de complementar o método com a adição de uma busca local, que tem como objetivo melhorar a solução construída na fase de construção. O Algoritmo 2.5 descreve um procedimento básico de busca local relativo a uma vizinhança $N(\cdot)$ de s para um problema de minimização. A qualidade da construção gerada causa um impacto direto na busca local, uma vez que essa solução inicial podem constituir pontos de partidas promissores para a busca local, permitindo assim agiliza-los.

Figura 2.5: Procedimento de busca local do GRASP.
Fonte: (FEO; RESENDE, 1995)

Algoritmo $BuscaLocal(f(\cdot), N(\cdot), s)$

```

1  $V \leftarrow \{s' \in N(s) \mid f(s') < f(s)\};$ 
2 enquanto  $|V| > 0$  faça
3   Selecione  $s'$  de  $V$ ;
4    $s \leftarrow s'$ ;
5    $V \leftarrow \{s' \in N(s) \mid f(s') < f(s)\};$ 
6 fim enquanto
7 devolva  $s$ ;
fim
```

O algoritmo de busca local define no passo 1 e 5 o conjunto de vizinhos da solução que melhoram o valor da função objetivo. Do passo 2 à 6 a solução corrente é atualizada enquanto houver uma solução melhor na vizinhança.

O GRASP apresenta basicamente o parâmetro α que pode ser ajustado. Valores de α que levem a uma LCR com tamanho bastante limitado (isto é, valor próximo da escolha gulosa) implicam soluções próximas as da solução gulosa, obtidas com um baixo esforço computacional. Porém, provocam uma baixa variedade de soluções construídas, que normalmente não é interessante para a busca local já que as soluções geradas são muito próximas. Por outro lado a escolha de valores de α muito elevados implicam na geração de uma grande diversidade de soluções mas, por outro lado, muitas das soluções construídas são de baixa qualidade.

Procedimentos GRASP mais sofisticados levam em consideração a mudança do valor de α ao longo das iterações. De acordo com os resultados obtidos em iterações anteriores. Estudos feitos em (PRAIS; RIBEIRO,

2000) indicam que essa adaptação do valor de α produz soluções melhores do que aquelas obtidas considerando-o fixo.

2.2.2 ILS

Essa seção descreve a metaheurística ILS (Iterated Local Search - Busca Local Iterativa) que se baseia na idéia de que um procedimento de busca local consegue melhores resultados quando a medida que a solução base é variada. Esses locais diferentes são obtidos a partir de perturbações em cima da solução ótima local corrente.

O Algoritmo 2.6 ilustra o pseudo-código do ILS. Nele pode-se perceber a necessidade da definição de quatro procedimentos: (a) *GeraSoluçãoInicial()* que obtém o ponto de partida s_0 para o problema; *BuscaLocal(s)*, que retorna o mínimo local da solução s , tendo como base as estruturas de vizinhança definidas; (c) *Perturbação(histórico, s)*, que altera a solução s para outra solução, e se utiliza do histórico para evitar repetir soluções bem como para inferir o grau de perturbação necessário para escapar do mínimo local. E o (d) *CritérioDeAceitação($s, s'', histórico$)*, que decide em qual solução a próxima perturbação será aplicada.

Figura 2.6: Procedimento Iterated Local Search. Fonte: (SOUZA, 2009)

Algoritmo ILS

```

1  $s_0 \leftarrow$  GeraSoluçãoInicial;
2  $s \leftarrow$  BuscaLocal( $s_0$ );
3 enquanto os critérios de parada não estiverem satisfeito faça
4    $s' \leftarrow$  Perturbação(histórico,  $s$ );
5    $s'' \leftarrow$  BuscaLocal( $s'$ );
6    $s \leftarrow$  CritérioAceitação( $s, s'', histórico$ );
7 fim enquanto
8 devolva  $s$ ;
fim
```

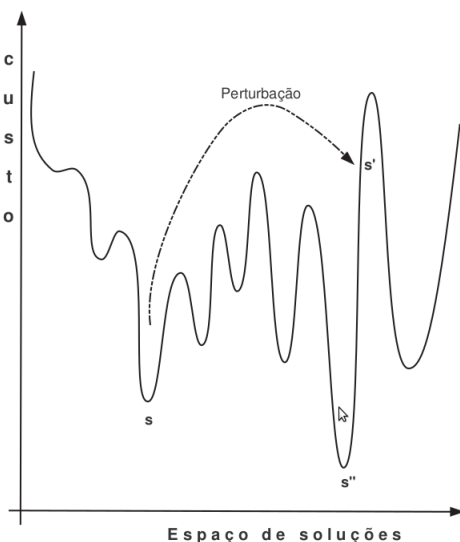
O ILS é dependente da escolha do método de busca local, das perturbações e do critério de aceitação. Normalmente um método de descida é utilizado, mas também é possível aplicar algoritmos mais sofisticados como Busca Tabu ou outras metaheurísticas.

A intensidade da perturbação deve ser forte o suficiente para permitir escapar do ótimo local corrente e permitir explorar diferentes regiões. Ao

mesmo tempo, ela precisa ser fraca o suficiente para guardar características do ótimo local corrente (SOUZA, 2009).

Um aspecto importante do critério de aceitação e da perturbação é que eles induzem aos procedimentos de intensificação e diversificação. A intensificação consiste em procurar melhores soluções nas área de busca corrente, isso acontece reduzindo a força da perturbação que faz com que as novas soluções de partida se encontrem nas proximidades da anterior. A diversificação acontece com a aplicação de grandes perturbações.

Figura 2.7: Representação esquemática do funcionamento do ILS.
Fonte: (SOUZA, 2009)



A Figura 2.7 demonstra o funcionamento do método ILS em um problema de minimização. Dado um ótimo local s , é realizada uma perturbação que lhe direciona para s' . Depois da aplicação da busca local, o novo mínimo s'' , melhor que a anterior, é encontrada. Ou seja $f(s'') < f(s)$.

Uma exemplo de perturbação seria a aplicação sucessiva de estruturas de vizinhança a solução corrente.

2.3 PROGRAMAÇÃO LINEAR

A programação linear é provavelmente a mais conhecida e utilizada técnica de otimização em todo o mundo. É geralmente utilizada para tomada de decisões gerenciais sobre a alocação de recursos para produção. Os custos dos recursos e as receitas geradas pelos produtos são usados para determinar a melhor solução. Qualquer problema que possa ser formulado com variáveis de decisão reais, tendo uma função-objetivo linear, e funções de restrição lineares, em princípio pode ser solucionado através da programação linear. Tais programas originariamente utilizavam o método *Simplex*, porém, mais recentemente, métodos de "*pontos interiores*" se mostraram mais eficientes.

Embora a programação linear seja muito eficiente para a resolução de problemas lineares, sua aplicação a problemas que apresentem objetivos ou restrições não-lineares tem levado a problemas e falhas de modelagem. Em alguns casos, funções não-lineares podem ser aproximadas por algumas funções lineares conjugadas, e a programação linear ainda pode ser utilizada. Contudo, isso leva a uma representação ineficiente do problema, podendo causar matrizes de decisão explosivamente grandes que demandam um tempo excessivo para resolução. Esta é uma dificuldade comum em problemas que envolvem, por exemplo, "*scheduling*" e "*sequenciamento*" de processos.

De forma equivalente, outros tipos de variáveis não podem ser tratadas diretamente com o uso de programação linear. Programação inteira usa programação linear para resolver problemas sobre variáveis inteiras, mas ainda com funções objetivo e restrições puramente lineares. As variáveis inteiras são representadas como variáveis reais no algoritmo de resolução do problema. Então um processo repetitivo é usado para "delimitar" o valor destas variáveis em valores inteiros, através da adição de restrições e reprocessamento da solução. Esse método, conhecido como "*branch & bound*", finaliza quando todas as variáveis assumem valores inteiros. Quando o número de variáveis inteiras é pequeno, a programação inteira soluciona o problema rapidamente. Infelizmente esse procedimento pode consumir muito tempo com um número grande de variáveis inteiras, podendo, em alguns casos, necessitar de milhões de iterações para serem resolvidos.

Essa técnica foi muito utilizada na segunda guerra mundial para otimizar as perdas inimigas e reduzir o custo das operações e também é utilizado no planejamento de algumas empresas.

2.4 TRABALHOS CORRELATOS

O trabalho de Argüello e Bard (ARGUELLO; BARD, 1997) apresenta um algoritmo baseado no GRASP para reconstruir trilhos de aeronaves que tenham sofrido atrasos durante o decorrer do dia e tem como principal objetivo a redução dos custos da reatribuição das aeronaves aos voos que é mensurado a partir do atraso dado aos voos e pelo número de voos cancelados.

Nesse trabalho foi utilizado a ideia de trilhos cancelado, que é formado pelos voos que não serão levados em consideração na solução final.

A reconstrução dos trilhos é feita com a utilização sucessiva de três estruturas de vizinhança, *flight routing augmentation*, *partial route exchange* e *simple circuit cancelation* onde as duas primeiras são aplicadas em um par de trilhos e a terceira é aplicada em trilhos individualmente

O *flight routing augmentation* remove uma sequência de voos do primeiro trilho e acrescenta eles no trilho de destino, ou seja, o segundo trilho é acrescido dos voos que foram removidos do primeiro. O trilho de destino pode crescer de três formas. Primeiro um circuito pode ser inserida no seu início. Um circuito é uma sequência de voos que se origina e termina no mesmo aeroporto. A segunda forma é a adição de um circuito em algum lugar entre o primeiro e o ultimo voo. A terceira forma envolve a adição de uma sequência de voos, que não precisa ter a mesma origem e destino, e a sua inserção no final do segundo trilho. Lembrando que apenas movimentos viáveis são avaliados.

O movimento de *partial route exchange* é uma simples troca de um par de sequências de voos. Dois tipos de trocas são possíveis. A primeira é a troca de duas sequências que possuam os mesmos extremos. E a segunda é uma troca que resulta na mudança do aeroporto de destino. Um trilho de cancelamento não pode trocar seus aeroportos de destino com outro trilho pois esse movimento poderia causar uma violação na restrição de balanceamento de aeronaves.

O *simple circuit cancelation* é feito em um único trilho e ela simplesmente remove um circuito desse trilho e efetua a criação de um novo trilho de cancelamento. Além disso foi desenvolvido um modelo matemático que foi utilizado apenas para a obtenção de um limite inferior (*lower bound*).

Mercier e Soumis (MERCIER; SOUMIS, 2007) resolveram o PCTA em conjunto com o problema de escala de tripulantes pois Cordeau et al. (CORDEAU *et al.*, 2001), Klabjan et al. (KLABJAN *et al.*, 2002) e Cohn e Barnhart (COHN; BARNHART, 2003) mostraram que a resolução desses problema de forma integrada pode gerar soluções que são significan-

temente melhor que as geradas de forma sequencial. Com essa finalidade eles proporam uma formulação compacta do problema e utilizaram o método de decomposição de Benders com um procedimento de geração de restrição dinâmica para resolvê-lo. Com a agregação desses dois problemas a resolução se tornou pesada e viável apenas para instâncias diárias. Os testes do algoritmo foram baseados em instâncias contendo no máximo 500 voos que foram fornecidas por duas grandes companhias aéreas, porém elas não se encontram disponíveis no artigo.

Pontes R., et al (PONTES *et al.*, 2002) utilizaram a fase de construção do GRASP para resolver o PCTA, também propuseram um modelo matemático que foi adaptado para auxiliar na geração da nossa solução. Além disso uma instância da Rio-Sul foi disponibilizada para a realização de testes. Com o solver eles conseguiram obter a solução ótima dessa instância mas o autor informou que essa resolução demorou dias para finalizar. Com a utilização da heurística eles conseguiram apenas se aproximar dessa solução porém com um tempo de 384 segundos.

Em (HAOUARI *et al.*, 2011) Mohamed et al. resolveu de forma integrada o problema de atribuição de frota e o problema de construção de trilhos de aeronaves, para uma pequena empresa de aviação a TunisAir. Além disso as restrições de manutenção não foram levadas em consideração pelo fato dela poder ser feita em todos os aeroportos em que as aeronaves passam a noite.

ão

3 GERAÇÃO DAS INSTÂNCIAS

Atualmente existem diversas fontes na qual se podem obter instâncias para problemas de otimização combinatória sendo uma das mais conhecidas a OR-Library ¹ que foi descrito inicialmente em J.E.Beasley (BEASLEY, 1990) permitindo o acesso a centenas de conjuntos de instâncias a partir da Internet.

Apesar da existência dessas entidades não foram encontrados nenhuma instância que fosse compatível com o problema de construção de trilhos de aeronaves, fazendo-se então necessário a criação de um conjunto de instâncias próprias que além de permitir a conclusão desse presente trabalho ainda irá servir como base para futuras propostas.

Atualmente estamos trabalhando com 2 instâncias. Uma é referente à uma malha diária da companhia aérea Rio-Sul, que é formada por 107 voos. Essa instância foi obtida a partir de um relatório técnico da Universidade Federal do Rio de Janeiro (PONTES *et al.*, 2002) e adaptada para corresponder as características necessárias do nosso modelo.

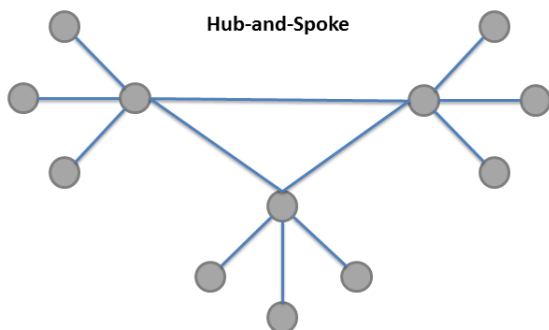
A outra instância trabalhada é a da empresa de transporte aéreo brasileira denominada TAM (<http://www.tam.com.br>). A obtenção desses dados foi feita através da seleção manual do conjunto de voos que ela operava em dezembro de 2010. Foram selecionados os voos operados pelo equipamento Airbus Industrie A310 que tinham o horário de partida iniciando em uma segunda-feira que foi identificada como sendo o dia 0 (zero) apenas para permitir sua utilização no algoritmo. Essa instância que foi obtida é composta por 241 voos e possui uma grande quantidade de ligações entre os 31 aeroportos envolvidos tornando o grau de complexidade mais elevado que instâncias com a características hub-and-spoke que é mais comum nas malhas comerciais norte-americanas.

Uma malha é considerada como sendo *hub-and-spoke* quando existe uma grande concentração de voos em poucos aeroportos como pode ser visto na Figura 3.1.

Diversas instâncias também serão geradas a partir dessa, variando o número de voos e as características das malhas com a finalidade de gerar instâncias com um variado grau de complexidade. Algum esforço também está sendo voltado para a obtenção de novas instâncias reais, porém pela falta de colaboração das empresas de transporte aéreo esse trabalho se torna demorado.

¹ A OR-Library pode ser acessado em <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>

Figura 3.1: Malha hub-and-spoke. Fonte: (Própria)



4 DESCRIÇÃO DO PROBLEMA

Os voos são organizados de forma que cada aeronave seja responsável por uma sequência válida, que é chamada de trilha. O conjunto desses trilhos é a solução do problema e é denominada de malha. Essa malha deve conter o menor número possível de trilhos que atenda todos os voos planejados com o mínimo de modificações em relação ao planejamento inicial.

As restrições que envolvem o PCTA induzem a formação de uma rede de possíveis conexões. Nessa rede os nós representam os voos e os arcos representam as conexões possíveis entre esses voos. Dessa forma, o problema pode ser formulado como um problema de minimização de fluxos em uma rede.

Dado a possibilidade de mudanças no tempo de partida sugerido dos voos e também a permissão para criar voos de reposicionamento, uma grande quantidade de soluções podem ser geradas. A ligação dos voos pode ocorrer de 6 formas distintas aqui denominado tipos de arcos. Os arcos do tipo 1 permitem a ligação de voos sem a utilização de atrasos e/ou reposicionamento. Os arcos do tipo 2 utilizam atrasos mas não o reposicionamento. Os arcos do tipo 3 permitem o sequenciamento com a utilização de um voo de reposicionamento mas sem inserir atraso em nenhum dos voos envolvidos. Os arcos do tipo 4 utilizam-se de atrasos e de um voo de reposicionamento para fazer a ligação entre dois voos. Os arcos do tipo 5 e 6 são usados no modelo, que é baseado no fluxo em grafo, para representar respectivamente o nó origem(source) e o destino(sink). Abaixo um maior detalhamento desses arcos:

- Conexão natural (Arco do tipo 1) - Os arcos desse tipo conectam dois voos respeitando o tempo de partida sugerido e a restrição geográfica. Eles são associados com as ligações que não requerem mudanças no tempo de partida e nem precisam de voos de reposicionamento. O arco do tipo 1 não apresenta custo para ser adicionado a solução.
- Mudança no tempo (Arco do tipo 2) - Apesar de ter os voos incidentes no mesmo aeroporto, os arcos desse tipo não permitem a ligação de forma direta pois o tempo de solo disponível não é suficiente para permitir a ligação. No entanto, a escolha desse tipo de arco implica em uma mudança no tempo de partida sugerido para quaisquer um dos voos envolvidos. O custo de um arco desse tipo é igual a soma dos valores absolutos dos atrasos (em minutos) dos horários de partida envolvidos.

Figura 4.1: Representação esquemática do arco do tipo 1. Fonte: (Própria)

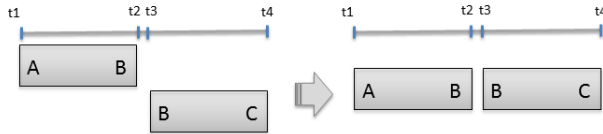
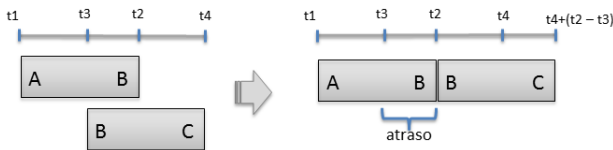
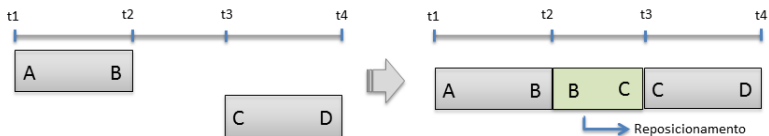


Figura 4.2: Representação esquemática do arco do tipo 2. Fonte: (Própria)



- Voos de reposicionamento (Arco do tipo 3) - Esses arcos representam conexões entre dois voos em que a origem parte de um aeroporto diferente do local de partida do voo de destino, no entanto, existe tempo suficiente para um voo de reposicionamento, entre os dois locais, sem violar as restrições de tempo de solo. Os custos de um arco do tipo 3 é igual a duração do voo de reposicionamento, incluindo o seu tempo de solo.

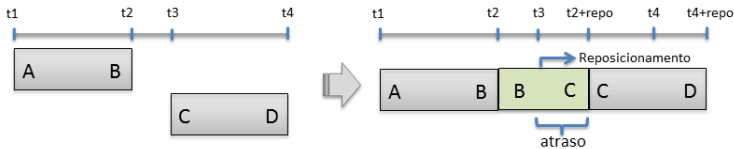
Figura 4.3: Representação esquemática do arco do tipo 3. Fonte: (Própria)



- Voos de reposicionamento mais mudança de tempo (Arco do tipo 4) - Esses arcos representam conexões que precisam de um voo de reposicionamento mais mudança no tempo de partida sugerido. O arco do tipo 4 tem custo igual tempo do voo de reposicionamento, incluindo o tempo de solo, mais a soma dos atrasos dos horários de partida envol-

vidos em valor absoluto.

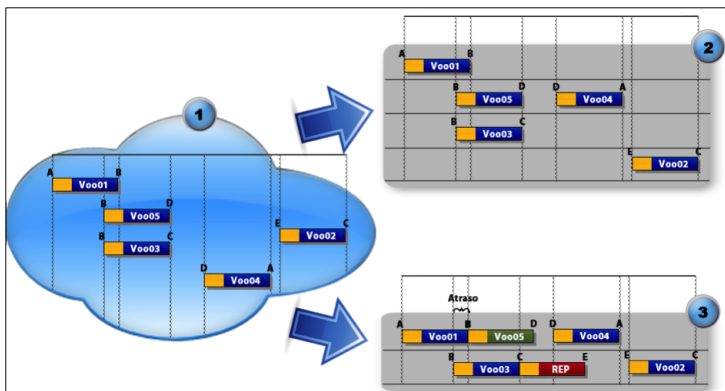
Figura 4.4: Representação esquemática do arco do tipo 4. Fonte: (Própria)



- Arcos do nó fonte ou *source* (Arco do tipo 5) - Esses arcos são criados para identificar o início de um trilho e é com ele também que se sabe a quantidade de trilhos necessários para resolver o problema. Cada arco do tipo 5 tem o custo igual a 1000.
- Arcos do nó final ou *sink* (Arco do tipo 6) - Esses arcos tem como destino o nó fictício que é usado para finalizar um trilho no modelo. Os arcos do tipo 6 não tem custo.

Abaixo na Figura 4.5 temos dois exemplos de montagem de trilhos feitas a partir de um conjunto fictícios de voos. Cada caixinha laranja e azul representa um voo, onde a parte laranja representa o tempo de solo que cada voo deve obedecer e a azul seria o tempo de voo da cidade de origem para a cidade de destino. As letras A, B, C, D, E representam as cidades e a linha pontilhada indica o tempo de início e de término de cada voo.

Figura 4.5: Construção de Trilhos de Aeronaves. Fonte: (Própria)



A parte 1 da Figura 4.5 representa os voos da companhia que ainda não foram cobertos por nenhuma aeronave e nas partes 2 e 3 são demonstrado duas formas de organizar esses voos em trilhos.

Na parte 2 temos a melhor forma possível de se organizar os voos da parte 1 utilizando apenas os arcos do tipo 1, ou seja sem a utilização de atrasos ou de voos de reposicionamento. Dessa forma se consegue uma formação com 4 trilhos.

Na parte 3 temos a melhor forma de organizar os voos utilizando todos os arcos e um atraso máximo equivalente a um tempo de solo. Dessa forma se consegue uma formação com apenas 2 trilhos.

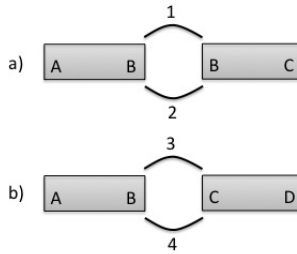
Pode-se verificar que a utilização de diferentes tipos de arcos pode proporcionar uma melhora significativa no número de trilhos. Porém essa abordagem faz com que o número de soluções possíveis tenha uma cardinalidade muito superior a utilização de arcos apenas do tipo 1 que por si só já gera uma quantidade de soluções bem elevada, por isso os arcos devem ser utilizados de forma controlada.

5 MODELO MATEMÁTICO

A idéia utilizada nessa formulação é que não há necessidade de modelar os 4 tipos de arcos para cada voo. Para dois voos quaisquer existem apenas 2 tipos de arcos que podem vir ocorrer como ilustrado na Figura 5.1.

Em a) os voos respeitam a restrição geográfica, dessa forma apenas os arcos de tipo 1 e 2 precisam ser modelados uma vez que não teria sentido fazer um voo de reposicionamento nessa situação. Em b) os aeroportos em questão são diferentes, sendo necessário apenas a modelagem dos arcos do tipo 3 e 4, percebe-se que não teria outra saída se não efetuar um voo de reposicionamento.

Figura 5.1: Arcos necessários para ligar dois voos. Fonte: (Própria)



Seja $D = (V, A)$ um grafo representando uma instância do PCTA, onde o conjunto de vértice $V = v_i : i \in I$ de D é indexado por $I = 1, 2, \dots, n+1, n+2$ onde v_{n+1} e v_{n+2} , identificam, respectivamente, os nós fonte e destino. E os nós restantes referem-se ao conjunto de arcos originais, com n elementos. Seja os custos $c_{ij} : (i, j) \in A$ introduzidos acima, estando associados com cada arco da instância.

Seja $x_{ij} : (i, j) \in A$ um conjunto binário 0-1 de variáveis usada para controlar a inclusão ($x_{ij} = 1$) ou a exclusão ($x_{ij} = 0$) de um arco (possível conexão) entre vértices (voos) v_i e v_j . O conjunto \bar{I} identifica o conjunto de nós excluindo o nó fonte (v_{n+1}) e o nó de destino (v_{n+2}). Variáveis reais δ_i e θ_i , $i \in \bar{I}$ são usados para representar, respectivamente, o desvio do tempo de partida sugerido e a norma desse desvio para v_i . Essas variáveis devem no entanto obedecer $-\gamma_i \geq \delta_i \geq \gamma_i$ e $0 \geq \theta_i \geq \gamma_i$, onde γ_i é o valor máximo de desvio permitido (em cada direção) do tempo de partida sugerido para o voo. Finalmente o tempo de partida sugerido que é dado por $s_i : i \in \bar{I}$.

5.1 FUNÇÃO OBJETIVO

$$\text{Minimizar } \sum_{i \in I} \sum_{j \in I} x_{ij} c_{ij} + \sum_{i \in \bar{I}} \theta_i \quad (1)$$

5.2 RESTRIÇÕES

a) Garantia de recobrimento dos voos

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in \bar{I} \quad (2)$$

$$\sum_{j \in I} x_{ij} = 1 \quad \forall i \in \bar{I} \quad (3)$$

b) Viabilidade das conexões

$$s_i + t_i x_{ij} - INF(1 - x_{ij}) + \delta_i \leq s_j + \delta_j \quad \forall i, j \in \bar{I} \quad (4)$$

c) Modulo do desvio do tempo de partida sugerido

$$\theta_i \geq \delta_i \quad \forall i \in \bar{I} \quad (5)$$

$$\theta_i \geq -\delta_i \quad \forall i \in \bar{I} \quad (6)$$

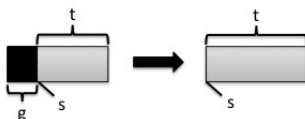
d) Limites das variáveis

$$-\gamma_i \geq \delta_i \geq \gamma_i \quad \forall i \in \bar{I} \quad (7)$$

$$0 \geq \theta_i \geq \gamma_i \quad \forall i \in \bar{I} \quad (8)$$

Pode-se perceber que o modelo matemático não faz menção ao tempo de solo (g), isso ocorre porque esse tempo é incorporado ao voo como demonstrado na Figura 5.2, ou seja o tempo de partida sugerido s passa a ter o valor $s - g$ e a duração t do voo passa a ter o valor $t + g$. Uma vantagem de usar essa abordagem que integra o tempo de solo ao voo é que a quantidade de restrições é reduzida.

Figura 5.2: Conversão de um voo para ser utilizado no solver. Fonte: (Própria)



Além disso o conjunto A contém apenas um tipo de arco, o arco do tipo 1 se os voos satisfazem a restrição geográfica e o arco do tipo 3 caso não satisfaçam. Os arcos do tipo 2 e 4 são modelados a partir da variável δ que tem seu custo acrescentado na função objetivo.

Essa estratégia permite a redução de 3 arcos para cada voo, o que deixa o modelo mais leve.

O cálculo dos custos são feitos através de um pré-processamento, onde os arcos viáveis recebem os valores referentes ao seu tipo, por exemplo, no caso de um arco originário do nó source, arco do tipo 6, um custo 1000 é atribuído. No caso de arcos que deverão ser evitados um custo elevado é atribuído.

6 MÉTODO PROPOSTO

O método proposto se utiliza do GRASP, do ILS e da abordagem exata através da programação linear inteira, ela visa tirar proveito das vantagens de cada uma dessas técnicas.

Da mesma forma que as outras abordagens heurísticas esse novo algoritmo consome pouco tempo computacional e possui uma forma de escapar de mínimos locais.

O framework utilizado é o GRASP, porém a fase de busca local foi adaptada para executar em conjunto com o ILS. A hibridização das metaheurísticas citadas com a programação linear inteira provê uma aceleração na obtenção de boas soluções através de uma intensificação em uma direção totalmente arbitrária. O uso dessas técnicas também proporciona um alto grau de convergência.

A busca local usa o método de descida variando entre 3 estruturas de vizinhança, o *swap-x*, o *crossover* e a *compactação*, que serão explicadas nas próximas seções.

6.1 FASE DE CONSTRUÇÃO

A construção da solução é feita elemento a elemento utilizando o GRASP. A primeira coisa a se fazer é ordenar o conjunto de voos a partir do seu tempo de partida sugerido. O algoritmo só termina quando todos os voos já foram alocados em algum trilha.

Existe duas formas de fazer a montagem da solução, uma seria a montagem trilha a trilha, onde um novo trilha só poderia ser criado quando o anterior já estivesse saturado. A outra forma é a montagem de trilhos de forma paralela, que, a priori, provocaria uma melhor distribuição dos voos. Na prática a primeira abordagem é adotada, pois, nas instâncias disponíveis ela apresentou, sempre, soluções de melhores qualidades. Uma maior quantidade de testes ainda é necessário para decidir qual a abordagem deverá ser utilizada ou se deverá, por exemplo, ser feita uma alteração na estratégia escolhida de acordo com alguma característica da instância.

6.1.1 Formação dos trilhos de forma sequencial

Quando se pensa na escolha do primeiro voo do trilho a decisão imediata é a escolha do voo que contenha o menor horário de partida sugerido. Porém essa escolha reduz a quantidade de soluções que podem ser geradas, pois a escolha dos voos restantes do trilho é diretamente influenciada pela escolha do voo inicial.

A abordagem utilizada para a escolha do primeiro voo inicia com a criação de uma lista de candidatos iniciais (LCI) que é formada pelos 5 voos com os menores horários de partida sugerido que ainda não foram alocados em nenhum outro trilho. A escolha do voo inicial é feita de forma aleatória entre os elementos da LCI.

6.1.2 Formação dos trilhos de forma paralela

Essa estratégia monta um conjunto de trilhos e constroi eles de forma paralela. Em cada iteração o trilho corrente é escolhido a partir desse conjunto de forma aleatória. O passo seguinte é a adição de um voo a esse trilho ou a sua remoção do conjunto de trilhos que estão em construção, esse segundo caso ocorre quando a lista de candidatos para esse trilho é vazia.

6.1.3 Escolha dos voos de um trilho

A escolha do primeiro voo de um trilho é feita como explicado nas seções anteriores. A escolha dos demais voos é feita com base no tipo de arco e na lista restrita de candidatos.

Os tipos de arcos foram definidos no Capítulo 4, porém nessa etapa apenas 4 tipos são considerados, o A_1, A_2, A_3, A_4 que representam formas de ligações entre os voos. Os arcos do tipo 5 e 6 só são utilizados apenas na modelagem matemática. Os arcos do tipo 1 permitem a ligação de voos sem a utilização de atrasos e/ou reposicionamentos. Os arcos do tipo 2 utilizam atrasos mas não o reposicionamento. Os arcos do tipo 3 permitem o sequenciamento com a utilização de um voo de reposicionamento mas sem inserir atraso em nenhum dos voos envolvidos. Os arcos do tipo 4 utilizam-se de atrasos e de um voo de reposicionamento para fazer a ligação entre dois voos. Os arcos do tipo 5 partem do nó *source* e servem para modelar o início de um trilho. Os arcos do tipo 6 tem chegam ao nó *sink* e indicam o fim de um

trilho.

O primeiro passo na escolha de um voo é a definição do tipo de arco que irá ser utilizado. Essa escolha é feita tendo como base as probabilidades 0.79, 0.16, 0.04, 0.01 que representam respectivamente os arcos do tipo 1 a 4. Esses valores foram obtidos a partir da porcentagem dos tipos de arcos presentes em uma solução ótima de um problema real.

De posse do tipo de arco, é feita então a formação da lista de candidatos. Essa lista é ordenada de acordo com o seu horário de partida sugerido, caso o arco seja do tipo A_1 , ou pelo custo associado a sua escolha para os demais tipos de arco. No caso da lista de candidatos não possuir nenhum voo, então outro tipo de arco é sorteado, até que não seja possível acrescentar nenhum voo ao trilho. Quando isso ocorre a construção desse trilho é finalizada.

Caso seja possível a obtenção de uma lista de candidatos então ela é reduzida tendo como base o passo 4 a 6 do algoritmo 2.4. Como está lista se encontra ordenada, então, o elemento de menor impacto (v_{menor}) na solução é o primeiro e o de maior impacto (v_{maior}) é o último. Dessa forma o elemento escolhido poderá ter o seu valor de impacto na solução de até $valor_{menor} + \alpha * (valor_{maior} + valor_{menor})$. O valor de α ainda é objeto de estudo, mas bons resultados tem sido obtido para α igual a 0.5.

6.2 FASE DE BUSCA LOCAL

A fase de busca local recebe uma solução e tem como objetivo melhorá-la. No método proposto essa fase foi substituída pelo ILS. Ou seja primeiro são aplicados as estruturas de vizinhança, visando obter o valor ótimo local da solução. Depois é feita uma perturbação que diversifica melhorando o valor da função objetivo. Quando nenhuma das duas estratégias consegue melhorar a solução então a busca local encerra e uma nova iteração do GRASP pode ser iniciada.

6.2.1 Vizinhança

Foram definidas 3 estruturas de vizinhança, o Swap-X e o Cross-Over, que tem o objetivo de remover modificações nos horários de partida sugeridos dos voos, e a Compactação, que promove a redução do número de trilhos. Abaixo essas estruturas são explicadas.

Swap-X

Esse operador efetua a troca de X voos de um trilho por um conjunto de voos de outro trilho. No método proposto apenas os movimentos do tipo Swap-1 e Swap-2 são utilizados, pois essa vizinhança é considerada grande. Na Figura X um caso de melhoria no custo dos trilhos é exemplificada.

Cross-Over

A ideia do operador *crossover* é a de efetuar troca entre dois segmentos de trilhos com a finalidade de gerar novos trilhos com menos modificações no horário de partida. A Figura X ilustra uma melhoria causada por um movimento desse tipo.

Compactação

A compactação é a única estrutura de vizinhança utilizada que é capaz de reduzir a quantidade de trilhos da solução final.

Isso ocorre porque ela consegue, insere um trilho em outro de forma direta ou com a utilização de um voo de reposicionamento.

A figura X mostra a redução de um trilho com a utilização desse movimento.

6.2.2 Perturbação usando o método exato

A perturbação normalmente é utilizada quando as estruturas de vizinhança não conseguem melhorar a solução. Quando isso ocorre pode-se dizer que a solução corrente é a ótima local com relação a vizinhança definida.

Para tentar encontrar outros mínimos locais aplica-se uma modificação na estrutura da solução, mesmo que isso provoque uma piora na sua qualidade, e depois procura-se melhorá-la aplicando novamente uma busca local.

O método de perturbação utilizado aqui difere do que normalmente é aplicado pois a solução, apesar de ter sua estrutura modificada, ainda consegue melhorar a sua qualidade.

A sua utilização ocorre com a seleção de um conjunto de trilhos,

que juntos definem um subproblema, e a posterior aplicação de um método exato no conjunto de voos que os formam. O método exato irá retornar a configuração ótima desses voos, que serão agrupados novamente a solução antiga. A seleção dos trilhos é feita com base no seu *grau de compactação*. O grau de compactação é definido como sendo a porcentagem de utilização efetiva de um trilho com relação ao tempo de partida do primeiro voo e o tempo de chegada do ultimo voo da instância. O calculo do grau de compactação não leva em consideração os voos de reposicionamento, pois eles não são passados para o modelo.

Os trilhos são adicionados a solução até o limite de 80 voos, pois o solver consegue, de forma imediata, resolver um problema desse porte.

Foram estudadas 3 formas de adicionar os trilhos ao solver:

- Adição dos trilhos com maior grau de compactação.
- Adição dos trilhos com menor grau de compactação.
- Alternar entre a adição de um trilho com maior grau de compactação e outro com o menor grau de compactação.

A utilização da segunda abordagem proporcionou melhores resultados.

7 RESULTADOS PRELIMINARES E DISCUSSÕES

Todos os algoritmos descritos foram desenvolvidas na linguagem C++ usando a biblioteca CPLEX Academic 12 para implementar o mecanismo de programação inteira. Todos os experimentos computacionais foram feitos em um notebook com a seguinte especificação: Pentium T4500 2.3 Ghz com 2 GB de RAM e rodando o sistema operacional sistema operacional Linux Ubuntu 11.04.

Foram efetuadas 10 iterações do GRASP usando um α de 0.5 e a busca local finalizava quando não conseguia melhorar o resultado. Para demonstrar a eficiência dos resultados foram realizadas comparações com o resultado ótimo obtido com um procedimento de programação inteira implementando o modelo descrito no capítulo 4. A coluna s^* indica o valor ótimo. O método híbrido foi executado 20 vezes e apenas a média dos valores obtidos foram levados em consideração. A coluna s indica a média dos valores obtidos com a execução do algoritmo, a coluna tempo indica a média da duração das execuções em segundos. A coluna final indicada por *GAP* indica a diferença percentual das soluções e é calculado como segue:

$$GAP = (s^* - s) / s$$

Para os testes foram utilizados duas instâncias diárias, uma da Rio-Sul(107 voos) e outra da TAM (241 voos). Com a finalidade de permitir estimar o tempo computacional necessário para resolver instâncias maiores foi proposto a extensão da frequência dos voos da instância Rio-Sul para uma semana, dessa forma foi gerado uma instância com 749 voos. Para simplificar foram adotados o tempo de solo de 20 minutos para todos os aeroportos, tendo em vista que a maioria deles opera em média com esse tempo.

A resolução dessas instâncias foram parametrizadas levando em consideração dois cenários. O cenário 1 faz o sequenciamento dos voos sem a permissão de utilizar nenhum atraso, essa representação é comum nas companhias que não aceitam a modificação do planejamento inicial. O cenário 2 se utiliza de atrasos permitindo assim uma maior liberdade na hora da montagem dos trilhos. Os parâmetros utilizados são detalhados na Tabela 7.1.

Pode-se observar que nos dois cenários a solução ótima foi obtida para a instância da Rio-Sul. Na instância da TAM a solução ótima foi encontrada, porém, na média o cenário 1 encontrou uma solução bem próxima. Essas

Tabela 7.1: Parametrização dos cenários

	Cenário 1	Cenário 2
Atraso Maximo	0	10
Prob. Arc. Tipo 1	0.92	0.69
Prob. Arc. Tipo 2	0	0.16
Prob. Arc. Tipo 3	0.08	0.04
Prob. Arc. Tipo 4	0	0.01

Tabela 7.2: Resultados do cenário 1

Instância	BKS	Resultado	Tempo(s)	GAP
Rio Sul	17.138	17.138	4.8	0
TAM	35.334	35.348	37	0.0004
Rio Sul Estendida	18.392	21.911	525	0.19

duas instâncias representam um horizonte de tempo de um dia. Na instância da Rio-Sul estendida que representam uma semana de operação as soluções ficaram em média 0.19 do ótimo para o cenário 1 e 0.18 no cenário 2 com um tempo aproximado de 8 minutos. Para o procedimento programação linear não foram inseridos limites previamente calculados, de modo que a ferramenta utilizou apenas a relaxação linear.

Alguns ajustes ainda podem melhorar o modelo híbrido para que ele possa se aproximar mais da solução ótima. A modificação da estrutura a ser otimizada na busca local pode ser um ponto que ajude a melhorar os resultados, pois a literatura mostra que esse é um dos pontos mais importantes de uma heurística híbrida.

Uma das grandes dificuldades encontradas no trabalho foi a falta de instâncias na literatura tornando difícil a comparação de resultados com outras abordagens. Com isso existe a necessidade de geração de um conjunto de instâncias e a sua publicação para fins comparativos.

Existe ainda a possibilidade de uma implementação paralela que ainda está em fase de planejamento e que se demonstrar resultados interessantes em tempo hábil será adicionada a dissertação.

Tabela 7.3: Resultados do cenário 2

Instância	BKS	Resultado	Tempo(s)	GAP
Rio Sul	16.158	16.158	5	0
TAM	35.015	35015	22	0
Rio Sul Estendida	17.433	20564	494	0.18

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDELGHANY, A.; ABDELGHANY, K. **Modeling Applications in the Airline Industry**. USA: Ashgate Publishing Limited, 2009.
- ARAÚJO, T. M. U. de. **Métodos híbridos baseados em continuous-GRASP aplicados à otimização global contínua**. Dissertação (Mestrado) — Universidade Federal da Paraíba, 2009.
- ARGUELLO, M. F.; BARD, J. F. A grasp for aircraft routing in response to groundings and delays. **Journal of Combinatorial Optimization**, v. 5, p. 211–228, 1997.
- BEASLEY, J. E. Or-library: distributing test problems by electronic mail. **Journal of the Operational Research Society**, v. 41, n. 11, p. 1069–1072, 1990.
- CLARKEA, L.; JOHNSONA, E.; NEMHAUSERA, G.; ZHUB, Z. The aircraft rotation problem. **Operations Research**, v. 69, p. 33–46, 1997.
- COHN, A. M.; BARNHART, C. Improving crew scheduling by incorporating key maintenance routing decisions. **Operations Research**, v. 51, n. 3, p. 387–396, May 2003.
- CORDEAU, J. F.; STOJKOVIC, G.; SOUMIS, F.; DESROSIERS, J. Benders decomposition for simultaneous aircraft routing and crew scheduling. **Transportation Science**, v. 35, p. 375–388, 2001.
- DIAS, T. **Algoritmos heurísticos e metaheurísticas híbridas aplicadas ao planejamento de uma rede de telecomunicações com topologia anel-estrela**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2006.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedure. **Journal of Global Optimization**, v. 6, n. 2, p. 109–133, 1995.
- HAOUARI, M.; SHERALI, H. D.; MANSOUR, F. Z.; AISSAOUI, N. Exact approaches for integrated aircraft fleet and routing at tunisair. **Computational Optimization and Applications**, 2011.
- KLABJAN, D.; JOHNSON, E. L.; NEMHAUSER, G. L.; GELMAN, E.; RAMASWAMY, S. Airline crew scheduling with time windows and plane-count constraints. **Transportation Science**, v. 36, n. 3, p. 337–348, 2002.

- LUCENA, A.; PONTES, R. **Aviação comercial controlada por máquinas inteligentes**. Rio de Janeiro: e-papers, 2007.
- MERCIER, A.; SOUMIS, F. An integrated aircraft routing, crew scheduling and flight retiming model. **Computers and Operations Research**, v. 34, p. 2251–2265, 2007.
- PIMENTEL, A. G. **Uma abordagem heurística para a solução de problemas de recobrimento de conjuntos de grande porte, com aplicação à alocação de tripulações para companhias aéreas**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2005.
- PONTES, R.; LUCENA, A.; CABRAL, L. **Exact and Heuristic Techniques for Solving the Aircraft Rotation Problem**. [S.l.], 2002.
- PRAIS, M.; RIBEIRO, C. Variação de parâmetros em procedimentos grasp. **Investigación Operativa**, 2000.
- SOUZA, M. J. F. **Inteligência Computacional para Otimização**. [S.l.], 2009.

A seguir temos o cronograma das atividades que foram e que serão desenvolvidas no decorrer do curso de pós graduação.

Atividades Realizadas	2010									2011								
	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez		Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
1. Pesquisa Bibliográfica																		
2. Estudo do problema																		
3. Análise sobre técnicas de otimização à utilizar																		
4. Construção da metodologia e implementação																		
5. Redação da proposta																		
6. Defesa da proposta																		
7. Redação da dissertação																		
8. Revisão dos métodos propostos																		
9. Análise crítica dos resultados																		
10. Defesa da dissertação																		

Inicialmente foi feito um levantamento bibliográfico sobre o PCTA e outros que são correlatos ou similares a ele, também foi pesquisado sobre metaheurísticas que tiveram bons resultados com esses tipos de problemas. Em seguida foi estudado a possibilidade de integração da metaheurística escolhida com alguma modelagem matemática eficiente. Após isso foi elaborado e implementado o método proposto.

Fica faltando ainda a finalização dessa implementação e a revisão do método proposto, bem como a análise crítica dos resultados. Após essa parte a redação da dissertação para posterior defesa da mesma. O mês de dezembro fica vago para o caso de ocorrer algum imprevisto no decorrer da execução do cronograma.

ANEXO A – REDE DE VOOS DA RIO-SUL

001 SL166 0 08:00 0 08:31 GYN BSB	055 SL598 0 16:00 0 17:55 CGH POA
002 SL155 0 08:32 0 10:33 GYN CGH	056 SL568 0 16:02 0 17:03 CGH PLU
003 SL330 0 08:34 0 09:59 CGH VIX	057 SL403 0 16:08 0 17:59 POA CGH
004 SL595 0 08:40 0 11:11 POA CGH	058 SL273 0 16:10 0 17:51 SJP CGH
005 SL596 0 08:44 0 10:39 CGH POA	059 SL391 0 16:12 0 17:55 FLN CGH
006 SL533 0 08:50 0 10:01 BSB PLU	060 JH343 0 16:12 0 18:57 SSA CGH
007 SL280 0 08:52 0 09:43 CGH CWB	061 SL481 0 17:00 0 17:45 CGH SDU
008 SL587 0 09:00 0 10:01 PLU CGH	062 JH506 0 17:02 0 18:33 CGH BSB
009 SL520 0 09:02 0 09:47 SDU PLU	063 SL405 0 17:14 0 19:57 CGH CGH
010 JH559 0 09:04 0 10:35 BSB CGH	064 SL569 0 17:32 0 18:33 PLU CGH
011 SL406 0 09:06 0 09:55 SDU CGH	065 SL152 0 17:44 0 19:35 CGH GYN
012 SL419 0 09:16 0 11:03 FLN CGH	066 JH546 0 17:56 0 19:27 CGH BSB
013 SL542 0 09:28 0 10:31 CGH PLU	067 SL572 0 17:58 0 18:59 CGH PLU
014 SL408 0 09:30 0 10:21 SDU CGH	068 SL482 0 18:30 0 19:21 SDU CGH
015 JH345 0 09:30 0 12:23 SSA CGH	069 JH341 0 18:50 1 00:19 JPA CGH
016 SL589 0 09:58 0 11:01 PLU CGH	070 SL592 0 18:54 0 20:45 CGH POA
017 JH502 0 10:02 0 11:33 CGH BSB	071 JH507 0 19:00 0 20:31 BSB CGH
018 SL281 0 10:08 0 10:59 CWB CGH	072 SL528 0 19:02 0 19:47 SDU PLU
019 SL521 0 10:16 0 11:01 PLU SDU	073 JH550 0 19:04 0 20:35 CGH BSB
020 SL532 0 10:20 0 11:31 PLU BSB	074 SL537 0 19:22 0 20:33 BSB PLU
021 SL510 0 10:22 0 11:13 CGH CWB	075 SL576 0 19:26 0 20:27 CGH PLU
022 SL544 0 10:32 0 11:33 CGH PLU	076 SL573 0 19:34 0 20:35 PLU CGH
023 SL331 0 10:40 0 12:11 VIX CGH	077 SL599 0 19:46 0 21:41 POA CGH
024 SL409 0 11:00 0 11:49 CGH SDU	078 JH547 0 20:00 0 21:31 BSB CGH
025 SL543 0 11:00 0 12:03 PLU CGH	079 SL153 0 20:04 0 22:05 GYN CGH
026 SL590 0 11:06 0 12:44 CGH POA	080 SL483 0 20:08 0 20:53 CGH SDU
027 SL336 0 11:16 0 15:05 CGH REC	081 SL412 0 20:10 0 21:29 CGH FLN
028 SL597 0 11:30 0 13:27 POA CGH	082 SL529 0 20:14 0 20:59 PLU SDU
029 SL522 0 11:32 0 12:17 SDU PLU	083 SL584 0 20:26 0 21:27 CGH PLU
030 SL150 0 11:42 0 13:33 CGH GYN	084 SL514 0 20:30 0 21:21 CGH CWB
031 SL511 0 11:50 0 12:41 CWB CGH	085 JH344 0 20:48 0 23:39 CGH SSA
032 JH503 0 12:02 0 13:33 BSB CGH	086 SL577 0 21:02 0 22:03 PLU CGH
033 SL545 0 12:04 0 13:05 PLU CGH	087 SL536 0 21:04 0 22:13 PLU BSB
034 SL470 0 12:28 0 13:19 SDU CGH	088 JH551 0 21:04 0 22:35 BSB CGH
035 SL523 0 12:46 0 13:31 PLU SDU	089 SL332 0 21:20 0 22:31 CGH VIX
036 JH342 0 13:00 0 15:45 CGH SSA	090 SL492 0 21:30 0 22:21 SDU CGH
037 JH500 0 13:02 0 14:33 CGH BSB	091 SL586 0 21:30 0 22:31 CGH PLU
038 SL147 0 13:10 0 18:03 BSB BSB	092 SL530 0 21:34 0 22:19 SDU PLU
039 JH340 0 13:22 0 18:25 CGH JPA	093 SL515 0 21:45 0 22:37 CWB CGH
040 SL591 0 13:30 0 14:57 POA CGH	094 SL593 0 21:52 0 23:19 POA CGH

041 SL402 0 13:38 0 15:33 CGH POA	095 SL585 0 22:04 0 23:05 PLU CGH
042 SL471 0 13:54 0 14:39 CGH SDU	096 SL518 0 22:16 0 23:07 CGH CWB
043 SL390 0 13:56 0 15:41 CGH FLN	097 SL413 0 22:22 0 23:31 FLN CGH
044 SL564 0 13:58 0 14:59 CGH PLU	098 SL407 0 22:24 0 23:09 CGH SDU
045 SL151 0 13:58 0 15:59 GYN CGH	099 SL167 0 22:42 0 23:13 BSB GYN
046 SL563 0 14:00 0 15:01 PLU CGH	100 SL531 0 22:48 0 23:33 PLU SDU
047 SL272 0 14:12 0 15:51 CGH SJP	101 JH558 0 22:56 1 00:27 CGH BSB
048 SL282 0 14:22 0 15:13 CGH CWB	102 SL493 0 23:02 0 23:47 CGH SDU
049 SL508 0 14:56 0 15:41 SDU PLU	103 SL333 0 23:04 1 00:15 VIX CGH
050 JH501 0 14:58 0 16:29 BSB CGH	104 SL588 0 23:06 1 00:07 CGH PLU
051 SL480 0 15:30 0 16:21 SDU CGH	105 SL594 0 23:18 1 01:57 CGH POA
052 SL283 0 15:42 0 16:37 CWB CGH	106 SL154 0 23:22 1 01:13 CGH GYN
053 SL337 0 15:46 0 19:29 REC CGH	107 SL418 0 23:32 1 01:19 CGH FLN
054 SL347 0 16:00 0 16:45 PLU SDU	

ANEXO B – TEMPO DOS VOOS DA RIO-SUL

BSB CGH 0091	CGH SJP 0049	FLN SSA 0225	PLU SDU 0045
BSB CWB 0125	CGH SSA 0171	FLN VIX 0136	PLU SJP 0067
BSB FLN 0152	CGH VIX 0087	GYN JPA 0216	PLU SSA 0113
BSB GYN 0031	CWB FLN 0028	GYN PLU 0076	PLU VIX 0044
BSB JPA 0197	CWB GYN 0114	GYN POA 0173	POA REC 0342
BSB PLU 0071	CWB JPA 0293	GYN REC 0210	POA SDU 0129
BSB POA 0185	CWB PLU 0095	GYN SDU 0109	POA SJP 0119
BSB REC 0191	CWB POA 0062	GYN SJP 0054	POA SSA 0267
BSB SDU 0107	CWB REC 0283	GYN SSA 0143	POA VIX 0177
BSB SJP 0066	CWB SDU 0078	GYN VIX 0118	REC SDU 0215
BSB SSA 0125	CWB SJP 0060	JPA PLU 0198	REC SJP 0242
BSB VIX 0109	CWB SSA 0208	JPA POA 0352	REC SSA 0075
CGH CWB 0051	CWB VIX 0125	JPA REC 0013	REC VIX 0169
CGH FLN 0105	FLN GYN 0142	JPA SDU 0226	SDU SJP 0079
CGH GYN 0095	FLN JPA 0311	JPA SJP 0251	SDU SSA 0130
CGH JPA 0255	FLN PLU 0114	JPA SSA 0085	SDU VIX 0048
CGH PLU 0058	FLN POA 0042	JPA VIX 0181	SJP SSA 0169
CGH POA 0097	FLN REC 0300	PLU POA 0155	SJP VIX 0110
CGH REC 0246	FLN SDU 0087	PLU REC 0187	SSA VIX 0097
CGH SDU 0049	FLN SJP 0088		

ANEXO C – RESULTADO ÓTIMO DA INSTÂNCIA RIO-SUL

Valor da função objetivo: 16158

Rota[01 - 8]

SL166 0 08:00 0 08:31 GYN BSB
JH559 0 09:04 0 10:35 BSB CGH
SL590 0 11:06 0 12:44 CGH POA
SL591 0 13:30 0 14:57 POA CGH
SL568 0 16:02 0 17:03 CGH PLU
SL569 0 17:32 0 18:33 PLU CGH
SL592 0 18:54 0 20:45 CGH POA
SL593 0 21:52 0 23:19 POA CGH

Rota[02 - 7]

SL155 0 08:32 0 10:33 GYN CGH
SL409 0 11:00 0 11:49 CGH SDU
SL470 0 12:28 0 13:19 SDU CGH
SL564 0 13:58 0 14:59 CGH PLU
SL347 0 16:00 0 16:45 PLU SDU
SL482 0 18:30 0 19:21 SDU CGH
JH344 0 20:48 0 23:39 CGH SSA

Rota[03 - 4]

SL330 0 08:34 0 09:59 CGH VIX
SL331 0 10:40 0 12:11 VIX CGH
JH340 0 13:22 0 18:25 CGH JPA
JH341 0 18:50 1 00:19 JPA CGH

Rota[04 - 7]

SL595 0 08:40 0 11:11 POA CGH
JH500 0 13:02 0 14:33 CGH BSB
JH501 0 14:58 0 16:29 BSB CGH
SL572 0 17:58 0 18:59 CGH PLU
SL573 0 19:34 0 20:35 PLU CGH
SL332 0 21:20 0 22:31 CGH VIX
SL333 0 23:04 1 00:15 VIX CGH

Rota[05 - 7]

SL596 0 08:44 0 10:39 CGH POA
SL597 0 11:30 0 13:27 POA CGH
SL282 0 14:22 0 15:13 CGH CWB
SL283 0 15:42 0 16:37 CWB CGH
SL152 0 17:44 0 19:35 CGH GYN
SL153 0 20:04 0 22:05 GYN CGH
JH558 0 22:56 1 00:27 CGH BSB

Rota[06 - 5]

SL533 0 08:50 0 10:01 BSB PLU(-1)
SL532 0 10:20 0 11:31 PLU BSB
SL147 0 13:10 0 18:03 BSB BSB
JH507 0 19:00 0 20:31 BSB CGH
SL586 0 21:30 0 22:31 CGH PLU

Rota[07 - 8]

SL280 0 08:52 0 09:43 CGH CWB
SL281 0 10:08 0 10:59 CWB CGH
SL150 0 11:42 0 13:33 CGH GYN
SL151 0 13:58 0 15:59 GYN CGH
SL405 0 17:14 0 19:57 CGH CGH
SL514 0 20:30 0 21:21 CGH CWB
SL515 0 21:45 0 22:37 CWB CGH
SL154 0 23:22 1 01:13 CGH GYN

Rota[08 - 8]

SL587 0 09:00 0 10:01 PLU CGH
SL544 0 10:32 0 11:33 CGH PLU
SL545 0 12:04 0 13:05 PLU CGH
SL402 0 13:38 0 15:33 CGH POA
SL403 0 16:08 0 17:59 POA CGH
SL483 0 20:08 0 20:53 CGH SDU
SL530 0 21:34 0 22:19 SDU PLU
SL531 0 22:48 0 23:33 PLU SDU

Rota[09 - 9]

SL520 0 09:02 0 09:47 SDU PLU
SL521 0 10:16 0 11:01 PLU SDU
SL522 0 11:32 0 12:17 SDU PLU
SL523 0 12:46 0 13:31 PLU SDU
SL508 0 14:56 0 15:41 SDU PLU
REPO 0 16:01 0 16:59 PLU CGH
JH546 0 17:56 0 19:27 CGH BSB
JH547 0 20:00 0 21:31 BSB CGH
SL493 0 23:02 0 23:47 CGH SDU

Rota[10 - 7]

SL406 0 09:06 0 09:55 SDU CGH
SL510 0 10:22 0 11:13 CGH CWB
SL511 0 11:50 0 12:41 CWB CGH
SL272 0 14:12 0 15:51 CGH SJP
SL273 0 16:10 0 17:51 SJP CGH(+1)
SL412 0 20:10 0 21:29 CGH FLN
SL413 0 22:22 0 23:31 FLN CGH

Rota[11 - 6]

SL419 0 09:16 0 11:03 FLN CGH
REPO 0 11:23 0 12:21 CGH PLU
SL563 0 14:00 0 15:01 PLU CGH
SL598 0 16:00 0 17:55 CGH POA
SL599 0 19:46 0 21:41 POA CGH
SL518 0 22:16 0 23:07 CGH CWB

Rota[12 - 7]

SL542 0 09:28 0 10:31 CGH PLU
SL543 0 11:00 0 12:03 PLU CGH
SL390 0 13:56 0 15:41 CGH FLN
SL391 0 16:12 0 17:55 FLN CGH
JH550 0 19:04 0 20:35 CGH BSB
JH551 0 21:04 0 22:35 BSB CGH
SL594 0 23:18 1 01:57 CGH POA

Rota[13 - 6]

SL408 0 09:30 0 10:21 SDU CGH
SL336 0 11:16 0 15:05 CGH REC
SL337 0 15:46 0 19:29 REC CGH
SL584 0 20:26 0 21:27 CGH PLU
SL585 0 22:04 0 23:05 PLU CGH
SL418 0 23:32 1 01:19 CGH FLN

Rota[14 - 6]

JH345 0 09:30 0 12:23 SSA CGH
JH342 0 13:00 0 15:45 CGH SSA
JH343 0 16:12 0 18:57 SSA CGH
SL576 0 19:26 0 20:27 CGH PLU
SL577 0 21:02 0 22:03 PLU CGH
SL407 0 22:24 0 23:09 CGH SDU

Rota[15 - 8]

SL589 0 09:58 0 11:01 PLU CGH
SL471 0 13:54 0 14:39 CGH SDU
SL480 0 15:30 0 16:21 SDU CGH
SL481 0 17:00 0 17:45 CGH SDU
SL528 0 19:02 0 19:47 SDU PLU
SL529 0 20:14 0 20:59 PLU SDU
SL492 0 21:30 0 22:21 SDU CGH
SL588 0 23:06 1 00:07 CGH PLU

ANEXO D – REDE DE VOOS DA TAM

001 JJ3458 0 00:05 0 01:05 MAB BEL	122 JJ3928 0 13:00 0 14:00 CGH SDU
002 JJ3585 0 01:10 0 06:21 RBR BSB	123 JJ3871 0 13:15 0 14:15 BEL MAB
003 JJ3585 0 01:10 0 06:21 RBR BSB	124 JJ3365 0 13:15 0 16:00 IOS SDU
004 JJ3595 0 01:30 0 06:16 PVH BSB	125 JJ3929 0 13:15 0 14:07 SDU CGH
005 JJ3459 0 01:35 0 06:25 BEL CNF	126 JJ3571 0 13:25 0 15:56 CGR BSB
006 JJ3459 0 01:35 0 02:35 BEL MAB	127 JJ3038 0 13:25 0 14:29 JOI CGH
007 JJ3069 0 01:50 0 02:51 CGB CGR	128 JJ3563 0 13:30 0 18:43 RBR BSB
008 JJ3739 0 02:30 0 05:59 FOR BSB	129 JJ3563 0 13:30 0 18:43 RBR BSB
009 JJ2101 0 02:40 0 04:10 NAT SSA	130 JJ3930 0 13:30 0 14:27 CGH SDU
010 JJ3201 0 02:40 0 07:20 NAT CNF	131 JJ3815 0 13:35 0 15:51 PMW BSB
011 JJ3201 0 02:40 0 09:22 NAT CGH	132 JJ3708 0 13:43 0 15:19 CGH BSB
012 JJ3065 0 02:50 0 06:22 AJU SDU	133 JJ3931 0 13:45 0 14:35 SDU CGH
013 JJ3459 0 03:10 0 06:25 MAB CNF	134 JJ3759 0 13:50 0 14:52 CNF SDU
014 JJ3409 0 03:23 0 05:27 BPS CNF	135 JJ3745 0 13:55 0 14:56 SJP CGH
015 JJ3069 0 03:24 0 06:24 CGR SDU	136 JJ3120 0 14:00 0 15:02 NVT CGH
016 JJ3775 0 03:35 0 06:03 CGR CGH	137 JJ3932 0 14:00 0 15:00 CGH SDU
017 JJ3201 0 04:45 0 07:20 SSA CNF	138 JJ3661 0 14:12 0 17:14 IOS CGH
018 JJ3201 0 04:45 0 09:22 SSA CGH	139 JJ3933 0 14:16 0 15:14 SDU CGH
019 JJ3737 0 05:50 0 06:45 SJP CGH	140 JJ3934 0 14:28 0 15:30 CGH SDU
020 JJ4723 0 06:00 0 07:43 BSB CGH	141 JJ3264 0 14:30 0 15:53 CWB SDU
021 JJ3597 0 06:00 0 08:42 CGB BSB	142 JJ3935 0 14:45 0 15:44 SDU CGH
022 JJ3770 0 06:00 0 07:00 GRU IOS	143 JJ3628 0 14:54 0 16:13 CGH SSA
023 JJ3900 0 06:04 0 07:04 CGH SDU	144 JJ4737 0 14:55 0 18:02 CGB CGH
024 JJ3100 0 06:05 0 07:15 FLN CGH	145 JJ3936 0 15:00 0 16:00 CGH SDU
025 JJ3211 0 06:07 0 07:21 CNF CGH	146 JJ3077 0 15:15 0 19:15 REC GIG
026 JJ3901 0 06:15 0 07:07 SDU CGH	147 JJ3077 0 15:15 0 21:11 REC CGH
027 JJ3768 0 06:20 0 07:27 LDB CGH	148 JJ3937 0 15:15 0 16:11 SDU CGH
028 JJ3119 0 06:24 0 07:35 CGH NVT	149 JJ3107 0 15:17 0 16:20 CGH FLN
029 JJ3902 0 06:30 0 07:30 CGH SDU	150 JJ3027 0 15:18 0 17:04 BSB SDU
030 JJ3758 0 06:34 0 07:47 SDU CNF	151 JJ3410 0 15:22 0 16:20 GIG VIX
031 JJ3903 0 06:45 0 07:49 SDU CGH	152 JJ3938 0 15:30 0 16:32 CGH SDU
032 JJ3242 0 06:47 0 07:50 CGH UDI	153 JJ3013 0 15:35 0 16:27 CGH CWB
033 JJ3035 0 06:49 0 08:00 CGH JOI	154 JJ3744 0 15:42 0 16:40 CGH SJP
034 JJ3370 0 06:52 0 07:49 CGH GIG	155 JJ3826 0 15:43 0 17:19 SDU BSB
035 JJ3370 0 06:52 0 10:27 CGH REC	156 JJ3939 0 15:45 0 16:35 SDU CGH
036 JJ3732 0 06:52 0 08:12 CGH SSA	157 JJ3652 0 15:57 0 16:40 CGH CGR
037 JJ3753 0 07:00 0 08:04 CNF SDU	158 JJ3940 0 16:01 0 17:02 CGH SDU
038 JJ3666 0 07:00 0 07:45 SDU IOS	159 JJ3723 0 16:03 0 17:38 BSB CGH
039 JJ3904 0 07:00 0 08:02 CGH SDU	160 JJ3941 0 16:15 0 17:11 SDU CGH
040 JJ3022 0 07:06 0 08:56 SDU BSB	161 JJ3029 0 16:22 0 17:58 BSB SDU

041 JJ3855 0 07:10 0 08:15 BSB CNF	162 JJ3942 0 16:30 0 17:28 CGH SDU
042 JJ3023 0 07:12 0 09:02 BSB SDU	163 JJ3857 0 16:41 0 17:52 BSB CNF
043 JJ3905 0 07:16 0 08:13 SDU CGH	164 JJ3943 0 16:45 0 17:34 SDU CGH
044 JJ3906 0 07:30 0 08:29 CGH SDU	165 JJ3263 0 16:47 0 18:55 SDU POA
045 JJ3907 0 07:45 0 08:51 SDU CGH	166 JJ3629 0 16:52 0 20:15 SSA CGH
046 JJ3740 0 07:55 0 08:55 CGH SJP	167 JJ3104 0 16:55 0 17:56 FLN CGH
047 JJ3740 0 07:55 0 10:05 CGH CGB	168 JJ3267 0 16:56 0 18:30 SDU CWB
048 JJ3201 0 08:00 0 09:22 CNF CGH	169 JJ3944 0 16:59 0 18:06 CGH SDU
049 JJ3908 0 08:00 0 09:00 CGH SDU	170 JJ3966 0 16:59 0 18:06 CGH SDU
050 JJ3130 0 08:12 0 09:35 CGH VIX	171 JJ3137 0 17:00 0 18:28 VIX CGH
051 JJ3667 0 08:15 0 11:00 IOS SDU	172 JJ3012 0 17:02 0 17:50 CWB CGH
052 JJ3118 0 08:15 0 09:09 NVT CGH	173 JJ3123 0 17:07 0 18:08 CGH NVT
053 JJ3909 0 08:15 0 09:15 SDU CGH	174 JJ3945 0 17:15 0 18:08 SDU CGH
054 JJ3411 0 08:15 0 09:22 VIX GIG	175 JJ3743 0 17:15 0 18:14 SJP CGH
055 JJ3269 0 08:18 0 09:48 GIG CWB	176 JJ3946 0 17:29 0 18:32 CGH SDU
056 JJ3385 0 08:30 0 09:38 CNF GIG	177 JJ3947 0 17:44 0 18:34 SDU CGH
057 JJ3910 0 08:30 0 09:30 CGH SDU	178 JJ3653 0 17:50 0 20:29 CGR CGH
058 JJ3239 0 08:30 0 09:41 UDI CGH	179 JJ3028 0 17:54 0 19:43 SDU BSB
059 JJ3032 0 08:40 0 09:52 JOI CGH	180 JJ3109 0 17:58 0 19:10 CGH FLN
060 JJ3261 0 08:42 0 10:27 SDU BSB	181 JJ3948 0 18:00 0 19:04 CGH SDU
061 JJ3911 0 08:44 0 09:55 SDU CGH	182 JJ3949 0 18:15 0 19:05 SDU CGH
062 JJ3911 0 08:44 0 09:55 SDU CGH	183 JJ3767 0 18:23 0 19:25 CGH LDB
063 JJ3370 0 08:45 0 10:27 GIG REC	184 JJ3755 0 18:26 0 19:26 CNF SDU
064 JJ3756 0 08:49 0 09:55 SDU CNF	185 JJ3950 0 18:29 0 19:30 CGH SDU
065 JJ3733 0 08:54 0 12:09 SSA CGH	186 JJ3827 0 18:32 0 20:01 BSB GIG
066 JJ3850 0 08:59 0 10:14 CNF BSB	187 JJ3224 0 18:38 0 19:55 CGH CNF
067 JJ3912 0 09:00 0 10:00 CGH SDU	188 JJ3951 0 18:46 0 19:37 SDU CGH
068 JJ3913 0 09:15 0 10:21 SDU CGH	189 JJ3033 0 18:46 0 19:45 CGH JOI
069 JJ3914 0 09:30 0 10:30 CGH SDU	190 JJ3122 0 18:48 0 19:49 NVT CGH
070 JJ3740 0 09:35 0 10:05 SJP CGB	191 JJ3966 0 18:48 0 19:15 SDU CFB
071 JJ3024 0 09:42 0 11:27 SDU BSB	192 JJ3952 0 19:00 0 20:06 CGH SDU
072 JJ3915 0 09:45 0 10:52 SDU CGH	193 JJ3238 0 19:04 0 20:10 CGH UDI
073 JJ3025 0 09:52 0 11:37 BSB SDU	194 JJ3712 0 19:12 0 20:58 CGH BSB
074 JJ3127 0 10:00 0 11:29 VIX CGH	195 JJ3268 0 19:14 0 20:46 CWB GIG
075 JJ3411 0 10:02 0 12:05 GIG POA	196 JJ3953 0 19:15 0 20:15 SDU CGH
076 JJ3916 0 10:02 0 11:00 CGH SDU	197 JJ3276 0 19:24 0 20:22 CGH RAO
077 JJ3372 0 10:04 0 12:12 CGH SSA	198 JJ3954 0 19:29 0 20:28 CGH SDU
078 JJ3660 0 10:04 0 10:58 CGH IOS	199 JJ3563 0 19:30 0 21:05 BSB GRU
079 JJ3660 0 10:04 0 12:12 CGH SSA	200 JJ3262 0 19:30 0 21:28 POA SDU
080 JJ3212 0 10:06 0 11:10 CGH CNF	201 JJ3955 0 19:44 0 20:38 SDU CGH
081 JJ3604 0 10:06 0 11:25 CGH SSA	202 JJ3967 0 19:45 0 20:12 CFB SDU
082 JJ3917 0 10:15 0 11:12 SDU CGH	203 JJ3110 0 19:50 0 21:05 FLN CGH
083 JJ3053 0 10:16 0 11:37 CGH POA	204 JJ3752 0 19:54 0 20:54 SDU CNF

084 JJ3856 0 10:29 0 11:43 CNF BSB	205 JJ3956 0 19:59 0 21:03 CGH SDU
085 JJ3820 0 10:30 0 12:20 GIG BSB	206 JJ3764 0 20:05 0 21:11 LDB CGH
086 JJ3918 0 10:30 0 11:28 CGH SDU	207 JJ4722 0 20:05 0 21:44 CGH BSB
087 JJ3246 0 10:32 0 11:39 CGH UDI	208 JJ3957 0 20:15 0 21:08 SDU CGH
088 JJ3266 0 10:36 0 11:57 CWB SDU	209 JJ3226 0 20:19 0 21:36 CGH CNF
089 JJ3745 0 10:45 0 13:15 CGB SJP	210 JJ3226 0 20:19 0 22:59 CGH SSA
090 JJ3745 0 10:45 0 14:56 CGB CGH	211 JJ3226 0 20:19 1 01:05 CGH NAT
091 JJ3919 0 10:45 0 11:48 SDU CGH	212 JJ3034 0 20:20 0 21:17 JOI CGH
092 JJ3920 0 11:00 0 12:00 CGH SDU	213 JJ3757 0 20:33 0 21:34 CNF SDU
093 JJ3574 0 11:04 0 12:20 BSB RBR	214 JJ3030 0 20:37 0 22:23 SDU BSB
094 JJ3825 0 11:12 0 12:50 BSB SDU	215 JJ3959 0 20:45 0 21:37 SDU CGH
095 JJ3921 0 11:16 0 12:24 SDU CGH	216 JJ3967 0 20:45 0 21:37 SDU CGH
096 JJ3922 0 11:30 0 12:25 CGH SDU	217 JJ3245 0 20:45 0 21:55 UDI CGH
097 JJ3660 0 11:33 0 12:12 IOS SSA	218 JJ3031 0 20:46 0 22:05 BSB SDU
098 JJ3923 0 11:45 0 12:48 SDU CGH	219 JJ3105 0 20:46 0 21:53 CGH FLN
099 JJ3215 0 11:50 0 12:57 CNF CGH	220 JJ3958 0 21:00 0 21:55 CGH SDU
100 JJ3039 0 11:50 0 12:50 CGH JOI	221 JJ3275 0 21:02 0 22:04 RAO CGH
101 JJ3364 0 12:00 0 12:45 SDU IOS	222 JJ3961 0 21:16 0 22:15 SDU CGH
102 JJ3605 0 12:00 0 15:34 SSA CGH	223 JJ3854 0 21:28 0 22:44 CNF BSB
103 JJ3924 0 12:00 0 13:00 CGH SDU	224 JJ3960 0 21:29 0 22:19 CGH SDU
104 JJ3754 0 12:04 0 13:03 SDU CNF	225 JJ3769 0 21:42 0 22:40 CGH LDB
105 JJ3260 0 12:06 0 13:31 BSB SDU	226 JJ3736 0 21:56 0 22:57 CGH SJP
106 JJ3243 0 12:10 0 13:11 UDI CGH	227 JJ3077 0 22:02 0 23:11 GIG CGH
107 JJ3925 0 12:15 0 13:14 SDU CGH	228 JJ3064 0 22:07 0 23:40 SDU AJU
108 JJ3265 0 12:19 0 13:45 SDU CWB	229 JJ3774 0 22:07 0 22:47 CGH CGR
109 JJ3570 0 12:21 0 12:50 BSB CGR	230 JJ3068 0 22:09 0 23:20 SDU CGR
110 JJ3052 0 12:22 0 13:48 POA CGH	231 JJ3226 0 22:20 0 22:59 CNF SSA
111 JJ3121 0 12:22 0 13:25 CGH NVT	232 JJ3226 0 22:20 1 01:05 CNF NAT
112 JJ3926 0 12:30 0 13:33 CGH SDU	233 JJ3177 0 22:20 0 23:35 GRU FLN
113 JJ3814 0 12:41 0 12:55 BSB PMW	234 JJ3458 0 22:25 0 23:35 CNF MAB
114 JJ3026 0 12:43 0 14:28 SDU BSB	235 JJ3458 0 22:25 1 01:05 CNF BEL
115 JJ3927 0 12:45 0 13:41 SDU CGH	236 JJ3584 0 23:11 1 00:20 BSB RBR
116 JJ3410 0 12:47 0 14:41 POA GIG	237 JJ3594 0 23:26 1 00:25 BSB PVH
117 JJ3373 0 12:52 0 17:14 SSA CGH	238 JJ3226 0 23:35 1 01:05 SSA NAT
118 JJ3661 0 12:52 0 13:37 SSA IOS	239 JJ3738 0 23:47 1 01:20 BSB FOR
119 JJ3661 0 12:52 0 17:14 SSA CGH	240 JJ3408 0 23:50 1 00:59 CNF BPS
120 JJ3963 0 12:58 0 14:01 SDU CGH	241 JJ3068 0 23:55 1 00:55 CGR CGB
121 JJ3709 0 13:00 0 14:38 BSB CGH	

ANEXO E – TEMPO DOS VOOS DA TAM

AJU SDU 0212	BSB SDU 0110	CGH NVT 0071	CNF SDU 0073
BEL CNF 0290	CFB SDU 0027	CGH POA 0086	CNF SSA 0155
BEL MAB 0060	CGB CGH 0251	CGH RAO 0062	CWB GIG 0092
BPS CNF 0124	CGB CGR 0061	CGH REC 0356	CWB SDU 0094
BSB CGB 0162	CGB SJP 0150	CGH SDU 0071	FLN GRU 0075
BSB CGH 0106	CGH CGR 0159	CGH SJP 0061	GIG POA 0123
BSB CGR 0151	CGH CNF 0082	CGH SSA 0277	GIG REC 0240
BSB CNF 0076	CGH CWB 0052	CGH UDI 0071	GIG VIX 0067
BSB FOR 0209	CGH FLN 0075	CGH VIX 0089	GRU IOS 0060
BSB GIG 0110	CGH GIG 0069	CGR SDU 0180	IOS SDU 0165
BSB GRU 0095	CGH IOS 0182	CNF GIG 0068	IOS SSA 0045
BSB PMW 0136	CGH JOI 0072	CNF MAB 0195	NAT SSA 0090
BSB PVH 0286	CGH LDB 0067	CNF NAT 0280	POA SDU 0128
BSB RBR 0313	CGH NAT 0402		

*O tempo de voos entre alguns aeroportos foram omitidos por falta de informação suficiente para inferi-las.

ANEXO F – RESULTADO ÓTIMO DA INSTÂNCIA TAM

Valor da função objetivo: 35015

Rota[01 - 12]

JJ3458 0 00:05 0 01:05 MAB BEL
 JJ3459 0 01:35 0 02:35 BEL MAB
 JJ3459 0 03:10 0 06:25 MAB CNF
 JJ3753 0 07:00 0 08:04 CNF SDU
 JJ3911 0 08:44 0 09:55 SDU CGH
 JJ3246 0 10:32 0 11:39 CGH UDI
 JJ3243 0 12:10 0 13:11 UDI CGH
 JJ3708 0 13:43 0 15:19 CGH BSB
 JJ3723 0 16:03 0 17:38 BSB CGH
 JJ3224 0 18:38 0 19:55 CGH CNF
 JJ3226 0 22:20 0 22:59 CNF SSA
 JJ3226 0 23:35 1 01:05 SSA NAT

Rota[04 - 9]

JJ3595 0 01:30 0 06:16 PVH BSB
REPO 0 06:36 0 08:26 BSB SDU
 JJ3911 0 08:44 0 09:55 SDU CGH
 JJ3918 0 10:30 0 11:28 CGH SDU
 JJ3364 0 12:00 0 12:45 SDU IOS
 JJ3365 0 13:15 0 16:00 IOS SDU
 JJ3945 0 17:15 0 18:08 SDU CGH
 JJ3226 0 20:19 0 21:36 CGH CNF
 JJ3458 0 22:25 0 23:35 CNF MAB

Rota[07 - 9]

JJ3739 0 02:30 0 05:59 FOR BSB
REPO 0 06:19 0 08:09 BSB SDU
 JJ3261 0 08:42 0 10:27 SDU BSB
 JJ3570 0 12:21 0 12:50 BSB CGR
 JJ3571 0 13:25 0 15:56 CGR BSB
 JJ3857 0 16:41 0 17:52 BSB CNF
 JJ3755 0 18:26 0 19:26 CNF SDU
 JJ3752 0 19:54 0 20:54 SDU CNF
 JJ3226 0 22:20 1 01:05 CNF NAT

Rota[10 - 5]

Rota[02 - 7]

JJ3585 0 01:10 0 06:21 RBR BSB
 JJ3855 0 07:10 0 08:15 BSB CNF
 JJ3850 0 08:59 0 10:14 CNF BSB
 JJ3574 0 11:04 0 12:20 BSB RBR
 JJ3563 0 13:30 0 18:43 RBR BSB
 JJ3563 0 19:30 0 21:05 BSB GRU
 JJ3177 0 22:20 0 23:35 GRU FLN

Rota[05 - 5]

JJ3459 0 01:35 0 06:25 BEL CNF
REPO 0 06:45 0 07:53 CNF GIG
 JJ3370 0 08:45 0 10:27 GIG REC
 JJ3077 0 15:15 0 21:11 REC CGH
 JJ3736 0 21:56 0 22:57 CGH SJP

Rota[08 - 7]

JJ2101 0 02:40 0 04:10 NAT SSA
 JJ3201 0 04:45 0 09:22 SSA CGH
 JJ3604 0 10:06 0 11:25 CGH SSA
 JJ3605 0 12:00 0 15:34 SSA CGH
 JJ3946 0 17:29 0 18:32 CGH SDU
 JJ3953 0 19:15 0 20:15 SDU CGH
 JJ3105 0 20:46 0 21:53 CGH FLN

Rota[11 - 10]

Rota[03 - 9]

JJ3585 0 01:10 0 06:21 RBR BSB
 JJ3023 0 07:12 0 09:02 BSB SDU
 JJ3915 0 09:45 0 10:52 SDU CGH
 JJ3922 0 11:30 0 12:25 CGH SDU
 JJ3927 0 12:45 0 13:41 SDU CGH
 JJ3934 0 14:28 0 15:30 CGH SDU
 JJ3943 0 16:45 0 17:34 SDU CGH
 JJ3948 0 18:00 0 19:04 CGH SDU
 JJ3957 0 20:15 0 21:08 SDU CGH

Rota[06 - 10]

JJ3069 0 01:50 0 02:51 CGB CGR
 JJ3069 0 03:24 0 06:24 CGR SDU
 JJ3666 0 07:00 0 07:45 SDU IOS
REPO 0 08:05 0 10:50 IOS SDU
 JJ3921 0 11:16 0 12:24 SDU CGH
 JJ3928 0 13:00 0 14:00 CGH SDU
 JJ3935 0 14:45 0 15:44 SDU CGH
 JJ3033 0 18:46 0 19:45 CGH JOI
 JJ3034 0 20:20 0 21:17 JOI CGH
 JJ3769 0 21:42 0 22:40 CGH LDB

Rota[09 - 9]

JJ3201 0 02:40 0 07:20 NAT CNF
 JJ3201 0 08:00 0 09:22 CNF CGH
 JJ3916 0 10:02 0 11:00 CGH SDU
 JJ3925 0 12:15 0 13:14 SDU CGH
 JJ3932 0 14:00 0 15:00 CGH SDU
 JJ3263 0 16:47 0 18:55 SDU POA
 JJ3262 0 19:30 0 21:28 POA SDU
 JJ3068 0 22:09 0 23:20 SDU CGR
 JJ3068 0 23:55 1 00:55 CGR CGB

Rota[12 - 7]

JJ3201 0 02:40 0 09:22 NAT CGH
 JJ3372 0 10:04 0 12:12 CGH SSA
 JJ3661 0 12:52 0 17:14 SSA CGH
 JJ3950 0 18:29 0 19:30 CGH SDU
 JJ3959 0 20:45 0 21:37 SDU CGH

JJ3065 0 02:50 0 06:22 AJU SDU
 JJ3903 0 06:45 0 07:49 SDU CGH
 JJ3910 0 08:30 0 09:30 CGH SDU
 JJ3917 0 10:15 0 11:12 SDU CGH
 JJ3039 0 11:50 0 12:50 CGH JOI
 JJ3038 0 13:25 0 14:29 JOI CGH
 JJ3013 0 15:35 0 16:27 CGH CWB
 JJ3012 0 17:02 0 17:50 CWB CGH
 JJ3238 0 19:04 0 20:10 CGH UDI
 JJ3245 0 20:45 0 21:55 UDI CGH

JJ3409 0 03:23 0 05:27 BPS CNF
 JJ3211 0 06:07 0 07:21 CNF CGH
 JJ3740 0 07:55 0 10:05 CGH CGB
 JJ3745 0 10:45 0 14:56 CGB CGH
 JJ3966 0 16:59 0 18:06 CGH SDU
REPO 0 18:26 0 19:39 SDU CNF
 JJ3757 0 20:33 0 21:34 CNF SDU

Rota[13 - 9]

JJ3775 0 03:35 0 06:03 CGR CGH
 JJ3242 0 06:47 0 07:50 CGH UDI
 JJ3239 0 08:30 0 09:41 UDI CGH
 JJ3660 0 10:04 0 10:58 CGH IOS
 JJ3660 0 11:33 0 12:12 IOS SSA
 JJ3661 0 12:52 0 13:37 SSA IOS
 JJ3661 0 14:12 0 17:14 IOS CGH
 JJ3767 0 18:23 0 19:25 CGH LDB
 JJ3764 0 20:05 0 21:11 LDB CGH

Rota[14 - 7]

JJ3201 0 04:45 0 07:20 SSA CNF
REPO 0 07:40 0 08:53 CNF SDU
 JJ3913 0 09:15 0 10:21 SDU CGH
 JJ3920 0 11:00 0 12:00 CGH SDU
 JJ3963 0 12:58 0 14:01 SDU CGH
 JJ3628 0 14:54 0 16:13 CGH SSA
 JJ3629 0 16:52 0 20:15 SSA CGH

Rota[15 - 6]

JJ3737 0 05:50 0 06:45 SJP CGH(-5)
 JJ3904 0 07:00 0 08:02 CGH SDU
 JJ3909 0 08:15 0 09:15 SDU CGH(+7)
 JJ3660 0 10:04 0 12:12 CGH SSA
 JJ3373 0 12:52 0 17:14 SSA CGH
 JJ3226 0 20:19 0 22:59 CGH SSA

Rota[16 - 8]

JJ4723 0 06:00 0 07:43 BSB CGH
 JJ3130 0 08:12 0 09:35 CGH VIX
 JJ3127 0 10:00 0 11:29 VIX CGH
 JJ3924 0 12:00 0 13:00 CGH SDU
 JJ3931 0 13:45 0 14:35 SDU CGH
 JJ3942 0 16:30 0 17:28 CGH SDU
 JJ3951 0 18:46 0 19:37 SDU CGH
 JJ3960 0 21:29 0 22:19 CGH SDU

Rota[17 - 6]

JJ3597 0 06:00 0 08:42 CGB BSB
 JJ3825 0 11:12 0 12:50 BSB SDU
 JJ3929 0 13:15 0 14:07 SDU CGH
 JJ3109 0 17:58 0 19:10 CGH FLN
 JJ3110 0 19:50 0 21:05 FLN CGH
 JJ3774 0 22:07 0 22:47 CGH CGR

Rota[18 - 7]

JJ3770 0 06:00 0 07:00 GRU IOS
 JJ3667 0 08:15 0 11:00 IOS SDU
 JJ3923 0 11:45 0 12:48 SDU CGH
 JJ3107 0 15:17 0 16:20 CGH FLN
 JJ3104 0 16:55 0 17:56 FLN CGH
 JJ3956 0 19:59 0 21:03 CGH SDU
 JJ3064 0 22:07 0 23:40 SDU AJU

Rota[19 - 8]

JJ3900 0 06:04 0 07:04 CGH SDU(-8)
 JJ3905 0 07:16 0 08:13 SDU CGH
 JJ3912 0 09:00 0 10:00 CGH SDU
 JJ3919 0 10:45 0 11:48 SDU CGH
 JJ3121 0 12:22 0 13:25 CGH NVT
 JJ3120 0 14:00 0 15:02 NVT CGH
 JJ3938 0 15:30 0 16:32 CGH SDU
 JJ3947 0 17:44 0 18:34 SDU CGH

Rota[20 - 10]

JJ3100 0 06:05 0 07:15 FLN CGH
 JJ3908 0 08:00 0 09:00 CGH SDU
 JJ3024 0 09:42 0 11:27 SDU BSB
 JJ3814 0 12:41 0 12:55 BSB PMW
 JJ3815 0 13:35 0 15:51 PMW BSB
 JJ3029 0 16:22 0 17:58 BSB SDU
 JJ3966 0 18:48 0 19:15 SDU CFB
 JJ3967 0 19:45 0 20:12 CFB SDU
 JJ3030 0 20:37 0 22:23 SDU BSB
 JJ3738 0 23:47 1 01:20 BSB FOR

Rota[21 - 11]

JJ3901 0 06:15 0 07:07 SDU CGH
 JJ3906 0 07:30 0 08:29 CGH SDU
 JJ3756 0 08:49 0 09:55 SDU CNF
 JJ3856 0 10:29 0 11:43 CNF BSB
 JJ3260 0 12:06 0 13:31 BSB SDU
 JJ3937 0 15:15 0 16:11 SDU CGH
 JJ3944 0 16:59 0 18:06 CGH SDU
 JJ3955 0 19:44 0 20:38 SDU CGH
 JJ3958 0 21:00 0 21:55 CGH SDU
REPO 0 22:15 0 23:28 SDU CNF

JJ3408 0 23:50 1 00:59 CNF BPS

Rota[22 - 9]

JJ3768 0 06:20 0 07:27 LDB CGH
 JJ3740 0 07:55 0 08:55 CGH SJP
 JJ3740 0 09:35 0 10:05 SJP CGB
 JJ3745 0 10:45 0 13:15 CGB SJP
 JJ3745 0 13:55 0 14:56 SJP CGH
 JJ3744 0 15:42 0 16:40 CGH SJP
 JJ3743 0 17:15 0 18:14 SJP CGH
 JJ3954 0 19:29 0 20:28 CGH SDU
 JJ3961 0 21:16 0 22:15 SDU CGH

Rota[25 - 8]

JJ3758 0 06:34 0 07:47 SDU CNF
 JJ3385 0 08:30 0 09:38 CNF GIG
 JJ3411 0 10:02 0 12:05 GIG POA
 JJ3410 0 12:47 0 14:41 POA GIG
 JJ3410 0 15:22 0 16:20 GIG VIX
 JJ3137 0 17:00 0 18:28 VIX CGH
 JJ3276 0 19:24 0 20:22 CGH RAO
 JJ3275 0 21:02 0 22:04 RAO CGH

Rota[28 - 5]

JJ3370 0 06:52 0 10:27 CGH REC
 JJ3077 0 15:15 0 19:15 REC GIG
REPO 0 19:35 0 20:43 GIG CNF
 JJ3854 0 21:28 0 22:44 CNF BSB
 JJ3594 0 23:26 1 00:25 BSB PVH

Rota[31 - 7]

JJ3411 0 08:15 0 09:22 VIX GIG
 JJ3820 0 10:30 0 12:20 GIG BSB
 JJ3709 0 13:00 0 14:38 BSB CGH
 JJ3936 0 15:00 0 16:00 CGH SDU
 JJ3267 0 16:56 0 18:30 SDU CWB
 JJ3268 0 19:14 0 20:46 CWB GIG
 JJ3077 0 22:02 0 23:11 GIG CGH

Rota[23 - 9]

JJ3119 0 06:24 0 07:35 CGH NVT
 JJ3118 0 08:15 0 09:09 NVT CGH
 JJ3212 0 10:06 0 11:10 CGH CNF
 JJ3215 0 11:50 0 12:57 CNF CGH
 JJ3930 0 13:30 0 14:27 CGH SDU
 JJ3939 0 15:45 0 16:35 SDU CGH
 JJ3123 0 17:07 0 18:08 CGH NVT
 JJ3122 0 18:48 0 19:49 NVT CGH
 JJ3226 0 20:19 1 01:05 CGH NAT

Rota[26 - 7]

JJ3035 0 06:49 0 08:00 CGH JOI
 JJ3032 0 08:40 0 09:52 JOI CGH
 JJ3053 0 10:16 0 11:37 CGH POA
 JJ3052 0 12:22 0 13:48 POA CGH
 JJ3940 0 16:01 0 17:02 CGH SDU
 JJ3028 0 17:54 0 19:43 SDU BSB
 JJ3584 0 23:11 1 00:20 BSB RBR

Rota[29 - 6]

JJ3732 0 06:52 0 08:12 CGH SSA
 JJ3733 0 08:54 0 12:09 SSA CGH
 JJ3926 0 12:30 0 13:33 CGH SDU
 JJ3933 0 14:16 0 15:14 SDU CGH
 JJ3652 0 15:57 0 16:40 CGH CGR
 JJ3653 0 17:50 0 20:29 CGR CGH

Rota[32 - 1]

JJ3871 0 13:15 0 14:15 BEL MAB

Rota[24 - 9]

JJ3902 0 06:30 0 07:30 CGH SDU(-5)
 JJ3907 0 07:45 0 08:51 SDU CGH
 JJ3914 0 09:30 0 10:30 CGH SDU
 JJ3754 0 12:04 0 13:03 SDU CNF
 JJ3759 0 13:50 0 14:52 CNF SDU
 JJ3826 0 15:43 0 17:19 SDU BSB
 JJ3827 0 18:32 0 20:01 BSB GIG
REPO 0 20:21 0 21:29 GIG CNF
 JJ3458 0 22:25 1 01:05 CNF BEL

Rota[27 - 7]

JJ3370 0 06:52 0 07:49 CGH GIG
 JJ3269 0 08:18 0 09:48 GIG CWB
 JJ3266 0 10:36 0 11:57 CWB SDU
 JJ3026 0 12:43 0 14:28 SDU BSB
 JJ3027 0 15:18 0 17:04 BSB SDU
 JJ3949 0 18:15 0 19:05 SDU CGH
 JJ4722 0 20:05 0 21:44 CGH BSB

Rota[30 - 7]

JJ3022 0 07:06 0 08:56 SDU BSB
 JJ3025 0 09:52 0 11:37 BSB SDU
 JJ3265 0 12:19 0 13:45 SDU CWB
 JJ3264 0 14:30 0 15:53 CWB SDU
 JJ3941 0 16:15 0 17:11 SDU CGH
 JJ3952 0 19:00 0 20:06 CGH SDU
 JJ3967 0 20:45 0 21:37 SDU CGH

Rota[33 - 2]

JJ3563 0 13:30 0 18:43 RBR BSB
 JJ3031 0 20:46 0 22:05 BSB SDU