

UNIVERSIDADE FEDERAL DE OURO PRETO

# **Metaheurísticas Busca Tabu Para o Problema de Rodízio de Tripulações de Ônibus Urbanos**

Suelaine D. G. Andrade  
Universidade Federal de Ouro Preto

Orientador: Gustavo Peixoto Silva

Dissertação submetida ao Instituto de Ciências  
Exatas e Biológicas da Universidade Federal de  
Ouro Preto para obtenção do título de Mestre  
em Ciência da Computação

Ouro Preto, Setembro de 2013



# **Metaheurísticas Busca Tabu Para o Problema de Rodízio de Tripulações de Ônibus Urbanos**

Suelaine D. G. Andrade  
Universidade Federal de Ouro Preto

Orientador: Gustavo Peixoto Silva



A553m      Andrade, Suelaine Débora Gonçalves de.  
Metaheurísticas busca tabu para o problema de rodízio de tripulações de  
ônibus urbanos [manuscrito]/ Suelaine Débora Gonçalves de Andrade. – 2013.

73 f.: il.; color; grafs.; tabs.

Orientador: Prof. Dr. Gustavo Peixoto Silva.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de  
Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-  
graduação em Ciência da Computação.  
Área de concentração: Otimização e Inteligência Computacional.

1. Programação heurística - Teses. 2. Otimização combinatória - Teses.  
3. Transporte urbano - Teses. I. Silva, Gustavo Peixoto. II. Universidade  
Federal de Ouro Preto. II. Título.

CDU: 004.023



MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO  
 Universidade Federal de Ouro Preto  
 Instituto de Ciências Exatas e Biológicas – ICEB  
 Programa de Pós-Graduação em Ciência da Computação



### Ata da Defesa Pública de Dissertação de Mestrado

Aos 05 dias do mês de setembro de 2013, às 10 horas na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Gustavo Peixoto Silva (presidente e orientador), Prof. Dr. Haroldo Gambini Santos e Prof. Dr. José Elias Claudio Arroyo**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem a mestranda **Suelaine Débora Gonçalves de Andrade**, com o título “**Metaheurísticas Busca Tabu para o Problema de Rodízio de Tripulações de Ônibus Urbano**”. Aberta a sessão pelo presidente, coube à candidata, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

( ) Aprovada sem recomendações

( ) Reprovada

(X) Aprovada com recomendações: correções de acordo q sugestões da banca.

Banca Examinadora:

Prof. Dr. Gustavo Peixoto Silva

Prof. Dr. Haroldo Gambini Santos

Prof. Dr. José Elias Claudio Arroyo

Prof. Dr. Haroldo Gambini Santos

Coordenador do Programa de Pós-Graduação em Ciência da Computação  
 DECOM/ICEB/UFOP

Ouro Preto, 05 de setembro de 2013.

*Dedico este trabalho a minha mãe Maria, que sempre foi o maior incentivo pra minha formação.*



# **Metaheurísticas Busca Tabu Para o Problema de Rodízio de Tripulações de Ônibus Urbanos**

## **Resumo**

O Problema de Rodízio das Tripulações (PRT) do sistema de transporte público consiste em atribuir a cada tripulação uma sequência de jornadas para os dias do horizonte de planejamento. Como as jornadas diárias tem durações diferentes, as sequências das jornadas podem resultar em um acúmulo de horas extras ou de horas ociosas. Assim o objetivo do PRT é minimizar as horas extras da escala, compensando-as com ociosidades entre jornadas. Este é o princípio do banco de horas permitido pela legislação, desde que sejam respeitadas as restrições operacionais e leis trabalhistas. Neste trabalho o problema foi resolvido utilizando diferentes implementações do Algoritmo de Busca Tabu.

Primeiramente é feita a geração da solução inicial através de uma heurística gulosa. A solução gerada é viável, no entanto os custos são altos. Depois são utilizadas as jornadas criadas e com base em trocas viáveis tenta diminuir o custo de cada rodízio com diferentes versões implementadas de Busca Tabu. Foram implementadas quatro versões: a primeira versão, BTMP, que possui menor tempo da busca local para quando encontra um vizinho melhor; a segunda, denominada BTMV, em que a busca local é efetuada sobre toda a vizinhança; a terceira, BTPV, que utiliza um critério de porcentagem variável para a busca pelo melhor vizinho e a quarta versão BTID, que utiliza critérios de intensificação e diversificação para a Busca Tabu. Ao montar um rodízio, devem ser consideradas as folgas das tripulações ao longo do período. Neste trabalho foi desenvolvido um modelo em dois cenários distintos: um que não considera a atribuição das folgas e outro que realiza esta atribuição. Posteriormente os resultados foram comparados aos obtidos no

trabalho de Prates e Silva (2012) através da metaheurística VNS. Os resultados mostram que as implementações do modelo desenvolvido se aproveitam das características de cada etapa, gerando soluções mais econômicas.

# **Tabu Search Metaheuristics for the Crew Rostering Problem**

## **Abstract**

The Crew Rostering Problem (CRP) of public transportation system consists in assigning, for each crew, a sequence of journeys over the days in the planning horizon. As the daily journeys have different durations, the sequences can result in an accumulation of overtime or idle hours. Thus, the goal of the CRP is to minimize the overtime in the schedule, compensating them with idleness between journeys. This is the principle of the bank of hours allowed by the legislation, subject to operational restrictions and labor laws. In this work, the problem was solved using Tabu Search algorithm.

First is generated the initial solution by greedy search. The generated solution is feasible, though the costs are high. Then the journeys are used to create the rosters and through viable changes it tries to decrease the cost of each rostering with different implemented versions of Tabu Search. Four versions have been implemented: the first version, BTMP, which has shorter local search steps when it finds a better neighbor; the second, so-called BTMV, wherein the local search is performed over the entire neighborhood; the third, BTPV, using a criterion percentage variable to search for the best neighbor and the fourth version BTID, which uses criteria intensification and diversification for Tabu Search. When mounting a rostering should be considered the respite of the crews throughout the period. Subsequently, the results were compared to those obtained in the work of Prates e Silva (2012) that uses VNS metaheuristic. The results show that the implementations of the model developed take advantage of the characteristics of each stage, generating more economic solutions.



# Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

Suelaine D. G. Andrade



## Agradecimentos

Muitas pessoas fizeram parte destes dois anos de caminhada e a construção e conclusão deste trabalho se deve também a elas. Nestes poucos parágrafos venho agradecer a atenção, carinho, respeito, paciência e apoio recebidos de todos aqueles que trilharam este caminho ao meu lado.

Primeiramente, agradeço a Deus. Depois à minha família, que sempre esteve a meu lado e tem tanto orgulho do meu trabalho, especialmente meu pai por sua bondade e prestatividade, minha irmã pela paciência, meus sobrinhos pelas muitas alegrias, minha mãe, que foi grande apoio e força e minha avó Judith, por me ceder a tranquilidade e aconchego de sua casa nos momentos mais complicados do mestrado.

Ao meu orientador, Gustavo Peixoto Silva, por ter me acolhido e orientado, por todo o conhecimento científico e profissional compartilhado. Ao Prof. Haroldo pelos muitos conselhos e orientações, ao Prof. Rabelo pelos puxões de orelha na hora certa. Ao colega Raphael Prates por toda a disponibilidade na orientação do uso do método VNS.

Agradeço em especial ao meu grande amigo e companheiro Sávio, por toda boa vontade, carinho e por estar ao meu lado nos momentos mais críticos. À minha grande amiga Francisca por ter me acolhido em sua vida e em sua casa. Agradeço imensamente a todos os amigos que tanto amo e que torceram por mim, como Paloma, Thiago, Gabriela, Tia Rosa, Melina, Meire, Geisa, Veronica, Welisangela, Ylana, Adriana, Giselle, Carol, Fabiana. Às amigas da república Bem Boladas que me receberam de braços abertos e me apoiaram desde o início. Às amigas de Barbacena, pela compreensão nos momentos difíceis, aos amigos do NTI, do CEAD, amigos do PPGCC/UFOP, os demais amigos da UFOP e aqueles que entraram na minha vida por outros meios, todos vocês, direta ou indiretamente, fazem parte desta conquista.

Por fim, agradeço a todos os funcionários e professores do DECOM/UFOP pela

atenção, dedicação e por me oferecer uma formação técnica e humana de qualidade da qual tenho muito orgulho. Aos que porventura tenha me esquecido, peço desculpas, vocês não deixam de ser importantes simplesmente por não terem sido citados, por isso, desde já agradeço.

Muito Obrigada.

# Sumário

Lista de Figuras	xix
Lista de Tabelas	xxi
Lista de Algoritmos	xxiii
Nomenclatura	1
<b>1 Introdução</b>	<b>3</b>
1.1 Sistemas de Transporte Público Urbano . . . . .	4
1.2 Objetivos do Trabalho . . . . .	7
1.3 Estrutura do Texto . . . . .	7
<b>2 Revisão Bibliográfica</b>	<b>9</b>
2.1 Rodízios de Tripulações . . . . .	9
2.1.1 Rodízio de Tripulações de Aviões . . . . .	10
2.1.2 Rodízio de Tripulações de Trens . . . . .	12
2.1.3 Rodízio de Tripulações de Ônibus . . . . .	13
<b>3 O Problema de Rodízio de Tripulações de Ônibus Urbanos</b>	<b>23</b>
3.1 Jornadas Diárias de Trabalho . . . . .	24

3.2	Escala Fixa . . . . .	24
3.3	Escala com Rodízio de Jornadas . . . . .	26
3.4	Restrições do Rodízio . . . . .	28
<b>4</b>	<b>Metodologia para Resolução do Problema do Rodízio de Tripulações</b>	<b>31</b>
4.1	Representação da Solução . . . . .	32
4.2	Solução Inicial . . . . .	33
4.2.1	Solução sem Folgas . . . . .	34
4.2.2	Solução com Folgas . . . . .	34
4.3	Função Objetivo . . . . .	35
4.4	Estruturas de Vizinhança . . . . .	37
4.5	Busca Local . . . . .	38
4.6	Busca Tabu . . . . .	39
4.6.1	Memória de Curto Prazo - Lista Tabu . . . . .	39
4.6.2	Critério de Aspiração e de Parada da Busca Tabu . . . . .	40
4.6.3	Seleção dos Atributos do Movimento . . . . .	40
4.7	Implementações da Metaheurística de Busca Tabu . . . . .	41
4.7.1	Melhor Vizinho - BTMV . . . . .	41
4.7.2	Primeira Melhora - BTPM . . . . .	42
4.7.3	Porcentagem de Vizinhança Variável - BTPV . . . . .	42
4.7.4	Algoritmo de Busca Tabu com Intensificação e Diversificação - BTID	42
4.8	Algoritmo VNS-VND . . . . .	43
<b>5</b>	<b>Experimentos Computacionais</b>	<b>49</b>
5.1	Instâncias de Teste . . . . .	50
5.2	Solução sem Folgas . . . . .	52

5.2.1	Resultados dos Algoritmos de Busca Tabu - Sem folgas . . . . .	52
5.2.2	Resultados - VNS-VND - Sem folgas . . . . .	55
5.2.3	Comparação entre os resultados dos algoritmos BTPM, BTMV, BTPV, BTID e VNS-VND - Sem folgas . . . . .	55
5.3	Solução com Folgas . . . . .	60
5.3.1	Resultados dos Algoritmos de Busca Tabu - Com folgas . . . . .	60
5.3.2	Resultados - VNS-VND - Com folgas . . . . .	60
5.3.3	Comparação entre os resultados dos algoritmos BTPM, BTMV, BTPV, BTID e VNS - Com folgas . . . . .	61
5.4	Discussão dos Resultados . . . . .	61
<b>6</b>	<b>Conclusões</b>	<b>67</b>
	<b>Apêndice</b>	<b>69</b>
	<b>Referências Bibliográficas</b>	<b>71</b>



# Lista de Figuras

1.1	Fases do Planejamento do Transporte Público . . . . .	5
2.1	Rede representativa do Rodízio . . . . .	14
2.2	Rede do Sequenciamento Semanal (Silva e Mayrink, 2010) . . . . .	18
3.1	Exemplo de Implementação do Modelo com Escala Fixa . . . . .	26
3.2	Exemplo de Implementação do Modelo com Rodízio . . . . .	26
3.3	Exemplo Fictício de um Conjunto de Jornadas Diárias de Trabalho e Folgas Naturais nos Finais de Semana. . . . .	28
4.1	Exemplo de Jornada . . . . .	32
4.2	Exemplo de Solução do PRT . . . . .	33
4.3	Exemplo de Rodízio com Utilização do Vetor de Dias Trabalhados . . . . .	35
4.4	Exemplo de um movimento de troca . . . . .	38
5.1	Gráfico de melhorias com relação a solução inicial de todos os algoritmos utilizando a versão sem folgas . . . . .	64
5.2	Gráfico de melhorias com relação a solução inicial de todos os algoritmos utilizando a versão com folgas . . . . .	65



# Lista de Tabelas

5.1	Tabela de tempos de execução - Sem folgas . . . . .	51
5.2	Tabela de tempos de execução - Com folgas . . . . .	51
5.3	Tabela de instâncias - Número de jornadas . . . . .	52
5.4	Tabela de características de dupla pegada e noturno das instâncias (Número de jornadas de cada tipo) . . . . .	53
5.5	Tabela de horas extras e ociosas das instâncias utilizadas (em minutos) .	54
5.6	Tabela de valores em minutos de horas extras e ociosas das soluções iniciais - Sem folga . . . . .	56
5.7	Resultados obtidos através da aplicação do Algoritmo BTPM - Sem folgas	57
5.8	Resultados obtidos através da aplicação do Algoritmo BTMV - SemFolgas	57
5.9	Resultados obtidos através da aplicação do Algoritmo BTPV - Sem folgas	58
5.10	Resultados obtidos através da aplicação do Algoritmo BTID - Sem folgas	58
5.11	Resultados obtidos através da aplicação do Algoritmo VNS-VND - Sem folgas . . . . .	59
5.12	Percentuais de melhorias com relação à solução inicial - Sem Folgas . . .	59
5.13	Tabela de horas extras e ociosas das soluções iniciais- Com folgas . . . . .	62
5.14	Valores em minutos dos resultados do Algoritmo BTPM - Com folgas . .	62
5.15	Valores em minutos dos resultados do Algoritmo BTMV - Com folgas . .	62
5.16	Valores em minutos dos resultados do Algoritmo BTPV - Com folgas . .	63

5.17	Valores em minutos dos resultados do Algoritmo BTID - Com folgas . . .	63
5.18	Valores em minutos dos resultados do Algoritmo VNS-VND - Com folgas	63
5.19	Resultado de melhorias em comparação com a solução inicial - Com folgas - Comparação entre os algoritmos . . . . .	64

# Lista de Algoritmos

4.1	Busca Tabu . . . . .	41
4.2	BTID - Implementação da Busca Tabu com critérios de intensificação e diversificação . . . . .	44
4.3	Pseudo-código do procedimento VNS-VND . . . . .	46
4.4	Pseudo-código do procedimento VND . . . . .	47



*“Computer science is no more about computers than astronomy is about telescopes.”*

— Edsger Dijkstra



# Nomenclatura

PRT	: <i>Problema de Rodízio de Tripulações;</i>
PPT	: <i>Problema de Programação de Tripulações;</i>
PPV	: <i>Problema de Programação de Veículos;</i>
PVRT	: <i>Problema de Integração entre Veículos e Rodízio de Tripulações;</i>
BIDLINES	: <i>Rodízios Anônimos;</i>
ROUNDTRIPS	: <i>Conjunto de Tarefas;</i>
MBA	: <i>Multilevel Bottleneck Assignment Problem;</i>
VNS	: <i>Variable Neighborhood Search;</i>
VND	: <i>Variable Neighborhood Descent;</i>
BTPM	: <i>Busca Tabu com Primeiro Melhor Vizinho;</i>
BTMV	: <i>Busca Tabu com Melhor Vizinho;</i>
BTID	: <i>Busca Tabu com Critérios de Intensificação e Diversificação;</i>
BTPV	: <i>Busca Tabu com Porcentagem de Vizinhança Variável;</i>
VNS-VND	: <i>Busca em Vizinhança Variável - Descida em Vizinhança Variável;</i>
PI	: <i>Programação Inteira;</i>
SIGULOSO	: <i>Solução Inicial Baseada em Método Guloso;</i>



# Capítulo 1

## Introdução

O Problema do Rodízio de Tripulações (PRT) também conhecido na literatura como *Crew Rostering Problem* é tratado e desenvolvido para a área de transporte rodoviário, ferroviário, metroviário e aeroviário, sendo mais comum neste último setor de acordo com Ernst, Jlang e Krishnamoorthy (2004). A frequente investigação na programação de tripulações dos diferentes modais do sistema de transporte público deve-se principalmente à busca pela redução de custos operacionais e manutenção da qualidade da operação para as empresas.

Conforme apontado em Vasconcelos (2001), as condições gerais de transporte e trânsito são insatisfatórias para a maioria da população, o que, acrescido do aumento no preço das passagens pode agravar este problema pela perda de passageiros, que muitas vezes optam pelo transporte privado, causando um aumento no número de veículos circulando na cidade, amplificando a poluição, os engarrafamentos e o desconforto causados pelo trânsito.

Segundo ANTP (1999) uma política de transporte urbano é de grande importância no sentido de garantir melhores condições de deslocamento de mercadorias e pessoas. Assim, o planejamento do sistema de transporte urbano torna-se fundamental para garantir um desempenho satisfatório do modelo de circulação urbana. Para isto, esta melhoria em qualidade deve ocorrer em vários contextos, como por exemplo, tecnológico, gerencial, de atendimento, integração e informação aos usuários. O aumento na eficiência de operação do sistema depende da garantia de espaço, condições adequadas de trânsito, como faixas e vias exclusivas apoiadas por sistemas eficientes de controle e sinalização. Por outro lado, deve ser lembrada também, a necessidade de garantir tarifas toleráveis

para os usuários, a fim de estimular a opção da população pelo transporte público, o que depende bastante não só do aumento da eficiência como também da redução dos custos por parte das empresas, conforme pode ser visto nas manifestações populares ocorridas recentemente.

Os custos dos transportes coletivos são classificados em fixos e variáveis de acordo com Bouzada (2003) e podem apresentar ou não economia de escala. No transporte público por ônibus, os custos variáveis são os gastos com combustíveis, pneus, peças e acessórios, e os fixos são os gastos com mão de obra de motoristas e cobradores, a depreciação dos equipamentos e as despesas administrativas. Como a mão de obra é um dos maiores pesos nesse cálculo, uma pequena redução nessa variável pode significar um ganho considerável no total. E o desenvolvimento de um método eficiente para resolver o PRT, através de uma escala de trabalho otimizada traz benefícios para motoristas e cobradores, que podem contar com uma divisão mais equilibrada das atividades, o que torna o ambiente de trabalho mais amigável, beneficiando também os passageiros. Por outro lado, as empresas poderão contar com uma diminuição nos gastos com horas extras, o que diminui a pressão sobre os preços das tarifas.

## 1.1 Sistemas de Transporte Público Urbano

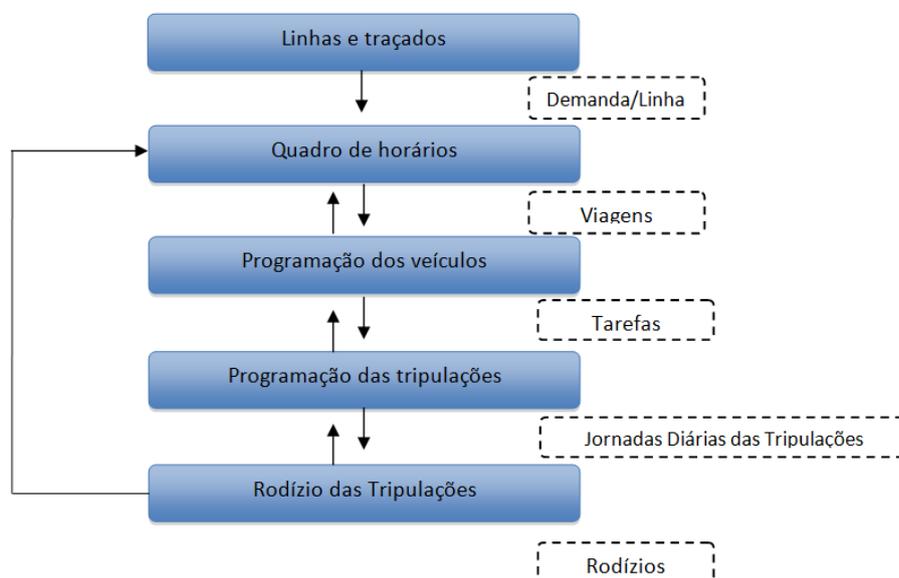
Segundo Ferraz e Torres (2004), a utilização do transporte público pode reduzir problemas como o deslocamentos de produtos e pessoas no interior das cidades. Primeiro pelo sistema ser essencial na mobilidade da força de trabalho, além de ser também, nas cidades grandes, uma alternativa para o uso do automóvel, proporcionando uma melhoria na qualidade de vida da comunidade, devido à diminuição na poluição ambiental, a utilização do transporte público pode reduzir problemas derivados do trânsito cheio e estressante, como acidentes, congestionamentos, necessidade de investimentos em obras caras, entre outros.

Outro aspecto pertinente do uso massivo do transporte público é uma ocupação e uso mais racional do solo urbano, o que contribui para tornar as cidades mais humanas e eficientes no tocante ao transporte, sistema viário e infra-estrutura de serviços públicos (Ferraz e Torres, 2004). Assim, o planejamento do sistema de transporte urbano torna-se de fundamental importância para garantir um desempenho satisfatório do modelo de circulação urbana.

De acordo com Lourenço, Paixão e Portugal (2001) as companhias de transporte público encontram muitos desafios na área de planejamento de transporte devido ao crescimento populacional, políticas ambientais, necessidade de serviços de qualidade, e pressão do governo por melhor uso de seus recursos. Diversos projetos tem sido desenvolvidos, ou estão em desenvolvimento para criar um Sistema de Planejamento de Transporte. Esse planejamento inclui a produção de tabelas de horários, escalas de veículos, geração de jornadas diárias e rodízios de tripulações para certos períodos (Paixão e Portugal, 2001).

O planejamento do sistema de transporte público urbano é uma tarefa delicada, pois envolve um complexo processo de planejamento sob a responsabilidade do poder público e das empresas prestadoras de serviços de transporte. Ele é decomposto em diversas fases de acordo com sua complexidade. De acordo com Lourenço, Paixão e Portugal (2001) as fases são: a definição de tabelas de horários, a geração das escalas de veículos, a programação e o rodízio da tripulação. A Figura 1.1 utilizada em Lourenço, Paixão e Portugal (2001) mostra a relação entre essas fases.

**Figura 1.1:** Fases do Planejamento do Transporte Público



Outros autores dividem este processo de planejamento em cinco fases, acrescentando também a definição das linhas e traçados na primeira. Nesta fase devem ser definidos os traçados a serem percorridos pelos ônibus cobrindo todas as áreas da cidade. Na segunda fase, é gerado um quadro de horários que define, para cada linha, o horário de partida das viagens, de acordo com os picos de demanda que ocorrem ao longo do dia.

Estas fases são de competência do poder público e não das empresas operadoras.

Na terceira fase é gerada a programação de veículos, também conhecida na literatura como *Vehicle Scheduling Problem*, é de responsabilidade das empresas e consiste em criar uma rotina diária de operações para uma frota de veículos de forma a atender as viagens definidas na etapa anterior. Seu objetivo é cumprir todas as viagens e reduzir os custos minimizando o número de veículos, os tempos de ociosidade e o número de deslocamentos sem passageiros. As restrições operacionais dizem respeito ao tempo mínimo em que o veículo deve permanecer na garagem, o tempo máximo que um veículo pode ficar parado em um ponto final, o máximo de trocas de linhas realizada pelo veículo entre outras.

A quarta fase engloba o Problema de Programação diária de Tripulações (PPT) ou *Bus Crew Scheduling Problem*, que consiste em gerar uma escala de trabalho para as tripulações que conduzirão a frota de ônibus em operação. Estas escalas, ou conjunto de jornadas devem contemplar todas as viagens definidas na fase anterior e ainda satisfazer as restrições legais e de operação impostas pela empresa. Este é um problema de particionamento para o qual não se conhece algoritmo polinomial para a sua resolução exata, uma vez que o problema de particionamento de conjuntos (*Set Partioning Problem*) e o problema de cobertura de conjuntos (*Set Covering Problem*), são NP - Difíceis (Garey Johnson, 1990).

Uma vez estabelecida a programação diária das tripulações, é necessário definir uma sequência das jornadas a serem executadas por cada tripulação durante todo o horizonte de planejamento, tal que seja minimizado o total de horas extras pagas pela empresa, assim como equilibrar o tempo médio de trabalho das tripulações. Este problema tem como dados de entrada as jornadas diárias de trabalho, sendo de três tipos: jornadas realizadas nos dias úteis, realizadas nos sábados e aquelas que devem ser realizadas aos domingos/feriados. Devido a demandas por viagens, o número de jornadas dos dias úteis é maior dos que as dos sábados, que por sua vez é maior do que as dos domingos. Isso se deve à redução de viagens nos finais de semana e feriados.

Neste processo pode ser utilizado o conceito de banco de horas, em que as horas extras realizadas por um condutor podem ser compensadas com horas ociosas realizadas pelo mesmo condutor em uma data posterior dentro do horizonte de planejamento. Este problema é conhecido como Problema do Rodízio de Tripulações (PRT) ou *Crew Rostering Problem*. O foco principal deste trabalho está nesta última fase do planejamento operacional, especificamente na implementação de metaheurísticas para resolver de forma eficiente o PRT de sistemas de transporte público.

## 1.2 Objetivos do Trabalho

Nesta dissertação, o PRT, será resolvido para cada uma das versões com e sem folgas das tripulações. Primeiramente será gerada uma solução inicial através de implementação de uma heurística construtiva baseada em uma função gulosa, proposta por Prates e Silva (2012). Depois, através de implementações da metaheurística Busca Tabu, os rodízios serão otimizados a fim de melhorar a função de custo, permitindo apenas trocas viáveis. Os resultados serão comparados com os resultados obtidos com o método VNS utilizado por Prates e Silva (2012).

## 1.3 Estrutura do Texto

No Capítulo 2 é feita uma revisão bibliográfica sobre o PRT. No Capítulo 3 o PRT é explicado em detalhes e no Capítulo 4 se encontra a metodologia utilizada para resolver o problema tratado neste trabalho. No Capítulo 5 estão apresentados os experimentos computacionais assim como seus resultados e discussões, e finalmente, no Capítulo 6 encontra-se a conclusão do trabalho.



# Capítulo 2

## Revisão Bibliográfica

Neste capítulo são destacados os principais trabalhos encontrados na literatura que dizem respeito ao PRT, nos diversos setores de transporte onde ele surge.

### 2.1 Rodízios de Tripulações

Rodízios de tripulações é o processo de construir escalas e jornadas de trabalho para os funcionários de forma que as empresas possam satisfazer as demandas por bens ou serviços. De acordo com Ernst, Jlang e Krishnamoorthy (2004) a primeira parte desse processo envolve determinar o número de funcionários, com habilidades particulares, necessários para atender as demandas de serviços.

É extremamente difícil encontrar boas soluções para esses problemas complexos e repletos de restrições, e mais difícil ainda é encontrar soluções ótimas que minimizem os custos, atendam as preferências dos trabalhadores, distribuam as jornadas equitativamente entre os funcionários e satisfaçam todas as restrições trabalhistas. Isto decorre do grande número de combinações, o que impede a resolução por enumeração explícita de soluções.

Em várias organizações, as pessoas envolvidas no desenvolvimento de rodízios precisam de ferramentas de suporte para prover os funcionários corretos nos horários certos. Normalmente são usadas planilhas, bancos de dados, possíveis ferramentas de rodízios, modelos matemáticos e algoritmos apropriados.

### 2.1.1 Rodízio de Tripulações de Aviões

Como já foi citado, o planejamento de tripulações de aviões é provavelmente a maior aplicação de rodízios de funcionários devido à escala e aos custos. Em média, os custos de tripulação podem representar de 10 a 15 % dos custos totais de transporte aéreo. De acordo com Ernst, Jlang e Krishnamoorthy (2004) a maioria dos artigos científicos da área tem se dedicado mais a tecnologias e aplicações nessa área do que em qualquer outra.

A indústria aérea refere-se ao rápido transporte de pessoas e mercadorias e é altamente competitiva. O sucesso de uma companhia aérea depende em primeiro lugar das rotas em que ela opera. Mas esta operação deve ser eficiente. Ao projetar a tabela de vôos de companhia aérea, deve-se considerar a demanda por transporte, os interesses econômicos das companhias aéreas e várias restrições operacionais.

Os passageiros, de um lado, estão interessados na frequência dos vôos e em minimizar os atrasos e cancelamentos. Por outro lado, os interesses das companhias aéreas são também levados em conta, como a alta utilização anual de aeronaves, baixos custos operacionais e lucro alto. As companhias aéreas estão interessadas em escalas de vôos com ocupação maximizada e boa utilização da capacidade de transporte existentes.

Ao criar uma escala de trabalho, vários requisitos operacionais devem ser cuidadosamente estudados e levados em conta, como:

1. requisitos do departamento de manutenção aérea como frequência, localização e duração são prescritos dentro do cronograma de manutenção de aeronaves,
2. a disponibilidade de aeronaves e tripulações (números de vários tipos de aeronaves, disponibilidade de portão, pilotos, comissários de bordo, pessoal de serviços em terra, e pessoal de atendimento ao cliente);
3. regulamentação das horas de trabalho das tripulações e requisitos de treinamento

Ao desenvolver as escalas das empresas, os requisitos do departamento de planejamento da tripulação também devem ser estudados em detalhes porque as tripulações das companhias aéreas tem legislações e regras sindicais que regulamentam o trabalho. Da mesma forma, é necessário cuidar do planejamento dos funcionários em terra, os que processam aeronaves no solo, e pessoal de apoio ao cliente, que servem os passageiros no terminal para cada estação na rede.

Devido ao fato de que uma tripulação de vôo tem licenças para um tipo específico de aeronave, os cuidados devem ser tomados para que haja pilotos suficientes, co-pilotos e em alguns casos engenheiros de vôo para cada tipo de aeronave utilizado. Razões de segurança, as leis relevantes, e da empresa devem definir com precisão as horas de trabalho, tempo de descanso, de vôo da tripulação.

Assim, ao desenvolver a escala da companhia aérea, a atenção também deve ser dada a janelas de tempo durante o dia e aeroportos onde substituições de tripulação serão feitas, o tempo que a tripulação vai ficar em certos aeroportos durante a espera para assumir a jornada, as viagens que a tripulação faz como passageiros normais a fim de ir ou voltar para a base, e todos os outros fatores relacionados com o trabalho da tripulação de vôo e o trabalho de comissários de bordo .

Os problemas de pareamentos (ou programação) de tripulações aéreas diferem entre as empresas aéreas com relação a custos e regras, mas as principais características permanecem as mesmas (Kohl e Karish, 2004). As estruturas das regras são comparáveis e as funções objetivo são baseadas em pagamentos e refletem principalmente os custos, como já mencionado. Ao contrário do pareamento, o rodízio de tripulações aéreas pode ser feitos de várias formas de acordo com a abordagem.

Na América do Norte, por exemplo, a maior parte dos rodízios aéreos é feita em dois passos: primeiro, rodízios anônimos (*bidlines*) são criados e posteriormente eles são associados a indivíduos baseados em ofertas para os rodízios anônimos. Em muitas empresas européias os rodízios individuais são construídos diretamente para cada membro da tripulação, chamados rodízios personalizados. Todos os membros de tripulações devem ter rodízios que satisfaçam certos critérios de qualidade ou preferências individuais. Os tripulantes colocam suas preferências pessoais que são consideradas durante a criação dos rodízios individuais.

Ao comparar as diferentes versões de rodízios aéreos, sob o ponto de vista da solução, os rodízios são similares, agora sob a visão de modelagem eles diferem basicamente na formulação da função objetivo (Kohl e Karish, 2004). Como os diferentes princípios de rodízios são aplicados de formas distinta nas empresas aéreas, e são normalmente combinados e estendidos, há uma demanda por sistemas de rodízios flexíveis e genéricos, que suportem a modelagem em ambientes diferentes e tenham métodos de otimização apropriados para resolver os problemas com eficiência e eficácia.

### 2.1.2 Rodízio de Tripulações de Trens

Rodízios e escalas de tripulações de trens somente tem aparecido recentemente na literatura de acordo com Ernst, Jlang e Krishnamoorthy (2004). A maior parte dos trabalhos diz respeito a aplicações reais de rodízios de tripulações de trens em diferentes países. Ele é um problema recente e um desafio difícil para os operadores de trens. A criação de rodízios de trabalho factíveis envolve a incorporação de requisitos legais, assim como preferências individuais advindas do motoristas ou condutores dos trens.

Em Hartog et al. (2009) e Huisman et al. (2005) são sugeridas três abordagens genéricas para o processo de rodízios de tripulações de trens.

1. Rodízios individuais são criados para todos os tripulantes, em que, por exemplo, desejos ou preferências (como a licença profissional ou obrigações privadas) podem ser levados em conta. Estes rodízios são geralmente criados para um curto espaço de tempo.
2. A abordagem de rodízio de linha de oferta (*bid-line*) permite que os membros da tripulação individuais façam uma oferta para concorrer a um determinado rodízio. Na prática, isso geralmente é feito através da criação de alternativas múltiplas de rodízios e dando aos tripulantes a oportunidade de escolher uma dessas alternativas.
3. Rodízios cíclicos são as escalas genéricas de trabalho nas quais sequências predefinidas de jornadas (normalmente por semana) variam entre os funcionários previstos neste planejamento. Portanto, os funcionários sabem com muito tempo de antecedência em que turno ou parte do dia eles vão trabalhar em uma determinada data.

Ernst, Jlang e Krishnamoorthy (2004) descrevem o processo de rodízio em seis módulos: modelo de demanda, escala de dias de folga, escala de viagens, construção das linhas de trabalho, atribuição de tarefas e atribuição das tripulações.

Na indústria ferroviária diversas especificidades são aplicadas ao problema de rodízio. Primeiro, a tripulação pode terminar a jornada em uma localização diferente de onde iniciou. Assim deve ser levado em conta o transporte de volta à sua base de origem e o tempo. As maiores dificuldades nesse planejamento são as restrições de tempo conforme os trabalhos de Hartog et al. (2009) e Huisman et al. (2005).

### 2.1.3 Rodízio de Tripulações de Ônibus

Assim como no planejamento de tripulações de aeronaves, as escalas e rodízios de ônibus também são criadas a partir de tabelas de horários de ônibus. Ao contrário do que ocorre no setor aéreo, neste caso, as escalas de tempo são de menor duração pois os conceitos de *roundtrips*, pareamentos ou sequência de tarefas são substituídos pelo conceito de jornadas e a maior parte destas pode ser realizada por uma tripulação sem a necessidade de longos intervalos de descanso durante a jornada diária. Embora a questão como deslocamentos também seja importante neste caso para a geração de jornadas, no planejamento das escalas e rodízios para ônibus urbano, o conceito de *home depots* ou base de casa, normalmente não se aplica o que significa que os locais de início e fim de uma jornada não precisam necessariamente ser os mesmos.

O rodízio de tripulações de ônibus urbanos tem por objetivo encontrar um conjunto de sequenciamentos de jornadas diárias, de menor custo, que cubra todas as jornadas diárias previstas dentro de um dado horizonte de planejamento. As sequências das jornadas, ou rodízios, são atribuídas a tripulações genéricas que posteriormente devem ser distribuídas entre as tripulações da empresa.

Uma jornada é formada por viagens que uma tripulação realiza durante um dia de trabalho. Para cada jornada um custo é associado, o qual representa os custos associados ao salário fixo, às horas extras ou ociosas realizadas durante a mesma. Custos artificiais tais como troca de veículos e dupla pegada também são considerados na resolução do problema.

O trabalho a ser realizado pelas tripulações está sujeito a várias restrições operacionais e legais dos rodízios, que em geral são relacionadas a:

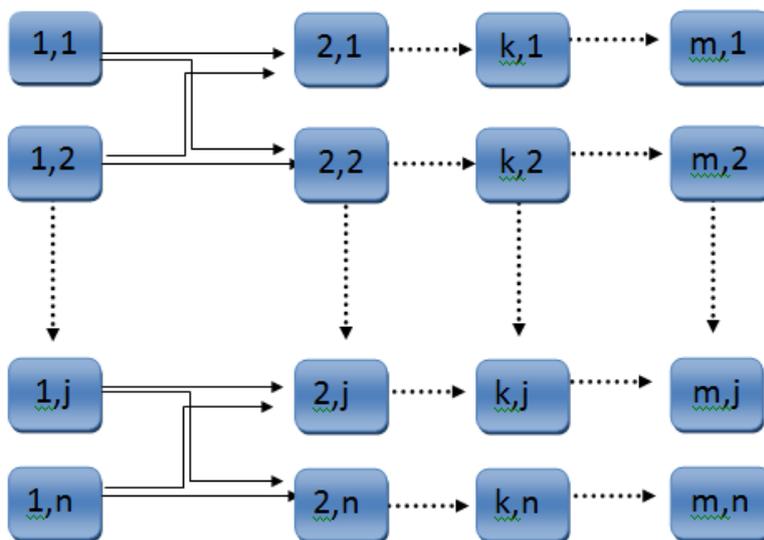
- Tempo máximo de trabalho
- Tempo máximo de intervalo entre uma viagem e outra
- Tempo mínimo de descanso entre uma jornada e outra

O rodízio de tripulações de ônibus urbanos tem recebido maior atenção nas últimas décadas. No trabalho de Carraresi e Gallo (1984) é proposto um modelo de assinalamento que atribui um peso a cada jornada, que representa o seu custo em função de suas horas extras ou ociosas. Este problema de encontrar um balanceamento das jornadas sobre um dado intervalo de tempo é formulado como um problema de gargalo

múltiplo, ou seja, minimizar o peso máximo dentre todas as jornadas. Nesse trabalho o PRT é formulado para um horizonte de  $m$  dias, cada dia com  $n$  jornadas, cada uma representada por um nó  $i, j$ , sendo  $i$  o dia e  $j$  a jornada como mostra a Figura 2.1. Os arcos representam os caminhos possíveis de ligação entre cada uma das jornadas (nós) de um dia com o próximo dia. Para  $m = 2$  temos o problema padrão de designação com gargalo. Já para  $m > 2$  o problema é chamado de Problema de Designação Multi-Nível com Gargalo (*Multilevel Bottleneck Assignment Problem - MBA*).

Para uma única tripulação, por exemplo, o problema corresponde a encontrar um caminho de um dos nós do primeiro conjunto a um dos nós do  $m$ -ésimo conjunto. O comprimento do caminho é dado pela soma dos pesos de cada nó utilizado no caminho. Os autores assumem que um mesmo conjunto de jornadas deve se repetir a cada dia. Um nó  $\langle k, i \rangle$  representa a jornada  $i$  no dia  $k$  (nível).

Figura 2.1: Rede representativa do Rodízio



A formulação matemática pode ser dada por:

Minimizar  $z$ , sujeito a:

$$x_{ij}^k \in \{0, 1\}; \quad k = 1, \dots, m - 1; \quad i, j = 1, \dots, n. \quad (2.1)$$

$$\sum_{j \in S_k(i)} x_{ij}^k = 1; \quad k = 1, \dots, m - 1; \quad i = 1, \dots, n. \quad (2.2)$$

$$\sum_{i \in P_k(j)} x_{ij}^k = 1; \quad k = 1, \dots, m - 1; \quad j = 1, \dots, n; \quad (2.3)$$

$$s_j^1 = w_{1j}; \quad j = 1, \dots, n. \quad (2.4)$$

$$s_j^k = w_{kj} + \sum_{i \in P_{k-1}(j)} s_i^{k-1} x_{ij}^{k-1}, \quad j = 1, \dots, n; \quad k = 2, \dots, m. \quad (2.5)$$

$$s_j^m \leq z; \quad j = 1, \dots, m. \quad (2.6)$$

Em que:

$x_{ij}^k$  indica se o arco que parte da jornada  $i$  no dia  $k - 1$  for associado a jornada  $j$  no dia  $k$ .

$S_{k(i)}$  e  $P_{k(i)}$  foram definidos anteriormente como o conjunto de nós sucessores e predecessores do nós  $k$ ,

$s_j^k$  é a soma dos pesos (pesos acumulados) de todas as jornadas utilizadas até o dia  $k$ , em que

$w_{kj}$  é o peso de associar cada jornada  $j$  ao dia  $k$

$z$  é o peso total (acumulado) que deve ser minimizado

A restrição (2.1) indica que  $x_{ij}^k$  é uma variável binária. As restrições (2.2) e (2.3) implicam que as variáveis binárias  $x_{ij}^k$  representarão uma partição dos nós em caminhos de tamanho  $n$ . Já (2.4) indica que a soma dos pesos de todas as jornadas utilizadas até o dia 1 deve ser igual ao peso de associar cada jornada  $j$  ao primeiro dia. As restrições (2.5), (2.6) induzem à interpretação de que  $s_j^k$  é o peso dos caminhos parciais, começando com um nó no nível  $l$  e terminando com um nó no nível  $k$ .

O problema (2.1) - (2.6) tem uma solução factível se e somente se, para cada  $k = 1, \dots, m - 1$ , o conjunto

$$Y_k = \{x^k : \sum_{j \in P_k(i)} x_{ij}^k = 1; \quad x_{ij}^k \in \{0, 1\}; \quad i, j = 1, \dots, n\} \quad (2.7)$$

não estiver vazio.

Conforme provado por Carraresi e Gallo (1984), o *MBA* é NP-Completo, mas se o número de dias  $m$  for fixado é possível encontrar um algoritmo que resolva o problema em tempo polinomial. Neste trabalho é apresentado um algoritmo de solução heurística que resolve iterativamente diversos problemas de gargalo de assinalamento sobre os nós no nível  $k$  e  $k + 1$ .

No entanto, de acordo com Bodin, Assad e Ball (1983), nenhuma destas aproximações fornece uma solução completa ao problema do rodízio de tripulações. A função objetivo utilizada no primeiro modelo não é facilmente aplicável a problemas reais e o segundo modelo não trata a questão das folgas. Nenhum dos dois modelos é facilmente extensível a horizontes maiores do que uma semana enquanto que nas situações reais, um único rodízio cobre várias semanas. Estes modelos poderiam entretanto ser úteis se combinados com procedimentos que tratem de outros aspectos do problema.

Em Townsend (1988) é proposta uma resolução do rodízio de tripulações de ônibus para o transporte regional de Londres. Um método é desenvolvido a fim de produzir um rodízio-base, que é composto de unidades padrão que contém padrões fixos de trabalho. O objetivo é determinar o número de unidades e então posicioná-las em uma sequência que concorde com objetivos operacionais e obedeça às restrições. O primeiro objetivo é produzir uma distribuição justa do trabalho nos finais de semana. Os experimentos indicaram que o procedimento foi capaz de produzir rapidamente rodízios de alta qualidade.

Em Catanas e Paixão (1995) é apresentado um algoritmo para determinar um limite inferior para o Problema de Rodízio de Tripulações (PRT). A formulação do PRT apresentado é basicamente a de um problema de cobertura de conjuntos com restrições adicionais. O método proposto para obter um limite inferior consiste em resolver um problema linear com base em uma adaptação da abordagem clássica de geração de colunas. O subproblema associado mostra ser um problema de caminho mínimo com restrições adicionais e é apresentado um corte muito eficiente para a formulação.

Em Souza, Yunes e Moura (2000), o PRT é modelado e resolvido com a técnica de Programação por Restrições e Programação Inteira - PI. Os autores tratam do sistema de transporte público brasileiro, especificamente de uma companhia que opera em Belo Horizonte, MG. Neste modelo, limitantes inferiores foram obtidos resolvendo o problema linear com a relaxação das restrições de integralidade das variáveis e usados para avaliar a qualidade da solução encontrada. É apresentada também uma heurística de geração de colunas combinando programação inteira e programação por restrições sobre uma

formulação de cobertura de conjuntos. O objetivo principal é encontrar a solução mais barata em termos de número de tripulantes e carga de trabalho balanceada em um horizonte de  $m$  dias.

Os resultados obtidos com o modelo de PI mostraram que encontrar solução ótima é bastante difícil apesar do tamanho das instâncias utilizadas não serem grandes. Com instâncias maiores o modelo não conseguiu um bom desempenho e por isto os autores decidiram utilizar a técnica de Programação por Restrições. Quando comparado com a formulação por PI o modelo teve um desempenho bem melhor tanto em qualidade da solução quanto em tempo computacional. Assim, foi possível encontrar soluções factíveis para grandes instâncias em um tempo de processamento reduzido. Outra abordagem do mesmo problema utilizada pelos autores foi a de um modelo híbrido utilizando particionamento de conjuntos. O modelo é híbrido pois a fase de geração de colunas do subproblema foi resolvida por um modelo de programação por restrições. Como o número de colunas foi muito grande, o algoritmo *branch and price* foi implementado para gerar as colunas. Com este modelo foi possível encontrar soluções prováveis ótimas para pequenas instâncias, o que é uma melhora do modelo de PI.

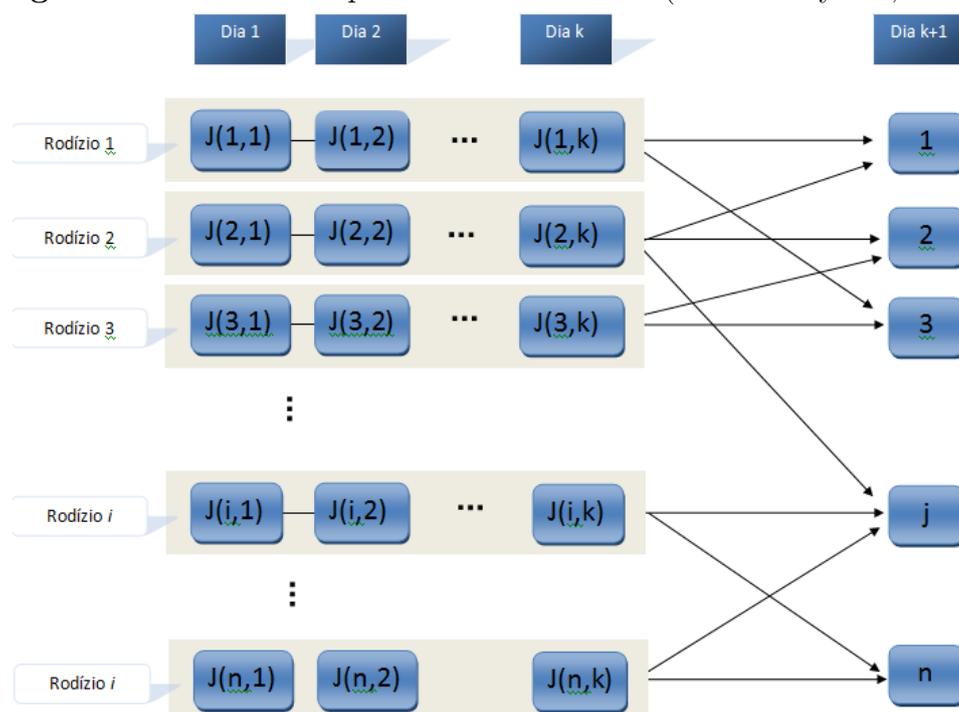
As conclusões gerais desse trabalho afirmam que o modelo de PI encontrou rodízios factíveis para instâncias pequenas. Por outro lado, a programação por restrições melhorou os resultados e factibilidade para instâncias reais em poucos segundos, enquanto o modelo de particionamento de conjuntos encontrou soluções ótimas para instâncias pequenas. Os autores afirmam também que encontrar prováveis soluções ótimas para grandes instâncias do problema ainda é uma tarefa difícil.

Em Toffolo, Souza e Silva (2005) são utilizadas as Meta-heurísticas *Simulated Annealing* e *Iterated Local Search* (ILS) para resolver o problema de rodízio de tripulações. São aplicados através da utilização de movimentos baseados na troca de jornadas entre tripulantes, sendo adotado como procedimento de busca local o método randômico de descida. De acordo com os autores, cada solução gerada é avaliada por uma função baseada em penalidades. Esta função possui componentes que avaliam restrições essenciais, tais como restrições legais referentes às folgas das tripulações, e restrições não essenciais, que são situações que apesar de indesejáveis podem ocorrer em determinadas circunstâncias. Os autores mostram que os métodos são capazes de gerar soluções viáveis rapidamente, sendo o método *Iterated Local Search* capaz de produzir soluções finais mais eficientes do que o *Simulated Annealing*, ou seja, produzem soluções com menor custo em um tempo de processamento reduzido.

Em um trabalho mais recente, Silva e Mayrink (2010) é construído o sequenciamento das jornadas da Figura 2.2 baseado no modelo de Carraresi e Gallo (1984), considerando as características particulares da realidade do caso abordado. Assim, a partir dos dados do primeiro dia do horizonte, é construída uma rede bipartida e resolvido um problema de designação para fazer a alocação do segundo dia do horizonte. Esse modelo minimiza o total de horas extras tentando compensá-las com horas ociosas. Nesse trabalho, a definição do rodízio mensal, sem as folgas, é dividida em duas etapas:

1. geração de um rodízio semanal
2. criação do rodízio mensal com o sequenciamento otimizado dos rodízios semanais

**Figura 2.2:** Rede do Sequenciamento Semanal (Silva e Mayrink, 2010)



Posteriormente é resolvido um modelo de PI para a alocação das folgas. Este modelo nem sempre atinge o ótimo, mas ainda assim, as soluções obtidas se mostraram superiores com consideráveis melhorias em relação à solução “bruta” pela empresa.

Em Leite e Silva (2012), também é utilizado um modelo de fluxo em redes, mas diferentemente de Carraresi e Gallo (1984) a rede é construída para o período completo de uma única vez. Este modelo também minimiza o total de horas extras tentando

compensá-las com as horas ociosas. Para o primeiro dia da semana, atribui-se a cada jornada um conjunto de arcos ligando-a com as possíveis jornadas do dia seguinte, respeitando as restrições do problema. Os custos nos arcos correspondem ao custo total de uma tripulação que executar as duas jornadas consecutivamente.

Os resultados de Leite e Silva (2012) não são mais eficientes do que aqueles obtidos por Silva e Mayrink (2010), considerando a soma do total de horas-extras e o total de horas ociosas. Entretanto, ao comparar a diferença entre as horas extras e as horas ociosas, nota-se que o valor obtido pelo modelo de Leite e Silva (2012) é menor. Naturalmente que as horas-extras de uma sequência de jornadas não podem ser compensadas com as horas ociosas de outra sequência, mas é possível combinar as duas propostas gerando uma escala para uma semana utilizando o modelo de Leite e Silva (2012) e combinar as semanas ao longo do horizonte de planejamento pelo método proposto por Silva e Mayrink (2010). Desta maneira, o objetivo é compensar o máximo de horas extras com ociosas de uma mesma sequência, minimizando o custo total.

Algumas tripulações não executam jornadas em um ou dois dias do final de semana, retornando à atividade no próximo dia útil. Assim, na formulação de Leite e Silva (2012), os arcos que partem das jornadas da sexta e se ligam às jornadas do domingo representam uma folga no sábado, os arcos de sexta para segunda representam uma folga dupla no sábado e domingo e os arcos que ligam as jornadas do sábado com as jornadas da segunda, representam uma folga no domingo. A rede é resolvida transformando-a em um problema de circulação, e aplicando-se o algoritmo *Out-of-Kilter* (Ahuja, Magnanti e Orlin, 1993) para escolher a melhor combinação de jornadas. A solução deste modelo permite definir um rodízio para o período mas não faz a alocação das folgas dos tripulantes, além das folgas naturais provenientes dos sábados e domingos. As restrições deste trabalho são similares às restrições utilizadas nesta dissertação, porém difere na forma de construção da solução inicial, na função objetivo e no tratamento das restrições de folgas.

Em Mesquita et al. (2011) é apresentada uma formulação de programação por metas binárias e não lineares para o Problema de Integração entre Veículos e Rodízio de Tripulações, o PVRT. O método proposto para obter soluções factíveis para este problema é um algoritmo sequencial considerado em um *framework* de programação por objetivos que dá maior prioridade à integração da programação de veículos/tripulações e menor para objetivos de rodízios de motoristas. A integração do PPV e do PPT consiste em três fases: preprocessamento, resolução de relaxação linear e obtenção de soluções factíveis. Posteriormente, conhecidas essas soluções (as jornadas diárias) como ponto inicial, e cobrindo o horizonte de planejamento, o rodízio de tripulações é resolvido em um contexto

não cíclico por um modelo binário bi-objetivo que considera interesse de motoristas e da gerência. Também é aplicada programação por objetivos que levou a soluções próximas da otimalidade.

De acordo com os autores, diferentes valores de alguns parâmetros do algoritmo, controlados pelo usuário, produzem diferentes soluções. Posteriormente, usando essas soluções como ponto de partida e cobrindo o horizonte de planejamento, o PRT é abordado em um contexto de programação não cíclica, por um modelo binário multi-objetivo que considera tanto os interesses da empresa quanto os dos motoristas.

Este problema também é resolvido por uma aproximação de programação por metas que levam a soluções mensais quase ótimas. A fim de avaliar a qualidade das diferentes soluções do PVRT, as soluções do rodízio foram medidas de perspectivas diferentes do custo. Por fim, os autores enfatizam que o PRT analisado tem grande gama de aplicações.

Em Prates e Silva (2012) o PRT é resolvido por meio da metaheurística *Variable Neighborhood Search* (VNS) tendo como método de busca local o *Variable Neighborhood Descent* - VND. As soluções respeitam o padrão de folga fixa do tipo 5/1 além das restrições legais e operacionais vigentes na empresa. São utilizadas diferentes estruturas de vizinhança. A implementação foi testada com dados de uma empresa de médio porte e os resultados mostraram melhorias significativas em relação à solução adotada pela empresa.

Em Nurmi, Kyngas e Post (2011) é apresentada uma forma de planejar os dias de folga das tripulações em uma base anual e as jornadas em uma base mensal em uma empresa finlandesa de transporte urbano. Os dias de folga e jornadas são planejados e alocados usando um algoritmo que inclui características de heurísticas populacionais, assim como heurísticas construtivas como *Simulated Annealing*, Busca Tabu e cadeias de ejeção. As restrições do problema são classificadas nos tipos: cobertura, regulamentação, operacionais, preferências operacionais e pessoais. O algoritmo para o método de solução do PRT é baseado em um método de busca local em população, na técnica de cadeias de ejeção.

A principal heurística adotada em Nurmi, Kyngas e Post (2011) é a busca gulosa de subida da encosta (*hill-climbing*). A solução inicial é criada alocando as jornadas em dias aleatórios, assim como neste trabalho que faz a associação das jornadas na solução inicial respeitando todas as restrições. As próximas jornadas são selecionadas de forma a causar a maior diminuição na função objetivo. O método, baseado em população, usa uma população de soluções em cada iteração. *Simulated Annealing* é implementado a

fim de evitar ficar preso em áreas que não são promissoras, mas esta estratégia não é suficiente.

Assim as piores soluções são substituídas pelas melhores quando  $n$  melhores programações são encontradas ( $n$  é o tamanho da população). Para adicionar penalidade são atribuídos pesos dinâmicos a restrições difíceis. Os autores citam também que a solução funciona melhor com dados de entrada randômicos.



## Capítulo 3

# O Problema de Rodízio de Tripulações de Ônibus Urbanos

O problema de rodízio de tripulação consiste basicamente na atribuição de dias de trabalho e de descanso a um conjunto de funcionários de forma que regras trabalhistas sejam satisfeitas, levando em conta restrições derivadas do tipo de trabalho e das preferências dos trabalhadores. Empresas pequenas costumam resolver este problema manualmente, tirando pouco proveito da possibilidade de fazer a compensação das horas extras dentro do horizonte, normalmente de um mês.

As empresas de transporte público são encarregadas de cumprir diariamente um quadro de horários de viagens para cada linha que esteja sob sua responsabilidade. Há a distribuição destas viagens entre os veículos da frota que entrará em operação. As tripulações devem realizar essas viagens, que juntas formam as jornadas que apresentam diferentes características como horários de início e término, tempo de duração e turno de trabalho.

As jornadas diárias de trabalho podem ser obtidas através de um planejamento manual ou através da utilização de técnicas computacionais para resolução do PPT. O período de planejamento da operação de uma empresa de ônibus normalmente corresponde a um mês. Há, contudo, estudos para horizontes de planejamento, ou seja, de rodízios das tripulações compostos de cinco, seis ou até sete semanas, como no caso abordado nesta dissertação.

### 3.1 Jornadas Diárias de Trabalho

Para cada tipo de dia do horizonte de planejamento, existe um conjunto distinto de jornadas diárias de trabalho, sendo que cada uma delas deve ser executada por alguma tripulação de modo que todas as viagens atribuídas à empresa sejam rigorosamente cumpridas. Este conjunto de jornadas diárias é, em geral, igual para todos os dias úteis, diferindo-se nos sábados e nos domingos. Esta característica se deve ao fato de existir um único quadro de horários para os dias úteis, outro para os sábados e um terceiro para os domingos/feriados.

Na ocasião de feriados o conjunto de jornadas é idêntico ao dos domingos, pois o quadro de horários dos feriados corresponde ao próprio quadro de horários do domingo. Estes três conjuntos distintos de jornadas de trabalho são denominados jornadas diárias. Elas apresentam características que as distinguem das outras, como:

1. *horário de início* indica o horário que a tripulação inicia sua jornada,
2. *horário de término* em que o funcionário está liberado do expediente de trabalho,
3. *turno da jornada* é estabelecido de acordo com o horário de início da jornada,
4. *ociosidade ou hora extra* quando a duração de uma jornada é inferior ou superior à carga normal de trabalho,
5. *dupla pegada* quando ocorre uma interrupção superior a duas horas durante o horário de trabalho. Intervalos menores que duas horas devem obrigatoriamente ser remunerados, e não constitui uma dupla pegada. Não pode haver duas ou mais interrupções não remuneradas em uma mesma jornada de trabalho,
6. *horário noturno* que aponta se a jornada inicia entre 22:00 e 5:00 horas do dia seguinte. Este tipo de jornada tem uma remuneração diferenciada.

### 3.2 Escala Fixa

Para realizar o planejamento destas jornadas, muitas empresas ainda utilizam o sistema de escala fixa. Neste sistema as tripulações realizam as mesmas jornadas de trabalho todos os dias ao longo de todo o período de planejamento. Este modelo ainda é muito

adotado na prática pois facilita a tarefa de alocação das jornadas que normalmente é feita de forma manual.

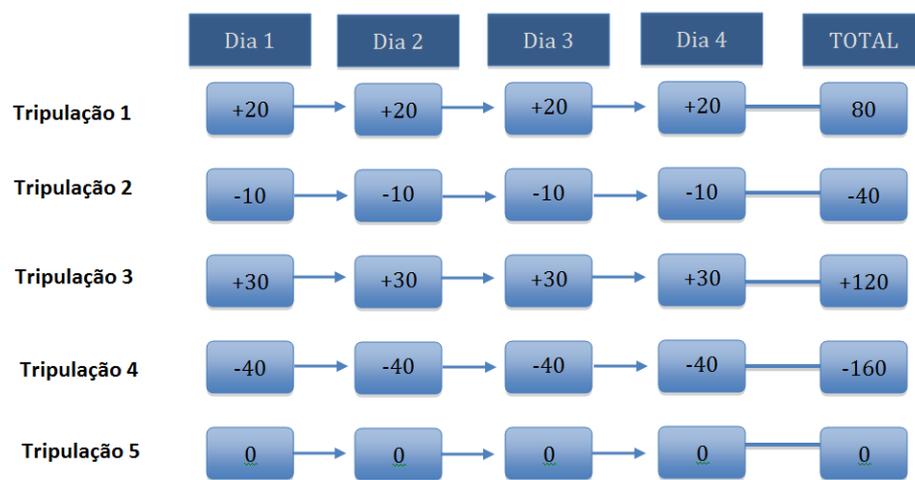
Vantagens e desvantagens da escala fixa:

- A vantagem deste sistema é a simplicidade na sua construção, além da facilidade de implementação pois não há variação nas jornadas a serem realizadas pelas tripulações ao longo do horizonte de planejamento.
- A alocação fixa também permite que os motoristas se familiarizem com a jornada de trabalho e com o traçado das linhas
- Uma desvantagem é que a escala resultante é bastante onerosa para as empresas, já que as jornadas com horas-extras são acumuladas ao final do período, e elas não são compensadas com as jornadas que têm horas ociosas.
- Outra desvantagem é que o esquema de jornadas fixa pode ser bastante injusto para os funcionários devido ao critério de seleção das jornadas, que é baseado no tempo de casa. Neste critério, os motoristas mais antigos podem escolher as jornadas que mais desejarem, que são aquelas com maior quantidade de horas-extras e que iniciam logo pela manhã.

A Figura 3.1 ilustra um pequeno exemplo fictício do modelo de escala fixa. Os nós representam os minutos extras ou ociosos das jornadas de trabalho e a sequência de jornadas que cada tripulação deverá realizar. Cada linha (ou sequência de nós) representa uma tripulação e o conjunto de jornadas associadas a ela. Os nós finais representam a soma das horas para cada tripulação.

Neste exemplo, a tripulação 5 representa um rodízio ideal para uma empresa, em que o funcionário não produz horas extras ou ociosas. as segunda e a terceira tripulações tendem a trabalhar mais do que as outras e conseqüentemente ficarem com um número maior de horas extras ao final do período de planejamento. Por outro lado, os outros rodízios contariam com um grande número de horas ociosas, o que representa menos trabalho recebendo o mesmo valor no final, resultando em um desperdício de mão de obra.

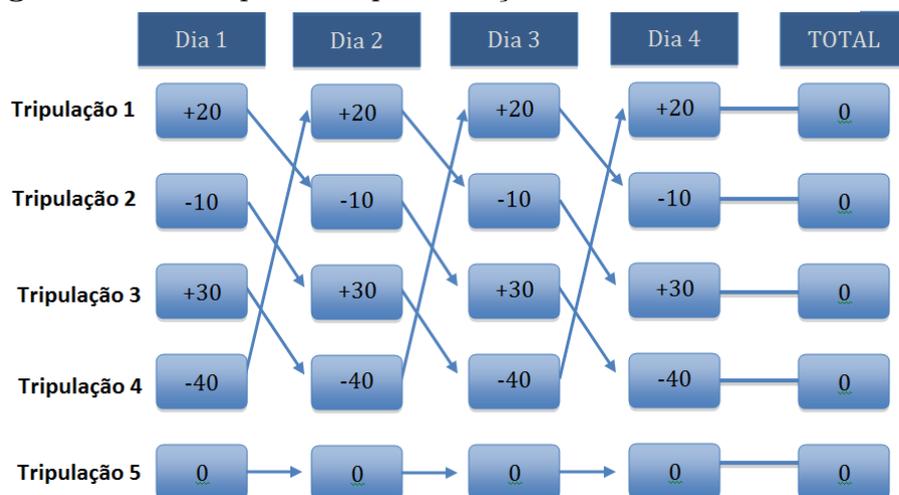
**Figura 3.1:** Exemplo de Implementação do Modelo com Escala Fixa



### 3.3 Escala com Rodízio de Jornadas

O rodízio das tripulações, em contraste com o sistema de escala fixa, busca estabelecer uma carga de trabalho mais imparcial aos funcionários e uma distribuição igualitária das horas-extras. A ideia principal é permitir que jornadas com diferentes características sejam intercaladas, com o objetivo de compensar as horas extras com horas ociosas de jornadas de dias posteriores, dentro do horizonte de planejamento.

**Figura 3.2:** Exemplo de Implementação do Modelo com Rodízio



Assim, o PRT visa determinar um conjunto de escalas mensais (ou periódicas) de trabalho, também denominado rodízio das tripulações, cada um composto de uma sequência

de jornadas diárias de trabalho a serem atribuídas a uma tripulação. A resolução do PRT tem o objetivo de se produzir sequências (rodízios) mais equilibradas em termos de tempo de trabalho e de horas extras, para reduzir os gastos, como mostra a Figura 3.1. As sequências produzidas na solução do PRT são tais que satisfazem as leis trabalhistas (CLT) e as convenções coletivas de trabalho, que podem apresentar variações de acordo com a região de atuação da empresa. Além disso, cada empresa também pode adotar critérios próprios, que, em conformidade com a CLT e convenções sindicais, são utilizados para gerar a escala mensal de trabalho.

Neste trabalho foi considerada uma carga normal de trabalho diário de seis horas e quarenta minutos, valor este que pode variar de acordo com as convenções sindicais de cada região. Jornadas diárias com duração superior a este valor são do tipo hora extra, enquanto as jornadas com duração inferior são chamadas de ociosas. Obviamente uma mesma jornada não pode ser do tipo ociosa e hora extra simultaneamente.

As jornadas do tipo dupla pegada ocorrem por causa dos horários de pico. Devido ao inconveniente das jornadas do tipo dupla pegada, as tripulações que realizam este tipo de jornada possuem um tratamento diferenciado na elaboração da escala mensal de trabalho tendo o seu descanso semanal ocorrendo sempre no domingo.

Para a maior parte das linhas de ônibus, o número de jornadas diárias de trabalho nos dias úteis é maior do que o número de jornadas dos sábados, que por sua vez é maior que o número de jornadas nos domingos. Desta maneira, surgem folgas nos finais de semana denominadas folgas naturais. Esta característica facilita a alocação das folgas previstas na lei, mas não elimina completamente o problema, pois a quantidade de folgas naturais, em geral, não é suficiente para contemplar todas as tripulações.

A Figura 3.3 retirada de Silva e Mayrink (2010) ilustra um exemplo didático de um conjunto de jornadas diárias para uma semana de planejamento. As jornadas de trabalho estão representadas por nós indicados com a letra  $T$  e as folgas naturais estão representadas por nós com a letra  $F$ . Analisando este exemplo, pode-se observar que a soma do número de folgas naturais que surgem nos sábados e domingos não é suficiente para contemplar todas as tripulações. Neste caso têm-se apenas seis folgas para um total de sete tripulações.

A insuficiência de folgas naturais deve ser suprida adicionando novas tripulações ao quadro de funcionários da empresa. Estas tripulações são denominadas *folguistas*. Os folguistas são um tipo especial de tripulação que trabalham para cobrir as jornadas de uma tripulação não folguista quando esta estiver gozando de seu direito de folga.

**Figura 3.3:** Exemplo Fictício de um Conjunto de Jornadas Diárias de Trabalho e Folgas Naturais nos Finais de Semana.

	Seg	Terça	Quar	Qui	Sex	Sab	Dom
Tripulação 1	T	T	T	T	T	T	T
Tripulação 2	T	T	T	T	T	T	T
Tripulação 3	T	T	T	T	T	T	T
Tripulação 4	T	T	T	T	T	T	F
Tripulação 5	T	T	T	T	T	T	F
Tripulação 6	T	T	T	T	T	F	F
Tripulação 7	T	T	T	T	T	F	F

### 3.4 Restrições do Rodízio

O sequenciamento das jornadas diárias deve obedecer a uma série de restrições, tanto de ordem operacional, adotadas pela empresa, quanto trabalhistas, que podem variar de acordo com o município. Neste trabalho o dia foi dividido em quatro turnos, cada um com seis horas de duração. O primeiro corresponde ao intervalo compreendido entre 04:00 e 09:59, o segundo entre 10:00 às 15:59 e assim sucessivamente. Jornadas cujo início se dá entre as 22:00 e 05:00 são consideradas noturnas. Neste intervalo a hora trabalhada é mais valorizada em 20 por cento em relação aos demais horários. As restrições consideradas para o caso estudado são:

1. O horizonte de planejamento sempre se inicia em uma segunda-feira;
2. O horizonte de planejamento sempre compreende um período de 7 semanas.
3. O tempo mínimo de descanso entre o término de uma jornada (conjunto de viagens que uma tripulação realiza durante um dia) e o início da próxima jornada deve ser maior ou igual a 11 horas;
4. Nenhuma tripulação pode trabalhar mais de seis dias consecutivos sem folgar;
5. Dentro de uma mesma semana, as tripulações devem executar jornadas dos dias úteis pertencentes ao mesmo turno de trabalho.

6. Uma jornada pode iniciar em um dia e terminar no outro
7. No decorrer de todos os dias do horizonte de planejamento as tripulações cumprem somente um tipo de jornada, ou seja elas fazem somente pegada simples, dupla pegada ou do tipo noturno ao longo da semana;
8. As tripulações que cumprem jornadas do tipo dupla pegada têm o direito de folgar todos os domingos do horizonte de planejamento;
9. Dentro do horizonte de planejamento todas as tripulações têm direito a pelo menos uma folga no domingo;
10. Para uma mesma tripulação, as horas-extras de uma jornada podem ser compensadas com as horas ociosas de outra jornada atribuída à tripulação deste que executadas no mesmo mês;
11. As folgas são consideradas fixas ao longo do horizonte de planejamento, utilizado o padrão fixo 5/1, ou seja, com um descanso após cinco dias de trabalho,

Baseado nestas características é que foram implementadas as metaheurísticas de resolução para o PRT, apresentadas no capítulo 4.



## Capítulo 4

# Metodologia para Resolução do Problema do Rodízio de Tripulações

Neste capítulo será apresentado o método utilizado neste trabalho para solucionar o PRT de uma empresa brasileira. Foram utilizadas restrições das empresas, assim como restrições legais. Como método de resolução foi empregando a heurística de Busca Tabu e tendo como procedimentos de busca local o método do melhor vizinho (*best improvement*) e o método do primeiro vizinho melhor (*first improvement*). Foi aplicado o padrão de folga 5/1 que é usado na prática e garante que as restrições de folgas sejam contempladas. A partir da solução inicial viável, construída através de um algoritmo de busca gulosa, foi implementada a heurística de Busca Tabu a fim de aprimorar os resultados da função objetivo já obtidos na solução inicial.

A metodologia utilizada para solucionar o Problema de Rodízio das Tripulações foi desenvolvida por meio de dois modelos de otimização distintos. Em uma primeira etapa o objetivo é obter o sequenciamento das jornadas diárias de trabalho para cada uma das tripulações ao longo de todo o horizonte de planejamento, sem considerar a restrição que impede as tripulações de trabalharem mais de seis dias consecutivos e a restrição que garante às tripulações ao menos uma folga no domingo dentro período de planejamento.

Assim foi obtida uma solução que pode apresentar incompatibilidades com as restrições das folgas. Esta solução inicial é utilizada como dados de entrada para o segundo modelo, cujo objetivo é inserir novas tripulações ao quadro de funcionários da empresa. Estas tripulações, denominadas folguistas, devem cobrir as jornadas de trabalho das demais tripulações em alguns dias do horizonte de planejamento de modo que se consiga

atribuir as folgas devidamente ao conjunto de todas as tripulações. O objetivo desta segunda fase consiste em minimizar o custo da função objetivo e o número de tripulações do tipo folguistas.

O PRT apresenta como dados de entrada as jornadas geradas na fase do Problema de Programação diária das Tripulações (PPT). Neste trabalho, a resolução do PRT, ou rodízio das tripulações, foi realizado para um período de sete semanas e foi em quatro etapas.

## 4.1 Representação da Solução

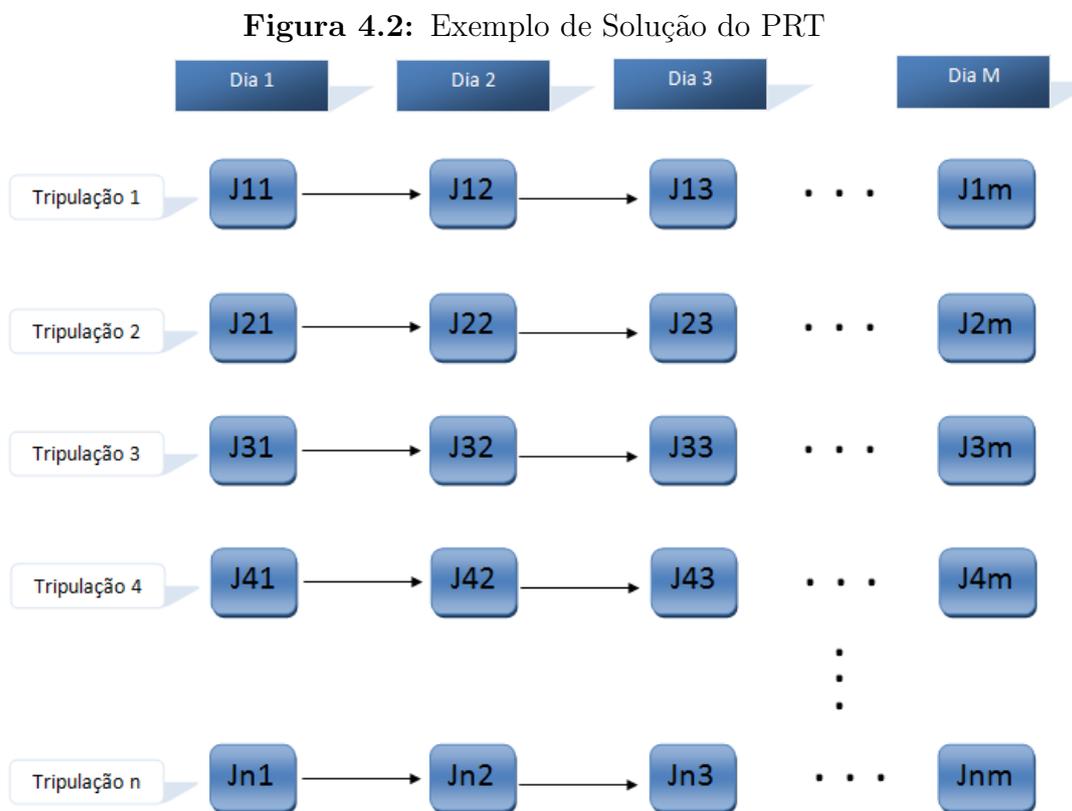
Uma solução para o PRT pode ser retratada como conjunto de rodízios, em que cada um, por sua vez, é associado a uma tripulação. Cada rodízio representa os trabalhos diários que cada tripulação deve executar durante o período de planejamento.

**Figura 4.1:** Exemplo de Jornada

< --- Jornada --- 1 >									
Tarefa	<----Partida---->			<----Chegada---->			Vc	Oc1	Oc2
1	G-02	04:11	D	3050	05:50	01	00	00	
7	D 3050	05:58	D	3050	07:33	01	00	08	
17	D 3050	07:38	D	3050	09:25	01	00	05	
26	D 3050	09:29	D	3050	11:27	01	00	04	
Finaliza a jornada às 11:27									
Horas Ociosas	00:17			Dupla Pegada			0		
Horas Extras	00:36			Troca Veiculo			0		
Troca Ponto	0								

Um rodízio é formado por uma lista de jornadas. Na Figura 4.1 está representada uma jornada real, ou seja, um conjunto de tarefas associadas e suas características. Nela encontram-se os dados das tarefas reunidas, como número, horário de partida, de chegada, e ociosidades. Ao final, são apresentadas as características da jornada como o cálculo final de horas-extras, ociosas, se há troca de ponto, se ela é do tipo dupla-pegada e se há troca de veículo. Na Figura 4.2 mostra-se um exemplo de uma solução do PRT. Cada quadrado é uma jornada associada a uma tripulação em um determinado dia. As

linhas representam as associações dos rodízios às tripulações e as colunas os dias do horizonte de planejamento. Por exemplo, na linha um da Figura 4.2 um rodízio para a tripulação 1 é: jornada  $j_{11}$  no primeiro dia,  $j_{12}$  no segundo, e assim sucessivamente até o último dia  $m$  em que é associada a jornada  $j_{1m}$ . Uma solução para o PRT é um conjunto de rodízios compreendendo todas as tripulações conforme pode ser visto na Figura 4.2.



## 4.2 Solução Inicial

A solução inicial é obtida de duas formas, já que uma contempla todas as restrições através da alocação das folgas e dos folguistas e a outra não.

### 4.2.1 Solução sem Folgas

A fim de padronizar a entrada e possibilitar a comparação com o trabalho de Prates e Silva (2012), foi utilizado o mesmo procedimento para geração das soluções iniciais. Foi utilizado SIGULOSO, um algoritmo baseado em um algoritmo de busca gulosa para geração das soluções iniciais viáveis. Uma solução inicial é formada por um conjunto de rodízios viáveis, porém com valor alto da função objetivo. Na Tabela 5.6 são apresentados os valores das soluções iniciais, a instância TR0, por exemplo possui um valor de custo, ou função objetivo igual a 504.854 e após aplicação do Algoritmo BTMV foi reduzido para 490.840. Cada solução inicial sem folgas foi construída através de um procedimento que começa pela primeira tripulação no primeiro dia e procede até as demais até encontrar uma associação das jornadas que seja viável no primeiro dia. Em cada passo as jornadas foram associadas às tripulações caso sejam viáveis. O procedimento é repetido até que todas as jornadas dos dias no horizonte de planejamento tenham sido associadas às tripulações.

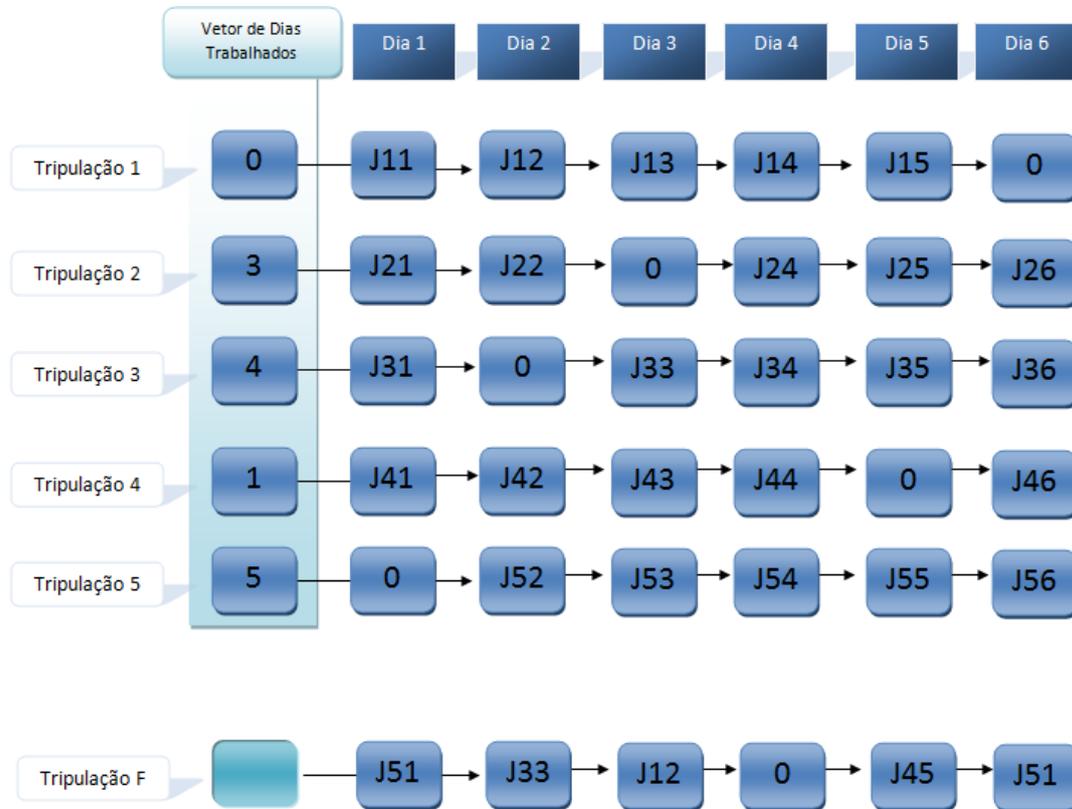
### 4.2.2 Solução com Folgas

Para o cenário em que as folgas foram consideradas na solução, foi criado um vetor com valores aleatórios no intervalo  $[0, 5]$  representam o número de dias trabalhados por cada tripulação. Com base neste vetor é calculado o número de folguistas necessários e os dias em que cada tripulação deve folgar. Para cada tripulação que deve ter uma folga no primeiro dia, considerando o vetor aleatório de dias trabalhados no período anterior, foi alocada uma tripulação do tipo folguistas. Assim, um processo iterativo foi criado a fim de gerar o vetor de dias trabalhados, alcançando o menor número de folguistas necessário para gerar a solução viável para cada instância. Este número é calculado após a criação de todas as folgas necessárias e soma do número de folguista em cada dia. O maior número de folguistas encontrado em um dia representa o número necessário de folguista para o rodízio.

Na Figura 4.3 há um exemplo fictício no qual um folguista é acrescentado para cobrir as folgas das cinco tripulações normais/regulares. Nela está apresentado um Vetor de Dias Trabalhados, em que, cada número contido nos retângulos representa o número de dias que cada tripulação trabalhou sem folgar. A partir desse número é calculada a próxima folga de cada tripulante.

Na Figura 4.3, a primeira tripulação inicia com zero (0) dias trabalhados, o que indica

Figura 4.3: Exemplo de Rodízio com Utilização do Vetor de Dias Trabalhados



que ela teria que trabalhar cinco dias e folgar no sexto. No dia 6 é então associada a uma folga. A segunda tripulação inicia o rodízio com três dias trabalhados. É então feita a vinculação à folga no dia 3, que é o próximo após os cinco dias sem folgar, e assim sucessivamente. É importante notar que o número de folguistas depende dos valores contidos no vetor de dias trabalhados. Diferentes valores podem levar a um número distinto de folguistas. Ao final da criação do vetor de dias trabalhados ele é utilizado como dado de entrada para cada instância. Uma vez definido o vetor de dias trabalhados, é criada uma solução inicial.

### 4.3 Função Objetivo

A função de avaliação, considerando que se deseja minimizar os custos de trabalho da tripulação, pode ser escolhida de duas maneiras distintas. A primeira função utilizada permite a geração de movimentos inviáveis, atribuindo pesos para cada termo de infacilidade de acordo com a prioridade, penalizando-os a fim de evitar que ocorram na

solução final e também que esta seja factível. A função objetivo referente a esta forma é definida da seguinte maneira:

$$fo(s) = \sum_{i=1}^n (\beta_1 \times quantTempoDescanso_i + \beta_2 \times nrTurno_i + \beta_3 \times nrDP_i + \beta_4 \times nrNot_i + \beta_5 \times he_i + \beta_6 \times oc_i) \quad (4.1)$$

em que

- $n$  = Número de tripulações
- $quantTempoDescanso_i$  = Número de horas que foram desrespeitadas (menores que) com relação às 11 horas de descanso previstas para uma tripulação  $i$  entre duas jornadas.
- $nrTurno_i$  = Número de vezes em que ocorreu a troca de turnos (desrespeitos à restrição de turnos) para a tripulação  $i$ .
- $nrDP_i$  = Número de vezes em que uma tripulação foi associada a uma jornadas de tipo dupla pegada diferente do seu para a tripulação  $i$ .
- $nrNot_i$  = Número de vezes em que ocorreu a troca de Jornadas do tipo noturnas para não noturnas.
- $he_i$  = Número de horas extras para a tripulação  $i$ .
- $oc_i$  = Número de horas em que foi menor o número de horas corresponde à jornada e
- $\beta_k$  é a penalização correspondente a cada quesito,  $k \in [1..6]$ .

A segunda função objetivo, que foi a utilizada neste trabalho, considera apenas soluções factíveis tanto para gerar a solução inicial quanto para melhorá-la. Neste caso a função de avaliação coincide com a própria função objetivo do problema, e a solução satisfaz todas as restrições do problema. Esta função representa o custo dos rodízio para as tripulações da empresa que atuam durante o período de planejamento. O custo foi calculado em função do total de horas acumuladas. Elas podem ser ociosas, extras ou nulas, como pode ser visto na equação 4.2, que mostra a função de custo utilizada

na implementação da Busca Tabu. Para cada tripulação é somado o valor das horas de cada jornada associados durante os dias do planejamento. Ao final é gerado um custo do rodízio de cada tripulação, obtido após a compensação, permitida por legislação, entre as horas-extras e ociosas dessa tripulação. Caso o valor seja maior do que zero, ele foi armazenado como horas-extras da tripulação, e as ociosas se tornam nulas. Caso seja menor que zero, as horas ociosas recebem este custo e as extras se tornam nulas. Caso contrário, ambas se tornam nulas

$$Custo(s) = \sum_{i=1}^n (w_1 \times horaextra_i + w_2 \times ociosidade_i) \quad (4.2)$$

Neste trabalho os pesos considerados foram unitários, imputando a mesma importância para ambos os fatores da função em que:

- $n$  = Número de tripulações
- $horaextra_i$  = valor de horas extras da tripulação  $i$ , caso ao final do período de  $m$  dias a soma das horas tenha excedido o número de horas normal.
- $ociosidade_i$  = valor de horas ociosas da tripulação  $i$ , caso ao final do período de  $m$  dias a soma das horas tenha sido menor do que o número de horas normal

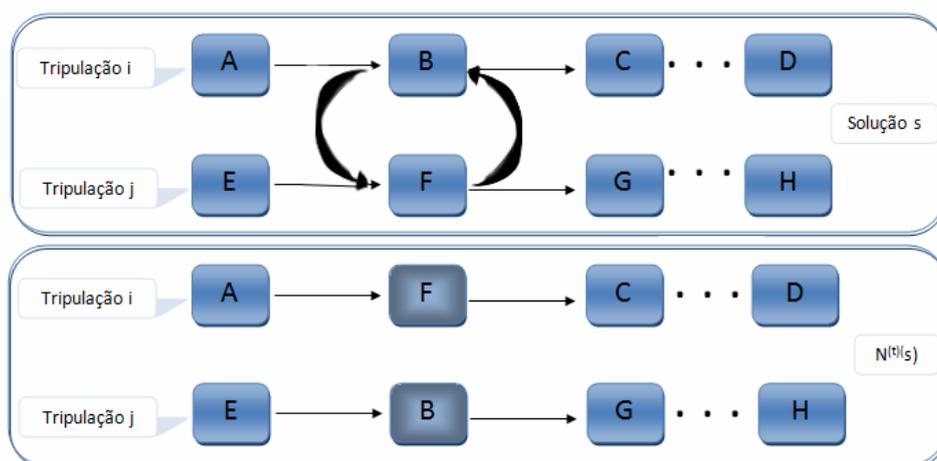
As diferentes funções objetivos utilizam soluções iniciais diversas. Neste problema foi considerada a segunda função, que não permite trocas inviáveis, e foram geradas apenas soluções factíveis. Esta escolha se deve ao fato de que durante o desenvolvimento da dissertação foi percebida a necessidade de se utilizar a função e os mesmos dados de entrada utilizados no trabalho de Prates e Silva (2012) a fim de equiparar os algoritmos para que os resultados pudessem ser comparados.

## 4.4 Estruturas de Vizinhança

Para explorar o espaço de soluções deste problema, foi definido um tipo de movimento que consiste em trocar a jornada entre duas tripulações em um mesmo dia. Desta forma, a vizinhança da solução  $s$  foi formada pelo conjunto de todas as soluções  $s'$  que podem ser obtidas a partir de um movimento de troca de duas jornadas entre duas tripulações em

um mesmo dia. As estruturas de vizinhança devem ser escolhidas para explorar de forma eficiente o espaço de soluções. Quanto maior a vizinhança, melhor será a qualidade da solução local ótima. Por outro lado, vizinhanças maiores requererem um grande tempo de pesquisa (Hertz, Taillard e Werra, 1995).

**Figura 4.4:** Exemplo de um movimento de troca



A Figura 4.4 mostra uma situação que representa um movimento. Na primeira parte é possível observar um pedaço da solução  $s$ , que compreende neste caso dois rodízios referentes às tripulações  $i$  e  $j$ . Nesta figura estão representadas as jornadas diárias com as letras de  $A$  a  $H$ , sendo de  $A$  a  $D$  a seqüência referente ao rodízio  $i$  e de  $E$  a  $H$  a que relacionada ao rodízio  $j$ . Na parte inferior da figura, é mostrada a troca da jornada  $B$  com a jornada  $F$  do rodízio  $j$ , o que representa uma possibilidade de vizinho da solução  $s$ , obtida pela troca entre duas jornadas dos rodízios  $i$  e  $j$ .

## 4.5 Busca Local

Neste trabalho foram exploradas três estratégias de busca local cuja descrição é feita a seguir.

- **Busca do tipo melhor vizinho** Na primeira estratégia, a vizinhança  $V$  corresponde a  $N(s)$ , ou seja, é feita uma busca de forma a encontrar o melhor vizinho dentre de todos os vizinhos obtidos pela troca entre duas jornadas quaisquer de tripulações diferentes que sejam factíveis

- **Busca do tipo primeiro melhor vizinho** Na segunda estratégia, a busca pelo primeiro melhor vizinho é feita sobre todo o conjunto de possíveis vizinhos, mas desta vez interrompe-se a exploração quando um vizinho melhor do que o primeiro é encontrado, parando no primeiro ótimo local. Assim, apenas no pior caso a vizinhança  $V$  equipara-se ao conjunto  $N(s)$ . Neste Algoritmo a ordem de exploração da vizinhança é alterada a cada passo a fim de evitar que favoreça sempre um mesmo caminho determinístico.
- **Busca do melhor vizinho para uma parte do espaço de soluções.** Na terceira estratégia, a vizinhança  $V$  equivale a um percentual variável de todo o conjunto de possíveis vizinhos  $N(s)$ . A busca ocorre de forma aleatória a cada iteração da Busca Tabu iniciando a busca num percentual de 5% da vizinhança, aumentando 10% cada vez que atinge um número máximo de iterações sem melhora até atingir 100%.

A estrutura de vizinhança está associada à quantidade de modificações realizadas a partir de cada estratégia. Na Figura 4.4 é possível ver a geração de vizinhos utilizando os tipos de movimentos desenvolvidos.

## 4.6 Busca Tabu

Nesta seção serão apresentadas as características utilizadas para as implementações de Busca Tabu utilizadas neste trabalho

### 4.6.1 Memória de Curto Prazo - Lista Tabu

As regras de proibição foram associadas aos atributos da solução de forma a evitar o retorno a uma solução visitada anteriormente. Os atributos dessa solução se tornam então tabu-ativos e uma solução que contenha os atributos tabu-ativos se torna tabu. Um movimento é tabu se ele der origem a uma solução tabu.

A lista clássica tabu, uma lista de movimentos proibidos, possui os movimentos reversos aos últimos  $|T|$  movimentos realizados, onde  $|T|$  é um parâmetro do Algoritmo que indica o tamanho da lista. Ela funciona como uma fila de tamanho fixo, isto é, quando um novo movimento é adicionado à lista, o mais antigo sai dela (Souza, 2013).

Assim, na exploração do subconjunto  $V$  da vizinhança  $N(s)$  da solução corrente  $s$ , ficam excluídos da busca os vizinhos  $s'$  que foram obtidos de  $s$  por movimentos  $m$  que constam na lista tabu a menos que o movimento gere uma solução melhor do que a melhor solução conhecida.

### **4.6.2 Critério de Aspiração e de Parada da Busca Tabu**

Para evitar proibição de movimentos que levem a soluções que ainda não foram visitadas existe também uma função de aspiração, que é um mecanismo que retira, sob certas circunstâncias, o status tabu de um movimento. Mais precisamente, para cada possível valor  $v$  da função objetivo existe um nível de aspiração  $A(v)$  que diz que uma solução  $s$  em  $V$  pode ser aceita, mesmo que o movimento esteja na lista tabu, caso o valor da função seja menor do que o valor do nível de aspiração.

Neste trabalho foi utilizada a aspiração por objetivo, que retira o status tabu caso um movimento que se encontre na lista possua valor melhor (neste caso menor) do que o melhor movimento encontrada até então  $s^*$ .

Foram definidos dois critérios para interrupção da busca Tabu:

- Número de iterações sem melhora
- Tempo máximo de processamento

### **4.6.3 Seleção dos Atributos do Movimento**

Quando se movimenta de uma solução para outra que está em sua vizinhança, há uma diferença muito pequena entre uma solução e outra. Apenas duas jornadas foram trocadas entre duas tripulações e conseqüentemente seus valores de horas e custos referentes às respectivas tripulações. Ao invés de armazenar toda uma solução na lista, armazena-se nela apenas os atributos do movimento que diferenciam uma solução da outra. Neste caso, para cada movimento da lista, são armazenados os seguintes atributos:

1. Número referente ao dia em que a troca ocorreu
2. Número da primeira tripulação da troca
3. Número da segunda tripulação da troca

A estrutura utilizada para a lista tabu é um vetor de estruturas (cada uma para os três inteiros dos atributos do movimento)

## 4.7 Implementações da Metaheurística de Busca Tabu

O pseudocódigo da heurística básica de Busca Tabu pode ser visto no Algoritmo 4.1. Primeiramente foram desenvolvidas três versões deste Algoritmo modificando apenas o critério de busca local. Posteriormente foi desenvolvida mais uma versão da Busca Tabu utilizando critérios de intensificação e de diversificação da Busca.

---

**Algoritmo 4.1:** Busca Tabu

---

```
1:  $s^* \leftarrow -s$ 
2:  $Iter \leftarrow -0$ 
3:  $MelhorIter \leftarrow -0$ 
4:  $T \leftarrow -\emptyset$ 
5: Inicialize a função de aspiração  $A$ 
6: while Critério de parada não satisfeito do
7:    $Iter \leftarrow -Iter + 1$ ;
8:   Seja  $s' \leftarrow -s \oplus m$  o elemento dentre a  $V \subset N(s)$  com menor valor de função de custo tal que o movimento  $m$  não seja tabu ( $m \notin T$ ) ou  $f(s') < A(f(s))$ 
9:   Atualize a lista tabu;
10:   $s \leftarrow -s'$ ;
11:  if  $f(s) < f(s^*)$  then
12:     $s^* \leftarrow -s$ ;
13:     $MelhorIter \leftarrow -Iter$ ;
14:  end if
15:  Atualize a função de aspiração  $A$ ;
16: end while
17:  $s \leftarrow -s^*$ ;
18: retorne  $S^*$ ;
```

---

### 4.7.1 Melhor Vizinho - BTMV

Nesta versão da busca local foram utilizados os procedimentos de geração das duas soluções iniciais descritos na seção 4.2, a primeira sem considerar as folgas e a segunda considerando-as. O melhor vizinho é buscado em toda a vizinhança a cada iteração até ocorrerem  $BTMax$  iterações sem melhora ou ser alcançado um tempo máximo de

execução. Toda a vizinhança é analisada independentemente de já se ter encontrado uma solução melhor. É importante salientar que neste caso a vizinhança  $V$  corresponde a  $N(s)$ .

### **4.7.2 Primeira Melhora - BTPM**

Esta versão corresponde ao Algoritmo BTMV de 4.7.1 com a diferença na busca pelo melhor vizinho, que neste caso a cada iteração as trocas são realizadas em uma nova sequência aleatória, através das trocas de jornadas entre duas tripulações em um mesmo dia. O procedimento é interrompido no primeiro movimento que supere a melhor solução já encontrada, sendo que no pior caso, de não haver vizinho melhor, é feita uma busca na vizinhança completa. A idéia de se reduzir o espaço de busca decorre de a iteração do Algoritmo de Busca Tabu ser custosa computacionalmente para este problema.

### **4.7.3 Porcentagem de Vizinhança Variável - BTPV**

Esta versão utiliza o procedimento de geração da solução inicial descrito na seção 4.2. Além disso, procura-se o melhor vizinho dentro de um percentual de 5% caso não haja sucesso sendo aumentado em 10% até o máximo de 100%. Neste caso embora a vizinhança varie a cada iteração, a análise é feita sobre todo o percentual da vizinhança independente de já existir alguma solução de melhora. Nesta estratégia o objetivo é baratear o custo no sentido de exploração de vizinhança.

### **4.7.4 Algoritmo de Busca Tabu com Intensificação e Diversificação - BTID**

De forma a intensificar a busca em regiões promissoras, é importante voltar às soluções elite já encontradas. Então o tamanho da lista tabu é diminuído de acordo com Hertz, Taillard e Werra (1995).

Na diversificação, conforme apontado em Hertz, Taillard e Werra (1995) o objetivo é evitar que uma grande região do espaço de soluções permaneça completamente inexplorada. A forma mais simples de se diversificar é fazer vários reinícios da busca partindo de diferentes soluções. Outra forma que para tentar explorar regiões não visitadas é penalizar movimentos realizados várias vezes ou soluções bastante visitadas.

No caso em que soluções factíveis tem que satisfazer um conjunto de restrições, elas se tornam montanhas intransponíveis (ou de tamanho infinito). Relaxar essas restrições e penalizar suas violações corresponde a reduzir o tamanho das montanhas e passar a barreira para alcançar outro vale em poucos passos. Durante a fase de diversificação, as soluções visitadas não são necessariamente factíveis. Para obter uma solução factível novamente as penalidades são gradualmente aumentadas (Hertz, Taillard e Werra, 1995).

A intensificação ocorre quando é atingido um número máximo de iterações sem melhora. A partir daí o tamanho da lista tabu é diminuído em 10% e o tamanho da vizinhança é alterado para 25% maior. O contrário ocorre com a diversificação em que o tamanho da vizinhança é diminuído em 25% e a lista aumentado em 10%, como pode ser visto no Algoritmo 4.2. Na linha 1, a solução inicial é associada à melhor solução atual. Na linha 7 é verificado o critério de parada (número de iterações sem melhora menor que BTMAX e o tempo menor que o tempo máximo para aquela instância). Os passos de 9 a 18 representam a intensificação e diversificação, que são as principais diferenças da Busca Tabu clássica.

## 4.8 Algoritmo VNS-VND

Prates e Silva (2012) propuseram o Algoritmo VNS-VND para abordar o Problema do Rodízio de Tripulações de uma empresa de transporte Público Urbano que opera na cidade de Belo Horizonte - MG. Este Algoritmo é baseado no VNS clássico e utiliza VND como método de busca local. Este trabalho será exposto com mais detalhes, visto que será objeto de comparação com os algoritmos propostos nesta dissertação.

O Algoritmo de Pesquisa em Vizinhança Variável conhecido como VNS (*Variable Neighborhood Search*), proposto por Mladenovic e Hansen (1997), é utilizado para a resolução de problemas combinatórios e globais. Ele é um algoritmo de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. À medida que as estruturas de vizinhança mudam, o caminho de busca é modificado explorando soluções mais distantes da solução corrente. A versão original utiliza o VND como método de busca local.

A implementação do VNS-VND parte de uma solução inicial  $s^0$  que é adotada como solução corrente  $s$  e a cada iteração uma nova solução  $s'$  é criada através de mudanças aleatórias na solução corrente dentro de uma estrutura de vizinhança  $N^k(s)$ . As mu-

---

**Algoritmo 4.2:** BTID - Implementação da Busca Tabu com critérios de intensificação e diversificação

---

```

1:  $s^* \leftarrow -s$ 
2:  $Iter \leftarrow -0$ 
3:  $MelhorIter \leftarrow -0$ 
4: limpaListaTabu
5:  $T \leftarrow \emptyset$ 
6: Inicialize a função de aspiração  $A$ 
7: while Critério de parada não satisfeito do
8:    $Iter \leftarrow -Iter + 1$ ;
9:   if critério de intensificação then
10:     intensifica;
11:      $|V| * = 1, 25$ ;
12:      $|T| * = 0, 9$ ;
13:   end if
14:   if critério de diversificação then
15:     diversifica;
16:      $|V| * = 0, 75$ ;
17:      $|T| * = 1, 1$ ;
18:   end if
19:   gerar um número aleatorio para limitar a vizinhanca;
20:   encontra o melhor vizinho em  $V$  Seja  $s1 =$  melhor elemento de  $V$  tal que não
   seja tabu ou obedeça um critério de aspiração;
21:   insere na Lista Tabu;
22:   troca a posicao do melhor vizinho encontrado;
23:   if  $f(s) < f(s^*)$  then
24:      $melhoriter = iteratual$ ;
25:      $fo^* = fo$ ;
26:      $s^* = s$ ;
27:   end if
28: end while
29:  $s \leftarrow -s^*$ ;
30: retorne  $s$ ;

```

---

danças foram dependentes do parâmetro  $k$ , que indica o tipo de vizinhança que será utilizada. Nesta implementação a vizinhança determina a quantidade de modificações realizadas.

Assim, ao iniciar em  $N^1(s)$ , é realizado um movimento de troca aleatório de uma jornada de duas tripulações em um determinado dia, atingindo uma nova solução  $s$ . Então se aplica um procedimento de busca local a partir de  $s'$  encontrando uma nova solução  $s''$ . Caso  $s''$  se mostre melhor do que a solução atual, uma atualização é feita a fim de retornar à primeira estrutura de vizinhança  $N^{k+1}(s)$ , caso contrário o valor de  $k$  é incrementado até  $N^{kmax}(s)$ . O valor de  $k$  é incrementado até o número máximo de estruturas de vizinhança  $N^1(s)$  e continua até que a condição de parada seja alcançada. Nesta implementação foi utilizado o tempo de processamento como condição de parada. O pseudocódigo do VNS é mostrado em 4.4

O Algoritmo VNS também possui um procedimento de busca local a ser aplicado sobre o vizinho da solução corrente, o Algoritmo VND, também proposto por Mladenovic e Hansen, (1997). Ele tem como dados iniciais uma solução viável, uma função de avaliação e um índice que indica as estruturas de vizinhança que serão utilizados pelo método de busca do melhor vizinho. Caso a solução resultante da busca obtenha uma melhora na função de avaliação com relação à solução atual, ela é escolhida como solução corrente e o procedimento então volta à primeira estrutura de vizinhança  $N^1(s)$ . Caso contrário, o processo vai para a próxima estrutura de vizinhança  $N^{k+1}(s)$ . O Algoritmo é interrompido quando todas as estruturas de vizinhança tenham sido exploradas, retornando a melhor solução encontrada, conforme apresentado no Algoritmo 4.4.

O VNS-VND desenvolvido por Prates e Silva (2012) utiliza três estruturas de vizinhança. A primeira estrutura consiste em fixar duas tripulações e realizar as trocas em  $k$  dias diferentes, caracterizando assim  $k$  estruturas de vizinhança. A escolha das duas tripulações diferentes  $i$  e  $j$  é feita de forma aleatória, além de um dia  $d1$ . De acordo com Prates e Silva (2012) é feita a troca da jornada da tripulação  $i$  com a jornada da tripulação  $j$  no dia  $d1$ . Um novo dia  $d2$  diferente do anterior é sorteado, e foram fixadas as tripulações  $i$  e  $j$ . As jornadas das duas tripulações foram trocadas para este novo dia. E assim foram realizadas  $k$  trocas de jornadas entre as mesmas tripulações anteriormente selecionadas. É então analisado o custo final das trocas com base na função de custo. Caso as trocas representem uma melhora no custo elas foram efetuadas, caso contrário foram sorteadas novas tripulações e reinicia-se o procedimento.

A segunda estrutura de vizinhança consiste no sorteio de um dia  $d$ , permanecendo

este fixo e os pares de tripulações foram escolhidos aleatoriamente. Os pares de jornadas candidatas à troca foram obtidos do sorteio aleatório das tripulações até atingir o número total de  $k$  trocas. Caso este conjunto melhore a função de custo, as trocas foram realizadas, caso contrário o processo se reinicia com o sorteio de outro dia.

Segundo os autores uma terceira estrutura de vizinhança foi implementada através da união das duas estruturas de vizinhança já mostradas. Neste caso foi utilizada a primeira estrutura de vizinhança para gerar as perturbações da solução corrente e a segunda estrutura de vizinhança foi utilizada para realizar a busca local do VND. Empregando as três estruturas de vizinhança foram obtidas três versões da metaheurística VNS para a resolução do PRT.

Os autores testaram os algoritmos considerando dados relativos a uma empresa que atua em Belo Horizonte - MG. Foram conduzidos testes iniciais para calibrar o parâmetro  $k$  de cada uma das estruturas de vizinhança. O número de execuções realizadas para cada valor de  $k$ , de acordo com os autores foi igual a cinco. Com os parâmetros que produziram as melhores soluções foram realizados outros dez testes, os quais produziram as diferentes soluções finais para o PRT.

---

**Algoritmo 4.3:** Pseudo-código do procedimento VNS-VND

---

**Require:**  $s$

- 1: Seja um conjunto de  $kmax$  estruturas de vizinhanças;
  - 2: Encontre uma solução inicial  $s$  e determine uma condição de parada;
  - 3: **while** critério de parada não satisfeito **do**
  - 4:    $k \leftarrow 1$ ;
  - 5:   **while**  $K \leq kmax$  **do**
  - 6:     Gere um novo vizinho  $s' \in N^k(s)$ ;
  - 7:      $s'' = VND(s \text{ de } s' \text{ na vizinhança } k)$ ;
  - 8:     **if**  $f(s'') < f(s)$  **then**
  - 9:        $s \leftarrow s''$ ;
  - 10:        $k \leftarrow 1$ ;
  - 11:     **end if**
  - 12:     **else**
  - 13:       **if**  $!f(s'') < f(s)$  **then**
  - 14:          $k \leftarrow k + 1$ ;
  - 15:       **end if**
  - 16:     **end while**
  - 17: **end while**
  - 18: **retorne**  $s$ ;
-

---

**Algoritmo 4.4:** Pseudo-código do procedimento VND

---

**Require:**  $s, k$ 

- 1: Seja  $s$  uma solução corrente dada
  - 2: Seja  $k$  uma estrutura de vizinhança dada;
  - 3:  $r \leftarrow -1$
  - 4: **while**  $r \leq k$  **do**
  - 5:   Encontre o melhor vizinho  $s'de(s)$  na vizinhança  $r$ ;
  - 6:   **if**  $f(s') < f(s)$  **then**
  - 7:      $s \leftarrow s'$ ;
  - 8:      $r \leftarrow r + 1$ ;
  - 9:   **end if**
  - 10:   **else**  $r \leftarrow r + 1$ ;
  - 11: **end while**
  - 12: **retorne**  $S$ ;
-



# Capítulo 5

## Experimentos Computacionais

Neste capítulo são mostrados os testes computacionais realizados com os algoritmos desenvolvidos. Para testar os algoritmos implementados foram utilizadas 13 instâncias-teste descritas na Seção 5.1. O número de instâncias para este problema é pequeno pois são derivados de dados reais e não foi possível obter dados de outras instâncias. Inicialmente, em 5.2.1, são comparados os resultados para as versões sem folgas dos Algoritmos baseados em Busca Tabu BTPM, BTMV , BTPV e BTID e o Algoritmo VNS-VND, proposto por Prates e Silva (2012), é apresentado na Seção 5.2.2. Esses algoritmos foram aplicados às instâncias da empresa e os resultados são apresentados e discutidos em 5.2.3. Posteriormente os mesmos procedimentos foram realizados para as versões com folgas, a partir da Seção 5.3. Todos os testes foram feitos em um computador pessoal com as seguintes características

- Processador Intel(R) Core I5
- CPU@2.3 GHz Memória
- 4 GB de Memória RAM
- Sistema Operacional Windows 7 de 32 Bits

As versões da Busca Tabu e da metaheurística VNS foram implementadas na linguagem C++ através da ferramenta Borland C++ Builder Professional Versão 5.0. Inicialmente foram realizados testes a fim de medir os melhores valores para parâmetros para cada instância:

- *MAXTABULISTSIZE* - 60 Tamanho da lista da Busca Tabu
- *BTMAX* - 600 Número máximo de iterações sem melhora da Busca Tabu
- *TEMPOEXECUCAO* - Tempo máximo de execução dos procedimentos - Tabela 5.2
- *MAXDIAS* - 100% Tamanho máximo da vizinhança para os procedimentos BTPV e BTID
- Parâmetro  $k$  de cada uma das estruturas de vizinhança para o VNS

O número de execuções realizadas para determinar o valor dos parâmetros acima foi igual a seis, sendo que cada execução teve o tempo de uma hora e meia para cada instância. Assim foi possível obter as melhores soluções. Todos os testes foram feitos considerando o mesmo tempo de execução e mesma entrada para cada comparação dos algoritmos. Com os parâmetros calibrados, as versões não determinísticas BTPM, BTPV, BTID e VNS-VND foram executadas 10 vezes para cada instância de forma a verificar a robustez dos algoritmos implementados. Outro detalhe importante é que os algoritmos foram testados para as soluções iniciais em função da contemplação ou não das folgas. As Tabelas 5.1 e 5.2 mostram os tempos para cada instância.

## 5.1 Instâncias de Teste

As instâncias utilizadas como dados de entrada para este problema são compostas pelas jornadas geradas na fase de programação de tripulações e são previstas para serem executadas em três tipos de dias diferentes do período de planejamento: dias úteis, sábados e domingos. Para este trabalho, foram utilizadas as 13 instâncias que estão apresentadas na Tabela 5.3. Esta tabela mostra a quantidade de jornadas em cada tipo de dia que devem ser cumpridas tanto para o rodízio com folgas quanto para o rodízio sem folgas. As instâncias denominadas TR0 e a NR104, se referem a dados reais de uma empresa que atua em Belo Horizonte(MG). As demais instâncias são derivadas do particionamento de suas linhas. Para cada instância derivada, foi gerado um conjunto de jornadas a serem utilizadas e verificada a viabilidade de realização do rodízio através da solução inicial que contempla as restrições. O período trabalhado não possui ocorrência de feriados.

**Tabela 5.1:** Tabela de tempos de execução - Sem folgas

<b>Instâncias</b>	<b>Tempo de execução</b>
TR0	365
TR1	245
TR3	67
TR5	16
TR4	30
TR6	600
TR7	875
TR8	108
TR9	30
TR10	30
NR24	30
NR104	134
V104	45

**Tabela 5.2:** Tabela de tempos de execução - Com folgas

<b>Instâncias</b>	<b>Tempo de execução</b>
TR3	67
TR5	16
TR4	30
TR7	875
TR10	30
NR24	30
NR104	134

As Tabelas 5.3, 5.4 e 5.5 apresentam características dos dados de entrada utilizados neste trabalho. Na Tabela 5.4 estão apresentados os dados de tipo noturno e dupla pegada de cada uma das instâncias, e na Tabela 5.5 são contabilizadas as horas extras e ociosas contidas nas jornadas dessas instâncias expressas em minutos. Por exemplo, na primeira instância, TR0, são realizadas 279 jornadas de trabalho em dias úteis sendo 101 jornadas do tipo noturno e 21 do tipo dupla pegada, com 14.770 minutos e 2.311 minutos de ociosidades.

**Tabela 5.3:** Tabela de instâncias - Número de jornadas

Instâncias	Número de jornadas		
	Dias Úteis	Sábados	Domingos
TR0	279	213	149
TR6	231	166	100
TR7	214	150	95
TR3	205	136	115
TR4	205	132	65
TR1	125	30	27
NR104	104	60	53
TR8	87	33	33
TR5	53	30	21
TR9	43	22	14
VR104	40	21	19
NR24	24	18	14
TR10	14	6	5

## 5.2 Solução sem Folgas

Primeiramente as soluções iniciais foram geradas através do Algoritmo SIGULOSO, definido no capítulo anterior, e os resultados foram salvos em arquivos texto para que todas as versões dos algoritmos utilizassem a mesma solução inicial. As características de horas-extras e ociosas das soluções iniciais sem folgas podem ser conferidas na Tabela 5.6.

### 5.2.1 Resultados dos Algoritmos de Busca Tabu - Sem folgas

As Tabelas 5.7, 5.8, 5.9, 5.10 mostram respectivamente os resultados dos algoritmos BTPM, BTMV, BTPV, e BTID. Para os dois últimos resultados mostrados nas Tabelas 5.9, 5.10, por serem resultados de algoritmos não determinísticos, foram calculadas as médias de todos os resultados, dentre as dez rodadas com cada instância. A Tabela 5.12 mostra as porcentagens de melhorias entre os resultados contidos nas Tabelas 5.7, 5.8, 5.9, 5.10 e a Tabela 5.6. As piores porcentagens de melhorias para todas as tabelas de

**Tabela 5.4:** Tabela de características de dupla pegada e noturno das instâncias (Número de jornadas de cada tipo)

Instâncias	Noturno			Dupla pegada		
	Dias Úteis	Sábados	Domingos	Dias Úteis	Sábados	Domingos
TR0	101	83	69	21	0	0
TR1	42	15	13	4	0	1
TR3	67	52	49	15	0	1
TR5	20	15	10	2	0	1
TR4	67	48	26	15	0	1
TR6	85	65	50	10	0	0
TR7	77	56	48	8	0	0
TR8	30	8	17	4	0	1
TR9	16	10	7	1	0	1
TR10	6	2	1	1	0	0
NR24	4	4	4	3	3	3
NR104	13	10	9	4	0	0
VR104	4	5	3	1	0	0

**Tabela 5.5:** Tabela de horas extras e ociosas das instâncias utilizadas (em minutos)

Instâncias	Horas extras			Ociosas		
	Dias Úteis	Sábados	Domingos	Dias úteis	Sábados	Domingos
<b>TR0</b>	14.770	8.491	6.259	2.311	4.501	2.540
<b>TR1</b>	5.494	1.523	1.045	6.618	1.051	3.201
<b>TR3</b>	10.142	5.787	4.396	4.745	3.625	4.269
<b>TR5</b>	2.596	1.523	924	4.522	1.051	3.065
<b>TR4</b>	10.142	5.586	2.406	4.745	3.601	3.255
<b>TR6</b>	11.657	6.631	4.276	3.987	3.206	2.423
<b>TR7</b>	10.340	5.764	3.928	4.023	2.603	2.285
<b>TR8</b>	4.643	1.021	1.386	4.711	1.033	3.209
<b>TR9</b>	2.011	1.031	573	4.646	1.027	3.011
<b>TR10</b>	900	243	180	3.361	555	454
<b>NR24</b>	1.161	1.161	2.589	2.589	2.589	1.161
<b>NR104</b>	3.748	2.233	1.645	4.702	1.277	961
<b>V104</b>	1.575	960	669	1.994	425	447

algoritmos tabu (5.7 5.8, 5.9, 5.10) foram da instância TR0 com 2,78%, 2,776%, 2,776%, 2,842% de melhoria respectivamente. As maiores melhorias foram da instância V104 com 31,35%, 35,014%, 34,313%, 34,756% de melhorias respectivamente.

### 5.2.2 Resultados - VNS-VND - Sem folgas

A fim de realizar a comparação dos resultados dos algoritmos, foram feitos 10 testes para o VNS-VND, com cada vizinhança, e calculada a média (de acordo com a função 5.1) para cada instância. Esta, por sua vez, foi utilizada para comparação com os algoritmos derivados da Busca Tabu. De forma a possibilitar a comparação, foi utilizada a mesma função objetivo mostrada no capítulo 4, seção 4.3, equação 4.2 e a mesma solução inicial para todos os testes.

$$Media = \frac{\sum_{i=1}^{10} CustoTotal_i}{10} \quad (5.1)$$

Em que  $i$  é o número de vezes que o Algoritmo foi testado  $CustoTotal$  é o custo final de cada teste do Algoritmo para cada instância.

Na Tabela 5.11 estão os resultados do Algoritmo VNS-VND sem folgas, cujos valores de horas-extras, ociosas e custo variam de acordo com o tamanho das instâncias.

### 5.2.3 Comparação entre os resultados dos algoritmos BTPM, BTMV, BTPV, BTID e VNS-VND - Sem folgas

Dos resultados obtidos foram calculadas os percentuais de melhoria com relação à solução inicial entre os custos (que levam em conta os valores de horas-extras e ociosas) das soluções e da solução inicial e na Tabela 5.12 estão apresentados os percentuais de melhoria e as médias de cada algoritmo em relação à solução inicial. A fórmula utilizada para cálculo destes percentuais pode ser vista em 5.2. Com base nesta tabela, pode-se verificar que o Algoritmo BTMV obteve a melhor média.

A fórmula utilizada pode ser dada por:

$$Melhoria(s) = \frac{(SolInicial - SolMetodo)}{SolInicial} \quad (5.2)$$

**Tabela 5.6:** Tabela de valores em minutos de horas extras e ociosas das soluções iniciais - Sem folga

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR0</b>	497.252	7.602	504.854
<b>TR1</b>	133.312	184.118	317.430
<b>TR3</b>	325.605	120.722	446.327
<b>TR5</b>	58.043	137.227	195.270
<b>TR4</b>	321.150	124.394	445.544
<b>TR6</b>	326.724	21.559	348.283
<b>TR7</b>	283.502	28.387	311.889
<b>TR8</b>	133.199	148.767	281.966
<b>TR9</b>	47.109	156.155	203.264
<b>TR10</b>	26.460	116.942	143.402
<b>NR24</b>	34.561	51.354	85.915
<b>NR104</b>	77.170	98.849	176.019
<b>V104</b>	33.040	42.518	75.558

**Tabela 5.7:** Resultados obtidos através da aplicação do Algoritmo BTPM - Sem folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR0</b>	490.245	595	490.840
<b>TR1</b>	92.160	142.966	235.126
<b>TR3</b>	313.708	108.825	422.533
<b>TR5</b>	47.544	126.728	174.272
<b>TR4</b>	310.129	113.373	423.502
<b>TR6</b>	305.750	585	306.335
<b>TR7</b>	255.680	565	256.245
<b>TR8</b>	118.134	133.702	251.836
<b>TR9</b>	36.981	146.027	183.008
<b>TR10</b>	17.745	108.227	125.972
<b>NR24</b>	24.437	41.230	65.667
<b>NR104</b>	63.190	84.869	148.059
<b>V104</b>	21.195	30.673	51.868

**Tabela 5.8:** Resultados obtidos através da aplicação do Algoritmo BTMV - SemFolgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR0</b>	490.245	595	490.840
<b>TR1</b>	91.225	142.031	233.256
<b>TR3</b>	311.003	106.120	417.123
<b>TR5</b>	47.684	126.868	174.552
<b>TR4</b>	302.036	105.280	407.316
<b>TR6</b>	305.725	560	306.285
<b>TR7</b>	255.675	560	256.235
<b>TR8</b>	117.808	133.376	251.184
<b>TR9</b>	34.076	143.122	177.198
<b>TR10</b>	17.745	108.227	125.972
<b>NR24</b>	24.239	41.032	65.271
<b>NR104</b>	63.272	84.951	148.223
<b>V104</b>	19.812	29.290	49.102

**Tabela 5.9:** Resultados obtidos através da aplicação do Algoritmo BTPV - Sem folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR0</b>	490.245	595	490.840
<b>TR1</b>	91.710	142.516	234.226
<b>TR3</b>	311.015	106.132	417.146
<b>TR5</b>	47.684	126.868	174.552
<b>TR4</b>	303.632	106.876	410.508
<b>TR6</b>	305.730	565	306.295
<b>TR7</b>	255.679	564	256.243
<b>TR8</b>	118.098	133.666	251.764
<b>TR9</b>	34.076	143.122	177.198
<b>TR10</b>	17.745	108.227	125.972
<b>NR24</b>	24.406	41.199	65.605
<b>NR104</b>	64.407	86.086	150.493
<b>V104</b>	20.077	29.555	49.632

**Tabela 5.10:** Resultados obtidos através da aplicação do Algoritmo BTID - Sem folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR0</b>	489.912	593	490.505
<b>TR1</b>	91.593	142.499	234.093
<b>TR3</b>	310.848	106.130	416.978
<b>TR5</b>	47.684	126.851	174.535
<b>TR4</b>	303.299	106.875	410.173
<b>TR6</b>	305.397	548	305.945
<b>TR7</b>	254.012	562	254.575
<b>TR8</b>	117.931	133.649	251.581
<b>TR9</b>	34.047	143.120	177.167
<b>TR10</b>	17.578	108.210	125.789
<b>NR24</b>	24.239	41.197	65.437
<b>NR104</b>	64.240	86.069	150.310
<b>V104</b>	19.744	29.553	49.297

**Tabela 5.11:** Resultados obtidos através da aplicação do Algoritmo VNS-VND - Sem folgas

Instâncias	Hora extra	Ociosidade	Custo
<b>TR0</b>	490.399	859	491.258
<b>TR1</b>	105.167	155.973	261.140
<b>TR3</b>	314.498	109.615	424.113
<b>TR5</b>	48.034	127.218	175.252
<b>TR4</b>	308.780	112.024	420.804
<b>TR6</b>	314.205	9.040	323.246
<b>TR7</b>	272.039	17.117	289.155
<b>TR8</b>	125.069	140.660	265.729
<b>TR9</b>	34.146	143.192	177.338
<b>TR10</b>	18.760	109.242	128.002
<b>NR24</b>	31.251	47.685	78.936
<b>NR104</b>	77.165	98.547	175.711
<b>V104</b>	23.320	32.870	56.190

**Tabela 5.12:** Percentuais de melhorias com relação à solução inicial - Sem Folgas

Instâncias	BTPM	BTMV	BTPV	BTID	VNS
<b>TR0</b>	2,78%	2,776%	2,776%	2,842%	1,378%
<b>TR1</b>	25,93%	26,517%	26,212%	26,254%	21,112%
<b>TR3</b>	5,33%	6,543%	6,538%	6,576%	3,411%
<b>TR5</b>	10,75%	10,610%	10,610%	10,618%	17,244%
<b>TR4</b>	4,95%	8,580%	7,864%	7,939%	3,852%
<b>TR6</b>	12,04%	12,059%	12,056%	12,156%	3,832%
<b>TR7</b>	17,84%	17,844%	17,842%	18,377%	4,043%
<b>TR8</b>	10,69%	10,917%	10,711%	10,776%	6,103%
<b>TR9</b>	9,97%	12,824%	12,824%	12,839%	27,517%
<b>TR10</b>	12,15%	12,155%	12,155%	12,282%	29,101%
<b>NR24</b>	23,57%	24,028%	23,640%	23,836%	9,577%
<b>NR104</b>	15,88%	15,791%	14,502%	14,606%	0,007%
<b>V104</b>	31,35%	35,014%	34,313%	34,756%	29,419%
<b>Média</b>	<b>14,09%</b>	<b>15,05%</b>	<b>14,77%</b>	<b>14,91%</b>	<b>12,05%</b>

## 5.3 Solução com Folgas

Assim como na solução sem folgas, as soluções iniciais foram geradas através do Algoritmo SIGULOSO e os resultados foram salvos em arquivos texto para que todas as versões dos algoritmos utilizassem a mesma solução inicial. No entanto foram utilizadas apenas as instâncias TR3,TR5,TR4,TR7,TR10,NR24 e NR104, pois foram as soluções que geraram soluções iniciais com folgas factíveis. Os dados de resultados das soluções iniciais com folgas podem ser conferidos na Tabela 5.13.

Os tempos correspondem ao tempo necessário para os algoritmos de melhoria da solução inicial. As soluções iniciais foram salvas em arquivos texto e foram lidas a cada execução dos algoritmos deste trabalho. Dessa forma há uma garantia de utilização da mesma solução inicial para cada instância em todos os algoritmos testados.

Posteriormente foram feitos vários testes a fim de calibrar os diversos parâmetros das versões BTPM, BTMV, BTPV, e BTID. Em resumo, foram adotados os mesmos utilizados na versão sem folgas.

### 5.3.1 Resultados dos Algoritmos de Busca Tabu - Com folgas

As Tabelas 5.14, 5.15, 5.16, 5.17 mostram respectivamente os resultados dos algoritmos BTPM, BTMV, BTPV, e BTID com folgas. Os dois últimos contém as médias de todos os resultados (dentro os dez testes com cada instância). As piores porcentagens de melhorias para todas as tabelas de algoritmos tabu (5.7, 5.8, 5.9, 5.10) foram da instância TR0 0,17%, 5,4%, 5,418%, 5,44% de melhoria respectivamente. As maiores melhorias foram da instância NR24 com 39,36%, 39,357%, 39,430%, 39,357% de melhorias respectivamente.

### 5.3.2 Resultados - VNS-VND - Com folgas

A fim de realizar a comparação dos resultados dos algoritmos, foram feitos 10 testes para o VNS-VND, com cada vizinhança, e calculada a média (de acordo com a função 5.1) para cada instância. Esta, por sua vez, foi utilizada para comparação com os algoritmos derivados da Busca Tabu. De forma a possibilitar a comparação, foi utilizada a mesma função objetivo mostrada no capítulo 4, seção 4.3, equação 4.2 e a mesma solução inicial para todos os testes. A Tabela 5.18 apresenta a média dos resultados dos testes de

acordo com a função 5.1.

### 5.3.3 Comparação entre os resultados dos algoritmos BTPM, BTMV, BTPV, BTID e VNS - Com folgas

Dos resultados obtidos foram calculadas a diferença entre os custos das soluções e da solução inicial e na Tabela 5.19 estão apresentados os percentuais de melhoria e as médias de melhorias de cada Algoritmo em relação à solução inicial. Com base nesta tabela, pode-se verificar que o Algoritmo BTMV obteve a melhor média. A fórmula utilizada para cálculo destes percentuais pode ser vista na equação 5.2.

## 5.4 Discussão dos Resultados

Os resultados, como pode ser visto no gráfico 5.1 mostram que para o cenário no qual não são consideradas as folgas, a versão BTMV foi aquela que produziu, em média, os melhores resultados. Em segundo lugar ficou a versão BTID, que conta com a técnica de intensificação e diversificação. Posteriormente são classificadas as soluções das versões BTPV, BTPM e por último o VNS.

Para a situação onde devem ser respeitadas as folgas, seguindo o critério 5/1, com um descanso após cinco dias de trabalho, a classificação das versões implementadas foi ligeiramente modificada. Conforme o gráfico 5.2 a versão mais eficiente continuou sendo o BTMV que busca o melhor vizinho dentre todos da vizinhança da solução corrente. Em segundo lugar ficou o VNS, em terceiro o BTPV seguido pela versão BTID e finalmente o BTPM. Portanto pode-se concluir que para os dois cenários do problema abordado, o Algoritmo mais eficiente foi a Busca Tabu associada ao movimento do Melhor Vizinho, aqui denominado BTMV. Para as demais versões da metaheurística, não é possível identificar uma tendência clara nos resultados. O VNS, por exemplo, passa da quinta e última posição, em relação ao primeiro cenário, para a segunda posição quando considerado o segundo cenário. Já a versão BTPV permanece na terceira posição em ambos os cenários operacionais e o BTPM passa da penúltima para a última posição.

**Tabela 5.13:** Tabela de horas extras e ociosas das soluções iniciais- Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	367.565	136.008	503.573
<b>TR5</b>	58.043	158.013	216.056
<b>TR4</b>	367.837	141.852	509.689
<b>TR7</b>	310.757	28.387	339.144
<b>TR10</b>	30.475	116.942	147.417
<b>NR24</b>	37.139	52.325	89.464
<b>NR104</b>	83.970	110.000	193.970

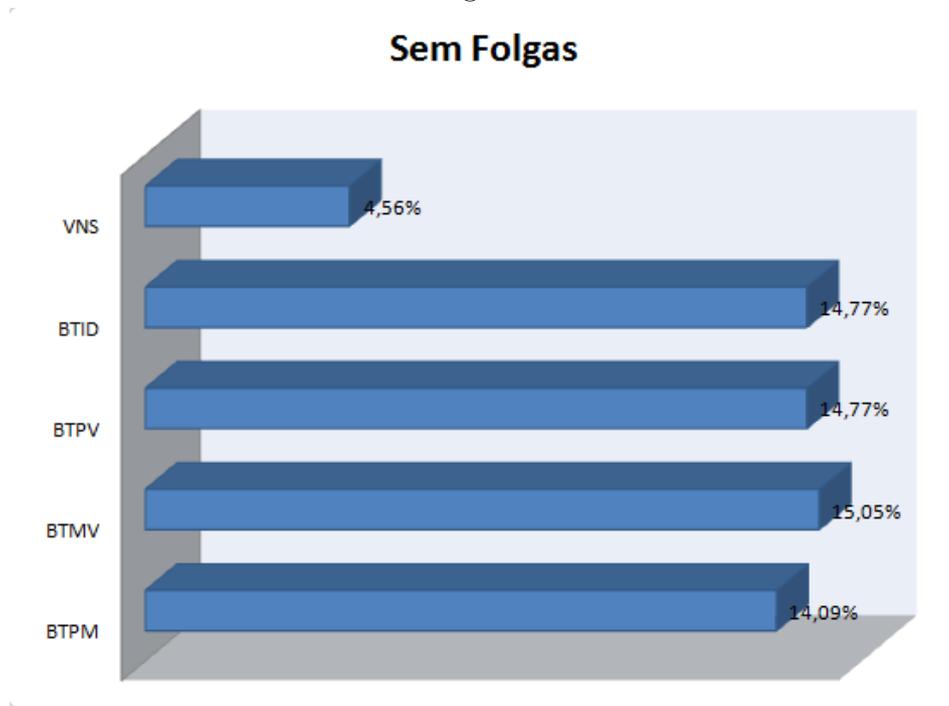
**Tabela 5.14:** Valores em minutos dos resultados do Algoritmo BTM - Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	367.129	135.572	502.701
<b>TR5</b>	57.245	157.215	214.460
<b>TR4</b>	366.116	140.131	506.247
<b>TR7</b>	309.866	27.496	337.362
<b>TR10</b>	29.042	115.509	144.551
<b>NR24</b>	19.534	34.720	54.254
<b>NR104</b>	83.577	109.607	193.184

**Tabela 5.15:** Valores em minutos dos resultados do Algoritmo BTMV - Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	353.969	122.412	476.381
<b>TR5</b>	43.980	143.950	187.930
<b>TR4</b>	332.175	106.190	438.365
<b>TR7</b>	282.930	560	283.490
<b>TR10</b>	18.214	108.227	122.895
<b>NR24</b>	19.534	34.720	54.254
<b>NR104</b>	52.932	78.962	131.894

**Figura 5.1:** Gráfico de melhorias com relação a solução inicial de todos os algoritmos utilizando a versão sem folgas



**Tabela 5.16:** Valores em minutos dos resultados do Algoritmo BTPV - Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	311.015	106.132	476.289
<b>TR5</b>	47.684	126.868	187.926
<b>TR4</b>	303.632	106.876	486.573
<b>TR7</b>	255.679	564	256.243
<b>TR10</b>	17.745	108.227	121.977
<b>NR24</b>	24.406	41.199	54.188
<b>NR104</b>	64.407	86.086	133.180

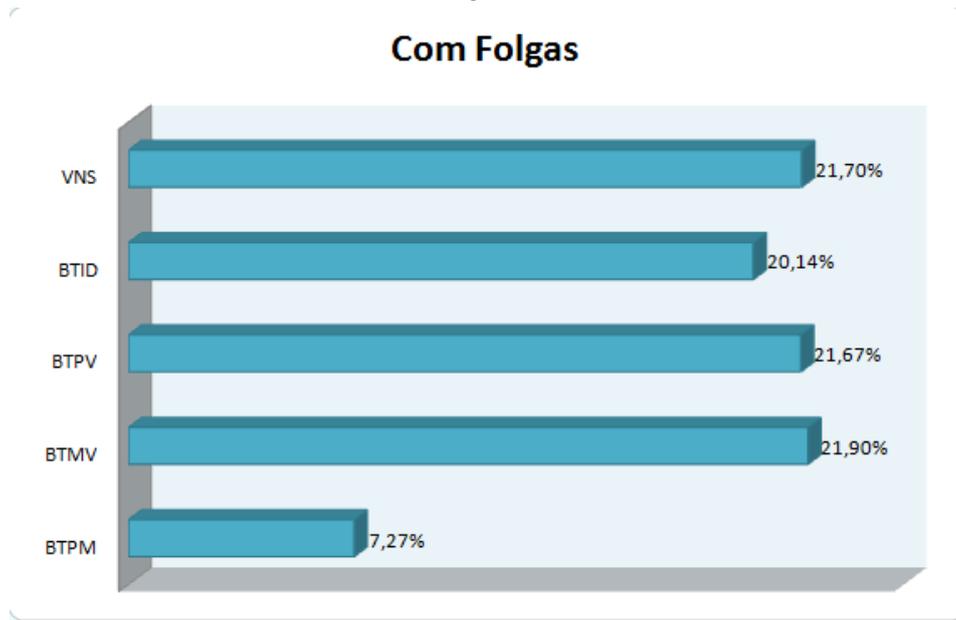
**Tabela 5.17:** Valores em minutos dos resultados do Algoritmo BTID - Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	353.868	122.311	476.179
<b>TR5</b>	43.951	143.921	187.872
<b>TR4</b>	332.175	106.190	438.365
<b>TR7</b>	305.349	22.979	328.328
<b>TR10</b>	18.214	104.681	122.895
<b>NR24</b>	19.534	34.720	54.254
<b>NR104</b>	50.393	76.423	126.816

**Tabela 5.18:** Valores em minutos dos resultados do Algoritmo VNS-VND - Com folgas

<b>Instâncias</b>	<b>Hora extra</b>	<b>Ociosidade</b>	<b>Custo</b>
<b>TR3</b>	348.970	120.054	470.688
<b>TR5</b>	34.967	130.947	164.173
<b>TR4</b>	333.151	118.858	452.816
<b>TR7</b>	296.914	17.234	316.422
<b>TR10</b>	10.658	99.556	120.110
<b>NR24</b>	24.864	40.749	61.421
<b>NR104</b>	46.163	62.254	119.394

**Figura 5.2:** Gráfico de melhorias com relação a solução inicial de todos os algoritmos utilizando a versão com folgas



**Tabela 5.19:** Resultado de melhorias em comparação com a solução inicial - Com folgas - Comparação entre os algoritmos

Instâncias	BTPM	BTMV	BTPV	BTID	VNS
<b>TR3</b>	0,17%	5,400%	5,418%	5,440%	6,530%
<b>TR5</b>	0,74%	13,018%	13,020%	13,045%	24,014%
<b>TR4</b>	0,68%	13,994%	4,535%	13,994%	11,158%
<b>TR7</b>	0,53%	16,410%	24,444%	3,189%	6,700%
<b>TR10</b>	1,94%	16,634%	17,257%	16,634%	18,524%
<b>NR24</b>	39,36%	39,357%	39,430%	39,357%	31,345%
<b>NR104</b>	0,41%	32,003%	31,340%	34,621%	38,447%
Média	<b>7,27%</b>	<b>21,90%</b>	<b>21,67%</b>	<b>20,14%</b>	<b>21,70%</b>



# Capítulo 6

## Conclusões

Este trabalho abordou o Problema de Rodízio de Tripulações (PRT) de uma empresa que opera no sistema de transporte público urbano da cidade de Belo Horizonte (MG). São apresentadas implementações de quatro versões da metaheurística Busca Tabu. A escolha dessa metaheurística se deve ao seu sucesso em outros problemas e pela sua robustez em comparação com outras metaheurísticas. A primeira versão desenvolvida, BTPM mostrou-se em ambos os casos, com e sem folgas, superior quanto ao resultado em comparação às outras versões. Todos os métodos apresentaram redução no número de horas extras, que representa um dos componentes com maior peso na planilha de custos.

Com o objetivo de comparar todos os métodos foram criadas 10 instâncias, além das 3 que foram obtidas da empresa para o cenário sem folgas e 5 para o cenário que considera as folgas das tripulações. A fim de gerar a solução inicial, para cada uma das instâncias foram consideradas todas as combinações de jornadas viáveis. Os algoritmos BTPM, BTMV, BTPV, BTID e VNS foram então aplicadas sobre cada uma das soluções iniciais geradas a fim de melhorar os resultados.

Após a criação da versão BTPM, foi criada a versão BTMV, cuja análise da busca local é feita sobre a vizinhança completa. Foi observada uma melhora significativa nos resultados de BTPM considerando ambos os cenários em relação às folgas. Posteriormente foram implementadas as versões BTPV e BTID, cujos resultados foram melhores que BTPM, mas piores do que BTMV. Para a versão sem folgas, VNS-VND apresentou os piores resultados (menor porcentagem de melhorias com relação à solução inicial), mas para a versão com folgas, ele ficou em segundo lugar.

Para a versão com folgas foi adotado o padrão 5/1, em que o motorista trabalha 5 dias e folga 1. Este padrão foi adotado por cumprir a restrição legal trabalhista de que as tripulações devem folgar pelo menos um domingo a cada período de planejamento. Para isto, foi criado um vetor de dias trabalhados que é utilizado para o cálculo das próximas folgas de cada tripulação.

Este trabalho mostrou-se relevante por resolver um problema real, configurado com todas as restrições operacionais e legais, e por auxiliar na disseminação de técnicas de resolução do PRT ao cenário brasileiro. A maioria das empresas ainda geram escalas de trabalho de forma manual. Foi demonstrado também neste trabalho a eficácia dos métodos heurísticos, em especial a metaheurística de Busca Tabu em tratar problemas reais em tempos de execução computacionais aceitáveis.

Como extensão deste trabalho pode-se propor primeiramente um aprimoramento no vetor de dias trabalhados antes do início do período, vetor este escolhido aleatoriamente. A definição deste vetor é fundamental para diminuir o número total de tripulações no padrão de folgas do tipo 5/1. Deve-se ainda considerar a possibilidade de utilizar também o padrão de folgas variáveis, o que também pode levar a uma redução do número de tripulações necessárias. Por fim, outras metaheurísticas podem ser testadas, como por exemplo os Algoritmos Genéticos, devido à estrutura do problema.

# Apêndice

Exemplo de entrada de um dia útil de uma das instâncias

Linha XXXX - XXX - XXX (Segunda - 00/00/0000)

04:10;12:20;8125;3346

05:00;12:55;9748;3348

05:20;13:04;10385;6444

05:30;13:13;8852;6442

05:40;13:22;9173;4990

05:41;13:40;9767;4999

05:47;09:00;7162;20065

05:57;09:10;9487;4932

06:02;12:30;9423;6445

06:05;09:30;9685;3347

06:13;09:40;10291;20063

06:22;12:42;8077;6443

06:23;10:00;10562;4980

06:28;10:00;10346;4998

11:58;19:00;10631;4998

12:20;19:10;10524;3346 (...)



# Referências Bibliográficas

- Ahuja, R. K., Magnanti, T. L. and Orlin, J. B.: 1993, *Network Flows: Theory, Algorithms and Applications*, Vol. I, Prentice Hall.
- Antp: 1999, O transporte na cidades do século 21, *Associação Nacional dos Transportes Públicos* .  
**URL:** <http://www.antp.org.br>
- Bodin, L. B., Assad, A. and Ball, M.: 1983, The state of the art in the routing and scheduling of vehicles and crews, *Computers and Operations Research* **10**(2), 63–211.
- Bouzada, C. F.: 2003, *Custo do transporte coletivo por ônibus*, Coleção Estado da arte, Editora C/Arte.
- Carraresi, P. and Gallo, G.: 1984, A multi-level bottleneck assignment approach to the bus drivers' rostering problem, *European Journal of Operational Research* **16**(2), 163 – 173.
- Catanas, F. and Paixão, M. P.: 1995, A new approach for the crew rostering problem, in J. R. Daduna, I. Branco and J. M. P. Paixão (eds), *Computer Aided Transit Scheduling*, Vol. 430 of *Lecture Notes in Economics and Mathematical Systems*, Springer Berlin Heidelberg, pp. 267–277.
- Ernst, A., Jlang, H. and Krishnamoorthy, M.: 2004, An annotated bibliography of personnel scheduling, *Annals of Operations Research* **127**, 21–144.
- Ferraz, A. and Torres, I. G.: 2004, *Transporte público urbano*, Rima Editora.
- Garey, M. R. and Johnson, D. S.: 1990, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA.

- Hartog, A., Huisman, D., Abbink, E. and Kroon, L.: 2009, Decision support for crew rostering at ns, *Public Transport* **1**(2), 121–133.  
**URL:** <http://dx.doi.org/10.1007/s12469-009-0009-6>
- Hertz, A., Taillard, E. and Werra, D.: 1995, A tutorial on tabu search, *Proceedings of Giornate di Lavoro AIRO'95 (Enterprise Systems: Management of Technological and Organizational Changes)*, Italy, pp. 13–24.
- Huisman, D., Kroon, L. G., Lentink, R. M. and Vromans, M. J. C. M.: 2005, Operations research in passenger railway transportation.
- Kohl, N. and Karisch, S. E.: 2004, Airline crew rostering: Problem types, modeling, and optimization, *Annals of Operations Research* **127**(1-4), 223–257.  
**URL:** <http://www.springerlink.com/content/10078166515u13u2/fulltext.pdf>
- Leite, P. V. S. and Silva, G. P.: 2012, Resolução do problema de rodízio de tripulações usando um modelo de fluxo em redes. Trabalho de Conclusão de Curso - Graduação em Engenharia de Controle e Automação - Universidade Federal de Ouro Preto.
- Lourenço, H. R., Paixão, J. M. P. and Portugal, R.: 2001, Multiobjective metaheuristics for the bus driver scheduling problem, *Transportation Science* **35**(3), 331–343.
- Mesquita, M., Moz, M., Paias, A., Paixão, J., Pato, M. and Respício, A.: 2011, A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters, *Journal of Scheduling* **14**(4), 319–334.  
**URL:** <http://dx.doi.org/10.1007/s10951-010-0195-8>
- Mladenović, N. and Hansen, P.: 1997, Variable neighborhood search, *Computers and Operations Research* **24**(11), 1097–1100.  
**URL:** [http://dx.doi.org/10.1016/S0305-0548\(97\)00031-2](http://dx.doi.org/10.1016/S0305-0548(97)00031-2)
- Nurmi, K., Kyngas, J. and Post, G.: 2011, Driver rostering for bus transit companies, *Engineering Letters* **19**(2), 125–132.
- Prates, R. F. C. and Silva, G. P.: 2012, Otimização da escala de motoristas de ônibus urbano utilizando a heurística variable neighborhood search, *XXVI ANPET - Congresso Nacional de Pesquisa e Ensino em Transportes ANPET* **1**, 1993–2004.
- Silva, G. and Mayrink, V.: 2010, Otimização da escala mensal de tripulações do sistema de transporte público., *XXIV ANPET - Congresso Nacional de Pesquisa e Ensino em Transportes* **1**, 185–197.

Souza, C., Yunes, T. H. and Moura, A. V.: 2000, Modeling and solving a crew rostering problem with constraint logic programming and integer programming, *Technical Report IC-00-04* .

Souza, M.: 2013, Inteligência computacional para otimização. Accessed: 2013-04-04.

**URL:** <http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional.htm>

Toffolo, T. A., Souza, M. J. F. and Silva, G. P.: 2005, Resolução do problema de rodízio de tripulações de ônibus urbano via simulated annealing e iterated local search, *XIX ANPET - Congresso de pesquisa e Ensino em Transportes* **2**, 657–668.

Townsend, W.: 1988, An approach to bus-crew roster design in london regional transport, *Operational Research Society* **39**(6), 543–550.

Vasconcellos, E.: 2001, *Transporte urbano, espaço e eqüidade: análise das políticas públicas*, Annablume.