

JONATA DE SOUZA CUNHA

Orientador: Túlio Ângelo Machado Toffolo

Co-orientador: Gustavo Peixoto Silva

**MODELOS E HEURÍSTICAS PARA OS PROBLEMAS DE
PROGRAMAÇÃO DE VEÍCULOS E DE TRIPULAÇÕES NO
SISTEMA DE TRANSPORTE PÚBLICO**

Ouro Preto
Agosto de 2021

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

**MODELOS E HEURÍSTICAS PARA OS PROBLEMAS DE
PROGRAMAÇÃO DE VEÍCULOS E DE TRIPULAÇÕES NO
SISTEMA DE TRANSPORTE PÚBLICO**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto.

JONATA DE SOUZA CUNHA

Ouro Preto
Agosto de 2021



FOLHA DE APROVAÇÃO

Jonata de Souza Cunha

Modelos e heurísticas para os problemas de Programação de Veículos e de Tripulações do sistema de transporte público

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

Aprovada em 27 de Agosto de 2021.

Membros da banca

Túlio Ângelo Machado Toffolo (Orientador) - Doutor - Universidade Federal de Ouro Preto
Gustavo Peixoto Silva (Coorientador) - Doutor - Universidade Federal de Ouro Preto
Rodrigo César Pedrosa Silva (Examinador) - Doutor - Universidade Federal de Ouro Preto
Jadson Castro Gertrudes (Examinador) - Doutor - Universidade Federal de Ouro Preto

Túlio Ângelo Machado Toffolo, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/08/2021.



Documento assinado eletronicamente por **Tulio Angelo Machado Toffolo, PROFESSOR DE MAGISTERIO SUPERIOR**, em 27/08/2021, às 11:28, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0210303** e o código CRC **09A89BBE**.

Resumo

Este trabalho aborda dois problemas comumente presentes no sistema de transporte público brasileiro: (i) o Problema de Programação de Veículos (PPV), que trata da atribuição de um conjunto de viagens programadas a um conjunto de veículos, e (ii) o Problema de Programação de Tripulação (PPT), que consiste em agrupar as viagens em jornadas de trabalho que serão atribuídas aos tripulantes. O objetivo central deste trabalho é avaliar modelos matemáticos e heurísticas para estes problemas, com o objetivo de minimizar a quantidade de veículos e de tripulantes necessários para atendimento às demandas de algumas empresas de ônibus da região metropolitana de Belo Horizonte. Para tratar o PPV é utilizado um modelo de fluxo em redes, capaz de rapidamente resolver o problema. O PPT, por outro lado, reconhecido pela sua dificuldade de resolução, é modelado como um problema de particionamento de conjuntos. Uma estratégia de geração de colunas envolvendo heurísticas para resolução do problema de *pricing* é empregada para resolver a relaxação linear do problema. A partir do modelo resultante, uma heurística de recobrimento é utilizada para obter soluções inteiras. Experimentos computacionais demonstram a qualidade dos limites inferiores obtidos com a relaxação linear, bem como o potencial da heurística de recobrimento utilizada. Soluções com *gap* de otimalidade inferior a cinco por cento foram geradas para a maioria das instâncias consideradas. Por fim, os resultados obtidos são comparados com resultados da literatura, evidenciando a qualidade das abordagens proposta.

Palavras-chave: Programação de Veículos, Programação de Tripulação, Geração de Colunas, Heurística de Recobrimento, Otimização Linear e Inteira, Pesquisa Operacional.

Abstract

This work addresses two problems present in the Brazilian public transportation system: (i) the Vehicle Scheduling Problem (VSP), which deals with the assignment of a set of jobs (trips) to a set of vehicles, and (ii) the Crew Scheduling Problem (CSP), which consists in grouping these jobs into daily work routines that will be assigned to the crew. The main goal of this work is to evaluate mathematical models and heuristics for these problems, with the objective of minimizing the number of vehicle and crew members necessary to cover the demand of bus companies from the metropolitan region of Belo Horizonte. To address the VSP, a network flow model is employed, being capable of quickly solving all problem instances. The CSP, in contrast, is well known as a hard problem to solve, being modeled as a set covering problem. A column generation strategy that includes heuristics to solve the pricing problem is used to solve the linear relaxation of the problem. Considering the resulting model, a cover heuristic is employed to obtain lower bounds. Computational experiments show the high quality of the lower bound obtained from the linear relaxation, as well as the potential of the cover heuristic to produce high quality solutions. In fact, solutions with up to 5% of optimality gap were produced for the majority of the instances. Finally, these results are compared against results from the literature, showing the quality of the proposed approaches.

Keywords: Vehicle Scheduling, Crew Scheduling, Column Generation, Cover Heuristic, Linear and Integer Programming, Operations Research.

Agradecimentos

O desenvolvimento deste trabalho só foi possível graças a participação de pessoas fundamentais em minha vida. Gostaria de agradecer:

Primeiramente ao meu irmão Eider Henrique da Cunha, pela ajuda financeira e principalmente por todo o incentivo para continuar os estudos. Aos meus familiares, Maria das Graças de Souza Cunha, Paulo Henrique da Cunha e Bruna Regina Cunha por sempre me fornecer todo apoio, incentivo, amor e companheirismo ao longo de toda a vida.

Em especial, a minha noiva Maria Juliana Coelho por sempre me apoiar mesmo nos momentos mais difíceis, pela paciência e pelo amor.

Ao meu orientador Túlio Ângelo Machado Toffolo e co-orientador Gustavo Peixoto Silva, por confiarem em minha capacidade, pela paciência e apoio para a realização deste trabalho.

Aos meus amigos Alan Erse, Luis Roberto, Gabriel Carvalho e Gustavo Presoti por toda energia positiva, amizade e apoio. Vocês deixaram essa jornada na universidade muito mais leve.

A todos os meus amigos e familiares, que me acompanharam nesta jornada sempre torcendo pelo meu sucesso.

A Universidade Federal de Ouro Preto (UFOP), por todo suporte. Por fim, a todos os professores do departamento de computação da UFOP, por todo o conhecimento repassado.

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos Geral e Específicos	2
1.3	Organização do Trabalho	2
2	Revisão de Literatura	3
2.1	Problema de Programação de Veículos - PPV	3
2.2	Problema de Programação da Tripulação	5
3	Descrição dos Problemas	7
3.1	Problema de Programação de Veículos - PPV	7
3.1.1	Restrições	8
3.2	Problema de Programação da Tripulação - PPT	9
3.2.1	Restrições	10
4	Metodologia Sequencial	11
4.1	Problema de Programação de Veículos	11
4.1.1	Modelo de Pseudo-Designação para o PPV	12
4.2	Problema de Programação da Tripulação	14
4.2.1	Modelo de cobertura de conjuntos para o PPT	14
4.2.2	Resolvendo o modelo de cobertura de conjuntos	15
5	Resultados	22
5.1	Resultados para o Problema de Programação de Veículos	22
5.2	Resultados para o Problema de Programação da Tripulação	24
5.3	Comparação com resultados da literatura	26
6	Conclusão e Trabalhos Futuros	29
	Referências Bibliográficas	30

Lista de Figuras

4.1	Rede GPD construído para o problema descrito anteriormente	13
5.1	Gráfico do Tempo x Viagens	24
5.2	Gráfico comparativo entre a função objetivo do modelo de Souza (2014) e da Heurística de Recobrimento	28

Lista de Tabelas

3.1	Tabela de horários	8
3.2	Tabela de Tarefas	9
4.1	Exemplo básico PPV	12
5.1	Resultado PPV	23
5.2	Resultados do PPT na metodologia sequencial	25
5.3	Comparativo modelo de cobertura de conjuntos x modelo de Souza (2014)	27

Lista de Algoritmos

1	Geração de Colunas	16
2	Heurística gulosa PPT	17
3	Heurística Journey Shortest Path	21

Capítulo 1

Introdução

Com o aumento da população a demanda por transportes tem se tornado cada vez maior. Isso tem como resultado um crescimento acelerado do tráfego de veículos nas cidades, que gera problemas na mobilidade urbana como: congestionamentos, poluição do ar, poluição sonora e acidentes. Segundo a pesquisa da Proteste (2015), feita em oito capitais, os brasileiros gastam em média 1 hora e 20 minutos por dia em transportes públicos. Esse tempo é muito elevado e faz com que muitas pessoas deixem de usar o transporte público, buscando o transporte individual, elevando assim os problemas supracitados. Uma forma de melhorar essa situação é com o aprimoramento do sistema de transporte público, tanto com a utilização de novas tecnologias, quanto com a otimização da operação do sistema.

Antes de uma empresa de transporte público por ônibus urbano começar a operar, ocorre o planejamento operacional, que segundo Visentini et al. (2017) é composto por quatro etapas: *(i)* a definição da tabela de horários, que mostra a sequência das viagens de cada linha da empresa, contendo os horários de partida e de chegada de cada viagem; *(ii)* o escalonamento da frota em operação, quando ocorre a atribuição das viagens aos veículos; *(iii)* o escalonamento da tripulação necessária para conduzir os veículos, gerando as jornadas de trabalho de cada tripulação e *(iv)* o rodízio das tripulações, que distribuem as diferentes jornadas de trabalho ao longo do mês.

Os maiores custos operacionais das empresas de transporte público estão concentrados na manutenção das frota rodante e principalmente na remuneração das tripulações. O Problema de Programação de Veículos - PPV consiste em minimizar os gastos referentes ao escalonamento de veículos. O Problema de Programação da Tripulação - PPT, por sua vez, consiste em minimizar os gastos relacionados ao escalonamento da tripulação.

Segundo (Silva e Gualda (2000)) o PPV trata da atribuição do conjunto de viagens contidas na tabela de horários da empresa a um conjunto de veículos disponíveis, de modo que cada viagem seja associada a um veículo e o custo total seja mínimo. Os veículos partem de uma ou mais garagens, realizam as suas viagens e devem retornar à garagem de origem após a última viagem. Já o PPT consiste em atribuir às tripulações (motoristas e cobradores quando

for o caso) a tarefa de conduzir os veículos em operação, de modo que todos os veículos, nas diferentes linhas atendidas pela empresa, sejam atendidos pelos tripulantes com o menor custo possível Silva e Cunha (2010).

Neste trabalho, é tratado o Problema de Programação dos Veículos e de suas Tripulações, considerando apenas uma garagem, através da metodologia sequencial. Sendo assim, primeiro é definida a programação dos veículos e, em seguida, é resolvido o PPT, quando são definidas as jornadas de trabalho das tripulações, tendo como base o resultado da programação dos veículos. Por fim, é feito um comparativo com outro trabalho da literatura mostrando qual metodologia obteve melhores resultados.

1.1 Motivação

O PPV e o PPT são problemas muito estudados na literatura, devido a inúmeras aplicações referentes à organização de transporte em muitas áreas, pois além de ser um problema pertencente à classe NP-Difícil (Freling et al., 1999) permite um planejamento operacional mais efetivo e possibilita economias significativas para as empresas de transporte. Além disso, com um planejamento operacional mais efetivo os problemas na mobilidade urbana supracitados podem ser amenizados.

1.2 Objetivos Geral e Específicos

O objetivo geral deste trabalho é realizar a implementação de modelos e heurísticas para tratar o PPV e o PPT de forma sequencial, visando minimizar a quantidade de veículos e de tripulantes necessários para realizar as viagens contidas na tabela de horários de algumas empresas de ônibus da região metropolitana de Belo Horizonte. Os objetivos específicos são: Realizar o escalonamento da tripulação e de veículos com custos mínimos, atender todas as restrições das empresas transportadoras em consonância com as restrições das leis vigentes do Brasil. Por fim, realizar um comparativo entre os resultados obtidos com os modelos implementados com outros resultados disponíveis na literatura.

1.3 Organização do Trabalho

O Capítulo 2 trata da revisão da literatura, apresentando trabalhos relacionados ao PPV e PPT. O Capítulo 3 fornece os detalhes de cada problema. A seguir, no Capítulo 4 são apresentados modelos e heurísticas empregados para resolver cada problema de forma sequencial. No Capítulo 5 experimentos e comparativo com outro trabalho da literatura são demonstrados. Por fim, o Capítulo 6 apresenta a conclusão e trabalhos futuros.

Capítulo 2

Revisão de Literatura

O Problema de Programação de Veículos (PPV) e o Problema de Programação da Tripulação (PPT) conta com muitas variações em função das situações vivenciadas em cada realidade. Desta forma, diferentes abordagens são encontradas na literatura na área de Pesquisa Operacional. Este capítulo apresenta uma breve revisão de alguns trabalhos que abordam os problemas estudados.

Este capítulo é dividido em duas seções. A Seção 2.1 contém os trabalhos relacionados ao PPV e A Seção 2.2 apresenta os trabalhos sobre o PPT.

2.1 Problema de Programação de Veículos - PPV

O PPV representa um importante tópico de pesquisa na literatura, devido a inúmeras aplicações referentes à organização do transporte em muitas áreas. Esse problema recebe muita atenção de pesquisadores de todo o mundo, tanto do ponto de vista teórico quanto do ponto de vista prático. Apesar de inúmeras variantes do problema de programação escalonamento de veículos terem sido introduzidas, considerando abordagens eficientes, novas restrições na organização de transporte motivam os pesquisadores a fazerem mais esforços na exploração deste problema.

O PPV trata da atribuição de um conjunto de viagens programadas a um conjunto de veículos disponíveis, de modo que cada viagem seja associada a um veículo. Os veículos partem de uma ou mais garagens e cada veículo, na última viagem, deve retornar a sua garagem inicial. O objetivo do PPV é minimizar a quantidade total de veículos, o total de deslocamentos sem passageiras e o tempo total de espera dos veículos nos terminais.

Para resolver o PPV com múltiplas garagens Dell'Amico et al. (1993) projetaram um novo algoritmo heurístico baseado no problema do caminho mínimo. Os testes computacionais foram realizados envolvendo até 1.000 viagens e 10 garagens. Os resultados mostraram que a nova abordagem sempre produz soluções aproximadas muito rígidas (cerca de 1% do ideal) e em um curto tempo de computação, superando outras heurísticas da literatura, à época. Por

outro lado, Ciancio et al. (2018) utilizam uma heurística gulosa para obter a solução inicial, com a técnica de *Simulated Annealing* a solução inicial é melhorada. Os resultados mostraram que o algoritmo proposto é capaz de encontrar rapidamente boas soluções dentro de um limite de tempo.

Um dos problemas das abordagens padrão é a impossibilidade de estudar um *trade-off* entre transferências oportunas de passageiros e os custos operacionais em termos de uso do veículo. Em outras palavras o que é mais importante, atender os passageiros de forma mais rápida ou ter uma quantidade menor de veículos? Para resolver esse problema, Ibarra-Rojas et al. (2014) apresentam dois modelos de programação linear inteira para os problemas de horário e agendamento de veículos em consonância com a combinação dos problemas em um modelo integrado com dois objetivos. Esse modelo conseguiu resolver instâncias de até 50 linhas e 5 garagens de ônibus em um tempo hábil. Visentini et al. (2017) propuseram três novas abordagens para resolver o problema integrado de geração de tabela de horários e escalonamento de veículos com frota heterogênea. Os resultados obtidos pelo modelo VTSP-TW (*Vehicle Type Scheduling Problem with Time Windows*) em instâncias reais, conseguiram reduzir em até 8 veículos, na maior instância, ao aplicar janela de tempo de 1 minuto. De modo geral, os modelos mostraram-se eficientes na resolução dessas instâncias, conseguindo gerar economias consideráveis.

Algumas empresas de transportes público utilizam veículos com diferentes características, gerando o Problema de Programação de Veículos com Diferentes Tipos (*vehicle scheduling problem with multiple vehicle types*). Hassold e Ceder (2014) propuseram uma nova metodologia baseada em uma rede de fluxo com custo mínimo utilizando conjuntos de horários ideais para Pareto em linhas de ônibus individuais. O método desenvolvido permite estipular o uso de um tipo de veículo específico para uma viagem ou permitir uma substituição por um veículo maior ou por uma combinação de veículos menores com a mesma ou maior capacidade total. Os resultados mostraram que uma substituição de veículos é benéfica e pode levar a reduções significativas de custos na faixa de mais de 27%.

As abordagens padrão do PPV combinam objetivos de forma ponderada para formar um único objetivo e, em seguida, usa um modelo de otimização mono objetivo para resolvê-lo podendo ter como resultado apenas uma solução. Isso é um problema, pois, várias soluções são altamente desejáveis para um tomador de decisão selecionar a solução com base em várias considerações e preferências. Para resolver este problema, Zuo et al. (2015) apresentaram uma metodologia que utiliza um Algoritmo Genético Multiobjetivo combinado com um procedimento de ajuste para obter um conjunto de soluções Pareto. Essa abordagem foi aplicada a um problema de programação de veículos real de uma linha de ônibus em Nanjing, na China. As experiências mostraram que essa abordagem superou uma abordagem baseada na experiência, um Algoritmo Genético Padrão e uma abordagem exata.

Uma outra possível abordagem para o PPV consiste em resolvê-lo em conjunto com pe-

pequenas modificações na tabela de horário, com o objetivo de obter melhorias na quantidade final de veículos. Sendo assim, Nakao e Silva (2017) aplicaram uma metaheurística VNS (*Variable Neighborhood Search*), que tem como entrada uma tabela de horários inicial, a partir da qual são realizadas pequenas modificações que não comprometem a qualidade do serviço e posteriormente é resolvido o respectivo PPV. Essa abordagem foi aplicada ao transporte público da região metropolitana de Belo Horizonte, levando em conta todas as restrições já existentes na empresa transportadora. Esse método proposto comparou os resultados da frota que era utilizada antes com a frota retornada pelo algoritmo. Os resultados demonstraram que o método proposto consegue melhorar a função objetivo e em alguns casos diminuir a quantidade de veículos.

2.2 Problema de Programação da Tripulação

Segundo Mauri et al. (2004), o PPT tem como objetivo atribuir as tarefas de condução dos ônibus aos motoristas e cobradores (tripulações), gerando suas jornadas de trabalho, de modo que as viagens das diferentes linhas atendidas pela empresa sejam realizadas com o menor custo possível. O PPT é um problema muito estudado na literatura, pois, um escalonamento de tripulação, ou seja, um conjunto de jornadas eficientes permite economias significativas para as empresas de transporte, em consonância com a dificuldade de realizar um escalonamento eficiente. Além disso, o PPT é um importante problema de pesquisa que pertence à classe NP-Difícil (Chen e Hou, 1988).

Uma abordagem de geração de colunas foi proposta por Desrochers e Soumis (1989) para resolver o PPT. Nessa abordagem, os autores decompõem o problema em duas partes: um problema mestre e um subproblema de *pricing*. O problema mestre consiste em um problema de cobertura de conjuntos, em que cada conjunto representa uma jornada de trabalho a partir de dias de trabalho já conhecidos e viáveis. O subproblema é usado para gerar novas jornadas viáveis para melhorar a solução atual do problema mestre. O método foi testado com sucesso em problemas reais. Ao comparar com as abordagens da época, essa tem duas vantagens, tendo em vista que pode produzir um limitante inferior muito bom e válido para o custo da solução bem como pode ser usado mesmo que haja vários dias de trabalho.

O PPT, para ser aplicado em algumas empresas transportadoras, precisa considerar restrições adicionais que podem dificultar ainda mais a resolução do problema. De Leone et al. (2011), ao aplicar o modelo com restrições adicionais, verificaram que ele é eficaz apenas para instâncias de tamanhos pequenos ou médios. Para resolver instâncias maiores, os autores propuseram um procedimento de busca GRASP, que foi testado em instâncias reais fornecidas pela PluService Srl, mostrando a eficácia da metaheurística projetada.

Um algoritmo heurístico foi proposto por Constantino et al. (2017) para resolver instâncias de grande escala (a maior instância é constituída de 2.300 tarefas). Esse algoritmo consiste

em duas fases. Primeiramente o algoritmo constrói uma solução inicial viável, em seguida, o algoritmo tenta melhorar a solução por dois procedimentos diferentes. Um procedimento considera todo o conjunto de tarefas e as divide em um conjunto de tarefas parciais que são recombinadas. O outro procedimento procura melhorar a alocação das tarefas longas, eliminando horas extras e inserindo-as em outra jornada. Os resultados gerais indicaram que a abordagem proposta é competitiva para solucionar instâncias grandes.

Para melhorar a eficiência da técnica de geração de colunas, Shijun (2013) propôs uma estratégia que consiste em pré-compilar um conjunto razoavelmente grande de mudanças “boas” em potencial usando o conhecimento específico do problema. Durante o processo de geração de colunas, o problema de caminho mínimo com restrição de recursos é construído para evitar mudanças em potencial até que nenhuma coluna com custo reduzido negativo (NRC) exista no conjunto de turnos. Os resultados experimentais mostraram que o algoritmo proposto supera os algoritmos tradicionais de geração de colunas.

A metodologia sequencial para a resolução do problema de programação de veículo e tripulação é uma metodologia ainda pouco explorada na literatura. Porém, é considerada promissora no que diz respeito à qualidade das soluções geradas.

Capítulo 3

Descrição dos Problemas

Nesse Capítulo, são apresentadas informações relacionadas ao Problemas de Programação de Veículos e de Programação de Tripulantes. São apresentadas as características específicas do caso estudado, envolvendo tanto restrições adotadas pela empresa quanto as restrições decorrentes das leis trabalhistas vigentes e os acordos coletivos firmados pela categoria.

A organização deste Capítulo está conforme descrito a seguir. Inicialmente, na Seção 3.1 é abordado o Problema de Programação de Veículos. Por fim, Na Seção 3.2 é apresentado o Problema de Programação da Tripulação.

3.1 Problema de Programação de Veículos - PPV

No Problema de Programação de Veículos (PPV) é realizado um escalonamento de veículos, que atribui um conjunto de viagens para um conjunto de veículos com o intuito de obter o menor custo em relação ao total de veículos e ao custo operacional. No custo operacional é considerado o tempo em que o veículo fica ocioso no terminal e o tempo em que o veículo gasta para se deslocar entre os terminais, estando fora de operação. Este deslocamento é dito *viagem ociosa*. Além disso, é indispensável atender todas as restrições da empresa de acordo com as restrições impostas por leis.

Para resolver o PPV é necessário ter como dados de entrada uma tabela de horários, que é composta pelas características das viagens que a empresa deve realizar. Nessa tabela de horários, cada viagem é identificada por meio de um conjunto de informações relevantes para o processo de resolução do problema.

Tabela 3.1: Tabela de horários

Hora Início	Local Início	Hora Fim	Local Fim	Embarque	Desembarque
00:30	DIAM3050	01:30	DIAM3050	1	2
01:00	BARR3055	02:10	BARR3055	1	2
01:35	DIAM3050	02:10	DIAM3050	1	1
02:10	DIAM3050	03:20	BARR3055	2	2
02:00	BARR3055	03:30	BARR3055	1	2

Fonte: Silva e Gualda (2000)

A Tabela 3.1 contém os seguintes dados de cada viagem: horário de início da viagem (em horas), o ponto inicial que corresponde ao local de início da viagem, horário de término da viagem (em horas), o ponto final que se refere ao local em que a viagem termina. As duas últimas colunas se referem respectivamente aos tempos destinados ao embarque e ao desembarque de passageiros (em minutos), sendo o primeiro relacionado ao horário de início da viagem e o segundo ao horário de fim da viagem. Estes tempos de embarque e desembarque são normalmente acrescentados ao tempo de duração da viagem.

Para exemplificar, considere a primeira viagem apresentada na Tabela 3.1. Essa viagem começa às 00:30 horas no ponto “DIAM3050” e termina às 01:30 horas no ponto “DIAM3050”. As duas últimas colunas indicam que o veículo deve permanecer no mínimo um minuto no terminal esperando para o embarque e dois minutos para o desembarque de passageiros ao final da viagem.

Uma outra informação relevante para o PPV é o tempo estimado de deslocamento entre os pontos de parada dos veículos, por exemplo: O tempo estimado de deslocamento entre a garagem e DIAM3050 é 18 minutos. Essa informação é utilizada para determinar o tempo em que um veículo gasta fora de operação, ou seja, é quando ocorre o deslocamento entre pontos sem estar carregando passageiros um ponto ao outro. Esse deslocamento é denominado *viagem ociosa* e entra no cálculo do custo da operação.

3.1.1 Restrições

Para que uma solução do PPV seja factível, é necessário atender uma série de restrições. Nesse trabalho as restrições levadas em consideração são referentes às leis vigentes no Brasil, e as restrições práticas são referentes a uma empresa de transporte público da região metropolitana de Belo Horizonte. As seguintes restrições foram implementadas:

1. No final de uma viagem o veículo pode esperar no local de término até iniciar uma nova viagem, se deslocar para outro ponto para iniciar uma outra viagem ou retornar temporariamente para a garagem.
2. Todos veículos iniciam e terminam a atividade na garagem.

3. Os veículos podem fazer “dupla pegada” que é quando ele fica ocioso entre duas viagens por um tempo maior que duas horas. Neste caso supõe-se que o veículo retornará temporariamente à garagem para depois voltar à operação. O tempo que o veículo permanece na garagem não é computado como ociosidade.
4. Se o tempo entre duas viagens for menor que duas horas ele é considerado como ociosidade.
5. Um veículo pode trocar de linha durante a operação, ou seja, ele pode realizar viagens de diferentes linhas ao longo do dia.

3.2 Problema de Programação da Tripulação - PPT

Para gerar as jornadas de trabalho das tripulações é necessário ter como dados de entrada o conjunto das tarefas a serem realizadas. Uma tarefa é identificada por um conjunto de viagens sequenciais, executadas por um mesmo veículo, sem que haja entre elas, condições para a troca de tripulações. O conjunto das tarefas é construída a partir da solução do Problema de Programação de Veículos.

O objetivo do PPT é atribuir um conjunto de tarefas para um conjunto de tripulações visando minimizar a quantidade total de tripulantes e o custo operacional, que envolve o tempo de horas extras, de horas ociosas entre outros fatores. Nesse escalonamento é obrigatório atender todas as restrições da empresa transportadora e as restrições impostas por leis e acordos trabalhistas.

Tabela 3.2: Tabela de Tarefas

Veículo	Hora Início	Local Início	Hora Fim	Local Fim
1	00:30	DIAM3050	02:20	GARAGEM
2	01:00	BARR3055	02:25	GARAGEM
3	02:10	DIAM3050	03:25	GARAGEM
4	02:00	BARR3055	03:45	GARAGEM

Fonte: Elaborado pelo autor

A Tabela 3.2 representa as tarefas, contendo as seguintes informações: o número do veículo que realizará a tarefa, o horário de início da tarefa (em horas), o ponto inicial que corresponde ao local de início da tarefa, horário de término da tarefa (em horas), o ponto final que se refere ao local em que a tarefa termina. Para exemplificar, considere a primeira tarefa apresentada na Tabela 3.2. Essa tarefa começa às 00:30 horas no ponto “DIAM3050” e termina às 02:20 horas no ponto “GARAGEM”.

3.2.1 Restrições

Para que uma solução do PPT seja viável é necessário atender a uma série de restrições. As restrições impostas por leis que foram levadas em consideração são referentes as leis vigentes no Brasil e as restrições da empresa transportadora são relacionadas a uma empresa da região metropolitana de Belo Horizonte. Neste trabalho foram consideradas as seguintes restrições:

1. Uma jornada é considerada “dupla pegada” quando o tempo entre duas tarefas consecutivas possui um tempo maior do que duas horas. Neste caso o motorista não é pago pelo tempo em que ele ficou ocioso entre as duas pegadas.
2. Se o tempo entre duas tarefas for menor que duas horas a jornada é considerada como “jornada corrida”, e o motorista é pago nesse tempo ocioso entre as tarefas.
3. O intervalo de tempo de descanso, em casa, entre o final e o início da jornada de trabalho no dia seguinte deve ser de pelo menos 11:00 corridas para todos os motoristas. Sendo assim, se um motorista terminar sua jornadas trabalhar às 00:00 horas, ele só poderá começar a trabalhar no outro dia a partir de 11:00 horas da manhã.
4. A duração da jornada de trabalho é de 06 horas e 40 minutos de duração, podendo ser acrescidas até duas horas extras.
5. Em jornadas com duração de até 4 horas, o motorista não necessita de tempo mínimo de descanso.
6. Em jornadas com duração entre 4 e 6 horas, o motorista necessita de 15 minutos como tempo mínimo de descanso.
7. Em jornadas com duração maior do que 6 horas, o motorista necessita de pelo menos 30 minutos de descanso.
8. Uma dupla pegada é dividida na 1^a e na 2^a pegada. Neste caso cada pegada não pode exceder 06:00 horas de duração. Neste intervalo não está previsto qualquer tempo de descanso.
9. Nas jornadas corridas não pode haver trocas de pontos. Nas duplas pegadas pode ocorrer a troca de pontos desde que essa troca ocorra entre a 1^a e 2^a pegada.
10. Cada jornada pode ter no máximo uma troca de veículo.

Capítulo 4

Metodologia Sequencial

Nesse Capítulo, é apresentada a modelagem do Problema de Programação de Veículos e Tripulação de forma sequencial, ou seja, primeiro é resolvido o Problema de Programação de Veículos e com o resultado é feito o escalonamento da tripulação.

A organização deste Capítulo está conforme descrito a seguir. Inicialmente, na Seção 4.1 é abordada a modelagem do Problema de Programação de Veículos. Na Seção 4.2 é apresentada a modelagem do Problema de Programação da Tripulação.

4.1 Problema de Programação de Veículos

A resolução do Problema de Programação de Veículos para uma única garagem foi baseada no trabalho de Silva e Gualda (2000). Neste artigo o PPV foi definido da seguinte maneira: Seja um conjunto V com n viagens, cada viagem $i \in V$ é representada por:

- b_i : o ponto inicial,
- e_i : o ponto final,
- d_i : o horário de partida de b_i ,
- a_i : o horário de chegada em e_i ,
- $t_{i,j}$: representa o tempo de viagem morta de i para j .

O tempo de espera da viagem é representado por $\epsilon_{i,j}$, o seu valor é dado por $d_j - a_i - t_{i,j}$. A garagem é representada pelos nós r e s , no qual r é o nó de partida da garagem e s é o nó de retorno à garagem. Um par de viagens (i, j) é dito compatível se $t_{i,j} \leq d_j - a_i$ e o custo do arco (i, j) ligando as viagens i e j é dado por:

$$\begin{array}{ll} \alpha t_{i,j} + \tau & \text{Se } i = r \\ \alpha t_{i,j} + \gamma & \text{Se } \epsilon_{i,j} > 120 \\ \alpha t_{i,j} + \beta \epsilon_{i,j} & \text{Se } \epsilon_{i,j} \leq 120 \end{array}$$

Onde $\alpha, \beta, \gamma, \tau$ são pesos definidos pelo usuário. Os valores destes pesos variam de acordo com a situação operacional, porém α é sempre maior do que β em situações reais. O custo de se utilizar o veículo é definido em τ . Esse custo é atribuído à primeira viagem de cada veículo, que necessariamente parte da garagem. Por fim, a última viagem de cada veículo deve retornar a garagem.

4.1.1 Modelo de Pseudo-Designação para o PPV

O modelo de pseudo-designação definido por Silva e Gualda (2000) atua sobre dois conjuntos disjuntos de nós, ligando cada nó do conjunto origem a pelo menos um nó do conjunto destino, de forma que o custo total das ligações sejam mínimos. Sendo assim, para cada viagem $i \in V$ são definidos os nós $\gamma(i)$ de início e $\delta(i)$ de final de viagem. A garagem é representada pelo nó r de partida e pelo nó s de retorno, com ofertas e demandas de $n, n = |V|$ veículos respectivamente.

Tabela 4.1: Exemplo básico PPV

Viagem (i)	d_i	b_i	a_i	e_i
1	07:00	DIAM3050	08:30	DIAM3050
2	08:00	DIAM3050	08:50	DIAM3050
3	09:00	DIAM3050	11:00	DIAM3050
4	09:30	DIAM3050	10:00	DIAM3050

Fonte: Silva e Gualda (2000)

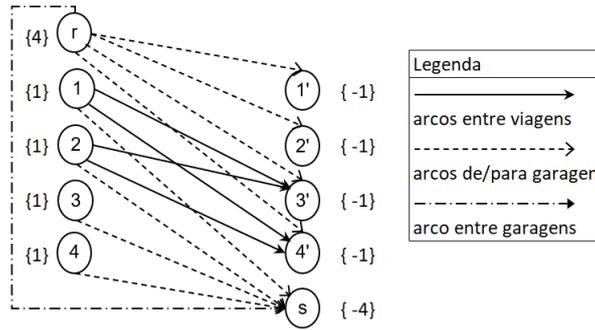
A partir da Tabela 4.1 é possível verificar que após executar a viagem 1, o mesmo veículo poderá executar as viagens 3 ou 4, pois estas viagens têm seus horários de partida posteriores ao horário de chegada da viagem 1, ou seja, as viagens 1 e 3 ou 1 e 4 são pares de viagens compatíveis. Por isso, são incluídos arcos ligando a viagem 1 às viagens 3 e 4. Esses arcos têm o custo de β vezes os tempos de espera de 30 e 60 minutos respectivamente. Os arcos que partem da garagem apresentam ainda o Custo Capital, o custo da viagem morta multiplicado por α , ou seja, o custo para o veículo ir da garagem até o terminal ou do terminal até a garagem.

Silva e Gualda (2000) realizam a divisão dos nós da rede em dois conjuntos disjuntos, sendo eles: $N_1 = \{r\} \cup \{\delta(i) | i \in V\}$ dos nós com oferta de fluxo e $N_2 = \{s\} \cup \{\gamma(i) | i \in V\}$ dos nós com demanda de fluxo. O conjunto de arcos ligando os nós de N_1 a N_2 é definido por $A = \{(r, \gamma(i)) | i \in V\} \cup \{(\delta(i), \gamma(j)) | (i, j) \text{ é um par de viagens compatíveis}\} \cup \{(\delta(i), s) | i \in V\}$. O primeiro subconjunto de A liga a partida da garagem até o início de cada viagem, o segundo subconjunto liga o final da viagem i com o início da viagem j , sendo (i, j) um par de viagens compatíveis, e o terceiro subconjunto liga o final de cada viagem ao retorno à garagem. Assim tem-se a rede bipartida $GPD = (N_1, N_2, A)$. O arco que liga as garagens tem custo nulo,

capacidade igual ao total de viagens n e serve para “absorver” os veículos economizados no processo de otimização.

Um exemplo deste tipo de rede é mostrado na Figura 4.1, na qual os valores entre chaves representam as respectivas ofertas e demandas de veículos, i corresponde ao final de cada viagem e i' ao início da viagem i . Os custos nos arcos $(\delta(i), \gamma(j))$, que na Figura 4.1 correspondem aos arcos (i, j') , seguem a expressão dada em (1).

Figura 4.1: Rede GPD construído para o problema descrito anteriormente



Fonte: Silva e Gualda (2000)

A formulação matemática para este modelo de pseudo-designação é dada então por:

$$\text{Minimizar} \quad \sum_{(i,j) \in A} c_{i,j} x_{i,j} \quad (4.1)$$

$$\text{s.t.} \quad \sum_{j \in N_2} x_{i,j} = 1 \quad \forall i \in T \setminus \{r\} \quad (4.2)$$

$$\sum_{i \in N_1} x_{i,j} = 1 \quad \forall j \in N_2 \setminus \{s\} \quad (4.3)$$

$$\sum_{j \in N_2} x_{r,j} = n \quad (4.4)$$

$$\sum_{i \in N_1} x_{i,s} = n \quad (4.5)$$

$$x_{i,j} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.6)$$

onde a variável de decisão $x_{i,j} = 1$ se a viagem j for executada pelo mesmo veículo imediatamente após a execução da viagem i , e $x_{i,j} = 0$ caso contrário. A expressão (4.1) minimiza o custo total no qual $c_{i,j}$ indica o custo de se utilizar o arco (i, j) , as restrições (4.2) e (4.3) asseguram que cada viagem será executada uma única vez, enquanto as restrições (4.4) e (4.5) estão relacionadas com as partidas e chegadas dos veículos à garagem. As restrições (4.6) garantem a integralidade da solução. A partir da solução deste modelo é possível construir os

blocos de viagens para os veículos. Para isso, basta percorrer nesta rede cada caminho saindo de r e chegando em s , correspondendo assim a um bloco de viagens a ser executado por um veículo. O conjunto de todos os caminhos disjuntos de r para s , contidos na solução, fornece a programação completa com o menor número de veículos e custo operacional.

4.2 Problema de Programação da Tripulação

Para tratar o Problema de Programação da Tripulação, primeiramente é resolvido o PPV através do Modelo de Pseudo-Designação que foi descrito na seção anterior. A solução do PPV retorna todas as atividades diárias de cada veículo, desde a sua partida da garagem, as viagens realizadas em serviço, possíveis viagens de reposicionamento, retorno temporário e finalmente, o seu retorno definitivo à garagem.

Com essas informações é necessário agrupar as viagens do veículo em “tarefas”, que são viagens consecutivas que devem necessariamente ser executadas por uma mesma tripulação. O agrupamento de 2 ou mais viagens em uma única tarefa ocorre por não haver uma oportunidade de troca da tripulação corrente por uma nova tripulação. Uma oportunidade de troca consiste de um intervalo de tempo mínimo que ocorre entre duas viagens (tempo de troca) em um ponto apropriado (pontos de troca) para que a troca das tripulações seja possível.

4.2.1 Modelo de cobertura de conjuntos para o PPT

A modelagem de cobertura de conjuntos proposta neste trabalho para tratar o PPT é definida da seguinte maneira: seja Ω o conjunto que contém todas as possíveis jornadas, e T o conjunto de todas as tarefas. O modelo considera então as variáveis λ_k , que assumem valor 1 se a jornada $k \in \Omega$ é escolhida para fazer parte da solução e valor 0 caso contrário.

Assim, o modelo considera:

- Ω : o conjunto de todas as jornadas viáveis;
- T_k : o conjunto de tarefas que pertencem à jornada k , $k \in \Omega$;
- c_k : o custo da jornada k , $k \in \Omega$.
- λ_k : variável binária que indica se a jornada $k \in \Omega$ faz parte da solução.

A formulação matemática de cobertura de conjuntos é dada pelas equações (4.7)-(4.9).

$$\text{Minimizar} \quad \sum_{k \in \Omega} c_k \lambda_k \quad (4.7)$$

$$\text{Sujeito a} \quad \sum_{k \in \Omega | t \in T_k} \lambda_k \geq 1 \quad \forall t \in T \quad (4.8)$$

$$\lambda_k \in \{0, 1\} \quad \forall k \in \Omega \quad (4.9)$$

A expressão (4.7) minimiza o custo total da solução, ou seja, busca selecionar as jornadas com o menor custo para o PPT. As Restrições (4.8) garantem que todas as tarefas vão ser executadas. Finalmente, a integralidade das variáveis λ_k é certificada pelas Restrições (4.9).

Karp (1972), demonstrou que o problema de cobertura de conjuntos é um problema que pertence à classe NP-Completo. Através da modelagem é possível verificar que utilizar o conjunto de todas as jornadas viáveis não é trivial, visto que a quantidade de todas jornadas viáveis é um número proibitivo. Assim, a próxima seção discute a estratégia adotada para resolver a formulação (4.7)-(4.9).

4.2.2 Resolvendo o modelo de cobertura de conjuntos

A estratégia de geração de colunas é utilizada para resolver a *relaxação linear* da formulação (4.7)-(4.9). Segundo Desrosiers e Lübbecke (2005), a geração de colunas é uma ferramenta indispensável na otimização computacional para resolver programas lineares que consideram de um número proibitivo de variáveis.

A estratégia de geração de colunas consiste em gerar as colunas necessárias para tratar o problema, sendo assim, ao invés de o modelo utilizar o conjunto de todas as jornadas viáveis ele vai utilizar somente as jornadas potencialmente úteis para a resolução do modelo. O modelo contendo um subconjunto das jornadas é chamado de *Mestre Restrito*, e esse subconjunto de jornadas é obtido através da resolução de um problema chamado *Pricing*.

A geração de colunas implementada neste trabalho recebe como entrada uma solução inicial heurística, cuja metodologia de geração é explicada na próxima seção. A partir da solução inicial é instanciado o modelo *Mestre* (modelo de cobertura de conjuntos), cujo objetivo é escolher dentre todas as jornadas aquelas que cobrem todas as tarefas de T com o menor custo. No entanto, o como dito anteriormente, apenas um subconjunto das jornadas é considerado. Cada jornada deste subconjunto é gerada pela resolução do problema de *Pricing*, que consiste em gerar uma coluna (ou jornada) válida que tenha o menor *custo reduzido*. O Algoritmo 1 apresenta a metodologia de geração de colunas implementada para resolver a relaxação linear da formulação (4.7)-(4.9).

Algoritmo 1: Geração de Colunas

```

1 jornadas  $\leftarrow$  solucaoHeuristica(tarefas)
2 mestre  $\leftarrow$  criaModeloMestre(jornadas, tarefas)
3 pricing  $\leftarrow$  criaModeloPricing(tarefas)
4 novaJornada  $\leftarrow$   $\emptyset$ 
5 faça
6   | mestre  $\leftarrow$  mestre + novaJornada
7   | duais  $\leftarrow$  resolverMestre()
8   | novaJornada  $\leftarrow$  resolverPricing(duais)
9 enquanto novaJornada  $\neq$   $\emptyset$ ;

```

O funcionamento do Algoritmo 1 de geração de colunas consiste nos seguintes passos:

1. O modelo de cobertura de conjuntos (mestre) recebe da solução inicial um conjunto de jornadas.
2. O modelo *Mestre* é resolvido e os duais das Restrições (4.8) são passados para o modelo de *Pricing*.
3. O *Pricing* por sua vez é resolvido e gera uma nova jornada que será incluída no modelo *Mestre*.
4. Esse processo termina quando o custo reduzido da jornada gerada pelo *Pricing* for maior ou igual a 0, ou seja, quando não for possível encontrar uma nova jornada que melhore a função objetivo do modelo *Mestre*.

Ao executar a estratégia de geração de colunas é resolvida a relaxação linear do modelo *Mestre*, ou seja, esse processo ignora a integralidade das variáveis. Sendo assim, é muito provável que o modelo use uma quantidade fracionária de uma jornada, tornando assim a solução inválida. Para gerar soluções inteiras uma heurística de recobrimento é aplicada, que consiste basicamente em resolver o *Mestre Restrito* (com um subconjunto das variáveis) considerando as restrições de integralidade.

Solução Inicial

Como apresentado na seção anterior, o algoritmo de geração de colunas apresentado neste trabalho necessita de uma solução inicial válida. Para isso, é necessário obter um conjunto de jornadas que atendam todas as tarefas sem infringir alguma das restrições citadas na Seção 3.2.

Segundo Davis e Impagliazzo (2004), o paradigma do algoritmo guloso é um dos mais importantes no projeto de algoritmos, devido à sua simplicidade e eficiência. Os algoritmos gulosos são comumente usados para fornecer resultados exatos para uma variedade de problemas. São frequentemente os melhores algoritmos de aproximação para problemas de

otimização complexos. Além disso, devido a sua simplicidade, são muito usados como heurísticas para problemas de otimização difíceis, mesmo quando suas razões de aproximação são desconhecidas ou reconhecidas como ruins no pior caso.

Sendo assim, devido as vantagens supracitadas a solução inicial é gerada a partir de uma heurística gulosa, representada pelo Algoritmo 2.

Algoritmo 2: Heurística gulosa PPT

```

1  solucao ← ∅
2  while tarefa ∈ entradaPPT do
3      melhorAtual ← ∅
4      novaTrip ← True
5      while trip ∈ solucaoTripulacao do
6          jornada ← geraJornada()
7          if jornadaValida(jornada) then
8              novaTrip ← False
9              if solucaoAtual.custo < melhorAtual.custo then
10                 melhorAtual ← solucaoAtual
11             end
12             if melhorAtual ∈ ∅ then
13                 melhorAtual ← solucaoAtual
14                 solucaoAtual ← solucao
15             end
16         end
17     end
18     if novaTrip then
19         tripulacao ← criaTripulacao()
20         solucaoTripulacao ← solucaoTripulacao ∪ {tripulacao}
21     else
22         solucao ← melhorAtual
23     end
24 end

```

O Algoritmo 2 consiste nos seguintes passos:

1. Percorrer todas as tarefas a serem realizadas pelo PPT.
2. Para cada tarefa, verificar se algum tripulante consegue realizá-la. Em caso afirmativo, alocar o tripulante que realiza a tarefa com o menor custo.
3. Se nenhum tripulante puder realizar a tarefa, é adicionado um novo tripulante para executar a tarefa.

Pricing

Segundo Lübbecke (2011) na etapa de *Pricing* do método simplex procuramos uma variável não básica de custo reduzido negativo para entrar na base do *Mestre*. Para conseguir isso na geração de colunas, resolve-se o problema de *Pricing*, que consiste em gerar uma jornada com custo reduzido negativo.

O modelo de *Pricing* considera as seguintes variáveis:

- $x_{i,j}$: define se a tarefa j vai ser executada imediatamente após a tarefa i ,
- e : tempo de hora extra da jornada,
- d : duração da jornada,
- p : 1 se a jornada possui uma dupla pegada, 0 caso contrário,
- v : 1 se a jornada possui uma troca de veículo, 0 caso contrário,
- r : tempo de repouso durante a jornada,
- r^{15} : 1 se o tempo de repouso é 15 min, 0 caso contrário,
- r^{30} : 1 se o tempo de repouso é 30 min, 0 caso contrário,
- ι : tempo de descanso de uma jornada até a outra,
- o : tempo ocioso externo,
- o^i : tempo ocioso interno,
- s : minuto em que inicia a jornada,
- f : minuto em que termina a jornada

Conjuntos:

- T : Conjunto de tarefas

Parâmetros:

- π_i : custo dual da tarefa i dado pelo modelo Mestre,
- α : custo de hora extra,
- β : custo da dupla pegada,
- γ : custo do tempo ocioso,
- σ : custo fixo do tripulante,

- ζ_i : hora de inicio da tarefa i ,
- δ_i : hora de fim da tarefa i ,
- $\rho_{i,j}$: 1 se a tarefa de ir de i para j é uma dupla pegada, 0 caso contrário,
- $\nu_{i,j}$: 1 se o veículo da tarefa i for diferente do veículo da tarefa j , 0 caso contrário,
- $\phi_{i,j}$: tempo ocioso entre a tarefa i e j ,
- d^+ : duração máxima da jornada considerando o tempo de hora extra,
- t^+ limite de duração da jornada,
- t^- : tempo mínimo de descanso entre uma jornada e outra,
- t^{j+} : tempo máximo da jornada sem hora extra,
- t^{e+} : tempo máximo de hora extra,
- m^{15} : tempo mínimo para o tempo de descanso ser 15 minutos,
- m^{30} : tempo mínimo para o tempo de descanso ser 30 minutos,

A formulação matemática para o modelo de *Pricing* é dada pelas Equações (4.10)–(4.31):

$$\text{Minimizar} \quad \sigma + (\alpha e) + (\beta p) + (\gamma o) + \sum_{i \in T} \sum_{j \in T} \pi_i x_{i,j} \quad (4.10)$$

$$\text{Sujeito a} \quad d = \sum_{i \in T} \sum_{j \in T} x_{i,j} (\delta_i - \zeta_i + \phi_{i,j}) \quad (4.11)$$

$$r^{15} \leq 1 - p \quad (4.12)$$

$$r^{30} \leq 1 - p \quad (4.13)$$

$$r^{15} \geq (d - m^{15}) / (d^+) - d^+ p \quad (4.14)$$

$$r^{15} \leq d / m^{15} \quad (4.15)$$

$$r^{30} \geq (d - m^{30}) / (d^+) - d^+ p \quad (4.16)$$

$$r^{30} \leq d / m^{30} \quad (4.17)$$

$$r = 15r^{15} + 15r^{30} \quad (4.18)$$

$$d \leq d^+ + r \quad (4.19)$$

$$e \geq d - t^{j+} - r \quad (4.20)$$

$$p \geq \sum_{i \in T} \sum_{j \in T} \rho_{i,j} x_{i,j} \quad (4.21)$$

$$v \geq \sum_{i \in T} \sum_{j \in T} \nu_{i,j} x_{i,j} \quad (4.22)$$

$$o^i = \sum_{i \in T} \sum_{j \in T} \phi_{i,j} x_{i,j} - r \quad (4.23)$$

$$o \geq d^+ - d \quad (4.24)$$

$$s = \sum_{i \in T} \zeta_i x_{0,i} \quad (4.25)$$

$$f = \sum_{i \in T} \delta_i x_{i,0} \quad (4.26)$$

$$f - s \leq t_+ \quad (4.27)$$

$$\sum_{i \in T} x_{i,0} = 1 \quad (4.28)$$

$$\sum_{i \in T} x_{0,i} = 1 \quad (4.29)$$

$$\sum_{j \in T} x_{i,j} = \sum_{j \in T} x_{j,i} \quad \forall i \in T \quad (4.30)$$

$$p, v, r^{15}, r^{30} \in \{0, 1\} \quad (4.31)$$

Na formulação (4.10)–(4.31), a variável de decisão $x_{i,j}$ é igual a 1 se a tarefa j for executada na jornada imediatamente após a execução da tarefa i , e igual a 0 caso contrário. A expressão (4.10) minimiza o custo total ao se utilizar o arco (i, j) . A Restrição (4.11) calcula a duração da jornada. As Restrições (4.12) e (4.13) limitam o tempo de descanso a 0 quando a jornada é do tipo dupla pegada, enquanto as Restrições (4.14), (4.15), (4.16), (4.17) e (4.18) determinam o tempo de repouso mínimo dos tripulantes durante a jornada. A Restrição (4.19) limita a duração da jornada, enquanto as Restrições (4.20) e (4.21) calculam respectivamente o tempo de hora extra e se a jornada é ou não de dupla pegada. A Restrição (4.22) calcula a quantidade de trocas de veículos, enquanto as Restrições (4.23) e (4.24) computam o tempo ocioso interno e externo respectivamente. As Restrições (4.25) e (4.26) determinam o tempo de início e fim da jornada. A Restrição (4.27) garante que o tempo de descanso da tripulação em casa seja maior do que o tempo mínimo. As Restrições (4.28) e (4.29) garantem que a primeira e a última tarefa do motorista sejam iguais a 1 e, por fim, a Restrição (4.30) garante a conservação do fluxo nos arcos.

Heurística para o *Pricing*

Como a formulação formulação (4.10)–(4.31) pode requer muito tempo de processamento para gerar uma jornada válida, utilizamos uma heurística de caminho mínimo (*Shortest Path* para gerar jornadas mais rapidamente. Segundo Fu et al. (2006), existem várias aplicações no ramo de transporte que requerem o uso de um algoritmo heurístico de caminho mínimo no lugar de modelos exatos padrões, devido à necessidades de resposta rápida.

A definição a seguir foi baseada no artigo de Fu et al. (2006). Uma lista de tarefas T é representada por um grafo $G = (N, A)$ direcionado e acíclico, que consiste em um conjunto de

arcos A , e um conjunto de nós N subdivididos em dois conjuntos, N_1 e N_2 . Seja o número de nós $|N_1| = T$ e $|N_2| = T$. Seja $i, j \in T$, e $a = (i, j) \in A$ um arco direcionado do nó i para o nó j . Tal arco indica que é possível realizar a tarefa j logo após a tarefa i . O custo de um arco a é dado pelo custo dual da tarefa i no modelo Mestre. O problema consiste em encontrar o caminho que tem custo mínimo partindo do nó da garagem até a volta à garagem. O Algoritmo 3 apresenta o pseudo-código desta heurística para gerar uma jornada válida.

Algoritmo 3: Heurística Journey Shortest Path

```

1 grafo ← geraGrafo(duais, tarefas)
2 caminhoMinimo ← geraCaminhoMinimo(grafo)
3 jornada ← ∅
4 custoCaminho ← 0
5 while tarefa ∈ caminhoMinimo do
6   | jornadaAtual ← tarefa ∪ {jornada}
7   | if jornadaValida(jornadaAtual) then
8     | | jornada ← jornadaAtual
9     | end
10  | custoCaminho ← custoCaminho + grafo.custo
11 end
12 custoReduzido ← custoCaminho + jornada.custo if custoReduzido < 0 then
13   | utilizaJornadaNoMestre()
14 end
15 if custoReduzido ≥ 0 then
16   | descartaJornada()
17 end

```

O Algoritmo 3 funciona da seguinte maneira:

- Primeiro é criado o grafo $G = (N, A)$.
- A partir do grafo é resolvido o problema de Shortest Path.
- Percorrer a solução do Shortest Path e Gerar a jornada.
- Ao gerar a jornada é necessário inserir todas as tarefas que não inflija nenhuma das restrições do PPT.
- Caso essa jornada tenha custo reduzido negativo, o modelo de *Pricing* não precisa ser resolvido.
- Caso contrário é resolvido o modelo de *Pricing* dado pela formulação (4.10)–(4.31).

Capítulo 5

Resultados

Neste Capítulo são apresentados os experimentos computacionais realizados com o modelo de Pseudo-Designação para o PPV e com o modelo de cobertura de conjuntos para o PPT. Os modelos foram implementados na linguagem de programação Python, com o framework Python-MIP (Santos et al., 2020) utilizando o Gurobi (Gurobi Optimization, LLC, 2021) como solver. Todos os experimentos foram realizados em um computador com 16GB de RAM e com um processador Intel i7 1.80GHz x 8.

A organização deste Capítulo está conforme descrito a seguir. Inicialmente, na Seção 5.1 é apresentada informações referente as instâncias testadas. Na Seção 5.2 é apresentada os testes do PPV. A Seção 5.3 reporta os testes realizados para o PPT a partir da solução do PPV. Por fim, a Seção 5.4 compara os resultados obtidos pela metodologia proposta com os resultados da literatura.

5.1 Resultados para o Problema de Programação de Veículos

Para realizar a validação da metodologia proposta para tratar o PPV, um conjunto de empresas transportadoras de ônibus da região metropolitana de Belo Horizonte disponibilizou 32 instâncias para o problema, sendo elas problemas reais de uma dada região da cidade.

A Tabela 5.1 apresenta, para cada instância, as seguintes informações: número da instância que está sendo resolvida (Instância), nome da instância (Nome), número de restrições (Restrições) que corresponde ao número de restrições do problema; número de variáveis (Variáveis) que representa o número de variáveis do problema; número de viagens (Viagens) que denota a quantidade de viagens da instância; valor da função objetivo que representa o custo da solução (F.O); Por fim, o tempo necessário para obter a solução ótima para o problema.

Tabela 5.1: Resultado PPV

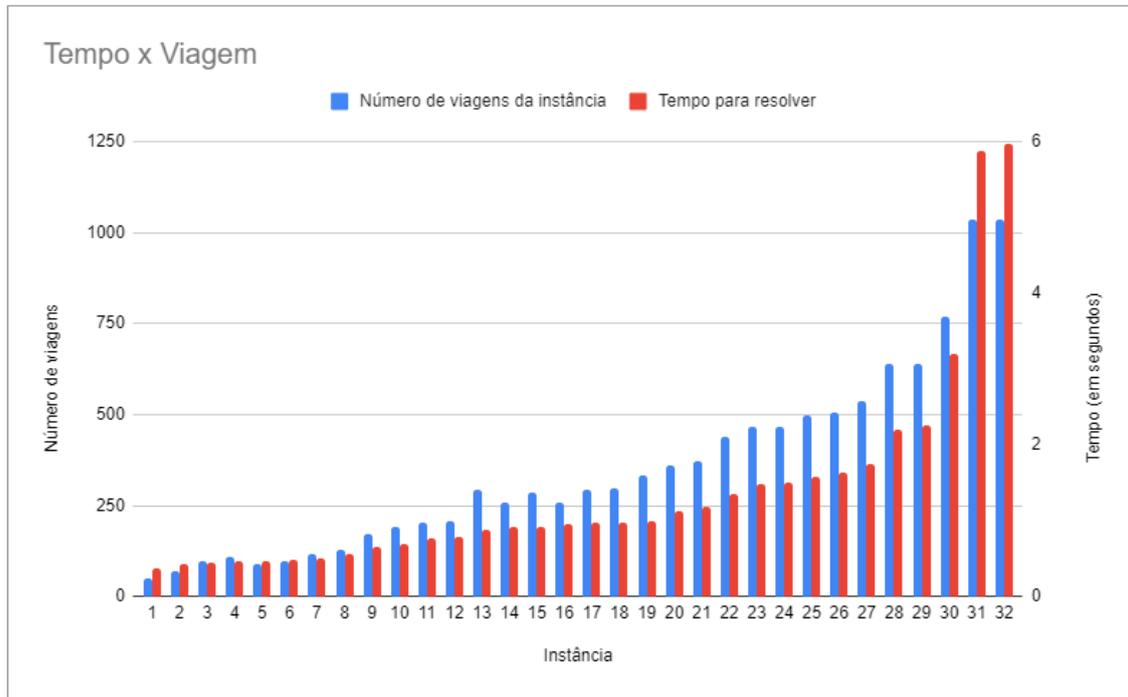
Instância	Nome	Restrições	Variáveis	Não zeros	Viagens	F.O	Tempo(s)
1	DOM-G27	106	1.303	2.606	52	51.982	0,37
2	SAB-G27	140	2.206	4.412	69	72.639	0,43
3	SEX-G27	198	4.255	8.510	98	134.131	0,45
4	DOM-G115	218	5.194	10.388	108	136.308	0,46
5	DOM-G02	182	3.592	7.184	90	113.108	0,47
6	SEG-G27	198	7.274	8.548	98	134.374	0,48
7	DOM-G99	232	6.499	12.998	115	52.143	0,50
8	SAB-G115	262	7.486	14.972	130	147.717	0,57
9	SAB-G02	346	12.797	25.594	172	235.081	0,65
10	SAB-G99	384	17.068	34.136	191	123.030	0,69
11	SEX-G115	408	17.659	35.318	203	295.492	0,77
12	SEG-G115	414	18.173	36.346	206	285.714	0,79
13	SEX-G99	586	38.698	77.396	292	265.338	0,88
14	SEG-G02	522	28.187	56.374	260	410.320	0,91
15	DOM-G46	544	37.372	74.744	286	224.756	0,92
16	SEX-G02	522	28.052	56.104	260	400.291	0,95
17	SEG-G99	588	39.160	78.320	293	255.536	0,97
18	DOM-G48	598	41.658	83.316	298	164.850	0,98
19	DOM-G69	666	50.221	100.442	332	237.391	0,99
20	SAB-G48	720	59.894	119.788	359	205.572	1,13
21	SAB-G46	746	62.792	125.584	372	295.172	1,19
22	SAB-G69	884	88.197	176.394	441	309.095	1,35
23	SEG-G48	938	100.606	201.212	468	327.761	1,48
24	SEX-G48	938	100.396	200.792	468	337.839	1,50
25	SEX-G46	998	109.621	219.242	498	497.952	1,57
26	SEG-G46	1.012	113.319	226.638	505	488.068	1,64
27	DOM-G61	1.078	135.317	270.634	538	247.473	1,74
28	SEX-G69	1.280	180.726	361.452	639	625.752	2,21
29	SEG-G69	1.280	181.915	363.830	639	585.355	2,26
30	SAB-G61	1.540	272.629	545.258	769	459.935	3,21
31	SEX-G61	2.074	487.830	975.660	1.036	745.063	5,87
32	SEG-G61	2.078	491.224	982.448	1.038	734.855	5,97

Fonte: Elaborado pelo autor

A partir da Tabela 5.1 é possível observar que o modelo foi resolvido em apenas 5,97 segundos para resolver a instância com o maior número de viagens, demonstrando que o modelo foi eficiente para todas as 32 instâncias reais.

O gráfico da Figura 5.1 contém um eixo X que representa o número da instância e possui dois eixos Y. O eixo Y à esquerda representa o número de viagens da instância, o eixo Y à direita representa o tempo necessário para resolver a instância. Note que todas as instâncias estão ordenadas com base no tempo de resolução. A partir dessa figura é possível verificar que o tempo necessário para resolver o PPV cresce na maioria das vezes com base no aumento da quantidade de viagens.

Figura 5.1: Gráfico do Tempo x Viagens



Fonte: Elaborado pelo autor

5.2 Resultados para o Problema de Programação da Tripulação

A metodologia sequencial para tratar o PPV e o PPT consiste em primeiro tratar o PPV e utilizar o resultado do PPV para resolver o PPT. Sendo assim, essa seção relata os resultados do PPT a partir das soluções do PPV reportados na seção anterior.

A Tabela 5.2 apresenta, para cada instância, as seguintes informações: identificador do problema que foi resolvido (Problema); função objetivo e tempo do limite inferior (Limite inferior); total de jornadas (Jornadas); total de horas extras (Horas Extras); quantidade de horas ociosas (Ociosos); número de duplas pegadas (Dupla Pegada); função objetivo e tempo da heurística de recobrimento (Heurística de Recobrimento); diferença em porcentagem da solução gerada pela heurística em relação a otimalidade (Gap); finalmente, tempo total para obter a solução em segundos (Tempo Total).

Tabela 5.2: Resultados do PPT na metodologia sequencial

Problema	Limite Inferior				Solução							Tempo Total (s)	Gap(%)	Tempo Total (s)
	F.O.	Tempo (s)	Jornadas	Horas Extras	Ocioso	Dupla Pegada	F.O.	Tempo(s)	Tempo(s)					
SEX-G99	450.878,3	1.300,2	43	1.881	275	22	450.999	0,1	0,03	1.300,3				
SEX-G115	526.829,3	2.056,9	51	2.252	290	13	527.098	0,1	0,05	2.057,0				
DOM-G69	-	3.601,0	45	4.427	0	25	482.708	5,6	-	3.606,6				
SAB-G99	228.521,1	461,3	22	1.839	15	11	233.971	1,3	2,33	462,6				
SEG-G48	-	3.604,6	57	4.949	18	40	613.814	17,9	-	3.622,5				
SEG-G61	-	3.612,6	126	8.056	2.385	100	1.354.609	1,4	-	3.613,9				
DOM-G99	111.851,7	878,4	11	1.017	0	1	114.668	1,0	2,46	879,4				
SAB-G61	-	3.611,1	82	6.785	744	62	885.084	8,6	-	3.619,7				
SEX-G48	-	3.600,8	58	5.177	0	45	627.708	4,2	-	3.604,9				
SAB-G27	140.209,0	115,3	14	1.058	0	2	145.432	0,3	3,59	115,6				
SEG-G69	-	3.602,9	101	6.138	1.772	71	1.078.924	5,6	-	3.608,5				
SAB-G115	284.800,0	169,1	27	2.504	0	10	286.016	0,0	0,43	169,1				
SEG-G99	438.459,0	1.768,7	42	1.728	242	19	438.554	0,1	0,02	1.768,7				
SAB-G02	354.271,8	619,2	34	1.613	164	13	354.416	0,0	0,04	619,2				
DOM-G02	192.080,7	107,3	18	1.424	0	11	192.296	0,0	0,11	107,3				
SEX-G61	-	3.618,9	145	6.322	6.003	111	1.547.891	0,2	-	3.619,1				
SEG-G02	649.767,0	1.810,5	63	1.506	1.899	22	651.123	0,0	0,21	1.810,6				
DOM-G46	415.974,2	3.043,6	39	3.925	0	23	419.500	6,4	0,84	3.050,0				
SAB-G48	438.655,8	3.220,8	41	4.060	39	26	441.879	7,7	0,73	3.228,5				
SEG-G27	225.184,0	148,1	21	1.573	58	15	225.350	0,1	0,07	148,1				
SEX-G02	670.062,5	3.468,3	65	1.906	2.174	20	671.798	0,0	0,26	3.468,4				
SEX-G27	243.397,5	162,3	24	927	43	7	247.951	0,1	1,84	162,4				
DOM-G48	-	3.617,3	33	2.980	23	16	351.543	80,4	-	3.697,7				
SAB-G69	-	3.601,6	60	5.584	56	44	648.792	31,3	-	3.633,0				
SEX-G46	-	3.611,2	82	4.058	229	39	859.861	0,2	-	3.611,4				
DOM-G115	238.826,7	282,9	23	1.578	0	5	239.312	0,2	0,20	283,1				
SAB-G46	-	3.602,0	52	4.997	0	40	563.988	1,9	-	3.604,0				
DOM-G27	102.054,2	19,8	10	835	33	5	106.373	0,0	4,06	19,9				
SEX-G69	-	3.604,3	113	5.624	3.273	81	1.204.369	0,7	-	3.605,0				
SEG-G46	-	3.609,7	79	4.397	701	49	837.689	0,4	-	3.610,0				
DOM-G61	-	3.611,5	54	4.647	90	49	588.078	83,4	-	3.694,8				
SEG-G115	531.335,5	1.472,9	52	1.940	240	6	531.600	0,0	0,05	1.472,9				

Fonte: Elaborado pelo autor

Os resultados da Tabela 5.2 demonstram que a relaxação linear foi resolvida em 19 das 32 instâncias. As soluções obtidas pela heurística de recobrimento são muito próximas do limite inferior nestas instâncias em que a relaxação linear foi resolvida, e tem gap máximo de 4,06% da solução ótima.

5.3 Comparação com resultados da literatura

Nesta seção é realizado um comparativo com os resultados obtidos por Souza (2014). Para realizar esta comparação, foi utilizado um conjunto de dados de entrada idêntico. No entanto, Souza (2014) tratam um problema com algumas restrições diferentes. Assim, para que os resultados sejam comparáveis, foi necessário adaptar o modelo proposto na Seção 4.2.2, responsável por gerar as jornadas. As restrições (4.14) até (4.18) foram removidas, e a Restrição (5.1) foi incluída no modelo.

$$r = 30 - 30p \tag{5.1}$$

Note que a Restrição (5.1) define o tempo de repouso como 0 quando a jornada é dupla pegada, e 30 no caso contrário. Isso ocorre, pois o modelo de Souza (2014) não considera a duração da jornada para definir o tempo de repouso.

A Tabela 5.3 apresenta, para cada instância, as seguintes informações: identificador do problema que foi resolvido (Problema); função objetivo (F.O) e tempo para obter limite inferior, quando disponível; total de jornadas (Jornadas), total de horas extras (Horas Extras); quantidade de horas ociosas (Ocioso); número de duplas pegadas (Dupla Pegada); função objetivo e tempo para a execução da heurística de recobrimento (Heurística de Recobrimento); diferença em porcentagem da solução gerada pela heurística em relação a otimalidade (Gap); por fim, tempo total para obter a solução em segundos (Tempo Total).

Tabela 5.3: Comparativo modelo de cobertura de conjuntos x modelo de Souza (2014)

Resultados da abordagem proposta:

Instância	Limite Inferior		Solução inteira obtida pela Heurística de Recobrimento						Gap(%)	Tempo Total (s)
	F.O.	Tempo (s)	Jornadas	Horas Extras	Ocioso	Dupla Pegada	F.O.	Tempo(s)		
1	46.824,0	7,9	5	217	242	2	52.310	0,02	10,4	7,9
2	61.499,5	16,6	6	424	186	3	63.682	0,03	3,4	16,7
3	77.400,4	28,4	8	597	308	2	83.896	0,23	7,7	28,6
4	90.190,3	47,0	9	772	292	4	95.780	0,37	5,8	47,3
5	100.211,5	89,3	10	614	340	7	106.996	0,56	6,3	89,8

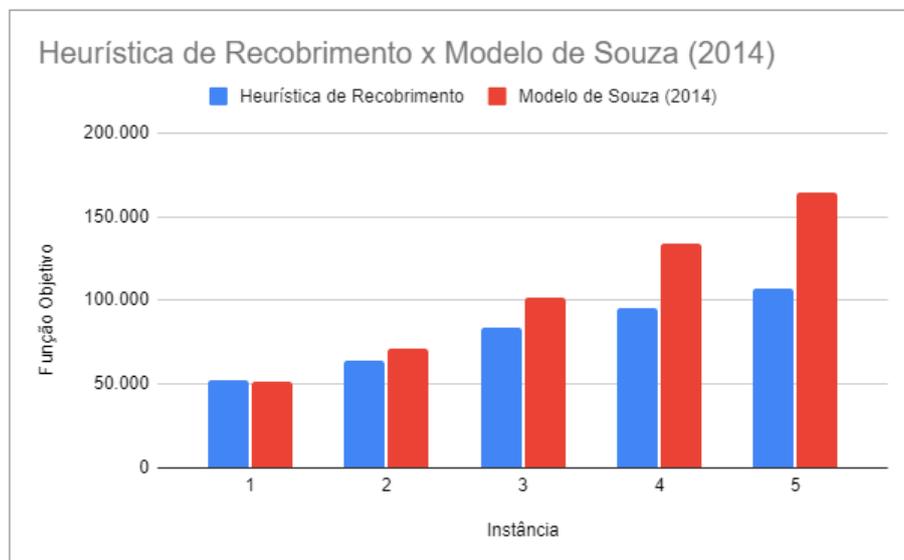
Resultados de Souza (2014):

Instância	Limite Inferior		Solução inteira obtida pelo Modelo de Souza (2014)						Gap(%)	Tempo Total (s)
	F.O.	Tempo (s)	Jornadas	Horas Extras	Ocioso	Dupla Pegada	F.O.	Tempo(s)		
1	51.409,0	-	5	146	225	1	51.409	-	0,0	18.300,0
2	43.794,8	-	7	151	403	1	71.607	-	38,8	28.800,0
3	46.052,0	-	10	98	451	2	102.043	-	54,8	28.800,0
4	63.518,5	-	13	454	1.046	2	134.062	-	52,6	28.800,0
5	53.104,0	-	16	257	1.989	3	164.817	-	67,7	28.800,0

Fonte: Elaborado pelo autor

A partir da Tabela 5.3 é possível observar que o modelo de cobertura de conjuntos consegue em quase todos os testes resultado melhor em relação a função objetivo. Em relação ao tempo de processamento o modelo de cobertura de conjuntos consegue obter o resultado em um tempo muito inferior. Na instância 5, por exemplo, o modelo exato (Souza, 2014) obteve como função objetivo (F.O.) o valor de 165.899 em um tempo de 8h, enquanto o modelo de cobertura de conjuntos obteve como função objetivo o valor de 108.189 em um tempo de 81,05 segundos. Além de resultar em uma solução de melhor qualidade, a abordagem proposta obteve esta solução em 0.31% do tempo utilizado pela abordagem de Souza (2014) na maior instância.

Figura 5.2: Gráfico comparativo entre a função objetivo do modelo de Souza (2014) e da Heurística de Recobrimento



Fonte: Elaborado pelo autor

O gráfico da Figura 5.2 contém um eixo X que representa o número da instância e possui um eixo Y que corresponde ao valor da função objetivo para aquela instância. A partir dessa figura é possível verificar que ao aumentar o tamanho do problema a diferença do custo do modelo de Souza (2014) para o modelo de cobertura de conjuntos aumenta consideravelmente. Além disso, o modelo de cobertura de conjuntos obteve melhor resultado em todas as instâncias.

Capítulo 6

Conclusão e Trabalhos Futuros

Neste trabalho foi implementado um modelo de pseudo-designação para o Problema de Programação de Veículos (PPV), que foi avaliado considerando um cenário real envolvendo um conjunto de empresas transportadoras da região metropolitana da cidade de Belo Horizonte. O modelo de pseudo-designação é capaz de resolver diversas instâncias em um tempo considerado muito pequeno, com a maior das instâncias contendo 1038 viagens sendo resolvida em apenas 5,97 segundos.

A partir da solução do PPV foi abordado o Problema de Programação da Tripulação (PPT), que consiste em escalonar os tripulantes, agrupando as tarefas em jornadas. Para esse escalonamento da tripulação foi implementado um modelo de cobertura de conjuntos utilizando uma estratégia de geração de colunas e uma heurística de recobrimento para obter soluções inteira. Como resultado dessa metodologia, a relaxação linear foi resolvida para 19 das 32 instâncias dentro do tempo limite especificado, gerando limites inferiores de alta qualidade. Além disso, as soluções obtidas pela heurística de recobrimento estão muito próximas da otimalidade nas instâncias em que a relaxação linear foi resolvida.

Dentre as contribuições deste trabalho, destacamos justamente os limites inferiores obtidos a partir da resolução da relaxação linear do modelo de cobertura de conjuntos. Tais limites poderão ser utilizados para aferir a qualidade de soluções geradas por heurísticas e demais metodologias para o PPT.

Ao realizar um comparativo com um modelo exato de um outro trabalho da literatura, observa-se que para quase todas as instâncias o modelo de cobertura de conjuntos obteve resultados com custos melhores e em um tempo muito inferior. Enquanto o modelo da literatura obteve cada resultado após cerca de 6 a 8 horas de processamento, o modelo de cobertura de conjuntos demorou apenas 89,8 segundos na maior instância.

Como trabalhos futuros, indicamos: (i) a implementação do *branch-and-price* para resolução do modelo de cobertura de conjuntos; (ii) a avaliação de outras estratégias heurísticas para geração de jornadas com custo reduzido negativo; e (iii) a utilização de meta-heurísticas mais refinadas para obter a solução inicial que é passada ao modelo de cobertura de conjuntos.

Referências Bibliográficas

- Chen, C. R. e Hou, C. Y. (1988). A new algorithm for cmos gate matrix layout. In *Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on*, pp. 138–141. IEEE.
- Ciancio, C.; Laganà, D.; Musmanno, R. e Santoro, F. (2018). An integrated algorithm for shift scheduling problems for local public transport companies. *Omega*, 75:139 – 153.
- Constantino, A.; Calvi, R.; Mendonça, C.; De Araujo, S.; Landa-Silva, D. e Flausino dos Santos, A. (2017). Solving a large real-world bus driver scheduling problem with a multi-assignment based heuristic. *JOURNAL OF UNIVERSAL COMPUTER SCIENCE*, 23:479–504.
- Davis, S. e Impagliazzo, R. (2004). Models of greedy algorithms for graph problems. *Algorithmica*, 54:269–317.
- De Leone, R.; Festa, P. e Marchitto, E. (2011). A bus driver scheduling problem: a new mathematical model and a grasp approximate solution. *J. Heuristics*, 17:441–466.
- Dell’Amico, M.; Fischetti, M. e Toth, P. (1993). Heuristic algorithms for the multiple depot vehicle scheduling problem. *Management Science*, 39:115–125.
- Desrochers, M. e Soumis, F. (1989). A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13.
- Desrosiers, J. e Lübbecke, M. E. (2005). *A Primer in Column Generation*, pp. 1–32. Springer US, Boston, MA.
- Freling, R.; Wagelmans, A. P. M. e Paixão, J. M. P. (1999). An overview of models and techniques for integrating vehicle and crew scheduling. In Wilson, N. H. M., editor, *Computer-Aided Transit Scheduling*, pp. 441–460, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Fu, L.; Sun, D. e Rilett, L. (2006). Heuristic shortest path algorithms for transportation applications: State of the art. *Computers Operations Research*, 33(11):3324–3343. Part Special Issue: Operations Research and Data Mining.

- Gurobi Optimization, LLC (2021). Gurobi Optimizer Reference Manual.
- Hassold, S. e Ceder, A. A. (2014). Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, 67:129 – 143.
- Ibarra-Rojas, O. J.; Giesen, R. e Rios-Solis, Y. A. (2014). An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. *Transportation Research Part B: Methodological*, 70:35 – 46.
- Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pp. 85–103. Springer US, Boston, MA.
- Lübbecke, M. (2011). *Column Generation*.
- Mauri, G.; Antonio, L. e Lorena, L. (2004). Método interativo para resolução do problema de escalonamento de tripulações.
- Nakao, C. A. e Silva, G. P. (2017). Otimização da tabela de horários de ônibus urbano. *XLIX Simpósio Brasileiro de Pesquisa Operacional Blumenau-SC*, 1(3):1643–1654.
- Proteste (2015). Transporte público deixa a desejar. *Proteste*.
- Santos, H. G.; Toffolo, T.; Tommy; Spoorendonk, S.; Larsen, P.; Jurasovic, M.; Marvin, A.; Bonelli, F.; Larchenko, I.; Vigerske, S.; Chen, X.; Iuvik; tekgrizzly e A, R. P. (2020). coin-or/python-mip: 1.11.0.
- Shijun, C. (2013). An improved column generation algorithm for crew scheduling problems. *Journal of Information and Computational Science*, 10:175–183.
- Silva, G. e Gualda, N. (2000). Um algoritmo de geração de arcos para o problema de programação de veículos. *TRANSPORTES*, 8.
- Silva, G. P. e Cunha, C. B. (2010). Uso da técnica de busca em vizinhança de grande porte para a programação da escala de motoristas de ônibus urbano. *Transportes*, 18(2).
- Souza, D. S. (2014). Uma abordagem híbrida para resolver o problema da escala de motoristas de ônibus urbano. *RIUFOP*, 8.
- Visentini, M.; Araújo, O.; Borenstein, D. e Neto, A. (2017). Resolução integrada de problemas do planejamento do transporte público: foco na tabela de horários e no escalonamento de veículos com frota heterogênea. *Revista de Administração da UFSM*, 10:828.
- Zuo, X.; Chen, C.; Tan, W. e Zhou, M. (2015). Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 16:1030 – 1041.