



Trabalho de Conclusão de Curso

**Uma Metaheurística Step Counting Hill
Climbing(SCHC) para Resolver o Problema de
Alocação da Tripulação de Ônibus Urbano**

Danilo Santos Souza
danilo.gdc@gmail.com

Orientador:
Balduino Fonseca dos Santos Neto

Maceió, Março de 2015

Danilo Santos Souza

Uma Metaheurística Step Counting Hill Climbing(SCHC) para Resolver o Problema de Alocação da Tripulação de Ônibus Urbano

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Baldoino Fonseca dos Santos Neto

Maceió, Março de 2015

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Baldoino Fonseca dos Santos Neto - Orientador
Instituto de Computação
Universidade Federal de Alagoas

Willy Carvalho Tiengo - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Joilson Batista de Almeida Rêgo - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Resumo

O Problema de Alocação da Tripulação consiste em gerar um conjunto de escalas de trabalho, dado um conjunto de viagens estabelecidas e seus devidos veículos alocados, atribuindo estas escalas aos Motoristas. O objetivo do problema é gerar essas escalas com o menor custo operacional possível, satisfazendo restrições trabalhistas, legislações, normas sindicais e normas internas das empresas. A obtenção de soluções com menor custo operacional está relacionada na redução da quantidade de jornadas, a quantidade de horas extras, o tempo ocioso entre as tarefas a serem realizadas pela tripulação e o número de jornadas do tipo dupla pegada. Nesta pesquisa é proposta a utilização de uma metaheurística Step Counting Hill Climbing(SCHC), para resolver este problema. Todos os testes computacionais realizados utilizaram instâncias geradas baseadas em dados reais de uma empresa do setor de transporte público da cidade de Belo Horizonte/MG.

Abstract

The Crew Scheduling Problem consists to create of a set of work schedules, given a set of established trips and their proper allocated vehicles, attributing these scales to drivers. The objective of the problem is to generate these scales with the lowest possible operating cost, satisfying work laws, trade union rules and internal regulations of the companies. Obtaining solutions with lower operating costs is related in reducing the amount of work schedules, the amount of overtime, the idle time between tasks to be performed by the crew and the number of split duty type journeys. In this research we propose the use of a metaheuristic Step Counting Hill Climbing (SCHC), to solve this problem. All computational tests used instances generated based on real data from a company in the public transport sector in the city of Belo Horizonte / MG.

Agradecimentos

Primeiramente agradeço à *Deus* por me amparar nos momentos difíceis e me dar força para superar as dificuldades.

Aos meus pais, *Gilsa e Denildo*, e a meu irmão *Diego*, pelo incentivo, apoio e afeto.

A minha avó *Carmelita*, pelo exemplo de dedicação e por ser uma mãe pra mim.

Ao meu Tio *Denilson*, por sempre acreditar em mim quando alguns não acreditavam, e por ser um exemplo de pai e homem.

À toda minha *família*, pelo carinho e força.

Agradeço ao meu orientador *Baldoino Fonseca dos Santos Neto* pela oportunidade concedida e pelo apoio no desenvolvimento do trabalho.

Aos meus amigos *Flavio, Bruna, Priscilla, Gabriela, Thayane, Lorrان, Alisson, Pereira, Lucas, Evellyn, Tamer, Rian, Marco, Emilia e Danielle* pela ajuda, pelas longas horas de estudo, amizade, carinho e força, fundamentais para que eu conseguisse chegar até aqui.

A *UFAL* e *IC* pelo apoio recebido no desenvolvimento deste trabalho.

Enfim, agradeço a *todos* que, de alguma forma, acreditaram e torceram por mim, participaram de minha vida e ajudaram na realização deste trabalho.

Danilo Santos Souza

Conteúdo

Lista de Figuras	v
Lista de Tabelas	vi
1 Introdução	1
1.1 Objetivo Geral	2
1.2 Objetivos Específicos	3
1.3 Organização do Trabalho	3
2 Revisão Bibliográfica	4
2.1 Métodos Exatos para Resolver o PPT	5
2.2 Metaheurísticas para Resolver o PPT	6
3 Problema de Programação da Tripulação de Ônibus Urbano	9
3.1 Restrições do Problema	10
4 A Metaheurística (SCHC)	12
4.1 Estruturas para a Representação do Problema	12
4.2 A Metaheurística SCHC	13
4.2.1 Função Objetivo	15
4.2.2 Geração da Solução Inicial	15
4.2.3 Estrutura de Vizinhança Realoca-Troca	15
5 Resultados Obtidos	17
6 Considerações Finais	21
Referências bibliográficas	22

Lista de Figuras

4.1 Movimentos de realocação e troca de uma tarefa da jornada j para a jornada k . 16

Lista de Tabelas

4.1	Estrutura de uma tarefa	12
4.2	Estrutura de uma jornada	13
5.1	Tabela com as informações de cada instância, contendo a quantidade de tarefas a serem realizadas e quantos veículos são utilizados.	17
5.2	Características dos melhores resultados para a instância 01	18
5.3	Características dos melhores resultados para a instância 02	18
5.4	Características dos melhores resultados para a instância 03	18
5.5	Características dos melhores resultados para a instância 04	19
5.6	Características dos melhores resultados para a instância 05	19
5.7	Características dos melhores resultados para a instância 06 com custo 5.000 para jornadas do tipo dupla pegada	19
5.8	Características dos melhores resultados para a instância 07	20

1

Introdução

Empresas de transporte público estão cada vez mais buscando utilizar ferramentas computacionais para diminuir o custo de alocação de suas tripulações na frota operacional, ou seja, definir quais serão os motoristas e cobradores que realizarão as viagens diárias programadas. Neste trabalho é abordado o Problema da Escala de Motoristas de Ônibus Urbano, também conhecido na literatura como Problema de Programação da Tripulação (PPT), que consiste em gerar um conjunto de jornadas de trabalho para as tripulações que conduzirão a frota em operação de uma empresa de transporte público. Além disso, estas jornadas deve obedecer as restrições impostas pelas empresas, as leis trabalhistas e os acordos contidos nas convenções coletivas de trabalho.

Este problema vem sendo estudado desde a década de 60, [7] como trabalho pioneiro, e devido as constantes mudanças no sistema de transporte publico varias pesquisas são realizadas para melhor resolver o problema.

A importância do PPT vem do fato que o custo de mão de obra operacional de uma empresa de transporte público compõe grande parte dos gastos da mesma [3]. A redução deste custo é de grande interesse das empresas, pois essa economia pode ser reinvestida em outras áreas dentro da empresa, como qualificação dos seus funcionários, manutenção da frota, entre outros. Desta forma, a qualidade do serviço pode ser melhorada, assim como o retorno à empresa.

Devido às restrições operacionais vigentes nas empresas, às leis trabalhistas e aos acordos contidos nas convenções coletivas de trabalho, o PPT torna-se de grande complexidade. Este pode ser modelado como um problema de particionamento ou de recobrimento com variáveis binárias, o que o torna NP-hard, para o qual não existe algoritmo de complexidade polinomial que obtenha a solução ótima [10].

O PPT tem como dados de entrada a Programação dos Veículos, que corresponde ao sequenciamento das viagens a serem realizadas por cada ônibus em operação de uma dada

empresa. Cada viagem a ser realizada apresenta uma hora de início e de fim, além do seu ponto inicial e o seu ponto final. O conjunto de todas as viagens realizadas por um veículo constitui o seu bloco de viagens. Desta forma é possível prever o horário de partida do veículo da garagem e o seu retorno ao final do dia, assim como o detalhamento de todas as viagens e eventuais pausas realizadas pelo veículo ao longo da operação. As viagens de cada veículo são agrupadas em tarefas, que são uma sequência de viagens consecutivas que deve ser executada por um único condutor, uma vez que entre estas viagens não existe tempo suficiente para que ocorra a troca da tripulação. Cada jornada corresponde a um conjunto de tarefas que será realizado por uma tripulação ao longo de um dia. As jornadas tem que respeitar as restrições impostas pela empresa e pelas leis trabalhistas e convenções coletivas.

No problema abordado estão presentes dois tipos de jornadas. A dupla pegada é uma jornada caracterizada por um conjunto de tarefas que tem um intervalo entre duas tarefas maior ou igual a um valor estipulado, no caso de duas horas. Este tipo de jornada é utilizado para a condução dos veículos que são operam durante os horários de pico, que ocorrem principalmente no início da manhã e no final da tarde. O intervalo entre os dois pedaços de jornada não é contabilizado como hora trabalhada.

Em uma jornada do tipo pegada simples os intervalos entre as tarefas são menores do que o valor estipulado para a dupla pegada. Independentemente do tipo de jornada, as horas extras correspondem ao tempo trabalhado além do tempo normal, que no caso é de seis horas e quarenta minutos. A resolução do PPT tem como objetivo atribuir as tarefas às tripulações de modo que cada tarefa seja realizada por uma tripulação, as restrições do problema sejam atendidas e que o custo total composto pelo número de tripulações e a quantidade de horas extras seja minimizado.

Devido à complexidade deste problema, a utilização de métodos exatos fica restrita, impossibilitando a resolução de problemas reais de grandes dimensões. Os métodos heurísticos são amplamente utilizados para resolver o problema [8].

Neste trabalho foi utilizada uma abordagem heurística utilizando o método *Step Counting Hill Climbing* (SCHC) que consiste em gerar jornadas de trabalho a partir das viagens a serem realizadas.

1.1 Objetivo Geral

Este trabalho tem como objetivo resolver o Problema da Programação de Tripulação de Ônibus Urbano de modo a reduzir o máximo possível os custos operacionais na atribuição das viagens às tripulações, utilizando uma técnica. Assim foi implementado uma metaheurística, para avaliar as vantagens e desvantagens desta abordagem na resolução de problemas reais.

1.2 Objetivos Específicos

- Resolver o PPT utilizando a metaheurística *Step Counting Hill Climbing* – SCHC;
- Avaliar os resultados obtidos com resultados da literatura e das soluções apresentadas pela empresa.

1.3 Organização do Trabalho

Os demais capítulos deste trabalho estão organizados da seguinte forma: No Capítulo 2 uma Revisão Bibliográfica sobre problema é apresentada. O Capítulo 3 define o problema da Programação da Tripulação. No Capítulo 4 aborda a metaheurística proposta para resolver o problema trabalhado. No Capítulo 5 são apresentados os testes computacionais e suas considerações para o problema. O Capítulo 6 contém as considerações finais e os trabalhos futuros.

2

Revisão Bibliográfica

 PPT é um problema clássico, sobre o qual existem diversos trabalhos que exploram diferentes técnicas de programação linear inteira para resolvê-lo. Entre estas técnicas são destacadas o *Branch-and-Bound* [11, 24], *Branch-and-Price* [6, 1] e *Branch-and-Cut* [13]. Por outro lado, diversas metaheurísticas foram implementadas para resolver problemas de grande porte. Dentre essas pode-se destacar: Algoritmos Genéticos, GRASP, Busca Tabu, Simulated Annealing, entre outros [21, 22, 15, 25]. Com o emprego destas meta-heurísticas foi possível obter resultados muito satisfatórios para o problema, embora a otimalidade das soluções não sejam garantidas.

O trabalho de [9] apresenta uma série de problemas relacionados à alocação de pessoal em diversas áreas, incluindo motoristas e cobradores de sistemas de transportes. Neste contexto, os autores descrevem problemas de alocação de funcionários em seus diferentes modais de transportes, ou seja, no aeroviário, ferroviário, transporte urbano e no transporte interurbano. No trabalho é apresentada uma revisão sobre as principais pesquisas tanto para o problema de programação da tripulação, como para o problema do rodízio das tripulações. No trabalho é mencionado também que a complexidade dos problemas dependem do tipo de transporte, da categoria da tripulação, dos tipos de frotas, das regras e regulamentações, da regularidade das viagens e dos custos a serem considerados. Segundo [9], uma das abordagens mais utilizadas para resolver os problemas de das tripulações é aquela que o decompõe nas seguintes etapas: geração das jornadas, otimização das jornadas e definição do rodízio da tripulação. Desta forma, a complexidade do problema é reduzida, permitindo sua resolução de forma satisfatória.

Algumas destas técnicas de otimização são utilizadas em pacotes comerciais voltadas para a área de programação de veículos e tripulação. Uma metaheurística do tipo *run-cutting* foi utilizada para resolver o problema no pacote RUCUS [27, 16], desenvolvido nos Estados Unidos da América a partir da década de 70, e um algoritmo desenvolvido por [17]

foi utilizado no sistema TRACS, o qual simulava o processo manual de alocação da tripulação. A formulação de recobrimento de conjuntos foi usada no pacote IMPACS do sistema BUSMAN e no pacote Crew-opt, que posteriormente foi utilizado na versão inicial do HASTUS que substituiu o TRACS [28]. A empresa brasileira Wplex, desenvolvedora de sistemas no âmbito de transporte, desenvolveu um sistema WplexON [12] para tratar problemas de alocação de tripulação. Neste trabalho a empresa utiliza a abordagem de *Branch-and-Price* para solucionar o problema modelado como particionamento de conjuntos, sendo que o conjunto de *Jornada de Trabalho* é gerada a partir do movimento de inserção de tarefas. Estas soluções obtidas pela Wplex são utilizadas em empresas no estado de Santa Catarina.

Nas subseções a seguir são apresentados alguns trabalhos que utilizam abordagens Exatas e Heurísticas.

2.1 Métodos Exatos para Resolver o PPT

Os métodos exatos se mostram mais eficientes quando utilizados para resolução de problemas de otimização combinatória quando os problemas são de pequena dimensão ou não são NP-Completo. No entanto, estes métodos podem ser ineficientes para problemas de grande dimensão, uma vez que o tempo de execução aumenta exponencialmente à medida que as instâncias aumentam. Ainda assim, existem várias abordagens exatas descritas na literatura para resolver o problema. Entre elas podemos citar a Programação Dinâmica, a Programação Linear e Inteira, Relaxação Lagrangeana, Programação por Restrições, entre outros. Muitas destas técnicas são flexíveis e independentes de domínio, e podem ser aplicadas à maioria dos problemas práticos [26].

Em [6] é proposta uma abordagem de geração de colunas para resolver o Problema de Programação da Tripulação. Nesta abordagem o problema é dividido em dois: um problema de cobertura de conjunto e um problema de caminho mínimo com limitação de recurso. O problema de cobertura de conjuntos escolhe uma escala de trabalho viável já conhecida. O problema de caminho mínimo é usado para propor novas escalas de trabalho para melhorar a solução obtida pelo problema de cobertura de conjuntos. Para o problema de cobertura de conjuntos, o método de geração de colunas é representado por linhas e colunas, onde as colunas representam uma escala viável e as linhas as tarefas a serem realizadas. O modelo deve selecionar as melhores escalas de modo que todas as tarefas de um dia de trabalho sejam cobertas. Porém as novas escalas são geradas usando um algoritmo de caminho mínimo com restrição de recursos. Cada caminho viável, da origem ao consumidor, representa uma jornada de trabalho viável, ou seja, uma nova coluna para o problema de cobertura de conjuntos. O método proposto apresenta resultados satisfatório para resolver problemas considerados pequenos, compostos de 167 e 235 tarefas.

[11] apresentam um sistema baseado em programação linear inteira, como melhoria do

sistema TRACS II desenvolvido pela Universidade de Leeds, Inglaterra. Diferente da abordagem utilizada no TRACS II, que utilizava duas funções objetivos para calcular as penalidades impostas na solução, o modelo proposto por [11] utiliza uma função objetivo composta, e o método de geração de colunas para obter a sua solução final. Os testes computacionais mostram que o modelo proposto consegue resultados melhores do que o modelo utilizado pelo sistema TRACS II, e obteve uma redução média de 41% no tempo de processamento para o mesmo conjunto de dados.

[20] utilizam uma abordagem exata de Geração de Colunas combinada com a metaheurística populacional Algoritmos Genéticos para resolver o PPT. Na geração de colunas, o problema é subdividido em dois: problema mestre e subproblema. O problema mestre seleciona as jornadas que serão realizadas pela tripulação em um conjunto de jornadas viáveis, de modo a conseguir cobrir todas as viagens necessárias. Já o subproblema é responsável por gerar novas jornadas (colunas). Estas jornadas são adicionadas ao problema mestre se possibilitarem a melhoria da solução. Para a resolução do subproblema é utilizado um Algoritmo Genético. Os resultados obtidos mostram que a utilização do Algoritmo Genético, em comparação com outras metaheurísticas, destaca-se pelo fato da possibilidade de geração de um conjunto maior de jornadas viáveis para a inclusão no problema mestre e consequentemente possibilita reduzir o tempo de processamento quando lida com grandes instâncias.

2.2 Metaheurísticas para Resolver o PPT

[15] utilizam duas metaheurísticas multiobjetivos para solucionar o PPT, a Busca Tabu e um Algoritmo Genético. Na abordagem apresentada, são consideradas duas funções objetivo, uma com os custos operacionais do problema ao que se refere a quantidade de tripulantes e tempo operacional, e custos por quebra de restrições operacionais, por exemplo exceder o máximo de trocas de veículos. Na Busca Tabu, os autores consideram duas listas tabu, uma de inserção e outra de remoção de jornadas. A primeira contém a lista de jornadas que foram removidas e não podem ser inseridas e a segunda lista contém as jornadas incluídas que não podem ser removidas. A Busca Tabu é realizada para cada função objetivo e posteriormente é aplicada com a função de soma ponderada das duas funções objetivos. Um processo para otimizar a solução é proposto, onde a Busca Tabu é realizada com várias iterações com objetivo de encontrar soluções viáveis, utilizando somente de inserção de tarefas, e a metaheurística GRASP é utilizada para selecionar as melhores jornadas geradas no processo de Busca Tabu. No algoritmo genético, a criação de novas populações se baseia na escolha aleatória de uma das funções objetivo para conseguir uma diversificação maior da população. A utilização de um método de cruzamento entre duas populações é proposto, utilizando GRASP para tentar resolver este cruzamento como um subproblema e encontrar uma população dita perfeita. Os teste foram comparados com dados utilizados em uma em-

presa de Portugal e um método de programação linear proposto por [2].

[14] propõe um estudo comparativo entre as metaheurísticas Busca Tabu e *Iterated Local Search* (ILS) e uma metodologia exata de programação matemática. Para a utilização das metaheurísticas foi necessário a geração de uma solução inicial, onde esta foi gerada de forma sequencial. A cada iteração, uma tarefa é escolhida sequencialmente e alocada à jornada existente, desde que a jornada permaneça viável. Como estrutura de vizinhança foram utilizados os movimentos de troca e realocação de tarefas, os mesmos utilizados no presente trabalho. Na Busca Tabu, a lista tabu é definida como uma fila circular, onde os vizinhos mais antigos são eliminados à medida em que os novos são adicionados e a fila está cheia. A metaheurística ILS é combinada com a metaheurística *Variable Neighborhood Descent* (VND), onde o VND é utilizado como método de busca local do ILS. Na utilização do ILS um processo de perturbação é apresentado, onde sucessivos movimentos aleatórios baseados na estrutura de vizinhança são empregados. Como critério de aceitação, uma nova solução é aceita se ela for melhor do que a melhor solução encontrada. Para as metaheurísticas a função de avaliação proposta leva em consideração soluções viáveis e inviáveis. As soluções inviáveis são penalizadas de forma a privilegiar a eliminação destas da solução final. A metodologia exata proposta utiliza a abordagem de geração de colunas onde as colunas são geradas a partir de Enumeração Exaustiva. No modelo exato só são aceitas soluções que atendam todas as restrições, ou seja soluções viáveis. Os resultados apresentados mostram que os métodos propostos obtêm soluções muito próximas e provou-se otimalidade em 93,75% dos casos de teste considerados.

No trabalho de [18] é apresentada a utilização da metaheurística VNS (*Variable Neighborhood Search*) combinada com dois métodos de busca distintos, o método clássico VND (*Variable Neighborhood Descent*) e o método de busca em vizinhança de grande porte VLNS (*Very Large-scale Neighborhood Search*). A metaheurística VNS é iniciada com uma solução factível (solução corrente) e um conjunto de estruturas de vizinhança é definido sequencialmente. Com a primeira estrutura de vizinhança, é gerado um vizinho da solução corrente. Em seguida é realizada uma busca local a partir deste vizinho. Se ao final da busca local for encontrado um valor melhor do que o da solução corrente então a solução é atualizada e o procedimento se repete considerando a estrutura de vizinhança inicial. Caso o valor da solução encontrada na busca local seja pior do que o valor atual, um novo vizinho é gerado considerando a próxima estrutura de vizinhança, e o procedimento de busca local é novamente realizado. O procedimento se repete até que a condição de parada seja atingida. Como procedimento de busca local foram utilizados o VND e o VLNS. Ambas as implementações foram testadas considerando que as tripulações podem realizar no máximo uma troca e duas trocas de veículos durante a jornada. Os custos para jornadas do tipo dupla pegada também foram variados na realização dos testes, assumindo os valores de 500 e 6000. Ambos os métodos utilizados obtiveram melhores resultados quando comparados com os custos adotados pela empresa da cidade de Belo Horizonte-MG que forneceu os dados. Parte des-

tes resultados serão utilizados como base de comparação para os métodos propostos neste trabalho.

3

Problema de Programação da Tripulação de Ônibus Urbano

Neste trabalho é abordado o Problema de Programação das Tripulações de Ônibus Urbano (PPT), também conhecido na literatura como *Crew Scheduling Problem*. O PPT tem como dados de entrada a solução do Problema de Programação dos Veículos, que corresponde ao sequenciamento das viagens a serem realizadas por cada ônibus em operação, de uma empresa, num determinado dia da semana. Cada viagem a ser realizada apresenta um horário de início e de fim, além do seu ponto inicial e ponto final. O conjunto de todas as viagens realizadas por um veículo constitui seu *bloco de viagens*. Desta forma é possível prever o horário de partida do veículo da garagem e o seu retorno ao final do dia, assim como o detalhamento de todas as viagens e eventuais pausas realizadas pelo veículo ao longo da operação. As viagens de cada veículo são agrupadas em *tarefas*. Uma tarefa é a sequência de viagens consecutivas que deve ser executada por um único condutor, uma vez que entre estas viagens não existe tempo suficiente para que ocorra a troca da tripulação. Cada *jornada* corresponde a um conjunto de tarefas que serão realizadas por uma tripulação ao longo de um dia. As jornadas são formadas a partir da atribuição das tarefas, a composição destas jornadas devem respeitar o pontos de trocas, a utilização dos veículos, o tempo máximo de trabalho, entre outros. Desta forma, podemos identificar uma jornada como uma tripulação e vice-versa, pois cada tripulante receberá uma jornada a ser realizada durante um dia útil de trabalho. As jornadas também devem respeitar as restrições impostas pela empresa, pelas leis trabalhistas e convenções coletivas.

No problema abordado, são considerados dois tipos de jornadas. A *dupla pegada* é uma jornada caracterizada por um conjunto de tarefas que têm um intervalo, entre duas tarefas, maior ou igual a um valor estipulado, no caso de duas horas. Este tipo de jornada é utilizado para a condução dos veículos que só operam durante os horários de pico, o que ocorre

principalmente no início da manhã e no final da tarde. O intervalo entre os dois *pedaços de jornada* não é contabilizado como hora trabalhada. Em uma jornada do tipo *pegada simples* os intervalos entre as tarefas são menores do que o valor estipulado para a dupla pegada. Independentemente do tipo de jornada, as horas extras correspondem ao tempo trabalhado além do tempo normal, que no caso é de seis horas e quarenta minutos. Por outro lado, se uma jornada durar menos do que o tempo normal de trabalho, então a diferença entre o tempo normal de trabalho e a duração da jornada é o *tempo ocioso* ou a *ociosidade* da jornada.

A resolução do PPT tem como objetivo atribuir as tarefas às tripulações de modo que cada tarefa seja realizada por uma única tripulação, as restrições do problema sejam atendidas e que o custo total composto pelo número de tripulações e a quantidade de horas extras seja minimizado.

Além das restrições mencionadas, as jornadas devem obedecer as restrições impostas pelas empresas, as leis trabalhistas e os acordos contidos nas convenções coletivas de trabalho. Entre essas regras podem se citadas: a carga horária máxima de trabalho de cada tripulante, o tempo mínimo de descanso de cada tripulação, o tempo mínimo para a realização da troca da tripulação, a possibilidade de troca de veículos durante a jornada, entre outras.

Nas subseções seguintes são apresentadas as restrições consideradas neste trabalho e uma abordagem matemática clássica de particionamento de conjuntos normalmente utilizada para resolver o problema.

3.1 Restrições do Problema

Para a construção das jornadas de trabalho foram levadas em consideração as seguintes restrições operacionais e trabalhistas:

- Sobreposição: Dada duas tarefas, i e j realizadas consecutivamente por uma mesma tripulação, então o horário de início do tarefa j deve ser posterior ao final da tarefa i ;
- Hora Extra: A duração de uma jornada deve conter no máximo 2 horas a mais da duração de trabalho regular, sendo este tempo de excesso contabilizado como hora extra;
- Intervalo de Descanso: Cada tripulação tem direito a no mínimo 30 minutos particionado ou não de folga (descanso, almoço, etc.). Esse tempo não é aplicado quando a jornada for do tipo dupla pegada;
- Troca de Terminal: Dada duas tarefas, i e j realizadas consecutivamente por uma tripulação, então o ponto final da tarefa i deve ser igual ao ponto inicial da tarefa j . Para as jornadas do tipo dupla pegada esta troca é permitida se o intervalo de tempo entre o término da tarefa i e o início da tarefa j for superior a 2 horas;

- Intervalo de Descanso Diário: Considerando que uma jornada pode ser executada pela mesma tripulação em dias consecutivos, então o intervalo entre o fim da jornada e o seu início no dia seguinte deve ser superior a 11 horas.
- Troca de Veículo: Em uma jornada de trabalho, dependendo das regras empresariais, pode ter como restrição um número limitado de troca de veículos;
- Minimização de Hora Extra: O tempo excedente da hora de trabalho normal será contabilizado em horas extras e o mesmo deve ser minimizado;
- Minimização do Número de Duplas Pegadas: Deve ser minimizado o número de jornadas do tipo dupla pegada o tanto quanto possível;
- Número de Jornadas: A quantidade de jornadas de trabalho deve ser mínimo, cobrindo todas as tarefas as diárias;
- Tempo Ocioso: Deve ser minimizado, tanto quanto possível, o tempo total ocioso das tripulações;



A Metaheurística (SCHC)

Neste capítulo é apresentado o método utilizado para a resolução do PPT. A resolução do PPT consiste em encontrar um conjunto de jornadas de tal forma que todas as tarefas sejam realizadas com o menor custo possível e que sejam atendidas todas as restrições consideradas para o problema (3.1). A seguir são descritas as estruturas utilizadas para representar os principais elementos que compõem uma solução do PPT. Em seguida é apresentada a Metaheurística (SCHC).

4.1 Estruturas para a Representação do Problema

Tarefa	
Identificador	
Horário de Início	Horário de Fim
Terminal de Início	Terminal de Fim
Número do Veículo	

Tabela 4.1: Estrutura de uma tarefa

A Tabela 4.1 apresenta os dados de cada tarefa a ser alocada. O campo *Identificador* recebe um inteiro que identifica unicamente cada tarefa. Os campos *Horário de Início* e *Horário de Fim* correspondem ao instante estimado em que a tarefa se inicia e se encerra, respectivamente. No campo *Terminal de Início* é atribuído um número inteiro para identificar o terminal onde a tarefa se inicia, e no campo *Terminal de Fim* um número inteiro para identificar o terminal onde a tarefa se encerra. O campo *Veículo* recebe um número inteiro que identifica qual veículo realiza a tarefa. Estes campos são as características relevantes de uma tarefa para a resolução do PPT.

Jornada	
Identificador	
Horário de Início	Horário de Fim
Horas Extras	Horas Ociosas
Dupla Pegada	Troca de Veículos
Lista das Tarefas	

Tabela 4.2: Estrutura de uma jornada

Na Tabela 4.2 é apresentada a estrutura de uma jornada, sendo que para cada jornada é atribuído um número inteiro único ao campo *Identificador*. Os campos *Horário de Início* e *Horário de Fim* recebem o instante estimado em que a jornada se inicia e instante que ele termina, respectivamente. No campo *Horas Ociosas* é atribuída a soma do tempo de *Ociosidade Interna* e *Ociosidade Externa*. Onde a ociosidade interna é a soma das ociosidade entre duas tarefas consecutivas. A ociosidade externa contém a ociosidade que ocorre quando uma jornada tem duração inferior ao tempo normal de trabalho. Portanto, a ociosidade externa é igual ao tempo normal de trabalho menos a duração da jornada. No campo *Hora Extra* é atribuído o valor do tempo de trabalho superior ao tempo normal de trabalho, ou seja, a duração da jornada menos o tempo normal de trabalho, quando houver. No campo *Dupla Pegada* é indicado se a jornada é do tipo dupla pegada ou não. A *Troca de Veículos* corresponde ao número de vezes que a tripulação deve trocar de veículo durante a jornada. O conjunto das tarefas a serem realizadas na jornada é armazenado na *Lista de Tarefas*.

4.2 A Metaheurística SCHC

A metaheurística *Step Counting Hill Climbing-SCHC*, proposta por [5], trata-se de uma adequação do método *Hill Climbing* Clássico [19]. Segundo [4], essa metaheurística é semelhante ao *Hill Climbing* operando com um parâmetro de controle (upper bound) *Custo Superior* b_c . Este custo representa o pior custo aceitável, ou seja a cada iteração do algoritmo será aceita qualquer solução candidata com o custo menor que o b_c e rejeita as soluções cujo o custo é mais elevado (pior) do que b_c . O algoritmo do SHCH é apresentado abaixo.

Algorithm 1: Implementação do SCHC para um problema de minimização

Entrada: Solução inicial s e o parâmetro c .
Saída: Melhor solução s^* encontrada.

```

 $i \leftarrow 0;$ 
 $b_c \leftarrow f(s);$ 
 $s^* \leftarrow s;$ 
enquanto Condição de parada não atendida faça
  |  $i \leftarrow i + 1;$ 
  | Gere um candidato  $s'$  na vizinhança de  $s$ ;
  | se  $f(s') < b_c$  ou  $f(s') \leq f(s^*)$  então
  | |  $s \leftarrow s';$ 
  | | se  $f(s') \leq f(s^*)$  então
  | | |  $s^* \leftarrow s';$ 
  | | fim
  | fim
  | se  $i \geq c$  então
  | |  $b_c \leftarrow f(s^*);$ 
  | |  $i \leftarrow 0;$ 
  | fim
fim
retorna  $s^*;$ 

```

O algoritmo SCHC deve ser inicializado a partir de uma solução inicial s e um parâmetro c , que define a quantidade máxima de soluções encontradas na busca para atualização do custo superior b_c . Um contador i é utilizado para realizar a contagem destas soluções, e ao atingir o valor de c este contador é reiniciado e o custo b_c é atualizado com o valor do custo da melhor solução encontrada $f(s^*)$. A função f calcula o custo da solução desejada, esta é apresentada na próxima seção. O valor inicial de b_c é definido pelo custo da solução inicial, essa considerada também a melhor solução encontrada s^* .

O processo de busca se encerra quando uma condição de parada for atendida, neste trabalho a condição adotada foi o tempo de processamento. A cada iteração uma nova solução s' é gerada a partir da solução s , esta atualizada sempre que seu custo é menor que b_c ou, menor ou igual a custo de s^* (melhor solução encontrada). Para geração de uma nova solução o método de realoca troca é utilizado, este é apresentado na seção 4.2.3. Caso o custo de s' seja menor que o custo da melhor solução atual encontrada, s^* é atualizada com s' . Ao final do algoritmo a saída desejada será a solução s^* que conterà a solução com menor custo encontrada.

Em [5] são apresentadas as seguinte propriedades deste algoritmo:

- O algoritmo requer apenas um parâmetro c ("limite do contador") a ser criada.
- Quando $c \leq 1$ a busca se transforma em Hill Climbing Guloso. Assim, o SCHC apresenta suas próprias propriedades com os valores de $c = 1$. Com o aumento de c o tempo

de pesquisa é também aumentada, o que (para muitos problemas) leva a melhores resultados finais.

Como pode ser visto no Algoritmo 1, o SCHC é de fácil implementação, porém algumas fases do algoritmo são de fundamental importância para o bom desempenho do mesmo. Estas fases são apresentadas nas próximas seções.

4.2.1 Função Objetivo

A função objetivo utilizada na metaheurística SCHC é apresentada abaixo.

$$f_{o_{schc}} = \sum_{j \in J} (\text{Custo}_J * y_j + \text{Custo}_{OC} * (\text{OC}_{ext_j} + \text{OC}_{int_j}) + \text{Custo}_{HE} * \text{Qnt}_{HE_j} + \text{Custo}_{DP} * \text{Dupla}_j) \quad (4.1)$$

Na expressão (4.1) o Custo_J se refere ao custo fixo de cada jornada, Custo_{OC} é o custo de cada hora ociosa, Custo_{HE} é o custo de cada hora extra e Custo_{DP} é o custo associado a cada jornada do tipo dupla pegada. Desta forma são minimizados os custos fixos, as horas extras, as horas ociosas e as duplas pegadas.

4.2.2 Geração da Solução Inicial

Uma fase importante da metaheurística SCHC é a geração da solução inicial. Para se obter uma solução inicial, foi utilizado um método construtivo guloso, onde o critério adotado é a ordem das tarefas, estas ordenadas a partir do tempo de início. Os dados de entrada, ou seja, as tarefas a serem programadas, são armazenadas na ordem crescente de seus horários de início, e em caso de empate, pelo horário de fim. A construção da solução é iniciada com uma jornada vazia, dita jornada corrente. A primeira tarefa k ainda não alocada é inserida nessa jornada. A partir de então, enquanto for possível, são inseridas as próximas tarefas ainda não alocadas, que pertencem ao mesmo veículo da tarefa k e que não geram inviabilidade na jornada corrente. Quando não for possível inserir qualquer tarefa na jornada corrente, uma nova jornada vazia é inicializada e o processo se repete até que todas as tarefas tenham sido alocadas.

4.2.3 Estrutura de Vizinhança Realoca-Troca

Os dois tipos de movimentos que caracterizam a estrutura de vizinhança adotada foram a realocação ou a troca de uma tarefa entre duas jornadas, sem gerar inviabilidade. Estes movimentos são realizados para encontrar um vizinho de uma solução corrente. Exemplificando, considere duas jornadas i e j , escolhidas aleatoriamente. Então é sorteada uma tarefa

a ser retirada da jornada i e introduzida na jornada j . Logo, pode ocorrer uma das seguintes situações:

- A tarefa retirada de i pode ser introduzida em j sem a necessidade de remover qualquer tarefa da jornada j . Neste caso é realizado um movimento de realocação, e a nova solução será avaliada.
- A introdução da tarefa em j exige a retirada de uma ou mais tarefas desta jornada. Neste caso, se a(s) tarefa(s) removida(s) de j puder(em) ser inserida(s) na jornada i , sem haver qualquer sobreposição com as tarefas remanescentes em i , então o movimento é aceito, caso contrário ele é descartado. Este é um movimento de troca.

Em ambos os casos, as modificações são aceitas se e somente se as jornadas resultantes respeitarem todas as restrições do problema. Uma representação gráfica do movimento é apresentada na figura 4.1.

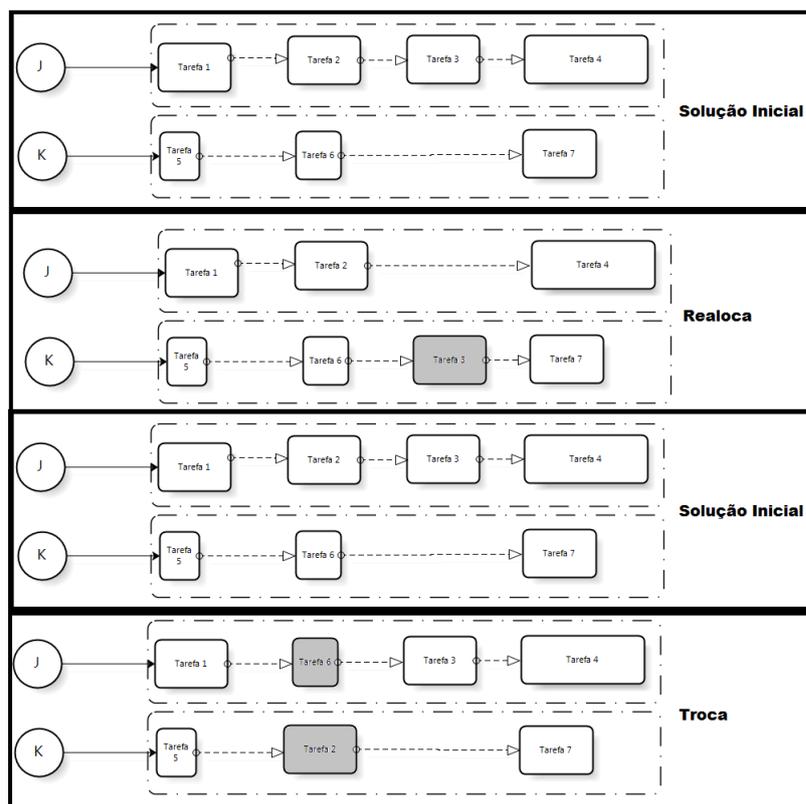


Figura 4.1: Movimentos de realocação e troca de uma tarefa da jornada j para a jornada k

5

Resultados Obtidos

A metaheurística SCHC foi testada em um conjunto de sete instâncias referentes a uma semana de operação de uma empresa que atua na cidade de Belo Horizonte-MG. A metaheurística foi implementada na linguagem C/C++. Todos os testes foram executados em um microcomputador com processador Intel(R) Core(TM) i7-2600, com clock de 3.40GHz, 8 GB de memória RAM, sob plataforma Windows Seven Professional.

Na Tabela 5.1 são apresentadas as informações de cada instância utilizada para a realização dos testes.

Instancia	Total de tarefas	Total de veículos
1 - Segunda	705	61
2 - Terça	674	58
3 - Quarta	814	68
4 - Quinta	872	76
5 - Sexta	787	71
6 - Sábado	644	56
7 - Domingo	345	34

Tabela 5.1: Tabela com as informações de cada instância, contendo a quantidade de tarefas a serem realizadas e quantos veículos são utilizados.

Nos experimentos, foram adotados os seguintes valores para os respectivos parâmetros:

- Custo por jornada de trabalho: 10.000
- Custo da dupla pegada: 5.000
- Custo da hora extra expressa em minutos: 4
- Máximo de trocas de veículos por jornada: 1
- Contador limite c: 1.000

Para cada instância, foram realizadas 10 execuções da metaheurística tendo como condição de parada o tempo em execução limitado a 60 minutos.

As Tabelas apresentadas a seguir mostram uma comparação dos resultados obtidos pela SCHC e aqueles produzidos pelas metaheurísticas VNS clássico e VNS combinado com VLNS, do trabalho de [23]. Também são apresentados o dados utilizados pela empresa.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.323.052	117	96:03	26
Empresa	1.390.804	134	86:41	6
VNS Clássico	1.368.520	120	98:00	29
VNS-VLNS	1.270.628	120	65:07	11

Tabela 5.2: Características dos melhores resultados para a instância 01

Analisando os dados apresentados nas Tabelas 5.2 pode-se concluir que a Metaheurística SCHC produziu uma solução aceiável. Porém, em sua melhor solução, há um acréscimo no custo final, quando comparado com a melhor solução encontrada pela metaheurística VNS-VLNS. A SCHC reduz o custo total, o número de jornadas e o número de duplas pegadas quando comparado com o VNS Clássico.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.322.420	113	94:02	25
Empresa	1.335.620	130	86:41	3
VNS Clássico	1.318.200	114	96:40	31
VNS-VLNS	1.213.176	114	75:44	11

Tabela 5.3: Características dos melhores resultados para a instância 02

Os resultados para a instância 02 são apresentados na Tabela 5.3. A metaheurística SCHC encontrou uma solução melhor do que a solução encontrada pelo VNS clássico, reduzindo tanto o número de duplas pegadas quanto a quantidade de horas extras. Por outro lado, o VNS-VLNS apresenta a melhor solução entre os métodos. A solução encontrada pela metaheurística LAHC possui 25 duplas pegada das 113 jornadas, se enquadrando nas métricas aceitas pelas empresas para este tipo de jornada.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.518.460	132	118:31	28
Empresa	1.540.460	149	106:05	5
VNS Clássico	1.527.904	135	116:16	30
VNS VLNS	1.471.176	140	67:24	11

Tabela 5.4: Características dos melhores resultados para a instância 03

A Tabela 5.4 apresenta os resultados obtidos para a terceira instância, uma quarta-feira. Os custos de Fo são reduzidos pelo SCHC quando comparados com aos dados da empresa e VNS clássico. O número de jornadas com duplas pegadas na solução obtida pela SCHC é de 28, de um total de 132 jornadas. Pode-se perceber também que a metaheurística VNS-VLNS apresenta os melhores resultados tendo um custo final de 1.471.176, enquanto a metaheurística obteve um custo de 1.518.460.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.621.732	148	111:23	23
Empresa	1.668.856	162	120:14	4
VNS Clássico	1.684.272	152	121:58	27
VNS-VLNS	1.583.644	148	77:41	17

Tabela 5.5: Características dos melhores resultados para a instância 04

Os resultados para a quarta instância são apresentados na Tabela 5.5. A solução com o menor custo final foi encontrada pela metaheurística VNS-VLNS. Porém a SCHC obteve melhores resultados que a metaheurística VNS clássico e o dados da empresa. Além disso, a SHCH apresenta a solução com número de jornadas igual ao VNS-VLNS.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.541.888	139	120:14	25
Empresa	1.580.964	155	108:11	1
VNS Clássico	1.562.864	142	116:06	23
VNS-VLNS	1.471.100	139	87:55	12

Tabela 5.6: Características dos melhores resultados para a instância 05

Para a quinta instância, os resultados obtidos são apresentados na tabela 5.6. Pode-se concluir que a SCHC obteve melhor solução que a metaheurísticas VNS Clássico e o dados da empresa. Porém, quando comparado com o VNS-VLNS, a solução obtida apresenta um aumento em todos os atributos avaliados, exceto o numero de jornadas.

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	1.169.588	100	120:27	29
Empresa	1.253.100	124	54:35	0
VNS Clássico	1.182.808	104	95:02	24
VNS-VLNS	1.152.552	109	52:18	10

Tabela 5.7: Características dos melhores resultados para a instância 06 com custo 5.000 para jornadas do tipo dupla pegada

Na Tabela 5.7 são apresentados os resultados das melhores soluções obtidas para a instância 6, correspondente a um dia de sábado. Para este problema, a SCHC, apesar de não

apresentar a melhor solução, é perceptível que ela reduz o número de jornadas. Novamente, a metaheurística VNS-VLNS apresenta a solução com o menor custo de "Fo".

Métodos	Fo	Total de jornadas	Total de horas extras	Total de duplas pegadas
SCHC	592.840	53	53:30	10
Empresa	686.468	68	26:57	0
VNS Clássico	596.088	54	46:12	9
VNS-VLNS	601.296	57	27:35	5

Tabela 5.8: Características dos melhores resultados para a instância 07

Para a última instância, as melhores soluções encontradas por cada método são apresentadas na Tabela 5.8. A metaheurística SCHC apresentou o melhor resultado nesta instância. O número de jornadas foi reduzido, consequentemente obtendo assim um custo de f_0 menor que as outras metaheurísticas e os dados da empresa.

6

Considerações Finais

A resolução do (PPT) é de grande importância para o planejamento operacional de empresas de transporte. Algumas empresas do setor de transporte ainda utilizam meios manuais para gerar as jornadas de trabalho de suas tripulações. Devido a esse fato há a necessidade de trabalhos que facilitem a geração das escalas(jornadas), como abordada nesse trabalho, especificamente para tripulação de ônibus urbano.

Neste trabalho foi proposto o desenvolvimento da metaheurística SCHC para resolver o Problema de Programação da Tripulação. A utilização desta abordagem, é justificada devido ao fato de que os métodos heurísticos atualmente são mais utilizados, além de apresentar um histórico com sucesso no resultados obtidos.

Os resultados obtidos pela metaheurística apresentam soluções com custo inferior às adotadas pela empresa e obtidas pela VNS Clássico. Nos resultados obtidos, as soluções de menores custos encontradas pela metaheurística SCHC apresentam uma quantidade de duplas pegadas dentro das limitações praticas da solução. Por outro lado, o SCHC apresentarão seu custo operacional maiores que as soluções encontradas pela metaheurística VNS-VLNS.

De modo geral, pode-se concluir que a utilização de métodos heurísticos para resolução do Problema da Programação da Tripulação mostrou resultados satisfatórios quando comparados a outros métodos. A utilização da metaheurística SCHC se mostrou eficiente, de modo a gerar soluções boas.

Como possíveis trabalhos futuros, propõe-se a utilização de um método híbrido, composto pelo SCHC e um possível método exato. Além de um melhor aprofundamento na calibração da metaheurística. Outro estudo posterior deve ser realizado na definição do limite do contador c , de modo ao mesmo ser alocado dinamicamente se adequando as soluções correntes encontradas.

Referências bibliográficas

- [1] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [2] John E Beasley. An algorithm for set covering problem. *European Journal of Operational Research*, 31(1):85–93, 1987.
- [3] Célio Freitas Bouzada. *Custo do transporte coletivo por ônibus*. Editora C/Arte, 2003.
- [4] Edmund K. Burke and Yuri Bykov. The late acceptance hill-climbing heuristic. In *Department of Computing Science and Mathematics, University of Stirling*, number CSM-192. June 2012.
- [5] Yuri Bykov and Sanja Petrovic. A step counting hill climbing algorithm. *Nottingham University Business School Research Paper*, (2013-10), 2013.
- [6] Martin Desrochers and Francois Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13, 1989.
- [7] SEG Elias. The use of digital computers in the economic scheduling for both man and machine in public transport. *Kansas, EUA: Technical Report*, 49, 1964.
- [8] Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, Bowie Owens, and David Sier. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1-4):21–144, 2004b.
- [9] Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27, 2004a.
- [10] Matteo Fischetti, Silvano Martello, and Paolo Toth. The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6):849–858, 1987.

- [11] Sarah Fores, Les Proll, and Anthony Wren. An improved ilp system for driver scheduling. In Nigel H. M. Wilson, editor, *Computer-Aided Transit Scheduling*, pages 43–61. Springer, 1999.
- [12] Sylvain Fournier. Branch-and-price algorithm for a real-life bus crew scheduling problem. *ERPOSul*, 2009.
- [13] Christian Friberg and Knut Haase. An exact branch and cut algorithm for the vehicle and crew scheduling problem. In Nigel H. M. Wilson, editor, *Computer-Aided Transit Scheduling*, pages 63–80. Springer, 1999.
- [14] Tiago Luiz Gonçalves. Meta-heurísticas para o problema de programação de tripulações. Master's thesis, Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, 2010.
- [15] Helena R Lourenço, José P Paixão, and Rita Portugal. Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, 35(3):331–343, 2001.
- [16] Lisa K Luedtke. Rucus ii: a review of system capabilities. *Computer Scheduling of Public Transport*, 2:61–116, 1985.
- [17] Margaret Parker and Barbara M e Smith. Two approaches to computer crew scheduling. *Computer Scheduling of Public Transport. Amsterdam: North-Holland*, pages 193–222, 1981.
- [18] Allexandre Fortes da Silva Reis and Gustavo Peixoto Silva. Um estudo de diferentes métodos de busca ea metaheurística vns para otimizar a escala de motoristas de ônibus urbano. *XXV ANPET - Congresso de Pesquisa e Ensino em Transportes*, 1:2374–2385, 2011.
- [19] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*, pages 111–114. Prentice Hall, December 2002.
- [20] André G Santos and Geraldo Robson Mateus. Crew scheduling urban problem: an exact column generation approach improved by a genetic algorithm. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1725–1731. IEEE, 2007.
- [21] Yindong Shen and Raymond SK Kwan. Tabu search for driver scheduling. In Stefan Voß and Joachim R. Daduna, editors, *Computer-Aided Transit Scheduling*, pages 121–135. Springer, 2001.
- [22] Gustavo Peixoto Silva and Claudio Barbieri da Cunha. Uso da técnica de busca em vizinhança de grande porte para a programação da escala de motoristas de ônibus urbano. *Transportes*, 18(2):37–45, 2010.

- [23] Gustavo Peixoto Silva and Allexandre Fortes da Silva Reis. A study of different metaheuristics to solve the urban transit crew scheduling problem. *Journal of Transport Literature*, 8:227–251, 2014.
- [24] Barbara M Smith and Anthony Wren. A bus crew scheduling system using a set covering formulation. *Transportation Research Part A: General*, 22(2):97–108, 1988.
- [25] Marcone Jamilson Freitas Souza, Leonardo Xavier Teixeira Cardoso, Gustavo Peixoto Silva, Margarida Maria Silva Rodrigues, and Silvia Maria Santana Mapa. Metaheurísticas aplicadas ao problema de programação de tripulações no sistema de transporte público. 2004.
- [26] Fernando Stefanello. Hibridização de métodos exatos e heurísticos para resolução de problemas de otimização combinatória. Master's thesis, Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Catarina, 2011.
- [27] EB Wilhelm. Overview of the rucus package driver run cutting program (runs). In *Transportation Science Section, Operations Research Society of America, etc., Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services. Preprints, Chicago*, 1975.
- [28] Anthony Wren. Scheduling vehicles and their drivers-forty years' experience. In *9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT), San Diego, California*, 2004.

Este trabalho foi redigido em \LaTeX utilizando uma modificação do estilo IC-UFAL. As referências bibliográficas foram preparadas no JabRef e administradas pelo \BIBTeX com o estilo LaCCAN. O texto utiliza fonte Fourier-GUTenberg e os elementos matemáticos a família tipográfica Euler Virtual Math, ambas em corpo de 12 pontos. A numeração dos capítulos segue com a família tipográfica Art Nouveau Caps.